

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

Факультет інформатики та обчислювальної техніки

(повне найменування інституту, факультету)

Автоматизованих систем обробки інформації і управління

(повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри

О.А.Павлов

(підпис)

(ініціали, прізвище)

“ ”

2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки 6.050103 «Програмна інженерія»

на тему Мобільне застосування для придбання квитків в кінотеатр

Виконав: студент IV курсу, групи ПП-51 Кравченко Олександр
Сергійович (підпис)
(прізвище, ім'я, по батькові)

Керівник доцент, к.т.н., Іванова Л.М. (підпис)
посада, науковий ступінь, вчене звання, прізвище, ініціали

Консультант з графічної документації ст. викл. Головченко М.М. (підпис)
посада, науковий ступінь, вчене звання, прізвище, ініціали

Рецензент: доц., к.т.н., доц. Марковський О.П. (підпис)
посада, науковий ступінь, вчене звання, прізвище, ініціали

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

Студент _____ (підпис)

Київ – 2019 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)

Напрямок підготовки (програма професійного спрямування) – *6.050103
«Програмна інженерія» (Програмне забезпечення систем)*

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

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

“ ” _____ 2019 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Кравченку Олександрю Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема проекту *«Мобільне застосування для онлайн-придбання квитків в кінотеатр»*

керівник проекту Іванова Любов Миколаївна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “23” квітня 2019 р. №1181-с

2. Термін подання студентом проекту «03» червня 2019 року

3. Вихідні дані до проекту

Технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: основні визначення та терміни, опис предметного середовища, огляд існуючих технічних рішень та відомих програмних продуктів, розробка функціональних та нефункціональних вимог

2) Моделювання та конструювання програмного забезпечення: моделювання та аналіз програмного забезпечення, засоби розробки, технічні рішення, архітектура програмного забезпечення

3) Аналіз якості та тестування програмного забезпечення

4) Впровадження та супровід програмного забезпечення

5. Перелік графічного матеріалу

1) Схема бази даних

2) Схема структурна класів програмного забезпечення

3) Креслення вигляду екранних форм

6. Консультанти розділів проекту

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

7. Дата видачі завдання «12» березня 2018 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення рекомендованої літератури	20.03.2019	
2.	Аналіз існуючих методів розв'язання задачі	23.03.2019	
3.	Постановка та формалізація задачі	25.03.2019	
4.	Аналіз вимог до програмного забезпечення	26.03.2019	
5.	Алгоритмізація задачі	28.03.2019	
6.	Моделювання програмного забезпечення	29.03.2019	
7.	Обґрунтування використовуваних технічних засобів	01.04.2019	
8.	Розробка архітектури програмного забезпечення	10.04.2019	
9.	Розробка програмного забезпечення	10.05.2019	
10.	Налагодження програми	15.05.2019	
11.	Виконання графічних документів	20.05.2019	
12.	Оформлення пояснювальної записки	27.05.2019	
13.	Подання ДП на попередній захист	28.05.2019	
14.	Подання ДП рецензенту	04.06.2019	
15.	Подання ДП на основний захист	08.06.2019	

Студент _____ Кравченко О.С.
(підпис)

Керівник проекту _____ Іванова Л.М.
(підпис)

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 61 сторінку, 11 рисунків, 36 таблиць, 7 джерел.

Дипломний проект присвячений розробці інформаційної системи онлайн придбання квитків в кінотеатр з метою підвищення ефективності процесу вибору та купівлі квитків за допомогою використання розробленої системи, яка містить серверне застосування та мобільний додаток.

В дипломному проекті були розглянуті: загальні положення автоматизації бізнес-процесів, існуючі технічні рішення, необхідний функціонал програмного забезпечення, сценарії його використання та тестування.

У розділі з інформаційного забезпечення були визначені вхідні та вихідні дані до комплексу задач, була розроблена структура бази даних для збереження даних, яка відповідає поставленим цілям проекту.

У розділі з програмного забезпечення описані основні засоби розробки комплексу задач, висунуті вимоги до технічного забезпечення, обрано та обґрунтовано архітектуру програмного забезпечення.

У технологічному розділі описана інструкція користувача та проведене тестування комплексу задач.

					КПІ.ІП-5109.045490.01.81	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

Structure and the scope of work. The explanatory note of the diploma project consists of five sections, containing 61 pages, 11 figures, 36 tables, 7 sources.

The diploma project is devoted to the development of an information system for online purchasing tickets to the cinema with the aim of increasing the efficiency of the process of selection and purchase of tickets through the use of a developed system that contains server application and mobile application.

In the thesis project were considered: general provisions of automation of business processes, existing technical solutions, necessary software functionalities, scenarios for its use and testing.

In the section on information provision input and output data were defined to a set of tasks, a database structure for data storage was developed which corresponds to the objectives of the project.

The software section describes the main tools for developing a set of tasks, the requirements for technical support, and chooses and justifies the software architecture.

The technology section describes the user's manual and tests a set of tasks.

					КПІ.ІП-5109.045490.01.81	Арк.
						6
ЗМН.	Арк.	№ докум.	Підпис	Дата		

**Пояснювальна записка
до дипломного проекту**

на тему: Мобільне застосування для онлайн-придбання квитків в кінотеатр

Київ – 2019 року

					КПІ.ІП-5109.045490.01.81	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП.....	11
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	13
1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	13
1.2 ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	14
1.2.1 <i>Принципи автоматизації процесів.....</i>	<i>14</i>
1.2.2 <i>Типи автоматизації систем.....</i>	<i>15</i>
1.2.3 <i>Стратегія автоматизації систем.....</i>	<i>15</i>
1.2.4 <i>Переваги та недоліки автоматизування систем.....</i>	<i>16</i>
1.3 АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ.....	17
1.3.1 <i>Аналіз відомих програмних продуктів.....</i>	<i>18</i>
1.4 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	21
1.4.1 <i>Розроблення функціональних вимог.....</i>	<i>22</i>
1.4.2 <i>Розроблення нефункціональних вимог.....</i>	<i>37</i>
1.4.3 <i>Постановка комплексу завдань модулю.....</i>	<i>38</i>
1.5 Висновки по розділу	38
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	40
2.1 МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	40
2.1.1 <i>Клієнтська частина.....</i>	<i>40</i>
2.1.2 <i>Серверна частина.....</i>	<i>41</i>
2.2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	47
2.3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	47
2.4 АНАЛІЗ БЕЗПЕКИ ДАНИХ.....	55
2.5 Висновки по розділу	56
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	57
3.1 АНАЛІЗ ЯКОСТІ ПЗ.....	57
3.2 ОПИС ПРОЦЕСІВ ТЕСТУВАННЯ.....	57
3.2.1 <i>Компоненти, що тестуються.....</i>	<i>57</i>
3.2.2 <i>Компоненти, що не тестуються.....</i>	<i>58</i>

3.2.3	Підхід до тестування	58
3.2.4	Критерій проходження тесту	59
3.3	ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ	59
3.4	Висновки до розділу	60
4	ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	61
4.1	РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	61
4.2	РОБОТА З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....	62
4.3	Висновки до розділу	62
	ВИСНОВКИ	63
	ПЕРЕЛІК ПОСИЛАНЬ.....	64

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

Swift - компільована мова програмування загального призначення.

Sequence Diagram – вид діаграм для опису бізнес-процесів програмного забезпечення

Heroku - хмарна PaaS-платформа, що підтримує ряд мов програмування.

					КПІ.ІП-5109.045490.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

ВСТУП

Автоматизація процесів на сьогоднішній день являє собою одну з концепцій управління ними, відмінна риса якої - використання інформаційних технологій. Вона передбачає широке застосування ЕОМ і програмного забезпечення і забезпечує управління інформацією, ресурсами і діями з мінімальною участю людини в даних процедурах або без такого в принципі.

Головне завдання, яке повинно реалізувати проектування автоматизації процесів - це вивід якісних показників процесів на принципово вищий рівень. Досягається вона головним чином завдяки тому, що основною перевагою автоматизованого режиму над ручним є його велика надійність, що, в свою чергу, сприяє зростанню продуктивності, прискоренню, здешевленню, збільшенню точності і стабільності.

На сьогоднішній день автоматизація процесів в світі використовується повсюдно - від координування найскладніших виробництв до здійснення придбань в супермаркетах. Спрямованість компанії, так само як і її масштаби, в даному випадку не принципові: автоматика буквально пронизує будь-яку з них. Одним із основних напрямків автоматизації є сфера послуг, а саме бронювання готелів, оплата таксі, продаж квитків на транспорт, та розважальні події, такі як кіно.

Організація бронювання і продажу квитків, моніторинг сеансів в реальному часі, активне використання сучасних технологій для обслуговування клієнтів дають можливість домогтися максимально продуктивної роботи на всіх процесах функціонування кінозалу. Реалізована автоматизація кінотеатру за допомогою сучасного програмного забезпечення підвищує зручність обслуговування клієнтів і спрощує роботу самого персоналу.

Мета створення даної роботи – підвищення ефективності процесу вибору та купівлі квитків в кінотеатр за допомогою використання інтегрованої інформаційної системи, яка містить серверне застосування та мобільний додаток.

					КП.ІП-5109.045490.01.81	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Завданням даної роботи є автоматизація бізнес-процесів кінотеатру, а саме розробка системи для онлайн купівлі квитків в кінотеатр, яку буде можливо використовувати для різних кінотеатрів. Компонентами розроблюваної системи є мобільний застосунок для платформи iOS, та серверне застосування. Результатом роботи є інтегрована система, яку можна використовувати у будь-якому кінотеатрі, де необхідно провести автоматизацію процесу купівлі квитків.

Змн.	Арк.	№ докум.	Підпис	Дата	КПІ.ІП-5109.045490.01.81	Арк.
						12

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Автоматизація бізнес-процесів - це впровадження програмної системи, яка виконує типові процедури по сучасним алгоритмам, таким як складання і виписка документів, контроль виконання облікових, складських операцій. Завдяки подібним нововведенням рівень роботи підприємства якісно підвищується.

Розбираючись в тому, що таке автоматизація бізнесу і що вона дає, слід зазначити наступні позитивні аспекти її введення:

- виняток рутинного і одноманітного ручної праці;
- інформація обробляється і передається набагато швидше;
- в єдиному інформаційному просторі обслуговуються окремі підрозділи або все підприємство;
- запрограмовані звіти і документи складаються автоматично;
- зростає зручність оперування базами даних, а робота стає більш точною, повною і несуперечливою;
- підрозділи або вся організація зводяться до єдиного комплексу, який об'єднують регулярний стандартизований документообіг та інформаційно-нормативна база.

Практика показує, що автоматизація процесів в переважній більшості випадків сприяє істотному підвищенню якості продукції та послуг. З огляду на предметну область, для забезпечення згаданого зростання якості послуг, необхідно використання інформаційної системи, яка включає аутентифікацію користувача, вибір та купівля квитків, та збереження куплених квитків для подальшого перегляду та пред'явлення у кінотеатрі.

1.2 Змістовний опис і аналіз предметної області

1.2.1 Принципи автоматизації процесів

Хоча автоматизація процесів і має можливість реалізовуватися на різних рівнях, її принципи для кожного з них, так само як і для будь-якого з процесів, є єдиними. Дана обставина ставить умови максимально ефективної реалізації процесів, що здійснюються в автоматичному режимі і визначає принципи управління ними з боку автоматики.

До таких принципів відносять:

1. Узгодженість. В процесі, який контролюється автоматикою, всі операції повинні бути узгоджені як взаємно, так і з його входами і виходами. Невиконання цієї умови нерідко призводить до збоїв у виконанні поставленого завдання.

2. Інтеграція. Що знаходиться під управлінням автоматики процес повинен мати можливість вбудовуватися в загальну систему організації. Відповідно до рівня автоматизації подібна інтеграція може бути реалізована по-різному. Що, втім, не змінює суті самого принципу, яка залишається без змін. Мета автоматизації будь-якого процесу - це забезпечення його взаємодії з середовищем, яка є по відношенню до нього зовнішньої.

3. Незалежність виконання. Будь-який процес, який реалізується за допомогою автоматики, повинен реалізовуватися без людської участі або ж зі зведеним до мінімуму рівнем контролю, здійснюваного оператором. При виконанні умови, що процес реалізується коректно, людське втручання в нього не повинно мати місця.

Варто зазначити, що перераховані вище принципи можуть бути істотно деталізовані, що залежить від типу процесів і рівня автоматизації.

1.2.2 Типи автоматизації систем

Кожну з систем автоматизації можна віднести до одного з трьох основних їх типів:

1. Незмінний. В ході процесу задана послідовність дій не підлягає ніяким коректувань.

2. Програмований. Залежно від використовуваної програми і особливостей процесу, послідовність дій може змінюватися. Вибір потрібної послідовності залежить від набору інструкцій, на який і орієнтується система.

3. Самоналагоджувальна. Потрібні варіанти вирішення завдання можуть вибиратися на розсуд системи під час роботи. Зміна їх послідовності та умов виконання проводиться за допомогою моніторингу ходу процесу.

Дані види автоматизації процесів можуть використовуватися відокремлено або як частина комбінованої системи: тут все залежить від ситуації.

1.2.3 Стратегія автоматизації систем

Будучи непростий і досить трудомістким завданням, автоматизація процесів вимагає чіткого дотримання певної стратегії, що сприяє зростанню ефективності діяльності і отримання чималої кількості очевидних переваг.

Щоб сформуванати стратегію, процес необхідно:

1. Зрозуміти. На підставі ретельно проведеного аналізу можна отримати максимально повне уявлення про процес, аж до найдрібніших подробиць. Так, слід отримати дані про його входах і виходах, послідовності операцій, наявності зв'язку з іншими процесами, склад його ресурсів і т.д.

2. Спростити. Зробити це можна шляхом видалення операцій, які не становлять цінності. Деякі операції можна об'єднувати, здійснювати паралельно і т.п. Тут вітається все, що є найбільш раціональним - наприклад, удосконалення технології реалізації процесу або її повна заміна.

					КПІ.ІП-5109.045490.01.81	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Автоматизувати. До даного пункту слід підходити лише після виконання попереднього. Причина цього проста: чим простішим є процес, тим легше він підлягає автоматизації і тим стабільніше і ефективніше він буде виконуватися.

1.2.4 Переваги та недоліки автоматизування систем

Практика показує, що автоматизація процесів в переважній більшості випадків сприяє істотному підвищенню якості - як продукції, так і управління її виробництвом. Проте, перед прийняттям остаточного рішення, що стосується впровадження автоматизації, рекомендується ознайомитися з основними її перевагами.

Отже, головні переваги автоматизації процесів - це:

1. Прискорення виконання операцій, які повторюються. Автоматика в процесі своєї роботи не схильна до втоми, що забезпечує її безпомилковість, не залежну від тривалості діяльності. Крім цього, однотипні завдання вона реалізує набагато швидше людини.

2. Поліпшення якісних показників діяльності. Головна причина цього - відсутність людського фактора, що сприяє зниженню в рази кількості різного роду помилок.

3. Підвищення точності управління. Завдяки застосовуваним інформаційним технологіям, обсяг даних про процес може бути дуже великим, що сприяє більш точному його виконання.

4. Збільшення швидкості прийняття рішення в ситуаціях, які є типовими. Дана перевага гарантує відсутність невідповідностей на наступних стадіях процесу.

Однак варто брати до уваги, що в деяких ситуаціях використання автоматизації процесів не є виправданим. Таке можливо, коли:

1. Дії надто складні технологічно або економічно.

