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

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК_004.896____

«До захисту допущено»
Завідувач кафедри
_____ О.В. Коваль
(підпис) (ініціали, прізвище)
"___"_____2018р.

Магістерська дисертація

зі спеціальності - 121 Інженерія програмного забезпечення
за спеціалізацією - Програмне забезпечення розподілених систем
на тему: «Інструментальні засоби визначеннння активності учасників колективної розробки»

Виконав: студент ____курсу, групи_ТВ-з7123МП__
___Васюк Наталія Сергіївна_ _____ _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник _к.т.н., доц. Гагарін О. О.____ _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент_____ _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2018

Національний технічний університет України
“ Київський політехнічний інститут ім. Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Інженерія програмного забезпечення
за спеціалізацією - Програмне забезпечення розподілених систем

ЗАТВЕРДЖУЮ
Завідувач кафедри
Коваль О.В. _____
(прізвище, ініціали) (підпис)
«____» _____ 2018р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

_____Васюк Наталія Сергіївна_____
(прізвище, ім’я, по батькові)

1. Тема дисертації __Інструментальні засоби визначеннння активності учасників колективної розробки_____
науковий керівник ___Гагарін Олександр Олександрович к. т. н., доц._____
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету №___ від “___”_____20___року
2. Строк подання студентом дисертації _____
3. Об’єкт дослідження Інструментальні засоби визначеннння активності учасників колективної розробки_____
4. Предмет дослідження __ побудова зв’язків у інструментальних засобах визначеннння активності учасників колективної розробки_____
5. Перелік питань, які потрібно розробити_____
1) Інструментальні засоби;
2) проаналізувати існуючі методи формалізації контенту в інструментальних засобах;
3) розробити алгоритм побудови зв’язків програмного комплексу;
4) розробити правила наглядної візуалізації інструментальних засобів;
5.) створити програмний продукт на основі розроблених правил.

6. Дата видачі завдання «_29_»__вересня____ 2017 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	29.09.17р.	
2	Збір інформації	09.10.17р. – 25.01.18р.	
3	Аналіз вимог завдання, вибір методів і засобів розв'язання поставленої задачі	26.01.18р. – 05.09.18р.	
4	Підготовка публікацій	19.07.18р., 02.09.18р.	
5	Підготовка доповідей на конференції	15.08.18р., 20.10.18р.	
6	Підготовка дисертації	03.07.18р. – 07.12.18р.	
7	Розробка програмного продукту	10.01.18р. – 09.10.18р.	
8	Захист програмного продукту	24.10.18р.	
9	Передзахист	28.11.18р.	
10	Захист	17.12.18р.	

Студент _____ ___Васюк Н.С._____
(підпис) (прізвище та ініціали)

Науковий керівник _____ ___Гагарін О.О._____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Структура й обсяг дипломної роботи.

Магістерська дисертація складається зі вступу, 6 розділів, висновку, переліку посилань з 36 найменувань, 2 додатки, і містить 15 рисунків, 20 таблиць. Повний обсяг магістерської дисертації складає 82 сторінок, з яких перелік посилань займає 4 сторінки, додатки – 9 сторінок.

Актуальність теми. В умовах колективної розробки програмного забезпечення як ніколи актуальними є інструменти для організації цього процесу. Завдяки автоматизації даного процесу можна було б досягти значного прискорення і ефективності у процесі роботи, більш впорядковане зберігання даних, а також їх аналіз. Завдяки рейтингу працівників можна найбільш ефективно розподіляти задачі між ними. Публічний ресурс на сьогоднішній день зазвичай має форму веб чи мобільного додатка, що є дуже популярною, а тому і звичною та зручною для користувачів ПК та мобільних девайсів реалізацією такої системи. В світі веб додатків на даний момент важкі обчислення потребують оптимізацій, більше того для систем реального часу в яких час є обмеженим ресурсом це питання постає особливо гостро.

Мета дослідження полягає у визначенні можливих способів реалізції системи для визначення ефективності учасників колективної розробки.

Для досягнення поставленої задачі були сформульовані наступні **завдання дослідження**, що визначили логіку дослідження та його структуру:

- проаналізувати та виявити найбільш вдалі технології для розробки програмного продукту;
- проаналізувати існуючі алгоритми побудови рейтингу працівників;
- вивести формулу для розрахунку ефективності працівників;

- розробити систему для колективної розробки.

Об’єктом дослідження є процеси обігу задач та проектів, а також побудова рейтингу користувачів в таких системах.

Предметом дослідження є пошук шляхів оптимізації та прискорення обміну задачами та побудови рейтингу.

Методи дослідження. Розв’язання поставлених задач виконувались такими засобами:

- дослідження сучасних технологій для прискорення будь-яких обчислень в браузерному середовищі. Пошук найефективніших варіантів алгоритмів побудови рейтингу;
- перевірка технологій та методів оптимізації емпіричним шляхом, побудова середовища для тестування обміну задачами та побудови рейтингу.

Практичне значення одержаних результатів роботи полягає в розробці програмного продукту, який буде надавати функції системи ведення задач та проектів, та побудови рейтингу працівників на основі цих даних.

Ключові слова. *Алгоритм побудови рейтингу*, *СУБД, таск-менеджер, оптимізація обчислювальних процесів*.

ABSTACT

The structure and volume of the thesis.

The master's dissertation consists of an introduction, 6 sections, a conclusion, a list of references from 36 titles, 2 applications, and contains 15 figures, 20 tables. The full volume of the master's dissertation is 82 pages, of which the list of links takes 4 pages, applications - 9 pages.

Topicality of the theme. In the conditions of collective software development, the tools for organizing this process are never more relevant. Thanks to the automation of this process, it would be possible to achieve significant acceleration and efficiency in the process of work, more orderly storage of data, as well as their analysis. Due to the ranking of employees, the most efficient way is to distribute tasks between them. Today's public resource usually takes the form of a web or mobile application that is very popular, and therefore the usual and user-friendly PC and mobile devices implementation of such a system. In today's world of web applications, heavier computing needs to be optimized, moreover, for real-time systems where time is a limited resource, this issue is particularly acute.

The purpose and problems of research is to identify possible ways of implementing the system to determine the effectiveness of participants in collective development.

To accomplish the task, the following research objectives were formulated, which determined the logic of the research and its structure:

- analyze and identify the most successful technologies for developing a software product;
- analyze existing algorithms for building a rating of employees;
- withdraw a formula for calculating the efficiency of employees;
- develop a system for collective development.

The object of research are processes for turning tasks and projects, as well as building user ratings in such systems.

The subject of research are the search for ways to optimize and accelerate the exchange of tasks and build a rating.

Research methods. The solution of the set tasks was carried out by the following means:

− research of modern technologies for acceleration of any calculations in the browser environment. Search for the most effective variants of image processing algorithms;

− testing technologies and methods for empirical optimization, building a medium for testing the speed of image processing.

The practical significance of the results work is to develop a software product that will provide functions for the task and project management system, and build a ranking of employees based on these data.

Keywords. *Algorithm of rating construction, DBMS, task manager, optimization of computational processes.*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД	База даних
ПЗ	Програмне забезпечення
HTML	Hyper Text Markup Language — Мова розмітки гіпертекстових документів
IDE	Integrated development environment — Інтегроване середовище розробки
UI	User Interface — Інтерфейс користувача
ПК	Персональний комп'ютер
XML	Extensible Markup Language — Розширювана мова розмітки
CSV	Comma-separated values — Значення, розділені комою
API	Application programming interface – Прикладний програмний інтерфейс

ВСТУП

Ефективе управління проектами і спілкування - це ряд дій, які виконує певна людина в групі людей, які працюють разом. Частіше за все це менеджер або директор.

Завдяки автоматизації даного процесу можна було б досягти значного прискорення і ефективності у процесі роботи, більш впорядковане зберігання даних, завдяки яким завжди можна було б отримати актуальну інформацію про співробітників та ефективність їх роботи. Швидкий пошук, а також значно більш зручний, швидкий спосіб організації та збереження даних, аналіз та побудова рейтингу дозволить керівнику заощадити багато часу при аналізі ефективності роботи певного працівника і команди вцілому.

Програмне забезпечення такого типу є одним з найбільш затребуваним в компаніях. Неперервний потік задач в компаніях змушує шукати шляхи, для забезпечення категоризації та правильного розподілення задач між працівниками.

Метою даної роботи є програмна реалізація системи, яка дозволить ефективно працювати в компанії над певною задачею всій команді, а також аналізувати керівнику дані по роботі працівників

Завдання полягає у створенні веб-додатку , який дозволяє користувачам швидко та без попередньої підготовки користуватися ним, та ефективно виконувати задачі по проектах. На основі вже існуючих данних по виконанню задач система будуватиме певний ретинг ефективності працівників.

Розробка надійної і ефективної системи можлива тільки при наявності достатньо потужного інструментарію і оволодіння принципами його вмілого застосування. Однією з найбільш поширених методик розробки подібних систем є схема розробки MVC. Систему для керування задачами доцільно розробляти як клієнт-серверний додаток. Засобами для розробки такої системи доцільно обрати

наступні: мова програмування php, а також СКБД MySQL, як досить популярні і підтримувані технології в наш час.

1.ЗАДАЧА РОЗРОБКИ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ВИЗНАЧЕННЯ АКТИВНОСТІ УЧАСНИКІВ КОЛЕКТИВНОЇ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ

Оскільки існує проблема керування процесами в різних компаніях, була поставлена задача створити засіб визначення активності учасників колективної розробки програмної системи.

Цю задачу намагаються вирішити вже декілька років різні компанії. Наприклад, компанія Atlassian зі своєю системою Jira дозволяє ефективно керувати проектами, окрім цього ведеться активна разробка плагінів для системи сторонніми програмістами за окрему плату. Є можливість знайти рішення практично на всі задачі, які необхідно вирішити. Jira залишається лідером в цій сфері.

Ще одним досить популярним інструментом цього призначення є Asana. Працює швидко, інтерфейс компактний. Багато гарячих клавіш, що може помітно прискорити роботу. Завдання створюються як розширені to-do, що теж непогано.

Перечислені вище системи попри свою популярність мають ряд недоліків - це платне користування а також незначний базовий функціонал, якого недостатньо для ефективної роботи команди. Для комфортної та ефективної роботи в Jira необхідно купувати ряд плагінів для розширення базового функціоналу. В Asana важко стежити за великою кількістю проектів - щоб знайти завдання, потрібно відкривати кожен проект і переглядати його. До того ж, немає клієнтського доступу і обліку часу. Тому в результаті цих досліджень постала задача релізувати новий програмний продукт, який дозволятиме ефективно визначати активність учасників колективної розробки програмної системи на основі данних, що містяться в системі.

Система має бути створена для роботи на сервері, оскільки основні

користувачі використовують браузер для роботи.

Система повинна уміти:

– авторизація та аутентифікація користувача;

– реєстрація користувача з підтвердженням пошти;

– генерування посилання для завершення реєстрації;

– відправка повідомлення для завершення реєстрації;

– аналізувати статус реєстрації, у разі непідтвердження електронної адреси протягом трьох днів видаляти дані про користувача з системи;

– дозволяти вхід користувачів на основі введених електронної адреси та паролю;

– давати змогу користувачу відновити забутий пароль;

– перевіряти коректність введених даних в формах входу та відновлення паролю;

– розрізняти три ролі користувачів - адміністратор, проектний менеджер і працівник та видавати їм привілегії в залежності від ролі;

– адміністратор повинен мати найвищі привілегії при роботі з системою, такі як - реєстрація нових користувачів, призначенння ролей користувачам, створення , редагування , видалення проектів а також призначення проектного менеджера на проект; створення нових задач, пизначення задач працівникам, а також перегляд рейтингу працівників по всіх проектах;

– проектний менеджер повинен мати можливість виконувати різні маніпуляції з проектом, на який він призначений - редагувати, створювати задачі в проекті а також переглядати рейтинг працівників зі свого проекту

– звичайний користувач повинен мати можливість переглядати текст задач, на які він призначений, а також змінювати їх статус;

– у своєму кабінеті користувач повинен мати можливість відредагувати дані про себе;

– у системі має бути розумний пошук по задачах, який дозволяє знайти задачі по назвах та виконавцях;

– система має включати різні параметри для задачі - важкість, важливість, тип, кількість повернень, оцінка проектного менеджера;

– кожна задача переведена в статус “виконано” має бути відображена у проектного менеджера в кабінеті у вкладці “List to be approved” , де проектний менеджер вирішуватиме прийняти задачу чи відхилити, у разі прийняття необхідно буде оцінити якість виконання задачі;

– система повинна зберігати дані про всіх користувачів, проекти та задачі в базі даних.

Користувач матиме змогу отримати запрошення на пошту від адміністратора системи, відредагувати інформацію про себе і виконувати дії в системі згідно з привілегіями його ролі.

Висновки до розділу 1

Задача розробки інструментальних засобів визначення активності учасників колективної розробки програмної системи полягає в тому, що створена система міститиме весь необхідний функціонал для ефективної роботи в команді і ефективної оцінки кожного працівника за данними системи.

2. НАУКОВА НОВИЗНА В ІНСТРУМЕНТАЛЬНИХ ЗАСОБАХ ВИЗНАЧЕННЯ АКТИВНОСТІ УЧАСНИКІВ КОЛЕКТИВНОЇ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ

В рамках дисертації були виконані наукові дослідження способів покращення інструментальних засобів визначення активності учасників колективної розробки.

Перед початком досліджень було проаналізовано декілька способів покращення роботи в команді, в тому числі:

- система штрафів за порушені терміни виконання;
- система співпраці декількох працівників над однією задачею;
- аналіз даних про задачу і побудова на основі цього рейтингу співробітників, що дозволить розподіляти задачі між працівниками найбільш ефективно.

В результаті аналізу був обраний третій варіант як найбільш вдалий при колективній розробці програмної системи.

Для досягнення цілі було проведено дослідження існуючих аналогів.

Існуючі системи не дають усього необхідного функціоналу, або ж ящо дають, то з допомогою платного плагіна тим самим дані системи поставлену проблему не вирішують.

Таким чином було прийняте рішення в розробці нового програмного продукту, який би не тільки надавав можливість керувати задачами та проектами, а ще й давав би рейтинг користувачів в проекті.

