

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації і управління

До захисту допущено:

В.о. завідувача кафедри

_____ *Олександр ПАВЛОВ*

(підпис)

(вл.ім'я, прізвище)

“ ___ ” _____ 2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»

спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: *«Інформаційна система підтримки діяльності балетної школи»*

(комплексна тема)

Загальна частина

Виконали:

студент IV курсу, групи ІС-62

Халін Олег Ігорович

(прізвище, ім'я, по батькові)

(підпис)

студент IV курсу, групи ІС-62

Кравчук Олександр Ігорович

(прізвище, ім'я, по батькові)

(підпис)

студент IV курсу, групи ІС-62

Щеголь Данило Андрійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник

ст. викл. Мажара Ольга Олександрівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

**Консультант з
графічної
документації**

доц., к.т.н., доц. Новінський Валерій Петрович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент

доц., к.т.н., доц. Михайлова Ірина Юріївна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Засвідчуємо, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студенти _____

(підпис)

(підпис)

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки

(повна назва)

Кафедра автоматизованих систем обробки інформації і управління

(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ

(підпис)

(вл.ім'я, прізвище)

“ ___ ” _____ 2020 р.

**ЗАВДАННЯ
на дипломний проєкт студенту**

Халіну Олєгу Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема проєкту «Інформаційна система підтримки діяльності
балетної школи» (комплексна тема)

керівник проєкту Мажара Ольга Олександрівна, ст. викладач

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7” травня 2020 р. №1081-с

2. Термін подання студентом проєкту “04” червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна варіантів використання

2. Схема бази даних

3. Схема структурна класів програмного забезпечення

4. Схема структурна послідовності

5. Схема структурна компонентів

6. Схема структурна розгортання програмного забезпечення

7. Креслення вигляду екранних форм

6. Консультанти розділів проєкту

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| | | | |

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

| № з/п | Назва етапів виконання дипломного проєкту | Термін виконання етапів проєкту | Примітка |
|-------|--|---------------------------------|----------|
| 1. | Вивчення рекомендованої літератури | 14.04.2020 | |
| 2. | Аналіз існуючих методів розв'язання задачі | 15.04.2020 | |
| 3. | Постановка та формалізація задачі | 16.04.2020 | |
| 4. | Розробка інформаційного забезпечення | 17.05.2020 | |
| 5. | Алгоритмізація задачі | 19.04.2020 | |
| 6. | Обґрунтування використовуваних технічних засобів | 20.04.2020 | |
| 7. | Розробка програмного забезпечення | 05.05.2020 | |
| 8. | Налагодження програми | 08.05.2020 | |
| 9. | Виконання графічних документів | 10.05.2020 | |
| 10. | Оформлення пояснювальної записки | 13.05.2020 | |
| 11. | Подання ДП на попередній захист | 15.05.2020 | |
| 12. | Подання ДП на основний захист | 08.06.2020 | |
| 13. | Подання ДП рецензенту | 09.06.2020 | |

Студенти

Олег ХАЛІН
Олександр КРАВЧУК
Данило ЩЕГОЛЬ

Керівник

Ольга МАЖАРА

**Пояснювальна записка
до дипломного проєкту**

Інформаційна система підтримки діяльності балетної школи.

(комплексна тема)

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проєкту складається з трьох розділів, містить 4 рисунків, 2 таблиці, 2 додатка, 20 джерел.

Призначенням дипломного проєкту є створення інформаційної системи підтримки діяльності балетної школи. Її реалізація у вигляді веб-застосунку, розміщеного на сервері та мобільного додатку, розміщеного на AppStore[1].

На сьогодні представлення компанії в інтернет просторі є необхідною умовою для її сприйняття користувачами. Найпоширенішою формою присутності у мережі є персоніфікована веб-сторінка. Веб-застосунок слугує не лише платформою для розміщення інформації, але й системою, що допомагає автоматизувати бізнес-процеси. Існує тенденція до створення окремих веб- та моб-додатків поряд з веб-сайтами (наприклад Instagram[2], Facebook[3], Privat24[4]). Актуальною є задача розробки персоніфікованих інформаційних систем для представників малого бізнесу та стартапів.

У розділі інформаційного забезпечення задаються необхідні вимоги до даних, якими оперує система. У нашому випадку це файли бази даних.

При виборі мов програмування один з основних критеріїв була швидкість розробки та простота підтримки. Для серверної частини була вибрана мова Python[5] з простим синтаксисом і великою спільнотою розробників. Для веб застосування використовується TypeScript[6] - мова, що розширює можливості JavaScript[7], завдяки системі типів застосунків має вищу надійність. Нативний мобільний застосунок розроблено на мові Swift[8], яка є новішою і має ширші можливості за Objective-c[9].

| | | | | | | | | |
|----------------|------------------------|-----------------|---------------|-------------|--|--|-------------|----------------|
| | | | | | ДП ІС-6226.00.000 ПЗ | | | |
| | | <i>Прізвище</i> | <i>Підпис</i> | <i>Дата</i> | | | | |
| <i>Розроб.</i> | <i>Халін О.І.</i> | | | | Інформаційна система підтримки діяльності балетної школи (комплексна тема) | <i>Літ.</i> | <i>Арк.</i> | <i>Аркушів</i> |
| | | | | | | | 2 | 36 |
| <i>Керівн.</i> | <i>Мажара О.О.</i> | | | | | КПІ ім. Ігоря Сікорського Каф. АСОІУ ар. ІС-62 | | |
| <i>Н. кон.</i> | <i>Новінський В.П.</i> | | | | | | | |
| <i>Зате.</i> | <i>Павлов О.А.</i> | | | | | | | |

Розділ математичного забезпечення присвячений вирішенню задачі теорії розкладів та обґрунтуванню використаних методів розв'язку. У ньому пояснюється обраний метод та доводиться його доцільність.

БАЛЕТНА ШКОЛА, ІНФОРМАЦІЙНИЙ ДОДАТОК, ВЕБ-ЗАСТОСУНОК, МОБІЛЬНИЙ ДОДАТОК

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 3 |

ABSTRACT

Structure and scope of the research work. There are three chapters, 4 images, 2 tables, 2 supplements, 20 sources in this explanatory note.

The purpose of the graduation project is to create an information system in order to support the activities of the ballet school. Its implementation in a web application form that will be hosted on a server and a mobile application that will be hosted on the App Store[1].

In nowadays' society, the company's representation in the Internet is a necessary condition for its perception by users. The most common form of online presence is a personalized web page. A web application serves not only as a platform for posting information, but also as a system that helps automate business processes. There is a tendency to create separate web and mobile applications along with websites (eg Instagram[2], Facebook[3], Privat24[4]). The task of developing personalized information systems for small businesses and startups is relevant.

The information support section defines the necessary requirements for the data that the system operates. In our case, these are database files.

When choosing programming languages, one of the main criteria was the speed of development and ease of support. Python[5] with a simple syntax and a large community of developers was chosen for the server part. For web applications, TypeScript[6] is used - a language that extends the capabilities of JavaScript[7], thanks to the system of application types has a higher reliability. The native mobile application is developed in the Swift[8] language, which is newer and has more features than Objective-c[9].

The section of mathematical support is devoted to solving the problem of schedule theory and substantiation of the used methods of solution. It explains the chosen method and proves its feasibility.

BALLET SCHOOL, INFORMATION SYSTEM, WEB APPLICATION,
MOBILE APPLICATION

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 4 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ЗМІСТ

| | |
|---|-----------|
| ВСТУП | 7 |
| 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ | 9 |
| 1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА..... | 9 |
| 1.1.1 <i>Опис процесу діяльності</i> | 9 |
| 1.1.2 <i>Опис функціональної моделі</i> | 10 |
| 1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ..... | 13 |
| 1.2.1 <i>BasePlan</i> | 13 |
| 1.2.2 <i>Отмечалка</i> | 14 |
| 1.2.3 <i>Mobifitness</i> | 14 |
| 1.2.4 <i>Параплан</i> | 14 |
| 1.3 ПОСТАНОВКА ЗАДАЧІ | 16 |
| 1.3.1 <i>Призначення розробки</i> | 16 |
| 1.3.2 <i>Цілі та задачі розробки</i> | 16 |
| Висновок до розділу..... | 18 |
| 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ | 19 |
| 2.1 ВХІДНІ ДАНІ | 19 |
| 2.3 СТРУКТУРА МАСИВІВ ІНФОРМАЦІЇ..... | 21 |
| Висновок до розділу..... | 26 |
| 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ | 27 |
| 3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ | 27 |
| 3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ..... | 27 |
| 3.3 ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ’ЯЗАННЯ | 29 |
| 3.4 ОПИС МЕТОДІВ РОЗВ’ЯЗАННЯ | 31 |
| Висновок до розділу..... | 33 |
| ЗАГАЛЬНІ ВИСНОВКИ | 34 |
| ПЕРЕЛІК ПОСИЛАНЬ | 35 |

ДП ІС-6226.00.000 ПЗ

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 6 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ВСТУП

У сучасному світі для залучення клієнтів малий бізнес повинен бути представлений у світовій мережі Інтернет. Бізнес повинен бути уніфікованим, персоніфікованим, ефективно використовувати ресурс, щоб виділятися з поміж конкурентів та нести цінність для клієнта. Для ефективної комунікації між бізнесом та його клієнтами, а також для підтримки бізнес — процесів, використовуються кастомізовані CRM/ERP системи. Слід враховувати, що сучасні CRM/ERP системи мають доволі широку функціональність. Освоєння, налаштування, підтримка таких системи потребує багато зусиль, часу, а також спеціальних знань. Тому створення системи, яка відповідає потребам конкретного замовника і містить лише необхідний функціонал є актуальною задачею.

Переважає більшість CRM/ERP орієнтовані на сам бізнес, а не на його клієнтів. В той же час клієнтоорієнтований підхід вимагає створення систем взаємодії бізнесу з користувачами. Особливо чітко таку тенденцію можна прослідкувати для нішового бізнесу, наприклад, балетної школи для дорослих.

Наявність мобільного додатку, який можна використовувати без доступу або при обмеженому доступі до мереж Інтернет — це перевага, яку може запропонувати далеко не кожна система взаємодії з клієнтами. Тому створення не лише веб-версії системи, але й мобільного застосунку підвищує конкурентоспроможність бізнесу та має практичне значення.

Дипломний проект присвячено розробці системи підтримки діяльності балетної школи. Переважає більшість CRM/ERP балетних шкіл є продуктом російського виробництва. Локалізацій таких систем не підтримується виробником, що унеможлиблює їх використання в Україні. Тому в роботі запропоновано спеціалізоване рішення для мережі балетних шкіл, яка працює у місті Києві.

Інформаційна система реалізована у вигляді веб-сайту та мобільного застосунку.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 7 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Систему реалізовано для трьох типів користувачів: учня, тренера, адміністратора. Для тренерів та учнів система забезпечує інформаційну підтримку, автоматизує процес складання розкладу, засоби зворотного зв'язку та обслуговування абонементів. Адміністратор має доступ до розширеного функціоналу, зокрема, до статистики відвідуваності занять, навантаження на педагогів, завантаженість учбових залів.

Практичне значення одержаних результатів. Розроблено інформаційну систему підтримки діяльності балетної школи та прийнято її в експлуатацію у мережу балетних шкіл «MaryBallet»[11].

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 8 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

1.1.1 Опис процесу діяльності

Ми розробляємо інформаційну систему підтримки процесів діяльності балетної школи, її реалізацію у вигляді веб та мобільного додатку. Однак ця система може використовуватись будь-якою навчальною організацією: починаючи з курсів і персональних занять та закінчуючи великими школами та навіть університетами. Вона є не просто корисною, а необхідною для значної кількості бізнесів, які переростають з маленьких персональних фірм до великих корпорацій, що обслуговують значну кількість людей.

На ринку існує безліч різноманітних CRM/ERP систем, якими користуються більшість розвинутих компаній, однак ми націлились на необхідні користувачу задачі, які не були реалізовані в більшості варіантів. А саме: автоматизація розкладу занять, що включають в себе різні рівні підготовки клієнтів, найбільш зручну для них локацію, час та певного тренера; вибір необхідного для клієнта абонементу, зручний перенос і подовження занять, що може відбуватися без витрачених ресурсів бізнесу; оплата онлайн; можливість тренера відмічати відвідуваність клієнтів і додавати коментарі, що допомагає йому стежити за необхідною навантажкою кожного клієнта індивідуально; автоматизація так званих «розумних повідомлень», що реалізуються за допомогою емейла та смс. Додаємо до цього всього зручний інтерфейс і кросплатформеність, щоб тренер міг це зробити у мобільному додатку в режимі офлайн. В результаті матимемо зручну систему, якою зможуть користуватися більшість шкіл і навчальних організацій.

Ми створюємо продукт персонально за заданими умовами і можемо в будь-який час його кастомізувати під потреби користувача. Для нас це не просто теорія або тестовий проект, для нас це реальна розробка, яка буде неодмінно в майбутньому приносити користь. Її функціонал буде економити

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 9 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

час та гроші власнику, а детально розроблений дизайн і юзер флоу клієнтської частини застосунку дозволить користувачам легко і інтуїтивно зрозуміло виконати всі необхідні задачі та дізнатися всю необхідну інформацію.

1.1.2 Опис функціональної моделі

Програмний продукт буде реалізовано у вигляді веб-застосунку, який буде розміщений на сервері. Користувачі матимуть змогу взаємодіяти з інтерфейсом застосунку за допомогою браузера.

У системі присутні 4 типи акторів:

1. Адміністратор;
2. Анонімний (неавторизований) користувач;
3. Учень;
4. Викладач.

Для кожного користувача в базі даних визначений його тип. Тому після авторизації анонімний користувач змінить свій тип на той, який визначений у базі даних.

У таблиці 1.1 описані актори, варіанти використання та їх описи дій.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 10 |

Таблиця 1.1 – Актори, варіанти використання та їх описи дій

| Актор | Варіант використання | Опис дії варіанта використання |
|---------------|--|--|
| Адміністратор | Додавання/редагування/видалення ресурсів сайту | <p>Адміністратор переходить в пункт ресурсу в адмінці і має можливість проглянути список цих ресурсів. Якщо ресурс підтримує операцію додавання, то на сторінці буде кнопка “Создать”, після натискання якої відкриється форма додавання ресурсу. Якщо ресурс підтримує видалення, то на сторінці редагування ресурсу буде присутня кнопка “Удалить”.</p> <p>Кожен ресурс має свою унікальну форму редагування/додавання, а також набір операцій.</p> <p>Ресурси:</p> <ul style="list-style-type: none"> - Зали (додавання, редагування, видалення); - Сторінки (редагування); - Шаблони уроку (додавання, редагування, видалення); - Локації (додавання, редагування, видалення); - Користувачі (додавання, редагування, видалення); - Абонемент (додавання, редагування, видалення); - Відвідування (редагування); - Історія оплат (лише перегляд); - Партнери (додавання, редагування, видалення); - Переноси уроків (додавання, редагування, видалення); - Заморозки абонементів (додавання, редагування, видалення). |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Продовження таблиці 1.1

| | | |
|----------------------|--------------------------|--|
| Анонімний користувач | Вхід | Користувач може авторизуватись заповнивши свій номер телефону та пароль, якщо дані будуть вказані вірно користувач стане авторизованим |
| Анонімний користувач | Реєстрація | Користувач може зареєструватись увівши своє ім'я та номер телефону, після введення користувачу на вказаний номер телефону прийде SMS з паролем для входу |
| Учень | Покупка абонементу | Учень в своєму особистому кабінеті може обрати абонемент по якому він бажає відвідувати заняття та придбати його. |
| Учень | Оплата | Учень в особистому кабінеті може оплатити абонемент онлайн. Система відмітить оплату, адміністратори та викладачі зможуть побачити, що користувач успішно оплатив урок. |
| Учень | Перенесення уроку | Якщо куплений користувачем передбачає перенесення уроків, то користувач має змогу перенести заняття на інший день, для цього йому потрібно обрати день у який проводиться заняття і обрати опцію перенесення |
| Учень | Заморока абонементу | Якщо куплений користувачем абонемент передбачає перенесення уроків, то користувач має змогу заморозити відвідування на строк, який дозволений абонементом. Система автоматично перенесе всі заняття. |
| Учень | Перегляд розкладу занять | Учень в своєму особистому кабінеті може переглянути його розклад, визначити коли буде наступний урок. |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Продовження таблиці 1.1

| | | |
|----------|---------------------------|--|
| Викладач | Відмічання оплати заняття | Викладач може отримати оплату безпосереднього після уроку і має можливість відмітити урок оплаченим для користувача |
| Викладач | Відмічання відвідування | Викладач може відмітити відвідування користувачем заняття, поставивши відповідну галочку навпроти користувача. |
| Викладач | Додавання коментарів | Викладач може додати коментар будь-якому учню на занятті, записавши його в необхідний текстовий блок біля користувача. |
| Викладач | Перегляд розкладу занять | Викладач в своєму особистому кабінеті може переглянути його розклад, визначити коли буде наступний урок. |

1.2 Огляд наявних аналогів

Для вирішення задач підтримки функціонування бізнесу, подібного до балетної школи, існують CRM/ERP системи. Однак здебільшого вони містять надлишковий функціонал, який універсальний для різних видів діяльності і не фокусуються на необхідних конкретному виду бізнесу задачах. Крім того, на забезпечення функціоналу щодо організації занять та оплати впливає також географічний регіон. Системи, створені для ринків Америки та Західної Європи складно адаптуються для використання в Україні.

1.2.1 BasePlan

Система розроблена російською компанією. Має в своєму функціоналі базу клієнтів, планувальник і рекламу заходів, модуль постановки задач, контроль за ресурсами компанії, а також емейл і смс повідомлення клієнтам.[12]

Однак даний сервіс не дає можливості використовувати різні способи оплати і можливості визначати різні види абонементів, відмічати присутніх і

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 13 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

жодних функцій, що дають можливість клієнтам взаємодіяти з системою. Також дана система має незручний інтерфейс, що відіграє велику роль на зручності використання.

1.2.2 Отмечалка

Також російське виробництво. Має більш великий функціонал ніж попередні приклади, а саме: робота з усією клієнтською базою, відмічання відвідування, налаштування розкладу, створення абонементів, ведення фінансів.[13]

Дана система доступна тільки у десктопному вигляді, не підтримує емейл і смс розсилки і найголовніше не має жодного представлення для клієнтів. Також не досить зручний і інтуїтивно зрозумілий функціонал.

1.2.3 Mobifitness

Дана система також російського виробника. І має наступний функціонал: облікова система для легкого управління, мобільний додаток для легкої комунікації з клієнтом, онлайн запис на заняття, онлайн оплату.[14]

Даний представник вже має декілька функцій, що дають можливість клієнтам взаємодіяти з системою і емейл/смс повідомлень. Вона має зручніший інтерфейс ніж попередній представник, однак він все ж не є дуже зрозумілим для легкого користування.

1.2.4 Параплан

Найкращий аналог на ринку СНД, однак також є продуктом російського виробництва. Він має досить великий функціонал, доступний і зрозумілий інтерфейс, мобільну версію. В своєму функціоналі сервіс надає наступні можливості: облік клієнтів компанії, додавання і редагування абонементів, планувальник задач для персоналу, аналітика і розрахунок зарплат, емейл і смс повідомлення.[15]

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 14 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Однак і ця система не має необхідного функціоналу, який був би необхідним для користувачів, які є відвідувачем школи. Не дозволяє тренерам школи контролювати відвідуваність і додавати коментарі щодо кожного клієнта.

Отже, як ми бачимо з аналізованих аналогів тільки один з представлених сервісів пропонує гарне рішення для персонального кабінету студентів школи. Однак сайт системи не надає жодної інформації чи демо версії, яка дозволяє переглянути усі функції і, що також є ключовим фактором, ціну.

Також слід відмітити, що більшість відомих CRM систем, таких як amocrm[16], hubspot[17], notion[18], bitrix24[19]... є досить великим продуктами, вони намагаються залучити до свого користування якомога більше користувачів тому додають все більше функціонала. В даному дослідженні як аналоги розглядалися лише вузьконаправленні системи для фітнесу, балету, спортивних секцій тощо.

Усі розглянуті системи розроблені в Росії. Для отримання більш детальної інформації, було проведено опитування служб підтримки, яке виявило відсутність забезпечення платіжних систем України, тому функції прийому платежу для українців недоступні. Також це може означати і те, що при використанні даних сервісів можуть виникати збої, так як більшість російських серверів заборонені на території України, а також, що гарного вітчизняного виробника не існує на ринку.

Особливістю даної роботи є те, що в ній створюється персональний продукт за заданими умовами і передбаченням подальшої переробки та кастомізації. Розробка виконувалась в співпраці з балетною школою та вже впроваджена в використання. Її функціонал дозволяє економити ресурси пам'яті та часу. Детально розроблений дизайн і юзер флоу клієнтської частини застосунку дозволяє користувачам легко і інтуїтивно виконати всі необхідні задачі та дізнатися всю необхідну інформацію.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 15 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

1.3 Постановка задачі

1.3.1 Призначення розробки

Розроблювана система призначена для автоматизованої роботи балетної школи. Автоматизація процесів у бізнесі надає купу переваг як підприємцям, так і клієнтам, а саме: значно менші фінансові витрати у довгостроковій перспективі, більша стресостійкість через меншу наявність людського фактору, економія часу та зручніший контроль даних. Окрім того у теперішній час інформаційні системи є невід'ємною частиною корпоративної культури та найціннішою складовою маркетингу.

Інформаційна система повинна бути:

- надійною — для того, щоб усі процеси відбувались без перешкод;
- адаптивною — тобто бути зручною для майбутнього розширення функціоналу або змін у дизайні;
- кросплатформною — для того щоб не обмежувати цільову аудиторію користувачів у рамках певної платформи;
- інтуїтивною у користуванні;
- функціонально повною, але не перенасиченою зайвими функціями.

Даний проект може бути використаний у якості прикладу клієнт-серверного застосунку у будь-якій предметній області. За мету було взято створити як можна гнучкішу архітектуру для подальшого використання у інших кросплатформних систем.

1.3.2 Цілі та задачі розробки

Ціль розробки – створення програмного продукту, який автоматизує створення та зміну індивідуальних розкладів відвідувачів балетної школи,

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 16 |

дозволяє здійснювати електронні платежі для оплати занять, спостерігати за новинами школи та полегшить взаємодію танцюрів та тренерів.

Для досягнення поставленої цілі необхідно виконати наступні задачі:

- розробити дизайн для десктопної та мобільної версій застосунку;
- спроектувати архітектуру серверної частини додатку;
- спроектувати архітектури клієнтських частин додатку для кожної платформи, намагаючись їх якомога більше уніфікувати;
- розробити усі необхідні екрани застосунку згідно дизайну;
- інтегрувати серверну частину у клієнтські частини застосунку;
- розробити платформиорієнтований функціонал, тобто функціонал, який буде унікальним у рамках певної платформи;
- провести тестування додатку на предмет наявності логічних та програмних помилок.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 17 |

Висновок до розділу

У цьому розділі описано предметне середовище, розглянуто наявні аналоги розробки, описано призначення розробки, обґрунтовано цілі та задачі розробки. Розділ містить відомості про типи користувачів в межах розробленого програмного продукту та функції, які необхідно реалізувати.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 18 |

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані отримуються з наступних джерел:

- Користувачі;
- Адміністратори;
- Платіжні системи.

Дані від користувачів надходять з форм в особистих кабінетах користувача, для адмінів це форми в адмінці.

В панелі адміністрування адміністратор може додавати, видаляти та модифікувати наступні дані:

1. Міста;
2. Вулиці;
3. Зали;
4. Користувачі;
5. Уроки;
6. Розклад уроків;
7. Підписки на уроки;
8. Куплені користувачами підписки.

Для реєстрації в системі користувач повинен заповнити форму реєстрації, яка включає у себе наступні поля:

1. Ім'я;
2. Номер телефон;
3. Код з смс, яке прийде на номер телефону.

Проведення платежу при покупці абонементу відбувається на сторонньому сервісі. Таким чином економиться час на розробку, а також

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 19 |

знімається відповідальність за безпеку коштів користувачів. Програмний продукт лише отримує підтвердження про здійснення платежу і зараховує покупку підписки.

Вихідні дані

Вихідними даними застосування є інформація, яку користувачі бачать на сторінках:

1. Розклад уроків;
2. Доступні підписки;
3. Найближче заняття;
4. Детальна інформація про заняття;
5. Статистика відвідування заняття.

Приклади наведено у графічних матеріалах у частині звітів екрану.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 20 |

2.2 Структура масивів інформації

Для розробки застосунку було використано базу даних PostgreSQL[20]

У таблиці 2.1 наведено перелік таблиць, які описують бізнес логіку застосунку.

Таблиця 2.1 – Перелік таблиць, які описують бізнес логіку застосунку.

| Таблиця | Опис таблиці | Поля | Тип | Опис поля |
|-----------|--------------|--------------|---------|--------------------------------|
| city | Місто | name | varchar | Ім'я міста |
| street | Вулиця | name | varchar | Ім'я вулиці |
| hall | Зал | id | serial | Ідентифікатор |
| | | city | varchar | Місто у якому знаходиться зал |
| | | street | varchar | Вулиця на якій знаходиться зал |
| | | number | varchar | Номер вулиці |
| auth_user | Користувач | id | serial | Ідентифікатор |
| | | email | varchar | Емейл |
| | | first_name | varchar | Ім'я |
| | | last_name | varchar | Прізвище |
| | | phone_number | varchar | Номер телефону |
| | | password | binary | Хешований пароль |
| | | avatar_url | varchar | URL на аватар користувача |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Продовження таблиці 2.1

| | | | | |
|--------|------|----------------------|------------|--|
| | | is_coach | boolean | Чи є користувач тренером |
| | | notes | varchar | Додаткові записи |
| | | is_staff | boolean | Чи є користувач персоналом (чи має доступ до панелі адміністрування) |
| lesson | Урок | id | serial | Ідентифікатор |
| | | lesson_template_id | integer | Шаблон за яким згенерований урок |
| | | hall_id | integer | Зал у якому проводиться заняття |
| | | booking_places_count | integer | Кількість доступних місць на уроці |
| | | trial_places_count | integer | Кількість доступних місць для пробного заняття |
| | | coach_id | integer | Тренер, який проводить заняття |
| | | time | timestamp | Час коли проведитиметься урок |
| | | duration | integer | Тривалість уроку у хвиликах |
| | | skill_level | skilllevel | Рівень підготовки |
| | | type | lesstypе | Тип уроку |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Продовження таблиці 2.1

| | | | | |
|-----------------|---|----------------------|------------|--|
| lesson_template | Шаблон уроку. Те що доступно для запису | id | serial | Ідентифікатор |
| | | hall_id | integer | Зал у якому проводиться заняття |
| | | weekday | integer | День тижня, коли проводиться заняття |
| | | time | time | Час коли проведитиметься урок |
| | | coach_id | integer | Тренер, який проводить заняття |
| | | booking_places_count | integer | Кількість доступних місць на уроці |
| | | trial_places_count | integer | Кількість доступних місць для пробного заняття |
| | | available | boolean | Чи доступний цей шаблон |
| | | duration | integer | Тривалість заняття |
| | | skill_level | skilllevel | Рівень підготовки |
| type | lessonstype | Тип уроку | | |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Продовження таблиці 2.1

| | | | | |
|---------------------|---------------------------------|--------------------|---------|---|
| lesson_visitor | Відвідувач уроку | id | serial | Ідентифікатор |
| | | user_id | integer | Користувач, який відвідує заняття |
| | | lesson_id | integer | Заняття, яке відвідується |
| | | lesson_purchase_id | integer | Куплена підписка, по якій відвідується заняття |
| | | is_first_visit | boolean | Чи це перше відвідування уроку в школі |
| | | pay_amount | integer | Сума, яка повинна бути оплачена за урок (грн) |
| | | moved_to_lesson_id | integer | Урок, на який було перенесене дане відвідування |
| | | visited | boolean | Чи відвідав користувач даний урок |
| | | paid_online | boolean | Оплачено онлайн |
| | | paid_offline | boolean | Оплачено на уроці (офлайн) |
| lesson_subscription | Підписка на відвідування уроків | id | serial | Ідентифікатор |
| | | lessons_count | integer | Кількість уроків у підписці |
| | | moves_count | integer | Кількість переносів уроків у підписці |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Продовження таблиці 2.1

| | | | | |
|---------------------|---|--------------------------------|-----------|--|
| | | freeze_da ys_count | integer | Кількість заморозок уроків у підписці |
| | | price | integer | Ціна підписки |
| | | available | boolean | Чи доступна підписка для покупки |
| | | created_at | timestamp | Час та дата створення підписки |
| | | updated_a t | timestamp | Час та дата модифікації підписки |
| lesson_pu rchase | Куплена підписка на відвідування уроків. | id | serial | Ідентифікатор |
| | | buyer_id | integer | Користувач, який купив підписку |
| | | lesson_su bscription _id | integer | Куплена підписка |
| | | freeze_left | integer | Кількість доступних заморозок |
| | | movement s_left | integer | Кількість доступних переносів |
| | | lessons_le ft | integer | Кількість доступних уроків для відвідування |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Висновок до розділу

У розділі інформаційного забезпечення визначаються формати вхідних та вихідних даних та опис структури бази даних. На вхід система отримує дані від різних типів користувачів: учнів, педагогів, адміністраторів. Вихідні дані також формуються в залежності від типу користувача. Для учня це дані про поточний стан абонементу, розклад персональних занять, інформаційні повідомлення тощо. Викладач отримує більш розширену інформацію про наявні групи, заняття, оплату уроків. Адміністратор крім загальних даних має можливість опрацювати інформацію про завантаження залів, впливати на зміни розкладу, отримувати статистику відвідувань для подальшого аналізу. В розділі в окрему таблицю зведено деталізовану структуру даних для всіх типів користувачів.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 26 |

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

У кожного тренера балетної школи є фіксований робочий графік на кожний день його роботи. При проведенні індивідуальних занять у власників балетної школи постає задача: скласти оптимальний розклад для кожного тренера на кожний день роботи зі студентами, що записались на індивідуальні заняття, враховуючи побажання кожного студента щодо часу і тривалості. В балетній школі не один тренер, тому постає задача розробити алгоритм, що за відносно короткий проміжок часу зміг би розробити оптимальний розклад з найменшими втратами часу тренера (вікнами у розкладу).