2. Продукт має короткий життєвий цикл. У разі, якщо продукція створюється і виводиться на ринок максимально швидко, або попит на неї буде

					КПІ.ІП-5109.045490.01.81	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

нетривалим - автоматизація може виявитися менш раціональною, ніж ручне виконання процесів.

3. Продукція є одиничною, виконаною майстром вручну.

4. Попит на продукт піддається надмірним коливанням. Дана обставина призводить до змін обсягів випуску, що робить використання автоматизації невиправданим.

У будь-якому випадку, перед прийняттям остаточного рішення щодо доцільності використання автоматизації слід провести якомога більш ретельний аналіз, щоб виключити ймовірність некоректно зробленого висновку.

1.3 Аналіз успішних ІТ-проектів

На сьогоднішній день за допомогою сучасних технологій і просунутого програмного забезпечення можна здійснити ефективну організацію практично будь-якого бізнесу і великої компанії. Підприємства в сфері мультимедійних розваг не є винятком, завдяки чому автоматизація кінотеатру для багатьох бізнесменів, що займаються прокатом кіно, є пріоритетною. Зараз існує велика кількість сервісів, що пропонують автоматизувати процес вибору та купівлі квитків на кіносеанс. Більшість сервісів пропонують ознайомчий період і, у подальшому, платний план з комісією, або ж створені для окремої мережі кінотеатрів і використовуються виключно нею. Однак обидві категорії сервісів пропонують ідентичні послуги, а саме:

- Перегляд актуальних фільмів, їх описання та сеанси
- Можливість вибору місця (або місць)
- Оплату квитків через Інтернет
- Зберігання куплених квитків у базі даних
- Повернення квитків
- Генерацію штрих-кодів або QR кодів для пропуску на сеанс

1.3.1 Аналіз відомих програмних продуктів

Найбільш популярні платформи для автоматизації бізнесу кінотеатрів:

Online Tickets System

Сервіс пропонує оренду програмного забезпечення і працює через SaaS [], а також повний викуп програмного забезпечення із можливістю встановлення на власний сервер.

Сервіс пропонує такі можливості у базовому комплекті:

Модуль «Касир»

- Робота з розкладом сеансів
- Фіксація проданих місць
- Продаж, бронювання, повернення грошей
- Монітор покупця
- Система фіксованих знижок

Модуль «Адміністратор»

- Служба захисту касирів
- Управління ціновими схемами
- Управління розкладом
- Управління фільмами
- Статистика продажів

Недоліком системи є відсутність графічного інтерфейсу (треба платити окремо), відсутність можливості у клієнтів купувати квитки онлайн, та відсутність дисконтної програми у базовому пакеті, відсутність мобільного додатку.

SaleTicket

Програма автоматизує продаж квитків для кінотеатрів, театрів, стадіонів, цирків та інших підприємств, відвідування яких здійснюється за вхідними квитками.

Основні можливості сервісу це:

- Продаж квитків, обмін і повернення куплених квитків.
- Онлайн продаж і бронювання квитків.
- Депозити, абонементи, подарункові сертифікати.
- Бронювання квитків і продаж їх з функцією пошуку заброньованих місць за номером броні або прізвища
- Реалізація квитків з різними знижками
- Нарахування бонусів за покупки і можливість придбання квитків за накопичені бонуси.

Система являє собою програмне забезпечення для операційної системи Windows, використовує застаріле обладнання (Windows XP, MS SQL Server 2005), та не має кабінету користувача для перегляду куплених квитків. Також цей сервіс, як і попередній, був розроблений у першу чергу для продажу кінотеатрам, тоді як для користувачів вони можуть не принести ніякої користі.

Vkino

Vkino – найбільш популярний сервіс продажу квитків в кінотеатр. Він найбільш сучасний, підтримує усі популярні платформи, та безліч кінотеатрів. Його основні переваги це:

- Наявність WEB інтерфейсу
- iOS та Android додатки
- Підтримка безлічі кінотеатрів по всій країні
- Кабінет користувача
- Фільтрація кінотеатрів за містом
- Пошук фільмів

					КПІ.ІП-5109.045490.01.81	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

- Перегляд трейлерів
- Онлайн оплата
- Повернення квитків

Цей сервіс міг би стати найкращім, якби не недоліки у його мобільних додатках. Мобільні телефони зараз є дуже популярними і в основному користувачі віддають перевагу мобільним додаткам, а не веб сайтам, адже телефон завжди з собою. Нажаль, мобільні додатки Vkinо – дуже застарілі, мають певну кількість багів, не адаптовані під останні моделі смартфонів, та періодично аварійно закінчують роботу. Їхній рейтинг 2 із 5 та багато негативних відгуків лише підтверджують висновки.

Kinoland

Сервіс складається з мобільних додатків для iOS та Android та використовуються локальною мережею кінотеатрів.

Його основні переваги:

- Сучасні додатки до найбільш популярних мобільних операційних систем iOS та Android
- Покупка квитків прямо із додатку
- Зручний інтерфейс
- Кабінет користувача з можливістю генерації штрих кодів, що дозволяє повністю самостійно проходити до кінозалу
- Система бонусів

До недоліків цієї системи можна віднести не адаптований інтерфейс до екранів нових смартфонів, та невелику кількість кінотеатрів (5 штук).

Переглянувши інформацію про найбільш популярні платформи для автоматичної покупки квитків в кінотеатр, можна виявити ряд переваг та недоліків. Частина платформ мають лише WEB інтерфейс, що очевидно недостатньо у сучасних умовах. Також, не усі мають кабінет користувача, який би дозволив клієнтам самостійно проходити на сеанс, без необхідності

					КПІ.ІП-5109.045490.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

звертатися до касира. Ті ж платформи, що мають і мобільні додатки, і кабінет користувача, морально та технічно застаріли, або були розроблені та працюють виключно локальною мережею кінотеатрів.

1.4 Аналіз вимог до програмного забезпечення

Для визначення вимог до програмного забезпечення необхідно визначити ролі користувачів та їх можливості в системі. Для досягнення мети розробки система повинна містити наступні типи користувачів:

- неавтентифікований користувач;
- автентифікований користувач;
- авторизований користувач.

Неавтентифікований користувач має можливість автентифікації за допомогою серверного програмного забезпечення, перегляду фільмів, сеансів, схем кінозалу та вільних місць.

Автентифікований користувач має доступ до збереження особистих даних для авторизації.

Авторизований користувач має можливість покупки квитків, перегляду куплених квитків, та перегляду власного профіля.

Загалом система повинна мати такий функціонал:

- автентифікація користувача (клієнтський застосунок та серверне застосування);
- реєстрація користувача (клієнтський застосунок та серверне застосування);
- перегляд списку доступних фільмів за обраною датою (клієнтський застосунок та серверне застосування);
- перегляд списку сеансів за обраним фільмом (клієнтський застосунок та серверне застосування);
- перегляд схеми кінозалу та вільних місць (клієнтський застосунок та серверне застосування);

– покупка квитків та перегляд куплених квитків користувачем (клієнтський застосунок та серверне застосування).

1.4.1 Розроблення функціональних вимог

В системі передбачено наступні варіанти використання:

Таблиця 1.1 – Варіант використання UC001

Назва	Вхід користувача в систему.
Опис	Користувач має можливість провести всі дії, необхідні для входу в систему: автентифікацію, реєстрацію, авторизацію.
Учасники	Користувач.
Передумови	Користувач не авторизований

Таблиця 1.2 – Варіант використання UC002

Назва	Автентифікація користувача
Опис	Користувач має можливість пройти автентифікацію за допомогою мобільного додатку.
Учасники	Неавтентифікований користувач.
Передумови	Користувач не автентифікований
Постумови	Користувач проходить автентифікацію в системі.

Продовження таблиці 1.2

Основний сценарій	<p>Користувач натискає кнопку позначену як “Профіль”.</p> <p>Система демонструє екран автентифікації.</p> <p>Користувач заповнює власну інформацію для автентифікації.</p> <p>Система демонструє кнопку “Авторизація”.</p> <p>Користувач натискає на кнопку “ Авторизація ”.</p> <p>Система надає користувачеві можливість зберегти особисті дані для подальшої авторизації у клієнтському застосуванні.</p>
Розширення сценаріїв	<p>1.1. Користувач не вводить необхідні дані для проходження коректної автентифікації.</p> <p>1.1.a. Система демонструє користувачеві неможливість автентифікації.</p> <p>2.1. Система виявляє, що дані, внесені користувачем, не є валідними даними для формату даних, необхідних для автентифікації, або існує користувач, зареєстрований в системі за такими даними для автентифікації.</p> <p>2.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.</p>

Таблиця 1.3 – Варіант використання UC003

Назва	Реєстрація користувача в системі.
-------	-----------------------------------

Продовження таблиці 1.3

Опис	Користувач має можливість для збереження особистих даних, а саме email та паролю, для подальшої авторизації в мобільному додатку та його подальшому використанні.
Учасники	Автентифікований користувач.
Передумови	Користувач мав пройти автентифікацію.
Постумови	Користувач реєструє особисті дані для авторизації в системі.
Основний сценарій	Система демонструє екран для введення особистих даних користувача для подальшої авторизації. Користувач вводить дані для подальшої авторизації. Користувач натискає кнопку "Реєстрація". Система надає користувачеві можливість підтвердити особисті дані.
Розширення сценаріїв	1.1. Система виявляє наявність зареєстрованого акаунту за вказаними особистими даними, або виявляє помилку автентифікації. 1.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.

Таблиця 1.4 – Варіант використання UC004

Назва	Авторизація користувача в мобільному застосунку.
Опис	Користувач, що пройшов авторизацію, має можливість авторизації в системі.

Продовження таблиці 1.4

Учасники	Автентифікований користувач
Передумови	
Постумови	Користувач проходить авторизацію у застосунку.
Основний сценарій	Система демонструє екран авторизації, яке містить поля для вводу його email та паролю. Користувач вводить відповідні дані. Система демонструє кнопку “Увійти”. Користувач натискає кнопку “Увійти”. Система авторизує користувача.
Розширення сценаріїв	1.1. Користувач не вводить необхідні дані для проходження коректної авторизації. 1.1.a. Система демонструє користувачеві неможливість авторизації. 2.1. Система виявляє, що дані, внесені користувачем, не є валідними даними для завершення автентифікації. 2.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.

Таблиця 1.5 – Варіант використання UC005

Назва	Відновлення пароля користувача.
Опис	Користувач має можливість відновити свій пароль.
Учасники	Автентифікований користувач

Продовження таблиці 1.5

Передумови	Автентифікований користувач підтвердив та зареєстрував особисті дані для входу в систему
Постумови	Користувач має можливість змінити пароль.
Основний сценарій	Користувач переходить на екран «Профіль» Користувач натискає кнопку «Загубив пароль». Система демонструє користувачеві вікно для введення email. Користувач вводить email. Користувач натискає на кнопку «Відновити пароль». Система надає користувачеві можливість змінити пароль, відправивши інструкцію на вказаний email.
Розширення сценаріїв	1.1. Користувач не вводить необхідні дані для проходження коректної процедури відновлення пароля. 1.1.a. Система демонструє користувачеві неможливість відновлення пароля. 2.1. Система виявляє, що дані, внесені користувачем, не є валідними даними для відновлення пароля, наприклад, введений email не зареєстрований в системі. 2.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.

Таблиця 1.6 – Варіант використання UC006

Назва	Отримання доступних фільмів.
Опис	Користувач має можливість переглядати список доступних фільмів.

Продовження таблиці 1.6

Учасники	Користувач.
Передумови	
Постумови	Користувач отримав список доступних фільмів.
Основний сценарій	Система демонструє користувачеві список актуальних фільмів. Користувач натискає на окремий фільм. Система демонструє користувачеві екран з деталями фільму.
Розширення сценаріїв	1.1. Фільм було видалено у проміжку часу між успішною авторизацією користувача та натисканням на фільм у застосунку. 1.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.

Таблиця 1.7 – Варіант використання UC007

Назва	Вибір дати.
Опис	Користувач має можливість вибрати бажану дату і переглянути доступні фільми.
Учасники	Користувач.
Передумови	
Постумови	Користувач отримав список доступних фільмів за обраною датою.

Продовження таблиці 1.7

Основний сценарій	Система демонструє користувачеві список актуальних фільмів. Користувач натискає на окремий фільм. Система демонструє користувачеві екран з деталями фільму.
Розширення сценаріїв	1.1. Фільм було видалено у проміжку часу між успішною авторизацією користувача та натисканням на фільм у застосунку. 1.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.

Таблиця 1.8 – Варіант використання UC008

Назва	Отримання інформації про фільм.
Опис	Користувач має можливість переглядати деталі обраного фільму.
Учасники	Користувач.
Передумови	Авторизований користувач пройшов авторизацію в застосунку та обрав фільм зі списку фільмів.
Постумови	Користувачеві надано детальну інформацію про обраний фільм.
Основний сценарій	Система демонструє користувачеві інформацію про обраний фільм.

Продовження таблиці 1.8

Розширення сценаріїв	1.1. Фільм було видалено під час отримання інформації про нього. 1.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.
----------------------	--

Таблиця 1.9 – Варіант використання UC009

Назва	Отримання сеансів на обраний фільм.
Опис	Користувач має можливість переглядати список доступних сеансів на обраний фільм.
Учасники	Користувач.
Передумови	Авторизований користувач пройшов авторизацію в застосунку, обрав фільм та запросив розклад сеансів.
Постумови	Користувачеві наданий розклад сеансів на обраний фільм.
Основний сценарій	Система демонструє користувачеві розклад сеансів на обраний фільм.
Розширення сценаріїв	1.1. Фільм було видалено під час отримання інформації про нього. 1.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.

Таблиця 1.10 – Варіант використання UC010

Назва	Отримання схеми залу та вільних місць на обраний сеанс.
Опис	Користувач має можливість переглядати список вільних місць та загальну схему залу на обраний сеанс.
Учасники	Користувач.
Передумови	Авторизований користувач пройшов авторизацію в застосунку, обрав фільм та сеанс.
Постумови	Користувачеві надана схема зали з позначеними вільними місцями.
Основний сценарій	Система демонструє користувачеві схему зали з позначеними вільними місцями.
Розширення сценаріїв	1.1. Фільм було видалено під час отримання інформації про нього. 1.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.

Таблиця 1.11 – Варіант використання UC011

Назва	Вибір місць для покупки.
Опис	Користувач має можливість вибрати бажані місця на схемі залу.
Учасники	Користувач.

Продовження таблиці 1.11

Передумови	Авторизований користувач пройшов авторизацію в застосунку, обрав фільм, сеанс, та бажані місця.
Постумови	Користувачеві наданий список обраних місць.
Основний сценарій	Система демонструє користувачеві список обраних ним місць, та позначає їх на схемі залу.
Розширення сценаріїв	1.1. Користувач обрав зайняте місце. 1.1.а. Система не дозволяє обрати зайняте місце.

Таблиця 1.12 – Варіант використання UC012

Назва	Покупка квитків.
Опис	Користувач має можливість купити квиткі на обраний сеанс.
Учасники	Користувач.
Передумови	Авторизований користувач пройшов авторизацію в застосунку, обрав фільм, сеанс, бажані місця, та нажав на кнопку «Купити Квитки».
Постумови	У особистий кабінет користувача збережено список куплених квитків.
Основний сценарій	Система пропонує користувачеві ввести номер картки і переадресує його на сайт платіжної системи. Після успішної оплати квитки зберігаються у особистому кабінеті, де можуть бути переглянути у будь-який час.

