

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра інформатики, програмної інженерії та економічної
кібернетики

WEB-ДОДАТОК ПІДТРИМКИ ПРОЦЕСУ ВИКОНАННЯ
МАГІСТЕРСЬКИХ РОБІТ З ПРОГРАМНОЇ ІНЖЕНЕРІЇ. БАЗИ
ДАНИХ ТА БІЗНЕС-ЛОГІКА

Кваліфікаційна робота(проект)
на здобуття ступеня вищої освіти «магістр»

Виконав: студент 2 курсу
Спеціальності 126 Інформаційні
системи та технології
Освітньо-професійної програми
«Інформаційні системи та технології»
другого (магістерського) рівня вищої
освіти

Савін Олег Андрійович
Керівник доктор фізико-математичних
наук, професор Львов Михайло
Сергійович
Рецензент кандидат педагогічних наук,
доцентка Гончаренко Тетяна
Леонідівна

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. Загальні положення при написанні магістерських робіт.	
Системи керування проєктами	6
1.1 Загальні положення написання магістерських робіт.....	6
1.2 Системи керування проєктами. Їх огляд та вимоги.....	9
РОЗДІЛ 2. Специфікація вимог програмного забезпечення для web-додатку підтримки процесу виконання магістерських робіт	14
2.1 Огляд програмних засобів для створення web-додатку.....	14
2.2 Специфікації вимог програмного забезпечення.....	16
РОЗДІЛ 3. Розробка web-додатку підтримки процесу виконання магістерських робіт з програмної інженерії	31
3.1 Реалізація web-додатку підтримки процесу виконання магістерських робіт з програмної інженерії.....	31
ВИСНОВКИ	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42
ДОДАТКИ	46
Додаток А.....	46

ВСТУП

Актуальність дослідження.

Процес виконання студентом магістерської роботи складається з великої кількості етапів, основними з яких є:

1. Визначення наукового керівника.
2. Вибір теми дослідження(зі списку який пропонує науковий керівник або власно запропонована тема).
3. Підготовка та написання індивідуального плану магістра.
4. Написання тексту магістерської дисертації.
5. Розробка програмного продукту з теми магістерського дослідження.
6. Написання презентації до виконаної магістерської роботи.

На кафедрі процедура виконання магістерських робіт не формалізована, а тому на будь-якому з вищезазначених етапів можуть виникнути певні проблеми. Щоб цього не траплялося весь процес написання та створення магістерської роботи повинен бути автоматизованим.

Система, що розробляється, є інструментом усунення невизначеностей які виникають внаслідок прийняття різних управлінських рішень. Ця система має допомогти підвищити рівень ефективності під час виконання магістерських робіт.

В умовах сучасного розвитку web-технологій все більше процесів переходять в електронний варіант. Головною причиною цього стала зручність та простота існуючих технологій та інструментів.

Запровадження інформаційно-комунікаційних технологій у закладах вищої освіти мають підвищити ефективність самостійної роботи у студентів та спостити контроль за виконанням робіт для наукових керівників.

2020-й рік продемонстрував необхідність існування системи для

підтримки виконання магістерських робіт. Під час пандемії коли зустріч між науковими керівниками і магістрами занадто ускладнилась, а отже стало складніше контролювати на якому етапі знаходиться виконання заданої роботи. Web-додаток підтримки виконання магістерських робіт повинен стати саме тим інструментом завдяки якому жодна, в тому числі, такі глобальні проблеми як пандемія, не стануть у заваді для наукового процесу.

Мета дослідження – спроектувати web-додаток підтримки процесу виконання магістерських робіт з програмної інженерії використання якого дозволить підвищити ефективність взаємодії студентів і наукових керівників.

Об’єкт дослідження – архітектура, інструменти та технології розробки web-додатка підтримки процесу виконання магістерських робіт.

Предмет дослідження – web-додаток підтримки процесу виконання магістерських робіт, його база даних та бізнес логіка.

Практичне значення одержаних результатів полягає у створенні інструмента для роботи над магістерськими роботами за допомогою web-додатку.

Методи дослідження – під час дослідження було використано:

- Опис
- Порівняння
- Моделювання

Відповідно до мети, об’єкту та предмету дослідження було сформульовано такі **завдання дослідження**:

1. Дослідити та проаналізувати основні вимоги до систем управління проектами.
2. Проаналізувати існуючі системи управління проектами.
3. Описати інструменти та технології розробки web-додатку підтримки процесу виконання магістерських робіт з програмної інженерії та обґрунтувати їх вибір.

4. Спроекувати web-додаток підтримки процесу виконання магістерських робіт з програмної інженерії.
5. Реалізувати web-додаток підтримки процесу виконання магістерських робіт з програмної інженерії.

Зв'язок роботи з науковими програмами, планами, темами.

Кваліфікаційна робота пов'язана з процедурою виконання та формалізації магістерських робіт з програмної інженерії кафедри інформатики, програмної інженерії та економічної кібернетики факультету комп'ютерних наук, фізики та математики Херсонського державного університету.

Наукова новизна одержаних результатів. Результати кваліфікаційної роботи є новими і полягають у реалізації web-додатку підтримки процесу виконання магістерських робіт з програмної інженерії, а саме модулю «представлення».

Практична цінність роботи полягає у тому, що представлені результати дозволяють застосовувати web-додаток підтримки процесу виконання магістерських робіт для допомоги студентам, науковим керівникам, гарантам та завідувачу кафедри для виконання та контролю над магістерськими роботами.

Апробація результатів роботи. Основні положення та результати роботи були обговорені на засіданні кафедри інформатики, програмної інженерії та економічної кібернетики. Також здійснено їх оприлюднення шляхом публікації у фаховому збірнику наукових праць.

Структура роботи. Робота буде складатися з наступних частин: вступ (актуальність дослідження, об'єкт дослідження, предмет дослідження, завдання дослідження), три розділи в котрих описані основні положення роботи, а також висновки та список використаних джерел.

РОЗДІЛ 1

ЗАГАЛЬНІ ПОЛОЖЕННЯ ПРИ НАПИСАННІ МАГІСТЕРСЬКИХ РОБІТ. СИСТЕМИ КЕРУВАННЯ ПРОЄКТАМИ

1.1 Загальні положення написання магістерських робіт

Магістерська робота – це самостійна випускна науково-дослідницька робота. Яка готується з метою подальшого публічного захисту та отримання академічного ступеня магістра, тобто виконує кваліфікаційну функцію.

Основним завданням магістранта – продемонструвати високий рівень своєї наукової кваліфікації, уміння вирішувати поставлені наукові задачі та самостійно здійснювати науковий пошук.

На підставі виконаної роботи під час захисту комісія ухвалює рішення про присвоєння або не присвоєння студенту кваліфікаційного рівня магістра.

Першим завданням студента є вибір теми майбутньої магістерської роботи. Тему для дослідження можна обрати або з запропонованих кафедрою тем, або самому запропонувати тему науковому керівнику та обґрунтувати її доцільність та актуальність.

Після обрання та узгодження тематики випускної роботи магістрант та його науковий керівник складають індивідуальний план на період навчання студента на магістратурі.

Науковий керівник під час процесу виконання магістрантом його наукової роботи повинен: перевіряти виконання студентом його індивідуального плану, рекомендувати вибір джерел інформації, рекомендувати необхідні до застосування методи наукового дослідження, контролювати дотримання магістрантом термінів підготовки магістерської роботи [29].

Тема магістерської роботи має бути сформована до початку проходження студентом магістрантом його виробничої практики на базі

практики.

Вимоги до магістерської роботи:

- Має висвітлювати тему, що недостатньо розроблена або яка має певний аспект вже відомої проблематики
- Виконуватися з актуальної на даний час теми
- Бути чітко побудовану та мати логічне, послідовне викладення матеріалів
- Обґрунтовувати використані методи дослідження
- Приводити до доказових висновків

Під час виконання студентом магістерської роботи будуть проводитися перед захисти виконання завдань, на якій здобувач освіти повинен надати комісії свої поточні результати виконання роботи. Таких перед захистів за час навчання буде декілька. На останньому перед захисті студент має надати написаний текст своєї роботи та виконати свою практичну частину на дев'яносто – сто відсотків. За результатами виконаної роботи комісія повинна прийняти рішення щодо допуску чи не допуску магістра до фінального захисту своєї роботи [22].

Основними процесами написання магістерської роботи є:

1. Вибір наукового керівника. Є важливою частиною майбутнього написання роботи.
2. Вибір тематики наукового дослідження:
 - a. Вибір теми з переліку запропонованим кафедрою.
 - b. Запропонувати і обґрунтувати власну тему.
3. Складання та узгодження з науковим керівником індивідуального плану магістранта на період здобування освіти. За цим планом керівник відстежує успіхи магістранта і те скільки роботи вже зроблено.
4. Сформування остаточної теми магістерської роботи перед початком проходження виробничої практики на обраній базі практики.

5. Проходження виробничої практики. Проводиться для перевірки отриманих за роки навчання практичних навичок магістранта.
6. захист виробничої практики. Практика не перескладається, тому важливість її захисту вкрай важлива для подальшого навчання.
7. Написання тексту дослідницької роботи. Студент має показати вміння отримувати та аналізувати отриману інформацію.
8. Розробка практичного завдання за темою роботи. Магістрант повинен показати свій високий рівень практичної кваліфікації з обраної їм теми.
9. Проходження перед захистів дипломної роботи. Є невід'ємною частиною під час написання магістерської роботи, перевірка того, що зробив студент на даний момент часу.
10. Підготовка презентації щодо виконаної дипломної роботи. За допомогою неї студент буде виступати перед комісією на захисті і показувати яких набув знань під час проходження навчання.
11. Здача екзаменів. Перед останній етап на шляху до отримання ступеня магістра. Перевірка знань яких набув магістрант за час його навчання. Провал цього етапу означає не допуск до захисту роботи.
12. захист магістерської роботи для отримання ступеня магістр. Заключний етап перед набуттям кваліфікації магістра.

Після написання тексту магістерської роботи вона повинна бути перевірена науковим керівником. Після перевірки робота відправляється на перевірку унікальності. Якщо перевірка не пройдена студент або втрачає можливість захистити свою роботу в цьому році, або йому

надається ще одна спроба перевірити роботу на унікальність. Пройдена на перевірку робота підлягає рецензуванню та повинен бути складений відгук про магістерську роботу від наукового керівника [19].

Рецензент повинен оцінити актуальність проведеного дослідження, недоліки зробленої роботи(якщо вони є), рівень знань яких набув магістрант за час свого навчання, ефективність методології яка була використана під час наукового пошуку.

У відгуку науковий керівник повинен позначити, як і якою мірою були виконані поставлені завдання, який було показано рівень теоретичної та практичної підготовки [27].