Розглядається задача теорії розкладів, в якій потрібно знайти оптимальну послідовність індивідуальних занять студентів балетної школи. Кожний студент заповнює форму, в якій вказує наступні дані:

- Тренер, до якого хоче потрапити студент або в якого вже вчиться;
- День, в який необхідно провести заняття;
- Час, в котрий повинен початися урок;
- Тривалість заняття.

3.2 Математична постановка задачі

Потрібно обробити пзаявок студентів на індивідуальне заняття для кожного тренера. Кожна заявка має наступні параметри: фактичний час початку заняття – позначимо як a_i , де $a_i \in [9,20]$ (балетна школа починає працювати не раніше дев'ятої години ранку та закінчує працювати не пізніше дев'ятої години вечора), та час тривалості заняття – позначимо як l_i , таке що $l_i \in [1,5]$ (заняття кожного студента не може продовжуватись менше однієї години і більше п'яти годин). Також у кожного тренера є свій графік роботи на день, позначимо час початку роботи x , таке що $x \in [9,20]$ та кінець його робочого дня y , таке що $y \in [10,21]$, $y > x$.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 27 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Знайдемо час закінчення заняття студента по формулі 3.1:

$$b_i = a_i + l_i, i = \overline{1, n}, \quad (3.1)$$

Таки чином ми знайшли час закінчення заняття, як суму часу початку заняття і часу тривалості заняття.

Знайдемо час тривалості роботи тренера по формулі 3.2:

$$k = y - x, \quad (3.2)$$

Таким чином ми знайшли тривалість роботи тренера як різницю часу кінця його робочого дня і часу початку роботи.

Тоді цільова функції задачі буде мати наступний вигляд:

$$z = k_i - \sum_{i=0}^n l_i \rightarrow \min, i = \overline{1, n}, \quad (3.3)$$

При виконанні умови 3.4:

$$\begin{cases} a_{i+1} \leq b_i \\ a_0 \geq x \\ b_n \leq y \end{cases}, \quad (3.4)$$

Тобто задача полягає в наступному: визначити порядок в якому повинні бути розташовані заявки на індивідуальні заняття, щоб час простою (коли тренер в робочій час відпочиває, через відсутність занять) був мінімальним.

Графічне представлення умови задачі наведено на рисунку 3.1:

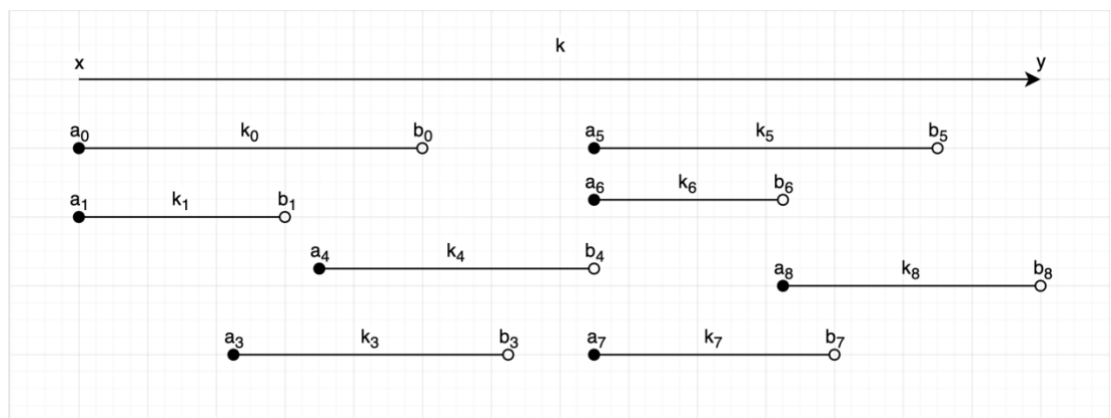


Рисунок 3.1 – Графічне представлення умови задачі

На рисунку 3.2 червоні лінії є часовими проміжками уроків, які входять в оптимальний розклад для тренера. Чорні проміжки при додаванні в розклад зроблять його не оптимальним і утворять більші затримки.

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

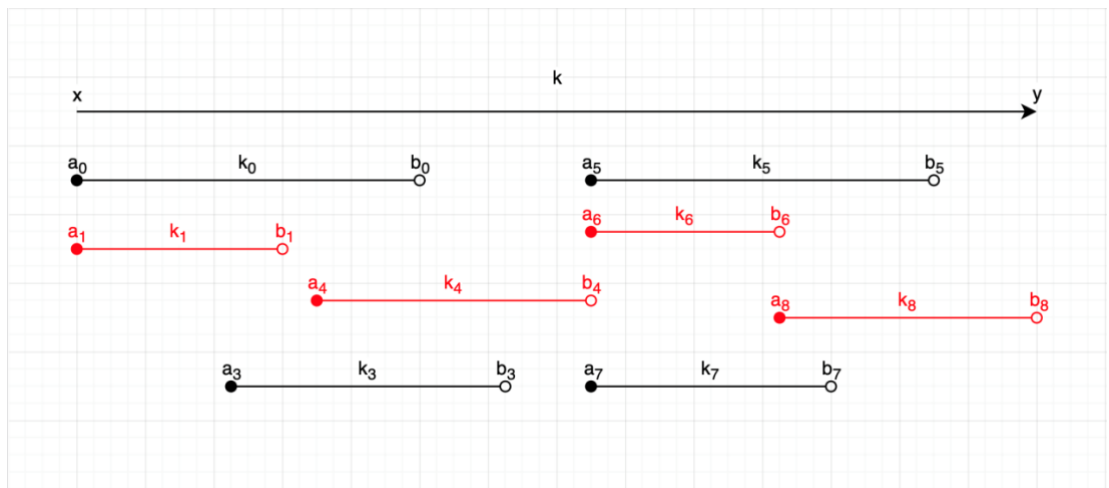


Рисунок 3.2 – Графічне представлення розв’язку задачі

3.3 Обґрунтування методу розв’язання

Часова складність алгоритму

Для вирішення даної задачі запропоновано наступні алгоритми:

- Повний перебір;
- Жадібний алгоритм.

Повний перебір нам гарантує правильне рішення але часова складність даного алгоритму представлена функцією факторіалу. А значить підходить для пошуку рішення лише для малої кількості учнів. На рисунку 3.3 показаний характер росту функції часу від кількості учнів для повного перебору.

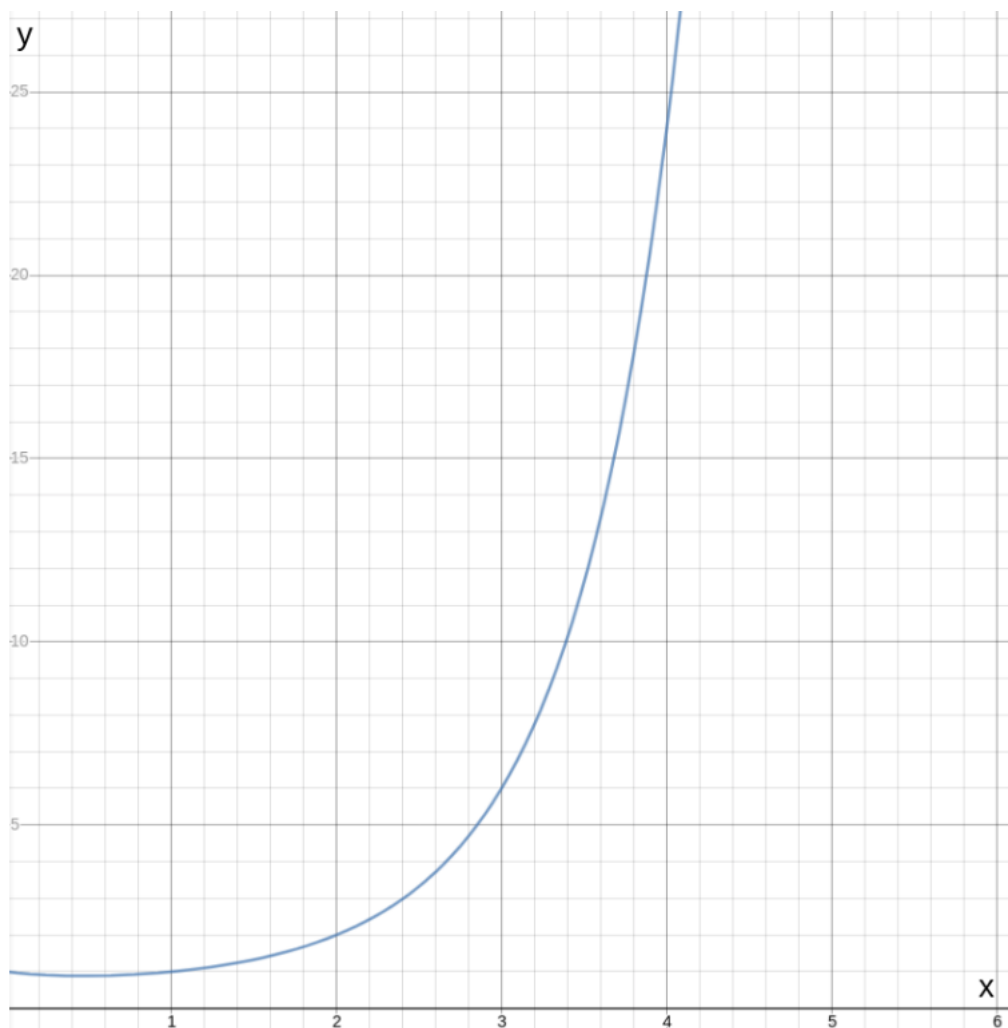


Рисунок 3.3 – Графік характеру зростання функції часу від кількості учнів. По осі абсцис - кількість студентів, по осі ординат - значення функції часу

Якщо кількість студентів буде перевищувати 10, швидкість пошуку оптимального розкладу буде надто низька. Аби прискорити швидкість пошуку вирішено застосувати жадібний алгоритм, який повинен за значно менший час визначити близький до оптимального розклад. Складність жадібного алгоритму: $O(n \cdot \log(n))$

На рисунку 3.4 представлено характер зростання функції часу від кількості студентів для жадібного алгоритму у порівнянні з алгоритмом повного перебору.

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

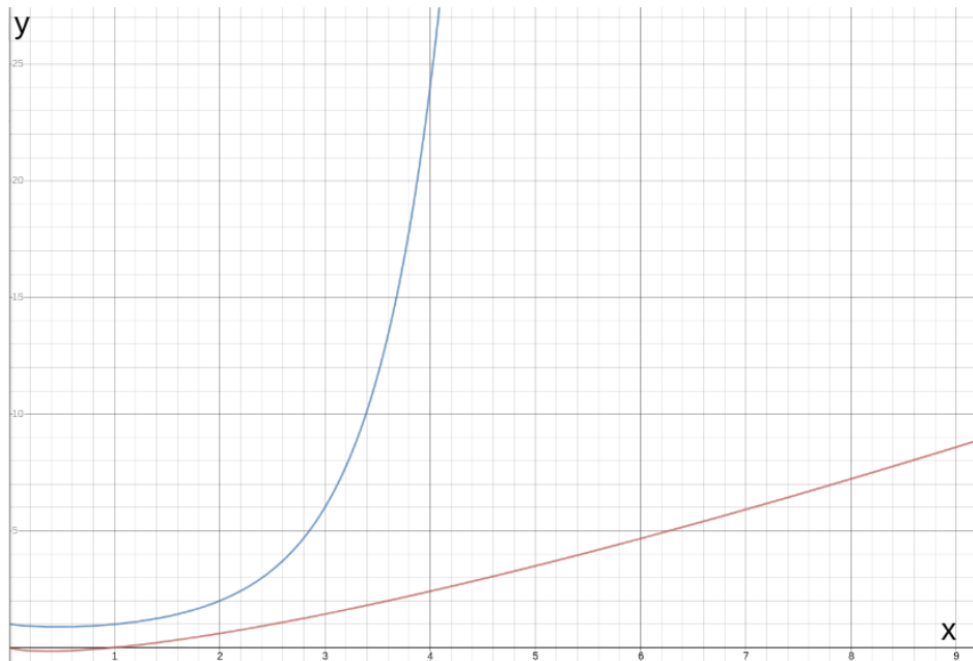


Рисунок 3.4 – Порівняння часу виконання жадібного алгоритму (нижній графік) з алгоритмом повного перебору (верхній графік). По осі абсцис - кількість студентів

Аналіз точності

Точність жадібного алгоритму повністю залежить від характеру даних, чим менша довжина уроку у студентів тим менша теоретична можлива затримка між уроками, а також чим більша кількість студентів, тим більше варіантів для вибору студента з найменшим часом затримки.

При повному переборі, ми гарантовано отримуємо вірний результат.

3.4 Опис методів розв'язання

Наша мета - отримати мінімальний час простою тренера, при розв'язанні жадібним алгоритмом ми намагаємось тримати загальний час простою мінімальним, для цього при виборі студента ми переглядаємо список усіх студентів і обираємо такого студента при якому різниця часу закінчення заняття попереднього студента і часу початку нового студента - мінімальна. Таким чином:

1. На першому кроці обираємо студента у якого різниця часу початку заняття з часом початку роботи тренера - мінімальна

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

2. Кожен наступний студент обирається так, щоб різниця часу закінчення заняття попереднього студента і обраного була мінімальна серед інших студентів:

$$a_{i+1} - b_i = \min, \text{де}$$

a_{i+1} - час початку заняття нового студенту,

b_i - час закінчення заняття попереднього студенту.

При повному переборі ми перебираємо усі набори студентів і обраховуємо час затримки тренера, при кожному покращенні значення цільової функції ми запам'ятовуємо набір студентів, які дали кращий результат. Після повного перебору усіх варіантів ми отримаємо оптимальне рішення.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 32 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Висновок до розділу

У даному розділі запропоновано математичну модель задачі планування розкладу занять студентів у тренера. Було розглянуто два методи пошуку оптимального розкладу та порівняно їх швидкість та точність результатів.

У результаті порівняння кращим з точки зору точності є алгоритм повного перебору, однак при кількості студентів більше 10, час його виконання зростає надто сильно, тому було вирішено використовувати алгоритм повного перебору лише для формування розкладу для тренерів, кількість студентів яких не перевищує 10, таким чином для цих тренерів буде сформований оптимальний розклад за прийнятний час. Якщо ж кількість студентів у тренера перевищує 10, то у цьому випадку застосовується жадібний алгоритм, який дає наближений до оптимального результат за значно менший час.

Таким чином поєднанням точного алгоритму для маленької кількості даних та швидкого неточного алгоритму для великої кількості даних, було досягнуто компроміс між швидкістю та оптимальністю вихідного розкладу.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 33 |

ЗАГАЛЬНІ ВИСНОВКИ

У дипломному проєкті було розроблено інформаційну систему підтримки діяльності балетної школи у вигляді веб застосунку і мобільного доданку. Розробка системи виконувалась для мережі балетних студій для дорослих «MaryBallet». Важливим аспектом було врахування специфіки бізнесу та забезпечення ефективної комунікації з клієнтом. В той же час, незважаючи на акцент на персоніфікацію продукту, система створювалась з забезпеченням можливої подальшої адаптації для подібних бізнесів, таких як, наприклад, спортивних шкіл, клубів за інтересами або навчальних закладів подібного типу.

Для реалізації клієнтського застосунку було розроблено прототипні версії сайту та протестована їх з участю учнів та педагогів балетної школи. На основі зворотного зв'язку оновлювалися вимоги до інтерфейсу та функціонального наповнення сайту. На основі виокремлених вимог було розроблено дизайн для веб-сайту та мобільної версії застосунку.

В роботі запропоновано та реалізовано загальне архітектурне рішення системи та її складових компонент. Обґрунтовано методи та засоби реалізації системи.

З метою перевірки відповідності вимогам та якості системи було проведено тестування всіх компонент. Крім того, система була надана для користування в тестовому режимі учням та тренерам школи. Для цього було розроблено стратегію розгортання системи в режимі розробки, тестовому та для загального користування.

На сьогодні веб-сайт з базовим функціоналом введено в експлуатацію мережею «MaryBallet», користування мобільним застосунком продовжується в тестовому режимі.

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 34 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ПЕРЕЛІК ПОСИЛАНЬ

1. AppStore [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:<https://apps.apple.com/ua/app/apple-store/id375380948?l=ru>;
2. Instagram[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.instagram.com> ;
3. Facebook[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.facebook.com> ;
4. Privat24 [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.privat24.ua> ;
5. Python[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.python.org> ;
6. TypeScript[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:<http://typescriptlang.org> ;
7. JavaScript[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:<https://javascript.info> ;
8. Swift[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.apple.com/ru/swift/> ;
9. Objective-c[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:<https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html> ;
10. Excel[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.microsoft.com/uk-ua/microsoft-365/excel> ;
11. Балетная школа в Киеве - студия балета MaryBallet, частные уроки, занятия у станка [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://maryballet.com.ua/>;
12. Управление клиентами от Baseplan. Crm системы для управления взаимоотношениями с клиентами. [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://baseplan.ru/>;

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 35 |

13. CRM система автоматизации учета посещаемости занятий, программа для электронного учета клиентов | Отмечалка [Электронный ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://otmechalka.com/>;

14. Программное обеспечение для управления фитнес-клубом - Mobifitness [Электронный ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://mobifitness.ru/>;

15. Параплан CRM - облачная CRM [Электронный ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://paraplancrm.ru/>;

16. Онлайн CRM система. Управление клиентами в современной облачной CRM — amoCRM [Электронный ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.amocrm.ru/>;

17. HubSpot | InboundMarketing, Sales, andServiceSoftware [Электронный ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.hubspot.com/>;

18. Notion – The all-in-one workspace for your notes, tasks, wikis, and databases. [Электронный ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.notion.so/>;

19. Бітрікс24: сервіс автоматизації і оптимізації бізнес-процесів компанії [Электронный ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.bitrix24.ua/>;

20. PostgreSQL: The world's most advanced opensource database [Электронный ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.postgresql.org/>;

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.000 ПЗ | Арк. |
| | | | | | | 36 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації і управління

УЗГОДЖЕНО

Керівник проекту

_____ О. О. Мажара
(підпис) (ініціали, прізвище)

“13” квітня 2020 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ О.А. Павлов
(підпис) (ініціали, прізвище)

“14” квітня 2020 р.

«Інформаційна система підтримки процесів
діяльності балетної школи»
ТЕХНІЧНЕ ЗАВДАННЯ

Шифр ДП ІС-6226.00.000 ТЗ

на 11 сторінках

Київ – 2020 року

ЗМІСТ

| | | |
|----------|--|-----------|
| 1 | ЗАГАЛЬНІ ПОЛОЖЕННЯ | 3 |
| 1.1 | ПОВНЕ НАЙМЕНУВАННЯ СИСТЕМИ ТА ЇЇ УМОВНЕ ПОЗНАЧЕННЯ..... | 3 |
| 1.2 | НАЙМЕНУВАННЯ ОРГАНІЗАЦІЇ-ЗАМОВНИКА ТА ОРГАНІЗАЦІЙ-УЧАСНИКІВ РОБИТ..... | 3 |
| 1.3 | ПЕРЕЛІК ДОКУМЕНТІВ, НА ПІДСТАВІ ЯКИХ СТВОРЮЄТЬСЯ СИСТЕМА..... | 3 |
| 1.4 | ПЛАНОВІ ТЕРМІНИ ПОЧАТКУ І ЗАКІНЧЕННЯ РОБОТИ ЗІ СТВОРЕННЯ СИСТЕМИ..... | 3 |
| 2 | ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ ЗАСОБІВ | 5 |
| 2.1 | ПРИЗНАЧЕННЯ ЗАСОБІВ..... | 5 |
| 2.2 | ЦІЛІ СТВОРЕННЯ ЗАСОБІВ..... | 5 |
| 3 | ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ | 7 |
| 4 | ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 8 |
| 4.1 | ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК..... | 8 |
| 4.2 | ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК..... | 9 |
| 4.3 | ВИМОГИ ДО СКЛАДУ І ПАРАМЕТРІВ ТЕХНІЧНИХ ЗАСОБІВ..... | 9 |
| 5 | СТАДІЇ І ЕТАПИ РОЗРОБКИ | 11 |
| 6 | ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ | 12 |

| | | | | | | | | |
|----------------|-----------------|-----------------|---------------|-------------|---|---|-------------|----------------|
| | | | | | ДП ІС-6226.00.000 ТЗ | | | |
| | | <i>Прізвище</i> | <i>Підпис</i> | <i>Дата</i> | | | | |
| <i>Розроб.</i> | Халін О.І. | | | | Інформаційна система підтримки діяльності балетної школи (комплексна тема) | <i>Літ</i> | <i>Арк.</i> | <i>Аркушів</i> |
| | | | | | | | 2 | 11 |
| <i>Керівн.</i> | Мажара О.О. | | | | | КПІ ім. Іоанна Сікорського Каф. АСОІУ гр. ІС-62 | | |
| <i>Н. кон.</i> | Новінський В.П. | | | | | | | |
| <i>Затв.</i> | Павлов О.А. | | | | | | | |

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Повне найменування системи: «Інформаційна система підтримки діяльності балетної школи».

Коротке найменування системи: «MaryBallet».

1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником проекту є кафедра автоматизованих систем обробки інформації та управління факультету інформатики та обчислювальної техніки Національного технічного університету України «КПІ імені Ігоря Сікорського».

Розробники застосунку – студенти групи ІС-62 факультету інформатики та обчислювальної техніки НГУУ «КПІ імені Ігоря Сікорського» Халін Олег Ігорович, Кравчук Олександр Ігорович, Щеголь Данило Андрійович.

1.3 Перелік документів, на підставі яких створюється система

Підставою для розробки “Інформаційна система підтримки діяльності балетної школи” є завдання на дипломне проектування, затвержене кафедрою автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку роботи над створенням застосунку – 1 грудня 2019 року.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.000ТЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 3 |

ДП ІС-6226.00.000ТЗ

Плановий термін по закінченню роботи над створенням застосунку – 25 травня 2020 року.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | | Арк. |
| | | | | | ДП ІС-6226.00.000ТЗ | 4 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ ЗАСОБІВ

2.1 Призначення засобів

Розроблювана система призначена для автоматизованої роботи балетної школи. Автоматизація процесів у бізнесі надає купу переваг як підприємцям, так і клієнтам. Автоматизовані системи мають наступні переваги: значно менші фінансові витрати у довгостроковій перспективі, більша стресостійкість через меншу наявність людського фактору, економія часу та зручніший контроль даних. Окрім того у теперішній час інформаційні системи є невід'ємною частиною корпоративної культури та найціннішою складовою маркетингу.

Інформаційна система повинна бути:

- надійною — для того, щоб усі процеси відбувались без перешкод;
- адаптивною — тобто бути зручною для майбутнього розширення функціоналу або змін у дизайні;
- кросплатформною — для того щоб не обмежувати цільову аудиторію користувачів у рамках певної платформи;
- інтуїтивною у користуванні;
- функціонально повною, але не перенасиченою зайвими функціями.

Даний проект може бути використаний у якості прикладу клієнт-серверного застосунку у будь-якій предметній області. За мету було взято створити як можна гнучкішу архітектуру для подальшого використання у інших кросплатформних систем.

2.2 Цілі створення засобів

Ціль розробки – створення програмного продукту, який автоматизує створення та зміну індивідуальних розкладів відвідувачів балетної школи,

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | | Арк. |
| | | | | | | 5 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДП ІС-6226.00.000ТЗ | |

дозволить здійснювати електронні платежі для оплати занять, спостерігати за новинами школи та полегшить взаємодію танцюрів та тренерів.

Для досягнення поставленої цілі необхідно виконати наступні задачі:

- розробити дизайн для десктопної та мобільної версій застосунку;
- спроектувати архітектуру серверної частини додатку;
- спроектувати архітектури клієнтських частин додатку для кожної платформи, намагаючись їх якомога більше уніфікувати;
- розробка усіх необхідних екранів застосунку згідно дизайну;
- інтеграція серверної частини у клієнтські частини застосунку;
- розробка платформиорієнтованого функціоналу, функціоналу який буде унікальним у рамках певної платформи;
- рефакторинг коду з метою оптимізації його швидкодії та лаконічності;
- тестування додатку на предмет наявності логічних та програмних помилок.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | | Арк. |
| | | | | | ДП ІС-6226.00.000ТЗ | 6 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

В роботі розробляється інформаційна система підтримки процесів діяльності балетної школи у вигляді веб та мобільного додатку. Однак ця система може використовуватись будь-якою навчальною організацією: починаючи з курсів і персональних занять та закінчуючи великими школами та навіть університетами. Вона є не просто корисною, а необхідною для значної кількості бізнесів, які переростають з маленьких персональних фірм до великих корпорацій, що обслуговують значну кількість людей.

На ринку існує безліч різноманітних CRM/ERP систем, якими користуються більшість розвинутих компаній, однак ми націлились на необхідні користувачу задачі, які не були реалізовані в більшості варіантів. А саме: автоматизація розкладу занять, що включають в себе різні рівні підготовки клієнтів, найбільш зручну для них локацію, час та певного тренера; вибір необхідного для клієнта абонементу, зручний перенос і подовження занять, що може відбуватися без витрачених ресурсів бізнесу; оплата онлайн; можливість тренера відмічати відвідуваність клієнтів і додавати коментарі, що допомагає йому стежити за необхідною навантажкою кожного клієнта індивідуально; автоматизація так званих «розумних повідомлень», що реалізуються за допомогою емейла та смс. Додаємо до цього всього зручний інтерфейс і кросплатформеність, щоб тренер міг це зробити у мобільному додатку в режимі офлайн. В результаті матимемо зручну систему, якою зможуть користуватися більшість шкіл і навчальних організацій.

В роботі створюється продукт персонально за заданими, що дозволяє в будь-який час його кастомізувати під потреби користувача. Його функціонал буде економити час та гроші власнику, а детально розроблений дизайн і юзер флоу клієнтської частини застосунку дозволить користувачам легко і інтуїтивно зрозуміло виконати всі необхідні задачі та дізнатися всю необхідну інформацію.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | | Арк. |
| | | | | | ДП ІС-6226.00.000ТЗ | 7 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Інформаційна система повинна надавати можливість переглядати обраний розклад занять, редагувати розклад, обираючи час та викладачів, а також отримувати повідомлення щодо змін у розкладі.

Система повинна мати бокове меню для зручної навігації по застосунку, яке повинне мати такі вкладки: розклад, повідомлення, поточне заняття, підтримка.

Система повинна виконувати наступні функції:

- Переглянути розклад – користувач обирає вкладку «Розклад» та на екрані відображається розклад занять на тиждень;
- Редагувати розклад – користувач обирає вкладку «Розклад», натискає на кнопку «Редагувати» та отримує можливість видаляти заняття з поточного розкладу та додавати заняття зі списку усіх занять;
- Переглянути інформацію про поточне заняття – користувач системи натискає на вкладку «Поточне заняття» для огляду учасників та інформації про тренера;
- Переглянути статус оплати зайняття – тренер натискає на вкладку «Поточне заняття» та переглядає статус оплати зайняття для кожного відвідувача поточного зайняття;
- Переглянути історію повідомлень – користувач натискає на вкладку «Повідомлення» та переглядає історію отриманих раніше повідомлень;
- Надавати доступ до контактних даних служби підтримки – користувач обирає вкладку «Підтримка» та отримує можливість подзвонити або написати у службу підтримки;

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | | Арк. |
| | | | | | ДП ІС-6226.00.000ТЗ | 8 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

- Система повинна мати змогу надсилати повідомлення окремим користувачам із текстовими даними. Користувачі в свою чергу повинні мати змогу отримувати дані повідомлення.]

Примечание [11]: Правила нумерації

4.2 Вимоги до функціональних характеристик

Система повинна працювати безвідмовно, незважаючи на наявність ймовірних дефектів, котрі можуть проявлятися під час експлуатації. Жодні зовнішні чи внутрішні умови не повинні бути причиною руйнування даних в інформаційних сховищах, збою чи закінченню життєвого циклу програми.

Розрив або відсутність Інтернет-з'єднання не повинні бути причиною критичних помилок у роботі мобільної версії додатку, адже вона підтримує кешування даних та не залежить від Інтернет-з'єднання для відображення певних даних.

Система повинна проходити тестування критичними ситуаціями ще на етапі розробки задля усунення подальших можливих дефектів вчасно.

4.3 Вимоги до складу і параметрів технічних засобів

Для коректного функціонування програмного продукту на стороні користувача повинні бути задовільнені певні вимоги щодо складу і параметрів технічних засобів.

Для мобільного додатку єдиною вимогою буде те, що пристрій користувача повинен працювати на операційній системі iOS 13. Дана вимога пов'язана із використанням найновіших фреймворків під час написання програмного продукту.

Для веб-застосунку список вимог є трохи ширшим, втім даним вимогам задовольняє майже будь-який комп'ютер на сьогоднішній день:

- Процесор з тактовою частотою не менше 1 ГГц;
- Об'єм оперативної пам'яті не менше 1 Гб;
- Підключення до мережі Інтернет.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | | Арк. |
| | | | | | ДП ІС-6226.00.000ТЗ | 9 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ДП ІС-6226.00.000ТЗ

Для взаємодію із веб-застосунком комп'ютер також повинен мати необхідну периферію:

- Монітор;
- Миша, трекпад (або дісплей із тачскріном);
- Клавіатура (або дисплей з тачскріном).

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|----------------------------|------|
| | | | | | | Арк. |
| | | | | | <i>ДП ІС-6226.00.000ТЗ</i> | 10 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з розробки наведені в таблиці 5.1.

Таблиця 5.1

| Назва етапу | Термін виконання |
|--|------------------|
| Вивчення рекомендованої літератури | 01.03.2020 |
| Аналіз існуючих методів розв'язання задачі | 07.03.2020 |
| Постановка та формалізація задачі | 15.03.2020 |
| Розробка програмного забезпечення | 11.05.2020 |
| Налагодження програми | 12.05.2020 |
| Оформлення пояснювальної записки | 14.05.2020 |
| Подання ДП на попередній захист | 15.05.2020 |
| Подання ДП на основний захист | 25.06.2020 |

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Задля перевірки правильності роботи програмного продукту буде проведено функціональне тестування. В ході тестування буде виконано перевірку всіх функціональних характеристик, як веб-застосунку, так і мобільного застосунку. Також, система буде перевірена на відмовостійкість шляхом виконання некоректних дій користувачем та надсиланням запитів із некоректними даними. Окремими тестами буде перевірена безпека системи в цілому. За допомогою спеціальних інструментів будуть перевірені швидкодія та енергоефективність мобільного застосунку.