Таким чином, наукова новизна розробленого прогамного додатку полягає в тому, що абсолютно бесплатний продукт даватиме можливість ефективно керувати проектами та задачами, а також організувати найбільш продуктивну роботу за рахунок призначення найбільш ефективних працівників на найважливіші задачі, а

рейтинг працівників будуватиметься в системі, базуючись на даних по раніше виконаних задачах.

Розроблений програмний продукт надає можливість отримати готовий і найбільш актуальний рейтинг ефективності працівників. Ця особливість дозволяє заощадити час на розрахунки вручну а також максимально правильно розподілити задачі між працівниками команди.

Висновки до розділу 2

Наукова новизна розробленого прогһамного додатку полягає в тому, що удосконалено спосіб проектування бази даних для мобільних застосунків, що дозволяє збільшити швидкість розробки мобільних застосунків для різних мобільних платформ за рахунок повторного використання схеми бази даних. Окрім цього набуло подальшого розвитку використання шаблонів Mustache в генерації програмного коду для мобільних застосунків на основі заданої схеми бази даних.

3. ТЕХНОЛОГІЇ ДЛЯ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ВИЗНАЧЕННЯ АКТИВНОСТІ УЧАСНИКІВ КОЛЕКТИВНОЇ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ

Для того, щоб створити інструментальні засоби визначення активності учасників колективної розробки програмної системи необхідно дослідити технології, які можуть бути використаними при розробці. Аналіз існуючих технологій є вкрай важливим на ранньому етапі досліджень, оскільки існує величезна кількість варіацій технологій, які можуть бути використані при розробці кінцевого програмного продукту. Вдало обрані технології є рушієм розробки програмного забезпечення і лежать в його основі.

3.1.Інструменти розробника

Для розробки програмного продукту використано середовище розробки PHP Storm, система бази даних MySQL, інтерфейс для СУБД phpmyadmin, веб сервер apache. Розробка велася на ОС Ubuntu 16.04. Для доступу до данного ресурсу необхідний встановлений на комп’ютері будь-який браузер і вихід в інтернет. Розробка велася в веб-браузері Mozila Firefox Developer Edition.

3.1.1 HTTP протокол передачі даних

Мережа інтернет- це велика кількість компютерів сполучених між собою кабелями, радіоканалами, супутниковими каналами зв’язку. У звязку з цим приймаючій і відправляючій стороні необхідно дотримуватися ряду домовленностей, які дозволять строго регламентувати передачу даних і гарантувати що ця передача пройде без ускладнень. Такий набір правил

називається протоколом передачі. Спрощено, протокол - це набір правил, який дозволяє системам взаємодіючим в рамках мережі обмінюватися данними в найбільш зручній для них формі.

Для веб-программмування використовується протокол передачі данних TCP(Transmittion Control Protocol - протокол управління передачею даних), а точніше HTTP - Hyper Transfer Protocol - протокол передачі гіпертекста. На рисунку 3.1 зображений механізм обміну данними в інтернеті.

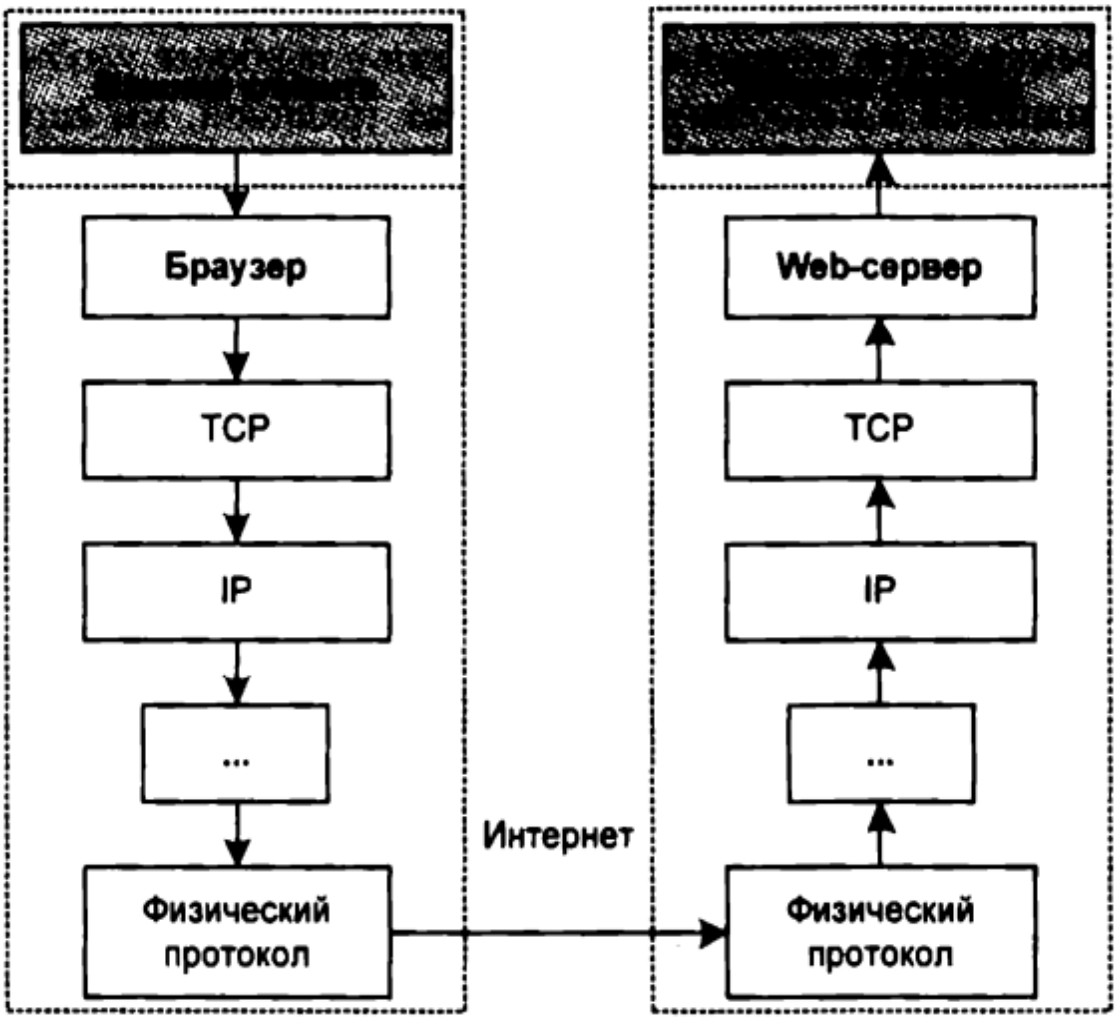

Рисунок 3.1 - механізм обміну данними в інтернеті

HTTP-протокол має ряд правил, яких необхідно дотримуватися при побудові запитів. Кожен запит має складатися з трьох частин - стартовий рядок, заголовки, тіло повідомлення. В даному протоколі розрізняють десять типів запитів:

OPTIONS

Повертає методи HTTP, які підтримуються сервером. Цей метод може служити для визначення можливостей веб-сервера.

GET

Запитує вміст вказаного ресурсу. Запитаний ресурс може приймати параметри (наприклад, пошукова система може приймати як параметр шуканий рядок). Вони передаються в рядку URI (наприклад: http://www.example.net/resource?param1=value1&param2=value2). Згідно зі стандартом HTTP, запити типу GET вважаються ідемпотентними — багатократне повторення одного і того ж запиту GET повинне приводити до однакових результатів (за умови, що сам ресурс не змінився за час між запитами). Це дозволяє кешувати відповіді на запити GET. Якщо назва ресурсу не вказана (у URI наявні лише схема та доменне ім'я), то веб-сервер повертає індекс директорії веб-сервера.

HEAD

Аналогічний методу GET, за винятком того, що у відповіді сервера відсутнє тіло. Це корисно для витягання мета-інформації, заданої в заголовках відповіді, без пересилання всього вмісту. Зокрема, клієнт чи проксі, перевіривши заголовок Last-Modified: (останній час модифікації), таким чином може переконатися, що сторінка на сервері не змінилася від часу попереднього запиту.

POST

Передає призначені для користувача дані (наприклад, з HTML-форми) заданому ресурсу. Наприклад, в блогах відвідувачі зазвичай можуть вводити свої коментарі до записів в HTML-форму, після чого вони передаються серверу методом POST, і він поміщає їх на сторінку. При цьому передані дані (у прикладі з блогами — текст коментаря) включаються в тіло запиту. На відміну від методу GET, метод POST не вважається ідемпотентним, тобто багатократне повторення одних і тих же запитів POST може повертати різні результати (наприклад, після кожного відправлення коментаря з'являтиметься одна копія цього коментаря).

PUT

Завантажує вказаний ресурс на сервер.

PATCH

Завантажує певну частину ресурсу на сервер.

DELETE

Видаляє вказаний ресурс.

TRACE

Повертає отриманий запит так, що клієнт може побачити, що проміжні сервери додають або змінюють в запиті.

CONNECT

Для використання разом з проксі-серверами, які можуть динамічно перемикатися в тунельний режим SSL.

Відповідь сервера має наступні коди статусів:

Коди статусу:

- 1xx — інформаційний: запит прийнятий, продовжуй процес;
- 2xx — успіх: дія була успішно передана, зрозуміла, та прийнята;
- 3xx — перенаправлення: наступні дії мають бути успішно виконані для реалізації запиту;
- 4xx — помилка клієнта: запит містить синтаксичні помилки або не може бути виконаний;
- 5xx — помилка сервера: сервер не зміг виконати правильно сформований запит.

3.1.2. Інструменти для локальної розробки веб-ресурсу

Веб-сервер - це сервер що приймає HTTP запити від клієнтів, частіше за все - веб-браузерів, видає їм HTTP-відповіді. Частіше за все веб-ресурси зберігаються на сторонніх серверах, деколи доцільно розмістити проект на своїй машині, тобто використати свій компютер як сервер. Для того щоб розмістити в себе проект необхідно налаштувати веб-сервер apache2 на своєму компютері.