Продовження таблиці 1.12

Розширення сценаріїв	1.1. Під час оплати виникла помилка. 1.1.а. Система демонструє користувачеві повідомлення про відповідну помилку.
----------------------	--

Таблиця 1.13 – Варіант використання UC013

Назва	Отримання списку куплених квитків.
Опис	Користувач має можливість переглянути список куплених квитків.
Учасники	Користувач.
Передумови	Авторизований користувач пройшов авторизацію в застосунку.
Постумови	Користувачеві наданий список куплених квитків.
Основний сценарій	Система демонструє користувачеві список куплених ним квитків.

Функціональні вимоги до клієнтського застосування

Вимоги до клієнтського застосування описано наступними таблицями.

Таблиця 1.14 – Опис функціональної вимоги REQ001

Номер	REQ001
Назва	Вхід у систему
Опис	Користувач має здійснити вхід в систему для можливості покупати квитки та переглядати список вже куплених.

Таблиця 1.15 – Опис функціональної вимоги REQ002

Номер	REQ002
Назва	Автентифікація користувача
Опис	Автентифікація користувача за допомогою введення інформації, яка може ідентифікувати користувача. Автентифікацію може пройти будь-який користувач мобільного застосунку. При успішній автентифікації користувач може зареєструвати особисті дані в системі для подальшої авторизації.

Таблиця 1.16 – Опис функціональної вимоги REQ003

Номер	REQ003
Назва	Реєстрація даних користувача в системі
Опис	Для реєстрації користувача в системі має бути використано його (email) та пароль.

Таблиця 1.17 – Опис функціональної вимоги REQ004

Номер	REQ004
Назва	Авторизація користувача
Опис	Користувач може авторизуватися в мобільному застосунку після того, як пройде етап автентифікації та реєстрації. Щоб авторизуватися, користувач вводить email та пароль. У разі успішної авторизації користувачеві надається доступ до покупки квитків та особистого кабінету.

Таблиця 1.18 – Опис функціональної вимоги REQ005

Номер	REQ005
Назва	Відновлення пароля користувача
Опис	Користувач, який пройшов автентифікацію та зареєстрував особисті дані в системі, може відновити пароль за допомогою email. Система має отримати email, для якого буде відновлено пароль, відправити користувачеві токен для підтвердження нового пароля на вказаний email. Користувач має ввести отриманий токен та новий пароль та підтвердити їх. Система має змінити пароль користувача на новий у разі коректності введених даних.

Таблиця 1.19 – Опис функціональної вимоги REQ006

Номер	REQ006
Назва	Отримання списку доступних фільмів
Опис	Користувач має можливість переглядати список доступних фільмів.

Таблиця 1.20 – Опис функціональної вимоги REQ007

Номер	REQ007
Назва	Вибір дати
Опис	Користувач має можливість переглядати список доступних фільмів за бажаною датою.

Таблиця 1.21 – Опис функціональної вимоги REQ008

Номер	REQ008
Назва	Перегляд інформації про фільм
Опис	Користувач має можливість переглянути деталі обраного фільма.

Таблиця 1.22 – Опис функціональної вимоги REQ009

Номер	REQ009
Назва	Перегляд доступних сеансів
Опис	Користувач має можливість переглядати список доступних сеансів на обраний фільм.

Таблиця 1.23 – Опис функціональної вимоги REQ010

Номер	REQ010
Назва	Перегляд схеми залу та вільних місць
Опис	Користувач має можливість переглядати схему зали з позначеними на ней вільними місцями.

Таблиця 1.24 – Опис функціональної вимоги REQ011

Номер	REQ011
Назва	Вибір бажаних місць для покупки
Опис	Користувач має можливість вибирати на схемі залу вільні місця для покупки.

Таблиця 1.25 – Опис функціональної вимоги REQ012

Номер	REQ012
Назва	Покупка квитків
Опис	Авторизований користувач має можливість купити квитки на обраний фільм.

Залежності між функціональними вимогами клієнтського застосунку зображено на рисунку 1.2

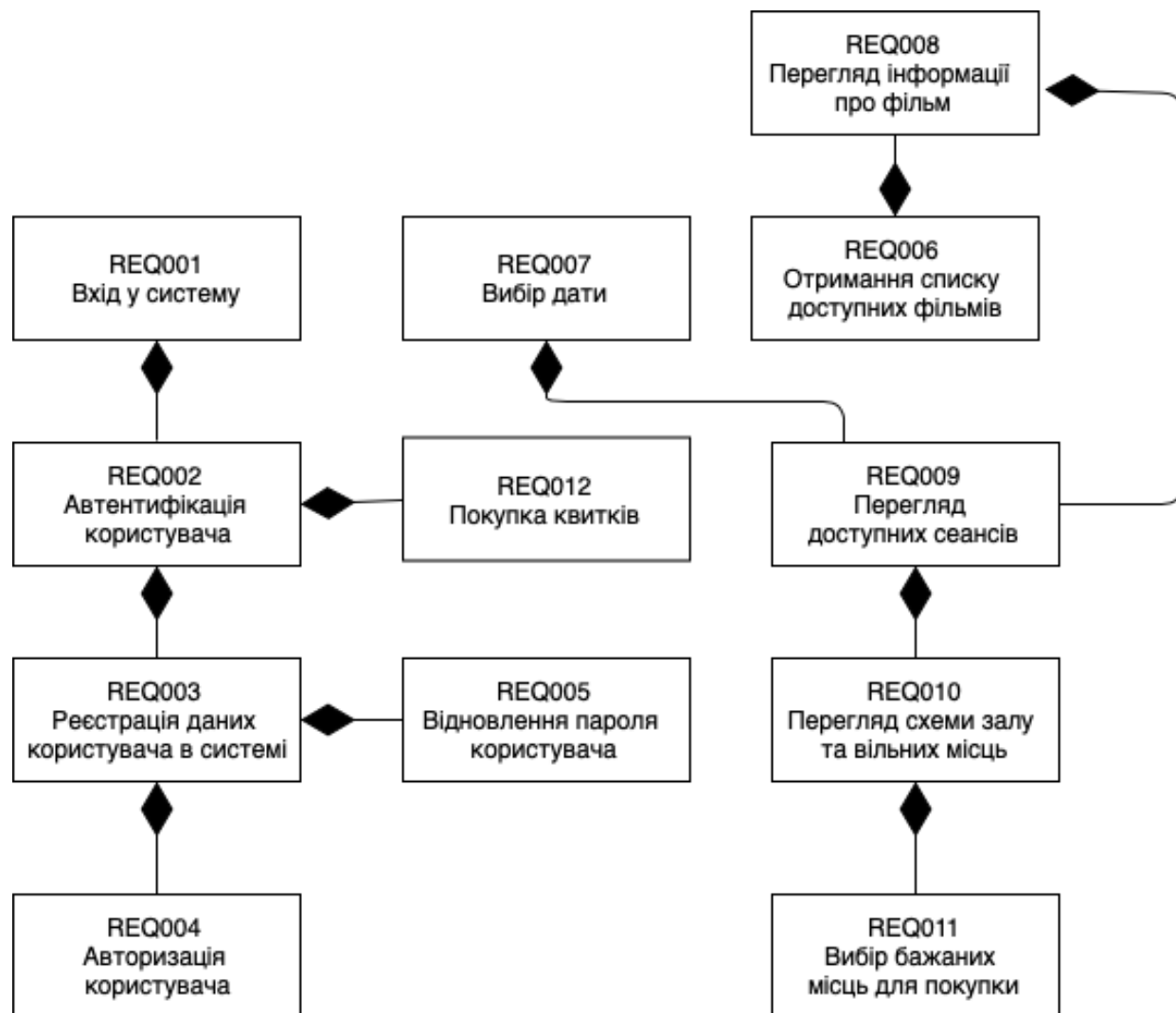


Рисунок 1.2 – Діаграма залежності між функціональними вимогами

1.4.2 Розроблення нефункціональних вимог

Програмне забезпечення повинне відповідати наступним нефункціональним вимогам:

- локалізація інтерфейсу – англійська для англійської локалізації пристрою;
- мінімальна підтримувана версія ОС для мобільних додатків iOS 12.
- клієнт-серверна взаємодія повинна бути реалізована у вигляді викликів REST API;
- авторизація повинна бути реалізована за технологією Bearer Auth.

1.4.3 Постановка комплексу завдань модулю

Розроблюване програмне забезпечення призначене для автоматизації процесу купівлі квитків в кінотеатр.

Мета створення даної роботи – підвищення ефективності процесу вибору та купівлі квитків в кінотеатр за допомогою використання інтегрованої інформаційної системи, яка містить серверне застосування та мобільний додаток.

Для досягнення мети даної роботи система повинна вирішувати наступні задачі:

- Автентифікація користувача.
- Реєстрація користувача.
- Авторизація користувача.
- Демонстрація списку доступних фільмів.
- Вибір бажаної дати.
- Демонстрація інформації про фільм та розкладу сеансів.
- Можливість вибору місця, купівлі та перегляду куплених квитків.

Мобільний застосунок повинен працювати на пристроях зі встановленою операційною системою iOS 12 та вище. Бізнес-логіка має бути реалізована на серверній частині застосунку.

1.5 Висновки по розділу

Автоматизація продажу квитків широко застосовується у підприємницькій діяльності. Основним завданням автоматизації є оптимізація діяльності, що, своєю чергою, призводить до зростання фінансових показників. Як засіб автоматизації в сучасних умовах раціонально використовувати інформаційні системи, які будуть задовольняти вимоги користувачів.

Наразі використовується ряд рішень для онлайн перегляду та покупки квитків, кожне має ряд переваг та недоліків. Жодна з переглянутих систем не має повного списку зазначеного вище функціоналу, являючись при цьому універсальною, а не націленою на окрему мережу кінотеатрів. Також, більшість існуючих рішень не має оптимізованих та сучасних мобільних додатків, які дозволяють користувачам повністю самостійно обирати місця та зберігати квитки на своєму смартфоні, що дає можливість самостійно проходити у кінозал без необхідності звертатися до каси.

Розробка системи, яка містила б також функціональність, що було описано вище, була б затребуваним рішенням серед аналогів. Додатковим фактором, який вказує на зручність використання подібної системи, є використання саме мобільного застосунку як клієнтської частини, адже це найбільш популярні рішення, за допомогою яких користувачі взаємодіють з онлайн сервісами.

Мобільний застосунок повинен мати сучасний інтерфейс та підтримувати як старі, так і актуальні моделі смартфонів. Серверне програмне забезпечення повинне містити можливість добавляти, змінювати, та видаляти фільми, сеанси, та схеми кінозалів.

					КПІ.ІП-5109.045490.01.81	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

2.1.1 Клієнтська частина

Для створення якісного програмного забезпечення необхідно попереднє детальне моделювання його бізнес процесів і архітектури. Для проектування бізнес-процесів було обрано методологію створення діаграм Sequence Diagram.

Схема містить загальні процеси, які проходить користувач під час використання застосунку, а також додаткові процеси, які не входять у потік загальних дій користувача. Загальні процеси, які проходить користувач, включають процеси аутентифікації, реєстрації особистих даних, авторизації, перегляд фільмів та покупку квитків.

Послідовний опис етапу входу до системи через мобільний додаток:

- Для реєстрації особистих даних користувач вводить email і пароль, які відправляються на сервер. У разі успішної реєстрації (дані, введені користувачем, знаходяться в базі даних серверної частини, а також немає акаунта, зареєстрованого з такими даними) користувач має можливість авторизуватися використовуючи введені дані.

- Для авторизації користувач також вводить свій email і пароль. Дані відправляються на сервер для підтвердження. У разі успішної авторизації користувач має можливість купляти квитки.

Послідовний опис роботи користувача з мобільним додатком:

- Користувачу продемонстровано список доступних фільмів за обраною датою. Користувач може обрати фільм зі списку фільмів.

- Користувачу продемонстровано інформацію про фільм і список доступних сеансів. Користувач може обрати сеанс.

- Після обрання сеансу на сервер відправляється запит на загрузку

схеми залу та доступних для покупки місць. Після отримання інформації, мобільний додаток демонструє її користувачу.

– Користувач може обрати вільні місця зі схеми залу. Після оплати місця маркеруються як зайняті, а користувачеві у особистий кабінет завантажено куплені квитки. Користувач може купити ще квитки, або повернутися до списку фільмів.

2.1.2 Серверна частина

Серверна архітектура та бізнес-логіка продемонстрована за допомогою sequence – діаграм.

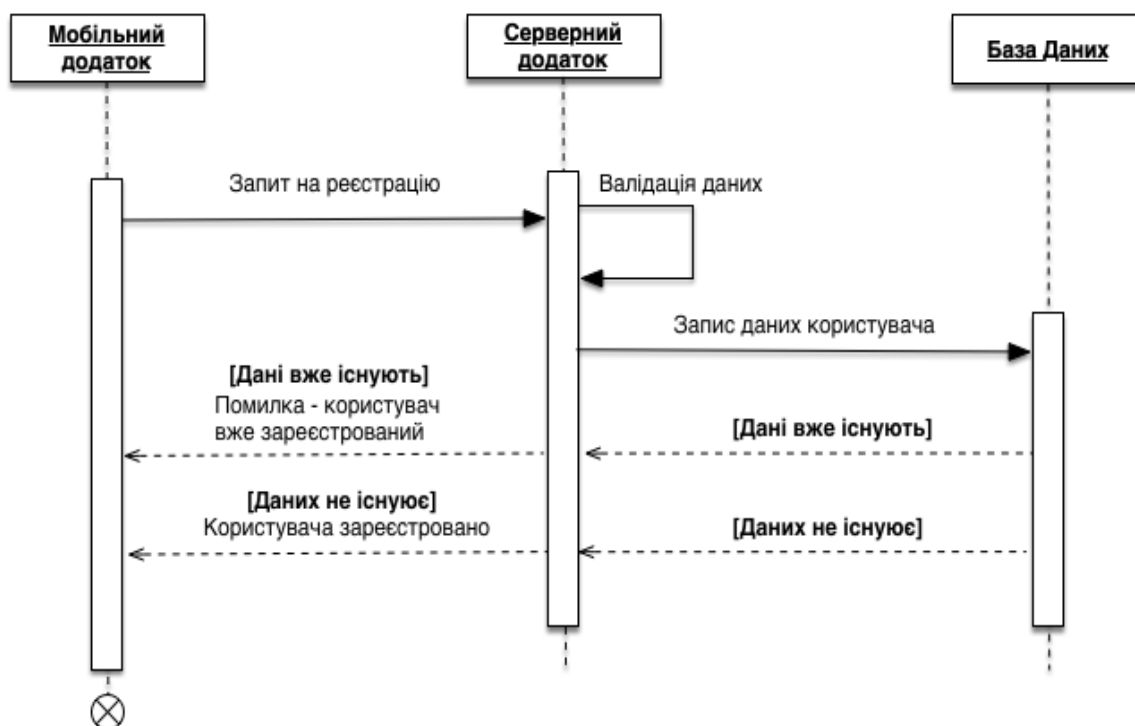


Рисунок 2.1 – Схема реєстрації клієнта

Послідовний опис процесів, зображених на діаграмі для реєстрації клієнта:

- Клієнт надсилає дані користувача для реєстрації на сервер
- Сервер проводить валідацію отриманих даних
- Якщо валідація успішна, сервер пробує записати дані до Бази Даних
- Якщо дані в БД вже існують, сервер надсилає повідомлення про

помилку.

– Якщо даних ще не існує, в БД створюється новий запис, після чого сервер надсилає повідомлення про успішну реєстрацію.