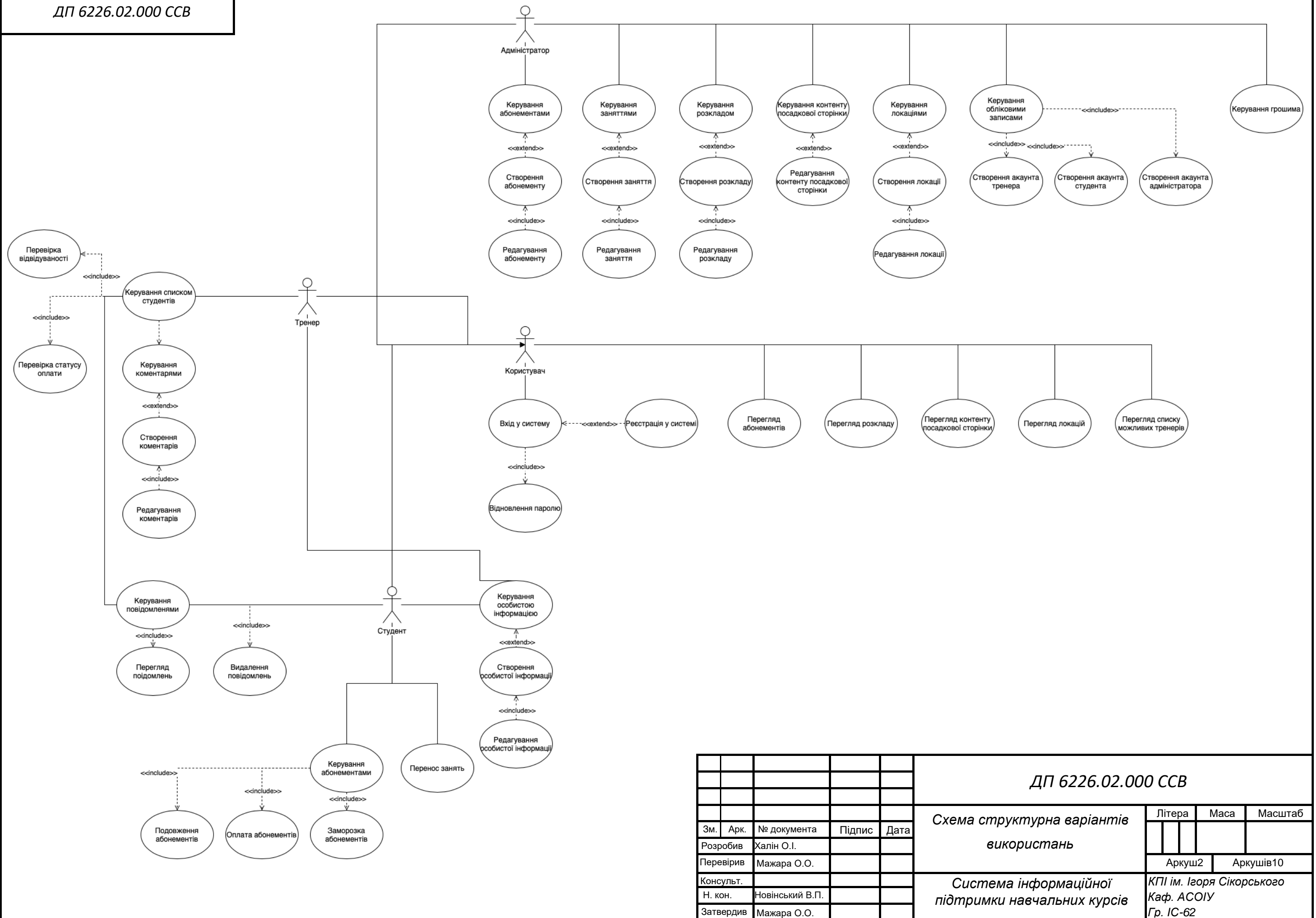
| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | | Арк. |
| | | | | | ДП ІС-6226.00.000ТЗ | 12 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Графічний матеріал до дипломного проєкту

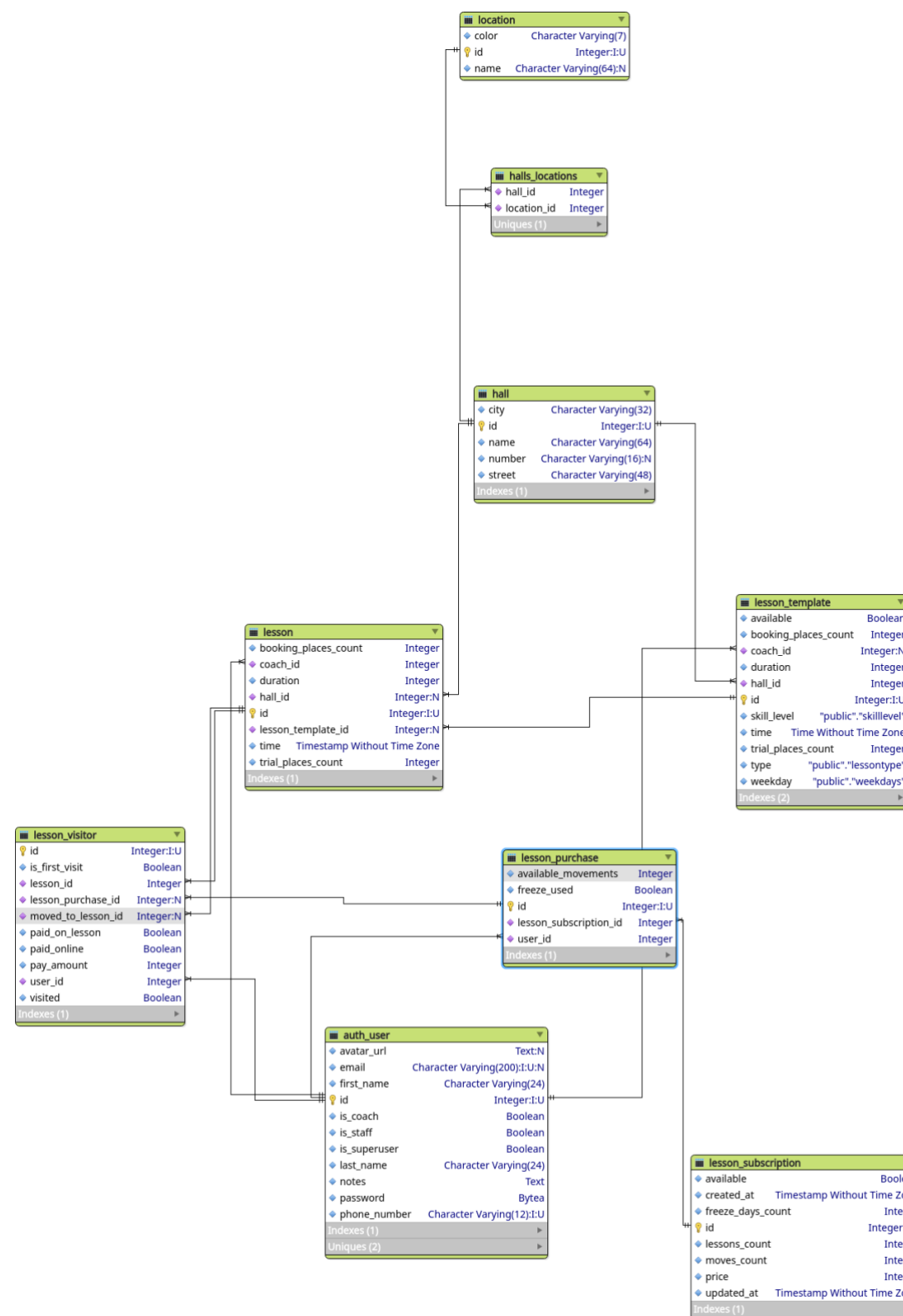
на тему: Інформаційна система підтримки діяльності балетної школи

(комплексна тема)

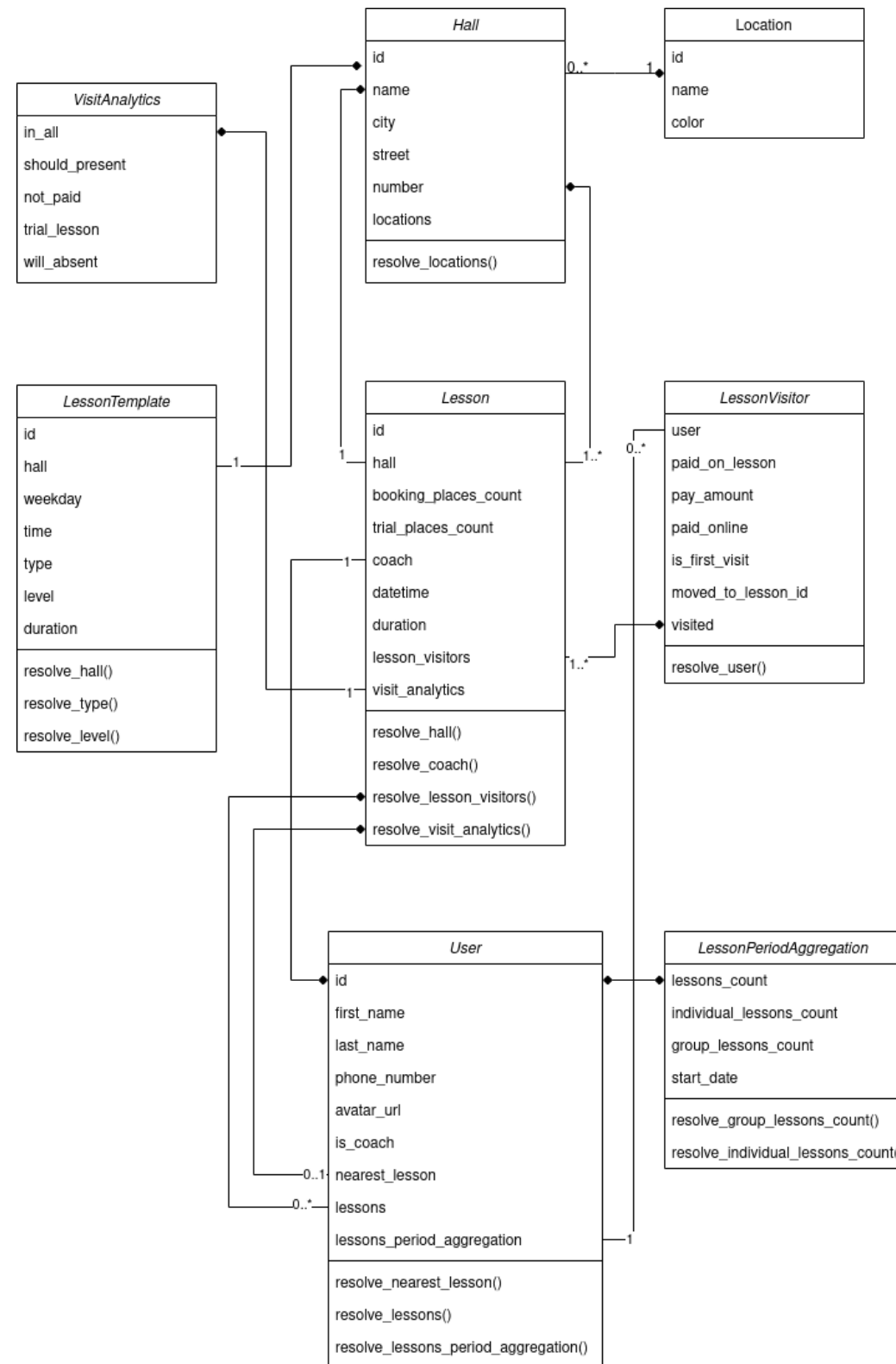
Київ – 2020 року



| | | | | | | | | | | |
|-----------|-----------------|-------------|--------|------|--|--|---|-----------|---------|--|
| | | | | | ДП 6226.02.000 ССВ | | | | | |
| Зм. | Арк. | № документа | Підпис | Дата | Схема структурна варіантів використань | | Літера | Маса | Масштаб | |
| Розробив | Халін О.І. | | | | | | | | | |
| Перевірив | Мажара О.О. | | | | Система інформаційної підтримки навчальних курсів | | Аркуш2 | Аркушів10 | | |
| Консульт. | | | | | | | <i>КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62</i> | | | |
| Н. кон. | Новінський В.П. | | | | | | | | | |
| Затвердив | Мажара О.О. | | | | | | | | | |



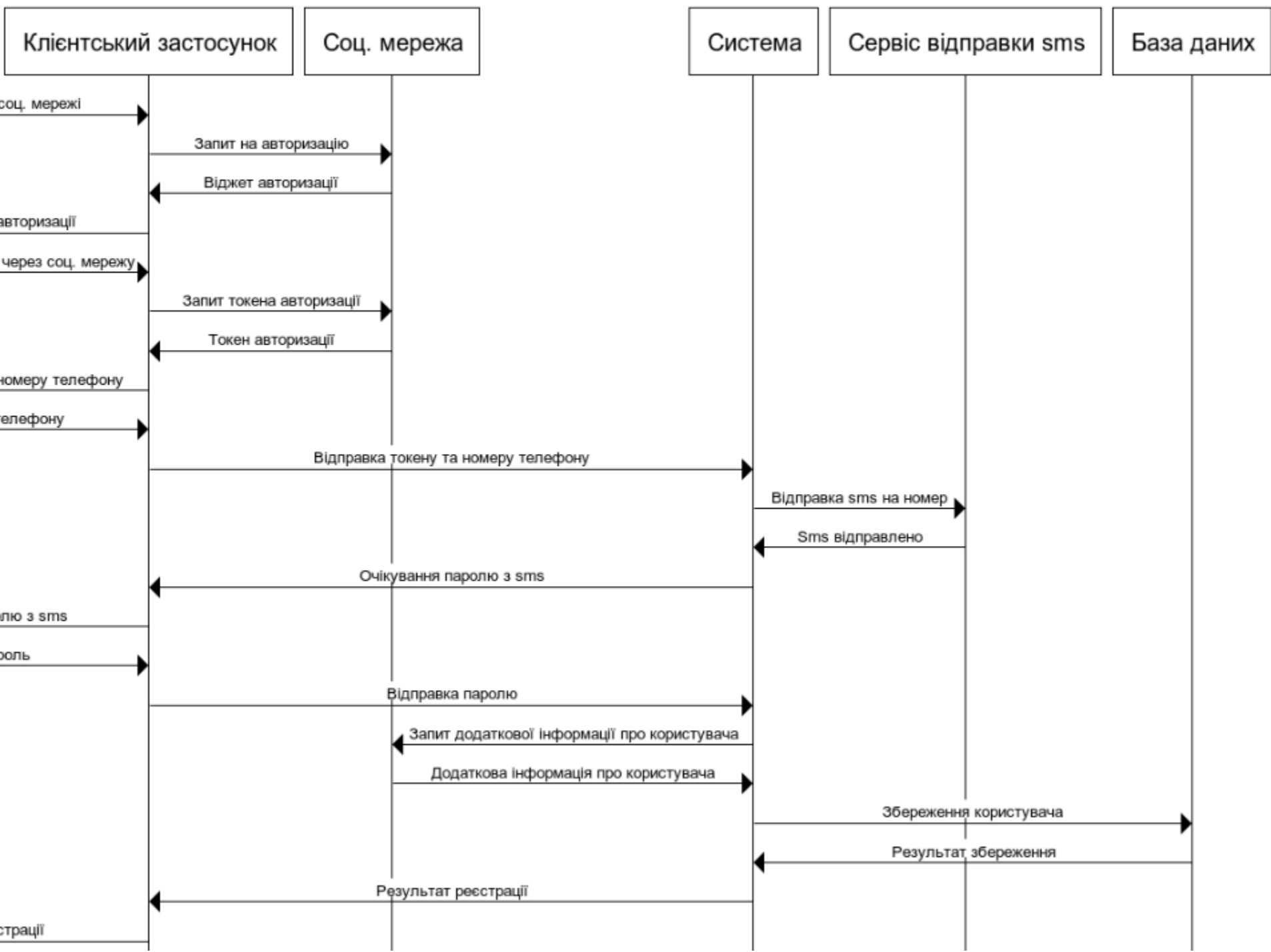
| | | | | | | | | | | |
|-----------|------|-----------------|--------|------|---|--|--|------|------------|--|
| | | | | | ДП 6226.03.000 СБД | | | | | |
| | | | | | Схема базиданих | | Літера | Маса | Масштаб | |
| Зм. | Арк. | № документа | Підпис | Дата | | | | | | |
| Розробив | | Халін О.І. | | | Система інформаційної підтримки навчальних курсів | | Аркуш 3 | | Аркушів 10 | |
| Перевірив | | Мажара О.О. | | | | | КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62 | | | |
| Консульт. | | | | | | | | | | |
| Н. кон. | | Новінський В.П. | | | | | | | | |
| Затвердив | | Мажара О.О. | | | | | | | | |



| | | | | | | | |
|-----------|------|-----------------|--------|------|---|------|------------|
| | | | | | ДП 6226.04.000 ССК | | |
| | | | | | Схема структурна класів програмного забезпечення | | |
| | | | | | Літера | Маса | Масштаб |
| Зм. | Арк. | № документа | Підпис | Дата | | | |
| Розробив | | Халін О.І. | | | | | |
| Перевірив | | Мажара О.О. | | | Аркуш 4 | | Аркушів 10 |
| Консульт. | | | | | Система інформаційної підтримки навчальних курсів | | |
| Н. кон. | | Новінський В.П. | | | | | |
| Затвердив | | Мажара О.О. | | | | | |
| | | | | | КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62 | | |



Користувач

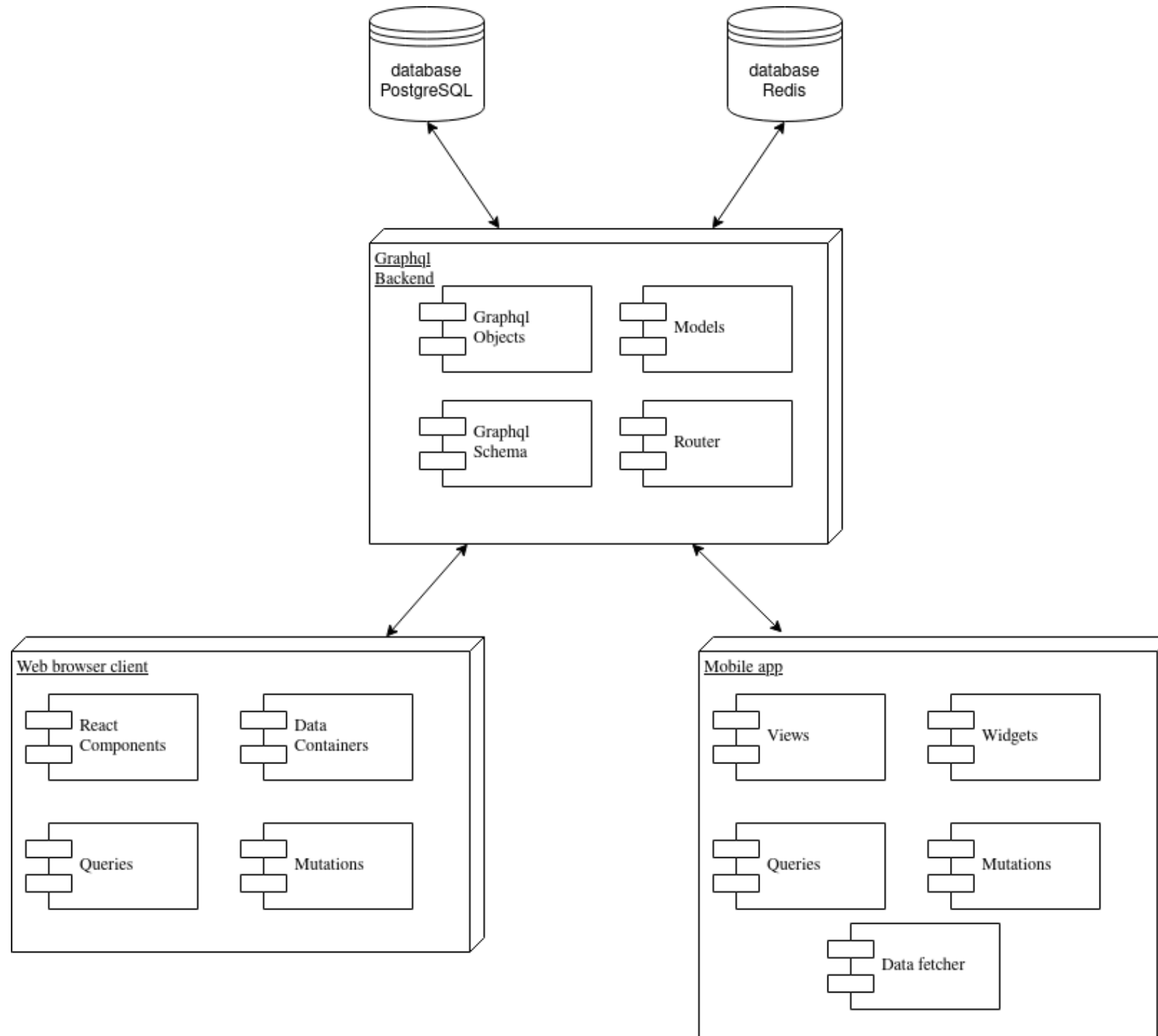


| | | | | | | | |
|-----------|------|-----------------|--------|------|--|------------|---------|
| | | | | | ДП 6226.05.000 ССП | | |
| | | | | | Схема структурна послідовності | | |
| | | | | | Літера | Маса | Масштаб |
| Зм. | Арк. | № документа | Підпис | Дата | | | |
| Розробив | | Халін О.І. | | | | | |
| Перевірив | | Мажара О.О. | | | Аркуш 5 | Аркушів 10 | |
| Консульт. | | | | | КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62 | | |
| Н. кон. | | Новінський В.П. | | | | | |
| Затвердив | | Мажара О.О. | | | | | |

Авторизація



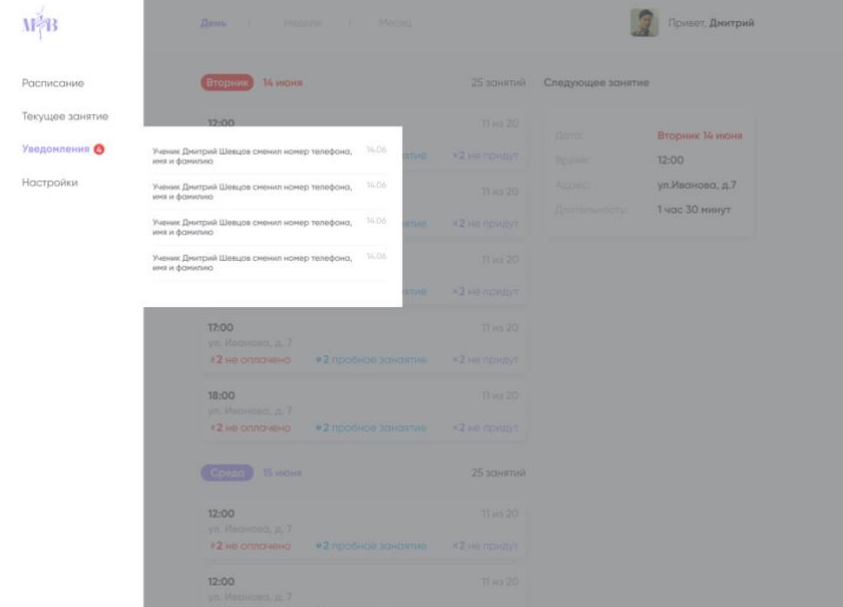
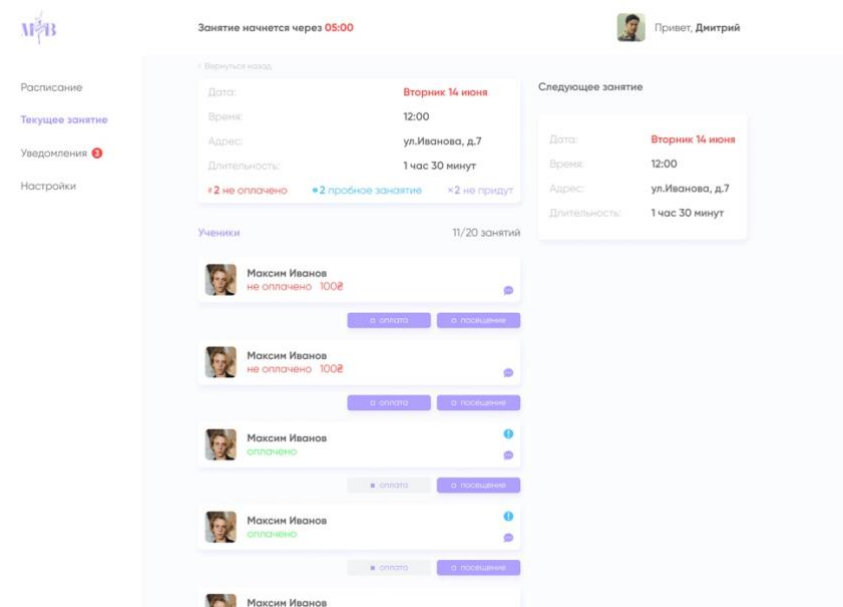
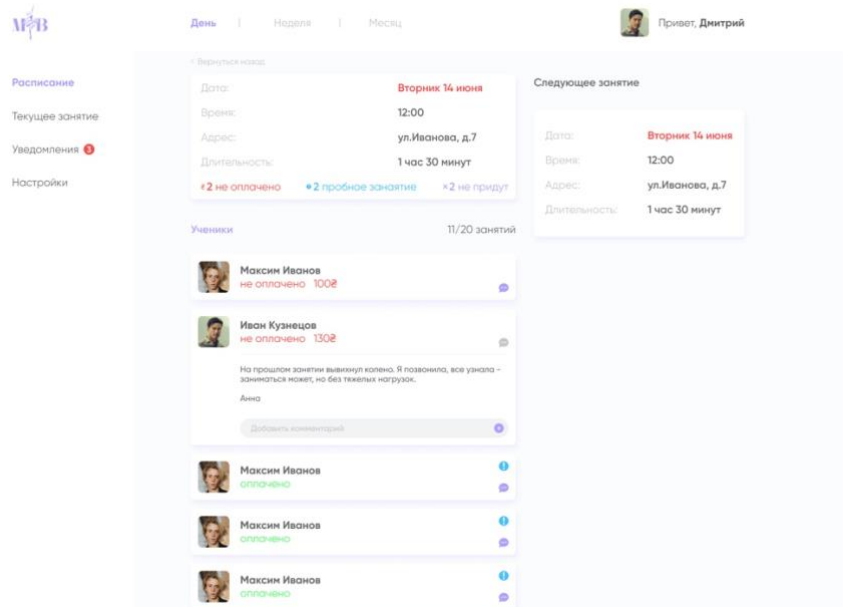
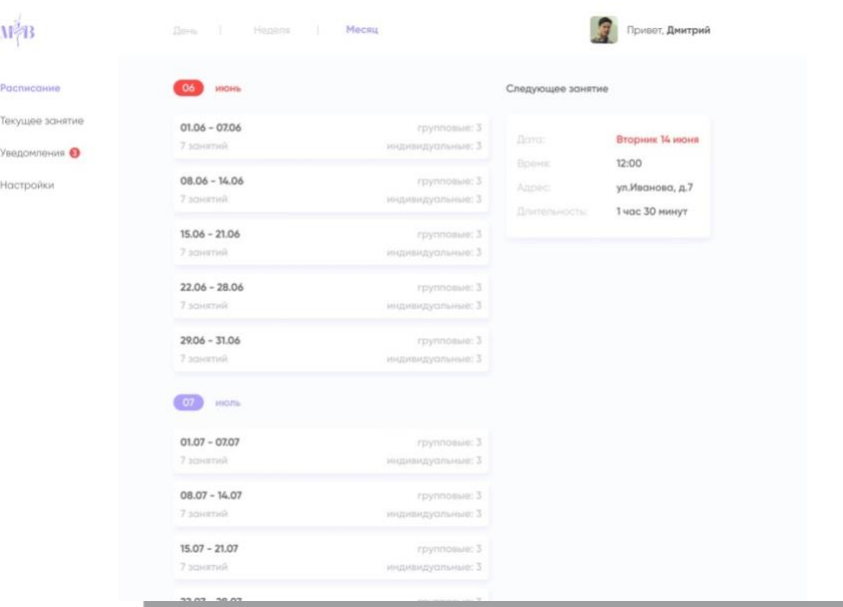
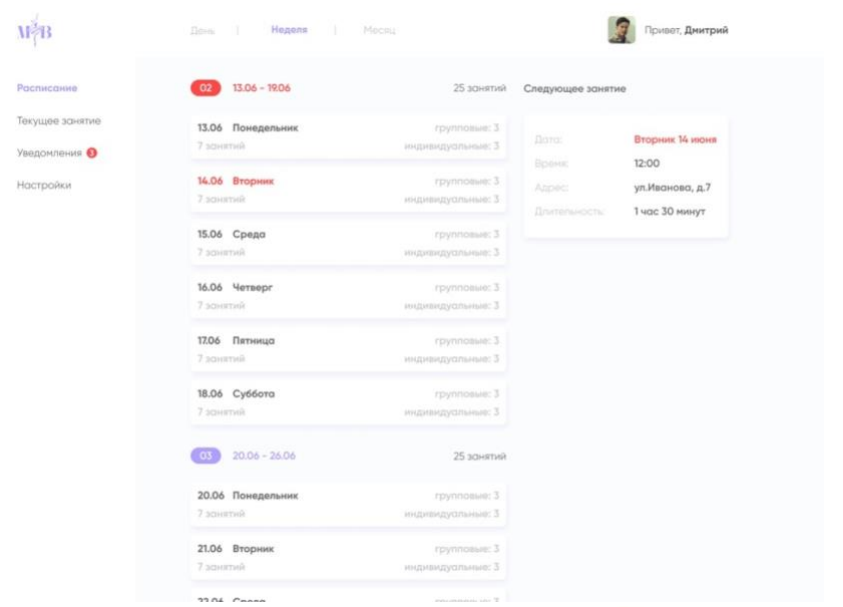
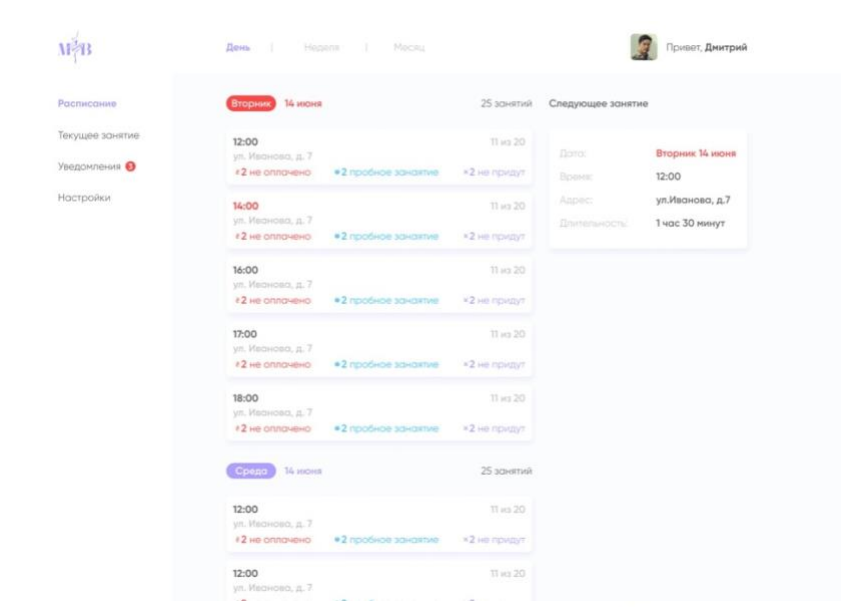
| | | | | | | | | | |
|-----------|------|-----------------|--------|------|--|--|---|------------|---------|
| | | | | | <i>ДП 6226.05.000 ССП</i> | | | | |
| | | | | | <i>Схема структурна послідовності</i> | | Літера | Маса | Масштаб |
| Зм. | Арк. | № документа | Підпис | Дата | | | | | |
| Розробив | | Халін О.І. | | | | | | | |
| Перевірив | | Мажара О.О. | | | | | Аркуш 6 | Аркушів 10 | |
| Консульт. | | | | | <i>Система інформаційної підтримки навчальних курсів</i> | | <i>КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62</i> | | |
| Н. кон. | | Новінський В.П. | | | | | | | |
| Затвердив | | Мажара О.О. | | | | | | | |



| | | | | | | | | | | |
|-----------|------|-----------------|--------|------|--|--|------------|--------|------|---------|
| | | | | | ДП 6226.06.000 ССК | | | | | |
| | | | | | Схема структурних компонентів програмного забезпечення | | | Літера | Маса | Масштаб |
| Зм. | Арк. | № документа | Підпис | Дата | | | | | | |
| Розробив | | Халін О.І. | | | Аркуш 7 | | Аркушів 10 | | | |
| Перевірив | | Мажара О.О. | | | Система інформаційної підтримки навчальних курсів КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62 | | | | | |
| Консульт. | | | | | | | | | | |
| Н. кон. | | Новінський В.П. | | | | | | | | |
| Затвердив | | Мажара О.О. | | | | | | | | |



| | | | | | | | | | | |
|-----------|------|-----------------|--------|------|--|--|--|---|------------|---------|
| | | | | | ДП 6226.06.000 ССК | | | | | |
| | | | | | <i>Схема структурна розгортання програмного забезпечення</i> | | | Літера | Маса | Масштаб |
| Зм. | Арк. | № документа | Підпис | Дата | | | | | | |
| Розробив | | Халін О.І. | | | | | | | | |
| Перевірив | | Мажара О.О. | | | | | | Аркуш 8 | Аркушів 10 | |
| Консульт. | | | | | <i>Система інформаційної підтримки навчальних курсів</i> | | | <i>КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62</i> | | |
| Н. кон. | | Новінський В.П. | | | | | | | | |
| Затвердив | | Мажара О.О. | | | | | | | | |



| | | | | | | | | | | |
|-----------|------|-----------------|--------|------|--|--|--|---|------------|---------|
| | | | | | ДП 6226.07.000 ЕК | | | | | |
| | | | | | Креслення вигляду екранних форм | | | Літера | Маса | Масштаб |
| Зм. | Арк. | № документа | Підпис | Дата | | | | | | |
| Розробив | | Халін О.І. | | | Система інформаційної підтримки навчальних курсів | | | Аркуш 9 | Аркушів 10 | |
| Перевірів | | Мажара О.О. | | | | | | КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62 | | |
| Консульт. | | | | | | | | | | |
| Н. кон. | | Новінський В.П. | | | | | | | | |
| Затвердив | | Мажара О.О. | | | | | | | | |