Для створення проекту необхідно вибрати найбільш зручне місце для зберігання файлів, та додавати їх потроху. Для того щоб ми могли використовувати код, необхідно додати файл virtual host(рисунок 3.2) для нашого ресурсу, в якому прописати дані конфігурації. Наступний крок - додати новий URL в файл hosts.

```
<VirtualHost *:80>
        ServerAdmin admin@test.laravel
        ServerName test.laravel
        ServerAlias www.test.laravel
        <Directory /var/www/test.laravel/diploma/public>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                allow from all
        </Directory>
        DocumentRoot /var/www/test.laravel/diploma/public
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Рисунок 3.2 - Файл vitual host для мого ресурсу

3.1.3. Середовище розробки PhpStrom

JetBrains PhpStorm — комерційне крос-платформове інтегроване середовище розробки для PHP, яке розробляється компанією JetBrains на основі платформи IntelliJ IDEA.

PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3/5.4/5.5/5.6/7.0/7.1 (сучасні і традиційні проекти), включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис коротких масивів. Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів. PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок установки плагінів, розроблених для платформи IntelliJ, або написавши власні плагіни.

PhpStorm надає багатий і інтелектуальний редактор коду для PHP з підсвічуванням коду, розширеною конфігурацією форматування коду, перевіркою на наявність помилок на льоту і розумним автодоповненням.

Середовище розробки має такі можливості:

- підтримка SQL і баз даних. А сааме рефакторинг схеми бази даних, генерація скриптів міграції схеми, експорт результатів виконання запиту у файл або буфер обміну, редагування збережених процедур і багато іншого;
- віддалене розгортання додатків і автоматична синхронізація з використанням FTP, SFTP, FTPS та ін протоколів;
- інтеграція з системами управління версіями (Git - включаючи спеціальний функціонал для роботи з GitHub, Subversion, Mercurial, Perforce, CVS, TFS), що дозволяє робити багато дій, наприклад commit, merge, diff та інші, прямо з PhpStorm;
- локальна історія (Local History) (локально відстежує будь-які зміни в коді);
- PHP UML (Діаграми класів UML для PHP коду з рефакторингом, що викликаються прямо з діаграми);
- підтримка Phing (надає автодоповнення, перевірку стандартних тегів, властивостей, імен цілей, значень атрибутів шляху в компонувальних файлах (build files);
- інтеграція з системами відстеження помилок;
- підтримка Vagrant, SSH консолі і віддалених інструментів;
- підтримка Google App Engine For PHP;
- PhpStorm також дозволяє різні поєднання клавіш для підвищення ефективності.

3.1.4.UML

UML - уніфікована мова моделювання. Мова - графічного опису для обєктного моделювання бізнес-процесів, системного проектування і відображення

організаційних структур. Зараз мова UML – активно використовується для роботи над проектом для детального опису структури системи. Для Ubuntu є UML-редактор Dia(рисунок 3.3), який включає в себе основний набір діаграм .

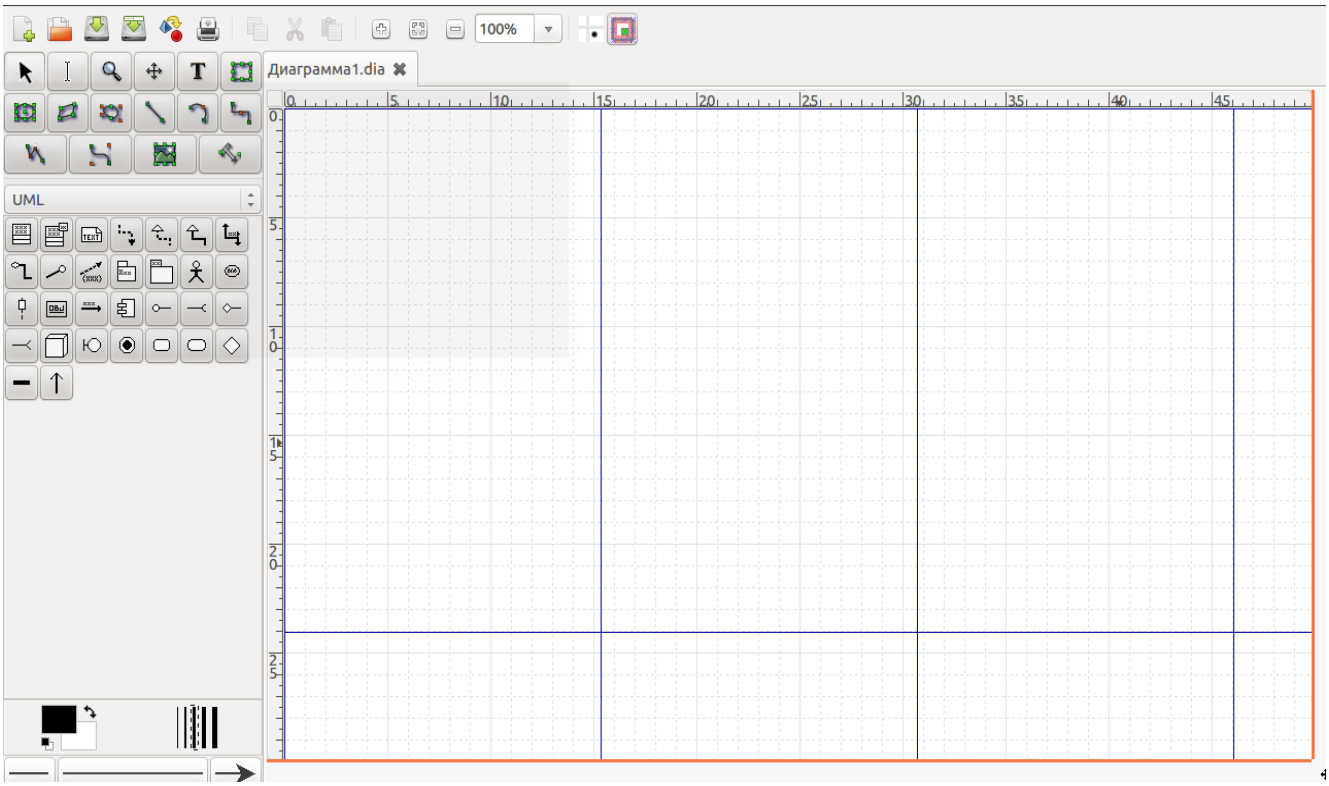

Рисунок 3.3. -конструктор UML діаграм Dia

3.1.5 Система контролю версій Git

Git — система керування версіями файлів та спільної роботи. Проект створив Лінус Торвальдс для управління розробкою ядра Linux, а сьогодні підтримується Джуніо Хамано (англ. Junio C. Hamano). Git є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, також можлива прив'язка цифрових підписів розробників до тегів і комітів. Програма є вільною і випущена під ліцензією GNU GPL версії 2. Система спроектована як набір програм,спеціально розроблених з врахуванням їхнього використання у скриптах. Це дозволяє зручно створювати спеціалізовані системи управління

версіями на базі Git або користувацькі інтерфейси. Наприклад, Cogito є саме таким прикладом фронтенда до репозиторіїв Git. A StGit використовує Git для управління колекцією латок. Система має ряд користувацьких інтерфейсів: наприклад, gitk та git-gui розповсюджуються з самим Git.

Віддалений доступ до репозиторіїв Git забезпечується git-демоном, SSH або HTTP сервером. TCP-сервіс git-daemon входить у дистрибутив Git і є разом з SSH найпоширенішимі надійним методом доступу. Метод доступу

HTTP,незважаючи на ряд обмежень, дуже популярний в контрольованих мережах,тому що дозволяє використання існуючих конфігурацій мережевих фільтрів. Git, на відміну від Subversion і подібних до неї систем, не зберігає інформацію як список змін (патчів) для файлів. Замість цього Git зберігає дані набором зліпків. Кожного разу при фіксації поточної версії проекту Git зберігає зліпок того, як виглядають всі файли проекту. Але якщо файл не змінювався, то дається посилання на раніше збережений файл (див. рис. 1). Git схожий на своєрідну файлову систему з інструментами, які працюють поверх неї. Для кожного відстежуваного файлу Git зберігає розмір, час створення і останньої зміни. Ці дані зберігаються у файлі index, який знаходиться у теці .git Вся база даних Git зберігається в теці з назвою .git. В Git файли можуть знаходитися в одному із 3-х станів: зафіксованому (файл вже збережено в локальній базі даних),зміненому (файл було змінено, але зміни не зафіксовано) і підготовленому(файли було змінено і відмічено для фіксації).

Більшість дій можна виконувати на локальній файловій системі без використання інтернет підключення. Вся історія змін зберігається локально і при необхідності вивантажується у віддалений репозиторій. На відміну від Subversion, де без підключення до інтернету можна лише редагувати файли, але зберегти зміни в вашу базу даних неможливо (оскільки вона відключена від репозиторія). Будь-який коміт спочатку робиться локально.

3.2. Мовні засоби розробки

3.2.1.СУБД MySQL

MySQL був розроблений компанією «TcX» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL включає наступні можливості:

- практично повна реалізація ANSI SQL-99 разом з розширеннями;
- міжплатформенна сумісність;
- незалежні типи таблиць (MyISAM для швидкого читання, InnoDB для транзакцій і цілісності посилань);
- транзакції;
- підтримка SSL;
- кешування запитів;
- реплікація: один головний сервер на одного підлеглого, багато підлеглих на одного головного;
- повнотекстова індексація і пошук з використанням типу таблиць MyISAM;
- імплементована бібліотека бази даних;
- підтримка Юнікоду (UTF-8);
- таблиці InnoDB забезпечують відповідність вимогам ACID;
- вбудований сервер, який дозволяє включати MySQL в автономні додатки;
- вкладені запити і похідні таблиці;
- нова система кодувань і сортувань;

- більш швидкий та гнучкий протокол клієнт-сервера із підтримкою підготовлених запитів, який забезпечує їх оптимальне виконання;
- нова програма установки та налаштування для Microsoft Windows і GNU/Linux;
- захищені через OpenSSL з'єднання між клієнтом та сервером;
- високо-оптимізована бібліотека, яка може бути використана в сторонніх програмах;
- повноцінна підтримка Юнікоду (UTF-8 і UCS2);
- стандартні просторові типи даних GIS для зберігання географічної інформації;
- покращений повнотекстовий пошук та система допомоги;
- збережувані процедури та функції;
- обробники помилок;
- курсори;
- тригери;
- представлення;
- інформаційна схема (так званий системний словник, що містить метадані);
- сегментування — можливість розбити одну велику таблицю на декілька частин, розміщених в різних файлових системах, базуючись на визначеній користувачем функції. При деяких умовах це може дати серйозне збільшення продуктивності та, крім того, полегшує масштабування таблиць;
- змінено поведінку ряду операторів для забезпечення більшої сумісності зі стандартом SQL:2003;
- порядкова реплікація (row-based реплікація), при якій в бінарний лог буде записуватись тільки інформація про реально змінені рядки таблиці замість оригінального (і, можливо, більш повільного) тексту запиту;

- вбудований планувальник робіт, що періодично запускаються. По синтаксису додання задачі схоже на додання тригера до таблиці; по ідеології — на crontab;
- додатковий набір функцій для обробки XML, реалізація підтримки XPath;
- нові засоби діагностики проблем і утиліти для аналізу продуктивності; Розширено можливості з керування вмістом лог-файлів, логи тепер можуть бути збережені і в таблицях general_log і в slow_log. Утиліта mysqlslap дозволяє провести тестування навантаження БД із записом часу реакції на кожний запит;
- для спрощення операції оновлення підготовлена утиліта mysql_upgrade, яка виконає перевірку всіх існуючих таблиць на предмет сумісності з новою версією, і при необхідності виконає належні коригування;
- MySQL Cluster тепер йде як окремий продукт, який базується на MySQL 5.1 і сховищі NDBCLUSTER;
- значні зміни в роботі MySQL Cluster, такі, як, наприклад, можливість зберігання табличних даних на диску;
- повернення до використання вбудованої бібліотеки libmysqld, відсутньої в MySQL 5.0;
- API для плагінів, що дозволяє завантажувати сторонні модулі, які розширюють функціональність (наприклад, повнотекстовий пошук), без перезапуску сервера;
- реалізація парсера повнотекстового пошуку у вигляді plug-in;
- новий рушій таблиць Maria (стійкий до збоїв клон MyISAM)[6], який у 2010 був перейменований на Aria та став основою форку MySQL від Монті Віденіуса під назвою MariaDB.

3.2.2. Мова програмування PHP

PHP — мова, код якої можна вбудовувати безпосередньо html-код сторінок, які у свою чергу, будуть коректно оброблені PHP-інтерпретатором. Обробник PHP просто починає виконувати код після відкриваючого тегу (<?php) і продовжує

виконання до того моменту, поки не зустріне закриваючий тег (?>), або кінець файлу.

Велика різноманітність функцій PHP дає можливість уникати написання багаторядкових функцій, призначених для користувача, як це відбувається в C або Pascal.Наявність інтерфейсів до багатьох баз даних в PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase.через стандарт відкритого інтерфейсу зв'язку з базами даних (Open Database Connectivity Standard — ODBC) можна підключатися до всіх баз даних, до яких існує драйвер.Традиційність Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. Багато конструкцій мови запозичені з C, Perl. Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP — мова, що поєднує переваги Perl та C і спеціально спрямована на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом. І хоча PHP є досить молодою мовою, вона здобула таку популярність серед web-програмістів, що в наш час є найпопулярнішою мовою для створення веб-застосунків (скриптів).Наявність сирцевого коду та безкоштовність