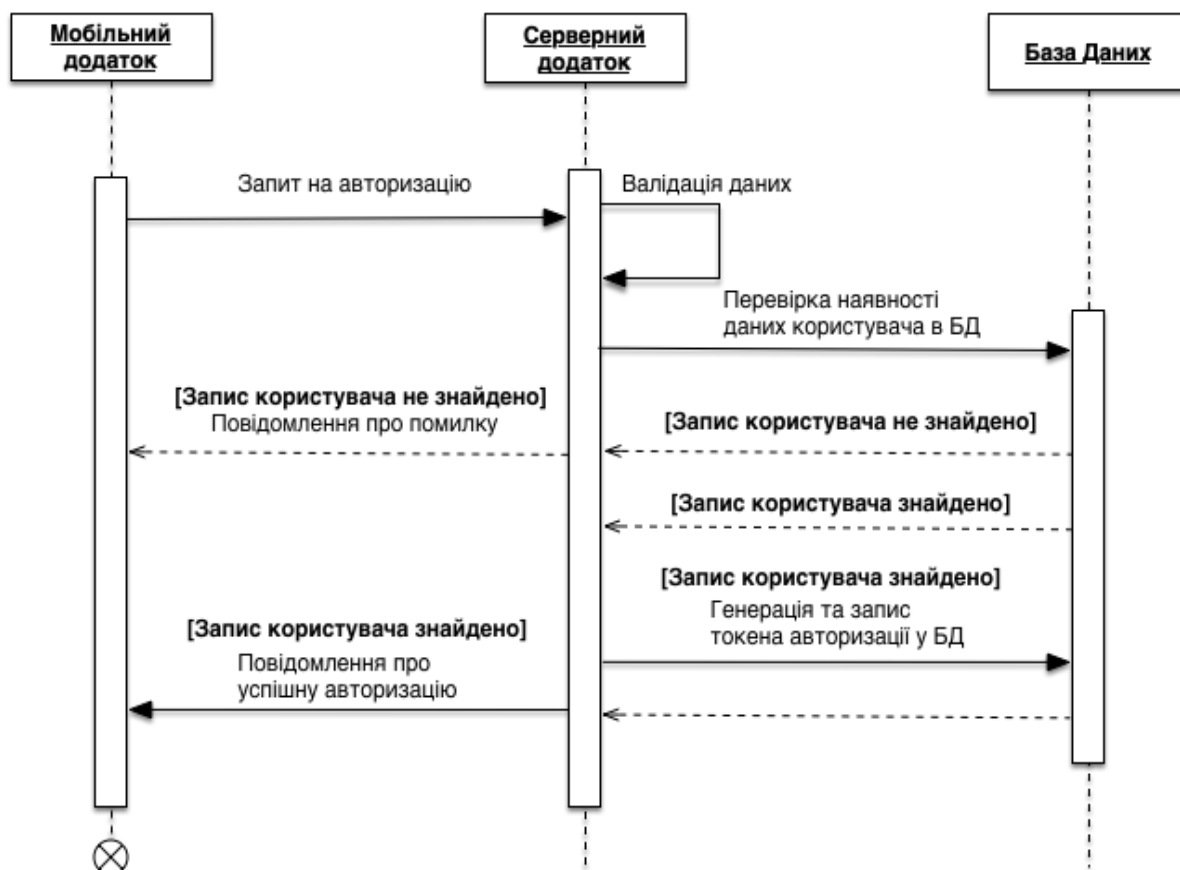


Рисунок 2. 2 – Схема авторизації клієнта

Послідовний опис процесів, зображених на діаграмі для авторизації клієнта:

- Клієнт надсилає запит на авторизацію.
- Сервер проводить валідацію отриманих даних
- Якщо валідація успішна, сервер email та пароль на наявність в БД.
- Якщо пароль не співпадає, сервер надсилає повідомлення про помилку.
- Сервер генерує та записує у БД токен авторизації
- Сервер надсилає повідомлення про авторизацію.



Рисунок 2.3 Схема отримання списку доступних фільмів

Послідовний опис процесів, зображених на діаграмі для отримання списку доступних фільмів:

- Клієнт надсилає запит на отримання списку доступних фільмів.
- Сервер вибирає доступні фільми з Бази Даних.
- Сервер повертає список доступних фільмів.

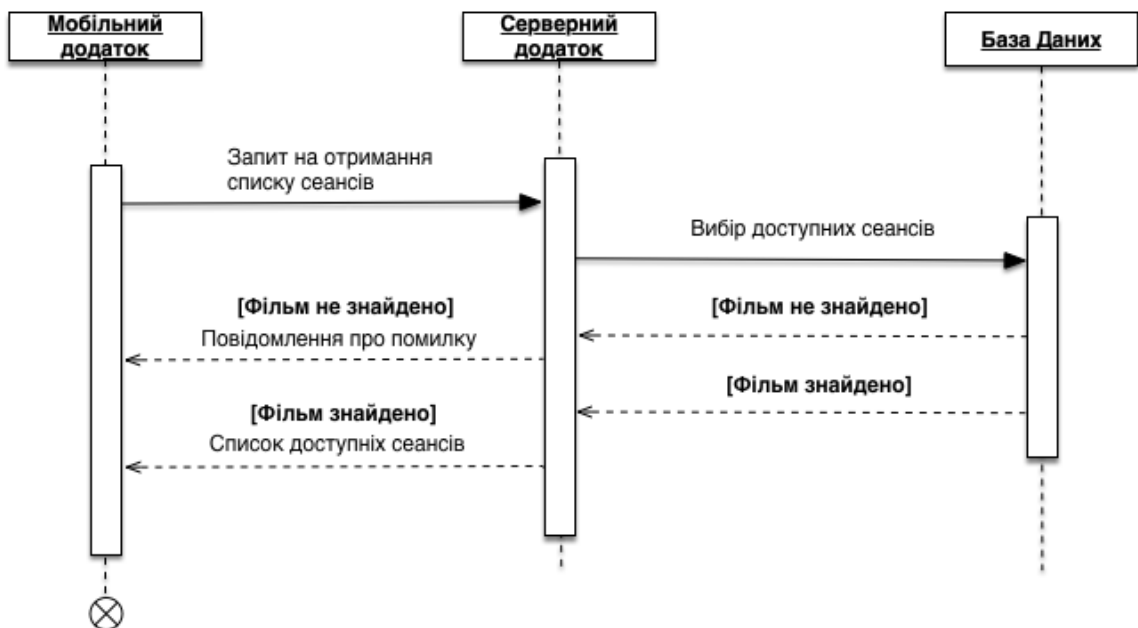


Рисунок 2.4 Схема отримання списку доступних сеансів на обраний фільм

Послідовний опис процесів, зображених на діаграмі для отримання списку доступних сеансів на обраний фільмів:

- Клієнт надсилає запит на отримання списку сеансів.
- Сервер вибирає доступні сеанси Бази Даних.
- Якщо фільм не знайдено, сервер відправляє повідомлення про помилку
- Сервер повертає список доступних сеансів.



Рисунок 2.5 Схема отримання списку вільних місць та схеми залу на обраний сеанс

Послідовний опис процесів, зображених на діаграмі для отримання списку вільних місць та схеми залу на обраний сеанс:

- Клієнт надсилає запит на отримання схеми залу та списку вільних місць.
- Сервер вибирає схему з Бази Даних.

- Якщо сеанс не знайдено, сервер відправляє повідомлення про помилку
- Сервер повертає схему та список вільних місць.

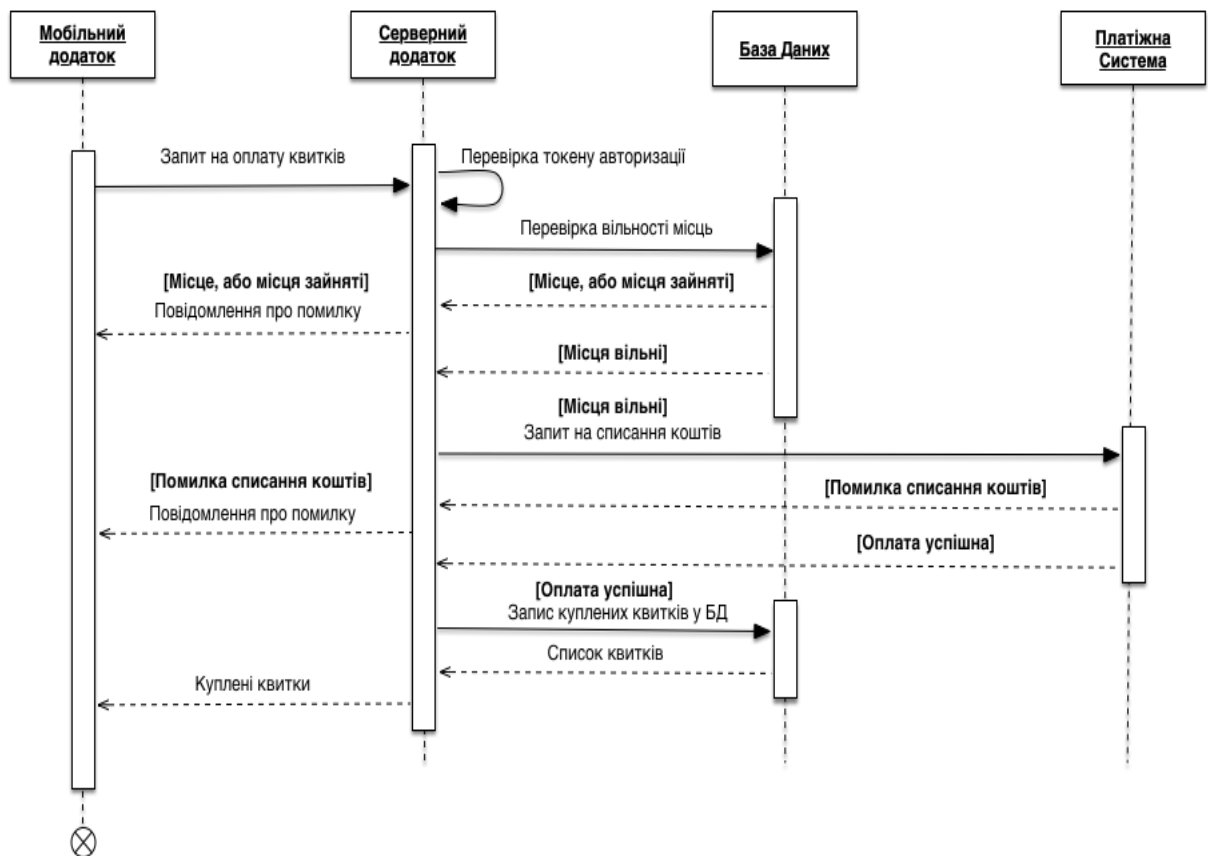


Рисунок 2.6 Схема оплати квитків

Послідовний опис процесів, зображених на діаграмі для оплати обраних квитків:

- Клієнт надсилає запит на оплату квитків.
- Сервер перевіряє токен авторизації.
- Якщо користувач авторизований, сервер перевіряє наявність обраних місць.
- Якщо не всі обрані місця вільні, сервер відправляє повідомлення про помилку.
- Сервер відправляє запит на списання коштів до платіжної системи.
- Якщо оплата не пройшла, сервер відправляє повідомлення про

помилку.

- Сервер записує куплені квитки у Базу Даних.
- Сервер повертає куплені квитки.

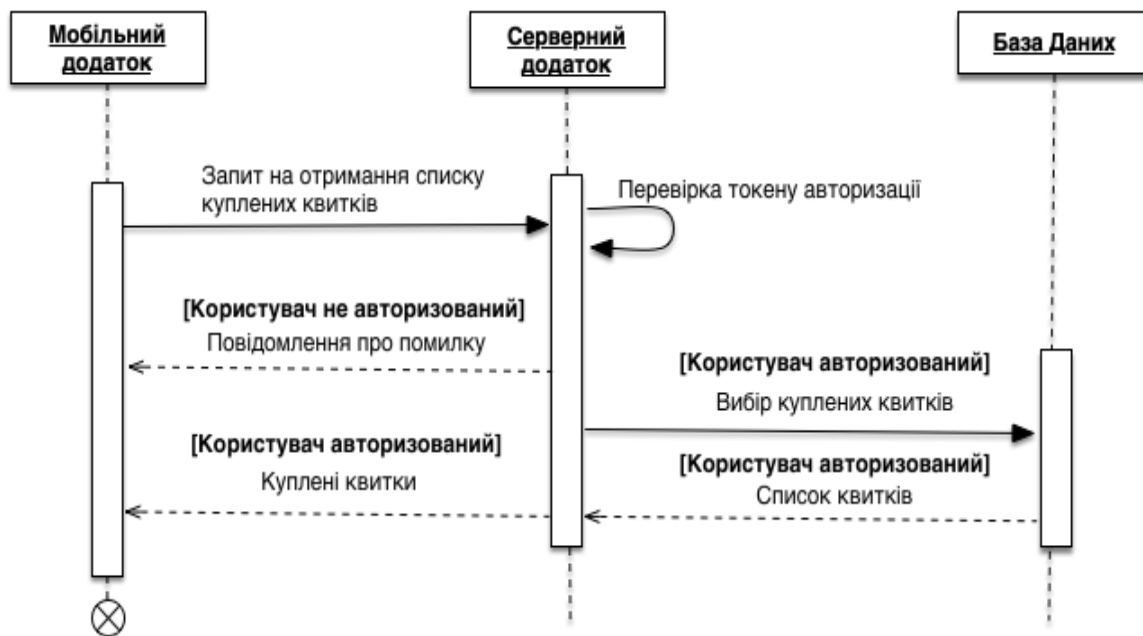


Рисунок 2.6 Схема отримання списку куплених квитків

Послідовний опис процесів, зображених на діаграмі для отримання списку куплених квитків:

- Клієнт надсилає запит на отримання куплених квитків.
- Сервер перевіряє токен авторизації.
- Якщо користувач не авторизований, сервер відправляє повідомлення про помилку.
- Сервер вибирає з Базу Даних квитки, куплені даним користувачем.
- Сервер повертає куплені квитки.

2.2 Архітектура програмного забезпечення

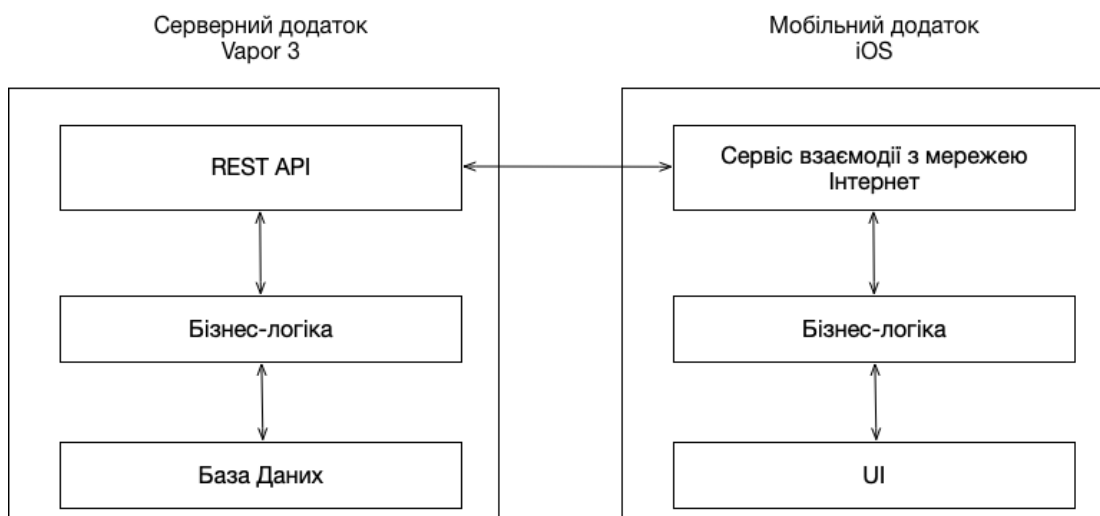


Рисунок 2.7 – Схема клієнт-серверної архітектури системи

Взаємодія між компонентами системи здійснюється за допомогою HTTP запитів до REST API, що підтримуються сервером. REST – це архітектурний стиль взаємодії компонентів розподіленого додатка в мережі. REST є узгоджений набір обмежень, що враховуються при проектуванні розподіленої системи. Властивості архітектури, які залежать від обмежень, накладених на REST-системи:

- Продуктивність - взаємодія компонентів системи може бути домінуючим фактором продуктивності і ефективності мережі з точки зору користувача;
- Масштабованість для забезпечення великої кількості компонентів і взаємодій компонентів.

2.3 Конструювання програмного забезпечення

Для взаємодії з клієнтським застосуном використовують наступні інтерфейси:

- Auth – інтерфейс для процесів авторизації, реєстрації, відновлення та зміни пароля.

- Showtime – інтерфейс для процесів отримання списку фільмів, інформації про фільм, списку сеансів, схеми залу та вільних місць.
- Payment – інтерфейс для процесу оплати квитків.
- Profile – інтерфейс для процесу завантаження куплених квитків.

Таблиця 2.1 Методи інтерфейсу Auth.

Назва метода	Аргументи	Значення, яке повертається	Призначення методу	Коди помилок
register	email, password	HTTP Code	Зареєструвати користувача у системі	001 004
login	email, password	Запис користувача	Увійти у застосунок із вказаними email та паролем	001
resetPassword	email	–	Відновити пароль за вказаним email	003

Таблиця 2.2 – Коди помилок інтерфейсу Auth

Код помилки	Системна назва помилки	Значення помилки
001	ERROR_AUTH_INVALID_CREDENTIALS	Email та/або пароль не пройшли валідацію

Продовження таблиці 2.2

002	ERROR_AUTH_NOT_AUTHENTICATED	Для здійснення дії користувач має пройти аутентифікацію
003	ERROR_AUTH_EMAIL_NOT_FOUND	Користувача з даним email не знайдено
004	ERROR_AUTH_EMAIL_ALREADY_IN_USE	Email вже використовується для іншого акаунта

Таблиця 2.3 Методи інтерфейсу Showtime.

Назва метода	Аргументи	Значення, яке повертається	Призначення методу	Коди помилок
loadMovies	date (Optional)	Масив доступних фільмів	Отримати доступні фільми за обраною датою	
loadMovieSessions	cinemaID, movieID, date	Масив доступних сеансів на обраний фільм у обраному кінотеатрі	Отримати доступні сеанси на обраний фільм у обраному кінотеатрі	101 102

Продовження таблиці 2.3

loadSessionPlaces	sessionID	Схема зали зі списком вільних місць	Отримати схему кінозалу і список вільних для вибору місць	103
loadCinemas	-	Масив доступних кінотеатрів	Отримати список доступних кінотеатрів	

Таблиця 2.4 – Коди помилок інтерфейсу Showtime

Код помилки	Системна назва помилки	Значення помилки
101	ERROR_ST_MOVIE_NOT_FOUND	Фільм не знайдено
102	ERROR_ST_CINEMA_NOT_FOUND	Кінотеатр не знайдено
103	ERROR_ST_SESSION_NOT_FOUND	Сеанс не знайдено

Таблиця 2.5 Методи інтерфейсу Payment.

Назва метода	Аргументи	Значення, яке повертається	Призначення методу	Коди помилок

Продовження таблиці 2.5

buyTickets	[Place]	Масив куплених квитків	Купити квитки на обраний фільм на обрані місця	201 202
validateCard	cardInfo	Повідомлення – валідна карта чи ні	Перевірити чи валідна кредитна картка користувача	202

Таблиця 2.6 – Коди помилок інтерфейсу Payment

Код помилки	Системна назва помилки	Значення помилки
201	ERROR_PAYMENT_PLACE_OCCUPIED	Місце або місця, обрані для покупки, вже зайняті
202	ERROR_PAYMENT_CUSTOM	Помилка від платіжної системи. Містить опис помилки.

Таблиця 2.7 Методи інтерфейсу Profile.

Назва метода	Аргументи	Значення, яке повертається	Призначення методу	Коди помилок
--------------	-----------	----------------------------	--------------------	--------------

Продовження таблиці 2.7

loadTickets	[Ticket]	Масив куплених квітків	Отримати список куплених квітків	301
-------------	----------	------------------------	----------------------------------	-----

Таблиця 2.8 – Коды помилок інтерфейсу Profile

Код помилки	Системна назва помилки	Значення помилки
301	ERROR_PROFILE_NOT_AUTHORIZED	Користувач має пройти аутентифікацію

Структури і типи даних, які використовуються в системі, вказано в таблиці 2.9.

Таблиця 2.9 – Типи даних

Назва типу	Представлення типу	Опис типу
Movie	<pre> { id: Int name: String about: String duration: String coverURL: String launchDate: Date ageRating: String genre: String country: String director: String actors: String is3D: Bool } </pre>	<p>Інформація про фільм.</p> <p>Містить його назву, опис, тривалість, та інше.</p>
Place	<pre> { id: Int sessionID: Int price: Int index: Int row: Int isFree: Bool xPos: Double yPos: Double width: Double } </pre>	<p>Інформація про місце.</p> <p>Містить посилання на сеанс, ціну, рядок та позицію, а також координати лівого верхнього кутка місця на умовній схемі залу, його розмір, та колір на схемі.</p>

Продовження таблиці 2.9

MovieSession	{ id: Int cinemaID: Int movieID: Int date: Date hallName: String }	Інформація про сеанс. Містить посилання на кінотеатр та фільм. Також містить дату сеансу, та назву кінозалу.
User	{ id: Int email: String accessToken: String }	Інформація про користувача. Містить email та токен авторизації.
TicketInfo	{ id: Int placeID: Int movie: Movie session: MovieSession }	Інформація про квиток. Містить посилання на місце, фільм та сеанс.
Cinema	{ id: Int name: String about: String longitude: Double latitude: Double imageURL: String }	Інформація про кінотеатр. Містить назву, опис, координати, та посилання на фотографію кінотеатру.

ключа, що використовується для безпечного зберігання паролів. Вcrypt використовує так звану «сіль» для ускладнення розшифрування паролю паролю, наприклад, через атаку перебором, або за допомогою райдужних таблиць.

2.5 Висновки по розділу

У даному розділі були розроблені структури даних та інтерфейси, що необхідні для функціонування системи, описані та проілюстровані бізнес-процеси розроблюваної системи з використанням діаграм типу Sequence Diagram.

Було побудовано архітектуру клієнтського та серверного застосунків, та їхньої взаємодії. Також було описано і розроблено API. Був обраний механізм для забезпечення безпеки зберігання інформації, та під час її передачі між компонентами системи.

					КПІ.ІП-5109.045490.01.81	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Тестування є невід'ємним етапом у процесі розробки будь-якого програмного забезпечення, як спосіб підтвердження працездатності продукту. У рамках даної роботи необхідність тестування також зумовлена бізнес-логікою програми, а саме взаємодією з онлайн-сервісами.

Тестування програмного забезпечення - це процес, що дозволяє оцінити функціональність програмного додатка з метою з'ясувати, чи відповідає розроблене програмне забезпечення зазначеним вимогам чи ні, та виявити дефекти, щоб гарантувати, що продукт не містить їх після випуску. Відповідно до стандарту ANSI / IEEE 1059, тестування - це «процес аналізу елемента програмного забезпечення для виявлення відмінностей між існуючими і необхідними умовами (тобто, дефектів) і для оцінки особливостей елемента програмного забезпечення».

На ринку існує кілька методів тестування програмного забезпечення, але по суті це є дослідницький процес і для ефективного тестування потрібно урахувати специфіку продукту, що розробляється.

План тестування програмного забезпечення включає в себе підхід, перелік компонентів, що тестуються, критерій проходження тесту, список тестових випадків, та створення звіту про завершення тестування системи, та звіту про помилки, що виникли в системі.

3.2 Опис процесів тестування

3.2.1 Компоненти, що тестуються

- Авторизація користувача
- Реєстрація користувача

- Перегляд доступних фільмів
- Перегляд доступних сеансів
- Перегляд схеми залу
- Вибір вільних місць
- Оплата квитків
- Перегляд куплених квитків

3.2.2 Компоненти, що не тестуються

- СУБД
- АРІ платіжної системи

3.2.3 Підхід до тестування

Підхід до тестування складається з наступних рівнів тестування:

- 1) Модульне тестування. Модульне тестування має бути виконане розробником та затверджене науковим керівником проекту. Модульні тести пишуться до, та під час написання самого проекту. Тести реалізуються за допомогою вбудованого у Cocoa Touch фреймворку XCTest.
- 2) Інтеграційне тестування. Інтеграційне тестування виконується розробником проекту та розпочинається після успішного завершення модульного. Не потребує специфічних інструментів. Також відбувається тестування сценаріїв навігації користувача по мобільному додатку.
- 3) Тестування інтерфейсу. Тестування інтерфейсу виконується тестувальником проекту за допомогою вбудованого у Cocoa Touch фреймворку XCTest. Починається після створення інтерфейсу (не залежно від інших частин проекту). Закінчується після затвердження науковим керівником проекту.

3.2.4 Критерій проходження тесту

Тестування вважається успішно закінченим, якщо не знайдено жодних помилок на усіх рівнях тестування, та виконані усі вимоги, зазначені у технічному завданні.

3.3 Опис контрольного прикладу

Ціль тестування, наведеного у контрольному прикладі – перевірка коректності роботи сценарію покупки квитків користувачем. У таблиці 3.1 продемонстровано покроковий опис тестування, що включає дії користувача, очікуваний та реальний результати.

Таблиця 3.1 – опис контрольного прикладу

Дія користувача	Очікуваний результат	Реальний результат
Користувач обирає фільм зі списку фільмів.	На екрані демонструється опис фільму, та кнопка «Купити квитки».	На екрані продемонстровано опис фільму, та кнопка «Купити квитки».
Користувач натискає на кнопку «Купити квитки».	На екрані демонструється список доступних сеансів.	На екрані продемонстровано список доступних сеансів.
Користувач обирає сеанс.	На екрані демонструється схема залу. На схемі позначені вільні та зайняті місця.	На екрані продемонстровано схема залу з позначеними вільними та зайнятими місцями.

Продовження таблиці 3.1

Користувач обирає кілька вільних місць.	Обрані користувачем місця позначаються іншим кольором на схемі. Під схемою з'являється таблиця, що містить більш детальне описання обраних місць.	Обрані користувачем місця позначені іншим кольором на схемі. Під схемою з'явилася таблиця, що містить більш детальне описання обраних місць.
Користувач натискає на кнопку «Оплатити квитки».	Користувачеві пропонується заповнити дані кредитної картки. Після цього з вказаної картки користувача списується зазначена сума, а на екрані демонструються куплені квитки.	Користувачеві пропонується заповнити дані кредитної картки. Після цього з вказаної картки користувача списується зазначена сума, а на екрані демонструються куплені квитки.
Користувач натискає на будь-який із квитків.	На екрані демонструється QR-код.	На екрані продемонстровано QR-код.

3.4 Висновки до розділу

У даному розділі був розглянутий аналіз якості програмного забезпечення, перелічені компоненти, що тестуються, та що не тестуються. Був описаний підхід до тестування з переліком та детальним описом видів тестування, та сформульовано критерій проходження тесту.

Був описаний контрольний приклад тестування, у якому перелічено дії користувача, та відповідні очікуваний та реальний результати.

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Для розгортання серверної частини інформаційної системи, необхідною є наявність компілятора мови програмування Swift 5, та його залежних бібліотек. У випадку Linux він може бути встановлений з офіційного сайту, на OS X компілятор вбудовано за замовчуванням.

Серверний додаток налаштовано для автоматичного розгортання на хмарний сервіс Heroku. Heroku використовує continuous integration для автоматичної збірки та запуску сервісів, після того як їхній код було завантажено у master-гілку Heroku репозиторію. Наглядно цей процес продемонстровано на рисунку 4.1.

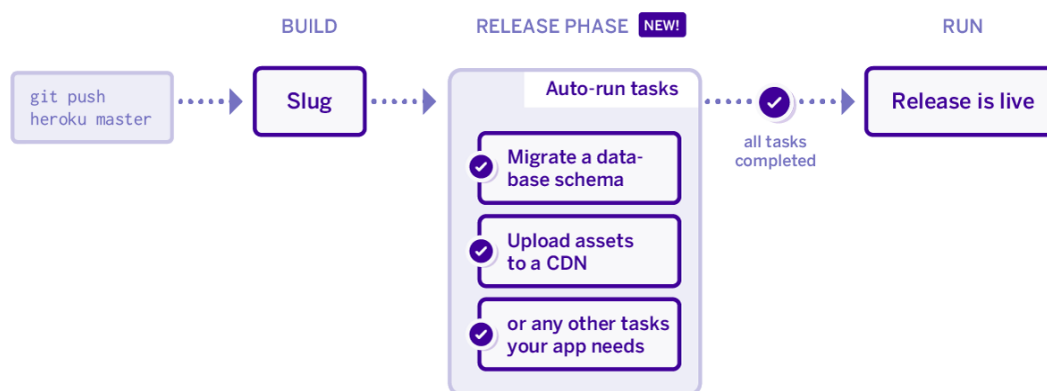


Рисунок 4.1 – процес автоматичного розгортання програмного забезпечення на хмарному сервісі Heroku

Для контролю та перегляду інформації про процес розгортання програмного забезпечення, треба перейти до адміністративної панелі на сайті Heroku, яка продемонстрована на рисунку 4.2. Це несе виключно інформаційний сенс та не є обов'язковим, оскільки весь процес виконується автоматично.

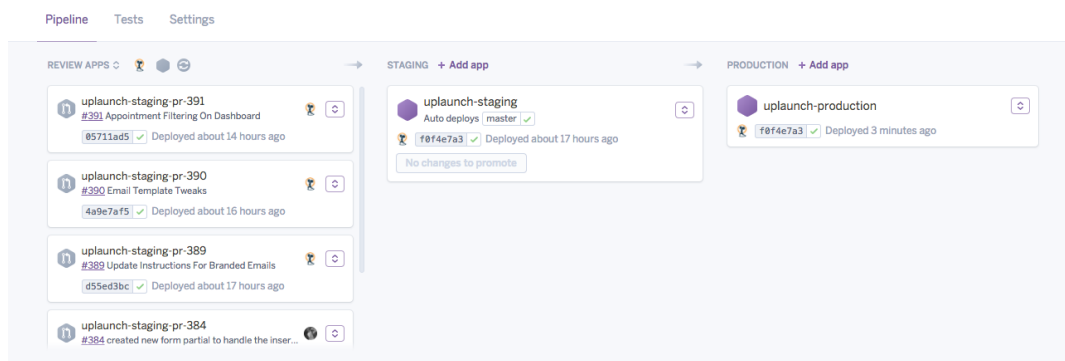


Рисунок 4.2 – адміністративна панель сервісу Heroku

Для встановлення мобільного додатку потрібен мобільний пристрій із встановленою операційною системою iOS 12, або новішою. Додаток може бути завантажений з сервісу App Store – онлайн-магазину додатків для мобільної операційної системи iOS. За необхідності, можна загрузити .ipa файл, підписати його власним сертифікатом, та завантажити на мобільний пристрій власноруч.

4.2 Робота з програмним забезпеченням

Детальну інструкцію роботи із клієнтською частиною програмного забезпечення наведено у додатку “Керівництво користувача”.

4.3 Висновки до розділу

У даному розділі був наведений запропонований сценарій розгортання програмного забезпечення – мобільного та серверного додатків. Був описаний процес розгортання серверного програмного забезпечення за допомогою обраного хмарного сервісу. Був описаний сценарій встановлення мобільного додатку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Cinema Automatization [Електронний ресурс]: (Стаття) http://www.1c.kiev.ua/about/useful_articles/avtomatizatsiya-raboty-kinoteatra-so-spetsializirovannym-po/
2. SaaS [Електронний ресурс]: (Стаття) <https://ru.wikipedia.org/wiki/SaaS>
3. Continuous_integration [Електронний ресурс]: (Стаття) https://en.wikipedia.org/wiki/Continuous_integration
4. Heroku [Електронний ресурс]: (Сайт) <http://www.heroku.com>
5. Mosproject [Електронний ресурс]: (Стаття) <https://mosproject-eng.ru/avtomatizacziya-proczessov.html>
6. Unisender [Електронний ресурс]: (Стаття) <https://www.unisender.com/ru/support/about/glossary/cto-takoe-avtomatizacija-business/>
7. Docker [Електронний ресурс]: (Сайт) <https://www.docker.com/resources/what-container>

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

МОБІЛЬНЕ ЗАСТОСУВАННЯ ДЛЯ ПРИДБАННЯ КВИТКІВ В
КІНОТЕАТР

Опис програми

КПІ.ІІ-5109.045490.02.13

“ПОГОДЖЕНО”

Керівник проекту:

_____ Л.М. Іванова

Нормоконтроль:

_____ М.М. Головченко

Виконавець:

_____ О.С. Кравченко

Київ – 2019 року

Тексти програмного коду

Мобільне застосування для придбання квитків в кінотеатр

(Найменування програми (документа))

DVD-R

(Вид носія даних)

10 арк, 6900 Кб

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

Київ – 2019

					КПІ.ІП-5109.045490.02.13	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		


```

    }
  }
}

extension MoviesVC: MoviesVMDelegate {

  func didLoadAllMovies(movies: [Movie]) {
    MBProgressHUD.hide(for: view, animated: true)
    let viewModels = movies.map { MovieTableViewCellVM(movie: $0) }
    controller.storage.update {
      $0.add(viewModels)
    }
  }

  func didFailWithError(error: Error) {
    MBProgressHUD.hide(for: view, animated: true)
    showErrorAlert(error.localizedDescription)
  }
}
}

```

SelectPlaceVC.swift

```

import UIKit
import MBProgressHUD
import AlisterSwift

class SelectPlaceVC: UIViewController {

  private let contentView = SelectPlaceView()
  private let viewModel: SelectPlaceVM

  private let tableController: TableController

  private let session: MovieSession
  private let movie: Movie

  init(session: MovieSession, movie: Movie) {
    self.session = session
    self.movie = movie
    self.viewModel = SelectPlaceVM(session: session)
    self.tableController = TableController(tableView: contentView.tableView)
    super.init(nibName: nil, bundle: nil)
  }

  required init?(coder aDecoder: NSCoder) {
    fatalError("init(coder:) has not been implemented")
  }

  override func loadView() {
    view = contentView
  }

  override func viewDidLoad() {
    super.viewDidLoad()
    loadData()
    setupContentView()
    configureTableView()
  }

  private func loadData() {
    MBProgressHUD.showAdded(to: contentView, animated: true)
    viewModel.delegate = self
    viewModel.loadPlaces()
  }

  private func setupContentView() {
    contentView.updateMovieInfo(movie: movie, session: session)
    contentView.placeSelectionDelegate = self
    contentView.closeButton.onTap { [weak self] in self?.dismiss(animated: true, completion: nil) }
    contentView.buyTicketsButton.onTap { [unowned self] in
      AppDataManager.isLoggedIn() ? self.buyTickets() : AlertPresenter.showEntrancePopup()
    }
  }

  private func configureTableView() {
    tableController.configureCells {
      $0.register(cell: SelectedPlaceTableViewCell.self, for: SelectedPlaceTableViewCellVM.self)
    }
  }

  private func buyTickets() {
    let price = contentView.buttons.filter { $0.isSelected }
      .map { $0.place }
      .reduce(0, { $0 + $1.price })
    let purchaseVC = PurchasePreviewVC(price: price)
    let naVC = UINavigationController(rootViewController: purchaseVC)
    purchaseVC.delegate = self
    present(naVC, animated: true, completion: nil)
  }
}

extension SelectPlaceVC: SelectPlaceVMDelegate {

  func didLoadPlaces(places: [Place]) {
    MBProgressHUD.hide(for: contentView, animated: true)
    contentView.setupCollectionView(places: places)
  }

  func didBuyTickets(tickets: [TicketInfo]) {
    MBProgressHUD.hide(for: contentView, animated: true)
  }
}

```

										Арк.
										69
ЗМН.	Арк.	№ докум.	Підпис	Дата	КП.ІП-5109.045490.02.13					

```

    dismiss(animated: true, completion: nil)
  }

  func didFailWithError(error: Error) {
    MBProgressHUD.hide(for: contentView, animated: true)
    showErrorAlert(error.localizedDescription)
  }
}

extension SelectPlaceVC: PlaceSelectionViewDelegate {

  func didPickPlace(place: Place) {
    let vm = SelectedPlaceTableViewCellVM(place: place)

    tableController.storage.animatableUpdate {
      $0.add(vm)
    }
  }

  func didRemovePlace(place: Place) {
    guard let vm = tableController.storage.itemsIn(section: 0).first(where: { viewModel -> Bool in
      guard let model = viewModel as? SelectedPlaceTableViewCellVM else { return false }
      return model.place == place
    }) as? SelectedPlaceTableViewCellVM else { return }

    tableController.storage.update {
      $0.remove(vm)
    }
  }
}

extension SelectPlaceVC: PurchasePreviewVCDelegate {

  func proceedPaymentWithToken(_ token: String) {
    MBProgressHUD.showAdded(to: view, animated: true)
    let places = contentView.buttons.filter { $0.isSelected }.map { $0.place }
    viewModel.buyTickets(places: places, token: token)
  }
}

```

SelectPlaceView.swift

```

import UIKit

protocol PlaceSelectionViewDelegate: class {
  func didPickPlace(place: Place)
  func didRemovePlace(place: Place)
}

class SelectPlaceView: UIView {

  private(set) var scrollView: UIScrollView()
  private(set) var tableView: UITableView()
  private let movieInfoLabel = UILabel()
  private(set) var buyTicketsButton = UIButton()
  private(set) var closeButton = UIButton()

  private(set) var buttons: [PlaceButton] = []

  weak var placeSelectionDelegate: PlaceSelectionViewDelegate?

  init() {
    super.init(frame: .zero)
    configureUI()
  }

  required init?(coder aDecoder: NSCoder) {
    fatalError("init(coder:) has not been implemented")
  }

  private func configureUI() {
    backgroundColor = .white

    addSubview(closeButton)
    closeButton.setImage(UIImage(named: "close-button-icon"), for: .normal)
    closeButton.snp.makeConstraints {
      $0.left.top.equalTo(safeAreaLayoutGuide).offset(12)
      $0.size.equalTo(36)
    }

    addSubview(movieInfoLabel)
    movieInfoLabel.font = UIFont.systemFont(ofSize: 16)
    movieInfoLabel.numberOfLines = 2
    movieInfoLabel.snp.makeConstraints {
      $0.top.equalTo(closeButton.snp.bottom).offset(22)
      $0.right.equalTo(safeAreaLayoutGuide)
      $0.left.equalToSuperview().offset(12)
    }

    addSubview(scrollView)
    scrollView.layer.borderColor = UIColor.lightGray.cgColor
    scrollView.layer.borderWidth = 1.0
    scrollView.snp.makeConstraints {
      $0.top.equalTo(movieInfoLabel.snp.bottom).offset(32)
      $0.left.right.equalToSuperview()
      $0.height.equalTo(280)
    }
  }
}

```

						Арк.
						70
ЗМН.	Арк.	№ докум.	Підпис	Дата	КП.ІП-5109.045490.02.13	


```

func format(identifier: String, message: String) -> String {
    return "\($identifier): \($message) \n"
}

func logNetworkRequest(_ request: URLRequest?) -> String {
    var output = ""

    if let httpMethod = request?.httpMethod {
        output += format(identifier: "HTTP Request Method", message: httpMethod)
    }

    output += format(identifier: "Request", message: request?.description ?? "(invalid request)")

    if let headers = request?.allHTTPHeaderFields {
        output += format(identifier: "Request Headers", message: headers.description)
    }

    if let bodyStream = request?.httpBodyStream {
        output += format(identifier: "Request Body Stream", message: bodyStream.description)
    }

    if let body = request?.httpBody, let stringOutput = requestDataFormatter?(body) ?? String(data: body, encoding: .utf8),
isVerbose {
        output += format(identifier: "Request Body", message: stringOutput)
    }

    return output
}

func logNetworkResponse(_ response: HTTPURLResponse?, data: Data?, target: TargetType) -> String {
    guard let response = response else {
        return format(identifier: "Response", message: "⚠ Received empty network response for \($target).")
    }

    var output = ""
    if 200..<400 ~= (response.statusCode) {
        output += "✅"
    } else {
        output += "❌"
    }
    output += format(identifier: "Response", message: "Status Code: \($response.statusCode) URL:\($response.url?.absoluteString ??
"")")

    if let data = data, let stringData = String(data: responseDataFormatter?(data) ?? data, encoding: String.Encoding.utf8),
isVerbose {
        output += stringData
        output += "\n"
    }

    return output
}

fileprivate extension NetworkLoggingPlugin {
    static func reversedPrint(_ separator: String, terminator: String, items: Any...) {
        for item in items {
            print(item, separator: separator, terminator: terminator)
        }
    }
}

func JSONResponseDataFormatter(data: Data) -> Data {
    do {
        let dataAsJSON = try JSONSerialization.jsonObject(with: data, options: [])
        let prettyData = try JSONSerialization.data(withJSONObject: dataAsJSON, options: .prettyPrinted)
        return prettyData
    } catch {
        return data //fallback to original data if it cant be serialized
    }
}

```

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ___ ” _____ 2019 р.

МОБІЛЬНЕ ЗАСТОСУВАННЯ ДЛЯ ОНЛАЙН-ПРИДБАННЯ КВИТКІВ
В КІНОТЕАТР

Технічне завдання

КП.П-5109.045490.03.91

“ПОГОДЖЕНО”

Керівник проекту:

_____ Л.М. Іванова

Нормоконтроль:

_____ М.М. Головченко

Виконавець:

_____ О.С. Кравченко

Київ – 2019 року

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
4.1	ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК.....	6
4.2	ВИМОГИ ДО НАДІЙНОСТІ	7
4.3	УМОВИ ЕКСПЛУАТАЦІЇ	7
4.4	ВИМОГИ ДО СКЛАДУ І ПАРАМЕТРІВ ТЕХНІЧНИХ ЗАСОБІВ.....	7
4.5	ВИМОГИ ДО ІНФОРМАЦІЙНОЇ ТА ПРОГРАМНОЇ СУМІСНОСТІ	8
4.6	ВИМОГИ ДО МАРКУВАННЯ ТА ПАКУВАННЯ.....	9
4.7	ВИМОГИ ДО ТРАНСПОРТУВАННЯ ТА ЗБЕРІГАННЯ	9
4.8	СПЕЦІАЛЬНІ ВИМОГИ.....	9
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	10
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	11
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	12

					КПІ.ІП-5109.045490.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Мобільне застосування для онлайн - придбання квитків в кінотеатр.

Галузь застосування: Кінотеатри.

Наведене технічне завдання поширюється на інтегрованої інформаційної системи, котра використовується для онлайн придбання квитків в кінотеатр та призначена для використання відвідувачами кінотеатрів.

					КП.ІП-5109.045490.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки системи онлайн купівлі квитків в кінотеатр є завдання на дипломне проектування, затверджене кафедрою автоматизованих систем обробки інформації і управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім.Ігоря Сікорського).

					КПІ.ІП-5109.045490.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для використання кінотеатрами та клієнтами.

Метою розробки є підвищення ефективності процесу вибору та купівлі квитків в кінотеатр за допомогою використання інтегрованої інформаційної системи, яка містить серверне застосування та мобільний додаток.

					КПІ.ІП-5109.045490.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1.1 Для користувача:

- Реєстрація;
- Перегляд розкладу доступних фільмів;
- Фільтрація фільмів за датою;
- Перегляд інформації про фільм;
- Вибір місця або місць;
- Покупка квитків;
- Перегляд куплених квитків;
- Генерація QR коду для зчитування квитка.

4.1.2 Розробку клієнтської частини виконати на платформі iOS на мові програмування Swift 5. Розробку серверної частини виконати з використанням серверного фреймворку Vapor 3.

4.1.3 Додаткові вимоги:

- Мобільний додаток має підтримувати версію мобільної операційної системи iOS не нижчу ніж iOS 12;
- Серверна частина повинна бути завантажена на хмарний сервіс Heroku.

					КП.ІП-5109.045490.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

4.2 Вимоги до надійності

4.2.1 Передбачити контроль введення інформації. Дані, що вводяться користувачами повинні проходити валідацію, а саме:

- 1) валідація email
- 2) пароль має містити не менше ніж 3 символи

4.2.2 Передбачити захист від некоректних дій користувача. Система має показувати повідомлення про помилку у випадку будь-яких некоректних дій користувача.

4.2.3 Забезпечити цілісність інформації в базі даних. Доступ до бази даних повинен бути обмежений за допомогою SSH ключів.

4.3 Умови експлуатації

4.3.1 Обслуговування:

Система не потребує періодичного обслуговування.

4.3.2 Обслуговуючий персонал:

Обслуговуючий персонал повинен мати навички роботи з базами даних та розподіленими системами.

4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Програмне забезпечення повинно функціонувати на смартфонах с версією операційної системи iOS 12 або вище. Серверне програмне забезпечення повинно функціонувати на комп'ютерах із встановленою версією операційної системи Ubuntu 16.04.

4.4.2 Мінімальна конфігурація технічних засобів:

Для серверного додатку:

Тип процесору Intel.

					КПІ.ІП-5109.045490.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

Об'єм ОЗП 512 Мб.

Об'єм накопичувача 20 ГБ

Для мобільного додатку:

Версія операційної системи iOS не нижча за 12.0

4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення повинно працювати під управлінням операційних систем сімейства Unix.

4.5.2 Вхідні дані повинні бути представлені в наступному форматі:
Для серверного додатку формат вхідних даних – це JSON об'єкт з певною інформацією.

Для мобільного додатку вхідними даними є взаємодія користувача із пристроєм.

4.5.3 Результати повинні бути представлені в наступному форматі:
Для серверного додатку формат вихідних даних – це JSON об'єкт з певною інформацією.

Для мобільного додатку вихідними даними є оновлення інтерфейсу додатку у відповідності із попередніми діями користувача.

4.5.4 Серверне програмне забезпечення повинно забезпечувати комунікацію із мобільним додатком за допомогою інтерфейсу REST API:

- Авторизація
- Список фільмів
- Список сеансів
- Схема залу
- Купівля квитків

Авторизація повинна бути реалізована за технологією Bearer Auth.
Паролі користувачів мають зберігатись у зашифрованому вигляді.
Для процесу оплати має бути використаний існуючий сервіс платежів,
наприклад, Stripe.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

4.8 Спеціальні вимоги

Згенерувати установчу версію програмного забезпечення. Завантажити і
провести необхідне налаштування серверного програмного забезпечення.

					КПІ.ІП-5109.045490.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка не менше ніж на 50 аркушах формату А4 (без додатків 5.3.2 - 5.3.6).

5.3.2 Технічне завдання.

5.3.3 Керівництво користувача.

5.3.4 Програма та методика тестування

5.4 Графічна частина повинна бути виконана на аркушах формату А3, котрі включаються у якості додатків до пояснювальної записки:

5.4.1 Схеми структури інформаційної системи.

5.4.2 Схеми структури програмного забезпечення.

5.4.3 Схеми функціоналу програмного забезпечення.

5.4.4 Схеми структури потоків даних програмного забезпечення або його частини.

5.4.5 Схеми структурна варіантів використання

5.4.6 Схеми структурна компонент.

5.4.7 Схеми структурна класів програмного забезпечення

					КП.ІП-5109.045490.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

	Назва етапу	Строк,	Звітність
1	Вивчення літератури за тематикою проекту	20.04.19	
2	Розробка технічного завдання	23.04.19	Технічне завдання
3	Аналіз вимог та уточнення специфікацій	25.04.19	Специфікації програмного забезпечення
4	Проектування структури програмного забезпечення, проектування компонентів	28.04.19	Схема структурна програмного забезпечення та специфікація компонентів
5	Програмна реалізація програмного забезпечення	15.05.19	Тексти програмного забезпечення
6	Тестування програмного забезпечення	19.05.19	Тести, результати тестування
7	Розробка матеріалів текстової частини проекту	27.05.19	Пояснювальна записка.
8	Розробка матеріалів графічної частини проекту	07.06.19	Графічний матеріал проекту
9	Оформлення технічної документації проекту	11.06.19	Технічна документація

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

7.1 Види випробувань

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІП-5109.045490.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ___ ” _____ 2019 р.

МОБІЛЬНЕ ЗАСТОСУВАННЯ ДЛЯ ОНЛАЙН-ПРИДБАННЯ
КВИТКІВ В КІНОТЕАТР

Програма та методика тестування

КПІ.ІІ-5109.045490.04.51

“ПОГОДЖЕНО”

Керівник проекту:

_____ Л.М. Іванова

Нормоконтроль:

_____ М.М. Головченко

Виконавець:

_____ О.С. Кравченко

ЗМІСТ

1	АНАЛІЗ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	3
2	ПІДХОДИ ДО ТЕСТУВАННЯ	4
3	КРИТЕРІЇ ПРОХОДЖЕННЯ ТЕСТУВАННЯ.....	5
4	ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ.....	6

1 АНАЛІЗ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування є невід'ємним етапом у процесі розробки будь-якого програмного забезпечення, як спосіб підтвердження працездатності продукту. У рамках даної роботи необхідність тестування також зумовлена бізнес-логікою програми, а саме взаємодією з онлайн-сервісами.

Тестування програмного забезпечення - це процес, що дозволяє оцінити функціональність програмного додатка з метою з'ясувати, чи відповідає розроблене програмне забезпечення зазначеним вимогам чи ні, та виявити дефекти, щоб гарантувати, що продукт не містить їх після випуску. Відповідно до стандарту ANSI / IEEE 1059, тестування - це «процес аналізу елемента програмного забезпечення для виявлення відмінностей між існуючими і необхідними умовами (тобто, дефектів) і для оцінки особливостей елемента програмного забезпечення».

На ринку існує кілька методів тестування програмного забезпечення, але по суті це є дослідницький процес і для ефективного тестування потрібно урахувати специфіку продукту, що розробляється.

План тестування програмного забезпечення включає в себе підхід, перелік компонентів, що тестуються, критерій проходження тесту, список тестових випадків, та створення звіту про завершення тестування системи, та звіту про помилки, що виникли в системі.

У даному плані будуть протестовані наступні функції:

- Авторизація користувача
- Реєстрація користувача
- Перегляд доступних фільмів
- Перегляд доступних сеансів
- Перегляд схеми залу
- Вибір вільних місць
- Оплата квитків
- Перегляд куплених квитків

					КПІ.ІП-5109.045490.04.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

2 ПІДХОДИ ДО ТЕСТУВАННЯ

Підхід до тестування складається з наступних рівнів тестування:

- 1) Модульне тестування. Модульне тестування має бути виконане розробником та затверджене науковим керівником проекту. Модульні тести пишуться до, та під час написання самого проекту. Тести реалізуються за допомогою вбудованого у Cocoa Touch фреймворку XCTest.
- 2) Інтеграційне тестування. Інтеграційне тестування виконується розробником проекту та розпочинається після успішного завершення модульного. Не потребує специфічних інструментів. Також відбувається тестування сценаріїв навігації користувача по мобільному додатку.
- 3) Тестування інтерфейсу. Тестування інтерфейсу виконується тестувальником проекту за допомогою вбудованого у Cocoa Touch фреймворку XCTest. Починається після створення інтерфейсу (не залежно від інших частин проекту). Закінчується після затвердження науковим керівником проекту.

					КПІ.ІП-5109.045490.04.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

3 КРИТЕРІЇ ПРОХОДЖЕННЯ ТЕСТУВАННЯ

Тестування вважається успішно закінченим, якщо не знайдено жодних помилок на усіх рівнях тестування, та виконані усі вимоги, зазначені у технічному завданні.

					КПІ.ІП-5109.045490.04.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

4 ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ

Таблиця 4.1 – опис контрольного прикладу

Дія користувача	Очікуваний результат	Реальний результат
Користувач обирає фільм зі списку фільмів.	На екрані демонструється опис фільму, та кнопка «Купити квитки».	На екрані продемонстровано опис фільму, та кнопка «Купити квитки».
Користувач натискає на кнопку «Купити квитки».	На екрані демонструється список доступних сеансів.	На екрані продемонстровано список доступних сеансів.
Користувач обирає сеанс.	На екрані демонструється схема залу. На схемі позначені вільні та зайняті місця.	На екрані продемонстровано схема залу з позначеними вільними та зайнятими місцями.
Користувач обирає кілька вільних місць.	Обрані користувачем місця позначаються іншим кольором на схемі. Під схемою з'являється таблиця, що містить більш детальне описання обраних місць.	Обрані користувачем місця позначені іншим кольором на схемі. Під схемою з'явилася таблиця, що містить більш детальне описання обраних місць.

Продовження таблиці 4.1

Користувач натискає на кнопку «Оплатити квитки».	Користувачеві пропонується заповнити дані кредитної картки. Після цього з вказаної картки користувача списується зазначена сума, а на екрані демонструються куплені квитки.	Користувачеві пропонується заповнити дані кредитної картки. Після цього з вказаної картки користувача списується зазначена сума, а на екрані демонструються куплені квитки.
Користувач натискає на будь-який із квитків.	На екрані демонструється QR-код.	На екрані продемонстровано QR-код.

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ___ ” _____ 2019 р.

МОБІЛЬНЕ ЗАСТОСУВАННЯ ДЛЯ ОНЛАЙН-ПРИДБАННЯ
КВИТКІВ В КІНОТЕАТР

Керівництво користувача

КПІ.ПІ-5109.045490.05.34

“ПОГОДЖЕНО”

Керівник проекту:

_____ Л.М. Іванова

Нормоконтроль:

_____ М.М. Головченко

Виконавець:

_____ О.С. Кравченко

ЗМІСТ

1	КЕРІВНИЦТВО КОРИСТУВАЧА	3
---	-------------------------------	---

Змн.	Арк.	№ докум.	Підпис	Дата	КПІ.ІП-5109.045490.05.34	Арк.
						2

1 КЕРІВНИЦТВО КОРИСТУВАЧА

При запуску мобільного додатку демонструється список доступних фільмів на поточну дату, як показано на рисунку 1.1.

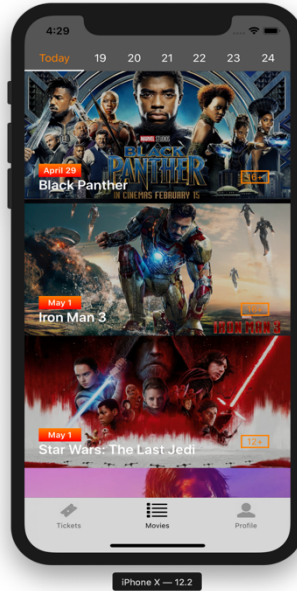


Рис. 1.1 – список фільмів

Вибрати іншу дату можна якщо зробити свайп по екрану у потрібному напрямку. Свайп справа наліво призведе до вибору наступної дати, свайп зліва направо – попередньої (рисунок 1.2).

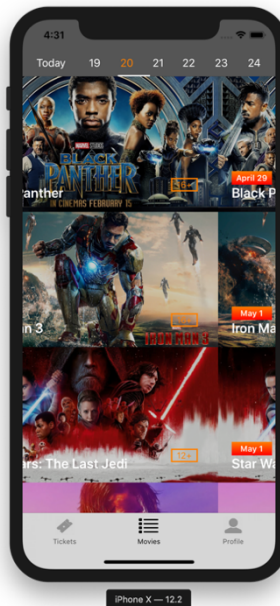


Рис. 1.2 – зміна дати

					КПІ.ІП-5109.045490.05.34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

Щоб продивитися повну інформацію про фільм потрібно натиснути на картинку обраного фільму. Буде відкрито екран з детальним описом фільму, що продемонстровано на рисунку 1.3.

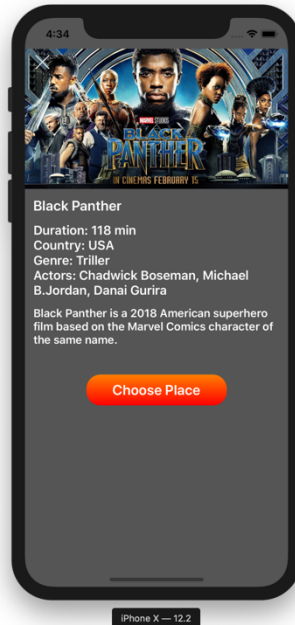


Рис. 1.3 – інформація про фільм

Щоб продивитись доступні сеанси треба натиснути на кнопку «Обрати Місця». Після цього знизу з'явиться панель зі списком доступних сеансів, рис. 1.4.

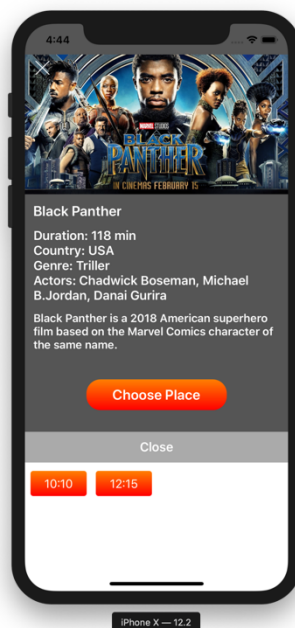


Рис. 1.4 - доступні сеанси

Після вибору сеансу відкриється екран зі схемою кінозалу, що показано на рисунку 1.5.

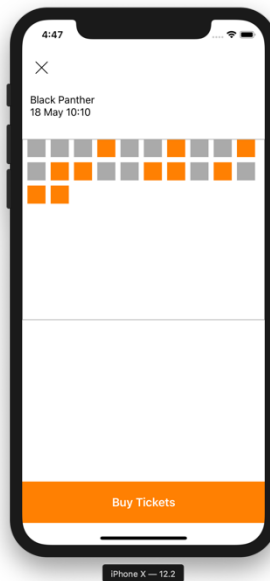


Рис. 1.5 – схема кінозалу

Вільні місця можна вибирати, натискаючи на них. Обрані місця з'являться у таблиці знизу, рис. 1.6.

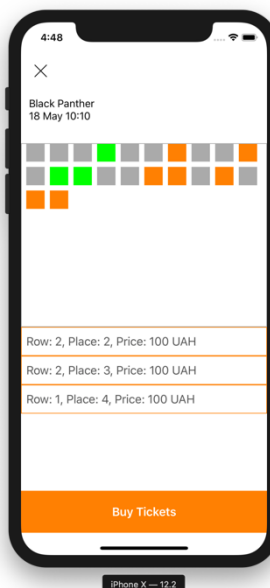


Рис. 1.6 – вибір місць

Після вибору місць, для того, щоб купити квитки, потрібно натиснути на кнопку «Купити квитки». Відкриється інтерфейс платіжної системи, куди потрібно ввести дані кредитної картки, що демонструє рисунок 1.7.

					КПІ.ІП-5109.045490.05.34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

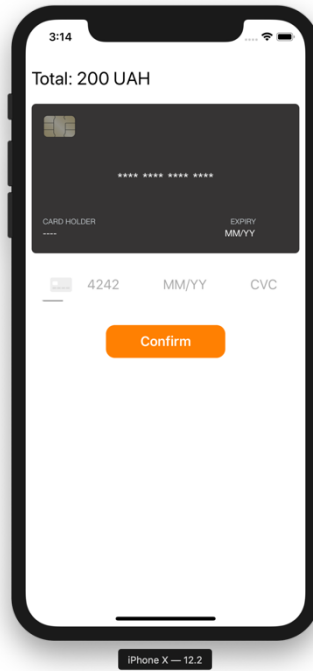


Рис. 1.7 – платіжна інформація

Після успішної оплати квитки з'являться у спеціальному розділі, куди можна потрапити натиснувши на крайню ліву кнопку на панелі навігації знизу, рис. 1.8.

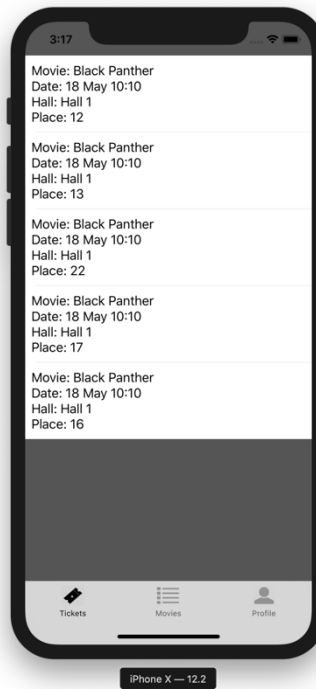


Рис. 1.8 – Список куплених квитків

					КПІ.ІП-5109.045490.05.34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

Для того щоб побачити згенерований QR код треба натиснути на обраний квиток. QR код відкривається у модальному вікні, що зображено на рисунку 1.9.



Рис. 1.9 – QR код

Щоб переглянути власний профіль треба натиснути на крайню ліву іконку на панелі навігації знизу, рис. 1.10.

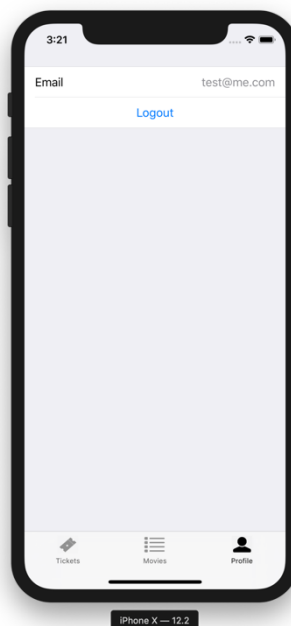


Рис. 1.10 - профіль

					КПІ.ІП-5109.045490.05.34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ___ ” _____ 2019 р.

МОБІЛЬНЕ ЗАСТОСУВАННЯ ДЛЯ ОНЛАЙН-ПРИДБАННЯ
КВИТКІВ В КІНОТЕАТР

Графічний матеріал

КПІ.ІІ-5109.045490.06.99

“ПОГОДЖЕНО”

Керівник проекту:

_____ Л.М. Іванова

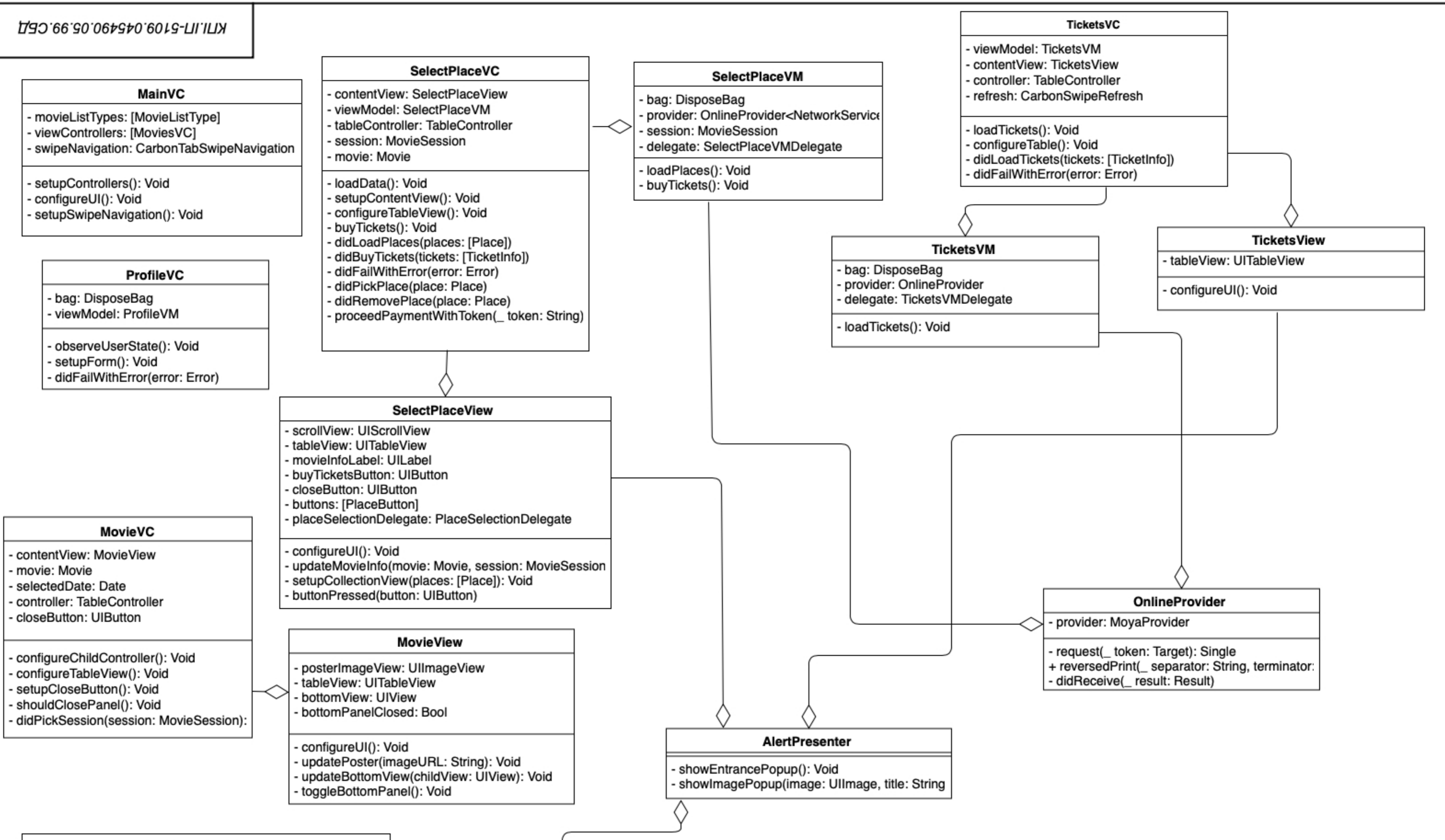
Нормоконтроль:

_____ М.М. Головченко

Виконавець:

_____ О.С. Кравченко

Київ – 2019 року



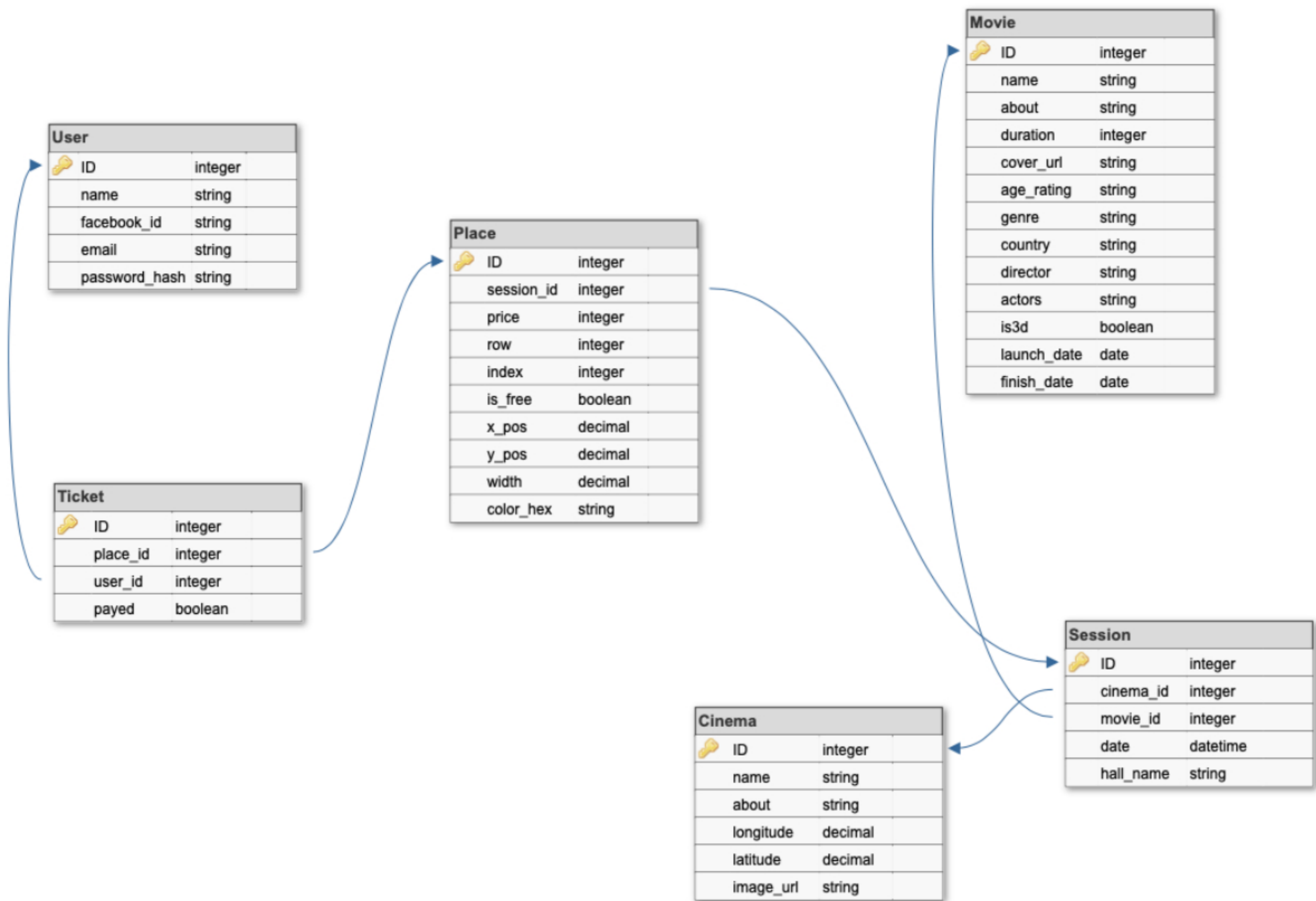
Зм.	Арк.	№ документа	Підпис	Дата
Розробив		Кравченко О.С.		
Перевірів		Іванова Л.М.		
Т. кон.				
Н. кон.		Ліщук К.І.		
Затвердив		Іванова Л.М.		

КЛІ.ІП-5109.045490.05.99.СБД

Схема структурна класів програмного забезпечення

Мобільне застосування для придбання квитків в кінотеатр

Літера	Маса	Масштаб
Аркуш	Аркушів	
КЛІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51		



User		
🔑 ID	integer	
name	string	
facebook_id	string	
email	string	
password_hash	string	

Ticket		
🔑 ID	integer	
place_id	integer	
user_id	integer	
payed	boolean	

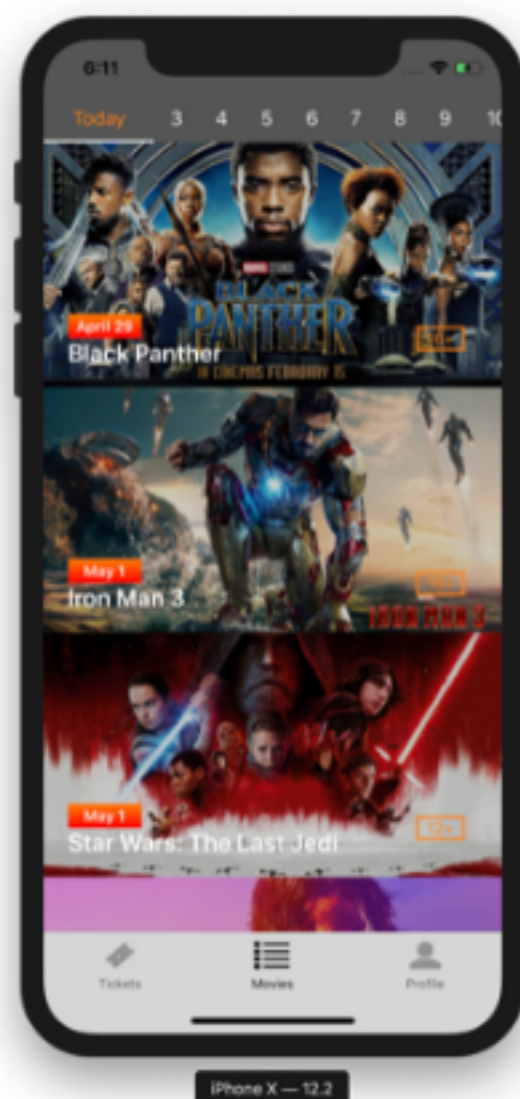
Place		
🔑 ID	integer	
session_id	integer	
price	integer	
row	integer	
index	integer	
is_free	boolean	
x_pos	decimal	
y_pos	decimal	
width	decimal	
color_hex	string	

Cinema		
🔑 ID	integer	
name	string	
about	string	
longitude	decimal	
latitude	decimal	
image_url	string	

Movie		
🔑 ID	integer	
name	string	
about	string	
duration	integer	
cover_url	string	
age_rating	string	
genre	string	
country	string	
director	string	
actors	string	
is3d	boolean	
launch_date	date	
finish_date	date	

Session		
🔑 ID	integer	
cinema_id	integer	
movie_id	integer	
date	datetime	
hall_name	string	

					<i>KPI.IT-5109.045490.05.99.CBD</i>				
Зм.	Арк.	№ документа	Підпис	Дата	Схема бази даних		Літера	Маса	Масштаб
Розробив		Кравченко О.С							
Перевірив		Іванова Л.М							
Т. кон.							Аркуш	Аркушів	
Н. кон.		Ліщук К.І.			Мобільне застосування для придбання квитків в кінотеатр		КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51		
Затвердив		Іванова Л.М							



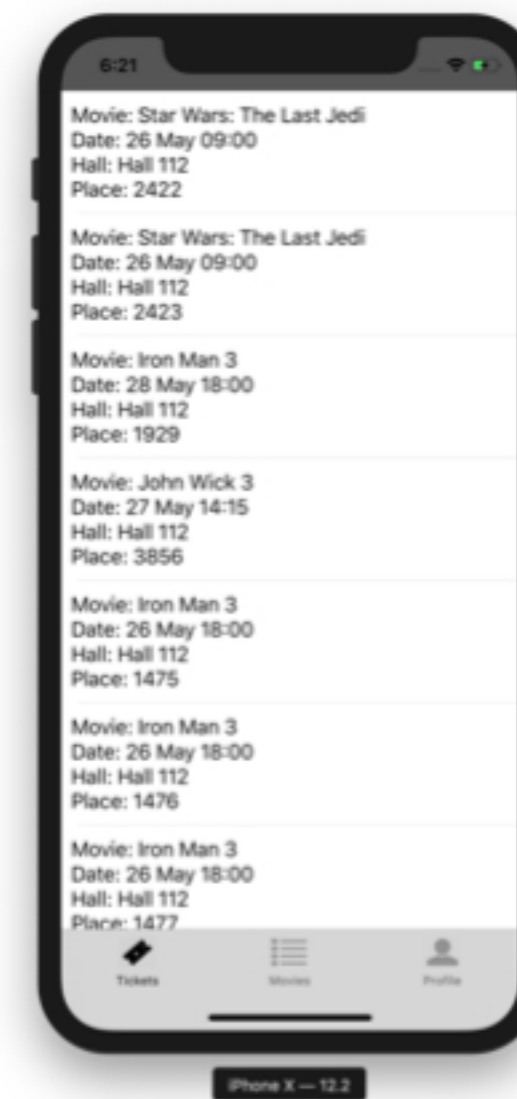
Список фільмів



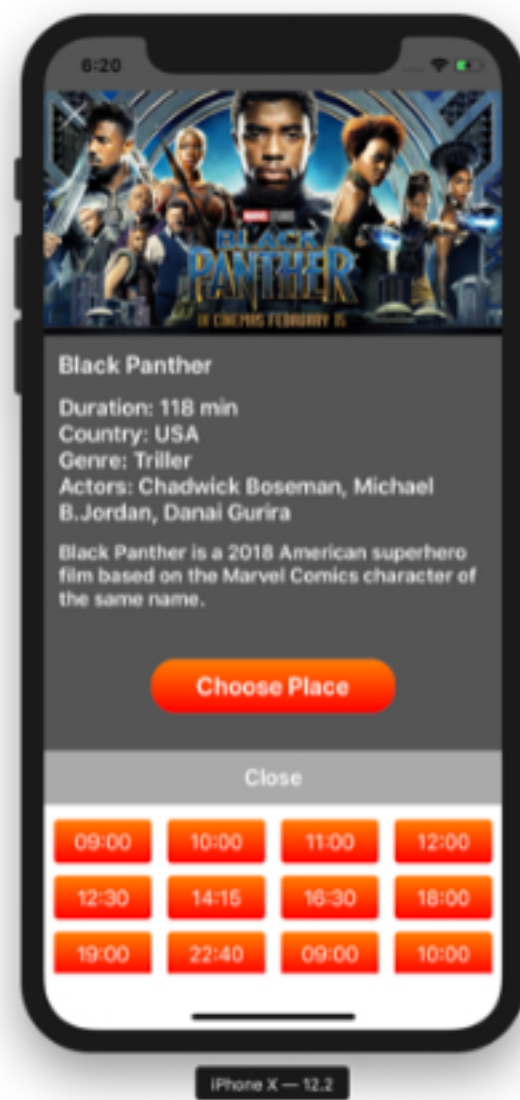
Платіжна інформація



QR код



Список квитків



Список сеансів



Вибір місць

					КПІ.ІП-5109.045490.05.99.СБД			
Зм.	Арк.	№ документа	Підпис	Дата	Креслення виду екранних форм	Літера	Маса	Масштаб
Розробив		Кравченко О.С.						
Перевірив		Іванова Л.М.			Мобільне застосування для придбання квитків в кінотеатр	Аркуш	Аркушів	
Т. кон.								
Н. кон.		Ліщук К.І.			КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51			
Затвердив		Іванова Л.М.						