The screenshot displays a mobile application interface with several main sections:

- Calendar View (Left):** Shows a monthly calendar for June and July. Each date is linked to a lesson plan with details like '3 groups / 2 individual' and the number of lessons.
- Current Lesson View (Middle):** Shows details for a lesson on Tuesday, June 14th at 12:00. It includes the address 'ул.Иванова, д.7', duration '1 час 30 минут', and a list of students with their payment status (e.g., 'Максим Иванов не оплачено 100 ₴').
- Notifications (Right):** A list of notifications such as 'Ученик Дмитрий Шевцов сменил номер телефона' with timestamps.
- Menu (Bottom Right):** Contains options like 'МОЕ РАСПИСАНИЕ', 'ТЕКУЩЕЕ ЗАНЯТИЕ (0)', 'УВЕДОМЛЕНИЯ (3)', 'НАСТРОЙКИ', 'ПОДДЕРЖКА', and 'ВЫХОД ИЗ АККАУНТА'.

| | | | | | | | |
|-----------|------|-----------------|--------|------|---|--|------------|
| | | | | | ДП 6226.07.000 ЕК | | |
| | | | | | <i>Креслення вигляду екранних форм</i> | | |
| Зм. | Арк. | № документа | Підпис | Дата | | | |
| Розробив | | Халін О.І. | | | | | |
| Перевірив | | Мажара О.О. | | | Аркуш 10 | | Аркушів 10 |
| Консульт. | | | | | <i>Система інформаційної підтримки навчальних курсів</i> | | |
| Н. кон. | | Новінський В.П. | | | | | |
| Затвердив | | Мажара О.О. | | | <i>КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62</i> | | |

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

До захисту допущено:

В.о. завідувача кафедри

_____ *Олександр ПАВЛОВ*
(підпис) (вл.ім'я, прізвище)

“ ____ ” _____ 2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційні управляючі системи
та технології»
спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: «Інформаційна система підтримки діяльності балетної
школи» (комплексна тема).

Підсистема розробки прототипів, дизайну та клієнтської
частини веб-застосунку. Індивідуальна частина № 1

Виконав:

студент IV курсу, групи ІС-62

_____ *Халін Олег Ігорович*

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник

_____ *ст. викл. Мажара Ольга Олександрівна*

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

_____ (підпис)

**Консультант з
графічної
документації**

_____ *доц., к.т.н., доц. Новінський Валерій Петрович*

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

_____ (підпис)

Рецензент

_____ *доц., к.т.н., доц. Михайлова Ірина Юріївна*

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

_____ (підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

Пояснювальна записка до дипломного проекту

на тему: Інформаційна система підтримки діяльності балетної школи.

Підсистема розробки прототипів, дизайну та клієнтської частини

веб-застосунку.

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка індивідуальної частини дипломного проекту складається з трьох розділів, містить 8 рисунків, 21 таблицю, 2 додатки, 28 джерел.

Індивідуальна частина дипломного присвячена реалізації прототипів та дизайну системи, а також розробці клієнтської частини веб-застосунку

У розділі постановки цілі і задач надається інформація щодо призначення розробки індивідуальної частини та необхідних кроків для її реалізації.

Розділ програмного забезпечення вводить та пояснює доцільність вибору засобів розробки та проектування, задає вимоги до системи та пояснює побудову архітектури додатку стосовно затребуваних вимог.

У технологічному розділі надається керівництво користувача з демонстрацією інтерфейсу застосунку та проводяться випробування готового продукту на відповідність вимогам.

БАЛЕТНА ШКОЛА, КЛІЄНТСЬКА ЧАСТИНА, ІНФОРМАЦІЙНА СИСТЕМА, ДИЗАЙН

| | | | | | | | | |
|----------------|------------------------|-----------------|---------------|-------------|---|---|-------------|----------------|
| | | | | | ДП ІС-6226.00.001 ПЗ | | | |
| | | <i>Прізвище</i> | <i>Підпис</i> | <i>Дата</i> | | | | |
| <i>Розроб.</i> | <i>Халін О.І.</i> | | | | Інформаційна система підтримки діяльності балетної школи. Підсистема розробки прототипів, дизайну та клієнтської частини веб-застосунку. | <i>Літ.</i> | <i>Арк.</i> | <i>Аркушів</i> |
| | | | | | | | 2 | 113 |
| <i>Керівн.</i> | <i>Мажара О.О.</i> | | | | | <i>КПІ ім. Ігоря Сікорського Каф. АСОІУ ар. ІС-62</i> | | |
| <i>Н. кон.</i> | <i>Новінський В.П.</i> | | | | | | | |
| <i>Зате.</i> | <i>Павлов О.А.</i> | | | | | | | |

ABSTRACT

Structure and scope of the research work. There are three chapters, 8 images, 21 tables, 2 supplements, 28 sources in this explanatory note.

The individual part of the diploma is devoted to the implementation of prototypes and system design, as well as the development of the client part of the web application

The goal and objectives set out provide information on the purpose of the development of the individual part and the necessary steps for its implementation.

The section of the software introduces and describes the expediency of choosing design and development tools, sets the system requirements, and explains the architecture of the application in relation to the requirements claimed.

The technology section provides a user guide with a demonstration of the application interface and tests of the finished product for compliance with the requirements.

BALLET SCHOOL, CLIENT SIDE, INFORMATION SYSTEM, DESIGN

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 3 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ЗМІСТ

| | |
|--|-----------|
| ВСТУП | 6 |
| 1 ПОСТАНОВКА ЗАДАЧІ | 7 |
| 1.1 ПРИЗНАЧЕННЯ РОЗРОБКИ..... | 7 |
| 1.2 ЦІЛІ ТА ЗАДАЧІ РОЗРОБКИ..... | 7 |
| 2 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ | 8 |
| 2.1 ЗАСОБИ РОЗРОБКИ..... | 8 |
| 2.1.1 <i>Figma</i> | 8 |
| 2.1.2 <i>InVision</i> | 9 |
| 2.1.3 <i>Zeplin</i> | 10 |
| 2.1.4 <i>WebStorm</i> | 11 |
| 2.1.5 <i>GitLab</i> | 11 |
| 2.1.6 <i>Trello</i> | 12 |
| 2.1.7 <i>React</i> | 12 |
| 2.1.8 <i>TypeScript</i> | 13 |
| 2.1.9 <i>Apollo та GraphQL</i> | 13 |
| 2.1.10 <i>Sass та Webpack</i> | 14 |
| 2.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ..... | 14 |
| 2.2.1 <i>Загальні вимоги</i> | 14 |
| 2.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... | 15 |
| 2.3.1 <i>Діаграма класів</i> | 15 |
| 2.3.2 <i>Діаграма компонентів</i> | 15 |
| 2.3.3 <i>Діаграма варіантів використання</i> | 15 |
| 2.3.4 <i>Діаграма послідовності</i> | 15 |
| 2.3.5 <i>Діаграма розгортання</i> | 15 |
| 2.3.6 <i>Специфікація функцій</i> | 16 |
| Висновок до розділу..... | 20 |
| 3 ТЕХНОЛОГІЧНИЙ РОЗДІЛ | 21 |

| | |
|--|-----------|
| 3.1 Керівництво користувача | 21 |
| 3.2 Випробування програмного продукту..... | 27 |
| 3.2.1 Мета випробувань | 27 |
| 3.2.2 Загальні положення..... | 27 |
| 3.2.3 Результати випробувань | 28 |
| Висновок до розділу..... | 38 |
| ЗАГАЛЬНІ ВИСНОВКИ..... | 39 |
| ПЕРЕЛІК ПОСИЛАНЬ..... | 40 |
| ДОДАТОК А..... | 42 |

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 5 |

ВСТУП

У межах розробки інформаційної системи підтримки діяльності балетної школи було створено підсистему розробки прототипів, дизайну та клієнтської частини веб-застосунку. Також було запропоновано дизайн веб-застосунку та мобільного додатку.

Враховуючи, що дана підсистема відповідає саме за візуальне, перше сприйняття користувачем продукту, значну увагу було приділено роботі з замовником. Розроблялись прототипи, дизайн системи, збиралися відгуки від потенційних користувачів. На основі зауважень та пропозицій здійснювалось ітеративне покращення продукту. В подальшому виконувалась розробка клієнтської частини та її інтеграція з серверною.

Для перевірки коректності роботи даної підсистеми було проведено тестування основних функціональних можливостей. Детальний опис процесу тестування надано у розділі «Результати випробувань».

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 6 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

1 ПОСТАНОВКА ЗАДАЧІ

1.1 Призначення розробки

Призначення підсистеми - забезпечити зручний і інтуїтивно зрозумілий дизайн усього додатку, його веб та мобільної версії. Створити клієнтську частину застосунку опираючись на розроблений дизайн. Зв'язати її з серверною частиною, щоб у користувача була змога отримувати та оперувати даними.

1.2 Цілі та задачі розробки

Перед створюваною підсистемою поставлена наступна ціль – створення дизайну програмного продукту та клієнтської частини його веб версії, розробка інтерфейсу для легкого управління даними, що отримуються з серверної частини.

Для досягнення поставленої цілі необхідно було виконати наступні задачі:

- Розробити дизайн для веб-застосунку та мобільного додатку;
- Спроекувати архітектуру клієнтської частини веб-застосунку;
- Розробити усі екрани застосунку опираючись на готовий дизайн;
- Зв'язати із серверною частиною продукту;
- Підключити сервіс аналітики до застосунку;
- Рефакторинг коду з метою покращення його швидкості роботи та лаконічності;
- Провести тестування застосунку та усунення логічних та програмних помилок.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 7 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

2 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Засоби розробки

Для розробки прототипів та дизайну а також для збору зворотного зв'язку від користувачів та замовників використовувався додаток Figma[1]. Для створення інтерактивного тестування використовувався веб-застосунок InVision[2]. Для представлення і легкої роботи з дизайном використовувався додаток Zeplin[3]. Для розробки використовувалося оточення розробки JetBrains WebStorm[4] та система контролю версій GitLab[5]. Для постановки задач і контролю ходу розробку використовувалась система управління проектами Trello[6]. Для створення проектної документації використовувалися текстовий процесор Microsoft Word[7] та веб-застосування Draw.io[8].

За основу для розробки клієнтської частини додатку було використано бібліотеку від компанії Facebook – React[9], мову програмування TypeScript[10], препроцесор Sass[11], мову запитів і маніпуляції даними GraphQL[12] та платформу Apollo[13], а також постачальник модулів JavaScript – webpack[14].

2.1.1 Figma

Figma[1]- це умовно безкоштовний крос-платформний векторний онлайн-сервіс для дизайну інтерфейсів.

Даний додаток було обрано з неведених нижче причин:

- Дозволяє працювати в команді і зберігати усі зміни у вигляді версій;
- Дизайн будується у векторній формі, на відміну від того ж Adobe Photoshop[15], що дозволяє розробнику отримувати на виході зручний і легкий у використанні макет;

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 8 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

- Передбачає хмарний сервіс зберігання – усі файли зберігаються у хмарі і не виникає проблем із втратою файлів або синхронізацією у команді дизайнерів;
- Надає можливість роботи з компонентами – необхідний функціонал для зручної і швидкої роботи дизайнера, а також для подальшої розробки. Дає можливість створювати один компонент і перевикористовувати надалі у всьому дизайні, при необхідності зміни – це займає декілька хвилин роботи. Також так як розробка клієнтської частини базується на React`i[9], а він в свою чергу базується на компонентах, ця функція є надзвичайно корисною;
- Фрейми і їх налаштування – дають змогу створити адаптивний дизайн під різні розміри екрану. Також вони мають список встановлених розмірів пристроїв: ноутбуки, планшети, телефони, годинники, соціальні мережі, листи бумагами;
- Прототипування і налаштування зв'язків, наприклад перехід на іншу сторінку при натисканні на кнопку. Є надзвичайно корисною функцією представлення зрозумілої форми дизайну для тесту і збору зворотного зв'язку;
- Доступ до макету прямо з браузера, що дозволяє швидко і легко для клієнта робити тестування і випробування дизайну системи.

2.1.2 InVision

InVision[2]- це графічний редактор, однак в даному дослідженні він використовувався тільки для прототипування і роботи з екранами, а також для представлення нових версій дизайну і легкою взаємодією з нашою аудиторією, щоб зібрати необхідний зворотній зв'язок і покращити інтерфейс додатку.

Основні причини використання InVision[2] для прототипування:

- Швидкий і легкий імпорт східних матеріалів з графічного редактора;

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 9 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

- Мультиплатформеність – легкий доступ до макетів з різних пристроїв;
- Анімація інтерфейсів – напевно найпростіший спосіб створення анімацій переходів між екранами або загрузки/підгрузки контенту. Має вбудований патерн для переходів, налаштування кривих, тривалості і затримки анімацій;
- Взаємодія з аудиторією – самий корисний функціонал для нашої команди. Користувачам легко і зрозуміло залишати свої коментарі та побажання до кожної сторінки сайту або веб додатку. В той же час, розробники мають змогу швидко відреагувати на запити і доробити дизайн для подальшого тестування і розробки.

2.1.3 Zeplin

Zeplin[3]- це зручний інструмент для спільної роботи UI дизайнерів та front-end розробників. Допомогає командам при передачі дизайну. Хоч в нашій команді дизайн і клієнтську частину розробки реалізовувала одна й та сама людина, використання цього сервісу значно прискорило роботу і отримання результатів.

Основні плюси використання даного продукту:

- Звісно ж кросс-платформеність. Даний додаток можна скачати на свій комп'ютер а також відкрити в браузері;
- Зручний і швидкий імпорт початкових файлів дизайну;
- Можливість експорту всіх необхідних зображень у потрібному розмірі та розширенні;
- Інтеграція зі Slack[16], для миттєвих сповіщень про оновлення макетів дизайну;
- Усуває необхідність створення документації специфікацій, що не є швидким завданням;

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 10 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

- Усі необхідні відступи в пару наведень мишкою;
- Завдяки плагінам – корисна генерація коду sass[11] із вже налаштованими змінними.

2.1.4 WebStorm

Webstorm[4] – розумний редактор коду. Націлений в першу чергу на мову програмування JavaScript[17] та TypeScript[10], на чому і ґрунтується клієнтська частина проекту.

Використання цього редактору надає наступні основні переваги:

- Дуже корисна швидка документація по усім атрибутам;
- Допомога з виявленням граматичних і стилістичних помилок;
- Підказки і автозаповнення, що допомагає значно скоротити час на написання коду;
- Підтримка останніх версій популярних технологій;
- Корисні плагіни і вбудована інтеграція з усіма необхідними в роботі технологіями;
- Юніт-тестування – запуск і налагодження тестів безпосередньо в інтерфейсі продукту;
- Інтеграція з системою контролю версій;
- Зручний пошук використання змінних і функцій.

2.1.5 GitLab

GitLab[5]- це система керування репозиторіями програмного коду для Git. Він був розроблений українським програмістом і створювався як аналог GitHub[18], який можна було б розгорнути на своєму сервері.

Основним плюсом використання даної системи є все ж таки контроль версій і процесу розробки, з можливістю ставити задачі і відстежувати проблеми.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 11 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

2.1.6 Trello

Trello[6] – система управління проектами у вигляді канбана.

Даний продукт було обрано, так як безкоштовну зручну у використанні програму. В цій системі були визначені всі задачі і правки, а також відслідковувався увесь процес роботи команди над продуктом.

2.1.7 React

React – це JavaScript-бібліотека для створення користувацьких інтерфейсів.

Основопологаючі функції React’у[9]:

- Заснований на компонентах. Необхідно створювати компоненти, які керують власним станом, та з них створювати складні багатофункціональні інтерфейси для взаємодії з користувачами. Подібно цеглинам з яких складаються великі витвори архітектури[20];
- При зміні рендириться лише необхідна частина інтерфейсу без перезавантаження сторінки. Наприклад, меню та футер не мають змінюватись/кожного разу створюватись заново при переході між сторінками одного застосунку. Це покращує швидкість роботи доданку і дозволяє користувачу швидко взаємодіяти з інтерфейсом[20];
- Він також може, використовуючи Node.js[19], рендеритись на сервері і приводити в дію мобільні додатки, за допомогою мови React Native[20].

В розробленій системі існує досить велика кількість компонентів, що використовуються на великій кількості сторінок. Також React[9] є порівняно новим і потужним інструментом. Використовувати його в розробці і при цьому

| | | | | | | | |
|------|------|----------|--------|------|--|---------------------|------|
| | | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | | 12 |
| Змн. | Арк. | № докум. | Підпис | Дата | | | |

посилити свої навички як front-end програміста були вирішальними при виборі технологій розробки.

2.1.8 TypeScript

В поєднанні з реактом було обрано TypeScript[10] як мова розробки клієнтської частини проекту. Це сучасний інструмент, розроблений компанією Microsoft у 2012 році, який позиціонується як засіб розробки веб-застосунків, що розширює можливості JavaScript[17].

Переваги:

- Статична типізація. Через це програмний код більш передбачуваний, в ньому легше відслідковувати помилки, але й допускати їх;
- Впорядкованість проекту навіть при великих масштабах завдяки підтримці ООП, просторів імен та модулів;
- На стадії компіляції проекту можна знайти досить багато помилок ще до стадії рантайм і поломки веб-застосунку;
- Завдяки підтримки ООП є можливість створювати і наслідувати класи, а синтаксис нагадує синтаксис C#[21] або Java[22];
- Можна створювати власні типи даних;
- Код написаний на Typescript[10] компілюється в JavaScript[17].

2.1.9 Apollo та GraphQL

GraphQL[12] – це синтаксис, який описує як запитувати дані, використовується для завантаження даних з серверу[23] і має три основні характеристики:

- Полегшує агрегацію даних з різних джерел;
- Використовує систему типів для опису даних;

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 13 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

- Допомагає на клієнтській частині точно вказати, які дані необхідні і отримати їх;
- Можна створювати власні типи даних.

По своїй суті GraphQL[12] є графом і ми можемо пройти по його ребрам і дістатися саме тих вузлів, які нам необхідні. Також ми можемо фільтрувати, сортувати і агрегувати дані так, як нам це необхідно.

2.1.10 Sass та Webpack

Sass – препроцесор для швидкої розробки CSS[24] частини застосунку зі спрощеним синтаксисом і зручними змінними і міксинами.

Webpack бере модулі з залежностями і генерує статичні ресурси, які представляють ці модулі. Він як раз і дає нам змогу використовувати ECMAScript9[25] (сучасну версію JavaScript[17]), Sass[11], та багато інших сучасних і корисних в розробці інструментів.

2.2 Вимоги до технічного забезпечення

2.2.1 Загальні вимоги

Для стабільної та правильної роботи клієнтської частини застосунку необхідні наступні умови:

- Широке та стабільне підключення до мережі Інтернет;
- Пристрої вводу-виведення: монітор, клавіатура, мишка або тачпад.

Наявність одного з веб-браузерів:

- Opera 60.0+;
- Internet Explorer 11+;
- Google Chrome 76.0+;
- Mozilla Firefox 68.0+;

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 14 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

– Safari.

2.3 Архітектура програмного забезпечення

2.3.1 Діаграма класів

Діаграму класів наведено у графічних матеріалах.

Діаграма класів описує схему графового матеріалу. У деяких класів присутні методи, які дозволяють визначити необхідні поля за певною логікою.

2.3.2 Діаграма компонентів

Діаграму компонентів наведено у графічних матеріалах.

На діаграмі компонентів відображені серверна та клієнтські частини застосунку з внутрішніми складовими.

2.3.3 Діаграма варіантів використання

Діаграму варіантів використання наведено у графічних матеріалах.

На діаграмі варіантів використання вказані актори системи та набір дій, які вони можуть виконувати. Система налічує 4 акторів: користувач, студент, тренер, адміністратор.

2.3.4 Діаграма послідовності

Діаграми послідовності описують процеси авторизації існуючого користувача та реєстрацію нового. Діаграми наведені у графічних матеріалах.

2.3.5 Діаграма розгортання

Діаграму розгортання наведено у графічних матеріалах.

На діаграмі розгортання вказані обчислювальні вузли системи і програмні компоненти, що виконуються на цих вузлах.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 15 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

2.3.6 Специфікація функцій

Специфікація функцій коду, задіяних у роботі програми наведено у таблиці 2.1.

Таблиця 2.1 – Перелік основних функцій

| Назва | Параметри | Примітка |
|-------------------------------|---|--|
| MainMenuWrapper() | | Слугує оболонкою меню. Ініціалізує і оперує стейтами для відображення нотифікацій і відкриття меню на мобільці |
| MainMenuVisible() | isUnreadNotification onRequestToggleNavbar onRequestToggleNotifications isNavbarOpen | Створює видиму частину меню з контейнером для посилань і кнопками для взаємодії на мобільних пристроях |
| MainMenuOpenButton() | isActive onClick | Створює кнопку для відкривання/закривання меню на мобільних пристроях |
| MainMenuNotificationsButton() | isActive onClick | Створює кнопку для відкривання/закривання сповіщень на мобільних пристроях |
| MainMenuNavigationBar() | isActive | Створює контейнер з посиланнями для переходу по пунктам меню на різні сторінки застосунку |
| MainMenuNavigationBarButton() | | Створює пункти меню – посилання для навігації по застосунку |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Продовження таблиці 2.1

| | | |
|---------------------------------------|--|---|
| MainMenuLogoutButton() | onClick | Створює пункт меню для виходу із застосунку |
| MainMenuNotificationsBar() | isActive | Створює контейнер для перегляду трьох останніх сповіщень |
| MainMenuNotificationsItem() | body date time onClick | Створює картку сповіщення з тілом повідомлення, датою и часом |
| MainMenuNotificationsShowMoreButton() | onClick | Створює кнопку для відображення усіх сповіщень |
| Overlay() | isVisible zIndex onClick | Створює оверлей – затемнення при відкриванні сповіщень і меню на мобільних додатках |
| ScheduleContainerWrapper() | date time address length | Створює контейнер для сторінки розкладу |
| ScheduleContainer() | isActive dayOfWeek date lessonsAmount children | Створює контейнер для карток розкладу із заголовком |
| ScheduleLesson() | time address firstLesson didntPay studentsAmount | Створює картку заняття з адресою, і всією необхідною інформацією |
| LessonVisitAnalytics() | notPaid trialLesson willAbsent className | Створює елемент детальної інформації по заняттю |

Продовження таблиці 2.1

| | | |
|----------------------------|--|--|
| ScheduleDay() | isActive time groupLessons personalLessons | Створює картку дня або місяцю з часом, і інформацією про кількість занять даного дня |
| OuterTeacherLessonCard() | title date time location duration level | Створює картку наступного заняття на сторінці розкладу з необхідною інформацією |
| TeacherLessonDetailsCard() | title date time location duration level | Створює картку наступного заняття з необхідною інформацією |
| ScheduleViewModeBar() | | Створює перемикач групування занять по дням, тижням, місяцям |
| ScheduleViewModeButton() | | Створює елемент для вибору групування |
| ScheduleViewModePerson() | | Створює картку особистого кабінету з інформацією про увійшовшого користувача |
| BaseCoachCabinet() | | Створює базу для особистого кабінету тренера |
| ArrowToTop() | isVisible onClick | Створює стрілку вгору |
| AddComment() | onSubmit | Створює форму для додавання коментарів на карточці студента |
| CoachLessonDetails() | | Створює сторінку поточного заняття |
| CoachPersonalPage() | | Створює сторінку розкладу |

Продовження таблиці 2.1

| | | |
|------------------------------------|---|---|
| LessonDetail StudentCard () | avatarUrl firstName lastName paymentInfo | Створює картку студента з аватаркою, ім'ям і всією необхідною інформацією |
|------------------------------------|---|---|

Висновок до розділу

У розділі програмного і технічного забезпечення обґрунтовуються засоби розробки, вимоги до технічного забезпечення та наведено архітектуру системи.

В якості програмного забезпечення було обрано ряд продуктів: Figma[1] для розробки прототипів і дизайну застосунку, InVision[2] для створення інтерактивних прототипів та тестування гіпотез, Zeplin[3] для зв'язки дизайну і front-end частини, для зручності розробки, WebStorm[4] як IDE для безпосередньої розробки клієнтської частини веб-застосунку, GitLab[5] для керування версіями та Trello[6] для керування задачами та роботи команди.

В якості технологій клієнтської частини додатку було обрано сучасну бібліотеку React[9] як зручну та водночас швидкому програмування для веб-систем. Основною мовою було обрано TypeScript[10], для створення потужних веб-додатків з чіткою типізацією, а також sass як препроцесор для розробки стилів компонентів системи та webpack як постачальник модулів JavaScript[17].

Діаграми були створені у програмному продукті draw.io[8] та зображені у графічному розділі. Вони дають уявлення про архітектуру продукту, основних акторів та принципи роботи програмного продукту. У разі передачі проекту іншій команді розробників, вони надають чітке пояснення зв'язків у проекті. Також наведені усі функції клієнтської частини застосунку з відповідними параметрами і описом їх роботи.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 20 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

3 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

3.1 Керівництво користувача

Даний розділ є інструкцією для користувача. В ньому описано, як використовувати даний продукт та що буде при тому чи іншому сценарії використання.

Вікна входу та реєстрації

Вікна входу та реєстрації – вікна для авторизації користувача щоб отримати доступ до системи. Авторизуватися можна за допомогою соціальних мереж або номеру телефону. Вікно реєстрації містить наступні поля: ім'я, прізвище, номер телефону. Вікно входу – номер телефону і пароль. Поле номеру телефону має маску для коректного заповнення. Після авторизації на номер телефону прийде код для входу в особистий кабінет. На рисунку 3.1 показано вікно реєстрацію із полями для вводу даних, а на рисунку 3.2 – вікно підтвердження коду з СМС.