Стратегія Open Source, і розповсюдження початкових текстів програм в масах, безсумнівно справили благотворний вплив на багато проектів, в першу чергу — Linux хоч і успіх проекту Apache сильно підкріпив позиції прихильників Open Source. Сказане відноситься і до історії створення PHP, оскільки підтримка користувачів зі всього світу виявилася дуже важливим чинником в розвитку проекту PHP.

Ухвалення стратегії Open Source і безкоштовне розповсюдження початкових текстів PHP надало неоціненну послугу користувачам. Окрім цього, користувачі PHP в усьому світі є свого роду колективною службою підтримки, і в популярних електронних конференціях можна знайти відповіді, навіть на найскладніші питання.Ефективність

Ефективність є дуже важливим чинником у програмуванні для середовищ розрахованих на багато користувачів, до яких належить і web. Важливою перевагою PHP є те, що ця мова належить до інтерпретованих. Це дозволяє обробляти сценарії з достатньо високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше за аналогічні їм програми, написані на Perl. Проте хоч би що робили розробники PHP, виконавчі файли, отримані за допомогою компіляції, працюватимуть значно швидше — в десятки, а іноді і в сотні разів. Але продуктивність PHP достатня для створення цілком серйозних веб-застосунків.

3.2.3. Фреймворк Laravel5

Безкоштовний, з відкритим кодом PHP-фреймворк, створений Taylor Otwell і призначений для розробки веб-додатків відповідно до шаблону model–view–controller (MVC). Деякі з особливостей Laravel є модульна система упакування з виділеним менеджером залежностей Composer, різні способи для доступу до реляційних баз даних, утиліти, які допомагають в розгортанні додатків і технічного обслуговування

- Пакети (англ. Packages) - дозволяють створювати і підключати модулі в форматі Composer до додатка на Laravel. Багато додаткові можливості вже доступні у вигляді таких модулів.
- Eloquent ORM - реалізація шаблону проектування ActiveRecord на PHP. Дозволяє строго визначити відносини між об'єктами бази даних. Стандартний для Laravel будівник запитів Fluent підтримується ядром Eloquent.
- Логіка програми - частина розробляється, оголошена або за допомогою контролерів, або маршрутів (функцій-замикань). Синтаксис оголошень схожий на синтаксис, який використовується в каркасі Sinatra.
- Зворотній маршрутизація пов'язує між собою генеруються додатком посилання і маршрути, дозволяючи змінювати останні з автоматичним оновленням

пов'язаних посилань. При створенні посилань за допомогою іменованих маршрутів Laravel автоматично генерує кінцеві URL.

- REST-контролери - додатковий шар для поділу логіки обробки GET- і POST-запитів HTTP.
- Автозавантаження класів - механізм автоматичного завантаження класів PHP без необхідності підключати файли їх визначень в include. Завантаження на вимогу запобігає завантаження непотрібних компонентів; завантажуються тільки ті з них, які дійсно використовуються.
- Укладачі уявлень (англ. View composers) - блоки коду, які виконуються при генерації уявлення (шаблону).
- Інверсія управління (англ. Inversion of Control) - дозволяє отримувати екземпляри об'єктів за принципом зворотного управління. Також може використовуватися для створення і отримання об'єктів-одинаків (англ. Singleton).

3.2.4. Шаблон проектування Model View Controller

MVC - архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими за рахунок зменшення складності.

У рамках архітектурного шаблону модель–вигляд–контролер (MVC) програма поділяється на три окремі, але взаємопов'язані частини з розподілом функцій між компонентами. Модель (Model) відповідає за зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача (зміна положення курсора миші, натискання кнопки, ввід даних в текстове поле) і передає дані у модель.

- Модель є центральним компонентом шаблону MVC і відображає поведінку застосунку, незалежну від інтерфейсу користувача. Модель стосується прямого керування даними, логікою та правилами застосунку.
- Вигляд може являти собою будь-яке представлення інформації, одержуване на виході, наприклад графік чи діаграму. Одночасно можуть співіснувати кілька виглядів (представлень) однієї і тієї ж інформації, наприклад гістограма для керівництва компанії й таблиці для бухгалтерії.
- Контролер одержує вхідні дані й перетворює їх на команди для моделі чи вигляду.

Модель інкапсулює ядро даних і основний функціонал їхньої обробки і не залежить від процесу вводу чи виводу даних.

Вигляд може мати декілька взаємопов'язаних областей, наприклад різні таблиці і поля форм, в яких відображаються дані.

У функції контролера входить відстеження визначених подій, що виникають в результаті дій користувача. Контролер дозволяє структурувати код шляхом групування пов'язаних дій в окремий клас. Наприклад у типовому MVC-проекті може бути користувацький контролер, що містить групу методів, пов'язаних з управлінням обліковим записом користувача, таких як реєстрація, авторизація, редагування профілю та зміна пароля.

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати

різні компоненти для відображення інформації. Таким чином, якщо користувач через контролер внесе зміни до моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

3.2.5. Технологія AJAX

AJAX (Asynchronous JavaScript And XML) — підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані. AJAX — один з компонентів концепції DHTML.

Про AJAX заговорили після появи в лютому 2005-го року статті Джесі Джеймса Гарретта (Jesse James Garrett) «Новий підхід до веб-застосунків». AJAX — не самостійна технологія.

AJAX — це не самостійна технологія, а швидше концепція використання декількох суміжних технологій. AJAX-підхід до розробки, який призначений для користувачів інтерфейсів, комбінує кілька основних методів і прийомів:

- використання DHTML для динамічної зміни змісту сторінки;
- використання XMLHttpRequest для звернення до сервера «на льоту», не перезавантажуючи всю сторінку повністю;
- альтернативний метод — динамічне підвантаження коду JavaScript в тег <SCRIPT> з використанням DOM, що здійснюється із використанням формату JSON);
- динамічне створення дочірніх фреймів.

Використання цих підходів дозволяє створювати набагато зручніші веб-інтерфейси користувача на тих сторінках сайтів, де необхідна активна взаємодія з користувачем. AJAX — асинхронний, тому користувач може переглядати далі контент сайту, поки сервер все ще обробляє запит. Браузер не перезавантажує web-сторінку і дані посилаються на сервер без візуального підтвердження (крім

випадків, коли ми самі захочемо показати процес з'єднання з сервером). Використання AJAX стало популярним після того, як компанія Google почала активно використовувати його при створенні своїх сайтів, таких як Gmail, Google Maps і Google Suggest. Створення цих сайтів підтвердило ефективність використання даного підходу.

3.2.6. HTML

HTML (англ. HyperText Markup Language — Мова розмітки гіпертекстових документів) — стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документа та ідеологію структурної розмітки тексту.

Попри те, що HTML — штучна комп'ютерна мова, вона не є мовою програмування.

HTML разом із каскадними таблицями стилів та вбудованими скриптами — це три основні технології побудови веб-сторінок.

HTML впроваджує засоби для:[1]

- створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації із Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.

Семантичний HTML — спосіб написання HTML, що віддає перевагу підкресленню смислу закодованої інформації радше за її подання (зовнішній вигляд). Ще з самого початку свого розвитку HTML мав у складі елементи семантичної розмітки[14], проте також мав і елементи презентаційної розмітки, такі як font, і та center. Також HTML має семантично-нейтральні елементи span та div. З

кінця 1990-х, коли Каскадні таблиці стилів почали належно працювати в більшості браузерів, авторам документів було рекомендовано уникати використання презентаційної розмітки HTML з метою розділення представлення і змісту.

У 2001 році в статті про Семантичну павутину Тім Бернерс-Лі та інші навели приклади шляхів, за якими одного дня «агенти» інтелектуального програмного забезпечення зможуть автоматично прочесати Всесвітню мережу та відшукати, відфільтрувати та встановити співвідношення попередньо непов'язаних фактів на благо користувачів.[16] Такі агенти є незвичайними навіть зараз, але деякі з ідей Web 2.0, мешапів та сервісів порівняння цін стають все ближчими до реалізації. Основна відмінність між цими гібридними веб-застосунками та семантичним агентом, який згадується у статті Бернерса-Лі, полягає в тому, що нинішні шляхи збирання та гібридизації інформації, як правило, створені веб-розробниками, які вже точно знають де шукати потрібну інформацію і яка в неї API-семантика.

Важливими типом веб-агента, який прочісує і читає веб-сторінки автоматично, проте без знання того, що він може виявити, є пошуковий робот.

Цей програмний агент залежить від семантичної ясності веб-сторінок, які він знаходить, оскільки в ньому використовуються різні методи і алгоритми зчитування та індексації мільйонів веб-сторінок в день, що забезпечує користувачів Інтернету пошуковими можливостями, без яких Всесвітня павутина була б корисна тільки на малу частину від її сучасних можливостей.

3.2.7.CSS

Каскадні таблиці стилів (англ. Cascading Style Sheets) — спеціальна мова,що використовується для опису сторінок, написаних мовами розмітки даних.Найчастіше CSS використовують для візуальної презентації сторінок,написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі. CSS має різні рівні та профілі. Наступний рівень CSS

створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

3.2.8. Bootstrap

Bootstrap − це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-додатків.