Виконана повністю робота з текстом, рецензією, відгуком та практичною частиною передається на кафедру за два тижні до фінального захисту. Текст, рецензія та відгук повинні бути надруковані та підшиті.

Під час захисту роботи студент має продемонструвати презентацію та виступити з доповіддю(5-7 хв.) перед комісією та продемонструвати результат виконаної, за період здобування освіти, роботи.

Оголошення результатів захисту проводить голова комісії після виступу усіх студентів, та обговорення питань з іншими членами комісії щодо оцінювання представлених робіт.

Було розглянуто основні стадії написання магістерської роботи студентами магістрантами. Завдання які вони повинні виконати під час навчання на магістратурі. Також були розібрані основні функції наукового керівника які він має виконувати сам та зі своїми студентами магістрантами.

1.2 Системи керування проєктами. Їх огляд та вимоги

Система яку необхідно розробити можна віднести до програмного забезпечення для керування проєктами.

Система керування проєктами – це система завдяки якій можна за допомогою планування, організації та управління ресурсами досягти

необхідних цілей та завершити завдання проєкту [32].

Для того щоб розробити необхідний Web-додаток підтримки процесу виконання магістерських робіт з програмної інженерії. Потрібно провести аналіз серед існуючих систем та відокремити найбільш вагомі частини які знадобляться при розробці нашої системи.

В результаті були розглянуті наступні системи:

1. Jira
2. Asana
3. Wrike
4. Trello
5. Freedcamp
6. Bitrix24
7. ProofHQ

Розглянемо кожну з них більш детально:

Jira – потужний інструмент, проте важкий в освоєнні. Можливість планувати проєкти, призначати виконавців завдань, виставляти пріоритети та дедлайни [7].

Плюси використання – зручно створювати та модифікувати робочі процеси різного обсягу, гнучкі панелі керування проєктами. Можливість створювати процеси, що налаштовуються з можливістю адаптації під будь-який підхід до роботи. Існує мобільна версія. Можливість використання більш ніж трьох тисяч додатків для розширення можливостей [42].

Мінуси використання – досить складна в освоєнні, при великій кількості під задач процес виконання основної задачі буде складніше сприймати [14].

Asana – інструмент, котрий дозволяє розбивати проєкт на розділи, та надає панель інструментів де можливо відстежувати яка частина проєкту вже закінчена [2].

З позитивного – простий для використання, існує можливість

керувати декількома проєктами в рамках однієї команди. Синхронізація в режимі реального часу для різноманітних пристроїв. Наявність версії для мобільних телефонів. Можливість перевіряти поштову скриньку із сервісу [34].

З негативного – неможливо оцінити проєкт в цілому, немає можливості призначати задачі декільком учасникам проєкта [38].

Wrike – система для роботи з проєктами. Є можливість призначати ролі та створювати робочий простір для кожної з них [1]. Присутня можливість стежити за статусом і прогресом завдання. Можливість автоматизувати усі періодично, постійно повторювані процеси. Наявність звітів з усіх проєктів [20]. Наявність мобільної версії додатку. Можливо відображати завдання на діаграмі Ганта(календарний графік).

З позитивних моментів – зручний як для малих так і для великих команд, можливість створювати окремі папки для окремих проєктів. Можливість підлаштувати додаток під себе [30].

З негативних моментів – створюючи розгалужені багаторівневі проєкти, можна легко загубитися в навігації.

Trello – сервіс для систематизації завдань, to-do листи та обговорення ті ідеї на одній дошці. Можливість призначати завдання одразу декільком членам команди [36].

Відгуки за – інтуїтивно зрозумілий, можливість виставляти пріоритети. Можливість синхронізувати з мобільними приладами. Легко додавати контрольні списки. Можливість відстежувати за усім через індикатор прогресу та журнал активності [21].

Відгуки проти – немає можливості додавати опис проєктів і модифікувати під задачі, відсутня можливість відстежувати час [24].

Freedcamp – сервіс з організованим робочим простором для миттєвого перегляду цілого проєкту, виставлення і призначення ряду завдань [6].

Плюси використання – можливість встановлення пріоритетів

задачам, можливість розділити усі проєкти на окремі вкладки, простий у використанні. Можливість відобразити діаграму Ганта [5]. Наявність додатку на мобільних платформах. Наявність файлового сховища.

Мінуси використання – немає можливості виставити прогрес та відсоток виконання завдань [41].

Bitrix24 – ідеальний варіант для великої або для розподіленої команди. Можна відстежувати завантаженість працівників та дедлайни [18].

З плюсів – можна використовувати як CRM(управління відносин з клієнтами) систему, має широкі можливості. Наявність файлового сховища. Наявність багатьох додаткових можливостей – чат, календар, соціальна мережа закритого типу. Ведення статистичної звітності [17].

З мінусів – складне в використанні, додаткова плата за оновлення, дуже мала кількість візуалізації задач і проєктів, відсутність кастомізації [33].

ProofHQ – додаток, який дозволяє ділитися документами, прототипами та ін., разом із командою коментувати і приймати рішення, дозволяти залишити коментарі клієнтам, завантажувати різні версії проєктів та ділитися їми [9].

Відгуки за – простота використання, можливість інтеграції з іншими сервісами. Постійна підтримка додатку. Можливість інтеграції з іншими сервісами [11].

Відгуки проти – відсутність безкоштовної версії.

Великим мінусом майже кожної з розглянутої системи є те що вони умовно безкоштовні, тобто велика кількість необхідних для роботи функцій або можливість додати до проєкту більшу кількість людей за усі ці можливості необхідно платити.

Після розгляду найбільш поширених систем керування проєктами ми змогли виділити основні вимоги які мають включати у себе наступні вимоги.

Вимоги до системи керування проєктами:

1. Керування окремим проєктом
2. Наявність мобільної версії додатку
3. Розподілення ресурсів організації між проєктами
4. Наявність звітності
5. Ефективне керування файлами
6. Відстеження робочого часу
7. Налаштовуванні робочі процеси [31, 37]

На основі отриманої інформації про системи керування проєктами було виділено найбільш загальні вимоги до розробки та створення нашого майбутнього проєкту. Ці вимоги будуть сформульовані, сформовані, деталізовані і записані у вигляді повноцінних вимог до нашого web-додатку у наступному розділі.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ WEB-ДОДАТКУ ПІДТРИМКИ ПРОЦЕСУ ВИКОНАННЯ МАГІСТЕРСЬКИХ РОБІТ

2.1 Огляд програмних засобів для створення web-додатку

Для створення web-додатку було розглянуто декілька варіантів програмних засобів. Було розглянуто наступні варіанти фреймворків для web-розробки:

1. Express JS
2. Django
3. Flask
4. Rails
5. Laravel
6. Spring

Коротко розглянемо їх:

Express JS – фреймворк на JavaScript для створення web-додатків з допомогою Node.js. Даний фреймворк позиціонує себе як швидкий, мінімалістичний та дуже гнучкий. Підтримує REST API [3]. Серед прикладів використання фреймворку – Accenture, IBM та Uber.

Плюсами системи є – гнучкість, простота, детальна документація, широкий вибір модулів [12].

Мінусами системи є – великий обсяг ручної роботи, використання застарілого підходу callbacks функцій [13].

Django – фреймворк на Python для web-розробки. Має структуру Модель-Представлення-Шаблон (Model-View-Template) [8]. Має багато важливих вбудованих функцій – аутентифікація, адміністративна панель, робота з базою даних. Особливе значення приділяється захисту. Серед прикладів використання фреймворку – Google, YouTube, Instagram.

Серед плюсів можна виділити – швидкість створення проєктів, повна

комплектація, масштабованість, безпечність, різнобічність [35].

Мінусами засобу є – усі компоненти розгортаються разом, необхідність володіти усією системою для роботи, використання шаблону маршрутизації з вказанням URL.

Flask – фреймворк на Python. Представляє лише базові можливості [4]. Легкий і гнучкий фреймворк. Він порожній, що дозволяє вибирати модулі для вирішення конкретних задач і встановлювати їх у міру необхідності.

Відгуками за є – легкий для освоєння, гнучкий, хороші інструменти для проведення тестування.

Відгуками проти виступають – не підтримує асинхронність, дефіцит можливостей [25].

Rails – фреймворк на Ruby. Має структуру Модель-Представлення-Контролер (Model-View-Controller). Основні принципи – за замовченням використовуються угоди по конфігурації, типові для більшості додатків, максимальне використання механізмів повторного використання. Серед прикладів використання фреймворку – Airbnb, GitHub, Hulu [16].

Основні переваги – швидкість розробки, висока надійність, масштабованість.

Основні недоліки - складний у вивченні, складність розгортання коробкових проєктів [28].

Laravel – фреймворк на PHP. Використовує архітектурний шаблон Модель-Представлення-Шаблон (Model-View-Template). Підтримка API доступна з коробки [23]. Має велику кількість корисних пакетів з додатковою функціональністю. Найбільша проблема – недостатня продуктивність в порівнянні з Django та Express JS, для великих проєктів це може стати суттєвим мінусом.

Плюси фреймворку – зрозуміла документація, гнучка система маршрутизації, зручний механізм обробки помилок і виключень.

Мінуси фреймворку – не логічне розташування каталогів і файлів,

порушена зворотна сумісність між версіями фреймворка.

Spring – фреймворк на Java. Професійний, гнучкий та дуже надійний. Представляє собою колекцію фреймворків у фреймворку, більшість з яких може працювати незалежно один від одного [10].

Перевагами є – підтримує різні способи конфігурації, підтримує декларативне програмування, дозволяє вільно зв'язувати модулі та легко їх тестувати.

Недоліками є – повільний запуск великих додатків.

В результаті розгляду та порівняння програмних засобів для розробки web-додатку було обрано мову Python.

Вибір на користь Python був зумовлений завдяки по-перше надзвичайно великій спільноті програмістів мови Python, а отже на майже кожне запитання котре виникало під час розробки програмного продукту вже є відповідь. По друге через власний досвід роботи з цією мовою програмування, а отже можливість одразу розпочати розробку нашого web-додатку.

Серед провідних засобів на Python були Flask та Django.

В результаті їх порівняння вибір був зроблений на користь Django. Це рішення супроводжувалося тим, що даний фреймворк має набагато більш широкий інструментарій для розробки web-додатків на його основі, а отже і можливість розробки якісного продукту була вище ніж при використанні іншого програмного засобу.

Для створення архітектури бази даних використовувався MySQL.

Серед переваг можна виділити:

- Гнучка система привілеїв і паролів
- Можливість запису фіксованої та змінної довжини
- Багатопоточність
- Підтримка декількох одночасних запитів [15]

2.2 Специфікації вимог програмного забезпечення

Специфікація вимог програмного забезпечення – це опис поведінки

розробляємої програми. Вона включає у себе ряд вимог, наприклад ряд користувальницьких історій задля опису усіх можливих варіантів взаємодії між користувачем системи та програмним засобом [39].