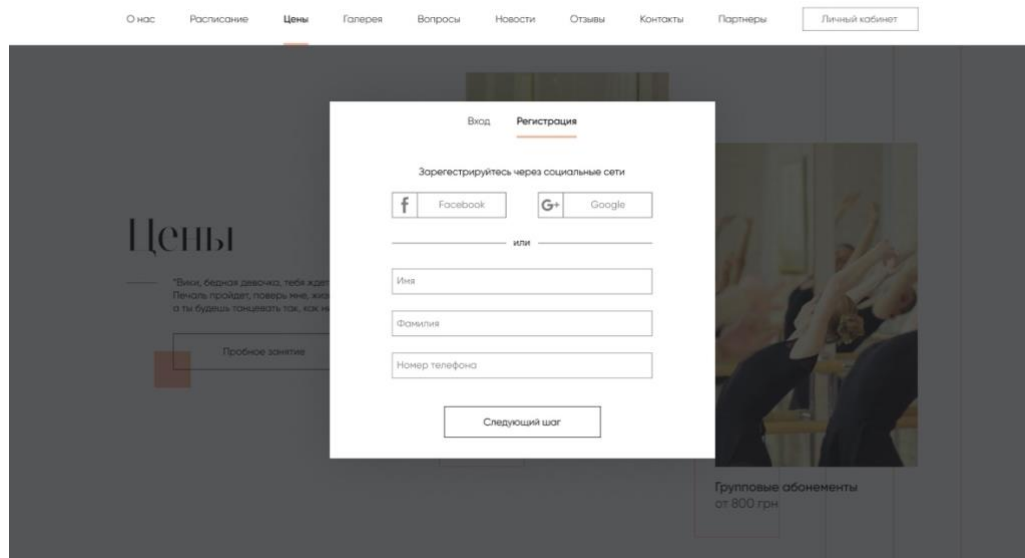


Рисунок 3.1 – Вікно реєстрації

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 21 |

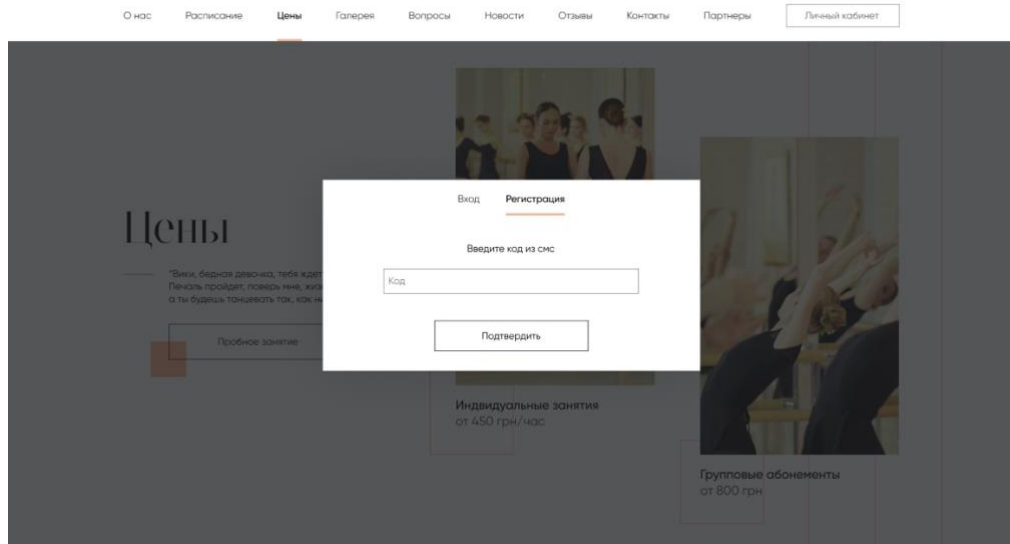


Рисунок 3.2 – Вікно підтвердження коду з СМС

Вікно розкладу занять

Після входу до особистого кабінету користувач побачить його найближче заняття за розкладом та його розклад групований по місяцям, тижням або дням. Розклад по місяцям наведено на рисунку 3.3. Там можна побачити найближчі заняття починаючи з поточного місяця. На карточці тижня можна бачити безпосередньо дати початку і кінця місяця, кількість групових, індивідуальних та загальну кількість занять. При кліці на карточку користувач потрапить на розклад групований по тижням з вибраним тижнем в фокусі.

Розклад по тижням наведено на рисунку 3.4. Там можна побачити найближчі заняття, починаючи з поточного тижня, у вигляді днів і групованих по тижням. На карточці дня можна бачити безпосередньо дату і день тижня, кількість групових, індивідуальних та загальна кількість занять. При кліці на карточку користувач потрапить на розклад групований по дням з вибраним днем в фокусі. Також можна бачити загальну кількість занять у тижні.

Розклад по дням наведено на рисунку 3.5. Там можна побачити найближчі заняття починаючи з поточного дня у вигляді занять і групованих по дням. На карточці заняття можна бачити час початку, адресу, кількість людей, які повинні бути присутні із загальної кількості місць, кількість людей,

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 22 |

які повинні зробити оплату, для яких дане заняття – пробне і які зазвичай ходять, але на даному занятті не будуть присутні. Також можна бачити загальну кількість занять обраного дня.

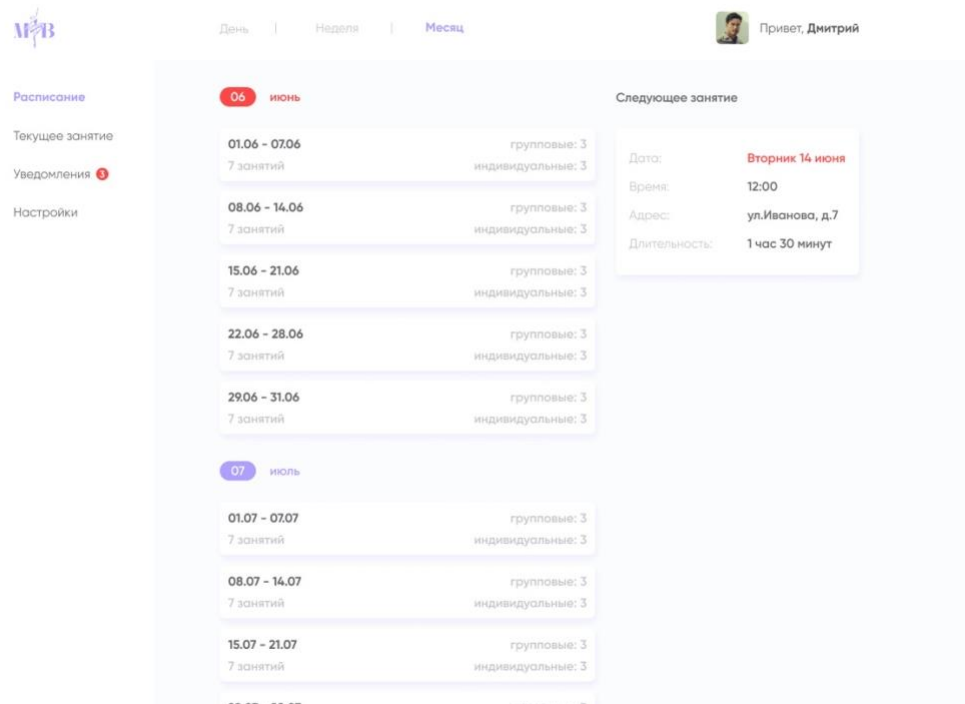


Рисунок 3.3 – Розклад групованих по місяцям

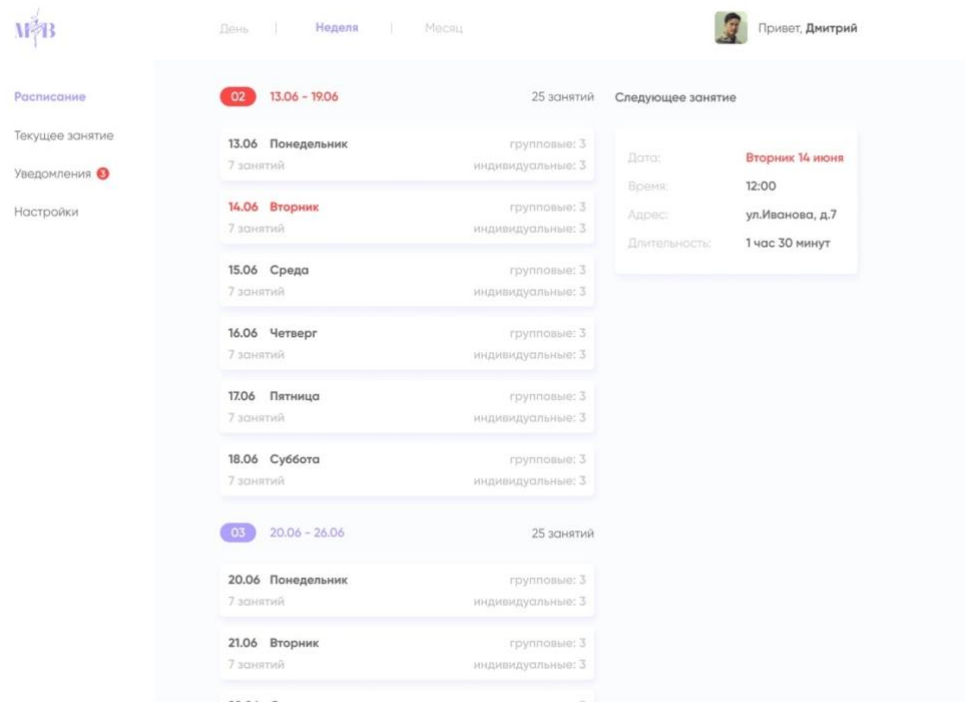


Рисунок 3.4 – Розклад групованих по тижням

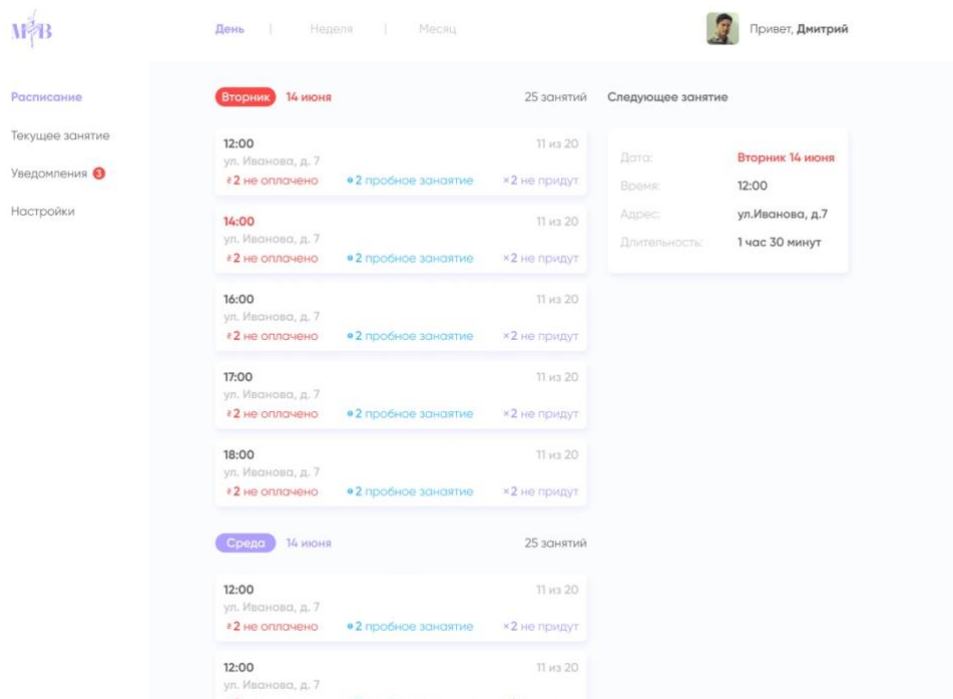


Рисунок 3.5 – Розклад групованих занять по дням

Вікно з інформацією про заняття

При натисканні на карточку заняття користувач потрапить на вікно з детальною інформацією. Тут користувач зможе побачити: дату, час, адресу і тривалість заняття, кількість людей, які повинні оплатити заняття, для яких це заняття буде пробним і які на даному занятті не будуть присутні. Також перелік студентів, які записані на дане заняття. На карточці студента видно наступну інформацію: фото, ім'я та прізвище, чи оплачено дане заняття, якщо ні – то яку суму він має оплатити, показуються коментарі та можливість додати свій коментар, також видно чи є це заняття пробним для студента балетної школи. Демонстрацію даного екрану можна побачити на рисунку 3.6:

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 24 |

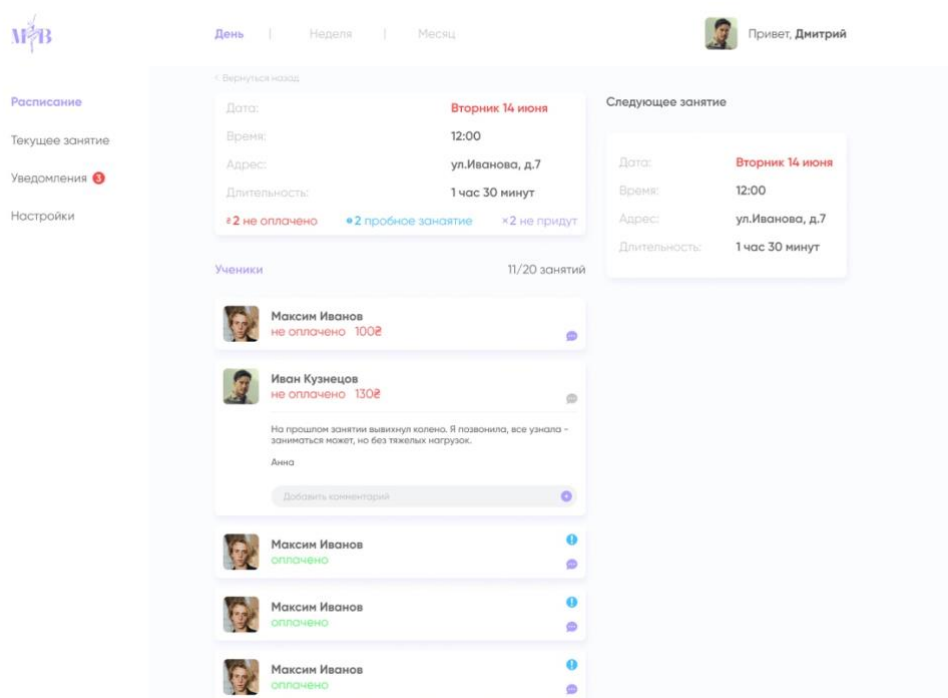


Рисунок 3.6 – Вікно з інформацією про заняття

Вікно з інформацією про поточне заняттям

При натисканні на пункт меню «Поточне заняття» користувач потрапить на вікно з детальною інформацією. Тут користувач зможе побачити: дату, час, адресу і тривалість заняття, кількість людей, які повинні оплатити заняття, для яких це заняття буде пробним і які на даному заняття не будуть присутні. Також перелік студентів, які записані на дане заняття. На карточці студента видно наступну інформацію: фото, ім'я та прізвище, чи оплачено дане заняття, якщо ні – то яку суму він має оплатити, показуються коментарі та можливість додати свій коментар, видно чи є це заняття пробним для студента балетної школи, також є дві кнопки для відмічання відвідування і для відмітки з оплати, якщо такі необхідні. Демонстрацію даного екрану можна побачити на рисунку 3.7:



Заняття почнеться через 05:00



Привет, Дмитрий

Расписание

Текущее занятие

Уведомления 3

Настройки

< Вернуться назад

Дата: **Вторник 14 июня**
 Время: 12:00
 Адрес: ул.Иванова, д.7
 Длительность: 1 час 30 минут
 +2 не оплачено • 2 пробное занятие × 2 не придут

Следующее занятие

Дата: **Вторник 14 июня**
 Время: 12:00
 Адрес: ул.Иванова, д.7
 Длительность: 1 час 30 минут

Ученики 11/20 занятий

Максим Иванов
не оплачено 100₽

Максим Иванов
не оплачено 100₽

Максим Иванов
оплачено

Максим Иванов
оплачено

Максим Иванов

Рисунок 3.7 – Вікно з інформацією про поточне заняття

Вікно сповіщення

При натисканні на пункт меню «Сповіщення» користувач побачить впливаюче вікно зі списком його непереглянутих повідомлень. На карточці повідомлення буде видно сам текст повідомлення і його час/дату. Демонстрацію даного екрану представлено на рисунку 3.8:

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

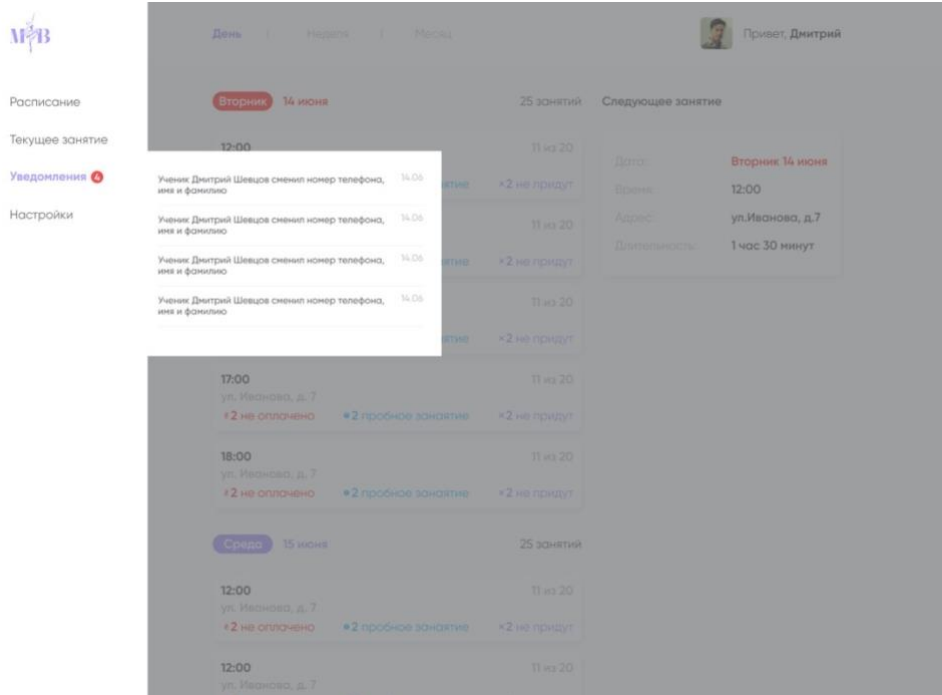


Рисунок 3.8 – Вікно сповіщення

3.2 Випробування програмного продукту

В цьому підрозділі наведено опис тестів і порядок їх виконання для перевірки відповідності програмного забезпечення комплексу задач функціональним вимогам, представленим у технічному завданні на створення інформаційної системи підтримки діяльності балетної школи.

3.2.1 Мета випробувань

Метою випробувань є перевірка відповідності функцій інформаційної системи підтримки діяльності балетної школи.

3.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603-92. Інформаційна технологія. Види випробувань автоматизованих систем;

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 27 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

– ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

3.2.3 Результати випробувань

В процесі тестування була перевірена уся функціональність комплексу задач (КЗ). У наступних таблицях наведений перелік випробувань основних функціональних можливостей (таблиці 3.1-3.20).

Таблиця 3.1 – Перевірка заповнення поля «Номер телефону» при реєстрації

| | |
|---------------------------------------|--|
| Мета тесту: | Перевірка функції «Реєстрація користувача» |
| Початковий стан КЗ | Відкрите вікно реєстрації |
| Вхідні данні: | 1. Ім'я «name»; 2. Прізвище «surname». |
| Схема проведення тесту: | 1. Ввести в поле «Ім'я» значення «name»; 2. Ввести в поле «Прізвище» значення «surname»; 3. Натиснути кнопку «Наступний крок». |
| Очікуваний результат: | Виведено повідомлення «Поле «Номер телефону» обов'язкове для заповнення» |
| Стан КЗ після проведення випробувань: | Виведено повідомлення «Поле «Номер телефону» обов'язкове для заповнення» |

Таблиця 3.2 – Перевірка унікальності введених реєстраційних даних

| | |
|--------------------|---|
| Мета тесту: | Перевірка функції «Реєстрація користувача» |
| Початковий стан КЗ | Відкрите вікно реєстрації. В системі зареєстрований користувач з номером телефону «phone» |
| Вхідні данні: | 1. Ім'я «name»; 2. Прізвище «surname»; 3. Номер телефону «phone». |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Продовження таблиці 3.2

| | |
|---------------------------------------|---|
| Схема проведення тесту: | <ol style="list-style-type: none"> 1. Ввести в поле «Ім'я» значення «name»; 2. Ввести в поле «Прізвище» значення «surname»; 3. Ввести в поле «Номер телефону» значення «phone»; 4. Натиснути кнопку «Наступний крок». |
| Очікуваний результат: | Виведено повідомлення «Користувач з таким номером телефону вже зареєстрований» |
| Стан КЗ після проведення випробувань: | Виведено повідомлення «Користувач з таким номером телефону вже зареєстрований» |

Таблиця 3.3 – Перевірка коректності вводу коду з СМС при реєстрації

| | |
|---------------------------------------|---|
| Мета тесту: | Перевірка функції «Реєстрація користувача» |
| Початковий стан КЗ | Відкрите вікно вводу коду з СМС. В системі не зареєстрований користувач з номером телефону «phone». На номер телефону користувача прийшло повідомлення з кодом «code» |
| Вхідні данні: | <ol style="list-style-type: none"> 1. Ім'я «name»; 2. Прізвище «surname»; 3. Номер телефону «phone»; 4. Код «code2». |
| Схема проведення тесту: | <ol style="list-style-type: none"> 1. Ввести в поле «Ім'я» значення «name»; 2. Ввести в поле «Прізвище» значення «surname»; 3. Ввести в поле «Номер телефону» значення «phone»; 4. Натиснути кнопку «Наступний крок»; 5. Ввести в поле «Код» значення «code2»; 6. Натиснути кнопку «Підтвердити». |
| Очікуваний результат: | Виведено повідомлення «Введено невірний код з СМС» |
| Стан КЗ після проведення випробувань: | Виведено повідомлення «Введено невірний код з СМС» |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Таблиця 3.4 – Перевірка переходу на вікно «Вхід» після успішної реєстрації

| | |
|---------------------------------------|--|
| Мета тесту: | Перевірка функції «Реєстрація користувача» |
| Початковий стан КЗ | Відкрите вікно реєстрації. В системі не зареєстрований користувач з номером телефону «phone». На номер телефону користувача прийшло повідомлення з кодом «code» |
| Вхідні данні: | <ol style="list-style-type: none"> 1. Ім'я «name»; 2. Прізвище «surname»; 3. Номер телефону «phone»; 4. Код «code». |
| Схема проведення тесту: | <ol style="list-style-type: none"> 1. Ввести в поле «Ім'я» значення «name»; 2. Ввести в поле «Прізвище» значення «surname»; 3. Ввести в поле «Номер телефону» значення «phone»; 4. Натиснути кнопку «Наступний крок»; 5. Ввести в поле «Код» значення «code»; 6. Натиснути кнопку «Підтвердити». |
| Очікуваний результат: | Відкрито вікно «Вхід» |
| Стан КЗ після проведення випробувань: | Відкрито вікно «Вхід» |

Таблиця 3.5 – Перевірка заповнення поля «Номер телефону» при вході в систему

| | |
|---------------------------------------|---|
| Мета тесту: | Перевірка функції «Вхід» |
| Початковий стан КЗ | Відкрите вікно входу |
| Вхідні данні: | Пароль «password» |
| Схема проведення тесту: | <ol style="list-style-type: none"> 1. Ввести в поле «Пароль» значення «password»; 2. Натиснути кнопку «Увійти». |
| Очікуваний результат: | Виведено повідомлення «Поле «Номер телефону» обов'язкове для заповнення» |
| Стан КЗ після проведення випробувань: | Виведено повідомлення «Поле «Номер телефону» обов'язкове для заповнення» |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Таблиця 3.6 – Перевірка заповнення поля «Пароль» при вході в систему

| | |
|---------------------------------------|--|
| Мета тесту: | Перевірка функції «Вхід» |
| Початковий стан КЗ | Відкрите вікно входу |
| Вхідні данні: | Номер телефону «phone» |
| Схема проведення тесту: | <ol style="list-style-type: none"> 1. Ввести в поле «Номер телефону» значення «phone»; 2. Натиснути кнопку «Увійти». |
| Очікуваний результат: | Виведено повідомлення «Поле «Пароль» обов'язкове для заповнення» |
| Стан КЗ після проведення випробувань: | Виведено повідомлення «Поле «Пароль» обов'язкове для заповнення» |

Таблиця 3.7 – Перевірка коректності заповнення поля «Номер телефону» при вході в систему

| | |
|---------------------------------------|--|
| Мета тесту: | Перевірка функції «Вхід» |
| Початковий стан КЗ | Відкрите вікно входу. В системі не зареєстрований користувач з номером телефону «phone» |
| Вхідні данні: | <ol style="list-style-type: none"> 1. Номер телефону «phone»; 2. Пароль «password». |
| Схема проведення тесту: | <ol style="list-style-type: none"> 1. Ввести в поле «Номер телефону» значення «phone»; 2. Ввести в поле «Пароль» значення «password»; 3. Натиснути кнопку «Увійти». |
| Очікуваний результат: | Виведено повідомлення «Некоректно введені дані. Користувача з таким номером телефона і паролем не існує» |
| Стан КЗ після проведення випробувань: | Виведено повідомлення «Некоректно введені дані. Користувача з таким номером телефона і паролем не існує» |

Таблиця 3.8 – Перевірка коректності заповнення поля «Пароль» при вході в систему

| Мета тесту: | Перевірка функції «Вхід» |
|---------------------------------------|---|
| Початковий стан КЗ | Відкрите вікно входу. В системі зареєстрований користувач з номером телефону «phone» та паролем «password» |
| Вхідні данні: | <ol style="list-style-type: none"> 1. Номер телефону «phone»; 2. Пароль «password2». |
| Схема проведення тесту: | <ol style="list-style-type: none"> 1. Ввести в поле «Номер телефону» значення «phone»; 2. Ввести в поле «Пароль» значення «password2»; 3. Натиснути кнопку «Увійти». |
| Очікуваний результат: | Виведено повідомлення «Некоректно введені дані. Користувача з таким номером телефона і паролем не існує» |
| Стан КЗ після проведення випробувань: | Виведено повідомлення «Некоректно введені дані. Користувача з таким номером телефона і паролем не існує» |

Таблиця 3.9 – Перевірка переходу на головну сторінку з розкладом групованим по дням після успішного входу в систему

| Мета тесту: | Перевірка функції «Вхід» |
|---------------------------------------|--|
| Початковий стан КЗ | Відкрите вікно входу. В системі зареєстрований користувач з номером телефону «phone» та паролем «password» |
| Вхідні данні: | <ol style="list-style-type: none"> 1. Номер телефону «phone»; 2. Пароль «password». |
| Схема проведення тесту: | <ol style="list-style-type: none"> 1. Ввести в поле «Номер телефону» значення «phone»; 2. Ввести в поле «Пароль» значення «password»; 3. Натиснути кнопку «Увійти». |
| Очікуваний результат: | Відкрито сторінку з розкладом групованим по дням |
| Стан КЗ після проведення випробувань: | Відкрито сторінку з розкладом групованим по дням |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Таблиця 3.10 – Перевірка заповнення поля «Номер телефону» при відновленні паролю

| | |
|---------------------------------------|--|
| Мета тесту: | Перевірка функції «Відновлення паролю» |
| Початковий стан КЗ | Відкрите вікно відновлення паролю |
| Вхідні данні: | |
| Схема проведення тесту: | Натиснути кнопку «Наступний крок» |
| Очікуваний результат: | Виведено повідомлення «Поле «Номер телефону» обов'язкове для заповнення» |
| Стан КЗ після проведення випробувань: | Виведено повідомлення «Поле «Номер телефону» обов'язкове для заповнення» |

Таблиця 3.11 – Перевірка коректності заповнення поля «Номер телефону» при відновленні паролю

| | |
|---------------------------------------|--|
| Мета тесту: | Перевірка функції «Відновлення паролю» |
| Початковий стан КЗ | Відкрите вікно відновлення паролю. В системі не зареєстрований користувач з номером телефону «phone» |
| Вхідні данні: | Номер телефону «phone» |
| Схема проведення тесту: | <ol style="list-style-type: none"> 1. Ввести в поле «Номер телефону» значення «phone»; 2. Натиснути кнопку «Наступний крок». |
| Очікуваний результат: | Виведено повідомлення «Некоректно введені дані. Користувача з таким номером телефона не існує» |
| Стан КЗ після проведення випробувань: | Виведено повідомлення «Некоректно введені дані. Користувача з таким номером телефона не існує» |

Таблиця 3.12 – Перевірка коректності вводу коду з СМС при відновленні паролю

| | |
|--------------------|--|
| Мета тесту: | Перевірка функції «Відновлення паролю» |
| Початковий стан КЗ | Відкрите вікно вводу коду з СМС. В системі зареєстрований користувач з номером телефону «phone». На номер телефону користувача прийшло повідомлення з кодом «code» |

Продовження таблиці 3.12

| | |
|---------------------------------------|--|
| Вхідні данні: | 1. Номер телефону «phone»; 2. Код з СМС «code2». |
| Схема проведення тесту: | 1. Ввести в поле «Номер телефону» значення «phone»; 2. Натиснути кнопку «Наступний крок»; 3. Ввести в поле «Код» значення «code2»; 4. Натиснути кнопку «Підтвердити». |
| Очікуваний результат: | Виведено повідомлення «Введено невірний код з СМС» |
| Стан КЗ після проведення випробувань: | Виведено повідомлення «Введено невірний код з СМС» |

Таблиця 3.13 – Перевірка переходу на вікно «Вхід» після успішного відновлення паролю

| | |
|---------------------------------------|---|
| Мета тесту: | Перевірка функції «Відновлення паролю» |
| Початковий стан КЗ | Відкрите вікно відновлення паролю. В системі зареєстрований користувач з номером телефону «phone». На номер телефону користувача прийшло повідомлення з кодом «code» |
| Вхідні данні: | 1. Номер телефону «phone»; 2. Код з СМС «code». |
| Схема проведення тесту: | 1. Ввести в поле «Номер телефону» значення «phone»; 2. Натиснути кнопку «Наступний крок»; 3. Ввести в поле «Код» значення «code»; 4. Натиснути кнопку «Підтвердити». |
| Очікуваний результат: | Відкрито вікно «Вхід» |
| Стан КЗ після проведення випробувань: | Відкрито вікно «Вхід» |

Таблиця 3.14 – Перевірка роботи пункту меню «Поточне заняття»

| | |
|--------------------|---|
| Мета тесту: | Перевірка роботи пункту меню «Поточне заняття» |
| Початковий стан КЗ | Відкрито сторінку з розкладом групованим по дням |