Bootstrap — це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері. Репозиторій з даним фреймворком є одним з найбільш популярних на GitHub.[4] Серед інших, його використовують NASA і MSNBC.

Bootstrap має модульну структуру і складається переважно з наборів таблиць стилів LESS, які реалізують різні компоненти цього набору інструментів. Розробники можуть самостійно налаштовувати файли Bootstrap, обираючи компоненти для свого проекту.

Основні інструменти Bootstrap:

- Сітки (grid) — наперед задані, готові до використання колонки.
- Шаблони (template) — фіксовані чи адаптивні шаблони сторінок.
- Типографіка (typography) — опис та визначення класів для шрифтів, таких як шрифти для коду, цитат тощо.
- Мультимедіа (media) — засоби управління зображеннями та відео.
- Таблиці (table) — засоби оформлення таблиць, які зокрема забезпечують сортування.
- Форми (form) — класи для оформлення як форм, так і деяких подій.

- Навігація (nav, navbar) — класи для оформлення вкладок, сторінок, меню і панелей навігації.
- Сповіщення (alert) — класи для оформлення діалогових вікон, підказок і спливаючих вікон.
- Іконочний шрифт (icon font) — набір іконок у вигляді шрифту, складається майже з 500 компонентів.

Bootstrap (початкова назва — Twitter Blueprint) був розроблений Марком Отто та Джейкобом Торнтоном (Twitter) у якості фреймворку для забезпечення однаковості внутрішніх інструментів Twitter. До появи Bootstrap у розробці інтерфейсу застосовувалися різні бібліотеки, що призводило до появи суперечностей та ускладнювало супровід. Через кілька місяців до розробки рішення долучилося багато розробників компанії Twitter. Проект було перейменовано з Twitter Blueprint на Bootstrap. Реліз із відкритим сирцевим кодом вийшов 19 серпня 2011 року. Нині проект підтримується невеликою групою розробників на чолі з Марком Отто та Джейкобом Торнтоном,

а також широкою спільнотою прихильників. Bootstrap сумісний з останніми версіями браузерів Google Chrome, Firefox, Internet Explorer, Opera і Safari (деякі з цих браузерів підтримуються не на всіх платформах). Bootstrap має модульну структуру і складається переважно з наборів таблиць стилів LESS, які реалізують різні компоненти цього набору інструментів. Розробники можуть самостійно налаштовувати файли Bootstrap, обираючи компоненти для свого проекту.

3.2.9. JavaScript

JavaScript—динамічна, об'єктно-орієнтованамова програмування.

Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати

структуру та зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні сервера (подібно до таких мов програмування, як Java, так і C#), розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо. JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

JavaScript (JS) − динамічна, об’єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки [2].

JavaScript класифікують як прототипну (підмножина об’єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, рограммного керування пам’яттю, прототипне наслідування, функції як об’єкти першого рогра.

При розробці великих і нетривіальних веб-застосунків з використанням JavaScript, критично важливим є доступ до інструментів зневадження. Оскільки браузери від різних виробників дещо відрізняються у поведінці JavaScript і реалізації Об'єктної Моделі Документа, треба мати в руках зневаджувач для кожного браузера, якщо веб-застосування орієнтовано на нього.

На даний час Internet Explorer, Firefox, Opera, Google Chrome та Safari мають зневаджувачі для себе.

Internet Explorer має три зневаджувача для себе: Microsoft Visual Studio є найпотужнішим з цих трьох, слідом йде Microsoft Script Editor (компонента Microsoft Office), і нарешті існує безкоштовний Microsoft Script Debugger з базовими функціями. Веб-застосування для Firefox допоможе привести до розуму додаток Firebug (зручно вбудований безпосередньо в браузер), або давніший відладчик Venkman, котрий також працює з браузером Mozilla. Drosera — це зневаджувач з WebKit engine[15], що супроводжує Apple Safari.

Також існують кілька інструментів, як вільних, наприклад JSLint[16], інструмент перевірки якості коду, що сканує JavaScript програму, шукаючи проблеми коду, так і комерційних продуктів типу інструменту з назвою JavaScript Debugger.

Оскільки JavaScript є інтерпретатором, без строгої типізації, і може виконуватися в різних середовищах, кожне зі своїми власними особливостями сумісності, програміст має бути дуже уважним, і повинен перевіряти, що його код виконується як очікується в широкому переліку можливих конфігурацій. Дуже часто трапляються випадки, коли скрипт, що чудово працює в одному середовищі, видає некоректні результати в іншому.

Кожен блок сценарію інтерпретатор розбирає окремо. На веб-сторінках, коли треба комбінувати блоки JavaScript та HTML, синтаксичні помилки знайти легше, якщо тримати функції сценарію в окремому блоці коду, або (ще краще) використовувати багато малих пов'язаних .js файлів. В такий спосіб синтаксична помилка не спричинятиме «падіння» цілої сторінки, і можна надати допомогу, елегантно вийшовши зі сторінки.

4. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ВИЗНАЧЕННЯ АКТИВНОСТІ УЧАСНИКІВ КОЛЕКТИВНОЇ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ

Програма написана на мові PHP з застосуванням Laravel Framework – популярних засобів розробки для додатків серверного типу.

Схема бази даних зберігається в *.sql файлі, який може бути відкритий програмою. Схема містить інформацію про сутності БД та версію БД.

Інструменти розробки, а в особливості фреймворк Laravel має ряд переваг для додатку такого типу. Laravel містить в собі реалізацію шаблону проектування MVC, а також шаблону проектування БД ActiveRecord. Міграції є зручним інструментом для організації системи контролю версій для БД. Також фреймворк містить модульне тестування і систему пакетів, завдяки якій можна створювати та підключати свої або чиїсь пакети з допомогою Composer.

Оскільки для ефективної роботи в системі лише одного типу користувачів недостатньо, в програмі передбачено використання 3 типів користувачів. Кожен тип користувачів має свої привілегії.

Загальна схема роботи виглядає як показано на рисунку 4.1.

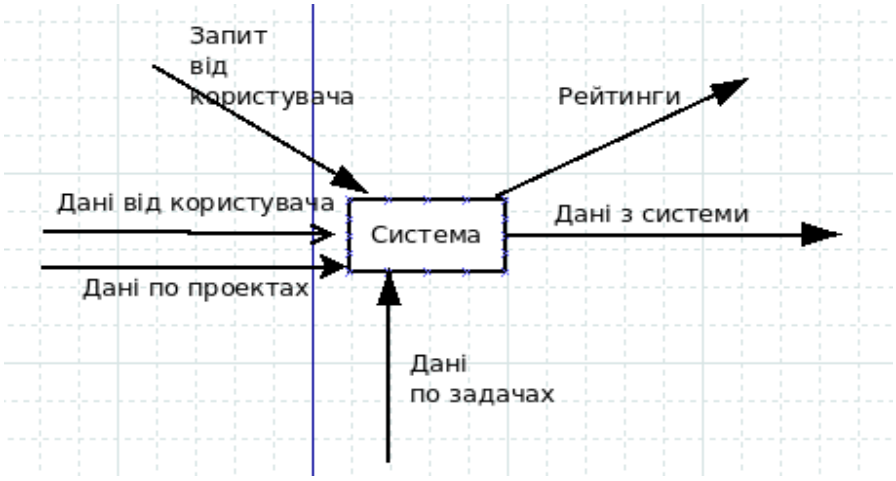

Рисунок 4.1 – Загальна схема роботи додатку

Система включає три ролі користувача, кожна з ролей має свої привілегії в системі. Адміністратор системи має найвищі привілегії в системі, саме адміністратор додає нових користувачів до системи відсилаючи з системи запрошення на пошту майбутнього користувача з посиланням для продовження реєстрації, якщо користувач погоджується, то перейшовши за посиланням завершує реєстрацію, створивши свій пароль, якщо ні то посилання стає неактивним через два дні після його створення. Адміністратор може призначати ролі користувачам, створювати проекти, а також переглядати інформацію по проектах та задачах, переглядати рейтинг користувачів по проектах, редагувати інформацію по проектах. Діаграма прецедентів адміністратора зображена на рисунку 4.2

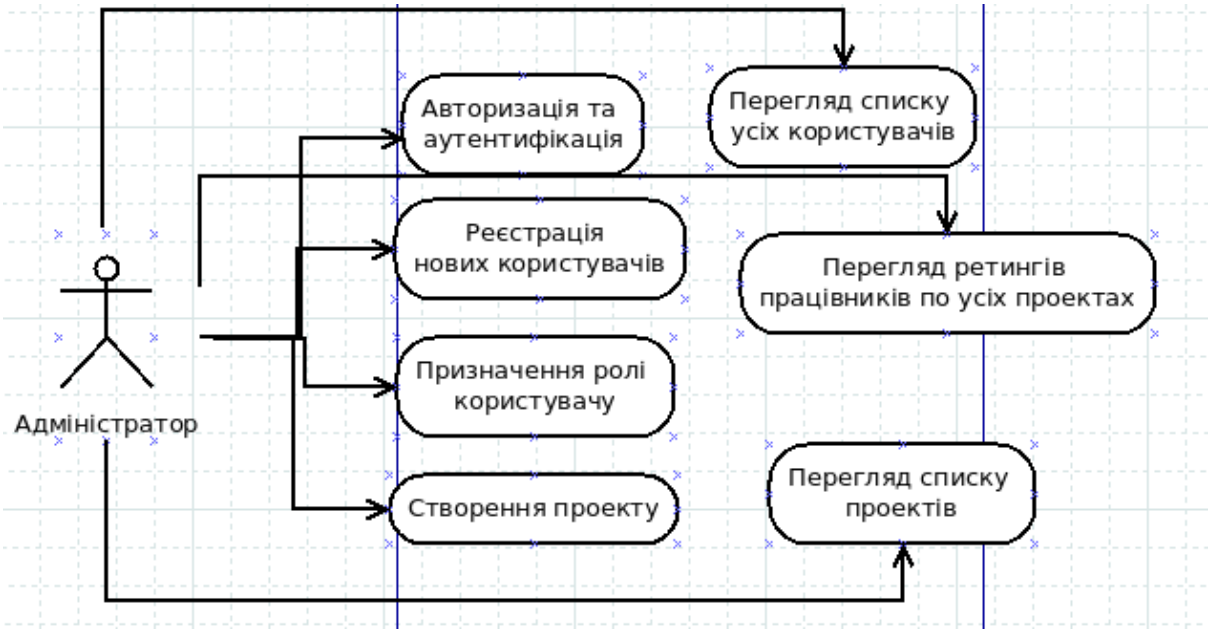

Рисунок 4.2. Діаграма прецедентів адміністратора

Роль проектного менеджера включає в себе повний контроль над процессами в проекті, на якому він головний. А саме проектний менеджер веде свій проект, може редагувати його, створювати, редагувати, та видаляти задачі в ньому. Також проектний менеджер розподіляє задачі між працівниками. В особистому кабінеті має свою вкладку “Оцінити виконані задачі”. На цій сторінці виводиться список

задач, які були переведені в статус “Виконано”, ут проектний менеджер вирішує приймати задачу, чи відхиляти, якщо задача приймається, то її треба оцінити від одного до пяти. Має можливість переглядати рейтинг працівників в проекті.

Звичайний працівник має можливість переглядати задачі, які були призначені на нього, змінювати статуси свої задач та редагувати інформацію про себе. Діаграма прецедентів проектного менеджера та користувача зображені на рисунку 4.3.

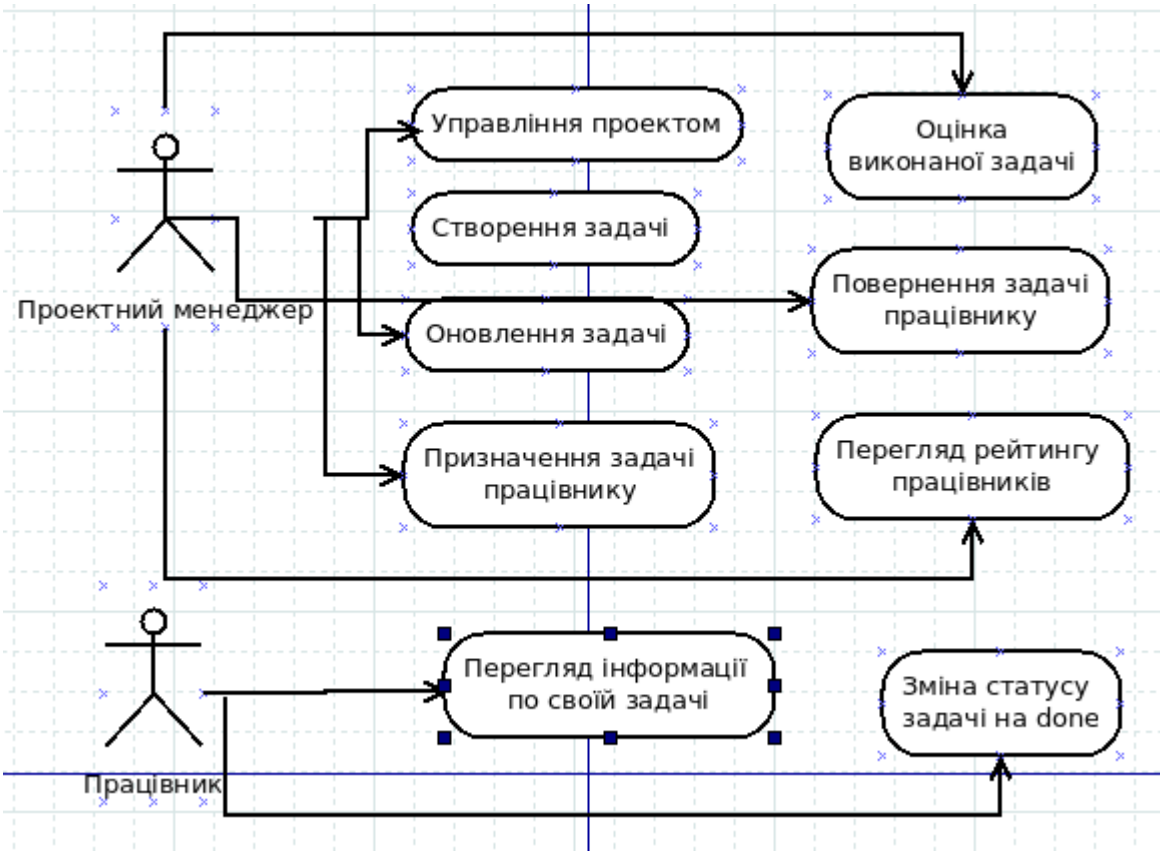

Рисунок 4.3. Діаграма прецедентів працівника і проектного менеджера

В програмі передбачено декілька критеріїв задачі , такі як тип, важкість, важливість, час на виконання, кількість повернень задачі проектним менеджером, оцінка задачі проектним менеджером. Задачі можуть бути двох типів, такі як bug і history, важкість задачі може бути від одного до пяти, так само і важливість. Оцінка проектного менеджера може бути від одного до пяти, а максимальна кількість повернень дорівнює десяти. На рисунку 4.4 зображено діаграму послідовностей створення задачі.

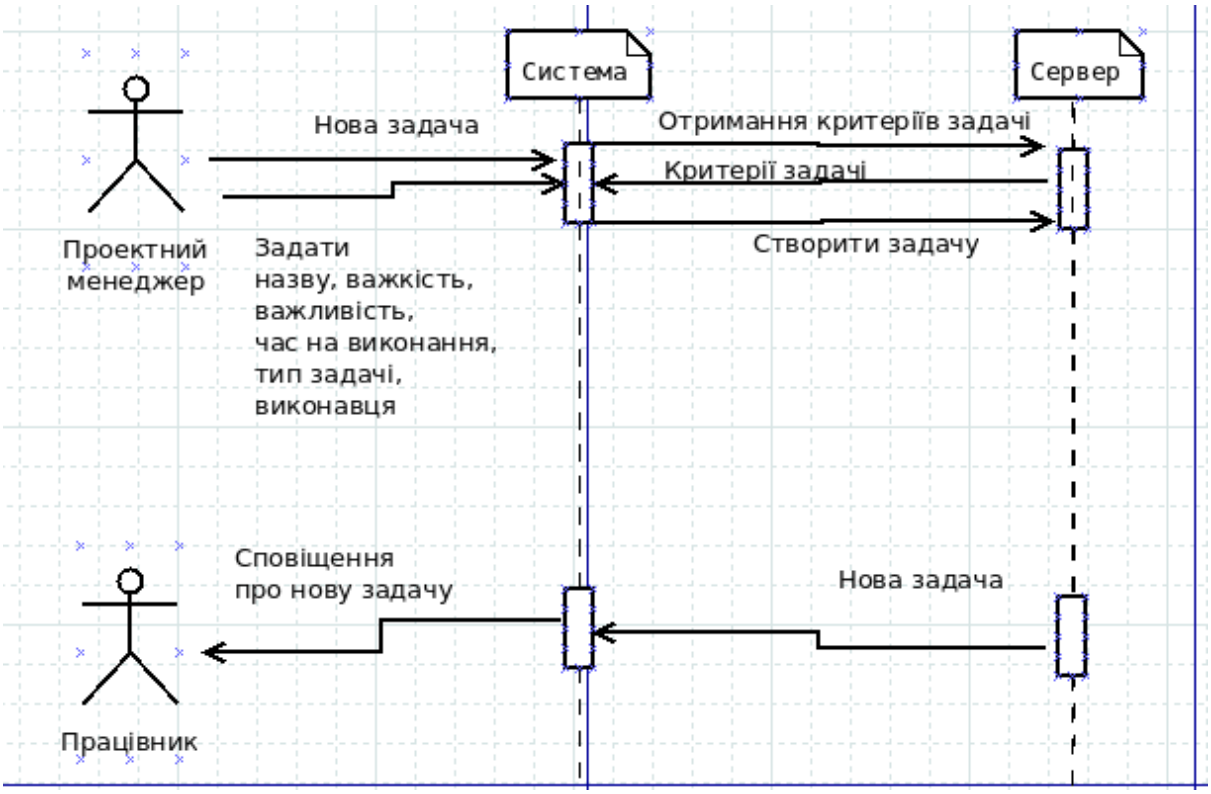

Рисунок 4.4. Діаграма послідовностей створення задачі

Програмна реалізація поділена на групи класів, класифіковані за категоріями:

- Controllers – загальні контролери;
- Migrations – класи для міграцій в базу даних;
- Seeds – класи, для сидінгу даних в базу;
- Models – класи, для роботи з сутністю з бази даних;
- Templates – класи, які відповідають за роботу user interface.

Для системи управління версіями бази даних організований механізм міграцій. Дозволяє зв'язувати зміни в коді програми зі змінами, які потрібно внести в структуру БД, що спрощує розгортання і оновлення програми. У ході виконання роботи було написано 10 міграцій, які дозволять розгорнути проект з нуля з потрібною структурою БД. Було написано наступні міграції:

- 2018_06_10_095710_create_roles_table - ролі користувачів;
- 2018_06_10_095710_create_issue_types_table - типи задач;
- 2018_06_10_095928_create_user_roles_table - ролі користувачів;
- 2018_06_11_094916_create_projects_table - таблиця проектів;

– 2018_06_19_145526_create_issue_statuses_table.php - статуси задач;

– 2018_06_19_152225_create_issues_table - таблиця задач;

– 2018_06_23_123841_create_permissions_table - дозволи користувача в системі;

– 2018_06_23_125455_create_role_permissions_table - таблиця відповідності дозволів ролям;

– 2018_07_13_170557_add_pm_scales_to_issues_table - додати оцінку проектного менеджера до задачі.

Структура БД побудована після запуску міграцій рисунок 4.5.

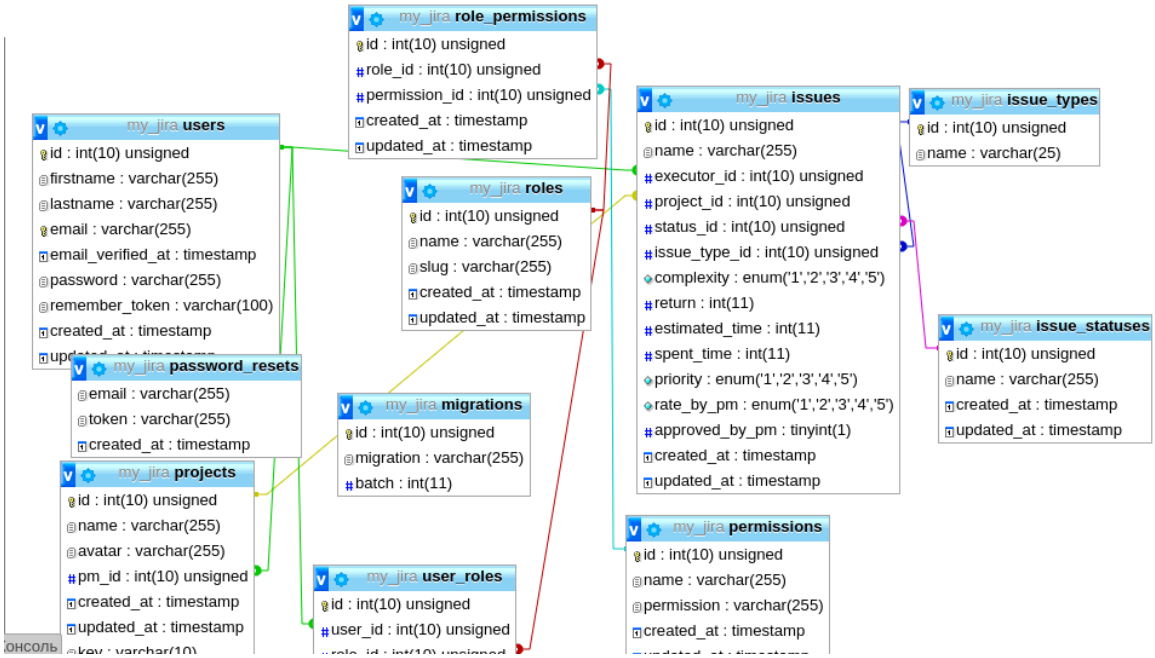

Рисунок 4.5. - структура БД