Специфікація вимог є основою для подальшого планування, дизайну та кодування. Це повний опис поведінки системи що розробляється, яка включає у себе множину прецедентів які показують усі взаємодії, які будуть мати користувачі з програмним кодом.

Специфікація складається з функціональних та нефункціональних вимог до системи.

Цей розділ покликаний забезпечити загальне розуміння задач та цілей web-додатку, що розробляється.

2.2.1 Ролі користувачів системи

Роль користувача в системі – це сукупність різноманітних можливостей котрі отримує користувач системи знаходячись в певній системній ролі.

Гість

Гостем може бути будь-яка людина котра потрапить на даний ресурс. Проте єдина можлива функція гостя це авторизуватися у системі якщо у нього є створений аккаунт(аккаунти створюють системні адміністратори сайту).

Системний адміністратор

Системним адміністратором є людина яка працює на факультеті комп'ютерних наук, фізики та математики у закладі Херсонського державного університету. Ця людина має доступ до усього функціоналу сайту. Також системний адміністратор слідкує за працездатністю системи.

Магістр

Магістром є людина котра поступила до Херсонського державного університету та обрала своїм призначенням факультет комп'ютерних

наук, фізики та математики. Це одна з основних ролей web-додатку та перша по кількості людей серед ролей. Магістри можуть обирати теми магістерських робіт, додавати файли пов'язані зі своєю темою та спілкуватися зі своїм науковим керівником.

Науковий керівник

Науковий керівник це працівник освіти який викладає на факультеті комп'ютерних наук, фізики та математики у Херсонському державному університеті. Серед основних функцій – додавання власних тем магістерських робіт та перевірка завантажених робіт своїх магістрантів.

Гарант

Гарант це працівник освіти на факультеті комп'ютерних наук, фізики та математики у Херсонському державному університеті який відповідає за свою спеціальність. Може переглядати магістерські роботи усіх студентів своєї спеціальності.

Завідувач кафедри

Завідувач кафедри це працівник кафедри на факультеті комп'ютерних наук, фізики та математики у Херсонському державному університеті. Серед основних функцій – встановлення строків на виконання магістрами їх робіт.

Були розглянуті основні ролі користувачів та опис людей які будуть виступати у цих ролях системи, що буде розроблятися.

2.2.2 Архітектура системи

Архітектура програмного забезпечення – це спосіб для структуризації обчислювальної або програмної системи.

Архітектура системи нашого web-додатку представляє собою структуру – модель-шаблон-представлення (рис. 2.1).

Моделі відповідають за бізнес-логіку, методи та властивості зв'язані з маніпуляціями над даними. Також моделі дозволяють створювати, редагувати, читати та видаляти об'єкти з бази даних.

Django для нормального функціонування системи та моделі які були створені в процесі розробки web-додатку підтримки процесу виконання магістерських робіт з програмної інженерії.

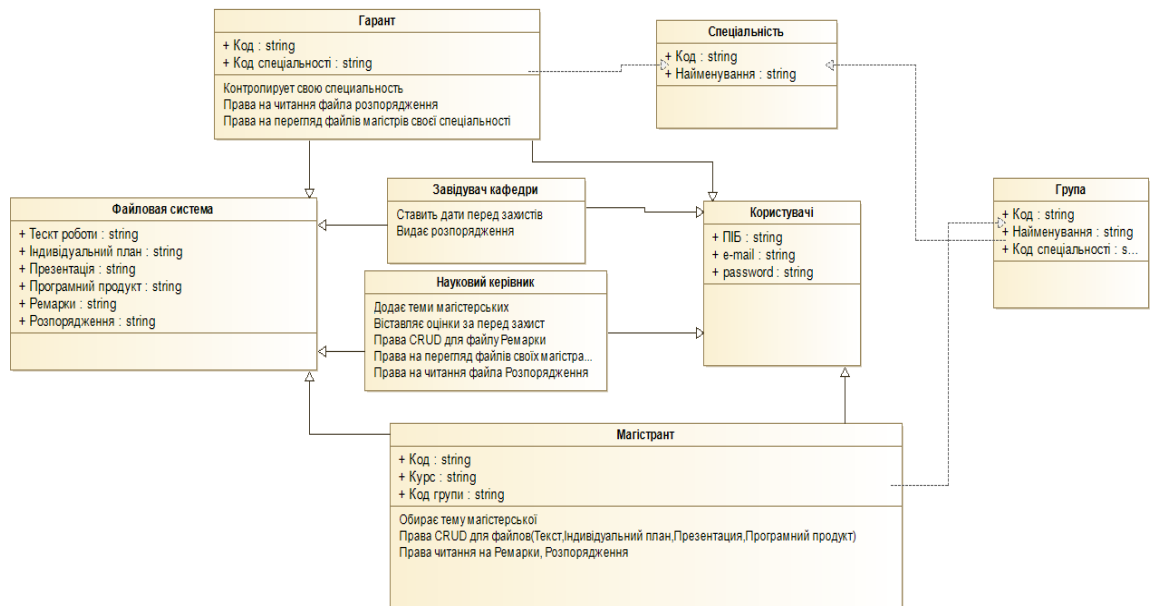


Рисунок 2.3 - Діаграма класів

Представлена діаграма класів демонструє структуру системи через опис класів та їх атрибутів, а також методи та відношення між об'єктами (рис. 2.3).

Було розглянуто архітектуру проекту на Django, її основні особливості. Архітектуру розробленої бази даних для нашого web-додатку. Та діаграму класів, що відображає існуючі у системі класи та їх атрибути.

2.2.3 Вимоги користувачів у форматі User Stories

Специфікація буде представлена у вигляді User Stories (Історії користувача).

Історія користувача (User Story) являє собою короткий виклад функціональних можливостей, реалізація яких необхідна для отримання конкретним зацікавленою особою користі від програмного продукту.

Як гість:

1. Можу увійти в систему

Як магістр:

1. Можу увійти в систему
2. Можу вийти з системи
3. Можу писати особисті повідомлення(завідувачу кафедри, науковому керівнику, гаранту)
4. Можу додавати свій індивідуальний план
5. Можу редагувати свій індивідуальний план
6. Можу видаляти свій індивідуальний план
7. Можу додавати свій текст магістерської роботи
8. Можу редагувати свій текст магістерської роботи
9. Можу видаляти свій текст магістерської роботи
10. Можу додавати свій програмний продукт магістерської роботи
11. Можу редагувати свій програмний продукт магістерської роботи
12. Можу видаляти свій програмний продукт магістерської роботи
13. Можу додавати свою презентацію магістерської роботи
14. Можу редагувати свою презентацію магістерської роботи
15. Можу видаляти свою презентацію магістерської роботи
16. Можу читати ремарки до свого індивідуального плану
17. Можу читати ремарки до свого тексту магістерської роботи
18. Можу читати ремарки до свого програмного продукту магістерської роботи
19. Можу читати ремарки до своєї презентації магістерської роботи
20. Можу обирати наукового керівника
21. Можу обирати тему магістерської роботи
22. Можу дивитися оцінки за попередні захисти дипломної роботи

23. Можу дивитися «Допуск/Не допуск» до захисту магістерської роботи

Як науковий керівник:

1. Можу увійти в систему
2. Можу вийти з системи
3. Можу додавати тему магістерської роботи
4. Можу читати індивідуальний план свого магістранта
5. Можу писати ремарки до індивідуального плану свого магістранта
6. Можу читати текст роботи свого магістранта
7. Можу писати ремарки до тексту роботи свого магістранта
8. Можу переглядати програмний продукт свого магістранта
9. Можу писати ремарки до програмного продукту свого магістранта
10. Можу переглядати презентацію свого магістранта
11. Можу писати ремарки до презентації свого магістранта
12. Можу писати особисті повідомлення(завідувачу кафедри, магістрантам, гаранту)
13. Можу виставляти оцінки за попередні захисти дипломної роботи магістрантом
14. Можу ставити магістранту «Допуск» чи «Не допуск» до захисту дипломної роботи

Як гарант:

1. Можу увійти в систему
2. Можу вийти з системи
3. Можу писати особисті повідомлення(завідувачу кафедри, науковим керівникам, магістрантам)
4. Можу переглядати індивідуальні плани магістрантів своєї спеціальності
5. Можу переглядати тексти магістрантів своєї спеціальності

6. Можу переглядати програмні продукти магістрантів своєї спеціальності
7. Можу переглядати презентації магістрантів своєї спеціальності

Як завідувач кафедри:

1. Можу увійти в систему
2. Можу вийти з системи
3. Можу писати особисті повідомлення(магістрантам, науковим керівникам, гарантам)
4. Можу виставляти строки вибору тем
5. Можу виставляти строки здачі індивідуальних планів магістрантів
6. Можу виставляти строки здачі тексту роботи магістрантів
7. Можу виставляти строки здачі програмних продуктів магістрантів
8. Можу виставляти строки здачі презентацій роботи магістрантів

Як системний адміністратор:

1. Можу увійти в систему
2. Можу вийти з системи
3. Можу додавати користувачів у систему
4. Можу видаляти користувачів з системи

Отже, за допомогою описаних вище історій користувача ми можемо легше зрозуміти, що потрібно робити на далі для розробки web-додатку підтримки процесу виконання магістерських програм з програмної інженерії.

Також завдяки формату User stories ми змогли легше розробляти подальші діаграми для проєкту.

2.2.4 Діаграми прецедентів для основних ролей

На основі описаних раніше «User story» були створені наступні

діаграми прецедентів.



Рисунок 2.4 - Діаграма прецедентів для актора «Гість»

Роль «Гість» відіграє у нашому проєкті найменшу роль (рис. 2.4).

Єдине, що можна зробити знаходячись у даній ролі – це увійти до системи якщо у тебе існує аккаунт на проєкті. Це зроблено для того щоб ніхто окрім учасників навчального процесу не зумів отримати жодних даних з нашого web-додатку.

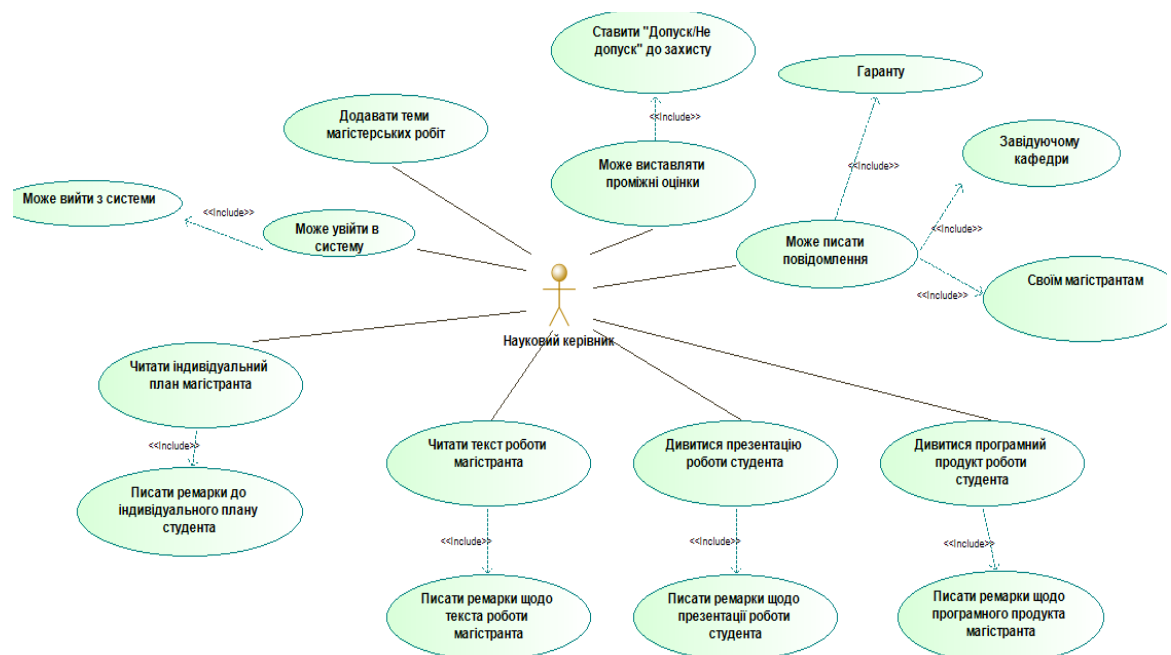


Рисунок 2.5 - Діаграма прецедентів для актора «Науковий керівник»

Роль «Науковий керівник» - одна з головних ролей системи, що розробляється (рис. 2.5).

Науковий керівник на початку року, за час відведених на це завідувачем кафедри, додає свої теми магістерських робіт у web-додаток. Та очікує коли магістранти почнуть обирати теми та керівників з дипломного проєкту.

Після цього з'являється можливість писати ремарки до доданих, своїми студентами, файлів.

Після проходження магістрантами перед захистів науковий керівник виставляє студентам проміжні оцінки. У нашому web-додатку є два типи оцінок – персональні та за перед захисти. Персональні оцінки науковий керівник виставляє під час усього навчального процесу за процес який виконав магістрант на даному етапі своєї роботи. Та оцінки за перед захист, ці оцінки виставляють після того як магістранти перед комісією розповідають та показують успіхи своєї роботи.

Після останнього перед захисту науковий керівник може вирішити допускати чи не допускати магістранта до захисту його магістерської дисертації.

Після фінального перед захисту у наукового керівника з'являється можливість додати до web-додатку файл з відгуком на текст магістерської роботи.

Роль «Магістрант» - найважливіша роль у розробці даної системи (рис. 2.6).

Після того як студент вибере тему магістерської роботи йому стане доступна можливість додавати файли своєї роботи до web-додатку.

Надалі під своїми доданими файлами він зможе спостерігати додані ремарки від свого наукового керівника, щоб надалі згідно з ними корегувати свою роботу.

Після проходження перед захистів магістрант матиме змогу бачити свої оцінки за нього.

Після останнього перед захисту студент дізнається про допуск чи не допуск до захисту всього дипломного проєкту.

Також після фінального перед захисту у магістранта є доступ до читання файлів відгука та рецензії щодо своєї виконаної роботи.

Роль «Завідувач кафедри» відповідає за встановлення строків здачі різноманітних файлів магістранта, перед захистів та захисту

магістерських робіт (рис. 2.7). Він може додавати до системи файли з розпорядженнями.

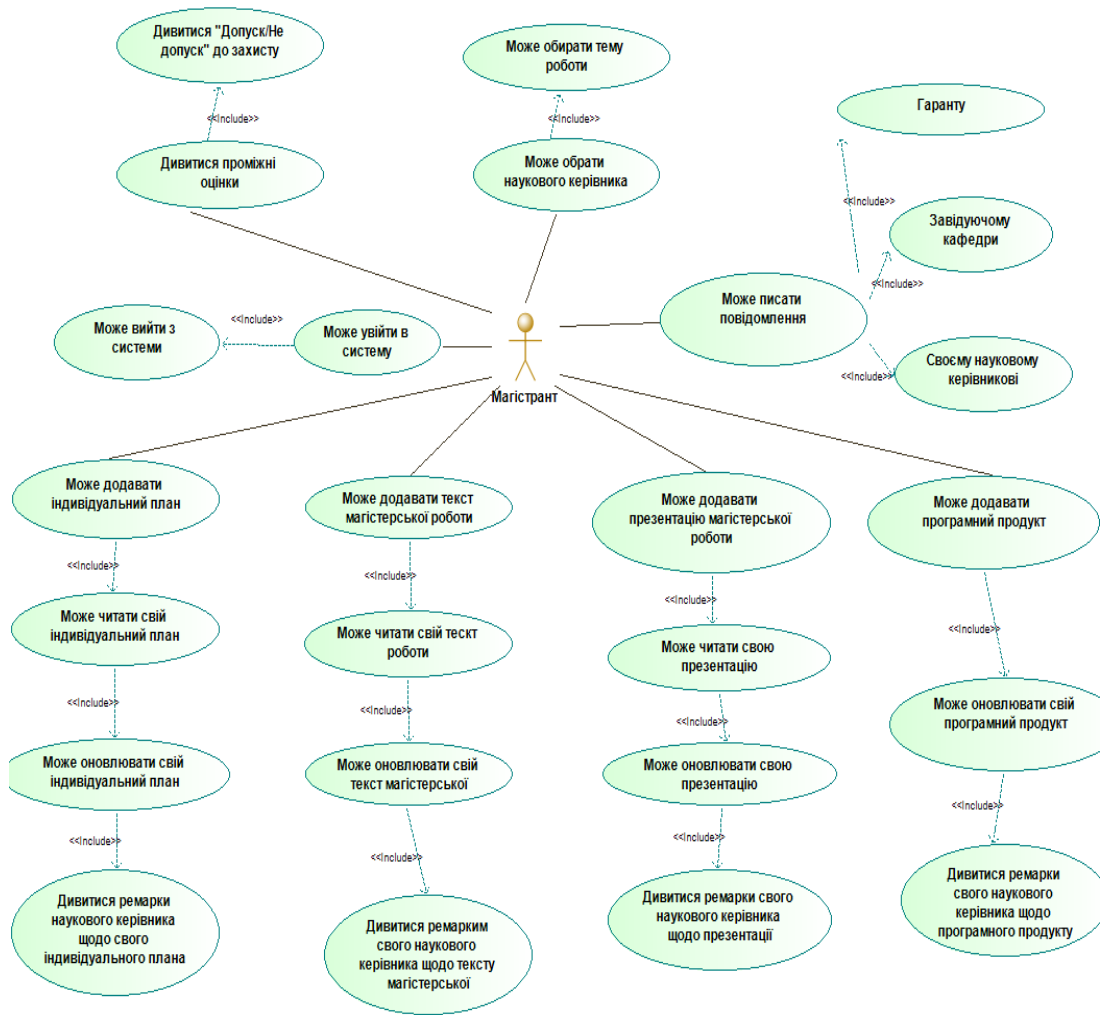


Рисунок 2.6 - Діаграма прецедентів для актора «Магістрант»

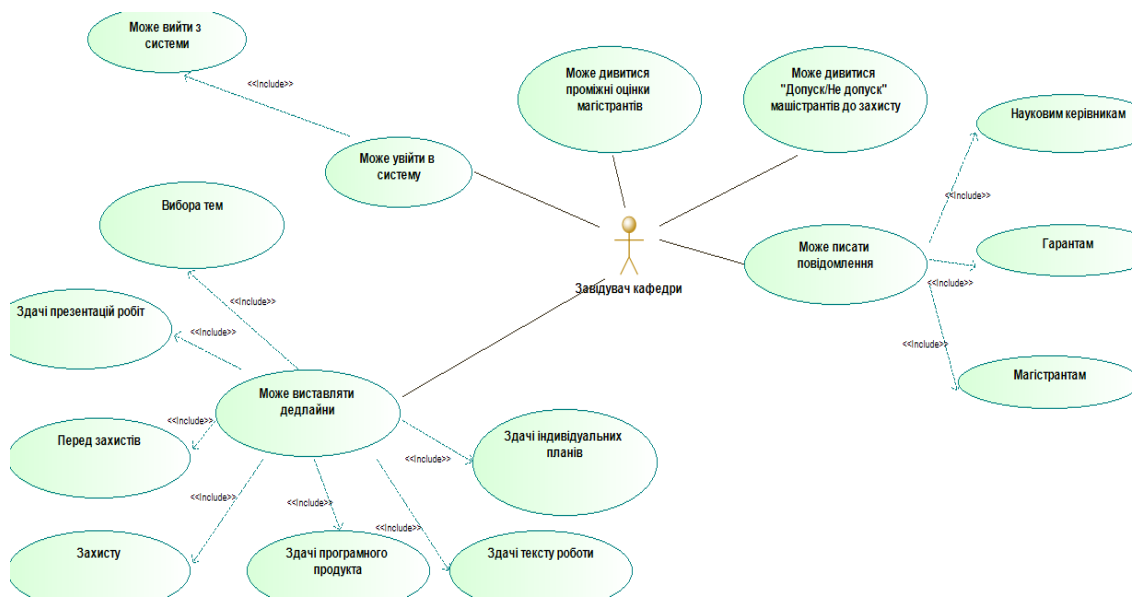


Рисунок 2.7 - Діаграма прецедентів для актора «Завідувач кафедри»

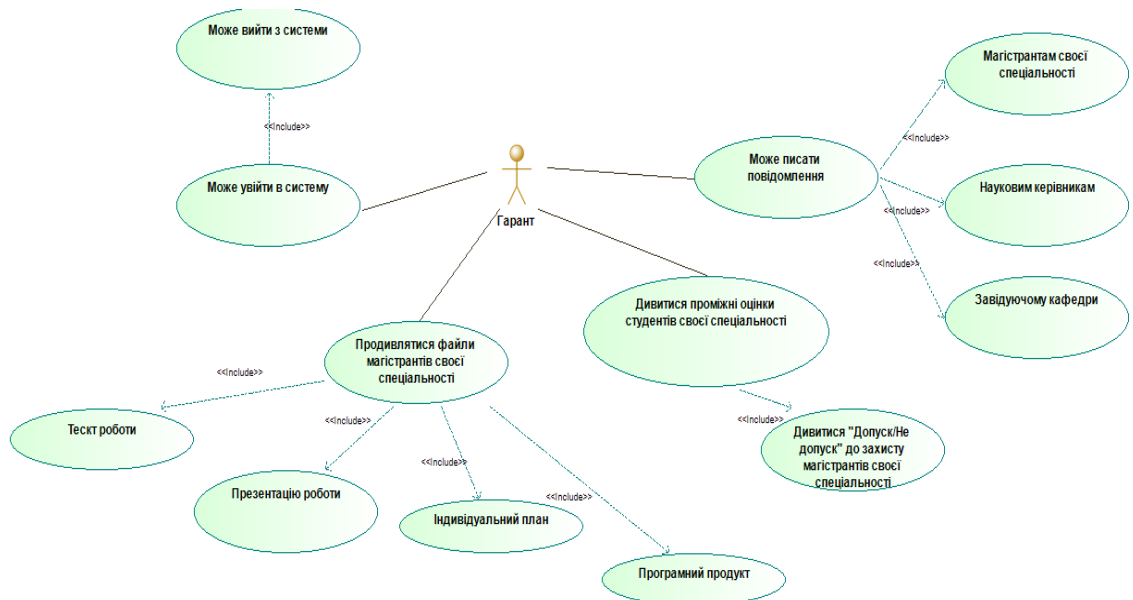


Рисунок 2.8 - Діаграма прецедентів для актора «Гарант»

Роль «Гарант» має більш ширші можливості ніж роль «науковий керівник» (рис. 2.8). Він може бути не лише керівником з дипломної роботи у магістрантів, а й може спостерігати за прогресом работ усіх студентів котрі належать до його спеціальності.

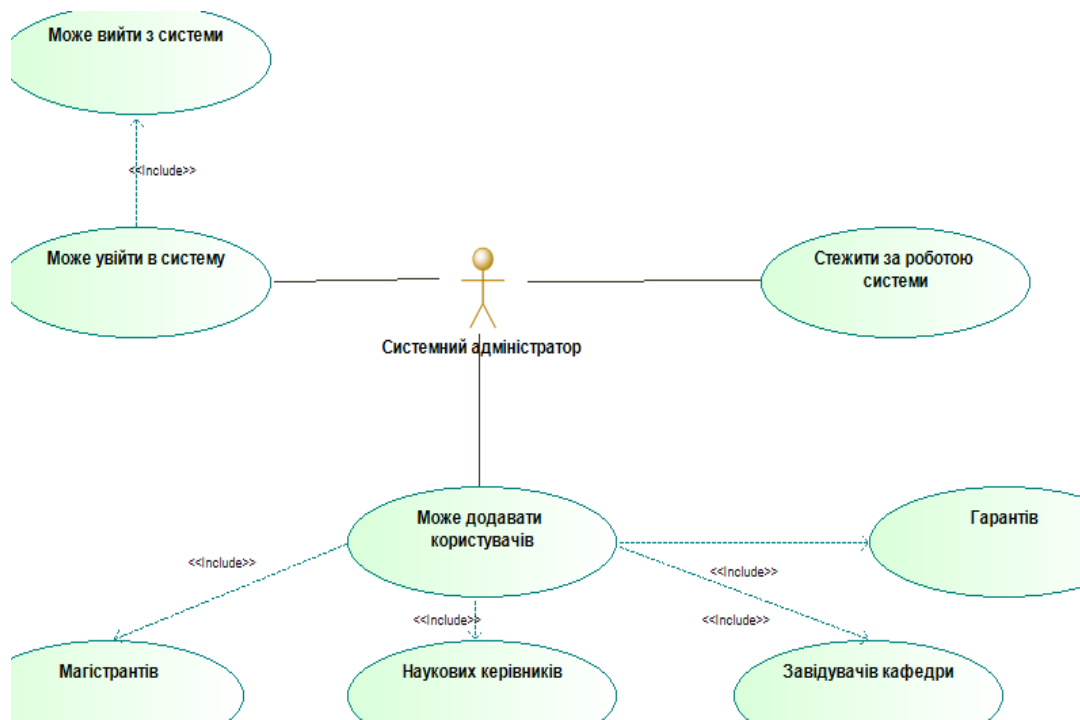


Рисунок 2.9 - Діаграма прецедентів для актора «Системний адміністратор»

Роль «Системний адміністратор» відповідає за стабільну роботу web-додатку (рис. 2.9).

За допомогою наведених вище діаграм прецедентів було відображено відносини між акторами та прецедентами. Це дозволяє описати систему на концептуальному рівні.

2.2.5 Функціональні вимоги

Web-додаток, що розробляється буде складатися з наступних програмних модулів:

1. Файлове сховище
2. Особистий кабінет
3. Чат
4. Авторизація
5. Інтерфейс
6. База даних
7. Календар

Функціональні вимоги до програмного модуля "Файлове сховище"

Для магістра

Можливість створювати, редагувати, читати та видаляти файли своєї магістерської роботи (індивідуальний план, текст роботи, програмний продукт та презентація). Також можливість читання файлів наукового керівника та завідувача кафедри(ремарки, розпорядження).

Для наукового керівника

Можливість читати та продивлятися файли своїх магістрантів(текст роботи, презентація, індивідуальний план, програмний продукт), та розпорядження від завідувача кафедри. Можливість до цих файлів додавати, редагувати, видаляти ремарки(тільки магістрам для яких ваша роль науковий керівник).

Для гаранта

Має ті самі можливості як і науковий керівник, але й може продивлятися файли студентів зі своєї спеціальності.

Для завідувача кафедри

Можливість створювати, видаляти, редагувати розпорядження.

Функціональні вимоги до програмного модуля "Особистий кабінет"

Користувач кабінету повинен мати змогу змінювати особисту інформацію про себе, наприклад фото, ідентифікатор в скайпі та телеграмі.

Для магістранта

Можливість перейти до перегляду власно доданих файлів та ремарок до них, доступ до чату з іншими користувачами, можливість переглянути розпорядження, можливість перейти до вибору наукового керівника та теми магістерської роботи.

Для наукового керівника та гаранта

Доступ до сторінки додавання теми дипломної роботи. Доступ до приватних повідомлень. Можливість перейти до перегляду файлів своїх магістрантів та додавання ремарок до переглянутих файлів студента.

Для гаранта

Має ті ж самі можливості як і науковий керівник. Проте може продивлятися усі файли магістрантів своєї спеціальності. Можливість писати ремарки тільки студентам, що пишуть наукову роботу у гаранта.

Для завідувача кафедри

Можливість додавати, редагувати чи видаляти розпорядження. Доступ до встановлення строків перед захистів, захисту, можливості додавання файлів магістрами та науковими керівниками. Доступ до чату з іншими користувачами системи. Доступ до списку тем-наукових керівників-магістрантів. Доступ до списку студентів.

Функціональні вимоги до програмного модуля "Чат"

Для усіх користувачів системи, окрім гостей

Доступ до обміну приватними повідомленнями між користувачами, та можливість обміну повідомленнями у групі.

Функціональні вимоги до програмного модуля "Авторизація"

Можливість здійснити вхід до системи за допомогою введення

логіну(e-mail) та пароллю.

Функціональні вимоги до програмного модуля "Інтерфейс"

1. Можливість повністю охопити весь набір завдань, які виникають під час користування системою.
2. Інтерфейс повинен мати елементи для керування системою, котрі дозволять налаштовувати web-додаток відповідно з бажанням користувача.
3. Він повинен мати просту та зрозумілу навігацію яка має спростити взаємодію користувачів і системи.
4. Інтерфейс повинен мати простий візуальний дизайн екрані програми
5. Коректно відображатися на різних налаштуваннях розширення екрану та різноманітних браузерах.

Функціональні вимоги до програмного модуля "База даних"

Швидкий пошук і отримання інформації, простота оновлення даних, захист даних від несанкціонованого доступу.

Функціональні вимоги до програмного модуля "Календар"

Цей модуль призначений для змоги користувачами відстежувати час виконання своїх обов'язків. Завідувач кафедри встановлює строки на додавання та вибір тем, на період написання магістерських робіт, на дати перед захистів та захисту.

За допомогою складених вище функції стало зрозуміло які вимоги ми ставимо до розроблюваного web-додатку, що і як він повинен робити і як реагувати на поставлені перед ним задачі.

Розробив зазначені вище вимоги можна приступати до останнього етапу нашої роботи – розробки нашого web-додатку.

РОЗДІЛ 3

РОЗРОБКА WEB-ДОДАТКУ ПІДТРИМКИ ПРОЦЕСУ ВИКОНАННЯ МАГІСТЕРСЬКИХ РОБІТ З ПРОГРАМНОЇ ІНЖЕНЕРІЇ

3.1 Реалізація web-додатку підтримки процесу виконання магістерських робіт з програмної інженерії

Як було зазначено у минулому розділі під час аналізу програмних засобів розроблятися наш web-додаток буде на мові Python за допомогою фреймворку Django.

Даний фреймворк надає великий функціонал та багато можливостей для розробки різноманітних web-додатків з його допомогою.

Розглянемо основний й важливий функціонал який представляє до нашої уваги Django.

Має вбудовану:

- Адміністративну панель
- Форму авторизації
- Переклад на багато мов
- Бібліотеку для роботи з формами
- API доступу до бази даних з підтримкою транзакцій

Отже за допомогою цих вбудованих, а також власно розроблених функцій та модулів і було створено наш web-додаток підтримки процесу виконання магістерських робіт з програмної інженерії.

На головній сторінці розробленого нами web-додатку буде форма входу до системи (рис. 3.1).

Увійти до неї можливо за допомогою логіну та паролю. Реєстрацію буде проводити системний адміністратор, щоб уникнути створення аккаунтів з недостовірними даними. Після цього логіни і паролі будуть надані майбутнім користувачам, які після першого входу у систему зможуть змінити ці дані, а також заповнити більш детальну інформацію

про себе.

Після того як буде здійснений вхід до системи нас буде перенаправлено на сторінку особистого кабінету.

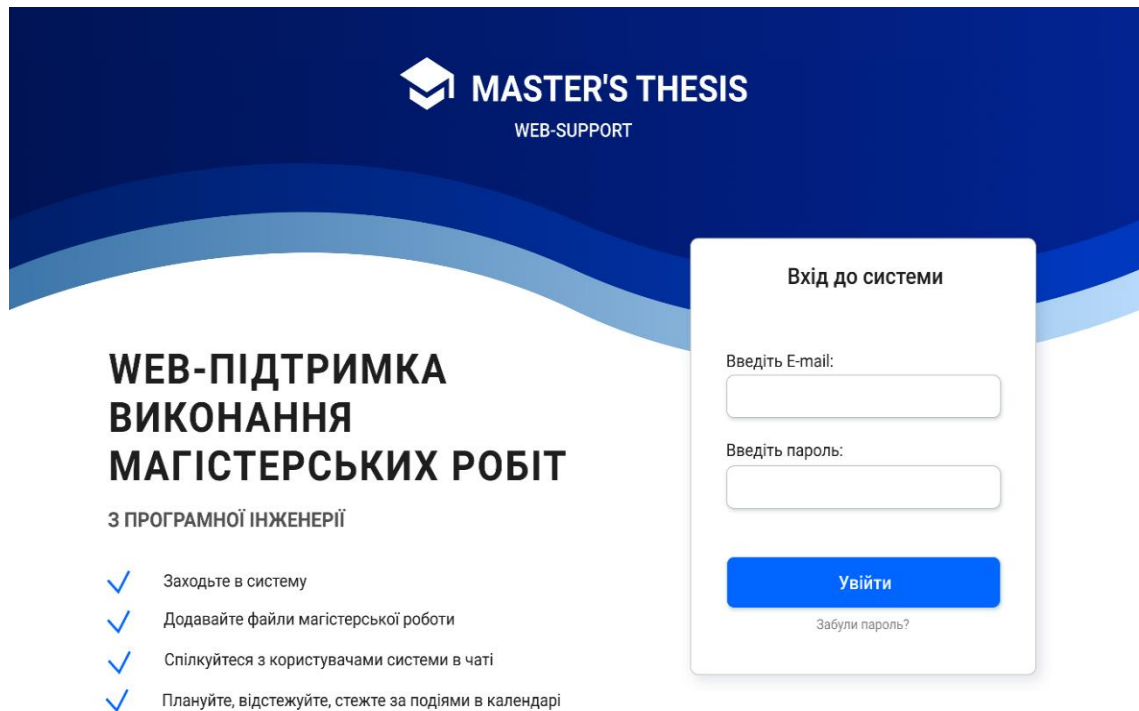


Рисунок 3.1 - Головна сторінка сайту

На цій сторінці можна буде переглянути інформацію про себе (рис. 3.2).

Також тут є кнопка «Редагувати інформацію». Натиснув на неї буде відкрита сторінка з формою на якій ми можемо змінити інформацію про нас, а саме – пароль, e-mail, прізвище, ім'я, по-батькові, логін у скайпі, у телеграмі, фото профілю. Після заповнення даної інформації натиснути кнопку «Зберегти» і нас поверне на сторінку з інформацією про нас, де окрім заповненої раніше інформації буде ще інформацію про нашу роль на сайті(магістр, завідувач кафедри, науковий керівник, гарант), згідно з цієї ролі буде відображена інформація про групу та спеціальність у магістра, наукового ступеня для наукового керівника та гаранта, та спеціальність за яку відповідає гарант.

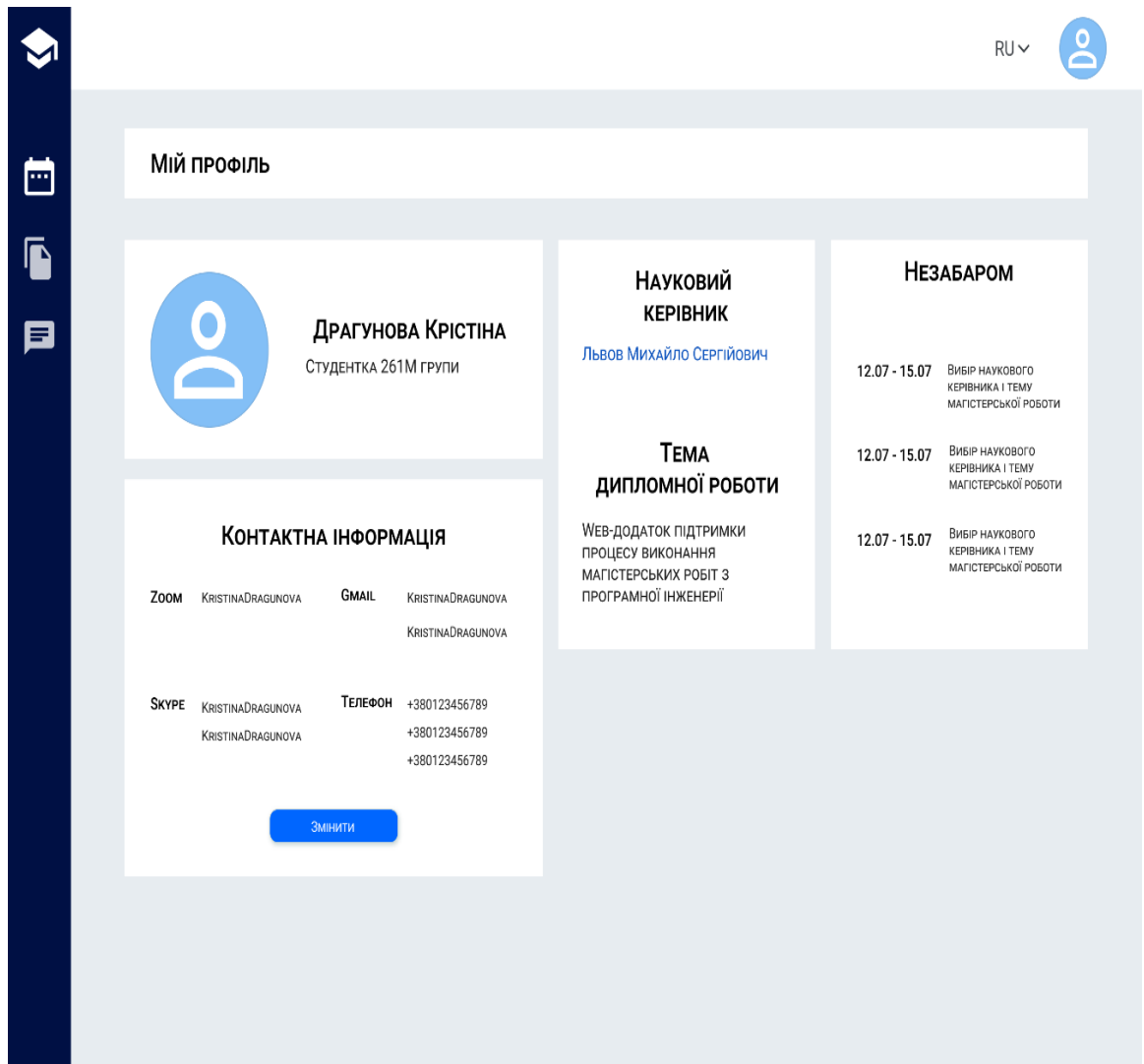


Рисунок 3.2 - Сторінка особистого кабінету

За допомогою панелі бокового меню можна перейти на список тем(для усіх ролей системи), до файлового сховища(для магістрів та їх наукових керівників), до сторінки з різноманітною інформацією яку надає завідувач кафедри, на сторінку з оцінками та отримати доступ до персональних повідомлень завдяки яким студенти можуть спілкуватися з науковими керівниками та завідувачем кафедри.

Після переходу на сторінку з темами стане доступна таблиця з інформацією «науковий керівник - тема - магістр» (рис. 3.3).

На цій сторінці наукові керівники можуть додавати свої теми магістерських робіт (рис. 3.4). А магістранти обирати серед цих робіт.

Після того як студент визначиться з керівником та темою він натисне навпроти неї кнопку «Обрати» і його прізвище з'явиться навпроти теми.

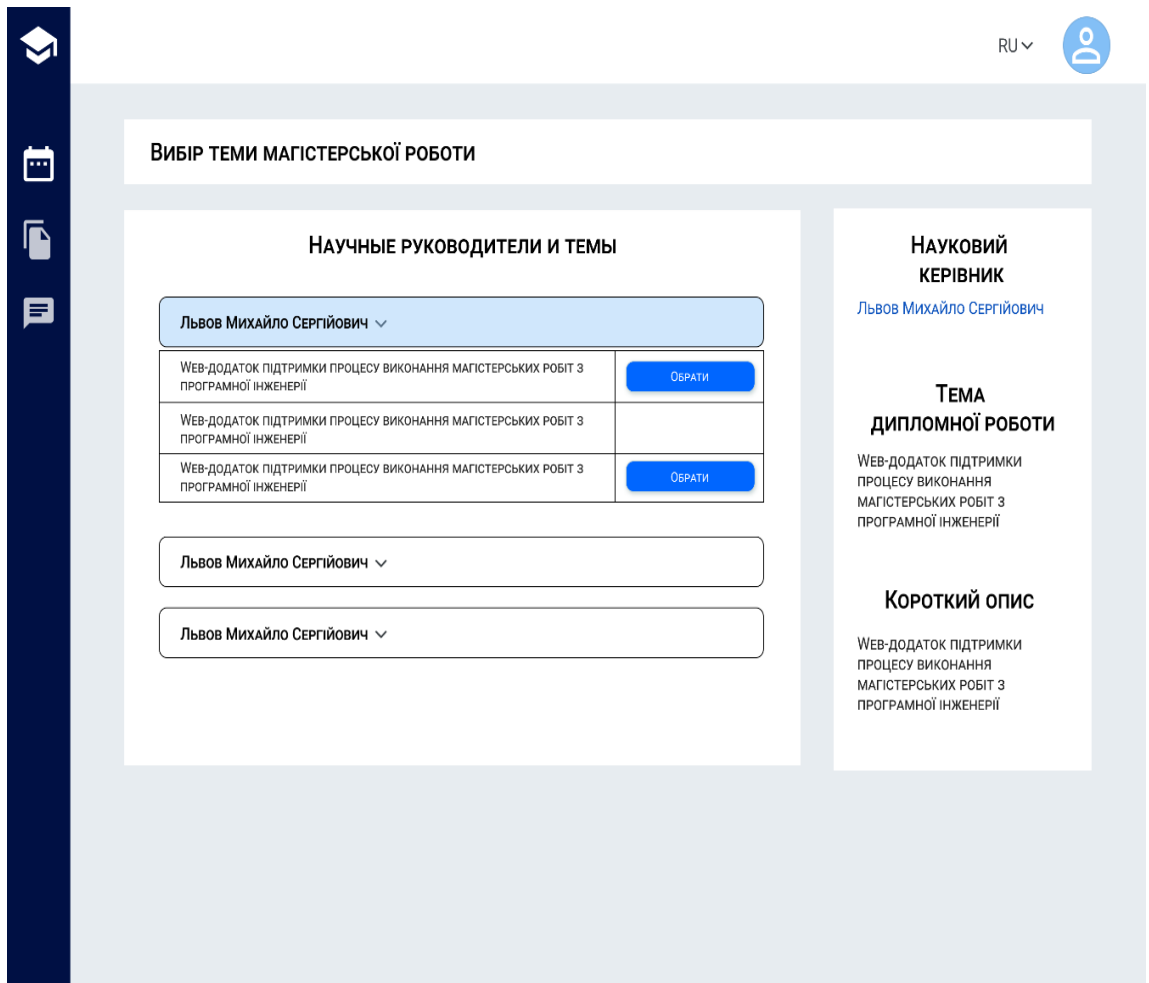


Рисунок 3.3 - Сторінка списку тем магістерських робіт

Рисунок 3.4 - Форма додавання теми роботи

Після того як буде обрана тема магістру становиться доступна можливість додавати файли своєї магістерської роботи (рис. 3.6). А саме текст роботи, індивідуальний план, програмний продукт та презентація роботи.

Усі додані магістрантом файли будуть відображатися на цій сторінці, вони будуть відсортовані за типом виконаної роботи (рис. 3.5).

Мої файли дипломної роботи

ІНДИВІДУАЛЬНИЙ ПЛАН ТЕКСТ ДИПЛОМНОЇ РОБОТИ ПРЕЗЕНТАЦІЯ ДИПЛОМНОЇ РОБОТИ

Код програмного продукту Архів з файлами

Архівні файли

Назва файлу	Тип файлу	Дата додавання	Дата видалення
Текст дипломної роботи	ТХТ	19.09.2020	19.09.2020
Текст дипломної роботи	ТХТ	19.09.2020	19.09.2020
Текст дипломної роботи	ТХТ	19.09.2020	19.09.2020
Текст дипломної роботи	ТХТ	19.09.2020	19.09.2020

Науковий керівник
Львов Михайло Сергійович

ТЕМА ДИПЛОМНОЇ РОБОТИ
WEB-ДОДАТОК ПІДТРИМКИ ПРОЦЕСУ ВИКОНАННЯ МАГІСТЕРСЬКИХ РОБІТ З ПРОГРАМНОЇ ІНЖЕНЕРІЇ

НЕЗАБАРОМ

12.07 - 15.07 ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ

12.07 - 15.07 ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ

12.07 - 15.07 ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ

Рисунок 3.5 - Сторінка з файлами магістранта

Raw data HTML form

Назва роботи

Назва роботи

Тип роботи

Опис роботи

Що нового

Файл роботи Файл не вибран
Завантажте файл вашої магістрської роботи

User

Додав

Theme

Тема роботи

Рисунок 3.6 - Форма додавання файлу магістрантом

Для наукового керівника на цій сторінці будуть відображатися прізвища його студентів, а під ними їх додані у систему файли.

Після переходу на огляд інформації про один конкретний файл,

науковий керівник зможе завантажити його до себе на комп'ютер.

```

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 1,
    "name": "Ремарка до роботи Савіна Олега Андрійовича",
    "description": "Опис ремарки до роботи магістра Савіна Олега Андрійовича",
    "remark": "http://127.0.0.1:8000/media/masterfiles/admin/unnamed.jpg",
    "user": "Львов Михайло Сергійович",
    "work": "Веб-додаток підтримки процесу виконання магістерських робіт з програмної інженерії. База даних та бізнес-логіка. Савін Олег Андрійович."
  }
]

```

Рисунок 3.7 - Ремарка наукового керівника

Після перевірки він може написати ремарку щодо переглянутого файлу (рис. 3.7).

За допомогою ремарок залишених науковим керівником магістранти можуть скорегувати свою роботу і зробити її більш якісною.

Ремарки до файлів магістрантів можна додавати у проміжок часу з початку написання магістерської роботи до моменту коли магістрантам буде заборонено додавати чи редагувати файли робіт. Час у який можна додавати визначається завідувачем кафедри.

Прочитанный: True

- С кем беседа: test1
- Последнее сообщение: 22 октября 2020 г. 18:52

Текст сообщения: привет тест 1

Прочитанный: True

- С кем беседа: Magistr
- Последнее сообщение: 22 октября 2020 г. 18:53

Текст сообщения: привет

Рисунок 3.8 - Список діалогів

На сторінці повідомлень користувачі системи бачать їхні діалоги з іншими користувачами системи (рис. 3.8). Натиснувши на повідомлення вони перейдуть до конкретного діалогу де будуть здатні обмінюватися повідомленнями з обраною людиною (рис. 3.9).

Користувачі системи здатні надсилати персональні повідомлення або використовувати груповий чат.

Модуль «Персональні повідомлення» дозволить не користуючись

іншими сторонніми ресурсами обмінюватися повідомленнями.

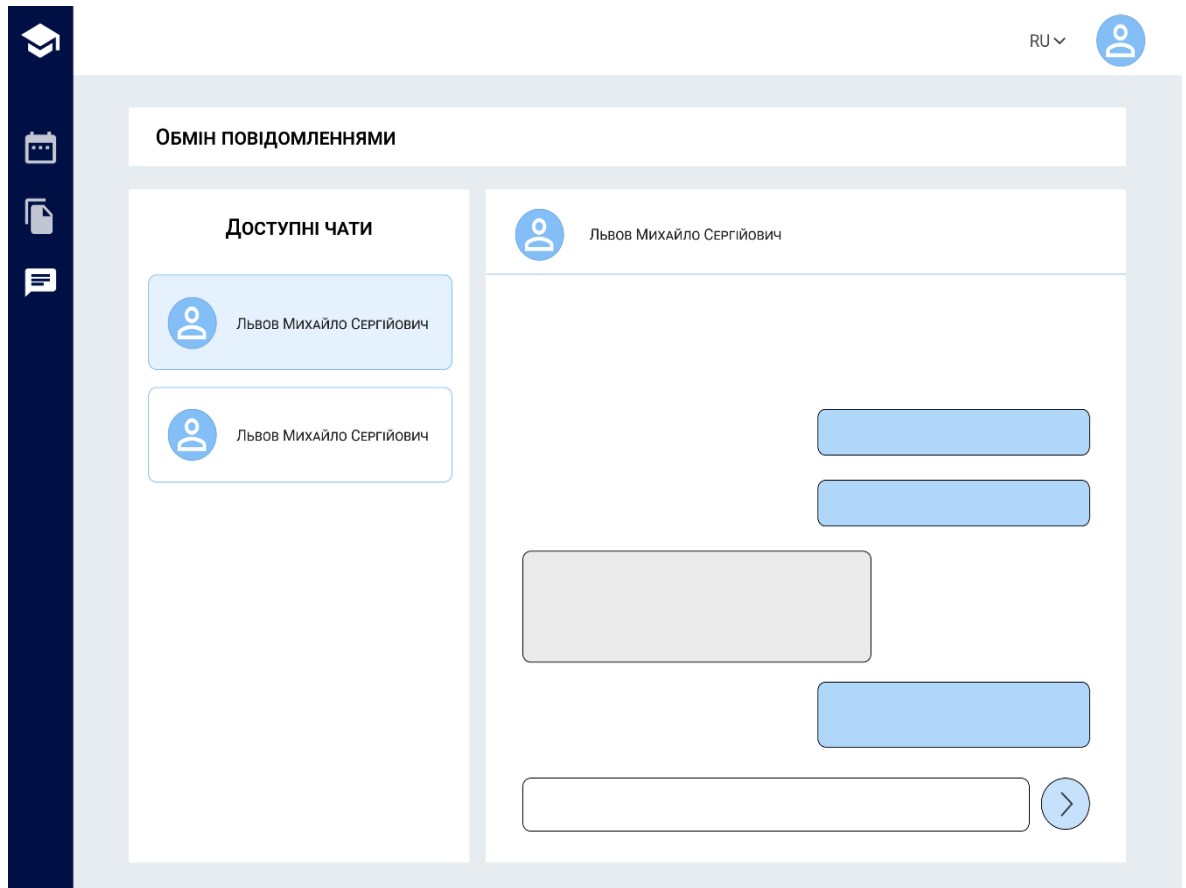


Рисунок 3.9 - Діалог з користувачами

Для того щоб написати повідомлення необхідно на сторінці необхідної людини натиснути кнопку «Надіслати повідомлення».

У системного адміністратора є доступ до адміністративної панелі. Завдяки чому він може додавати чи видаляти користувачів з системи (рис. 3.10).

Загальна функція адміністратора це відстеження безперебійної роботи web-додатку, пошук та виправлення можливих недоліків та оновлення системи за необхідністю.

За допомогою програмного модуля «Календар» завідувач кафедри має можливість виставляти різноманітні повідомлення про строки виконання магістрами їх робіт, чи про дату перед захисту (рис. 3.11).

Коли підходить час зазначений на календарі, наприклад закінчення прийому файлів магістерських робіт, то магістранти більше не можуть додавати, оновлювати чи видаляти файли своїх дипломних робіт, а

наукові керівники не можуть додавати ремарки до цих файлів.

Администрирование Django

Администрирование сайта

CABINET

Группы + Добавить ✎ Изменить

Специальности + Добавить ✎ Изменить

CHAT

Chats + Добавить ✎ Изменить

Messages + Добавить ✎ Изменить

MASTERFILES

Магістерські роботи + Добавить ✎ Изменить

Ремарки роботи + Добавить ✎ Изменить

Темы + Добавить ✎ Изменить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить ✎ Изменить

Пользователи + Добавить ✎ Изменить

Последние действия

Мои действия

- + Презентація до роботи
Магістерська робота
- + Ремарка до роботи
Ремарка роботи
- + Тестк дипломної
Магістерська робота

Рисунок 3.10 - Адміністративна панель

RU

КАЛЕНДАРЬ НА МІСЯЦ ▾

ПН	ВТ	СР	ЧТ	ПТ
	1	1	1	1
1	1	7	1	1
1	12	13	14	15

Сьогодні
07.09.2020

ОБРАТИ НАУКОВОГО КЕРІВНИКА І ТЕМУ ДИПЛОМНОЇ РОБОТИ

ДІЙСЬЕ ДО: 07.09.2020 20:00

УСТАНОВЧИЙ ДОКУМЕНТ: №1060 ПРО ЗАТВЕРДЖЕННЯ ТЕМАТИКИ КВАЛІФІКАЦІЙНИХ РОБІТ ДФН НАКАЗ ПРО ЗАТВЕРДЖЕННЯ ТЕМАТИКИ КВАЛІФІКАЦІЙНИХ РОБІТ ЗДОБУВАЧАМ ДРУГОГО (МАГІСТЕРСЬКОГО) РІВНЯ ВИЩОЇ ОСВІТИ

НЕЗАБАРОМ

12.07 - 15.07 ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ

12.07 - 15.07 ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ

12.07 - 15.07 ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ

НАЙВАЖЛИВІШЕ

12.07 - 15.07 <small>ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ</small>	12.07 - 15.07 <small>ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ</small>
12.07 - 15.07 <small>ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ</small>	12.07 - 15.07 <small>ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ</small>
12.07 - 15.07 <small>ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ</small>	12.07 - 15.07 <small>ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ</small>
12.07 - 15.07 <small>ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ</small>	12.07 - 15.07 <small>ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ</small>
12.07 - 15.07 <small>ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ</small>	12.07 - 15.07 <small>ВИБІР НАУКОВОГО КЕРІВНИКА І ТЕМУ МАГІСТЕРСЬКОЇ РОБОТИ</small>

Рисунок 3.11 - Календар

Цим модулем встановлюється строки додавання науковими керівниками тем магістерських робіт, можливість обирати ці роботи студентами магістрантами, дати перед захистів робіт, момент до якого магістри можуть додавати файли своїх робіт, дата фінального захисту та багато інших можливостей.

Після проходження студентами перед захистів їм за це ставляться оцінки які переносяться у даний модуль (рис. 3.12). Оцінки за перед захист доступні для усіх користувачів системи, а виставлятиме їх завідувач кафедри, який після перед захисту отримає усі оцінки.

ПІБ Студента	Оцінки за перед захист		Допуск/Не допуск
Студент	3	3	
Студент	4	4	
Студент	3	3	
Студент	4	4	
Студент	4	4	
Студент	3	3	
Студент	4	3	

Рисунок 3.12 - Оцінки студентів

Також існує модуль в якому науковий керівник виставляє оцінки тільки своєму магістранту за виконану на даний момент роботу. На підставі цих оцінок буде вирішуватися чи буде допущений магістрант до захисту своєї магістерської роботи чи ні.

Розроблений нами web-додаток дозволяє перевести процес написання магістерських робіт з очного режиму до дистанційного, коли для того щоб написати свою дипломну роботу студенту потрібен був лише доступ до інтернету.

ВИСНОВКИ

На сучасному етапі розвитку суспільства все частіше спостерігається тенденція на впровадження у всі сфери людської діяльності інформаційно-комунікаційних технологій.

2020-чи рік показав нам, що за умови наявності такої системи раніше можна було б вкрай полегшити процес написання магістерських робіт. Адже зараз вкрай складно а й інколи небезпечно очно зустрічатися науковим керівникам і їх магістрам.

Розроблена нами система повинна допомогти в близькому майбутньому в освітньому процесі більш якісніше та прозоріше виконувати контроль над виконанням студентами магістрантами своєї роботи.

Під час огляду існуючих систем для керування проєктами були порівняні та розглянуті найзатребуваніші з них. Було виявлено їх слабкі та сильні сторони.

Для розробки серверної частини проєкту були порівняні та проаналізовані основні інструменти сучасної web-розробки. На основі цих порівнянь було обрано мову програмування Python та фреймворк Django, за допомогою яких надалі і була розроблена наша система.

Для створення бази даних було обрано MySQL, через її швидкість та простоту у використанні.

Перед розробкою були складені основні функціональні та нефункціональні вимоги до майбутнього проєкту. А саме були створені UML-діаграми описані вимоги до користувачів та вимоги до майбутніх модулів проєкта. Також була спроектована база даних нашого web-додатку.

За результатами дипломної роботи були виконані наступні завдання:

1. Було досліджені та проаналізовані основні вимоги до систем керування проєктами

2. Були проаналізовані сучасні існуючі системи керування проектами
3. Були розглянуті та описані технології та інструменти для розробки web-додатку
4. Було обґрунтовано вибір технології для створення web-додатку
5. Було спроектовано web-додаток для підтримки процесу виконання магістерських робіт з програмної інженерії
6. Було реалізовано web-додаток підтримки процесу виконання магістерських робіт з програмної інженерії

Отже, розроблений нами web-додаток можна використовувати під час навчального процесу у закладах вищої освіти для більш чіткого та прозорого контролю виконання студентами магістрантами їх випускних магістерських робіт.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 7 лучших программ для менеджмента проектов. URL: <https://techrocks.ru/2018/11/13/best-free-project-management-software/>
2. Asana vs Trello: Какой инструмент лучше для управления проектами и задачами в компании? URL: <https://rocketsales.ru/blog-posts/asana-vs-trello-kakoy-instrument-luchshe-dlya-upravleniya-proektami-i-zadachami-v-kompanii>
3. Express – фреймворк веб-приложений Node.js. URL: <https://expressjs.com/ru/>
4. Flask против Django: почему Flask может быть лучше. URL: <https://python-scripts.com/flask-vs-django>
5. Freedcamp приложение таск-менеджер. Обзор Freedcamp. URL: <https://coba.tools/freedcamp>
6. Freedcamp. URL: <https://ru.qaz.wiki/wiki/Freedcamp>
7. Jira Software – Функции. URL: <https://www.atlassian.com/ru/software/jira/features>
8. Kyle. Чем Джанго лучше/хуже Фласка? URL: <https://devman.org/qna/18/chem-dzhango-luchshehuzhe-flaska/>
9. ProofHQ. URL: <https://www.workfront.com/proofhq>
10. Spring Framework и Spring Boot. URL: <https://topjava.ru/blog/spring-framework-vs-spring-boot-differences>
11. stannislav. 43 полезных сервиса для управления проектами. Без эпитетов. URL: <https://habr.com/ru/post/276873/>
12. Umbrella IT. Выбор лучшего Node.js фреймворка: Express, Коа или Sails. URL: <https://umbrellait.com/ru/blog/choosing-the-best-nodejs-framework/>
13. WebforMyself. Node.js - Фреймворк Express. URL: <https://webformyself.com/node-js-frejmvork-express/>
14. YANA GUSTI. Jira – инструмент для тестирования и управления

- проектами. URL: <https://geteasyqa.com/ru/blog/jira-test-case-project-management-tool/>
15. Александр Овсянников. База данных сайта MySQL: что это такое, преимущества перед другими СУБД. URL: <https://imajor.ru/razrabotka/hosting/baza-dannih-mysql>
 16. Библиотека программиста. ТОП-10 фреймворков для веб-разработки в 2019. URL: <https://proglib.io/p/web-frameworks-2019/>
 17. Битрикс24 – подробное описание системы и особенности использования. URL: <https://cetera.ru/about/articles/bitrix-24-detailed-description-of-system-and-features-of-use>
 18. Битрикс24: Что такое Битрикс24? URL: <https://www.bitrix24.ru/whatisthis/>
 19. В.В. Корнешук. Методичні рекомендації до написання випускної кваліфікаційної роботи магістра. Одеса, 2018. URL: <https://opu.ua/sites/default/files/publicFiles/op/disciplines/2944/mv231mag.pdf>
 20. Добивайтесь более высоких результатов с меньшими затратами – используйте Wrike. URL: <https://try.wrike.com/ru-branded/>
 21. Женя Орлов. Что такое Trello: преимущества и как пользоваться доской. URL: <https://baku.deveducation.com/blog/chto-takoe-trello-preimushchestva-i-kak-polzovatsya-doskoy/>
 22. Жулінська К. М., Кузьменко О. В., Подоляка О. І., Яровенко Г. М. Дипломна робота магістра : методичні рекомендації щодо підготовки, оформлення та захисту. Суми : УАБС, 2014. 56 с
 23. Илья. Фреймворк Laravel. URL: <https://unetway.com/blog/laravel-framework-review>
 24. Используя Trello, можно добиться более эффективной и слаженной работы команды. URL: <https://trello.com/>
 25. Лучшие веб-фреймворки Python. URL: <https://all-python.ru/osnovy/veb-frejmvorki.html#flask>

26. Львов М.С. Основные принципы построения педагогических программных средств поддержки практических занятий. *Управляющие системы и машины*. 2006. № 6. С. 70 - 75.
27. Магістерська робота як кваліфікаційне дослідження. URL: <http://www.info-library.com.ua/books-text-3040.html>
28. Мария Багулина. 8 основных языков для бэкенда. URL: <https://tproger.ru/translations/7-basic-languages-for-backend-development/>
29. Методичні вказівки щодо написання, оформлення та захисту кваліфікаційної магістерської роботи. *Сумський науковий університет*. Суми, 2019. URL: https://cyber.uabs.sumdu.edu.ua/images/department/cyber/stories/files/Metodychni_vkazivky.pdf
30. Обзор универсальной платформы для управления проектами | Wrike. URL: <https://www.wrike.com/ru/tour/>
31. Определение требований и ограничений к системе управления проектами. URL: <http://www.b-solutions.ru/serv/projects/pmo/services3-PMO-3.html>
32. Организация управления проектами. URL: <https://www.prostoy.ru/495.html>
33. Особенности Битрикс24. URL: <https://www.aniart.com.ua/blog/bitrix24-bitrix24-vnedrenie-obuchenie-i-podderzhka/>
34. Плюсы и минусы 5 основных инструментов управления проектами. URL: <https://dou.ua/forums/topic/30400/>
35. Плюсы и минусы Django. URL: <https://python-scripts.com/django-obzor>
36. Погружение в особенности | Начало работы с Trello. URL: <https://trello.com/ru/guide/feature-deep-dive.html>
37. Разработка и внедрение Системы управления проектами. URL: <http://www.tbsconsulting.ru/services/pm-menu/84.html>
38. Роман Яковенко. Обзор Asana. Индивидуальный таск-менеджер и рабочее пространство для команды. URL: <https://ya->

roman.com/blog/asana-obzor

39. Специфікація вимог до програмного забезпечення. URL: https://studopedia.ru/19_19058_spetsifikatsiya-vimog-do-programnogo-zabezpechennya.html
40. Співаковський О.В., Львов М.С., Кравцов Г.М. Педагогічні технології та педагогічно орієнтовані програмні системи: предметно-орієнтований підхід. *Комп'ютер у школі й сім'ї*. 2002. № 4 (22). С. 24-28.
41. Сравнение сервисов Earliz и Freedcamp. URL: <https://startpack.ru/compare/earliz/freedcamp>
42. Яна Руднева. Сравнение Jira и Trello: что выбрать для малого и среднего бизнеса. URL: <https://vc.ru/services/93252-sravnenie-jira-i-trello-что-vybrat-dlya-malogo-i-srednego-biznesa>

ДОДАТКИ

Додаток А

**КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО
ДЕРЖАВНОГО УНІВЕРСИТЕТУ**


Я, Савин Олег Андрійович,
учасник(ця) освітнього процесу Херсонського державного університету, УСВІДОМЛЮЮ, що академічна доброзесність – це фундаментальна етична цінність усієї академічної спільноти світу.

ЗАВЯЛЮЮ, що у своїй освітній і науковій діяльності **ЗОБОВ'ЯЗУЮСЯ**:

- дотримуватися:
 - вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
 - принципів та правил академічної доброзесності;
 - нульової толерантності до академічного плагіату;
 - моральних норм та правил етичної поведінки;
 - толерантного ставлення до інших;
 - дотримуватися високого рівня культури спілкування;
- надавати згоду на:
 - безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
 - оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
 - використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;
- самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;
 - надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;
 - не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;
 - своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;
 - не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;
 - підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;
 - поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;
 - не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;
 - відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науково-дослідницькі завдання;
 - запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;
 - не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;
 - не підроблювати документи;
 - не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;
 - не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;
 - не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;
 - не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;
 - не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;
 - не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягати власних корисних цілей;
 - не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.

УСВІДОМЛЮЮ, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброзесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброзесності.

05.05.2020
(дата)


(підпис)

Савин Олег
(ім'я, прізвище)