Продовження таблиці 3.14

| | |
|---------------------------------------|---|
| Вхідні данні: | |
| Схема проведення тесту: | Натиснути пункт меню «Поточне заняття» |
| Очікуваний результат: | Відкрито сторінку з інформацією про поточне заняття |
| Стан КЗ після проведення випробувань: | Відкрито сторінку з інформацією про поточне заняття |

Таблиця 3.15 – Перевірка роботи пункту меню «Повідомлення»

| | |
|---------------------------------------|--|
| Мета тесту: | Перевірка роботи пункту меню «Повідомлення» |
| Початковий стан КЗ | Відкрито сторінку з розкладом групованим по дням |
| Вхідні данні: | |
| Схема проведення тесту: | Натиснути пункт меню «Повідомлення» |
| Очікуваний результат: | Відкрито сторінку зі списком повідомлень |
| Стан КЗ після проведення випробувань: | Відкрито сторінку зі списком повідомлень |

Таблиця 3.16 – Перевірка роботи пункту меню «Розклад»

| | |
|---------------------------------------|--|
| Мета тесту: | Перевірка роботи пункту меню «Розклад» |
| Початковий стан КЗ | Відкрито сторінку з повідомленнями |
| Вхідні данні: | |
| Схема проведення тесту: | Натиснути пункт меню «Розклад» |
| Очікуваний результат: | Відкрито сторінку з розкладом групованим по дням |
| Стан КЗ після проведення випробувань: | Відкрито сторінку з розкладом групованим по дням |

Таблиця 3.17 – Перевірка роботи групування розкладу по місяцям

| | |
|--------------------|--|
| Мета тесту: | Перевірка роботи групування розкладу по місяцям |
| Початковий стан КЗ | Відкрито сторінку з розкладом групованим по дням |
| Вхідні данні: | |

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Продовження таблиці 3.17

| | |
|---------------------------------------|---|
| Схема проведення тесту: | Натиснути підпункт меню «Місяць» |
| Очікуваний результат: | Відкрито сторінку з розкладом групованим по місяцям |
| Стан КЗ після проведення випробувань: | Відкрито сторінку з розкладом групованим по місяцям |

Таблиця 3.18 – Перевірка роботи групування розкладу по тижням

| | |
|---------------------------------------|---|
| Мета тесту: | Перевірка роботи групування розкладу по тижням |
| Початковий стан КЗ | Відкрито сторінку з розкладом групованим по дням |
| Вхідні данні: | |
| Схема проведення тесту: | Натиснути підпункт меню «Тиждень» |
| Очікуваний результат: | Відкрито сторінку з розкладом групованим по тижням |
| Стан КЗ після проведення випробувань: | Відкрито сторінку з розкладом групованим по тижням |

Таблиця 3.19 – Перевірка роботи відмічання присутності

| | |
|---------------------------------------|---|
| Мета тесту: | Перевірка роботи відмічання присутності |
| Початковий стан КЗ | Відкрито сторінку з поточним заняттям |
| Вхідні данні: | |
| Схема проведення тесту: | Натиснути кнопку «Відвідування» |
| Очікуваний результат: | Кнопка «Відвідування» змінила свій стан на активний, що демонструє зміна її кольору |
| Стан КЗ після проведення випробувань: | Кнопка «Відвідування» змінила свій стан на активний, що демонструє зміна її кольору |

Таблиця 3.20 – Перевірка роботи відмічання оплати

| | |
|-------------------------|---|
| Мета тесту: | Перевірка роботи відмічання оплати |
| Початковий стан КЗ | Відкрито сторінку з поточним заняттям |
| Вхідні данні: | |
| Схема проведення тесту: | Натиснути кнопку «Оплата» |

Продовження таблиці 3.20

| | |
|---------------------------------------|--|
| Очікуваний результат: | Кнопка «Оплата» змінила свій стан на активний, що демонструє зміна її кольору Текст «Не оплачено» змінився на «Оплачено» і змінилось забарвлення з червоного на зелений |
| Стан КЗ після проведення випробувань: | Кнопка «Оплата» змінила свій стан на активний, що демонструє зміна її кольору Текст «Не оплачено» змінився на «Оплачено» і змінилось забарвлення з червоного на зелений |

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 37 |

Висновок до розділу

В цьому розділі в таблицях наведено перелік випробувань програмного продукту, а саме функції реєстрації, авторизації в системі та встановлення паролю. Описана робота основних функцій навігації і зміну стану відвідувачів занять балетної школи.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 38 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ЗАГАЛЬНІ ВИСНОВКИ

У дипломному проекті реалізовано клієнтської частини веб-застосунку системи інформаційної підтримки балетної школи. Окремо вирішено підзадачі розробки дизайну та прототипних версій, а також організації ефективної комунікації з клієнтом, а саме мережею балетних шкіл для дорослих “MaryBallet”. В ході роботи над проектом було проведено аналіз діяльності компаній та визначено вимоги до розроблюваної системи. Виконано огляд аналогів подібних рішень. Представлено прототипні рішення системи у чорно-білому варіанті і у вигляді макетів дизайну сайту. На основі зворотного зв’язку з користувачем було виправлено недоліки дизайну. Обґрунтовано методи та засоби реалізації.

Крім вирішення безпосередньо технічних завдань, було розроблено процедуру взаємодії з користувачем та всередині команди. Обрано та впроваджено технологічні рішення, які допомагають організувати роботу над проектом та звітування щодо етапів виконання.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 39 |

ПЕРЕЛІК ПОСИЛАНЬ

1. Figma[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.figma.com/>;
2. InVision[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:https://www.invisionapp.com;
3. Zeplin[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://zeplin.io>;
4. WebStorm[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.jetbrains.com/ru-ru/webstorm/>;
5. GitLab[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://gitlab.com>;
6. Trello[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://trello.com>;
7. MicrosoftWord[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.microsoft.com/uk-ua/microsoft-365/word>;
8. Draw.io[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<http://draw.io>;
9. React[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://uk.reactjs.org>;
10. TypeScript[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.typescriptlang.org>;
11. Apollo[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.apollographql.com>;
12. GraphQL[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://graphql.org>;
13. SASS[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://sass-lang.com>;
14. Webpack[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://webpack.js.org>;

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 40 |

15. AdobePhotoshop[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.adobe.com/ua/products/photoshop.html>;
16. Slack[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://slack.com/>;
17. JavaScript[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://developer.mozilla.org/uk/docs/Web/JavaScript>;
18. Github[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://github.com>;
19. Node.js[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://nodejs.org/en/>;
20. Документація з використання React [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://en.reactjs.org/docs>;
21. C#[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://docs.microsoft.com/ru-ru/dotnet/csharp/>;
22. Java[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.java.com/ru/>;
23. CSS[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<http://htmlbook.ru/css>;
24. So what’s this GraphQL thing I keep hearing about?[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.freecodecamp.org/news/so-whats-this-graphql-thing-i-keep-hearing-about-baf4d36c20cf/>;
25. EcmaScript 9[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.ecma-international.org/ecma-262/9.0/index.html>
26. John Resig, Bear Bibeault, and Josip Maras. Secrets of the JavaScript Ninja, Second Edition. 2016;
27. Ethan Brown. Learning JavaScript: JavaScript Essentials for Modern Application Development, 3rd Edition. 2016;
28. Alex Banks, Eve Porcello. Learning React: Functional Web Development with React and Redux 1st Edition. 2017.

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 41 |

Додаток А

Тексти програмного коду

Інформаційна система підтримки діяльності балетної школи.

*Підсистема розробки прототипів, дизайну та клієнтської частини
веб-застосунку.*

(Найменування програми (документа))

DVD-R

(Вид носія даних)

196 арк., 242 600Кб

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2020 року

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 42 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import {
  BrowserRouter,
  Route,
  Redirect,
  Switch,
} from 'react-router-dom';

import { withApolloHooksProvider } from '@client';
import CoachPersonalPage from '@pages/CoachPersonalPage';
import {
  teacherCabinet,
  teacherCabinetSchedule,
  teacherCurrentLesson,
  teacherDaysSchedule,
  teacherLessonDetail,
} from '@urls';
import CoachLessonDetails from '@pages/CoachLessonDetails';

const AppRouter: React.FC = () => (
  <BrowserRouter>
  <Switch>
    { /* Schedule */ }
  <Route
    exact={true}
    path={teacherCabinetSchedule}
    component={() =><Redirect to={teacherDaysSchedule} />}
  />
  <Route
    path={teacherCabinetSchedule}
    component={CoachPersonalPage}
  />

    { /* Lesson details */ }
  <Route
    path={[
      teacherLessonDetail,
      teacherCurrentLesson,
    ]}
    component={CoachLessonDetails}
  />

  <Route
    path={teacherCabinet}
    component={() =><Redirect to={teacherDaysSchedule} />}
  />
  </Switch>
</BrowserRouter>
);

const AppRouterWithHooks = withApolloHooksProvider(AppRouter);

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 43 |

```

ReactDOM.render(
  <AppRouterWithHooks />,
  document.querySelector('#app')
);

import * as React from 'react';

import { ApolloClient } from 'apollo-client';
import { InMemoryCache } from 'apollo-cache-inmemory';
import { HttpLink } from 'apollo-link-http';
import { onError } from 'apollo-link-error';
import { ApolloLink } from 'apollo-link';
import { ApolloProvider as ApolloHooksProvider } from '@apollo/react-hooks';

const client = new ApolloClient({
  link: ApolloLink.from([
    onError(({ graphQLErrors, networkError }) => {
      if (graphQLErrors) {
        graphQLErrors.map(
          ({message, locations, path}) =>
            console.log(
              `[GraphQL error]: Message: ${message}, Location: ${locations}, Path: ${path}`,
            ),
        );
      }
      if (networkError) {
        console.log(`[Network error]: ${networkError}`);
      }
    }
  ]),
  new HttpLink({
    uri: 'https://maryballet.com.ua/api/graph',
    credentials: 'same-origin'
  })
  ],
  cache: new InMemoryCache()
});

export const withApolloHooksProvider = <T extends {}>(Component: React.ComponentType<T>) => (
  (props: T) => (
    <ApolloHooksProvider client={client}>
    <Component {...props} />
    </ApolloHooksProvider>
  )
);

import * as React from 'react';

import ScheduleViewModeBar from '@components/ScheduleViewModeBar';
import ScheduleContainerWrapper from '@components/ScheduleContainerWrapper';
import MainMenuWrapper from '@components/MainMenuWrapper';

import BaseCoachCabinet from '@pages/BaseCoachCabinet';

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 44 |

```

import * as s from './CoachPersonalPage.sss';

export const CoachPersonalPage: React.FC = () => {

  return (
    <BaseCoachCabinet className={s.root}>
    <MainMenuWrapper/>

    <ScheduleContainerWrapper
      date="14 июня, Вторник"
      time="12:00"
      address="ул.Иванова, д.7"
      length="1 час 30 минут"
    />

    <ScheduleViewModeBar/>
    </BaseCoachCabinet>
  );
};

export default CoachPersonalPage;

import * as React from 'react';
import * as classNames from 'classnames';

// import ArrowToTop from '@components/ArrowToTop';

import * as s from './BaseCoachCabinet.sss';

type BaseCoachCabinetProps = {
  outerClassName?: string
  className?: string
};

export const BaseCoachCabinet: React.FC<BaseCoachCabinetProps> = ({
  outerClassName,
  className,
  children,
}) => (
  <div className={classNames(s.root, outerClassName)}>
  <div className={classNames(s.content, className)}>
    {children}
  </div>
  { /*<ArrowToTop/>*/ }
  </div>
);

export default BaseCoachCabinet;

import * as React from 'react';
import * as classNames from 'classnames';

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

import MainMenuVisible from '@components/MainMenuVisible';
import MainMenuNavigationBar from '@components/MainMenuNavigationBar';
import MainMenuNotificationsBar from '@components/MainMenuNotificationsBar';
import Overlay from '@components/Overlay';

type MainMenuWrapperProps = {
  isActive?: boolean
  className?: string
};

import * as s from './MainMenuWrapper.sss';

export const MainMenuWrapper: React.FC<MainMenuWrapperProps> = ({
  className,
}) => {
  const [isNavbarOpen, setIsNavbarOpen] = React.useState(false);
  const [isNotificationOpen, setIsNotificationOpen] = React.useState(false);

  return (
<div className={classNames(s.root, className)}>
<MainMenuVisible
  isNavbarOpen={isNavbarOpen}
  onRequestToggleNavbar={() => {
    setIsNavbarOpen(!isNavbarOpen);
    if (isNotificationOpen) {
      setIsNotificationOpen(!isNotificationOpen);
    }
  }}
  onRequestToggleNotifications={() => {
    setIsNotificationOpen(!isNotificationOpen);
    if (isNavbarOpen) {
      setIsNavbarOpen(!isNavbarOpen);
    }
  }}
/>
<MainMenuNavigationBar isActive={isNavbarOpen}/>
<MainMenuNotificationsBar isActive={isNotificationOpen}/>
<Overlay
  onClick={() => {
    if (isNavbarOpen) {
      setIsNavbarOpen(!isNavbarOpen);
    } else {
      setIsNotificationOpen(!isNotificationOpen);
    }
  }}
  zIndex={ 9000 }
  isVisible={ isNavbarOpen || isNotificationOpen }
/>
</div>
);
};

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 46 |

```

export default MainMenuWrapper;

import * as React from 'react';
import * as classNames from 'classnames';

import * as s from './MainMenuVisible.sss';

import GridContainer from './GridContainer';
import MainMenuOpenButton from './MainMenuOpenButton';
import MainMenuNotificationsButton from './MainMenuNotificationsButton';

type MainMenuVisibleProps = {
  isUnreadNotification?: boolean
  className?: string
  onRequestToggleNavbar: () => void
  onRequestToggleNotifications: () => void
  isNavbarOpen: boolean
};

export const MainMenuVisible: React.FC<MainMenuVisibleProps> = ({
  isUnreadNotification = true,
  className,
  onRequestToggleNavbar,
  onRequestToggleNotifications,
  isNavbarOpen
}) => (
  <div className={classNames(s.mainMenuVisible, className)}>
    <GridContainer>
    <div className={classNames(s.mainMenuVisibleInner)}>
      <MainMenuOpenButton
        isActive={isNavbarOpen}
        onClick={onRequestToggleNavbar}
      />
      <span>{ isNavbarOpen ? 'Меню' : 'Мое расписание' }</span>
      <MainMenuNotificationsButton
        isActive={true}
        onClick={onRequestToggleNotifications}
      />
    </div>
  </GridContainer>
</div>
);

export default MainMenuVisible;

import * as React from 'react';
import * as classNames from 'classnames';

import * as s from './GridContainer.sss';

type GridContainerProps = {

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 47 |

```

    className?: string
  };

  export const GridContainer: React.FC<GridContainerProps> = ({
    className,
    children
  }) => (
    <div
      className={classNames(s.gridContainer, className)}
    >
      {children}
    </div>
  );

  export default GridContainer;

  import * as React from 'react';
  import * as classNames from 'classnames';

  import * as s from './MainMenuOpenButton.sss';

  type MainMenuOpenButtonProps = {
    onClick: () => void
    isActive: boolean
    className?: string
  };

  export const MainMenuOpenButton: React.FC<MainMenuOpenButtonProps> = ({
    className,
    isActive = true,
    onClick,
  }) => (
    <div
      className={
        classNames(
          s.mainMenuOpenButton,
          {[s.active]: isActive}
        )
      }
      onClick={onClick}
    >
      <span></span>
    </div>
  );

  export default MainMenuOpenButton;

  import * as React from 'react';
  import * as classNames from 'classnames';

  import * as s from './MainMenuNotificationsButton.sss';

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

import * as SvgMainMenuNotificationsButton from './static/SvgMainMenuNotificationsButton.svg';

type MainMenuNotificationsButtonProps = {
  onClick?: () => void
  isActive: boolean
  className?: string
};

export const MainMenuNotificationsButton: React.FC<MainMenuNotificationsButtonProps> = ({
  className,
  isActive = true,
  onClick,
}) => (
<div
  className={
    classNames(
      s.mainMenuNotificationsButton,
      {[s.active]: isActive}
    )
  }
  onClick={onClick}
  dangerouslySetInnerHTML={{ __html: SvgMainMenuNotificationsButton }}
>
</div>
);

export default MainMenuNotificationsButton;

import * as React from 'react';
import * as classNames from 'classnames';

import GridContainer from '@components/GridContainer';
import MainMenuNavigationButton from '@components/MainMenuNavigationButton';
import MainMenuLogoutButton from '@components/MainMenuLogoutButton';
import {
  teacherCabinetSchedule,
  teacherCurrentLesson,
  teacherSupport
} from '@urls';

import * as SvgLogo from './static/SvgLogo.svg';

import * as s from './MainMenuNavigationBar.sss';

type MainMenuNavigationBarProps = {
  isActive?: boolean
  className?: string
};

export const MainMenuNavigationBar: React.FC<MainMenuNavigationBarProps> = ({
  isActive = true,
  className,

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 49 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

    }) => (
    <div
      className={
        classNames(
          s.mainMenuNavigationBar,
          {[s.active]: isActive},
          className
        )
      }
    >
    <GridContainer className={s.maxHeight}>
    <div className={s.mainMenuNavigationWrap}>
    <div className={s.mainMenuNavigationButtonWrap}>
    <div
      className={s.logo}
      dangerouslySetInnerHTML={{ __html: SvgLogo }}
    />
    <MainMenuNavigationButton to={teacherCabinetSchedule}>Мое расписание</MainMenuNavigationButton>
    <MainMenuNavigationButton to={teacherCurrentLesson}>Текущее занятие</MainMenuNavigationButton>
    <MainMenuNavigationButton
      to={'#'}
      style={{ color: '#ccc' }}
    >
      Уведомления
    </MainMenuNavigationButton>
    <MainMenuNavigationButton to={teacherSupport}>Поддержка</MainMenuNavigationButton>
    </div>
    <MainMenuLogoutButton>Выход из аккаунта</MainMenuLogoutButton>
    </div>
    </GridContainer>
    </div>
  );

  export default MainMenuNavigationBar;

  import * as React from 'react';
  import * as classNames from 'classnames';
  import {
    NavLink,
    NavLinkProps
  } from 'react-router-dom';

  import * as s from './MainMenuNavigationButton.sss';

  type MainMenuNavigationButtonProps = NavLinkProps;

  export const MainMenuNavigationButton: React.FC<MainMenuNavigationButtonProps> = ({
    className,
    children,
    ...props
  }) => (
    <NavLink

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

        className={classNames(s.root, className)}
        activeClassName={s.active}
        {...props}
    >
        {children}
    </NavLink>
);

export default MainMenuNavigationButton;

import * as React from 'react';
import * as classNames from 'classnames';

import * as s from './MainMenuLogoutButton.sss';

type MainMenuLogoutButtonProps = {
    onClick?: () => void
    className?: string
};

export const MainMenuLogoutButton: React.FC<MainMenuLogoutButtonProps> = ({
    className,
    onClick,
    children
}) => (
    <div
        className={
            classNames(
                s.mainMenuLogoutButton,
                className
            )
        }
    >
        {children}
    </div>
);

export default MainMenuLogoutButton;

import * as React from 'react';
import * as classNames from 'classnames';

import GridContainer from '@components/GridContainer';
import MainMenuNotificationsItem from '@components/MainMenuNotificationsItem';
import MainMenuNotificationsShowMoreButton from '@components/MainMenuNotificationsShowMoreButton';

import * as s from './MainMenuNotificationsBar.sss';

type MainMenuNotificationsBarProps = {
    isActive?: boolean
    className?: string
};

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 51 |

```

export const MainMenuNotificationsBar: React.FC<MainMenuNotificationsBarProps> = ({
  isActive = true,
  className,
}) => (
  <div
    className={
      classNames(
        s.root,
        {[s.active]: isActive},
        className
      )
    }
  >
    <GridContainer>
    <div className={classNames(s.rootWrap)}>
    <h3>Новые</h3>
    <MainMenuNotificationsItem
      body="Ученик Дмитрий Шевцов сменил номер телефона, имя и фамилию"
      date="14.06"
      time="12:30"
    />
    <MainMenuNotificationsItem
      body="Ученик Дмитрий Шевцов сменил номер телефона, имя и фамилию"
      date="14.06"
      time="12:30"
    />
    <MainMenuNotificationsItem
      body="Ученик Дмитрий Шевцов сменил номер телефона, имя и фамилию"
      date="14.06"
      time="12:30"
    />
    <MainMenuNotificationsShowMoreButton>Смотреть все</MainMenuNotificationsShowMoreButton>
  </div>
</GridContainer>
</div>
);

export default MainMenuNotificationsBar;

import * as React from 'react';
import * as classNames from 'classnames';

import * as s from './MainMenuNotificationsItem.sss';

type MainMenuNotificationsItemProps = {
  onClick?: () => void
  className?: string
  body: string
  date: string
  time: string
};

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 52 |

```

export const MainMenuNotificationsItem: React.FC<MainMenuNotificationsItemProps> = ({
  className,
  onClick,
  body,
  date,
  time
}) => (
  <div
    className={
      classNames(
        s.mainMenuNotificationsItem,
        className
      )
    }
  >
    <div className={s.mainMenuNotificationsItemBody}>{body}</div>
    <div className={s.mainMenuNotificationsItemDateTime}>
      <div className={s.mainMenuNotificationsItemDate}>{date}</div>
      <div className={s.mainMenuNotificationsItemTime}>{time}</div>
    </div>
  </div>
);

export default MainMenuNotificationsItem;

import * as React from 'react';
import * as classNames from 'classnames';

import * as s from './MainMenuNotificationsShowMoreButton.sss';

type MainMenuNotificationsShowMoreButtonProps = {
  onClick?: () => void
  className?: string
};

export const MainMenuNotificationsShowMoreButton: React.FC<MainMenuNotificationsShowMoreButtonProps> = ({
  className,
  onClick,
  children
}) => (
  <div
    className={
      classNames(
        s.mainMenuNotificationsShowMoreButton,
        className
      )
    }
  >
    {children}
  </div>
);

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 53 |

```

export default MainMenuNotificationsShowMoreButton;

import * as React from 'react';
import ReactDOM from 'react-dom';

import * as classNames from 'classnames';

import * as s from './Overlay.sss';

type OverlayProps = {
  isVisible: boolean
  zIndex?: number
  onClick: () => void
  className?: string
};

export const Overlay: React.FC<OverlayProps> = ({
  isVisible,
  className,
  zIndex = 1,
  onClick,
}) => {
  const component = (
    <div className={
      classNames(
        s.overlay,
        {[s.active]: isVisible},
        className
      )
    }
    style={{ zIndex }}
    onClick={ onClick }
  />
  );

  return ReactDOM.createPortal(
    component,
    document.body
  );
};

export default Overlay;

import * as React from 'react';
import * as classNames from 'classnames';
import { Route } from 'react-router-dom';

import ScheduleContainer from '@components/ScheduleContainer';
import ScheduleLesson from '@components/ScheduleLesson';
import ScheduleDay from '@components/ScheduleDay';
import OuterTitleTeacherLessonCard from '@components/OuterTitleTeacherLessonCard';

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

import {
  teacherDaysSchedule,
  teacherMonthsSchedule,
  teacherWeeksSchedule
} from '@urls';

import * as s from './ScheduleContainerWrapper.sss';

type ScheduleContainerWrapperProps = {
  className?: string
  date: string
  time: string
  address: string
  length: string
};

export const ScheduleContainerWrapper: React.FC<ScheduleContainerWrapperProps> = ({
  className,
  date,
  time,
  address,
  length
}) => (
  <div className={classNames(s.root, className)}>
    <div className={s.schedules}>
      <Route
        path={teacherDaysSchedule}
        component={() => (
          <ScheduleContainer dayOfWeek="Вт" date="14 июня" lessonsAmount={2}>
            <ScheduleLesson
              time="12:00"
              address="ул. Иванова, д. 7"
              firstLesson={2}
              didntPay={2}
              studentsAmount={11}
            />
            <ScheduleLesson
              time="13:30"
              address="ул. Крупской, д. 17"
              firstLesson={0}
              didntPay={4}
              studentsAmount={13}
            />
          </ScheduleContainer>
        )}
      />

      <Route
        path={teacherWeeksSchedule}
        component={() => (
          <ScheduleContainer dayOfWeek="03" date="20 - 26 июня" lessonsAmount={25}>

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 55 |

```

<ScheduleDay
  time="Пн 20.06"
  groupLessons={3}
  personalLessons={4}
/>

<ScheduleDay
  time="Вт 21.06"
  groupLessons={3}
  personalLessons={0}
/>

<ScheduleDay
  time="Ср 22.06"
  groupLessons={0}
  personalLessons={4}
/>

</ScheduleContainer>
  )}
/>

<Route
  path={teacherMonthsSchedule}
  component={() => (
<ScheduleContainer dayOfWeek="07" date="июль" lessonsAmount={38}>
<ScheduleDay
  time="01.07 - 07.07"
  groupLessons={3}
  personalLessons={4}
/>

<ScheduleDay
  time="08.07 - 14.07"
  groupLessons={3}
  personalLessons={0}
/>

<ScheduleDay
  time="15.07 - 21.07"
  groupLessons={0}
  personalLessons={4}
/>

</ScheduleContainer>
  )}
/>
</div>

<OuterTitleTeacherLessonCard
  className={s.currentLesson}
  title="Следующее занятие"
  date={date}
  time={time}
  location={address}
  duration={length.length}
  level={1}
/>
</div>

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

);

export default ScheduleContainerWrapper;

import * as React from 'react';
import * as classNames from 'classnames';

import * as s from './ScheduleContainer.sss';

type ScheduleContainerProps = {
  className?: string
  isActive?: boolean
  dayOfWeek: string
  date: string
  lessonsAmount: number
};

export const ScheduleContainer: React.FC<ScheduleContainerProps> = ({
  className,
  isActive = false,
  dayOfWeek,
  date,
  lessonsAmount,
  children
}) => (
<div
  className={
    classNames(
      s.scheduleContainer,
      {[s.active]: isActive},
      className
    )
  }
>
<div className={s.scheduleContainerHeader}>
<div>
<span className={s.scheduleContainerDayOfWeek}>
  {dayOfWeek}
</span>
<span className={s.scheduleContainerDate}>{date}</span>
</div>
<div className={s.scheduleContainerLessonsCount}>
  {lessonsAmount} заняття
</div>
</div>
  {children}
</div>
);

export default ScheduleContainer;

import * as React from 'react';

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

import * as classNames from 'classnames';

import LessonVisitAnalytics from '@components/LessonVisitAnalytics';

import * as s from './ScheduleLesson.sss';

type ScheduleLessonProps = {
  className?: string
  time: string
  address: string
  firstLesson: number
  didntPay: number
  studentsAmount: number
};

export const ScheduleLesson: React.FC<ScheduleLessonProps> = ({
  className,
  time,
  address,
  firstLesson,
  didntPay,
  studentsAmount
}) => {
  return(
    <div className={classNames(s.scheduleLesson, className)}>
      <div className={s.scheduleLessonTime}>
        {time}
      </div>
      <LessonVisitAnalytics
        className={s.scheduleLessonInfo}
        notPaid={didntPay}
        trialLesson={firstLesson}
        willAbsent={0}
      />
      <div className={s.scheduleLessonAddress}>
        {address}
      </div>
      <div className={s.scheduleLessonPeople}>
        {studentsAmount} чел.
      </div>
    </div>
  );
};

export default ScheduleLesson;

import * as React from 'react';
import * as classNames from 'classnames';

import * as Close from '@static/svg/Close.svg';
import * as ExclamationMark from '@static/svg/ExclamationMark.svg';
import * as UAH from '@static/svg/UAH.svg';

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 58 |

```

import * as s from './LessonVisitAnalytics.sss';

type VisitTypeInfoProps = {
  icon: string
  count: number
  className?: string
};

type LessonVisitAnalyticsProps = {
  notPaid: number
  trialLesson: number
  willAbsent: number
  className?: string
};

export const VisitTypeInfo: React.FC<VisitTypeInfoProps> = ({
  icon,
  count,
  children,
  className,
}) => (
  <div className={classNames(s.visitTypeInfo, className)}>
    <div
      className={s.icon}
      dangerouslySetInnerHTML={{ __html: icon }}
    />
    <span>{count}</span>
    <span className={s.tabletOrWider}>{children}</span>
  </div>
);

export const LessonVisitAnalytics: React.FC<LessonVisitAnalyticsProps> = ({
  notPaid,
  trialLesson,
  willAbsent,
  className,
}) => (
  <div className={className}>
    <VisitTypeInfo
      className={s.red}
      icon={UAH}
      count={notPaid}
    >
      не оплачено
    </VisitTypeInfo>
    <VisitTypeInfo
      className={s.blue}
      icon={ExclamationMark}
      count={trialLesson}
    >
      пробное занятие
    </VisitTypeInfo>
  </div>
);

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 59 |

```

<VisitTypeInfo
  className={s.fiolet}
  icon={Close}
  count={willAbsent}
>
  не придут
</VisitTypeInfo>
</div>
);

export default LessonVisitAnalytics;

import * as React from 'react';
import * as classNames from 'classnames';

import * as s from './ScheduleDay.sss';

type ScheduleDayProps = {
  className?: string
  isActive?: boolean
  time: string
  groupLessons: number
  personalLessons: number
};

export const ScheduleDay: React.FC<ScheduleDayProps> = ({
  className,
  time,
  isActive = false,
  groupLessons,
  personalLessons,
}) => {

  return(
<div
  className={
    classNames(
      s.scheduleDay,
      className
    )
  }
>
<div
  className={
    classNames(
      s.scheduleDayTime,
      {[s.active]: isActive}
    )
  }
>
    {time}
</div>

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

    {groupLessons !== 0 &&
    <div
      className={
        s.scheduleDayInfoGroup
      }
    >
      групповые: {groupLessons}
    </div>
    }
    {personalLessons !== 0 &&
    <div
      className={
        classNames(
          s.scheduleDayInfoPersonal,
          {[s.scheduleDayInfoPersonalSwap]: groupLessons === 0}
        )
      }
    >
      индивидуальные: {personalLessons}
    </div>
    }
    <div
      className={
        s.scheduleDayInfo
      }
    >
      { groupLessons !== 0 &&<div>{groupLessons} групп</div>}
      { (groupLessons !== 0 && personalLessons !== 0) &&<span>/</span> }
      { personalLessons !== 0 &&<div>{personalLessons} инд</div> }
    </div>
    <div
      className={
        s.scheduleDaySumUp
      }
    >
      {groupLessons + personalLessons} занятий
    </div>
  </div>
);
};

export default ScheduleDay;

import * as React from 'react';

import {
  TeacherLessonDetailsCard,
  TeacherLessonDetailsCardProps,
} from '@components/TeacherLessonDetailsCard';

import * as s from './OuterTitleTeacherLessonCard.sss';