Для заповнення бази початковими даними було написано сіди для статусів - IssueStatusesTableSeeder та типів задач - IssueTypesTableSeeder, ролей - RolesTableSeeder ,дозволів в системі - PermissionsTableSeeder, дозволів розподілених по ролях - RolePermissionsTableSeeder.

Для всього функціоналу написано 5 контроллерів. Діаграма класів контроллерів системи показана на рисунку 4.6.

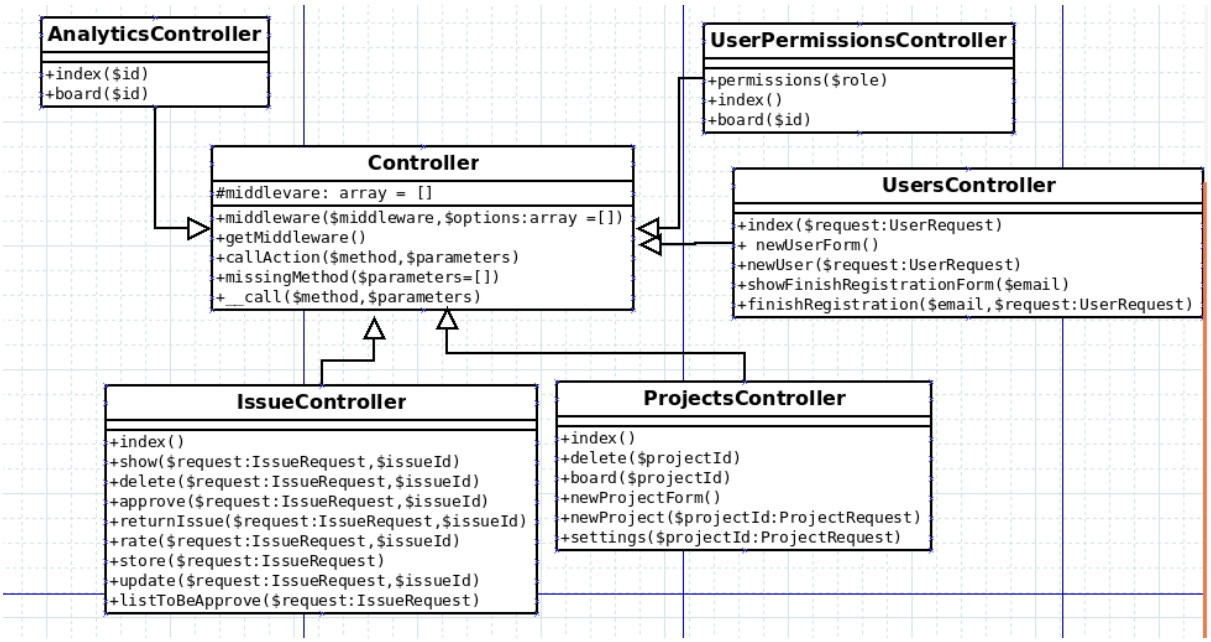

Рисунок 4.6 діаграма класів контроллерів системи

Controller - базовий клас контроллер.

UsersController - контроллер для роботи з користувачами, додавання, редагування, видалення, завершення реєстрації користувача.

ProjectsController - контроллер для роботи з проектами, створення, редагування, видалення, налаштування, виведення дошки з задачами по проекту.

UserPermissionsController - контроллер для управління пермішнами в системі, створення нових, видалення, редагування, призначенння користувача чи ролі в системі.

AnalyticsController - контроллер для аналітики в системі, прорахунку рейтингу користувачів.

IssuesController - контроллер для управління задачами. Вивести список, вивести конкретну задачу, додати нову, відредагувати існуючу, видалити задачу, підтвердити виконання задачі або повернути її - для ПМ, оцінити виконану задачу, повернути список задач для оцінки ПМ.

Для роботи з базою даних використовуються класи пронаслідувані від базового Model, який реалізує шаблон ActiveRecord, а саме:

- IssueStatus – статуси задач.

- IssueType – типи задач.
- Permission – дозволи в системі.
- Role – ролі в системі.
- RolePermission – дозволи по ролях.
- Issue – задачі.
- User – користувачі.
- UserRole – ролі користувачів.

Система передбачає побудову рейтингу працівників у проекті, для розрахунку рейтингу було виведено формулу (рисунок 5.7)

$$(\sum_{c} \frac{0.5 * G + 0.7(\frac{D}{T}) * \frac{S}{5}}{0.6 * R})/c$$

Рисунок 5.7. - Формула для розрахунку рейтингу працівників

Дана формула використовує дані про задачу, які зберігаються в системі, де

- c – кількість виконаних задач;
- G – оцінка ПМ;
- D – важкість задачі;
- T – час на виконання;
- S – важливість;
- R – повернення на доопрацювання.

Рейтинг рахується як середнє арифметичне по всіх задачах, що виконав працівник. Дана формула повертає число 10 при найуспішнішому виконанню задачі по параметрах і число близьке до нуля при найгіршому варіанті. Для розрахунку рейтингу береться середнє арифметичне по всіх задачах.

Висновки до розділу 4

Розроблений програмний додаток виконано відповідно заданих вимог та з використанням патерну MVC та сучасних підходів до розробки програмних застосунків.

5. ІНСТРУКЦІЯ РОБОТИ ПРОГРАМИ

Після запуску програми, користувач може увійти в обліковий запис - рисунок 5.1. Для того щоб зайти в систему користувачу необхідно ввести свої дані(логін та пароль), якщо користувач забув пароль, то його можна відновити, натиснувши “Forgot your password?”

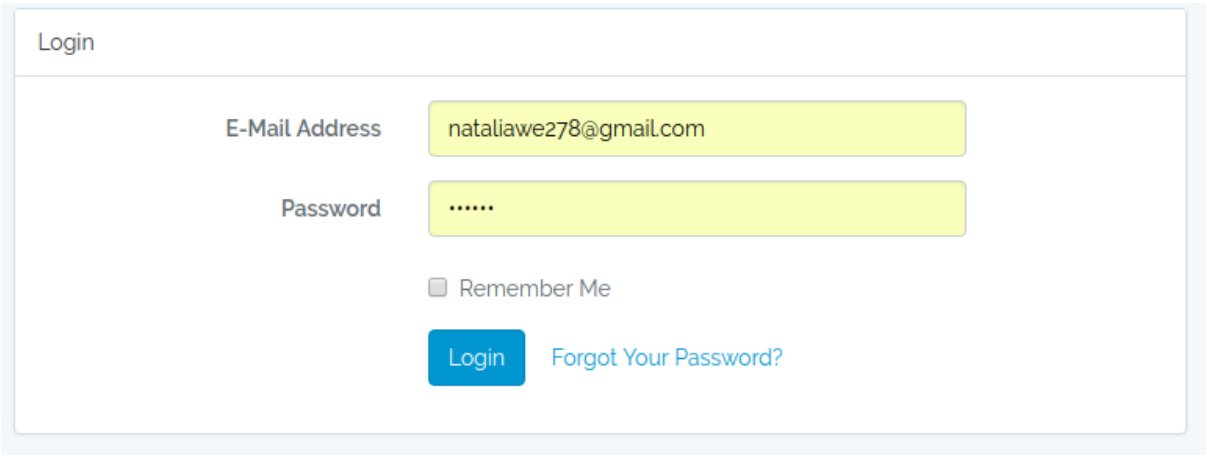

Рисунок 5.1 - форма для входу

Після аутентифікації користувач може виконувати ряд дій дозволених його роллю, так адміністратор може переглядати списки користувачів, перейшовши на вкладку “Користувачі” разом з їх ролями та проектами а також створювати нових. Для створення нового користувача необхідно натиснути кнопку “Новий користувач”, після чого з’явиться вікно у якому треба буде ввести емейл користувача та задати його роль. Після цього система створить посилання для завершення реєстрації цього користувача і відправить повідомлення на пошту з цим посиланням. Для завершення реєстрації користувач матиме 2 дні, якщо за 2 дні реєстрація не завершиться, то користувач видалиться з системи. Для завершення реєстрації користувач повинен двічі ввести пароль і заповнити особисті дані. Перейшовши на вкладку проекти, адміністратор може бачити список активних

проектів разом з їх проектними менеджерами (рисунок 5.2).

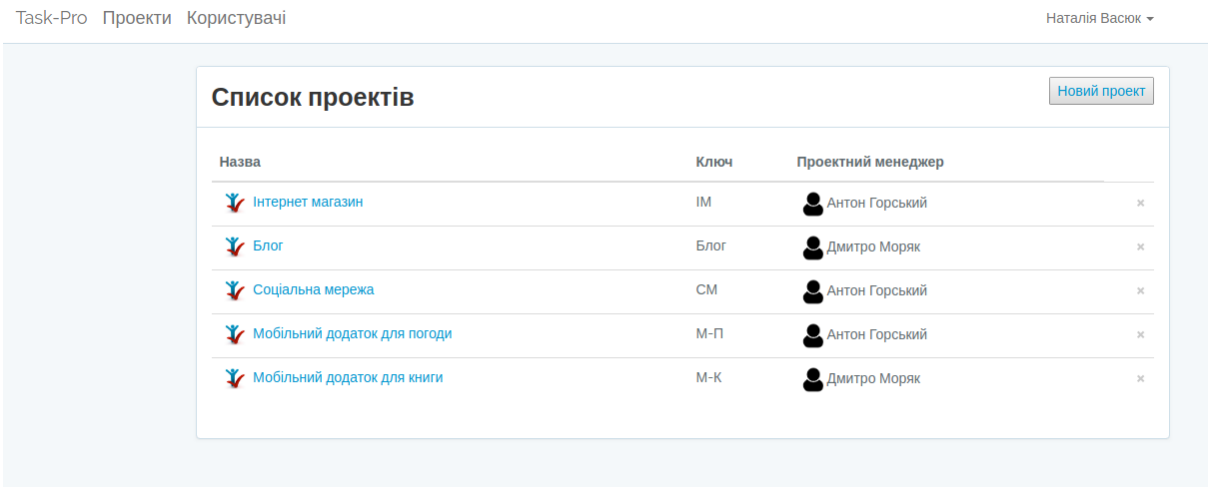

Рисунок 5.2. - Список проектів

Натиснувши на кнопку “Новий проект” адміністратор може створити новий проект. Для створення нового проекту, адміністратору необхідно ввести його ім’я, ключ, завантажити аватар та призначити проектного менеджера (рисунок 5.3).

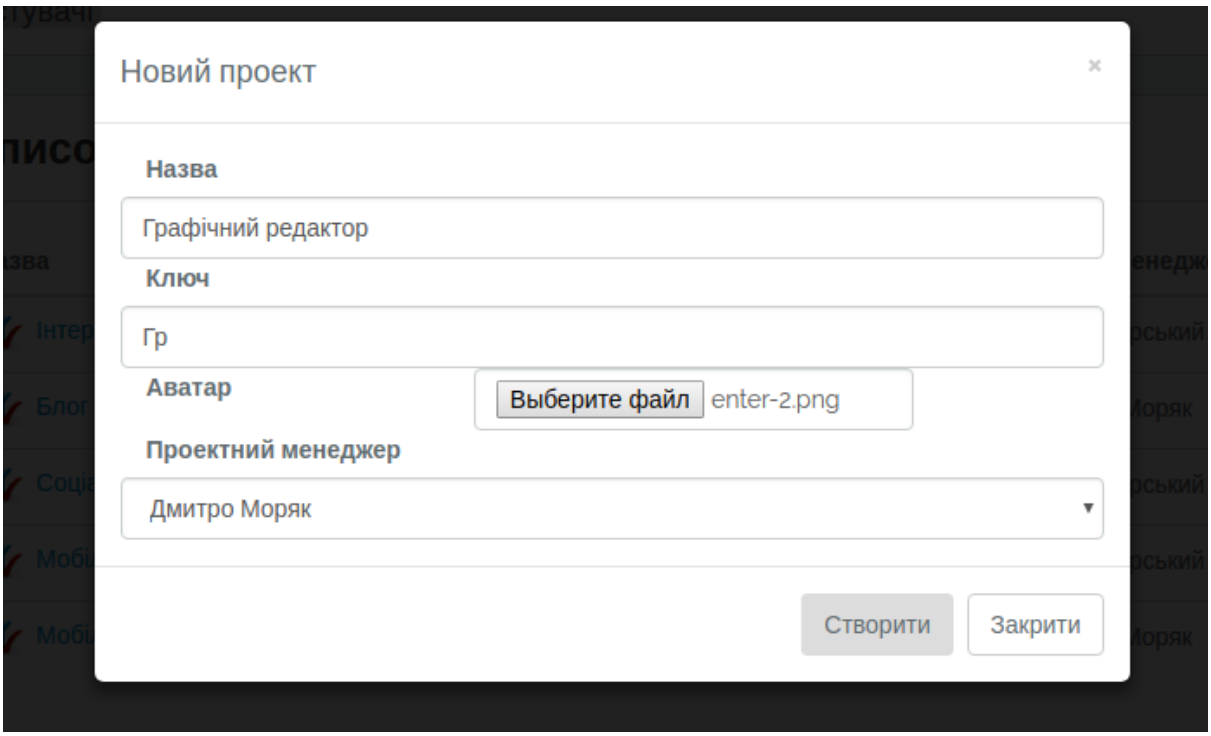

Рисунок 5.3 - Створення нового проекту

Після цього проект з’явиться у списку. Для перегляду дошки користувач має натиснути на назву проекту в списку проектів. Після цього з’явиться дошка з

задачами, адміністратор зможе переглядати усі задачі по всім проектам, редагувати інформацію по проектах вибравши пункт меню “Налаштування”. Адміністратор може відредагувати інформацію по будь-якому проекту, а також змінити програмного менеджера. Також для адміністратора є можливість додавання задач в проекти видалення, та редагування. Для того щоб створити нову задачу необхідно натиснути на кнопку в будь-якій з трьох секцій(“Зробити”, “В процесі”, “Зроблено”). Після натиснення на кнопку з’явиться форма (рисунок 5.4) з полями, кожне з яких є критерієм задачі.

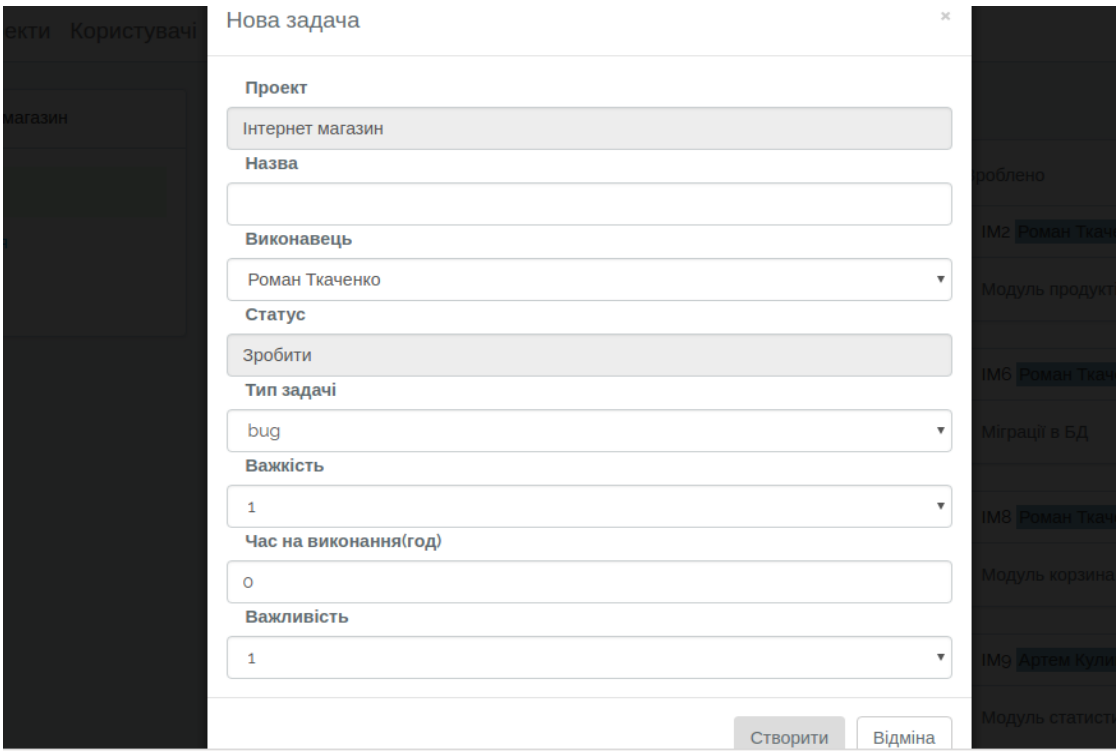

Рисунок 5.4. - Створення задачі

Форма містить 8 полів:

- проект - в якому сворюється задача(заповнюється автоматично);
- назва задачі;
- виконавець задачі;
- статус задачі - один із статусів передбачених в системі(заповнюється автоматично);
- тип задачі;

- важкість - величина, яка характеризує важкість задачі, значення від одного до п’яти;
- час на виконання задачі в годинах;
- важливість задачі - значення від одного.

Також адміністратор має можливість переглядати рейтинг усіх співробітників по всіх проектах. Для перегляду рейтингу нелбхідно вибрати пункт меню “Аналітика”.

Ще одна не менш важлива роль в системі - проектний менеджер. Має можливість переглядати співробітників свого проекту, створювати та редагувати задачу в своєму проекті. Проектний менеджер має повну владу над задачами в своєму проекті. Після того як співробітник переводить задачу в статус “Зроблено” її має оцінити проектний менеджер. Для того щоб оцінити задачу, проектний менеджер має авторизуватися в системі згори у проектного менеджера є додаткова вкладка “Заапрувити”, перейшовши на цю вкладку буде показано список задач, які були переведені в статус “Зроблено”, але ще не підтверджені. Клацнувши по одній з задач відкриється модальне вікно з інформацією по задачі(рисунок 5.5), проектний менеджер має вирішити прийняти чи відхилити задачу. У разі прийняття задачі її необхідно оцінити значенням від одного до п’яти.

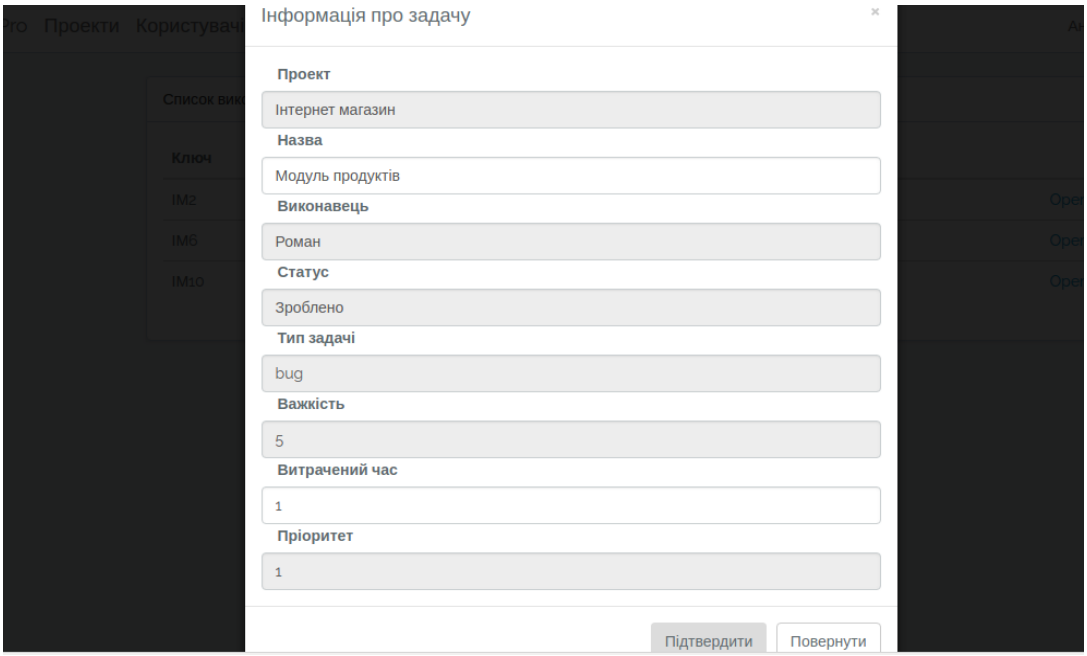

Рисунок 5.5. - модальне вікно виконаної задачі

У випадку якщо задача була відхилена, то вона повертається на опрацювання, а лічильник повернень задачі збільшується.

Користувач має можливість авторизуватися в системі, відредагувати інформацію про себе. Переглядати свої задачі та змінювати їх статуси.

6. СТАРТАП ПРОЕКТ

Розділ має на меті проведення маркетингового аналізу стартап проекту задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження. Проведення маркетингового аналізу передбачає виконання нижченаведених кроків.

6.1 Опис ідеї проекту

В межах підпункту слід проаналізувати та подати у вигляді таблиць:

- зміст ідеї (що пропонується);
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару;
- чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подаються у вигляді таблиці (таблиця 6.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки.

Таблиця 6.1. Опис ідеї стартап-проекту

Зміст ідеї	**Напрямки застосування**	**Вигоди для користувача**
Інструментальні засоби визначення активності учасників колективної розробки програмної системи	1. ІТ компанія	1. Пришвидшення розробки завдяки ефективному розподіленню задач

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик

ідеї;

− визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

− проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (таблиця 6.2). [58]

Таблиця 6.2. Визначення сильних, слабких та нейтральних характеристик

No п/п		**(потенційні) товари/концепції конкурентів**	
		Мій проект	**Jira**
1	W слабка сторона	Відсутність офлайн версії	Відсутність офлайн версії
2		Відсутність можливості створення плагінів	Платне використання
3	N нейтральна сторона	Можливість керування задачами та проектами	Можливість керування задачами та проектами
4	S сильна сторона	Можливість перегляду коефіцієнтів ефективності на основі розрахунків за унікальною формулою	Можливість створення плагінів

6.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту. Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 6.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту;
- чи існують такі технології, чи їх потрібно розробити/доробити;
- чи доступні такі технології авторам проекту.

Таблиця 6.3. Технологічна здійсненність ідеї проекту

No п/п	**Ідея проекту**	**Технології її реалізації**	**Наявність технологій**	**Доступність технологій**
1	Нативний інтерфейс користувача	macOS Laravel Framework	Наявна	Доступна безкоштовно
2	Веб-інтерфейс користувача	HTML + Javascript	Наявна	Доступна безкоштовно
3	Крос-платформений інтерфейс користувача	Web browsers	Наявна	Доступна безкоштовно
Висновок: проект реалізувати можливо. Обрана технологія реалізації ідеї проекту: Нативний інтерфейс користувача, Генерація коду на основі заданих шаблонів mustache				

За результатами аналізу таблиці робиться висновок щодо можливості технологічної реалізації проекту: так чи ні, а також технологічного шляху, яким це доцільно зробити (з поміж названих технологій обираються такі, що доступні авторам проекту та є наявними на ринку).

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 6.4).

Таблиця 6.4. Попередня характеристика потенційного ринку стартап-проекту

No п/п	**Показники стану ринку (найменування)**	**Характеристика**
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	270 грн
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обме- жень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	50 %

Середня норма рентабельності в галузі (або по ринку) порівнюється із банківським відсотком на вкладення. За умови, що останній є вищим, можливо, має сенс вкласти кошти в інший проект.

За результатами аналізу таблиці робиться висновок щодо того, чи є ринок привабливим для входження за попереднім оцінюванням.

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (таблиця 6.5).

Таблиця 6.5. Характеристика потенційних клієнтів стартап-проекту

No п/п	**Потреба, що формує ринок**	**Цільова аудиторія (цільові сегменти ринку)**	**Відмінності у поведінці різних потенційних цільових груп клієнтів**	**Вимоги споживачів до товару**
1	Інструментальні засоби визначення активності учасників колективної розробки програмної системи	Компанії та стартапери, що розробляю ть ПЗ	Безкоштовне користування продуктом	Безкоштовна Наявність документації Підтримка необхідних платформ Генерація оптимізованого коду

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиці 5.6-5.7).

Таблиця 6.6. Фактори загроз

No п/п	**Фактор**	**Зміст загрози**	**Можлива реакція компанії**
1	Підходить для нових проектів	Для існуючих проектів виникне потреба переносити базу даних. Продукт більше підходить для нових проектів	Додавання можливості автоматизованого імпорту з різних типів сховищ
2	Власний формат зберіганн я	При необхідності змінити інструмент, компанії доведеться це робити вручну, оскільки використовується власна структура збереження даних	Додавання можливості автоматизованого експорту в різні типи сховищ
3	Обмеженіс ть функцій	Інструмент обмежений наявними функціями і не має деяких функцій, які мають конкуренти (наприклад: індекси)	Додавання нових функцій за потреби

Таблиця 6.7. Фактори можливостей

No п/п	**Фактор**	**Зміст можливості**	**Можлива реакція компанії**
1	Популярність веб застосунків	Веб індустрія зараз перша за розвитком	Вихід на вебринок
2	Потреба у використанні баз даних	SQL бази даних використовуються у більшості серверних програм	Надання інструменту для генерування нативного коду для роботи з БД
3	Відсутність повноцінних альтернатив	Існуючі альтернативи не надають можливості генерувати код для декількох платформ	Розширена підтримка існуючих платформ

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку. Аналіз пропозиції необхідно виконати аналізуючи існуючі види конкуренцій.

Пропозиції повинні відповідати на питання “Як просувати продукт”.

Аналіз пропозицій зображено на таблиці 5.8.

Таблиця 6.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	**В чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
1. Вказати тип конкуренції - монополія/олігополія/ монополістична/чиста	чиста	Прямі договори з стартапами, презентація продукту на виставках
2. За рівнем конкурентної боротьби - локальний / національний / ...	національний	Публікація статей на міжнародних сайтах

Таблиця 6.8. (продовження)

3. За галузевою ознакою - міжгалузева/ внутрішньогалузева	внутрішньогалузева	Розвивати напрямки, нерозвинуті конкурентами
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	товарно-видова	Розповідати про свої переваги перед конкурентом у цій галузі
5. За характером конкурентних переваг - цінова / нецінова	нецінова	Надання функцій, які не надають конкуренти
6. За інтенсивністю - марочна/не марочна	марочна	Надання функцій, які не надають конкуренти

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (таблиця 5.9). [59]

Таблиця 6.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	**Прямі конкуренти в галузі**	**Потенційні конкуренти**	**Постачальники (Архітектори БД)**	**Клієнти (Розробники)**	**Товари-замінники**
	- Jira - Asana	Проект з підтримкою офлайн	Мінімізація витрат часу постачальників	Контроль якості	Лояльність споживачів
Висновки:	Визначити інтенсивність конкурентної боротьби з боку прямих конкурентів	Є можливості виходу на ринок, оскільки існуючі рішення не надають потрібних переваг	Постачальники підлаштовуються під ринок	Клієнти диктують вимоги згідно з умовами експлуатації	Обмеження для роботи на ринку через товари замінники

На основі аналізу конкуренції, проведеного в п. 3.5 (таблиця 6.9), а також із урахуванням характеристик ідеї проекту (таблиця 6.2), вимог споживачів до товару

(таблиця 6.5) та факторів маркетингового середовища (таблиця 6.6-6.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за таблицею 10. [60]

Таблиця 6.10. Обґрунтування факторів конкурентоспроможності

No п/п	**Фактор конкурентоспроможності**	**Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)**
1	Наявність унікального алгоритму розрахунку коефіцієнту важливості працівника	Існуючі конкуренти або не надають такого функціоналу і його потрібно додатково купувати, або є платними
2	Безкоштовність для функціоналу керування задачами та проектами	Існуючі конкуренти надають платне ПЗ.

За визначеними факторами конкурентоспроможності (таблиця 6.10) проводиться аналіз сильних та слабких сторін стартап-проекту (таблиця 6.11). [61]

Таблиця 6.11. Порівняльний аналіз сильних та слабких сторін

No п/п	**Фактор конкурентоспроможності**	**Бали 1-20**	**Рейтинг товарів-конкурентів у порівнянні з Database Generator (даним продуктом)**						
			-3	**-2**	**-1**	**0**	**1**	**2**	**3**
1	Підтримка декількох платформ	20	+						
2	Генерація додаткових зручних методів	10			+				

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 6.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця

6.11). [62]

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 6.12. SWOT-аналіз стартап-проекту

Сильні сторони: Алгоритм для розрахунку важливості працівника	**Слабкі сторони:** Відсутність дизайнерського інтерфейсу
Можливості: Популярність мобільних платформ Потреба у використанні баз даних Відсутність повноцінних альтернатив	**Загрози:** Підходить для нових проектів Власний формат зберігання Обмеженість функцій

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. [63]

Таблиця 6.13. Альтернативи ринкового впровадження стартап-проекту

No п/п	**Альтернатива (орієнтовний комплекс заходів) ринкової поведінки**	**Ймовірність отримання ресурсів**	**Строки реалізації**
1	Орієнтація поточної моделі на ринок стартаперів	50 %	40 год
2	Орієнтація поточної моделі на ринок державних установ	20 %	160 год

Таблиця 6.13. (продовження)

3	Орієнтація поточної моделі на ринок ентерпрайз	10 %	200 год
4	Переорієнтація на генерацію серверної частини	80 %	120 год
5	Переорієнтація на веб-розробку	35 %	80 год
6	Переорієнтація на перенесення БД з одних платформ на інші	60 %	160 год

Альтернатива, де отримання ресурсів є більш простим та ймовірним – №4 "Переорієнтація на генерацію серверної частини", що становить 80 відсотків. Це значення перевищує інші альтернативи.

Альтернатива, де строки реалізації є більш стислими – №1 " Орієнтація поточної моделі на ринок стартаперів". Терміни реалізації в цьому разі становлять лише 40 годин.

6.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 6.14).

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку. [64]

Таблиця 6.14. Вибір цільових груп потенційних споживачів

No п/п	**Опис профілю цільової групи потенційних клієнтів**	**Готовність споживачів сприйняти продукт**	**Орієнтовний попит в межах цільової групи (сегменту)**	**Інтенсивність конкуренції в сегменті**	**Простота входу у сегмент**
1	Стартапери	Готові	Високий	Висока	Просто
2	Державні установи	Потребують недовгих переговорів	Середній	Середня	Складно
3	Ентерпрайз	Потребують довгих переговорів	Низький	Низька	Дуже складно
Які цільові групи обрано: стартапери					

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (таблиця 6.15).

Таблиця 6.15. Визначення базової стратегії розвитку

No п/п	**Обрана альтернатива розвитку проекту**	**Стратегія охоплення ринку**	**Ключові конкурентоспроможні позиції відповідно до обраної альтернативи**	**Базова стратегія розвитку***
1	Орієнтація поточної моделі на ринок стартаперів	Стратегія концентрованого маркетингу	Стартапери потребують швидкості розробки, яку надає підтримка декількох платформ даним продуктом	Стратегія спеціалізації (спирається на диференціацію)

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Перелік ринкових загроз та ринкових можливостей складається на основі

аналізу факторів загроз та факторів можливостей маркетингового середовища. Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту.

Наступним кроком є вибір стратегії конкурентної поведінки (таблиця 6.16).

Таблиця 6.16. Визначення базової стратегії конкурентної поведінки

No п/п	**Чи є проект «першопрохідцем» на ринку?**	**Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?**	**Чи буде компанія копіювати основні характеристики товару конкурента, і які?**	**Стратегія конкурентної поведінки***
1	Ні	шукати нових споживачів	Так, розділення на генерований код та код вільний до редагування	Стратегія заняття конкурентної ніші

З обраних сегментів до постачальника (стартап-компанії) та до продукту розробляється стратегія позиціонування (таблиця 6.17). що полягає у формуванні ринкової позиції, за яким споживачі мають ідентифікувати торгівельну марку/проект. [65]

Таблиця 6.17. Визначення стратегії позиціонування

No п/п	**Вимоги до товару цільової аудиторії**	**Базова стратегія розвитку**	**Ключові конкуренто-спроможні позиції власного стартап-проекту**	**Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)**
1	Стабільність роботи Безкоштовність Наявність документації Підтримка необхідних платформ	Стратегія спеціалізації (спирається на диференціацію)	Стартапери потребують швидкості розробки, яку надає підтримка декількох платформ даним продуктом	пришвидшення розробки ПЗ підтримка декількох платформ графічний редактор схеми БД

6.5 Розроблення маркетингової програми стартап-проекту

Для цього у таблиці 6.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару. [66]

Таблиця 6.18. Визначення ключових переваг концепції потенційного товару

No п/п	**Потреба**	**Вигода, яку пропонує товар**	**Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)**
1	Пришвидшення розробки ПЗ	Підтримка декількох платформ	Більшість конкуренти підтримують лише одну платформу (окрім Realm)
2	Використання нативного коду	Генерує нативний код	Конкурент Realm генерує код для роботи з власною БД, не нативною

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 6.19).

Таблиця 6.19. Опис трьох рівнів моделі товару

Рівні товару	**Сутність та складові**		
I. Товар за задумом	Генератор нативного модельного коду для мобільних платформ на основі заданої схеми БД		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	можливість оптимізації витрат часу	М	Тл
	можливість оптимізації витрат коштів	М	Вр

Таблиця 6.19. (продовження)

	відповідність актуальним технологіям	Нм	Тх
	Відповідає вимогам ДСТУ ISO/IEC 25030:2015 Програмна інженерія. Вимоги щодо якості та оцінювання програмного продукту (SQuaRE). Вимоги щодо якості		
	Пакування: готовий до використання dmg пакет		
	Марка: Database Generator		
III. Товар із підкрі- пленням	Потенційний користувач може ознайомитись з поточним товаром з наукових конференцій та публічних виступів, а також наукових вісників на яких була представлена інформація про даний продукт		
За рахунок чого потенційний товар буде захищено від копіювання: Назва і контент захищені ліцензією MIT; захист інтелектуальної власності			

М/Нм – монотонні або немонотонні;

Вр/Тх/Тл/Е/Ор – вартісні, технічні, технологічні, ергономічні або органолептичні (останній – для продуктів харчування)

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання. Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу-хау, чи комплексне поєднання властивостей і характеристик, закладене на другому та третьому рівнях товару. [67]

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (таблиця 6.20).

Таблиця 6.20. Визначення меж встановлення ціни

No п/п	**Рівень цін на товари-замінники**	**Рівень цін на товари-аналоги**	**Рівень доходів цільової групи споживачів**	**Верхня та нижня межі встановлення ціни на товар/послугу**
1	27…270 грн	135…270 грн	27000…98000 грн	27…135 грн

Висновки до розділу 6

Розроблений програмний продукт має переваги над існуючими конкурентами та є конкурентноздатним на ринку. Програма має шляхи подальшого розвитку, визначені маркетингові стратегії та шляхи збуту. Основна цільова аудиторія – це ІТ компанії, для яких важлива швидкість та ефективність розробки ПЗ.

ВИСНОВКИ

Задача інструментальних засобів визначення активності учасників колективної розробки полянає у визначенні можливих способів реалізції системи для визначення ефективності учасників колективної розробки. В якості вхідних параметрів даються дані щодо проектів та задач, які заповнюються проектним мнеджером. Дані по задачах аналізуються програмою і на основіх цих даних будується рейтинг працівників.

Наукова новизна розробленого прогамного додатку полягає в тому, що абсолютно бесплатний продукт даватиме можливість ефективно керувати проектами та задачами, а також організувати найбільш продуктивну роботу за рахунок призначення найбільш ефективних працівників на найважливіші задачі, а рейтинг працівників будуватиметься в системі, базуючись на даних по раніше виконаних задачах.

Практичне значення одержаних результатів роботи полягає в розробці програмного продукту, який полегшить розробку продуктів, працюючи в команді.

База даних підтримує велику кількість різноманітних функцій. З усіх потужностей бази даних обрано базові можливості: сутності, атрибути та зв'язки. Наявність чи відсутність ключів визначає шаблон, оскільки для iOS ця функціональність не потрібна.

Генерований код використовує мову PHP за основу для роботи системи.Використовується досить потужний та популярний PHP фреймворк Laravel.

Для зберігання данних обрано реляційну базу даних з системою управління базою даних MYSQL.

Для фронтенду обрано фреймворк bootstrap, для динамічності мову javascript, а саме фреймворк jquery

Розроблений програмний додаток виконано відповідно заданих вимог та з використанням патерну MVC та сучасних підходів до розробки програмних застосунків.

Для користування даним продуктом необхідий встановлений браузер і вихід в інтеренет. Так як йде звернення до віддаленого сервера.

Розроблений програмний інтерфейс дозволяє виконати всі задані вимоги, а саме створити користувачів, призначити їм відповідні ролі, Корувати інформацією по користувачах, створювати, редагувати та видаляти проекти, призначати проектного менеджера для проекта. Створювати, редагувати та видаляти задачі. Також система будує рейтинг працівників, який потім може аналізувати проектний менеджер і керуватися ним при призначенні задач.

Розроблений програмний продукт має переваги над існуючими конкурентами та є конкурентноздатним на ринку. Програма має шляхи подальшого розвитку, визначені маркетингові стратегії та шляхи збуту. Основна цільова аудиторія – це стартапери, для яких важлива швидкість розробки, та в яких немає достатньо коштів для покупки подібно системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дронов В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов; БХВ-Петербург - М., 2007. - 480 с.
2. Жадаев Александр PHP для начинающих; "Издательство "Питер" - М., 2014. - 288 с.
3. Леонтьев Борис PHP 5.0 для начинающих, или как создать динамический WEB-сайт; Новый издательский дом - М., 2005. - 176 с.
4. Ховард Майкл , Лебланк Дэвид , Виега Джон Как написать безопасный код на C++, Java, Perl, PHP, ASP.NET; ДМК Пресс - М., 2014. - 288 с.
5. Шкрыль А. PHP - это просто. Программируем для Web-сайта; БХВ-Петербург - М., 2006. - 368 с.
6. WebAssembly Specifications [Електронний ресурс] // Специфікація – Режим доступу до ресурсу: https://webassembly.github.io/spec/.
7. WebAssembly Docs [Електронний ресурс] // Документація – Режим доступу до ресурсу: https://webassembly.org/docs/high-level-goals/.
8. Ванесса В. Подробное руководство HTML5
9. WebSocket. / Ванесса Ванг. – Апресс, 2013. – 205с. – (1-е).
10. Мельников А. The WebSocket Protocol: Стандарт [Електронний ресурс] / Мельников А., Фіт І. - 2011 - Режим доступу: https://tools.ietf.org/html/rfc6455.
11. Кормік Л. MediaDevices: Стаття [Електронний ресурс] / Кормік Л. — 2016 — Режим доступу: https://developer.mozilla.org/ru/docs/Web/API/MediaDevices.
12. Кантелон М. Node.js in Action / Кантелон М., Хартер М., Головайчук Т., Райлих Н. — Manning 2013 — 416 с.

13. Jason L. Realtime Web Apps: With HTML5 WebSocket, PHP, and jQuery / Jason Lengstorf, Phil Leggetter, Alex Newman - Apress, 2013 - 312p.
14. Бази даних [Електронний ресурс] // Чернігівський Національний Технологічний Університет – Режим доступу до ресурсу: http://kpi.stu.cn.ua/bazi-danih/.
15. Дж. Дейт К. Введение в системы баз данных / К. Дж. Дейт. – Москва: ООО "Издательский дом "Вильямс", 2005. – 1328 с. – (8-е).
16. Лекція 5. Бази даних [Електронний ресурс] // Електрона бібліотека МГПУ – Режим доступу до ресурсу: http://library.lp.edu.ua/yak-oformyty-spysok-literatury.
17. Лекція 9 [Електронний ресурс] // Електрона бібліотека Інгулецького коледжу – Режим доступу до ресурсу: http://itdvnzknu.com.ua/188/Лекція%209.doc.
18. Світличний О. О. Моделі даних. Ієрархічна модель даних. Мережна модель даних. Реляційна модель даних [Електронний ресурс] / О. О. Світличний, С. В. Плотницький // Пізнавальний сайт "Географія". Основи геоінформатики – Режим доступу до ресурсу: http://geoknigi.com/book_view.php?id=589.
19. Поняття про бази даних та їх види. Повні уроки [Електронний ресурс] // Гіпермаркет Знань – Режим доступу до ресурсу: http://edufuture.biz/index.php?title=Поняття_про_бази_даних_та_їх_види._Повні_уроки.
20. Объектные СУБД: ситуация смены парадигмы [Электронный ресурс] // Объектно-ориентированный анализ и проектирование – Режим доступа к ресурсу: http://ooad.asf.ru/standarts/Oobd/OCYBD/list02.aspx.
21. Программирование баз данных Oracle для профессионалов / [Р. Гринвальд, Р. Стаковьяк, Г. Додж та ін.]., 2007. – 784 с. – (1-е).
22. Архитектура «файл-сервер» в информационных системах [Электронный ресурс] // Введение в базы данных – Режим доступа к ресурсу: http://pivot-table.ru/arxitektura-fajl-server-v-informacionnyx-sistemax.html.

23. Технологія “клієнт-сервер”. Моделі реалізації цієї технології. [Електронний ресурс] – Режим доступу до ресурсу: http://lubbook.org/book_499.html.
24. Модель п'яти сил конкуренції за М. Портером [Електронний ресурс] – Режим доступу до ресурсу: http://stud.com.ua/45490/ekonomika/.
25. Цибульов П. М. Управління інтелектуальною власністю : монографія/ Цибульов П. М., Чеботарьов В. П., Зінов В. Г. , Суїні Ю., за ред. П. М. Цибульова. – К. : «К. І. С.», 2005. – 448 с.
26. Квашнин А. Как управлять портфелем технологий и интеллектуальной собственностью : серия методических материалов «Практические руководства для центров коммерциализации технологий» / под рук. П. Линдхольма, проект EuropeAid «Наука и коммерциализация технологий», 2006. – 60 с.
27. Квашнин А. Как продвигать проекты коммерциализации технологий : серия методических материалов «Практические руководства для центров коммерциализации технологий» / М. Катешова, А. Квашнин, под рук. П. Линдхольма, проект EuropeAid «Наука и коммерциализация технологий», 2006. – 52 с.
28. Петруненков А. Оценка коммерческой привлекательности проекта [Электронный ресурс] // Технологический бизнес. – 1999. – № 2. Режим доступа:http://www.techbusiness.ru/tb/archiv/number2/page01.htm
29. Тиль, П. От нуля к единице : как создать стартап, который изменит будущее / П. Тиль, Б. Мастерс; перевод с англ. – Москва : Альпина паблишер, 2015. – 188 с.
30. Харниш, В. Правила прибыльных стартапов : как расти и зарабатывать деньги / В. Харниш ; пер. с англ. В. Хозинского. – Москва : Манн, Иванов и Фербер, 2012. – 279 с.
31. Экланд С. Ангелы, драконы и стервятники : как привлечь правильных инвесторов в свой стартап и сохранить бизнес / С. Экланд ; пер. с англ. О. Терентьевой. – Москва : Манн, Иванов и Фербер, 2011. – 275 с.

32. Маллинс, Дж. Поиск бизнес-модели : как спасти стартап, вовремя сменив план / Дж. Маллинс, Р. Комисар ; пер. с англ. М. Пуксант и Е. Бакушевой. – Москва : Манн, Иванов и Фербер, 2012. – 329 с.
33. Паттерны для новичков: MVC [Электронный ресурс] // Хабрахабр. – 2014. – Режим доступа к ресурсу: https://habrahabr.ru/post/215605/.
34. Фримен А. LINQ. Язык интегрированных запросов в C# 2010 для профессионалов / А. Фримен, Д. С. Раттц., 2011. – 656 с. – (Expert's Voice).
35. CodeSmith Tools: Tutorial [Electronic resource] – Access mode: http://www.codesmithtools.com/features/tutorial.
36. Mustache manual [Electronic resource] – Access mode: https://mustache.github.io/mustache.5.html.

ДОДАТОК А

Апробації

УКР.НТУУ “КПІ”.ТВ-з7123МП

Аркушів 7

2018

ДОДАТОК Б

Акт впровадження

УКР.НТУУ “КПІ”. ТВ-з7123МП

Аркушів 2

2018