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

type OuterTitleTeacherLessonCardProps = TeacherLessonDetailsCardProps & {
  title: string
  className?: string
};

export const OuterTitleTeacherLessonCard: React.FC<OuterTitleTeacherLessonCardProps> = ({
  title,
  className,
  ...props
}) => {
  return (
    <div className={className}>
      <div className={s.title}>{title}</div>
      <TeacherLessonDetailsCard {...props} />
    </div>
  );
};

export default OuterTitleTeacherLessonCard;

import * as React from 'react';
import * as classNames from 'classnames';

import ScheduleViewModeButton from '@components/ScheduleViewModeButton';
import ScheduleViewModePerson from '@components/ScheduleViewModePerson';
import GridContainer from '@components/GridContainer';
import {
  teacherDaysSchedule,
  teacherWeeksSchedule,
  teacherMonthsSchedule,
} from '@urls';

import * as s from './ScheduleViewModeBar.sss';

type ScheduleViewModeButtonProps = {
  className?: string
};

export const ScheduleViewModeBar: React.FC<ScheduleViewModeButtonProps> = ({
  className,
}) => (
  <div className={classNames(s.root, className)}>
    <GridContainer>
      <div className={classNames(s.scheduleViewModeWrap)}>
        <div className={classNames(s.scheduleViewModeButtons)}>
          <ScheduleViewModeButton to={teacherDaysSchedule}>День</ScheduleViewModeButton>
          <ScheduleViewModeButton to={teacherWeeksSchedule}>Неделя</ScheduleViewModeButton>
          <ScheduleViewModeButton to={teacherMonthsSchedule}>Месяц</ScheduleViewModeButton>
        </div>
        <ScheduleViewModePerson />
      </div>
    </GridContainer>
  )

```

| | | | | |
|------|------|-----------|--------|------|
| | | | | |
| Змн. | Арк. | № док.ум. | Підпис | Дата |

```

</div>
);

export default ScheduleViewModeBar;

import * as React from 'react';
import * as classNames from 'classnames';
import {
  NavLink,
  NavLinkProps
} from 'react-router-dom';

import * as s from './ScheduleViewModeButton.sss';

type ScheduleViewModeButtonProps = NavLinkProps;

export const ScheduleViewModeButton: React.FC<ScheduleViewModeButtonProps> = ({
  className,
  children,
  ...props
}) => (
<NavLink
  {...props}
  className={classNames(s.root, className)}
  activeClassName={s.active}
>
  {children}
</NavLink>
);

export default ScheduleViewModeButton;

import * as React from 'react';
import * as classNames from 'classnames';

import MainMenuNotificationsButton from '@components/MainMenuNotificationsButton';

import * as s from './ScheduleViewModePerson.sss';

type ScheduleViewModePersonProps = {
  className?: string
};

export const ScheduleViewModePerson: React.FC<ScheduleViewModePersonProps> = ({
  className,
}) => (
<div
  className={
    classNames(
      s.scheduleViewModePerson,
      className
    )
  }

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

    }
  >
  
  <div>Привет, <span>Дмитрий</span></div>
  <MainMenuNotificationsButton
    isActive={true}
  />
</div>
);

export default ScheduleViewModePerson;

import * as React from 'react';
import * as classNames from 'classnames';

import * as s from './MainMenuNotificationsButton.sss';

import * as SvgMainMenuNotificationsButton from './static/SvgMainMenuNotificationsButton.svg';

type MainMenuNotificationsButtonProps = {
  onClick?: () => void
  isActive: boolean
  className?: string
};

export const MainMenuNotificationsButton: React.FC<MainMenuNotificationsButtonProps> = ({
  className,
  isActive = true,
  onClick,
}) => (
<div
  className={
    classNames(
      s.mainMenuNotificationsButton,
      {[s.active]: isActive}
    )
  }
  onClick={onClick}
  dangerouslySetInnerHTML={{ __html: SvgMainMenuNotificationsButton }}
>
</div>
);

export default MainMenuNotificationsButton;

import * as React from 'react';

import BaseCoachCabinet from '@pages/BaseCoachCabinet';
import ScheduleViewModeBar from '@components/ScheduleViewModeBar';
import MainMenuWrapper from '@components/MainMenuWrapper';
import TeacherLessonDetailsCard from '@components/TeacherLessonDetailsCard';

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 64 |

```

importLessonDetailStudentCardfrom '@components/LessonDetailStudentCard';
importLessonVisitAnalyticsfrom '@components/LessonVisitAnalytics';
importLessonDetailStudentsListfrom '@components/LessonDetailStudentsList/LessonDetailStudentsList';
importOuterTitleTeacherLessonCardfrom '@components/OuterTitleTeacherLessonCard';
importTeacherLessonFullStudentAnalyticsfrom '@components/TeacherLessonFullStudentAnalytics';
import { PaymentSource } from '@constants';

import * as s from './CoachLessonDetails.sss';

constCoachLessonDetails: React.FC = () => {
conststudents = [
<LessonDetailStudentCard
  avatarUrl={'https://i.pinimg.com/originals/5e/81/37/5e81371c890a88377dd2195a99d3deba.jpg'}
  firstName={'Вікторія'}
  lastName={'Глушко'}
  paymentInfo={{
    paid: true,
    source: PaymentSource.Online,
    amount: 100
  }}
  isFirstVisit={true}
  newMessages={true}
  />,
<LessonDetailStudentCard
  avatarUrl={'https://i.pinimg.com/originals/5e/81/37/5e81371c890a88377dd2195a99d3deba.jpg'}
  firstName={'Вікторія'}
  lastName={'Глушко'}
  paymentInfo={{
    paid: true,
    source: PaymentSource.OnLesson,
    amount: 100
  }}
  isFirstVisit={true}
  newMessages={true}
  />,
<LessonDetailStudentCard
  avatarUrl={'https://i.pinimg.com/originals/5e/81/37/5e81371c890a88377dd2195a99d3deba.jpg'}
  firstName={'Вікторія'}
  lastName={'Глушко'}
  paymentInfo={{
    paid: false,
    source: null,
    amount: 300
  }}
  isFirstVisit={true}
  newMessages={true}
  />,
<LessonDetailStudentCard
  avatarUrl={'https://i.pinimg.com/originals/5e/81/37/5e81371c890a88377dd2195a99d3deba.jpg'}
  firstName={'Вікторія'}
  lastName={'Глушко'}
  paymentInfo={{

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

paid: false,
source: null,
amount: 145
  }}
isFirstVisit={true}
newMessages={true}
  />
];

return (
<BaseCoachCabinetclassName={s.root}>
<MainMenuWrapper/>

<div className={s.lessonInfo}>
<TeacherLessonDetailsCard
className={s.lesson }
date={'14 июня, Вторник'}
time={'12:00'}
location={'ул.Иванова, д.7'}
duration={90}
level={1}
>
<LessonVisitAnalytics
className={s.visitAnalytics }
notPaid={4}
trialLesson={3}
willAbsent={0}
  />
</TeacherLessonDetailsCard>
<LessonDetailStudentsList
className={s.studentsList }
cards={students}
analytics={
<TeacherLessonFullStudentAnalytics
inAll={20}
shouldPresent={11}
notPaid={3}
trialLesson={2}
willAbsent={9}
  />
}
  />
</div>

<OuterTitleTeacherLessonCard
className={s.nextLesson }
title="Следующеезанятие"
date={'14 июня, Вторник'}
time={'12:00'}
location={'ул.Иванова, д.7'}
duration={90}
level={1}

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

    />
<ScheduleViewModeBar/>
</BaseCoachCabinet>
);
};

exportdefaultCoachLessonDetails;

import * as React from 'react';
import * as classNames from 'classnames';

import { PaymentSource } from '@constants';
// import { AddComment } from '@components/AddComment/AddComment';

import * as SvgDialog from '@static/svg/Dialog.svg';
import * as SvgExclamationMark from '@static/svg/ExclamationMark.svg';

import * as s from './LessonDetailStudentCard.sss';

typePaidPaymentInfo = {
  paid: true
  source: PaymentSource.Online | PaymentSource.OnLesson
  amount: number
};
typeNotPaidPaymentInfo = {
  paid: false
  source: null
  amount: number
};

typeLessonDetailStudentCardProps = {
  avatarUrl: string
  firstName: string
  lastName: string
  paymentInfo?: PaidPaymentInfo | NotPaidPaymentInfo
  isFirstVisit: boolean
  newMessages: boolean
  className?: string
};

exportconstLessonDetailStudentCard: React.FC<LessonDetailStudentCardProps> = ({
  avatarUrl,
  firstName,
  lastName,
  paymentInfo,
  className,
}) => {
  letpaymentInfoComponent;
  if (paymentInfo) {
    if (!paymentInfo.paid) {
      paymentInfoComponent = (
        <div className={s.notPaid}>

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 67 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        не оплачено {paymentInfo.amount}
    </div>
    );
    } elseif (paymentInfo.source === PaymentSource.Online) {
    paymentInfoComponent = (
    <div className={s.paidOnline}>
    оплачено
    </div>
    );
    } elseif (paymentInfo.source === PaymentSource.OnLesson) {
    paymentInfoComponent = (
    <div className={s.paidOnLesson}>
    оплачено
    </div>
    );
    }
    }

    return (
    <div className={classNames(s.root, className)}>
    <div className={s.profile}>
    <div className={s.leftSide}>
    <div className={s.avatarOuter}>
    <img
    className={s.avatar}
    src={avatarUrl}
    alt={` ${firstName} ${lastName}`}
    />
    </div>
    <div className={s.info}>
    <div className={s.fullName}>{firstName} {lastName}</div>
    {paymentInfoComponent}
    </div>
    </div>
    <div className={s.rightSide}>
    <div
    className={s.icon}
    dangerouslySetInnerHTML={{ __html: SvgExclamationMark }}
    />

    <div
    className={s.icon}
    dangerouslySetInnerHTML={{ __html: SvgDialog }}
    />
    </div>
    </div>
    <div className={s.comments}>
    { /*<AddCommentonSubmit={console.log} />*/ }
    </div>
    </div>
    );
    };

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

export default LessonDetailStudentCard;

import * as React from 'react';
import * as classNames from 'classnames';

import * as s from './TeacherLessonDetailsCard.sss';

export type TeacherLessonDetailsCardProps = {
  date: string
  time: string
  location: string
  duration: number
  level: 1 | 2 | 3 | null
  className?: string
};

export const TeacherLessonDetailsCard: React.FC<TeacherLessonDetailsCardProps> = ({
  date,
  time,
  location,
  duration,
  level,
  className,
  children,
}) => {
  return (
    <div className={classNames(s.root, className)}>
      <div className={s.row}>
        <span>Дата:</span>
        <span>{date}</span>
      </div>
      <div className={s.row}>
        <span>Время:</span>
        <span>{time}</span>
      </div>
      <div className={s.row}>
        <span>Адрес:</span>
        <span>{location}</span>
      </div>
      <div className={s.row}>
        <span>Продолжит:</span>
        <span>{duration}</span>
      </div>
      <div className={s.row}>
        <span>Уровень:</span>
        <span>{level}</span>
      </div>
      {children}
    </div>
  );
};

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

export default TeacherLessonDetailsCard;

import * as React from 'react';
import * as classNames from 'classnames';

import * as s from './ArrowToTop.sss';
import * as SvgArrowToTop from './static/SvgArrowToTop.svg';

type ArrowToTopProps = {
  isVisible?: boolean
  onClick?: () => void
  className?: string
};

export const ArrowToTop: React.FC<ArrowToTopProps> = ({
  isVisible = true,
  className,
  onClick,
}) => (
  <div
    className={
      classNames(
        s.arrowToTop,
        className,
        { [s.active]: isVisible }
      )
    }
    onClick={onClick}
    dangerouslySetInnerHTML={{ __html: SvgArrowToTop }}
  />
);

export default ArrowToTop;

.arrowToTop
position: fixed
width: 36px
height: 36px
border-radius: 50%
box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.1)
background-color: #ffffff
bottom: 55px
cursor: pointer
transition: background-color .3s, right .3s
display: flex
justify-content: center
align-items: center
right: -46px
svg
width: 13px
path

```

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.001 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 70 |

```

stroke: $fiolet
&:hover, &:active
background-color: $fiolet
svg
path
stroke: $white

.active
right: 10px

@media ($min-tablet)
  .arrowToTop
bottom: 70px
  .active
right: 45px

.gridContainer
margin: 0 auto
max-width: 290px
position: relative
display: flex
width: 100%

.maxHeight
height: 100%

@media ($min-tablet)
  .gridContainer
max-width: 678px

@media ($min-desktop)
  .gridContainer
max-width: 1100px

.root
box-shadow: 2px 6px 10px rgba(176, 158, 255, 0.15)
border-radius: 7px
padding: 12px

.profile
display: flex
justify-content: space-between

.leftSide
display: flex

.avatarOuter
height: 52px
width: 52px
border-radius: 7px

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

overflow: hidden
display: flex
align-items: center

.avatar
width: 100%
height: auto

.info
margin-left: 32px
display: flex
flex-direction: column

.fullName
color: \$darkGray
font-weight: 600

.paidOnline
color: \$blue

.paidOnLesson
color: \$green

.notPaid
color: \$red

.rightSide
display: flex
flex-direction: column
justify-content: space-between

.icon
svg
height: 14px
width: 14px

.header
display: flex
justify-content: space-between

.title
font-size: 18px
color: \$fiolet
font-weight: 500

.cards
margin-top: 30px

&> *
margin-bottom: 9px

.visitTypeInfo

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 72 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

display: flex

.icon

padding-top: 4px

margin-right: 2px

svg

height: 11px

width: 11px

.red

color: \$red

fill: \$red

.blue

color: \$blue

fill: \$blue

.fiolet

color: \$fiolet

fill: \$fiolet

.tabletOrWider

display: none

@media (\$min-tablet)

.tabletOrWider

display: block

margin-left: 4px

@media (\$min-desktop)

.icon

padding-top: 7px

.mainMenuLogoutButton

display: block

font-size: 14px

font-weight: 500

color: \$red

text-transform: uppercase

padding-top: 15px

padding-bottom: 15px

width: 100%

transition: color .3s

cursor: pointer

&:hover, &:active, &.active

color: \$red

@media (\$min-desktop)

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 73 |

.mainMenuLogoutButton

display: none

.mainMenuNavigationBar

height: calc(100vh - 37px)

display: flex

flex-direction: column

justify-content: space-between

position: absolute

background: \$white

bottom: calc(100% - 37px)

box-sizing: border-box

transition: transform .5s ease

transform: none

z-index: -1

width: 100%

padding-top: 20px

padding-bottom: 15px

&.active

transform: translateY(calc(100%))

transition-delay: .2s

.logo

display: none

.mainMenuNavigationButtonWrap

display: flex

flex-direction: column

width: 100%

.mainMenuNavigationWrap

display: flex

flex-direction: column

justify-content: space-between

width: 100%

height: 100%

.maxHeight

height: 100%

@media (\$min-tablet)

// Нужно прописать через джсчтобыотступ справа рассчитывался по разнице контейнера и шириныэкрана

.mainMenuNavigationBar

max-width: 235px

padding-left: 47px

padding-top: 39px

@media (\$min-desktop)

.mainMenuNavigationBar

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 74 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```
max-width: 260px
position: fixed
top: 0
left: 0
bottom: calc(100% - 20px)
padding-top: 16px
height: 100vh
padding-left: 58px
.logo
display: block
margin-bottom: 42px
cursor: pointer
svg
width: 59px
height: auto
```

```
.root
display: block
font-size: 14px
font-weight: bold
color: $darkGray
text-transform: uppercase
padding-top: 15px
padding-bottom: 15px
width: 100%
transition: color .3s
cursor: pointer
text-decoration: none
```

```
&:hover,
&:active,
&.active
color: $fiolet
```

```
@media ($min-tablet)
```

```
.root
padding-top: 24px
padding-bottom: 24px
```

```
@media ($min-desktop)
```

```
.root
padding-top: 17px
padding-bottom: 17px
font-size: 18px
```

```
.root
display: flex
flex-direction: column
justify-content: space-between
```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

position: absolute
background: $white
bottom: calc(100% - 37px)
box-sizing: border-box
transition: transform .5s ease
transform: none
z-index: -1
width: 100%
padding-top: 9px
padding-bottom: 23px

&.active
transform: translateY(calc(100%))
transition-delay: .2s

.rootWrap
width: 100%

//h3
// display: none

@media ($min-tablet)
.root
background-color: transparent
padding: 0
padding-top: 14px

.rootWrap
margin-left: auto
background-color: $white
max-width: 450px
box-sizing: border-box
padding: 14px 15px 28px
border-radius: 7px
position: relative

&::after
position: absolute
bottom: 100%
right: 6px
content: "
width: 0
height: 0
border-left: 4px solid transparent
border-right: 4px solid transparent
border-bottom: 4px solid $white
h3
display: block
font-size: 18px
font-weight: bold
color: $red

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

margin: 0
padding: 0
margin-bottom: 8px

.mainMenuNotificationsButton

position: relative
width: 17px
cursor: pointer
svg
width: 100%
height: auto
&.active
&::after
content: "
width: 7px
height: 7px
position: absolute
top: 1px
right: 0
background-color: \$red
border-radius: 50%

.mainMenuNotificationsItem

display: flex
padding-top: 9px
padding-bottom: 17px
font-size: 14px
color: \$darkGray
font-weight: 500
border-bottom: 1px solid \$muteGray
&Body
flex: 1
&DateTime
margin-left: 9px
text-align: right
&Time
color: \$lightGray

@media (\$min-tablet)

.mainMenuNotificationsItem
padding-top: 9px
padding-bottom: 9px
border-bottom: none
&Body
border-radius: 16px
background-color: \$muteGray
padding: 9px 12px
&Date
color: \$red

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 77 |

```

&DateTime
margin-left: 17px
display: flex
flex-direction: column
justify-content: center

.mainMenuNotificationsShowMoreButton
border-radius: 16px
background-color: $muteGray
font-size: 14px
font-weight: bold
color: $fiolet
padding: 8px 0
width: 266px
text-align: center
margin-left: auto
margin-right: auto
margin-top: 23px

@media ($min-tablet)
.mainMenuNotificationsShowMoreButton
margin-top: 20px

.mainMenuOpenButton
width: 17px
height: 12px
display: flex
justify-content: flex-start
align-items: center
cursor: pointer
span
transition: transform .3s
width: 12px
height: 1px
background-color: $darkGray
display: block
position: relative
&::after, &::before
transition: transform .3s
content: "
height: 1px
width: 100%
position: absolute
background-color: $darkGray
left: 0
&::after
bottom: -4px
&::before
top: -4px
&.active
span
transform: rotate(45deg)
    
```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```
&::before
opacity: 0
&::after
transform: translateY(-4px) rotate(90deg)
```

```
.mainMenuVisible
display: flex
box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.05)
background-color: $white
padding: 10px 0
font-weight: bold
font-size: 14px
text-transform: uppercase
color: $darkGray
line-height: normal
&Inner
display: flex
width: 100%
height: 100%
justify-content: space-between
align-items: center
```

```
@media ($min-desktop)
.mainMenuVisible
display: none
```

```
.root
width: 100%
background-color: $white
z-index: 10000
position: fixed
left: 0
top: 0
```

```
.title
font-size: 18px
font-weight: 600
color: $darkGray
margin-bottom: 30px
```

```
.overlay
position: fixed
width: 100%
height: 100%
top: 0
left: 0
background-color: rgba(101, 101, 101, 0.62)
z-index: 0
opacity: 0
transition: visibility 0s .5s, opacity .3s .2s
visibility: hidden
```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

.active
opacity: 1
transition: visibility 0s, opacity .3s
transition-delay: 0s
visibility: visible

.scheduleContainer
width: 100%
&:not(:last-child)
padding-bottom: 30px
&Header
width: 100%
display: flex
justify-content: space-between
font-size: 14px
font-weight: 500
margin-bottom: 8px
&DayOfWeek
padding: 4px 11px
background-color: $fiolet
border-radius: 12px
margin-right: 7px
color: $white
line-height: 1
&Date
color: $fiolet
&LessonsCount
color: $darkGray
>div:not(.scheduleContainerHeader):not(:last-child)
margin-bottom: 9px

.active
.scheduleContainer
&DayOfWeek
background-color: $red
&Date
color: $red

@media ($min-tablet)
.scheduleContainer
&Header
margin-bottom: 30px
&:not(:last-child)
padding-bottom: 40px
>div:not(.scheduleContainerHeader):not(:last-child)
margin-bottom: 15px

@media ($min-desktop)
.scheduleContainer

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 80 |

```

&Header
font-size: 18px

.root
box-sizing: border-box
width: 100%
display: flex

.schedules
width: 100%

.currentLesson
display: none

@media ($min-desktop)
.schedules
flex: 1

.currentLesson
width: 350px
margin-left: 30px
display: block

.scheduleDay
display: flex
padding: 12px
border-radius: 7px
box-shadow: 2px 6px 10px 0 rgba(176, 158, 255, 0.15)
background-color: $white
font-size: 14px
justify-content: space-between
flex-wrap: wrap
&Time
font-weight: bold
color: $darkGray
width: 50%
margin-bottom: 5px
&Info, &SumUp, &InfoGroup, &InfoPersonal
font-weight: 500
color: $lightGray
&Info
display: flex
span
margin-left: 3px
margin-right: 3px
&Group, &Personal
display: none
&SumUp
width: 100%

.active
    
```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

color: \$red

@media (\$min-tablet)

.scheduleDay

&Time, &SumUp

min-width: 60%

flex: 1

&InfoPersonal

order: 3

&Swap

order: 1

&SumUp

order: 2

&InfoGroup

order: 1

.scheduleDayInfo

display: none

&Group, &Personal

display: block

.scheduleLesson

display: flex

padding: 12px

border-radius: 7px

box-shadow: 2px 6px 10px 0 rgba(176, 158, 255, 0.15)

background-color: \$white

font-size: 14px

justify-content: space-between

flex-wrap: wrap

&Time

font-weight: bold

color: \$darkGray

width: 50%

margin-bottom: 5px

&Info

display: flex

width: 90px

justify-content: space-between

&Address, &People

font-weight: 500

color: \$lightGray

&Address

flex: 1

min-width: 75%

&People

width: auto

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

@media ($min-tablet)
  .scheduleLesson
    &Address
    order: 2
    &People
    order: 1
    &Info
    order: 3
    margin-top: 7px
    width: 100%

@media ($min-desktop)
  .scheduleLesson
    font-size: 18px
    &Address
    order: 2
    &People
    order: 1

.root
  position: fixed
  display: flex
  width: 100%
  bottom: 0
  left: 0
  box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.1)
  background-color: $white

.scheduleViewModeWrap
  width: 100%
  display: flex
  justify-content: center
  align-items: center

.scheduleViewModeButtons
  display: flex
  justify-content: center

.scheduleViewModePerson
  display: none

@media ($min-desktop)
  .root
  top: 0
  bottom: auto
  box-shadow: none
  .scheduleViewModeWrap
  justify-content: space-between
  padding-left: 190px
  .scheduleViewModePerson
  display: flex

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```
.root
padding: 11px 25px
position: relative
cursor: pointer
font-size: 14px
font-weight: 500
color: $lightGray
text-decoration: none
```

```
&:not(:last-child)
```

```
&::after
content: "
position: absolute
top: 50%
left: 100%
width: 1px
height: 16px
background-color: $lightGray
transform: translateY(-50%)
```

```
.active
color: $fiolet
```

```
@media ($min-tablet)
```

```
.root
padding: 11px 50px
```

```
@media ($min-desktop)
```

```
.root
padding: 35px 50px
font-size: 18px
```

```
.scheduleViewModePerson
display: none
```

```
@media ($min-desktop)
.scheduleViewModePerson
display: flex
justify-content: center
align-items: center
cursor: pointer
img
width: 47px
height: auto
border-radius: 7px
```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 84 |

```
div
margin-left: 16px
font-size: 18px
font-weight: normal
color: $darkGray
span
font-weight: bold

.root
display: block
padding: 20px
border-radius: 7px
box-shadow: 2px 6px 10px 0 rgba(176, 158, 255, 0.15)
background-color: $white
font-size: 18px
box-sizing: border-box
font-weight: 600
color: $lightGray

// TODO (Inalkman): replacetags on classes
.row
display: flex
justify-content: space-between
span
width: 100%
&:nth-child(2)
color: $darkGray

&:not(:last-child)
margin-bottom: 20px

&:first-child
span
&:nth-child(2)
color: $red

.root
display: flex

.baseAnalytics
display: flex
justify-content: space-between
margin-right: 12px
width: 80px

.tabletOrWider
display: none

@media ($min-tablet)

.baseAnalytics
```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

display: none

    .tabletOrWider
display: inline

import 'lazysizes';

importGlidefrom '@glidejs/glide';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import axios from 'axios';

import Input from '@IndexLanding/components/Input';
import TrialLessonScrollModalWrapper from '@IndexLanding/render_blocks/TrialLessonScrollModalWrapper';

// styles
import '@glidejs/glide/dist/css/glide.core.css';
import './IndexLanding.sass';

const req = require
    .context('@/images/IndexLanding', true, /\.(png|svg|jpg|gif|ico)$/);
req
    .keys()
    .forEach(key => req(key));

ReactDOM.render(
    <TrialLessonScrollModalWrapper />,
    document.querySelector('.trialLessonScrollModal')
);

const questionForm = document.querySelector('#questionForm');
const temp = document.createElement('div');

ReactDOM.render(
    <Input
    placeholder="Ваш номер телефона"
    name="phone_number"
    />,
    temp
);

questionForm.querySelector('input[name="phone_number"]').replaceWith(
    temp.querySelector('input')
);

document.querySelector('#questionForm').addEventListener(
    'submit',
    async event => {
        event.preventDefault();

        let withErrors = false;

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

const formData = new FormData(event.target as HTMLFormElement);

const name = formData.get('name') as string;
const phoneNumber = formData.get('phone_number') as string;
const questionBody = formData.get('question_body') as string;

const nameError = document.querySelector('#nameInputError');
const phoneNumberError = document.querySelector('#phoneNumberInputError');
const questionBodyError = document.querySelector('#questionBodyInputError');

const normalizedPhoneNumber = phoneNumber.replace(/\s/g, '');
if (normalizedPhoneNumber.length) {
    const isNumber = +normalizedPhoneNumber;
    const isValidPhoneNumber = (
        isNumber &&
        (
            normalizedPhoneNumber.startsWith('+380') && normalizedPhoneNumber.length === 13 ||
            normalizedPhoneNumber.startsWith('380') && normalizedPhoneNumber.length === 12 ||
            normalizedPhoneNumber.startsWith('80') && normalizedPhoneNumber.length === 11 ||
            normalizedPhoneNumber.startsWith('0') && normalizedPhoneNumber.length === 10
        )
    );

    if (!isValidPhoneNumber) {
        phoneNumberError.textContent = (
            'Неверный формат номера телефона. ' +
            'Введите номер в формате +380123456789'
        );
        withErrors = true;
    } else {
        phoneNumberError.textContent = 'Это поле должно быть заполнено';
        withErrors = true;
    }

    if (!name.length) {
        nameError.textContent = 'Это поле должно быть заполнено';
        withErrors = true;
    }

    if (!questionBody.length) {
        questionBodyError.textContent = 'Это поле должно быть заполнено';
        withErrors = true;
    }

    if (!withErrors) {
        const response = await axios.post(
            '/send-question',
            formData
        );

        if (response.status === 200) {

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

constsubmitButton = document.querySelector('#submitQuestionButton') asHTMLButtonElement;
submitButton.textContent = 'Готово';
submitButton.disabled = true;
    }
    }
    }
);

// Video
constvideoContainer =
document
    .querySelector('.videoContainer') asHTMLMElement;
constvideoClose =
document
    .querySelector('.videoClose') asHTMLMElement;
constwatchVideoButtons =
document
    .querySelectorAll('.watchVideo');
constvideo =
document
    .querySelector('.videoContainervideo') asHTMLVideoElement;
constvideoControllersStatus =
document
    .querySelector('.videoControllersStatus') asHTMLMElement;
constvideoControllersStatusCurrent =
document
    .querySelector('.videoControllersStatus .videoControllersStatusCurrent') asHTMLMElement;
constvideoControllersStatusCurrentTime =
document
    .querySelector('.videoControllersStatus .videoControllersStatusCurrentTime') asHTMLMElement;
constvideoControllersStatusEndTime =
document
    .querySelector('.videoControllersStatus .videoControllersStatusEndTime') asHTMLMElement;
constvideoControllersPlayPause =
document
    .querySelector('.videoControllersPlayPause') asHTMLMElement;
constvideoControllers =
document
    .querySelector('.videoControllers') asHTMLMElement;
constvideoControllersHideTime = 2000;
letinstrI = null;

videoContainer.addEventListener('mousemove', () => {
if (!sm.matches) {
videoControllers.classList.remove('hide');
clearTimeout(instrI);
instrI = setTimeout(
    () => {
videoControllers.classList.add('hide');
    },
videoControllersHideTime
);

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

    }
  });

  // Function for playorpausevideo
  consttogglePlayPause = () => {
    if (video.paused) {
      videoControllersPlayPause
        .classList
        .add('play');
      video
        .play();
    } else {
      videoControllersPlayPause
        .classList
        .remove('play');
      video
        .pause();
    }
  };

  // Function for parsingtime (minutes:seconds)
  consttimeParse = time => {
    constminutes = Math.floor(time / 60);
    constseconds = Math.floor(time) % 60;
    constshownSeconds = seconds < 10 ? '0' + seconds : seconds;
    return `${minutes}:${shownSeconds}`;
  };

  // Function for detectcurrentvideotime
  conststatusClick = e => {
    if (e.which === 1) {
      conststatus = e.offsetX / videoControllersStatus.offsetWidth;
      videoControllersStatusCurrent
        .style
        .width = `${status * 100}%`;
      videoControllersStatusCurrentTime
        .style
        .left = `${status * 100}%`;
      video.currentTime = status * video.duration;
    }
  };

  // Closevideo
  videoClose.addEventListener('click', () => {
    // fullpage_api.setAllowScrolling(true);
    // fullpage_api.setKeyboardScrolling(true);
    videoControllersPlayPause
      .classList
      .remove('play');
    video.pause();
    videoContainer
      .classList

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

        .remove('active');
    if (!teachersScreen.classList.contains('active')) {
        body.classList.remove('lock-screen');
    }
});

// Pause/Play on videoclick
video.addEventListener('click', togglePlayPause);

constbody = document.querySelector('body');

// Click on watchvideobutton
watchVideoButtons.forEach(link => {
    link.addEventListener('click', e => {
        e.preventDefault();
        // Showvideo'sduration
        videoControllersStatusEndTime
            .textContent = timeParse(video.duration);
        // fullpage_api.setAllowScrolling(false);
        // fullpage_api.setKeyboardScrolling(false);
        videoContainer
            .classList
            .add('active');
        videoControllersPlayPause
            .classList
            .add('play');
        video.play();
        body.classList.add('lock-screen');
        if (!sm.matches) {
            setTimeout(
                () => {
                    videoControllers.classList.add('hide');
                },
                videoControllersHideTime
            );
        }
    });
});

// Pause/Play on buttonplay/pauseclick
videoControllersPlayPause.addEventListener('click', togglePlayPause);

// On videogoeshangecurrenttime
video.addEventListener('timeupdate', () => {
    constcurrentTime = video.currentTime;
    conststatus = currentTime / video.duration;
    videoControllersStatusCurrent
        .style
        .width = `${status * 100}%`;
    videoControllersStatusCurrentTime
        .style
        .left = `${status * 100}%`;
});

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

videoControllersStatusCurrentTime
    .textContent = timeParse(currentTime);
});

// Videokeyboard
letvideoCurrentVolume = 1;
document.addEventListener('keydown', event => {
if (teachersScreen.classList.contains('active')) {
if (event.code === 'Escape') {
if (! videoContainer.classList.contains('active')) {
teachersScreen
    .classList
    .remove('active');
body.classList.remove('lock-screen');
    }
    }
    }
if (newsMoreWrapper.classList.contains('active')) {
if (event.code === 'Escape') {
    // fullpage_api.setAllowScrolling(true);
    // fullpage_api.setKeyboardScrolling(true);
newsDisableActive();
darkEffect
    .classList
    .remove('active');
darkEffectWrapper
    .classList
    .remove('active');
body.classList.remove('lock-screen');
    }
    }
if (priceMoreWrapper.classList.contains('active')) {
if (event.code === 'Escape') {
    // fullpage_api.setAllowScrolling(true);
    // fullpage_api.setKeyboardScrolling(true);
priceDisableActive();
darkEffect
    .classList
    .remove('active');
darkEffectWrapper
    .classList
    .remove('active');
body.classList.remove('lock-screen');
    }
    }
if (videoContainer.classList.contains('active')) {
if (event.code === 'Space') {
if (video.paused) {
videoControllersPlayPause
    .classList
    .add('play');
video.play();

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 91 |

```

if (!sm.matches) {
instrI = setTimeout(
    () => {
videoControllers.classList.add('hide');
    },
videoControllersHideTime
    );
}
} else {
videoControllersPlayPause
    .classList
    .remove('play');
video.pause();
if (! sm.matches) {
clearTimeout(instrI);
videoControllers.classList.remove('hide');
    }
    }
    } elseif (event.code === 'ArrowRight') {
video.currentTime = video.currentTime + 5;
    } elseif (event.code === 'ArrowLeft') {
video.currentTime = video.currentTime - 5;
    } elseif (event.code === 'ArrowUp') {
if (video.volume>= .9) {
video.volume = 1;
    } else {
video.volume += .1;
    }
    } elseif (event.code === 'ArrowDown') {
if (video.volume<= .1) {
video.volume = 0;
    } else {
video.volume -= .1;
    }
    } elseif (event.code === 'KeyM') {
if (video.volume === 0) {
video.volume = videoCurrentVolume;
    } else {
videoCurrentVolume = video.volume;
video.volume = 0;
    }
    } elseif (event.code === 'Escape') {
// fullpage_api.setAllowScrolling(true);
// fullpage_api.setKeyboardScrolling(true);
videoControllersPlayPause
    .classList
    .remove('play');
video.pause();
videoContainer
    .classList
    .remove('active');
if (!teachersScreen.classList.contains('active')) {

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

body.classList.remove('lock-screen');
    }
  }
});

// video.addEventListener('mousemove', () => {
//   videoControllers.classList.remove('hide');
//   waitForMouseStop() => {
//     console.log('stopped');
//   });
//   // setTimeout(() => {
//     //   videoControllers.classList.add('hide');
//     // }, 1500);
// })

// Changevideostatus
videoControllersStatus
  .addEventListener('mousemove', statusClick);
videoControllersStatus
  .addEventListener('click', statusClick);

// Selectortoshowfirstscreenmenu
constmenuHome =
document
  .querySelector('.menuHome') asHTMLElement;
// Selector for mobilemenuopen
constmobileMenuOpen =
document
  .querySelector('.mobileMenuOpen') asHTMLElement;
constmobileMenu =
document
  .querySelector('.mobileMenu') asHTMLElement;
constmobileMenuItem =
document
  .querySelectorAll<HTMLElement>('.mobileMenu .mobileMenuInner .menuInner li a');
// SelectortoshowteachersScreen
constteachersScreenShow =
document
  .querySelector('.aboutScreen> .section-inner> .blockButton') asHTMLElement;
// Selectorstoshowcurrent and allamountofslides on testimonialsScreen
// Also a variabletochangelocationof the rectangle on .buttonLesson
constslidesCount =
document
  .querySelector('.testimonialsSlideCountAll');
constslidesCountActive =
document
  .querySelector('.testimonialsSlideCountActive');
constlessonButtonTestimonials =
document
  .querySelector('.testimonialsScreen .buttonLesson');
// Variablestodefineheight and locationof .blockButton on teachersScreen

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

constteachersScreen =
document
    .querySelector('.teachersScreen') asHTMLElement;
constteachersScreenContainer =
document
    .querySelector('.teachersScreen .container') asHTMLElement;
constteachersScreenBlockButton =
document
    .querySelector('.teachersScreen .blockButton') asHTMLElement;
constteachersClose =
document
    .querySelector('.teachersClose') asHTMLElement;
// Variables for glides
constglideQuestions = '.glideQuestions';
constglideTestimonials = '.glideTestimonials';
constglideTeachers = '.glideTeachers';
constglideNews = '.glideNews';
constglideContacts = '.glideContacts';
// Variablestoadd a carouseltoquestionsScreen on mobiledevices
constquestionsAddGlide =
document
    .querySelector(glideQuestions);
constquestionsAddGlideTrack =
document
    .querySelector('.questionsAddGlideTrack');
constquestionsAddGlideSlides =
document
    .querySelector('.questionsAddGlideSlides');
// Variablestoadd a carouseltonewsScreen on tabletdevices
constnewsAddGlide =
document
    .querySelector(glideNews);
constnewsAddGlideTrack =
document
    .querySelector('.newsAddGlideTrack');
constnewsAddGlideSlides =
document
    .querySelector('.newsAddGlideSlides');
// Variables for responsibility
constlg =
window
    .matchMedia( '(max-width: 1199.98px)' );
const_lg =
window
    .matchMedia( '(min-width: 1199.98px)' );
constsm =
window
    .matchMedia( '(max-width: 767.98px)' );

// NewsMore
constdarkEffect =
document

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

        .querySelector('.darkEffect') asHTMLElement;
constdarkEffectWrapper =
document
    .querySelector('.darkEffectWrapper') asHTMLElement;
letisActiveNewsMore = false;
letisActivePriceMore = false;
constnewsMoreWrapper =
document
    .querySelector('.newsMoreWrapper') asHTMLElement;
constpriceMoreWrapper =
document
    .querySelector('.priceMoreWrapper') asHTMLElement;
constnewsMoreButtonClose =
document
    .querySelector('.newsMoreWrapper .buttonClose') asHTMLElement;
constpriceMoreButtonClose =
document
    .querySelector('.priceMoreWrapper .buttonClose') asHTMLElement;
constnewsItem =
document
    .querySelectorAll<HTMLElement>('.newsItem');
constnewsMoreSelection =
document
    .querySelector('.newsMoreSelection') asHTMLElement;
constnewsMoreSelectionItem =
document
    .querySelectorAll<HTMLElement>('.newsMoreSelectionItem');
constnewsMoreActive =
document
    .querySelector('.newsMoreActive') asHTMLElement;

constnewsDisableActive = () => {
newsMoreSelection
    .querySelector('.newsMoreSelectionItem.hidden')
    .classList
    .remove('hidden');
newsMoreActive
    .querySelector('.newsMoreActiveItem.active')
    .classList
    .remove('active');
newsMoreWrapper
    .classList
    .remove('active');

isActiveNewsMore = false;
};

constpriceDisableActive = () => {
priceMoreWrapper
    .classList
    .remove('active');

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

isActivePriceMore = false;
};

constremoveDarkEffect = () => {
if (isActiveNewsMore) {
newsDisableActive();
}
if (isActivePriceMore) {
priceDisableActive();
}
darkEffect
.classList
.remove('active');
darkEffectWrapper
.classList
.remove('active');
body.classList.remove('lock-screen');
// fullpage_api.setAllowScrolling(true);
// fullpage_api.setKeyboardScrolling(true);
};

darkEffect
.addEventListner('click', removeDarkEffect);
newsMoreButtonClose
.addEventListner('click', removeDarkEffect);

newsItem.forEach(element => {
element.addEventListener('click', () => {
// fullpage_api.setAllowScrolling(false);
// fullpage_api.setKeyboardScrolling(false);
isActiveNewsMore = true;
constdataId =
element
.dataset
.id;
newsMoreSelection
.querySelector(`.newsMoreSelectedItem[data-id='${dataId}']`)
.classList
.add('hidden');
newsMoreActive
.querySelector(`.newsMoreActiveItem[data-id='${dataId}']`)
.classList
.add('active');
darkEffect
.classList
.add('active');
darkEffectWrapper
.classList
.add('active');
body.classList.add('lock-screen');
newsMoreWrapper
.classList

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

        .add('active');
    });
});

const pricesItemOpen =
document
    .querySelectorAll('.pricesItemSelf');

pricesItemOpen.forEach(element => {
    element.addEventListener('click', e => {
        e.preventDefault();
        isActivePriceMore = true;
        darkEffect
            .classList
            .add('active');
        darkEffectWrapper
            .classList
            .add('active');
        body.classList.add('lock-screen');
        priceMoreWrapper
            .classList
            .add('active');

        // fullpage_api.setAllowScrolling(false);
        // fullpage_api.setKeyboardScrolling(false);
    });
});

priceMoreButtonClose
    .addEventListener('click', removeDarkEffect);

newsMoreSelectedItem
    .forEach(element => {
        element
            .addEventListener('click', () => {
                const dataId =
                element
                    .dataset
                    .id;
                newsMoreSelection
                    .querySelector('.newsMoreSelectedItem.hidden')
                    .classList
                    .remove('hidden');
                element
                    .classList
                    .add('hidden');
                newsMoreActive
                    .querySelector('.newsMoreActiveItem.active')
                    .classList
                    .remove('active');
                newsMoreActive
                    .querySelector(`.newsMoreActiveItem[data-id='${dataId}']`)

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

        .classList
        .add('active');
    });
});

/// Fullpageinitializing
/* tslint:disable:no-unused-expression */
/// newfullpage('#fullPage', {
///   menu: '.menuInner',
///   normalScrollElements:
///     '.newsMore, .newsMoreSelection, .newsMoreActiveItemContentBody, .priceMoreWrapper, .scheduleImage, ' +
///     '.questionsItem p',
///   autoScrolling: true,
///   onLeave: (origin, destination) => {
///     removeDarkEffect();
///     if (!lg.matches) {
///       if (origin.anchor === 'mainScreen') {
///         menuHome.style.display = 'none';
///       }
///       if (destination.anchor === 'mainScreen') {
///         menuHome.style.display = 'flex';
///       }
///     }
///     if (origin.anchor === 'aboutScreen') {
///       teachersScreen
///       .classList
///       .remove('active');
///     }
///   }
/// });

constsection = document.querySelectorAll('.section');
constsections = {};

Array.prototype.forEach.call(section, e => {
  sections[e.id] = e.offsetTop;
});

window.addEventListener('scroll', () => {

  constscrollTop = document.documentElement.scrollTop || document.body.scrollTop;

  let i;
  for (i in sections) {
    if (sections[i] <= scrollTop + 75) {
      if (
        typeof(document.querySelector('.menuInnera.active')) !== 'undefined'
        &&document.querySelector('.menuInnera.active') !== null
      ) {
        document.querySelector('.menuInnera.active').classList.remove('active');
      }
      document.querySelector('a[href*=' + i + ']').classList.add('active');
    }
  }
}

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 98 |

```

    }
    }

    if (pageYOffset >= 100) {
        menuHome.style.display = 'none';
    } else {
        menuHome.style.display = 'flex';
    }
});

const scrollIt = (destination, duration = 200, easing = 'linear') => {

    const easings = {
        linear(t: number) {
            return t;
        },
        easeInQuad(t: number) {
            return t * t;
        },
        easeOutQuad(t: number) {
            return t * (2 - t);
        },
        easeInOutQuad(t: number) {
            return t < 0.5 ? 2 * t * t : -1 + (4 - 2 * t) * t;
        },
        easeInCubic(t: number) {
            return t * t * t;
        },
        easeOutCubic(t: number) {
            return (--t) * t * t + 1;
        },
        easeInOutCubic(t: number) {
            return t < 0.5 ? 4 * t * t * t : (t - 1) * (2 * t - 2) * (2 * t - 2) + 1;
        },
        easeInQuart(t: number) {
            return t * t * t * t;
        },
        easeOutQuart(t: number) {
            return 1 - (--t) * t * t * t;
        },
        easeInOutQuart(t: number) {
            return t < 0.5 ? 8 * t * t * t * t : 1 - 8 * (--t) * t * t * t;
        },
        easeInQuint(t: number) {
            return t * t * t * t * t;
        },
        easeOutQuint(t: number) {
            return 1 + (--t) * t * t * t * t;
        },
        easeInOutQuint(t: number) {
            return t < 0.5 ? 16 * t * t * t * t * t : 1 + 16 * (--t) * t * t * t * t;
        }
    }

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

    };

    conststart = window.pageYOffset;
    conststartTime = 'now' in window.performance ? performance.now() : newDate().getTime();

    constdocumentHeight = Math.max(
    document.body.scrollHeight,
    document.body.offsetHeight,
    document.documentElement.clientHeight,
    document.documentElement.scrollHeight,
    document.documentElement.offsetHeight
    );
    constwindowHeight = window.innerHeight
    || document.documentElement.clientHeight
    || document.getElementsByTagName('body')[0].clientHeight;
    constdestinationOffset = typeofdestination === 'number' ? destination : destination.offsetTop;
    constdestinationOffsetToScroll = Math.round(
    documentHeight - destinationOffset < windowHeight
    ? documentHeight - windowHeight - 25
    : destinationOffset - 25);

    if ('requestAnimationFrame' in window === false) {
    window.scroll(0, destinationOffsetToScroll);
    return;
    }

    functionscroll() {
    constnow = 'now' in window.performance ? performance.now() : newDate().getTime();
    consttime = Math.min(1, ((now - startTime) / duration));
    consttimeFunction = easings[easing](time);
    window.scroll(0, Math.ceil((timeFunction * (destinationOffsetToScroll - start)) + start));

    if (window.pageYOffset === destinationOffsetToScroll) {
    return;
    }

    requestAnimationFrame(scroll);
    }

    scroll();
    };

    document.querySelectorAll('.mainMenu .menuInner > li > a').forEach(el => {
    el.addEventListener('click', e => {
    // Allowclick on the link for canaryusers
    if (!window.IS_CANARY) {
    e.preventDefault();
    }
    if (priceMoreWrapper.classList.contains('active')) {
    priceDisableActive();
    darkEffect
    .classList

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 100 |

```

        .remove('active');
    darkEffectWrapper
        .classList
        .remove('active');
    body.classList.remove('lock-screen');
    }
    if (newsMoreWrapper.classList.contains('active')) {
    newsDisableActive();
    darkEffect
        .classList
        .remove('active');
    darkEffectWrapper
        .classList
        .remove('active');
    body.classList.remove('lock-screen');
    }
    if (teachersScreen.classList.contains('active')) {
    teachersScreen
        .classList
        .remove('active');
    body.classList.remove('lock-screen');
    }
    scrollIt(
    document.querySelector(`${el.getAttribute('href').replace('/', '')}`),
        700,
        'easeInOutQuad'
    );
    });
});

// Ifmobiledevice
if (sm.matches) {
    questionsAddGlide
        .classList
        .add('glide');
    questionsAddGlideTrack
        .classList
        .add('glide__track');
    questionsAddGlideSlides
        .classList
        .add('glide__slides');

    newGlide(glideQuestions, {
    startAt: 0,
    perView: 1,
    gap: 5
    })
        .mount();

    lessonButtonTestimonials
        .classList
        .replace('leftCorner', 'rightCorner');

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

mobileMenuItem
    .forEach(el => {
el.addEventListener('click', () => {
mobileMenu.classList.remove('active');
scrollIt(
document.querySelector(`${el.getAttribute('href')}`),
    700,
    'easeInOutQuad'
);
// fullpage_api.setAllowScrolling(true);
// fullpage_api.setKeyboardScrolling(true);
});
});

// Additionalmenus for main and gallerypages
constmobileMenuOpenAdditionalMain =
document
    .querySelector('.mobileMenuInner .showMain .menuInnerAdditionalOpen') asHTMLElement;
constmobileMenuInnerAdditionalMain =
document
    .querySelector('.mobileMenuInner .showMain .menuInnerAdditional') asHTMLElement;

mobileMenuOpenAdditionalMain.addEventListener('click', () => {
mobileMenuOpenAdditionalMain.classList.toggle('open');
mobileMenuInnerAdditionalMain.classList.toggle('is-visible');
});
constmobileMenuOpenAdditionalGallery =
document
    .querySelector('.mobileMenuInner .showGallery .menuInnerAdditionalOpen') asHTMLElement;
constmobileMenuInnerAdditionalGallery =
document
    .querySelector('.mobileMenuInner .showGallery .menuInnerAdditional') asHTMLElement;

if (mobileMenuOpenAdditionalGallery) {
mobileMenuOpenAdditionalGallery.addEventListener('click', () => {
mobileMenuOpenAdditionalGallery.classList.toggle('open');
mobileMenuInnerAdditionalGallery.classList.toggle('is-visible');
});
}
}

// Iftabletdevice
if (lg.matches) {
newsAddGlide
    .classList
    .add('glide');
newsAddGlideTrack
    .classList
    .add('glide__track');
newsAddGlideSlides
    .classList

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

        .add('glide__slides');

newGlide(glideNews, {
    // type: 'carousel',
startAt: 0,
perView: 2,
gap: 30,
peek: {
before: 0,
after: 100
},
breakpoints: {
    767.98: {
perView: 1,
gap: 18,
peek: {
before: 0,
after: 25
}
}
}
})
.mount();
}

// Ifdesktopdevice
if (_lg.matches) {
    // Definitionofheight and locationof .blockButton on teachersScreen
constteachersScreenBlockButtonHeight =
    (teachersScreen.offsetWidth - teachersScreenContainer.offsetWidth) / 2;
teachersScreenBlockButton
.style
.height = `${teachersScreenBlockButtonHeight}px`;
teachersScreenBlockButton
.style
.transform = `translateX(-${teachersScreenBlockButtonHeight}px) rotate(90deg)`;
}

// ShowteachersScreen
teachersScreenShow
.addEventListener('click', e => {
e.preventDefault();
teachersScreen
.classList
.toggle('active');
body.classList.add('lock-screen');
});

teachersClose.addEventListener('click', () => {
    // fullpage_api.setAllowScrolling(true);
    // fullpage_api.setKeyboardScrolling(true);
teachersScreen

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

        .classList
        .remove('active');
body.classList.remove('lock-screen');
});

// Carousel for testimonialsScreen
const glide = new Glide(glideTestimonials, {
  type: 'carousel',
  startAt: 0,
  perView: 2,
  gap: 30,
  breakpoints: {
    767.98: {
      perView: 1
    }
  }
});
glide
  .on('mount.before', () => {
    slidesCount.textContent = String(
      document
        .querySelectorAll('.testimonialsItem')
        .length
    );
  });
glide
  .on('move', () => {
    slidesCountActive
      .textContent = glide.index + 1;
  });
glide
  .mount();

// Carousel for testimonialsScreen
new Glide(glideTeachers, {
  startAt: 0,
  perView: 4,
  gap: 0,
  bound: true,
  peek: {
    before: 0,
    after: 50
  },
  breakpoints: {
    991.98: {
      perView: 2,
      peek: {
        before: 0,
        after: 127
      }
    },
    767.98: {

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

perView: 2,
peek: {
before: 0,
after: 50
}
}
}).mount();

letmap;

// Createmarkers
constallMarks = [];

constdefaultIcon = 'https://maryballet.fra1.digitaloceanspaces.com/static/jsbuild/images/IndexLanding/@2x/marker.png';
constactiveIcon = 'https://maryballet.fra1.digitaloceanspaces.com/static/jsbuild/images/IndexLanding/@2x/marker_active.png';

constinitMap = addresses => {
map = newgoogle.maps.Map(
document.getElementById('contactsContentMap'), {
disableDefaultUI: true,
zoomControl: true,
mapTypeControl: false,
scaleControl: true,
streetViewControl: false,
rotateControl: false,
fullscreenControl: true,
styles: [
{
elementType: 'geometry',
stylers: [
{
color: '#f5f5f5'
}
],
},
{
elementType: 'labels.icon',
stylers: [
{
visibility: 'off'
}
],
},
{
elementType: 'labels.text.fill',
stylers: [
{
color: '#616161'
}
],
},

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 105 |

```

    {
      elementType: 'labels.text.stroke',
      stylers: [
        {
          color: '#f5f5f5'
        }
      ]
    },
    {
      featureType: 'administrative.land_parcel',
      elementType: 'labels.text.fill',
      stylers: [
        {
          color: '#bdbdbd'
        }
      ]
    },
    {
      featureType: 'poi',
      elementType: 'geometry',
      stylers: [
        {
          color: '#eeeeee'
        }
      ]
    },
    {
      featureType: 'poi',
      elementType: 'labels.text.fill',
      stylers: [
        {
          color: '#757575'
        }
      ]
    },
    {
      featureType: 'poi.park',
      elementType: 'geometry',
      stylers: [
        {
          color: '#e5e5e5'
        }
      ]
    },
    {
      featureType: 'poi.park',
      elementType: 'labels.text.fill',
      stylers: [
        {
          color: '#9e9e9e'
        }
      ]
    }
  ]

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

    },
    {
      featureType: 'road',
      elementType: 'geometry',
      stylers: [
        {
          color: '#ffffff'
        }
      ]
    },
    {
      featureType: 'road.arterial',
      elementType: 'labels.text.fill',
      stylers: [
        {
          color: '#757575'
        }
      ]
    },
    {
      featureType: 'road.highway',
      elementType: 'geometry',
      stylers: [
        {
          color: '#dadada'
        }
      ]
    },
    {
      featureType: 'road.highway',
      elementType: 'labels.text.fill',
      stylers: [
        {
          color: '#616161'
        }
      ]
    },
    {
      featureType: 'road.local',
      elementType: 'labels.text.fill',
      stylers: [
        {
          color: '#9e9e9e'
        }
      ]
    },
    {
      featureType: 'transit.line',
      elementType: 'geometry',
      stylers: [
        {
          color: '#e5e5e5'
        }
      ]
    }
  ]
}

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

    }
  ]
},
{
  featureType: 'transit.station',
  elementType: 'geometry',
  stylers: [
    {
      color: '#eeeeee'
    }
  ]
},
{
  featureType: 'transit.station',
  elementType: 'labels',
  stylers: [
    {
      visibility: 'on'
    }
  ]
},
{
  featureType: 'water',
  elementType: 'geometry',
  stylers: [
    {
      color: '#c9c9c9'
    }
  ]
},
{
  featureType: 'water',
  elementType: 'labels.text.fill',
  stylers: [
    {
      color: '#9e9e9e'
    }
  ]
}
];

const bounds = new google.maps.LatLngBounds();

for (const address of addresses) {
  const marker = new google.maps.Marker({
    position: new google.maps.LatLng(address[0], address[1]),
    icon: defaultIcon,
    map
  });
  bounds.extend(marker.position);
}

```

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.001 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 108 |

```

allMarks.push(marker);

google.maps.event.addListener(marker, 'click', function() {
constlatToGo = this.position.lat().toFixed(6);
constlenToGo = this.position.lng().toFixed(6);
constglideSlide = document
    .querySelectorAll<HTML<Element>(
        `<div class="contactsInfoItem">glide__slide:not(.glide__slide--clone)`
    );
letinx;
glideSlide.forEach((sl, index) => {
if (sl.dataset.lat === latToGo&&sl.dataset.len === lenToGo) {
    // glideC.go`={1}`);
    // console.log(index);
inx = index;
    }
});

glideC.go`=${inx}`);
allMarks.forEach(m => {
m.setIcon(defaultIcon);
m.setZIndex(1);
    });
this.setIcon(activeIcon);
this.setZIndex(5);

map
    .setZoom(14);
map
    .panTo(this.getPosition());
});
}
map.fitBounds(bounds);
});

constgoogleMapsUrl =
    'https://maps.googleapis.com/maps/api/js?key=AIzaSyAGvadG6rYjdr86rF_1j7sVEZXXIORMdto&callback=initMap';

constaddMapsScript = addressx => {
if (!document.querySelector(`script[src="${googleMapsUrl}"]`).length) {
document.body.appendChild(Object.assign(
document.createElement('script'), {
type: 'text/javascript',
src: googleMapsUrl,
onload: () =>initMap(addressx)
}));
} else {
// this.initMap(addressx);
}
});
};

```

```

document.getElementById('contactsContentMap').addEventListener('mouseover', () => {
    // fullpage_api.setAllowScrolling(false);
    // fullpage_api.setKeyboardScrolling(false);
});

document.getElementById('contactsContentMap').addEventListener('mouseleave', () => {
    // fullpage_api.setAllowScrolling(true);
    // fullpage_api.setKeyboardScrolling(true);
});

constaddressesAll = [];

constglideC = newGlide(glideContacts, {
    type: 'carousel',
    startAt: 0,
    perView: 1,
    gap: 0,
    dragThreshold: false
});

glideC
    .on('mount.before', () => {
        constaddresses = document.querySelectorAll<HTMLElement>('.contactsInfoItem');
        addresses.forEach(a => {
            addressesAll.push([a.dataset.lat, a.dataset.len]);
        });
    });

glideC
    .on('mount.after', () => {
        addMapsScript(addressesAll);
    });

glideC
    .on('run.after', () => {
        constcurrentLatLen = document
            .querySelector('.contactsInfoItem.glide__slide--active') asHTMLElement;
        // letcurrentMark;
        allMarks.forEach(m => {
            if (
                m.position.lat().toFixed(5) === parseFloat(currentLatLen.dataset.lat).toFixed(5)
                &&
                m.position.lng().toFixed(5) === parseFloat(currentLatLen.dataset.len).toFixed(5)
            ) {
                allMarks.forEach(m => {
                    m.setIcon(defaultIcon);
                    m.setZIndex(1);
                });
                mr.setIcon(activeIcon);
                mr.setZIndex(5);
            }
            map
                .setZoom(14);
        });
    });

```

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

map
  .panTo(mr.getPosition());
}
});
});

glideC.mount();

// addMapsScript();

import * as React from 'react';
import * as ReactDOM from 'react-dom';

import TrialLessonModal from '@IndexLanding/containers/TrialLessonModal';
import AuthModal from '@IndexLanding/containers/AuthModal';

const req = require
  .context('/images/IndexLanding', true, /\.(png|svg|jpg|gif|ico)$/);
req
  .keys()
  .forEach(key => req(key));

const trialLessonClickFlag = 'trialLessonClickFlag';
const TrialLessonModalWrapper: React.FC = () => {
  const [isOpen, setIsOpen] = React.useState(false);

  if (!window[trialLessonClickFlag]) {
    const trialLessonModalOpeners = document
      .querySelectorAll('.registerMainButton, .buttonLesson, .toPersonalPage, .pricesItemGroup, +
        '.mobileMenuInner .blockButton');

    for (const link of Array.from(trialLessonModalOpeners)) {
      link.addEventListener(
        'click',
        e => {
          e.preventDefault();
          setIsOpen(true);
        }
      );
    }
  }

  return (
    <TrialLessonModal
      isOpen={isOpen}
      onRequestClose={() => setIsOpen(false)}
    />
  );
};

ReactDOM.render(
  <TrialLessonModalWrapper />,

```

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | ДП ІС-6226.00.001 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 111 |

```

document.querySelector('.trialLessonModal')
);

if (window.IS_CANARY) {
constAuthModalWrapper: React.FC = () => {
const [isOpen, setIsOpen] = React.useState(false);

React.useEffect(
  () => {
constpersonalOfficeOpeners = document
  .querySelectorAll('.toPersonalPageCanary, .mobileMenuPersonal');

  for (constlinkofArray.from(personalOfficeOpeners)) {
if (link.getAttribute('href') === '#') {
link.addEventListener(
  'click',
  () => {
setIsOpen(true);
  }
);
  }
  },
  []
);

return (
<AuthModal
isOpen={isOpen}
onRequestClose={() =>setIsOpen(false)}
/>
);
};

ReactDOM.render(
<AuthModalWrapper />,
document.querySelector('.authModal')
);
}

// Selector for mobilemenuopen
constmobileMenuOpen =
document
  .querySelector('.mobileMenuOpen') asHTMLElement;
constmobileMenu =
document
  .querySelector('.mobileMenu') asHTMLElement;

// Initializemobilemanu
mobileMenuOpen.addEventListener(
  'click',
  () => {

```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 112 |

```
if (mobileMenu.classList.contains('active')) {  
  mobileMenu.classList.remove('active');  
  } else {  
  mobileMenu.classList.add('active');  
  }  
}  
);
```

| | | | | | | |
|------|------|----------|--------|------|---------------------|------|
| | | | | | ДП ІС-6226.00.001ПЗ | Арк. |
| | | | | | | 113 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |