

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко
підпис

«__» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Вебзастосунок ведення розкладу занять

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.21810921

Студент _____ І. А. Пастушок
підпис
«__» _____ 2022 р.

Керівник канд. тех. наук, доцент. _____ Г. В. Горбань
підпис
«__» _____ 2022 р.

Консультант канд. тех. наук, доцент. _____ А. О. Алексєєва
підпис
«__» _____ 2022 р.

Миколаїв – 2022

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ АНАЛОГІЧНИХ ЗАСТОСУНКІВ ТА ОБҐРУНТУВАННЯ ЗАДАЧ РОЗРОБКИ.....	6
1.1 Порівняльний аналіз аналогів.....	6
1.2 Методи розробки вебзастосунків	13
1.3 Специфікація вимог до програмного забезпечення	16
Висновки до розділу 1	17
2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ	18
2.1 Бізнес-процес системи	18
2.2 Моделювання прототипу користувацького інтерфейсу	22
Висновки до розділу 2	30
3 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ.....	32
3.1 Сценарії використання системи.....	32
3.2 Діаграма класів	38
3.4 Діаграма послідовності.....	40
3.4 Технології.....	41
3.4.1 Webstorm.....	41
3.4.2 Vue	42
3.4.3 Vuetify	44
3.4.4 Firebase	45
Висновки до розділу 3	46
4 ПРОЄКТУВАННЯ І ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ.....	48
4.1 Налаштування середовища для розробки.....	48
4.3 Розробка бази даних.....	50

4.3 Розробка клієнтської частини	52
4.3.1 Головна сторінка	56
4.3.2 Сторінка «Календар»	57
4.3.3 Сторінка «Лист завдань»	60
4.3.4 Сторінка «Викладачі»	62
4.3.5 Сторінка «Нотатки»	64
Висновки до розділу 4	65
ВИСНОВКИ	66
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	67

ВСТУП

Однією з найважливіших проблем якісної організації навчального процесу у закладі вищої освіти є створення автоматизованого навчального розкладу.

Незалежно від об'єкта автоматизації, чи то викладацького складу чи адміністрації університету, в освітньому закладі такі системи впроваджують, переслідуючи кінцеву мету – підвищення якості освіти.

Об'єктом кваліфікаційної роботи є розклад предметів студента факультету комп'ютерних наук.

Предметом кваліфікаційної роботи технології створення вебзастосунків для відображення розкладу студента.

Актуальність теми кваліфікаційної роботи полягає в підвищенні ефективності роботи та економії часу студента. На даний час інтернет є невідкладною частиною життя людини. Можна просто взяти телефон чи сісти за комп'ютер, зайти в Google і знайти усю необхідну інформацію. Інформація оточує нас, а всі цікаві нам новини легко можна тримати у пристрої. Наприклад студент, якому потрібно завжди тримати власний розклад занять, роботи та дедлайни.

Метою кваліфікаційної роботи є підвищення ефективності та економії часу студента шляхом розробки вебзастосунку, який дозволяв би переглядати статистичну інформацію про викладачів, розклад, створення динамічної інформації про завдання предметів розкладу, його редагування та видалення.

Для досягнення визначеної мети необхідно вирішити такі завдання:

1. Дослідити об'єктну та предметну область.
2. Проаналізувати аналоги, що вже існують.
3. Спроектувати базу даних.
4. Розробити користувацький інтерфейс.
5. Побудувати та протестувати вебзастосунок.

Кафедра інженерії програмного забезпечення
Вебзастосунок веденя розкладу занять

1 АНАЛІЗ АНАЛОГІЧНИХ ЗАСТОСУНКІВ ТА ОБҐРУНТУВАННЯ ЗАДАЧ РОЗРОБКИ

1.1 Порівняльний аналіз аналогів

Для аналізу новочасного стану предметної області проведено дослідження існуючих вебзастосунків зі схожим застосуванням. Для дослідження обрано декілька найпоширеніших застосунків, таких як: iSpring, Moodle і Teachbase.

iSpring пропонує комплексне рішення для корпоративного онлайн-навчання. До нього входить навчальний портал iSpring Learn та конструктор курсів iSpring Suite. Рішення орієнтоване на швидкий запуск навчання [1].

iSpring Suite володіє великою кількістю можливостей та функцій. До основних функцій можна віднести:

- конструктор курсів. За допомогою iSpring Suite можна створювати навчальний контент: курси, тести, діалогові тренажери та скрінкасти;
- безлімітне сховище. У iSpring Learn можна завантажувати необмежену кількість файлів;
- інтеграція з іншими сервісами. iSpring Learn легко інтегрується з іншими системами клієнта через API. Наприклад, це може бути кадрова система чи корпоративний портал;
- швидкий запуск платформи. Платформу не потрібно довго налаштовувати. Достатньо зареєструватися, завантажити курси та запросити користувачів.

До основних мінусів можна віднести:

- не можна додати одразу велику кількість файлів, потрібно додавати кожен файл по одному;
- тести можна створювати в окремій програмі iSpring Suite - конструкторі курсів;

- перед тим як додавати користувачів у систему, треба створити організації та групи (якщо це не зробити, то потім не дуже зручно групувати користувачів);
- користувачі не отримують повідомлень про нові коментарі та відповіді в обговореннях;
- щоб додати користувачів до групи, як кожного користувача треба вводити вручну і не можна вибрати зі списку.

Інтерфейс застосунку можна побачити на рис. 1.1.

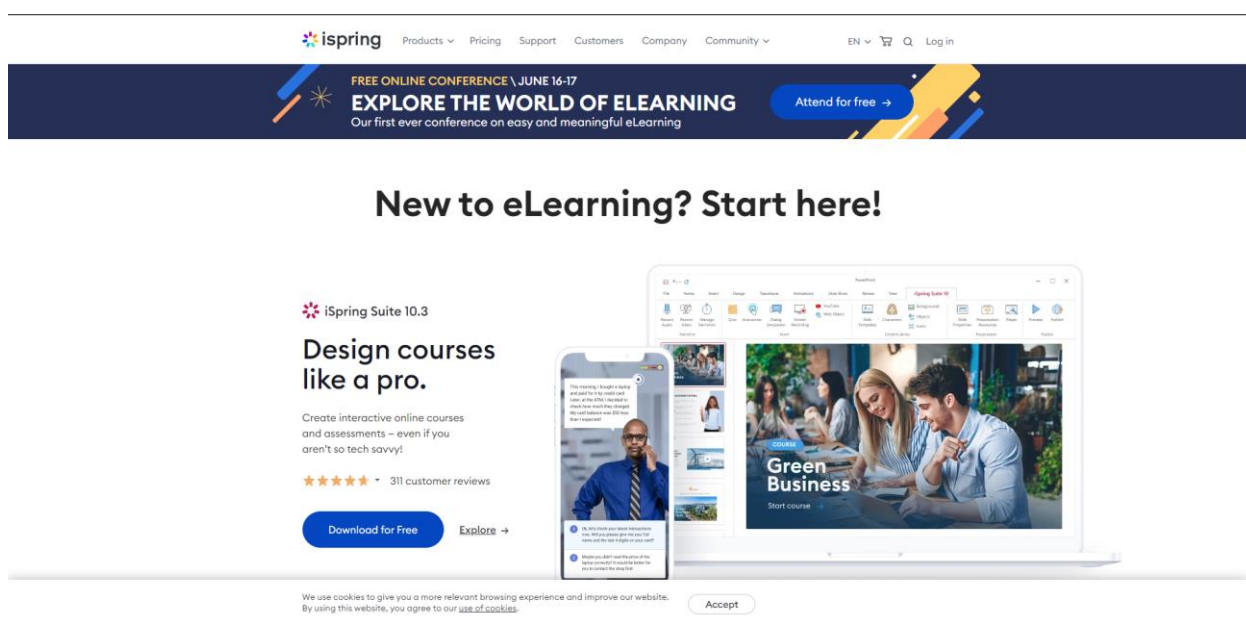


Рисунок 1.1 – Інтерфейс десктопного застосунку iSpring Suite

Розробляється вебзастосунок, адаптований під мобільний пристрій, тому буде розглянуто також зовнішній вигляд інтерфейсу під телефон.

Інтерфейс телефону для iSpring можна побачити на рис. 1.2.

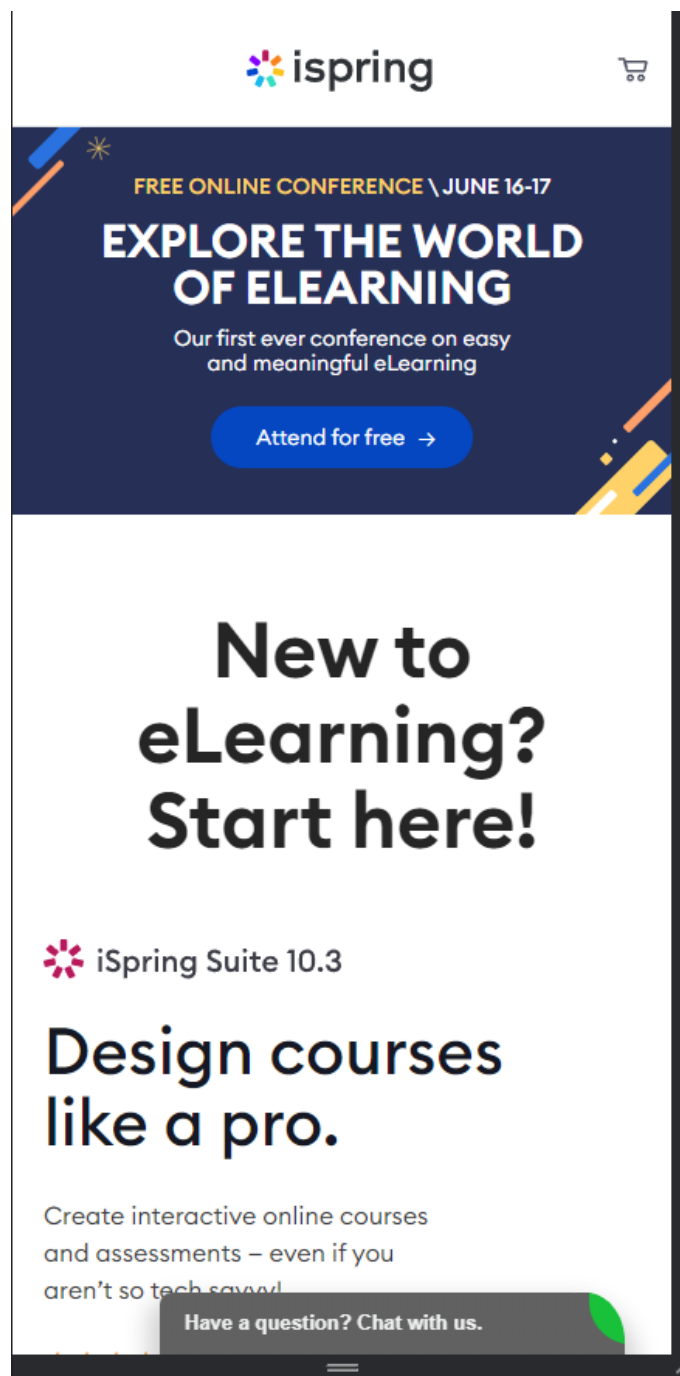


Рисунок 1.2 – Інтерфейс мобільного застосунку iSpring Suite

Другим застосунком, який досліджувався, є Moodle [2].

Moodle – безкоштовна система електронного навчання з відкритим кодом. За роки розробки навколо Moodle сформувалося сильне співтовариство, яке створює нові модулі (плагіни) для платформи. Модуль – це ZIP-архів, який при встановленні на платформу додає нові функції або

змінює дизайн. Сьогодні Moodle перекладена більш ніж 100 мовами і підтримує понад 1500 плагінів.

До основних переваг Moodle можна віднести:

- можливість її безкоштовного використання. При цьому функціональність системи дистанційного навчання Moodle не поступається комерційним аналогам;
- поширення системи у відкритому вихідному коді, що дозволяє адаптувати її під специфіку завдань, які мають бути вирішені за її допомогою;
- легкість інсталяції та оновлення системи при переході на нові версії.

Основним недоліком цього застосунку є відсутність професійної технічної підтримки.

Інтерфейс застосунку Moodle можна побачити на рис. 1.3 – 1.4.

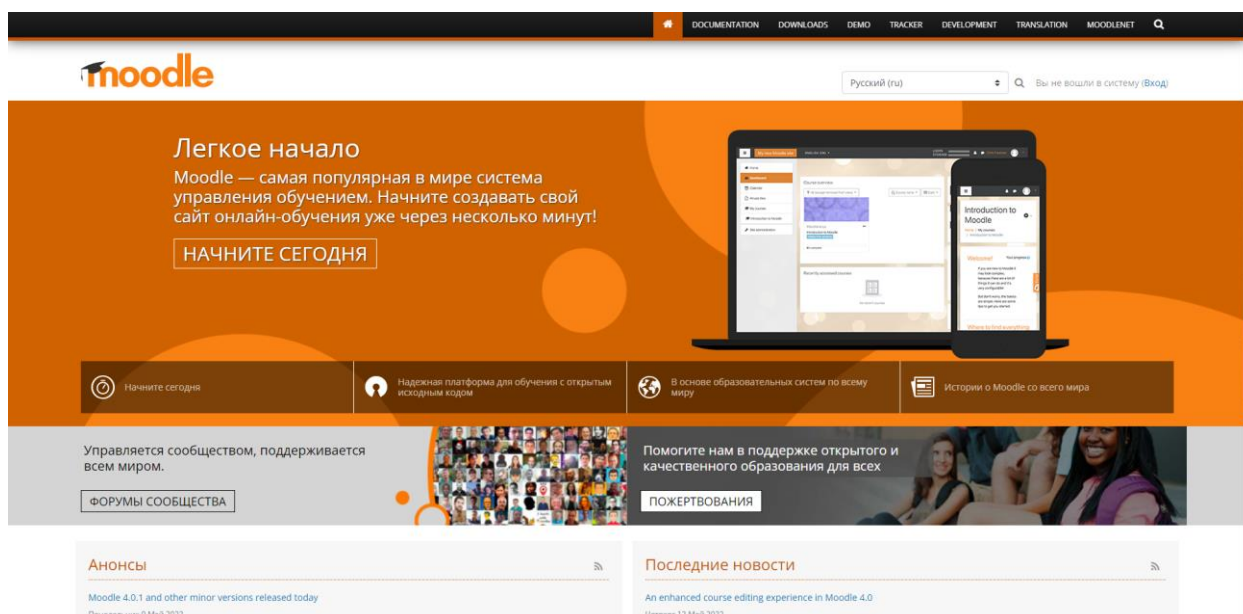


Рисунок 1.3 – Интерфейс десктопного застосунку Moodle

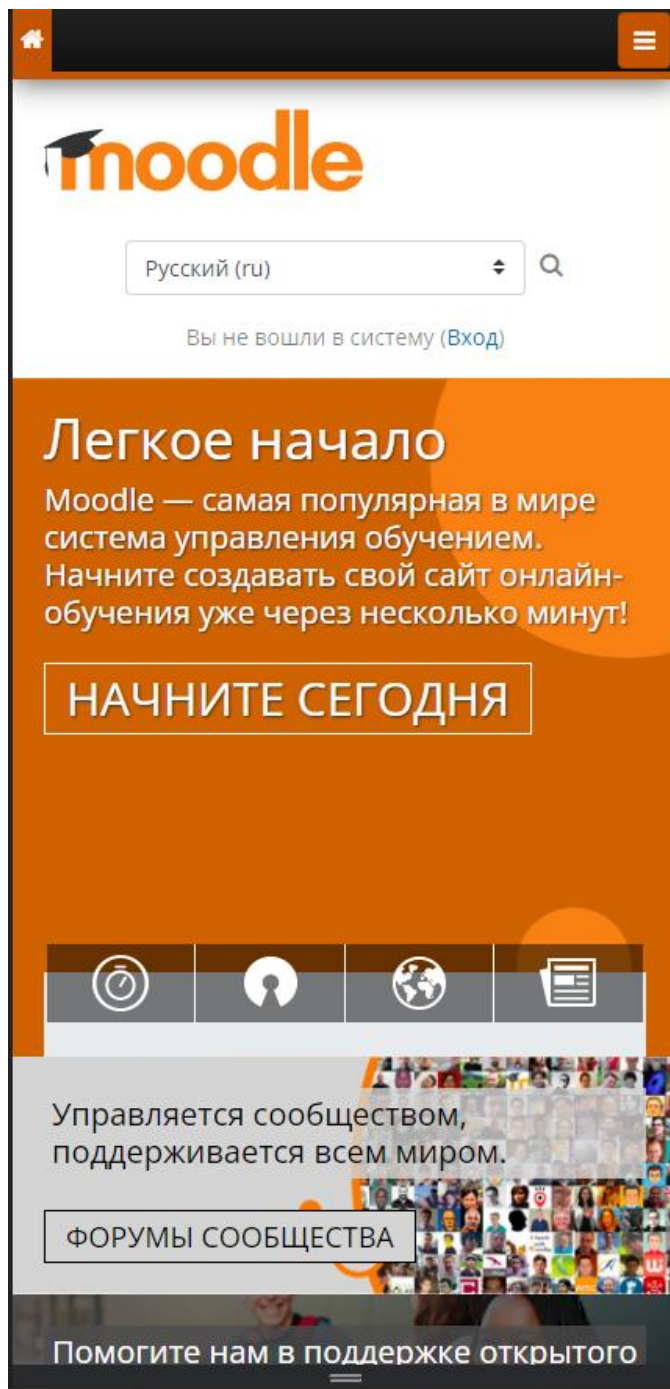


Рисунок 1.4 – Інтерфейс мобільного застосунку Moodle

Останнім вебсервісом для дослідження став Teachbase [3].

Teachbase – вебсервіс для дистанційного навчання. На платформі можна організувати як дистанційне навчання співробітників, так і продаж курсів.

Даний застосунок має наступні особливості:

- інтеграція здійснюється з іншими системами. Teachbase можна інтегрувати зі сторонніми CRM та платіжними системами;
- майданчик для вебінарів. Teachbase дозволяє проводити вебінари безпосередньо на платформі без сторонніх сервісів;
- наявність вбудованого редактора курсів – сторінка з курсом збирається на Tilda як звичайна посадкова сторінка;

Недоліками можна визначити наступні пункти:

- простий інтерфейс, але завжди інтуїтивно зрозуміло «куди натискати» (на одній сторінці може бути дві кнопки з назвою «Зберегти», деякі кнопки дуже маленькі і без назви);
- при створенні тестів не можна налаштувати дизайн питань – можна тільки додати один файл, при цьому додана картинка або відео відображається дрібно і відкривається поверх вікна з питанням;

Інтерфейс застосунку Teachbase можна побачити на рис. 1.5 – 1.6.

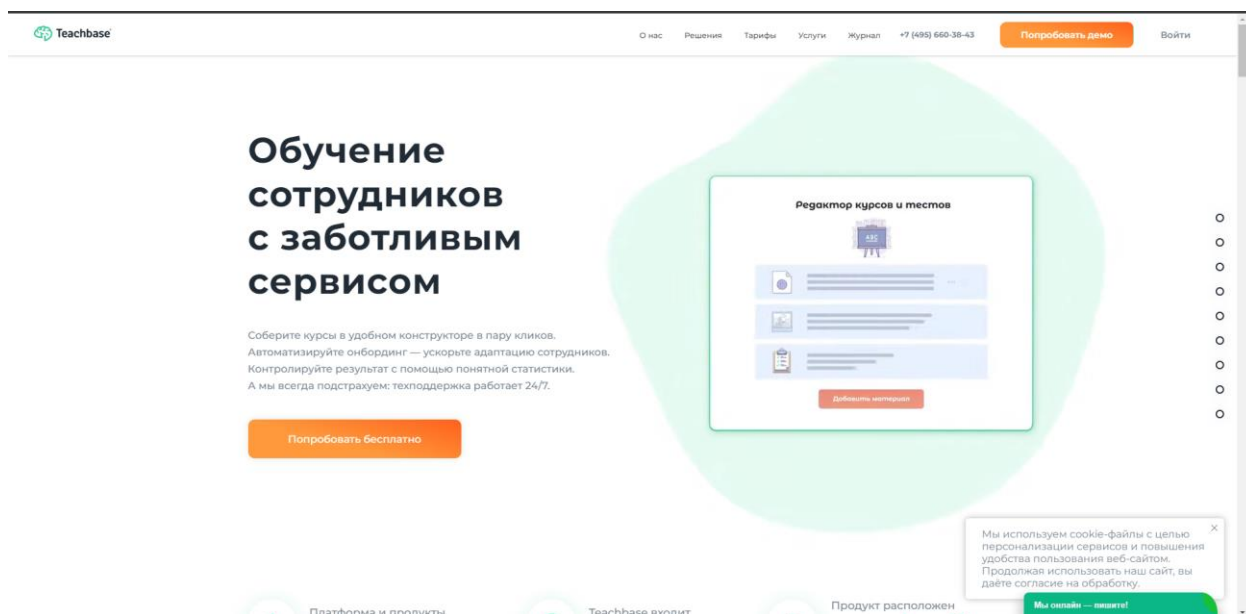


Рисунок 1.5 – Интерфейс десктопного застосунку Teachbase

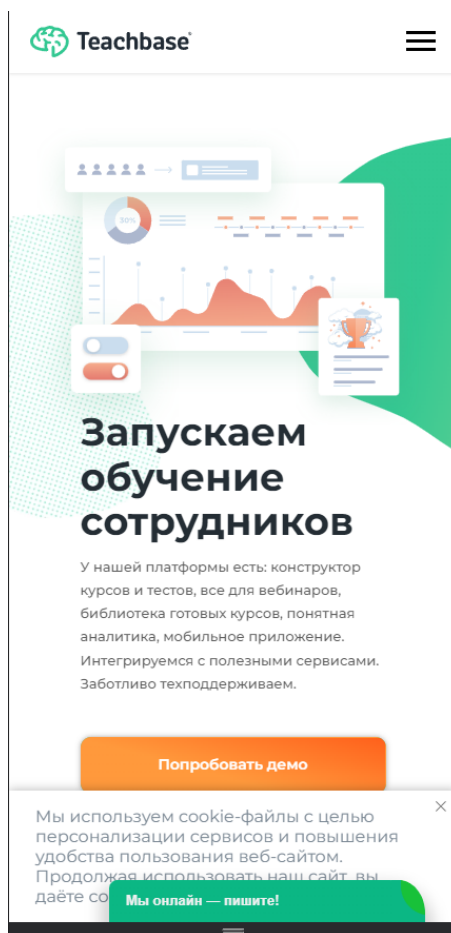


Рисунок 1.6 – Інтерфейс мобільного застосунку Teachbase

В табл. 1.1 наведено порівняння систем.

Таблиця 1.1 – Порівняння вебзастосунків

Показники	iSpring Suite	Moodle	Teachbase
Створення контенту	+	+	+
Продаж курсів	-	+	+
Мобільне навчання	+	+	+
Хмарна версія	+	+	+
Коробкова версія	+	+	-
Сумарний показник	4	5	4

На основі таблиці можна побачити, що фаворитом є Moodle. Він за своєю метою схожий на майбутній вебзастосунок, тому він буде невеликим

орієнтиром при створенні, проте головним фактором, що буде виділяти проєкт – те, що він буде таким собі «Органайзером» для студента.

1.2 Методи розробки вебзастосунків

При розробці сайтів, програм та інших програмних продуктів використовується достаньо багато різних методологій. Яку з них вибрати, багато в чому залежить від конкретної ситуації – особливостей проєкту, бюджету, термінів, особистих уподобань розробника тощо.

«Водоспад» або каскадна модель (Waterfall Model) [4] – класична методологія, що використовується «з давніх-давен». Є суворо послідовне виконання всіх стадій розробки. Іншими словами, нова стадія не починається доти, доки не буде повністю закінчена попередня.

Каскадна модель дуже зручна для управління проєктом, оскільки процес розробки легко відстежується. Це дає можливість жорсткого контролю за процесом розробки, що дозволяє досить точно заздалегідь визначити терміни закінчення та загальну вартість проєкту.

Проте така жорсткість має негативний бік. «Водоспад» добре підходить для проєктів із гранично чіткими вимогами та заздалегідь продуманими способами реалізації. Але якщо в технічному завданні є туманні моменти, які можна трактувати двозначно, каскадна методологія стає вкрай незручною. Водоспад не передбачає можливості відкотити розробку на одну-дві стадії назад, відсутня можливість протестувати окремий аспект до повного закінчення розробки. З цієї причини неможливо вносити зміни, поправки та коригування до вже зробленої частини роботи, або внесення таких поправок різко підвищує вартість проєкту.

Таким чином, каскадна методологія підходить виключно для тих проєктів, де вимоги в технічному завданні гранично точні, зрозумілі та зафіксовані на папері, а будь-які різночитання чи непорозуміння відсутні.

Також дуже бажано використати цю методологію лише для відносно невеликих проєктів.

V-подібна модель (V-Model) [5] - ця модель має загалом самі принципи послідовної «крок-за-кроком» розробки, як і каскадна, але відрізняється від неї одним важливим моментом — кожному етапі здійснюється тестування готової частини проєкту.

V-подібну модель зазвичай використовують при розробці програмного забезпечення, призначеного для важливих систем, де неприпустимі перебої у роботі. Наприклад, під час створення програмного забезпечення для моніторингового медичного устаткування, різних систем безпеки тощо. Словом скрізь, де помилки та недоліки у програмному продукті можуть мати серйозні наслідки.

Загалом можна впевнено говорити про перевагу V-подібної моделі для тих проєктів, які вимагають ретельного тестування всіх аспектів від зручності інтерфейсу до системної стабільності та відсутності вразливостей для зовнішнього втручання.

Інкрементна модель (Incremental Model) [6]. Дана методологія використовується для проєктів, що передбачають кілька варіантів (складання) готового продукту. Найчастіше розробка ведеться кількома циклами, тобто у результаті виходить свого роду "мульти-водоспад". При цьому в кожному циклі є свої етапи та створювані модулі. Для кожного модуля передбачено власні етапи уточнення вимог, створення проєкту, кодування, тестування тощо.

Інкрементна модель передбачає особливу послідовність створення складання: спочатку реалізується основний проєкт (базове складання), потім на її основі створюються нові складання з новими функціями, званими «інкрементами».

Розробка за інкрементною моделлю хороша для тих проєктів, у яких чіткі та зрозумілі не лише базові вимоги до системи, а й запити на внесення

змін також зрозумілі, а самі зміни легко реалізуються. При цьому цілком допускається, що окремі функції та нові складання можуть допрацьовуватися вже після впровадження на практиці базового складання.

Швидка розробка програм або «RAD Model» є різновидом описаної вище інкрементної моделі. Ключовою відмінністю є те, що компоненти проекту (модулі) або різні збирання розробляються не по черзі однією командою, а паралельно кількома командами. У разі жорстко лімітованого часу створені одночасно модулі збирають у єдиний робочий зразок. У результаті вдається надати замовнику робочу систему в стислі терміни.

Важливою умовою застосування даної методології є кілька висококваліфікованих команд фахівців. Наслідком такого підходу стають високі витрати на оплату послуг великої кількості задіяних спеців та робочих інструментів, які вони використовують.

Гнучка модель розробки (Agile Model) [7]. Ключова особливість даної методології полягає в максимальній прозорості процесу розробки для замовника, який має можливість відстежувати буквально кожну ітерацію і схвалювати її або вимагати переробки. Таким чином, повністю виключається найменша можливість зробити зовсім не те, чого хотів клієнт.

Очевидним недоліком гнучкої моделі є складність попередньої оцінки трудовитрат та вартості проекту. Однак в умовах відсутності чітких вимог і невиразного ТЗ, коли замовник сам дуже неявно розуміє, що йому потрібно, гнучка модель є єдиною можливою для використання.

Важливим атрибутом гнучкої методології є проведення нетривалих щоденних зустрічей, які називаються «Scrum», і навіть регулярних зборів щотижня чи рідше, іменованих «Sprint».

Методологія добре себе демонструє розробки великих проектів, чи проектів, які необхідно постійно адаптувати до мінливим умовам ринку.

1.3 Специфікація вимог до програмного забезпечення

На основі аналізу предметної області та дослідження існуючих вебзастосунків сформовані вимоги до застосунку, що розроблюється, відповідно до мети. Сформовано Специфікацію вимог до вебзастосунку, що представляє собою опис поведінки застосунку, що розроблюється.

ПРИЗНАЧЕННЯ ПРОЄКТУ

Застосунок буде являтися вебзастосунком та частково органайзером для перегляду розкладу, та створення власних списків даних, зв'язаних с розкладом чи університетським життям студента.

ФУНКЦІЇ СИСТЕМИ

Функція системи 1:

Перегляд розкладу предметів.

Функціональні вимоги

Застосунок повинен виводити на головну сторінку розклад, який буде створений студентом.

Функція системи 2:

Створення, редагування та видалення даних на сторінці «Календар» та «Лист задач».

Функціональні вимоги

На сторінці «Календар» має бути змога створити та видалення даних, які будуть відображатися в календарі, а на сторінці «Лист задач», будуть відображатися список даних, що було створено на попередній сторінці, зі змогою їх видалити.

Функція системи 3:

Створення, та видалення записів на сторінці «Примітки»

Функціональні вимоги

На сторінці «Примітки» має бути змога створити, закреслити та видалити дані, які будуть відображатися на сторінці.

ВИМОГО ДО ПРОГРАМНОГО ЗЕБЕЗПЕЧЕННЯ

Мова і технологія розробки

Для розробки застосунку використано середовище розробки WebStorm та JavaScript-фреймворк Vue.js з UI бібліотекою Vuetify.

Вимоги:

- Доступ до інтернету;
- Мобільний телефон чи комп'ютер

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Вебзастосунок повинно бути адаптоване під мобільний пристрій.

Практичність

- Застосунок має бути зручним у використанні;
- Дизайн застосунку має бути максимально зручним для використання.

Висновки до розділу 1

У результаті проведеного аналізу визначено, що створення вебзастосунку для введення власного розкладу є актуальним, тому що дане завдання допоможе студентам зі створенням та редагуванням розкладу, завдань предметів семестру.

Проведено дослідження схожих систем та вебзастосунків та визначено необхідність створення даного застосунку. На основі проведеного дослідження предметної області сформовано вимоги до програмного забезпечення, що розроблюється.

2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ

2.1 Бізнес-процес системи

VRwin – дуже якісний інструмент моделювання, який був створений фірмою Computer Associates Technologies [8]. Продукт використовується для аналізу, запису до документів та реорганізації різних та складних бізнес-процесів. Модель, створена в програмі VRwin, дозволяє чітко документувати різні фрагменти діяльності - дії, які потрібні зробити, способи їх здійснення, необхідні для цього ресурси та ін. Таким чином формується цілісна картина діяльності підприємства - від моделей організації роботи в маленьких відділах до складних ієрархічних структур. При розробці або закупівлі програмного забезпечення моделі бізнес-процесів є чудовим засобом документування потреб, допомагаючи забезпечити високу ефективність інвестицій у сферу ІТ. У руках системних аналітиків і розробників VRwin – ще й потужний засіб моделювання процесів при створенні корпоративних інформаційних систем (КІС).

Моделі VRwin дають основу для осмислення бізнес-процесів та оцінки впливу тих чи інших подій, а також описують взаємодію процесів та потоків інформації в організації. Неefективна, високовитратна чи надмірна діяльність може бути легко виявлена і, отже, удосконалена, змінена чи усунена відповідно до спільних цілей організації.

Зовнішні обставини часто змушують вносити зміни у діяльність організації. Наслідки цих змін мають бути ретельно вивчені та осмислені перед тим, як система буде перероблена з їх врахуванням. VRwin може допомогти користувачеві протягом усього циклу, надавши можливість оптимізувати бізнес-процес, якого торкнуться ці зміни.

VRwin дозволяє:

- забезпечити ефективність операцій, розглядаючи поточні бізнес-операції через потужні інструменти моделювання;

- удосконалювати бізнес-процеси, формулюючи та визначаючи альтернативні реакції на впливи ринку;
- швидко виключати непродуктивні операції, легко та інтуїтивно зіставляючи операційні зміни. Неефективні, неекономічні або надмірні операції можуть бути легко виявлені і, отже, покращені, змінені або зовсім виключені відповідно до цілей компанії.

BPwin (тепер AllFusion Process Modeler) - програмний продукт у сфері реалізації засобів CASE-технологій. Дозволяє проводити опис, аналіз та моделювання бізнес-процесів. Займає одне з провідних місць у своєму сегменті ринку. В даний час випускається компанією Computer Associates. Розповсюджується на комерційній основі.

Включає три стандартні методології: IDEF0 (функціональне моделювання), DFD (моделювання потоків даних) та IDEF3 (моделювання потоків робіт). Ці методології по-своєму є унікальними. Кожна з них може бути виконана окремо за допомогою BPwin, але їхня сукупність укладена в модель дає аналітику повну картину предметної області клієнта.

Інтерфейс програми можна побачити на рисунку 2.1.

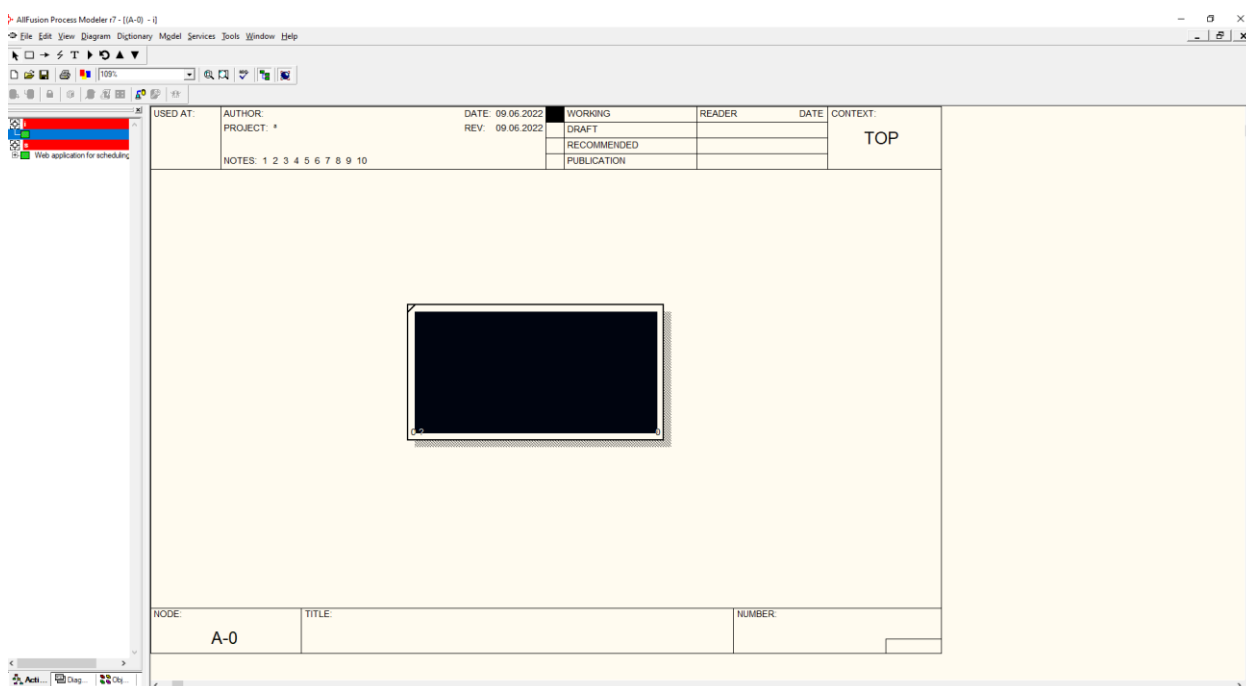


Рисунок 2.1 – Інтерфейс BPwin

Ключовим бізнес-процесом системи, що проєктується, вважається процес створення та збереження результатів роботи студента.

Суб'єктом бізнес-процесу є «Вебзастосунок введення розкладу занять».

На вхід надходять дані:

- список вчителів;
- завдання;
- записки;
- розклад.

Механізмами виконання процесу вважається:

- студент.

Керуванням є:

- створення та перегляд завдання для розкладу.

На виході отримуємо:

- виведення обробленої інформації.

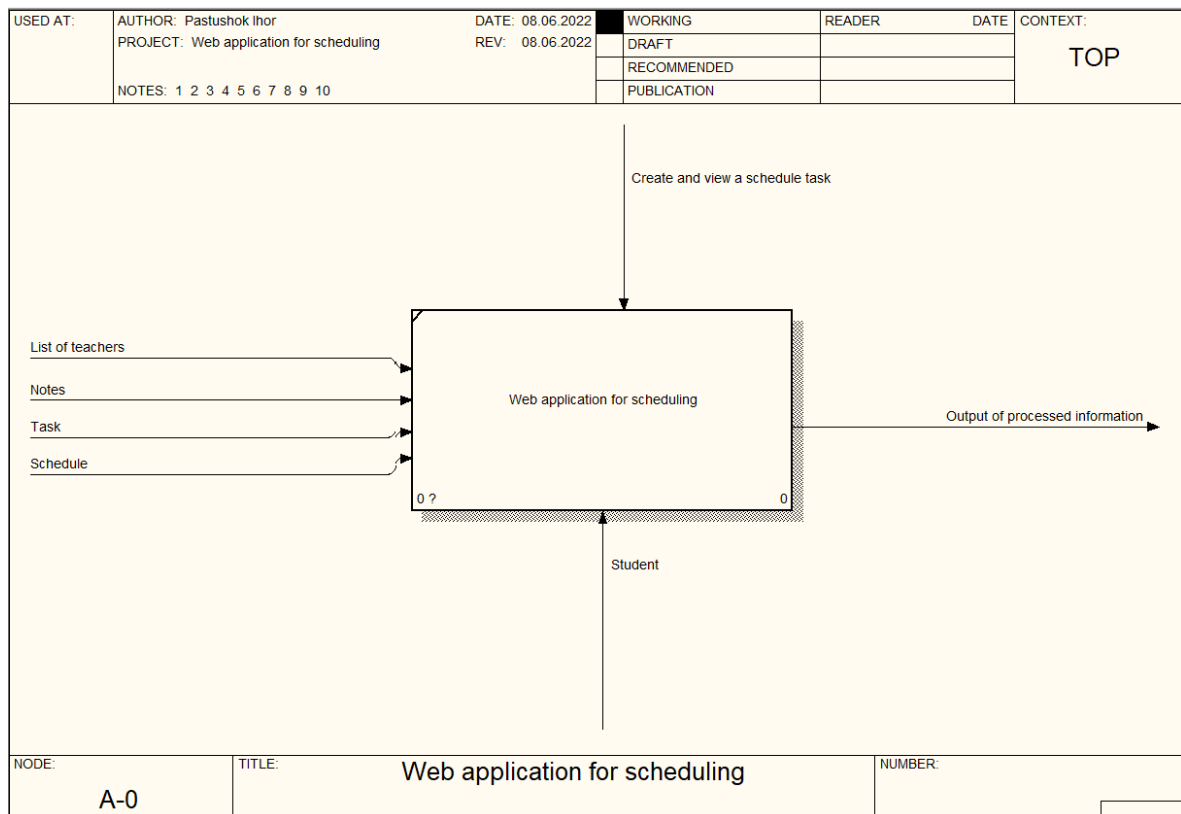


Рисунок 2.2 – Діаграма IDEF0 «Вебзастосунок ведення розкладу занять»

Провідними бізнес-функціями процесу «створення та збереження результатів роботи студента» стануть:

1. створення бази даних з викладачами;
2. створення бази даних з завданнями;
3. створення бази даних з записками.
4. створення бази даних з розкладом.
5. формування запиту.

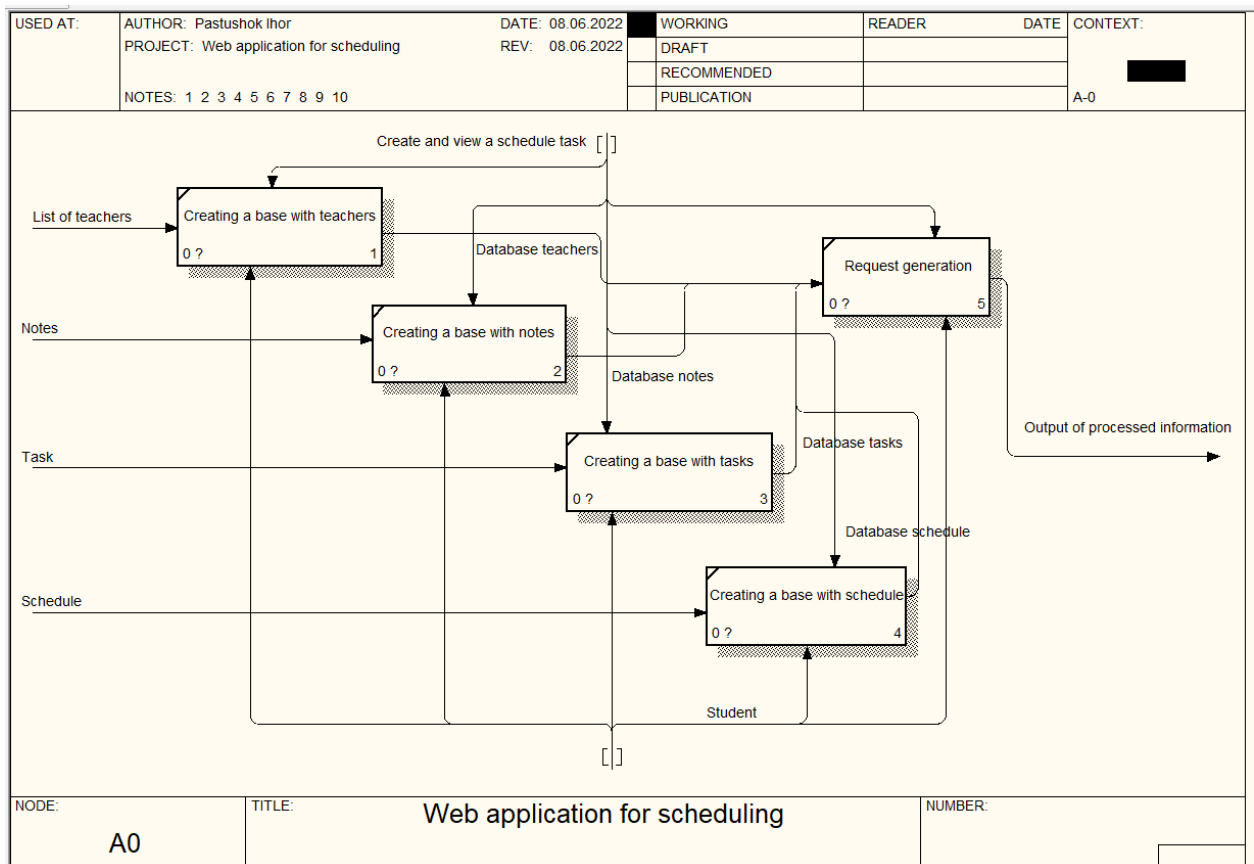


Рисунок 2.3 – Повна діаграма IDEF0 «Вебзастосунок введення розкладу занять»

2.2 Моделювання прототипу користувацького інтерфейсу

Моделювання зовнішнього вигляду є дуже важливою частиною в проектуванні застосунку. Зручний і інтуїтивно зрозумілий користувацький дуже спрощує роботу із застосунком. Розташування елементів повинне бути зручним і інтуїтивно зрозумілим будь-якому користувачу.

Для розробки прототипів користувацького інтерфейсу обрано інструмент Figma. Figma простий та потужний інструмент, який має необхідний функціонал для розробки прототипу дизайну мобільного застосунку. Інтерфейс інструменту можна побачити на рис. 2.4.

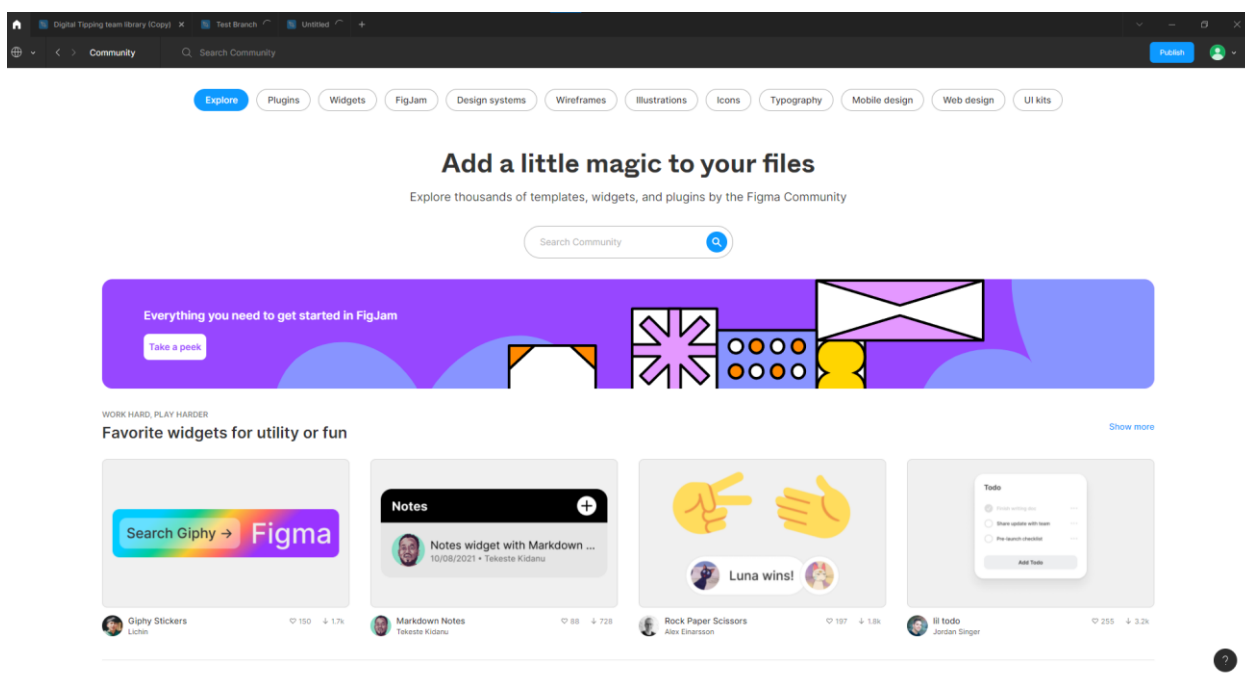


Рисунок 2.4 – Інтерфейс Figma

В першу чергу створено макет інтерфейсу основної сторінки застосунку, а саме сторінка «Розклад» (рис. 2.5 – 2.8). На головній сторінці виводиться розклад над межею та під межею. На рисунку 2.7 створений інтерфейс сторінки з навігаційним баром.

Кафедра інженерії програмного забезпечення
Вебзастосунок ведення розкладу занять

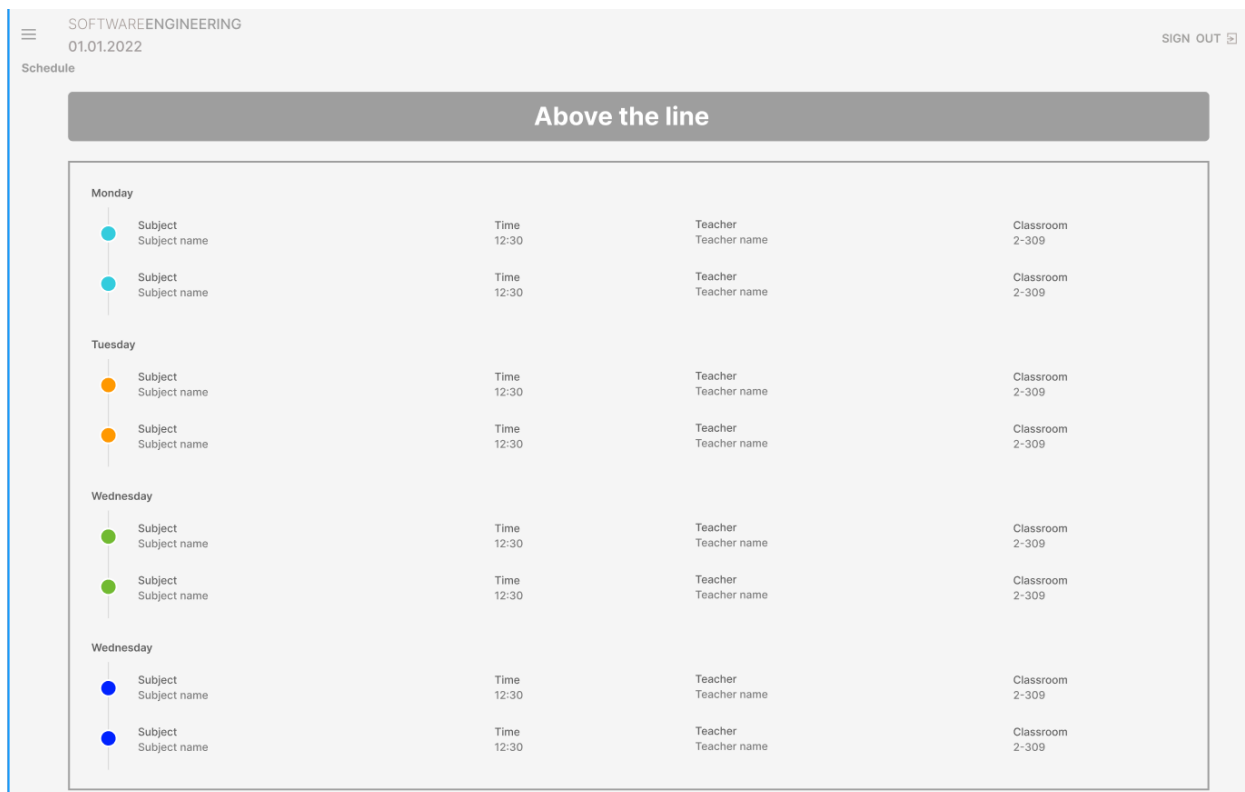


Рисунок 2.5 – Розклад над межею

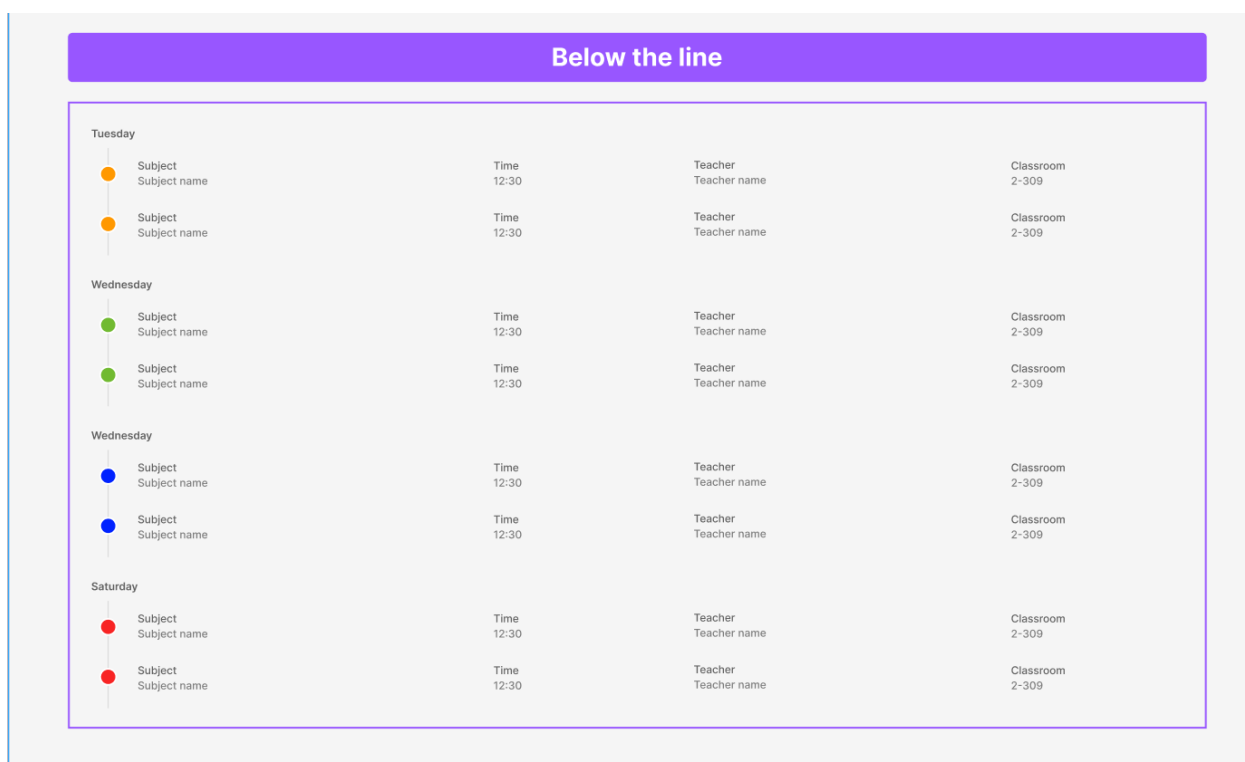


Рисунок 2.6 – Розклад під межею

Кафедра інженерії програмного забезпечення
Вебзастосунок ведення розкладу занять

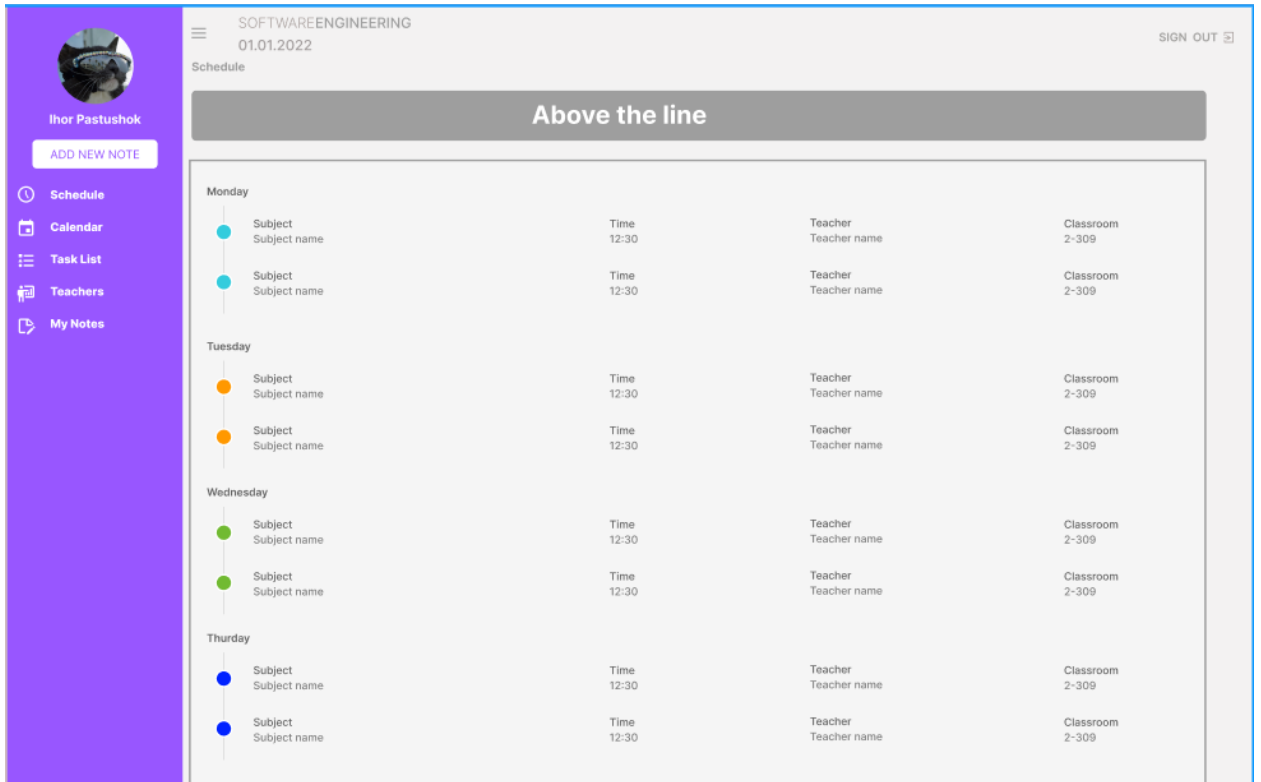


Рисунок 2.7 – Сторінка з навігаційним баром

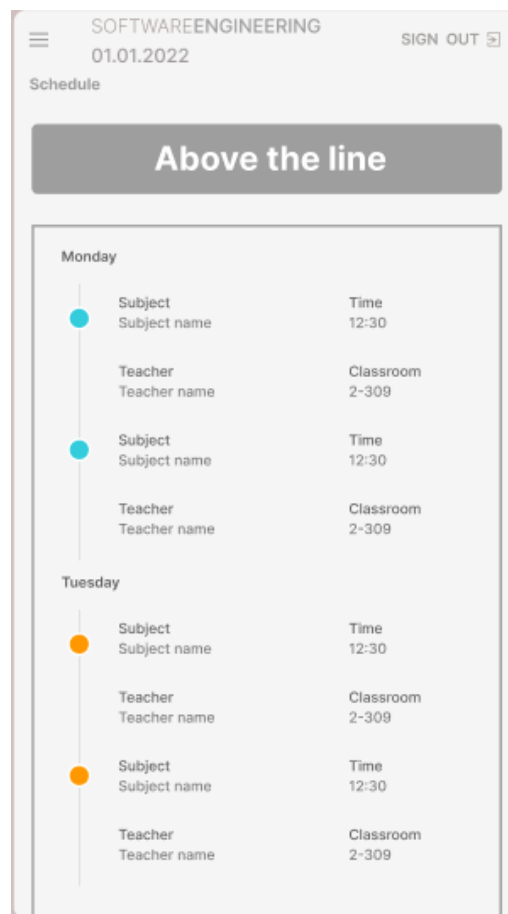


Рисунок 2.8 – Інтерфейс для мобільного пристрою

Друга сторінка – сторінка «Календар» (рис. 2.9 – 2.10). На ній можна побачити календар, у якому будуть показуватися роботи, які студент буде записувати. При створенні завдання, можна ввести назву, опис, вибрати дату та колір, яким буде помічатися завдання у календарі.

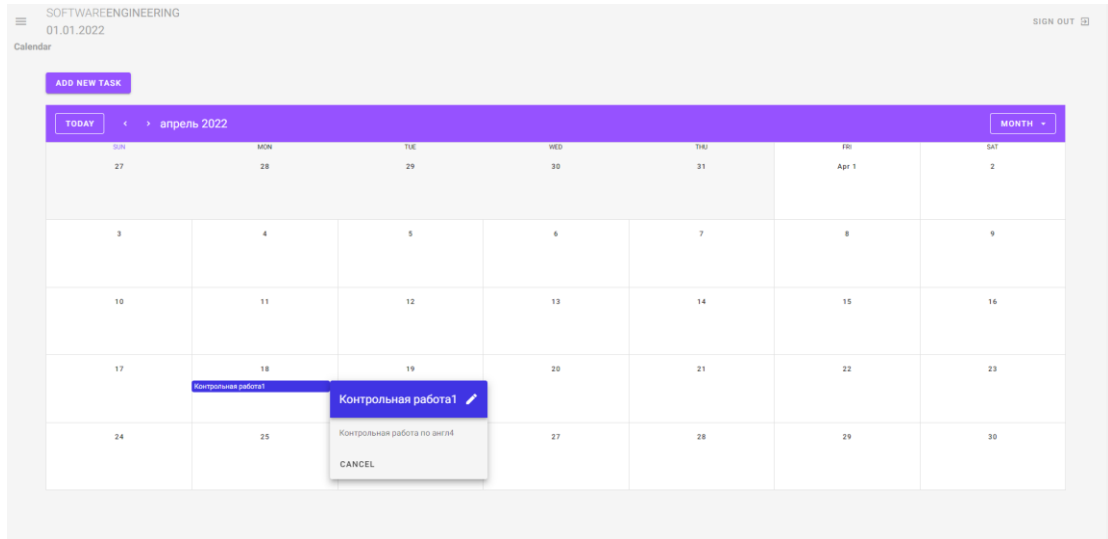


Рисунок 2.9 – Сторінка «Календар»

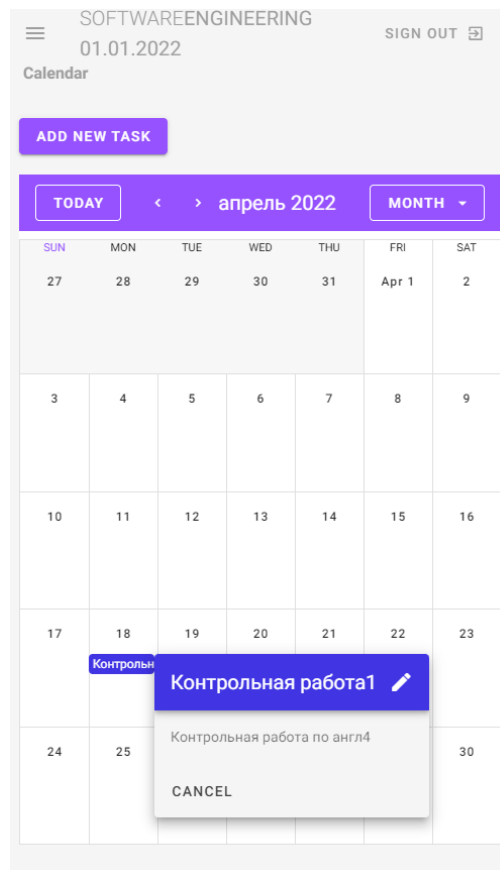


Рисунок 2.10 – Інтерфейс для мобільного пристрою

Сторінка «Лист задач» іде третьою у черзі (рис. 2.11 – 2.12). Вона виводить у вигляді списку усі завдання, які можна відкрити, щоб переглянути опис, та видалити його. Видалення працює і для календаря. Завдання можна фільтрувати за назвою або датою.

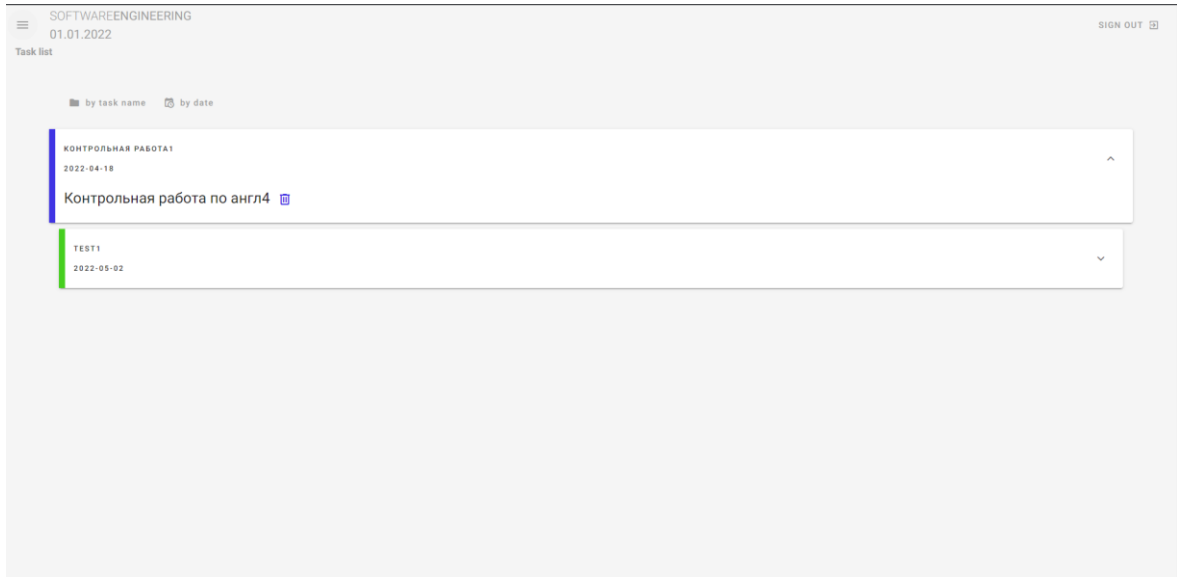


Рисунок 2.11 – Сторінка «Лист задач»

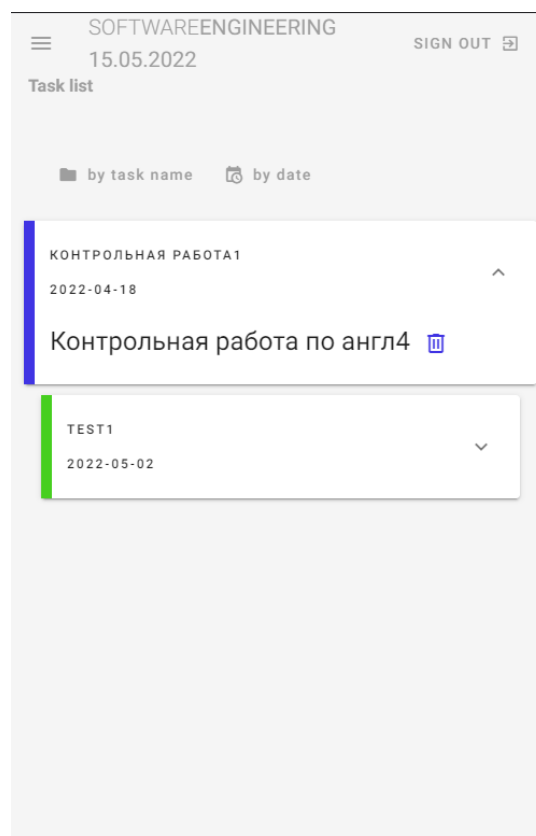


Рисунок 2.12 – Інтерфейс для мобільного пристрою

Четверта сторінка – сторінка «Викладачі» (рис. 2.13 – 2.15). На ній виводиться список у вигляді карточок, де знаходиться інформація про викладачів факультету комп'ютерних наук. При натисканні на кнопку, має відкриватися модальне вікно, де буде детальна інформація про викладача.

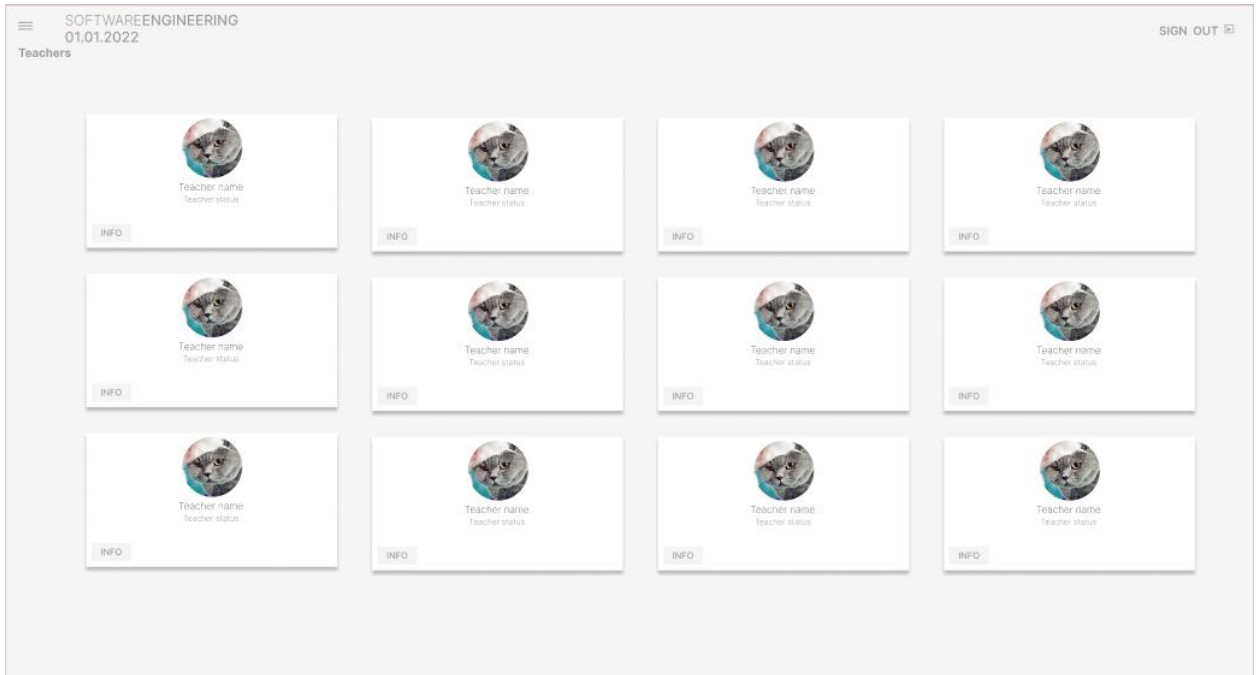


Рисунок 2.13 – Сторінка «Викладачі»

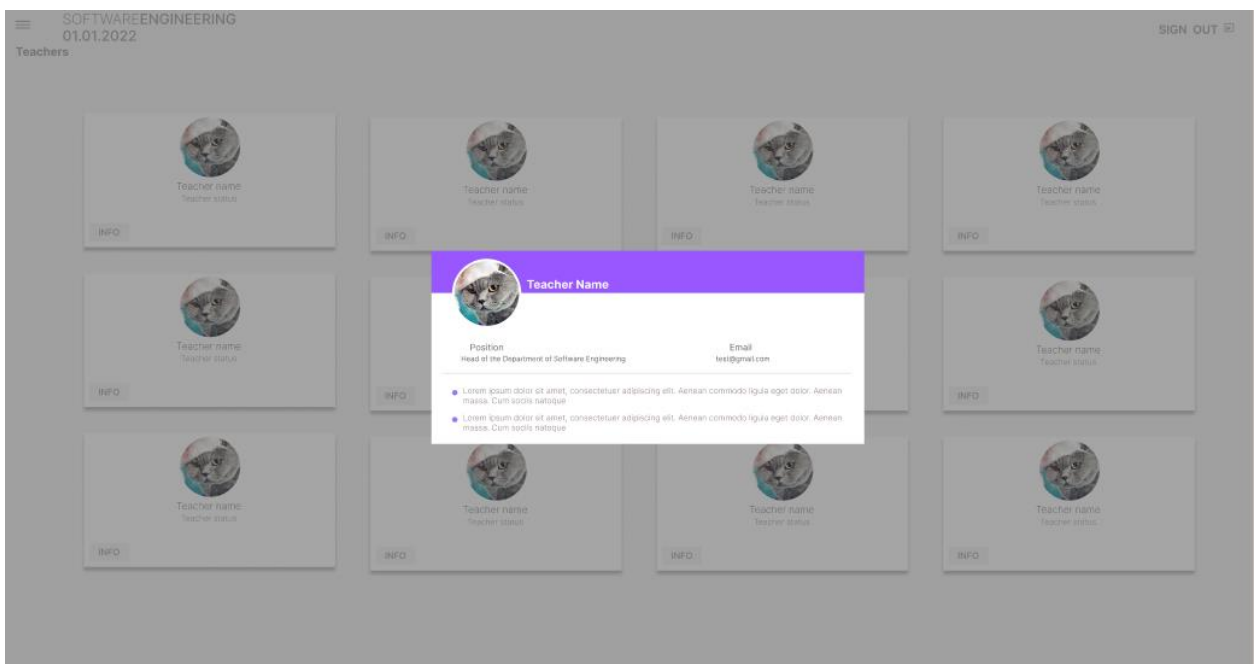


Рисунок 2.14 – Модальне вікно обраного викладача

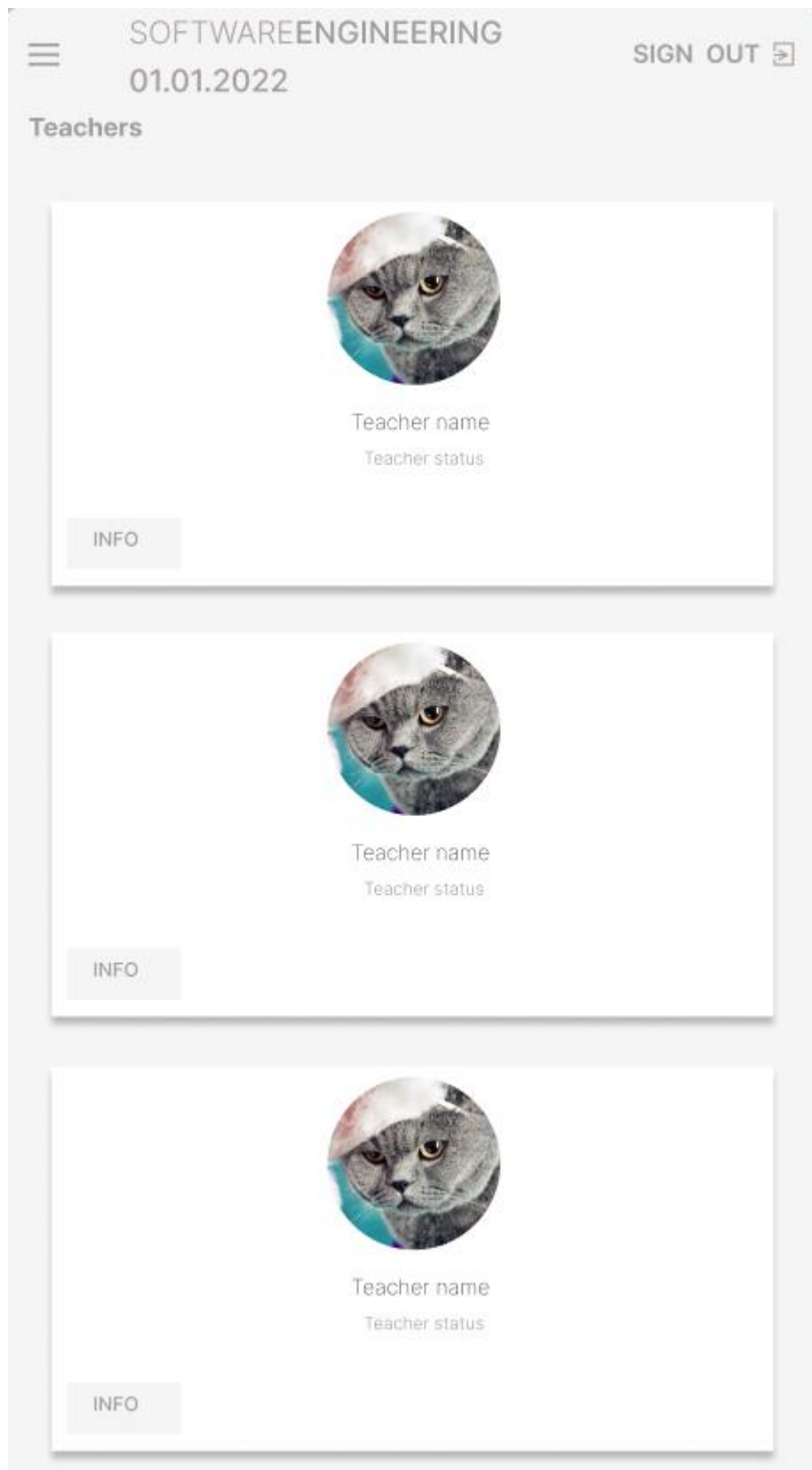


Рисунок 2.15 – Інтерфейс для мобільного пристрою

Остання сторінка має назву «Примітки». Результат створеного макету представлено на рис. 2.16 – 2.17. Ця сторінка буде записувати примітки, якщо студенту знадобиться швидко додати якусь інформацію.

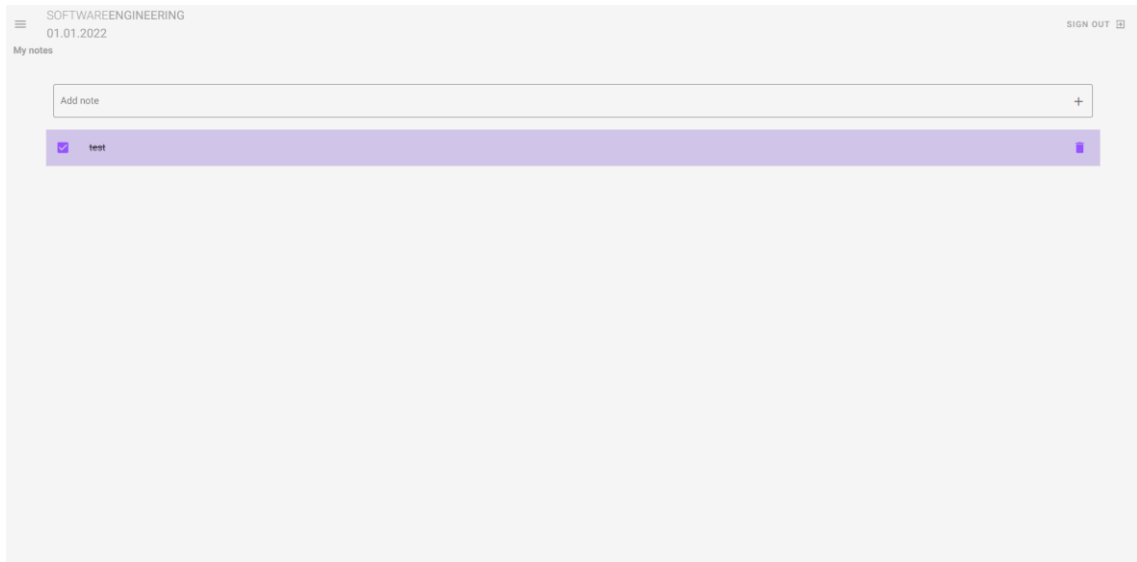


Рисунок 2.16 – Інтерфейс для мобільного пристрою

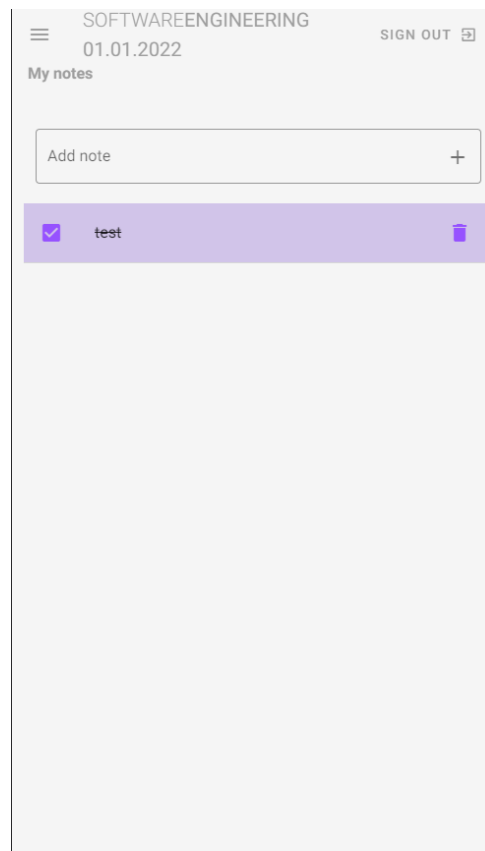


Рисунок 2.17 – Інтерфейс для мобільного пристрою

На навігаційному барі є кнопка «Додати нову записку», при натиску якої, визивається модальне вікно. Макет створено на рис. 2.18.

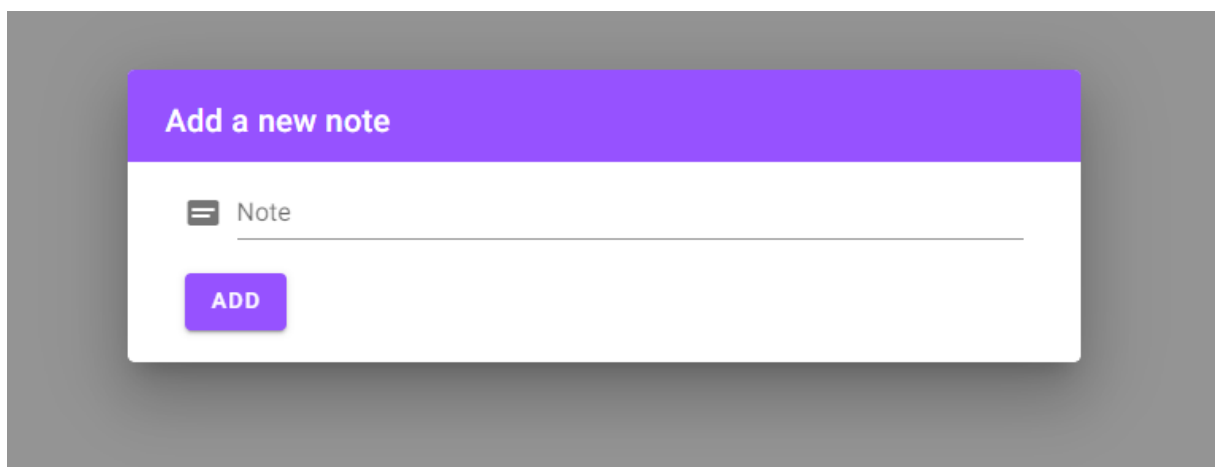


Рисунок 2.18 – Інтерфейс модального вікна для створення записки

Також присутня сторінка авторизації до вебзастосунку на рис. 2.19.



Рисунок 2.19 – Інтерфейс сторінки авторизації користувача

Висновки до розділу 2

У другому розділі кваліфікаційної роботи бакалавра розглянуто такий етап розробки застосунку як моделювання. Був проаналізований продукт BРwin, створено бізнес-процес системи. В результаті проектування було розроблені діаграми IDEF0.

В кінці розділу було змодельовано дизайн користувацького інтерфейсу вебзастосунку.

Кафедра інженерії програмного забезпечення
Вебзастосунок веденя розкладу занять

3 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ

3.1 Сценарії використання системи

В цьому розділі буде проведено ознайомлення зі сценаріями використання веб застосунку. Для цього буде розроблено діаграми варіантів використання та будуть описані варіанти використання системи за допомогою сценарної техніки опису взаємодії Use Case.

Для створення діаграми обрано інструмент для розробки UML-діаграм StarUML від розробника MKLab [9].

Інтерфейс даного інструменту можна подивитися на рис. 3.1.

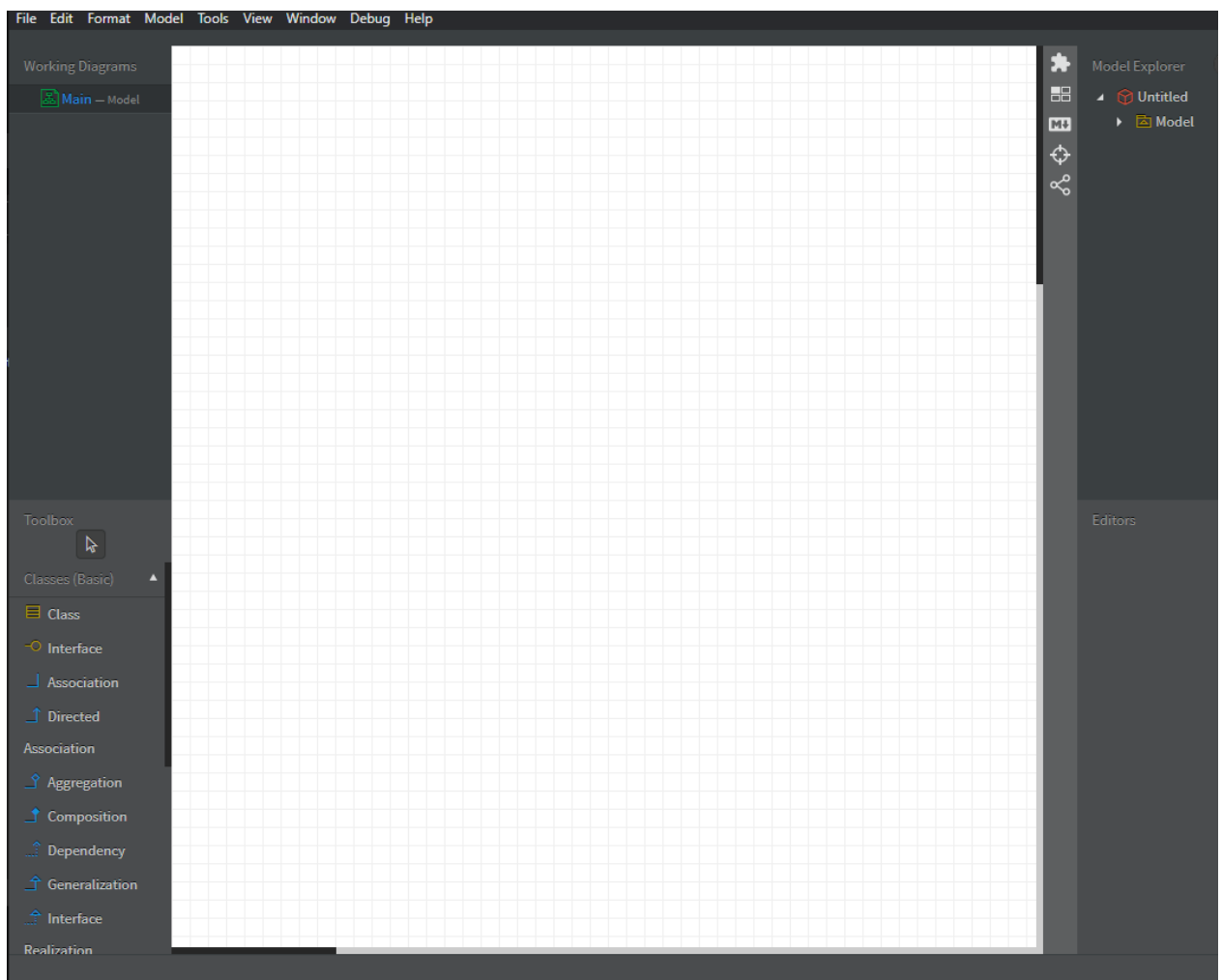


Рисунок 3.1 – Інтерфейс StarUML

Система містить лише одну дійову особу – Користувача. Діаграму варіантів використання можна побачити на рис. 3.2. На ній зображено дійову особу та доступні їй функції вебзастосунку.

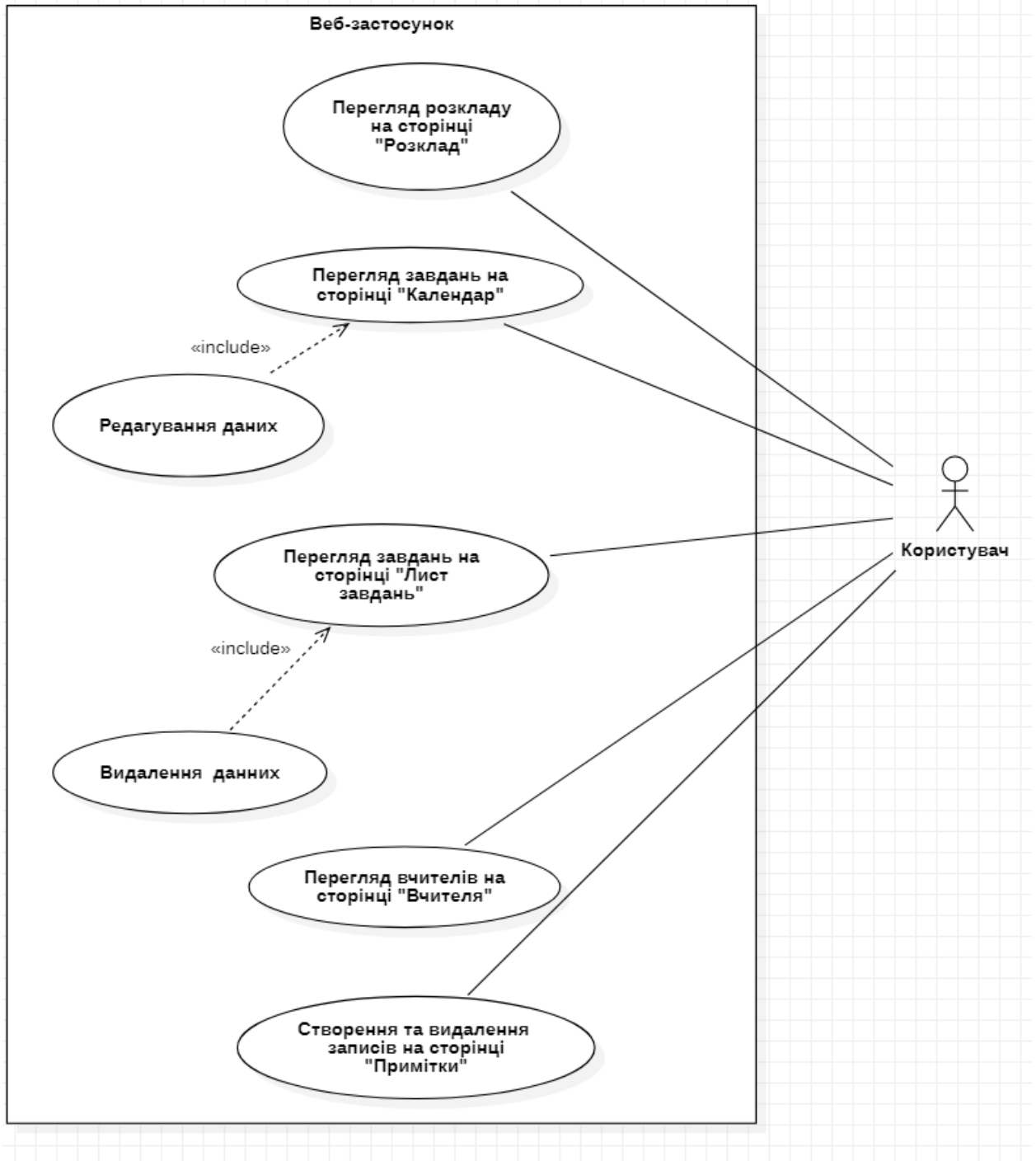


Рисунок 3.2 – Діаграма варіантів використання

Тепер потрібно створити приклади сценаріїв використання системи.

Сценарій 1. Вхід до системи.

Користувач потрапляє на сайт і йому необхідно увійти під своїм логіном та паролем. Після введення, система зв'язується з базою даних на предмет збігу. Користувач потрапляє на головну сторінку сайту. Діаграма представлена на рис. 3.3.



Рисунок 3.3 – Вхід до системи

Сценарій 2. Вхід до системи с альтернативними сценаріями

Головний сценарій.(Успішний)

Користувач потрапляє на сайт і йому необхідно увійти під своїм логіном та паролем. Після введення, система зв'язується з базою даних на предмет збігу. Користувач потрапляє на головну сторінку.

Альтернативні сценарії:

1) Помилка системи. З невідомих причин не має доступу до даних або сторінка недоступна.

2) Відказано в доступі. Введений логін або пароль невірні.

Діаграма представлена на рис. 3.4.

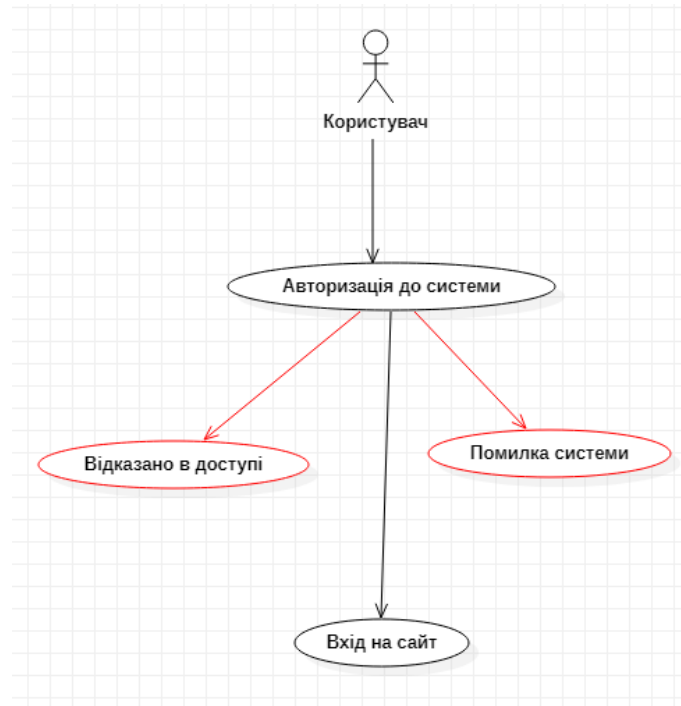


Рисунок 3.4 – Авторизація с альтернативними сценаріями

Сценарій 3. Перегляд розкладу та створення завдання на сторінці «Календар».

В табл. 3.1 можна побачити реалізацію 3 сценарію, а на рис. 3.5 – її діаграму.

Таблиця 3.1 – Перегляд розкладу та створення завдання на сторінці «Календар»

Use Case Name	Перегляд розкладу та створення завдання на сторінці «Календар»
Scope	Вебзастосунок ведення розкладу занять

Level	Додати завдання на сторінці «Календар»
Primary Actor	Користувач(студент)
Stakeholders and interests	Студент: Зацікавлений в швидкому додаванні завдання до власних даних та доступу до перегляду та\або редагуванні чи видаленні їх.
Precondition	Користувач відкрив застосунок на смартфоні чи комп'ютері, та виконав вхід до системи
Success guarantee	Користувач успішно авторизувався та перейшов на необхідну сторінку для додавання потрібного йому завдання.
Main Success Scenario	<ol style="list-style-type: none"> 1.Студент авторизується на сайті. 2.Попадає на сторінку «Розклад» та переглядає його. 3.Переходить на сторінку «Календар». 4.Натискає на кнопку «Додати». 5.У відкритшомуся модальному вікні ввести необхідні дані та обрати дату і зберігає їх.
Extensions	<ol style="list-style-type: none"> 1.Студент може відкрити сторінку «Розклад» та переглянути усі його заняття 2.Помилка системи. 3.Відказано в доступі до сайту.

Special Requirements	Пристрій з доступом до інтернету та який задовольняє мінімальні системні вимоги браузера.
Frequency of Occurrence	Система може працювати безперервно.

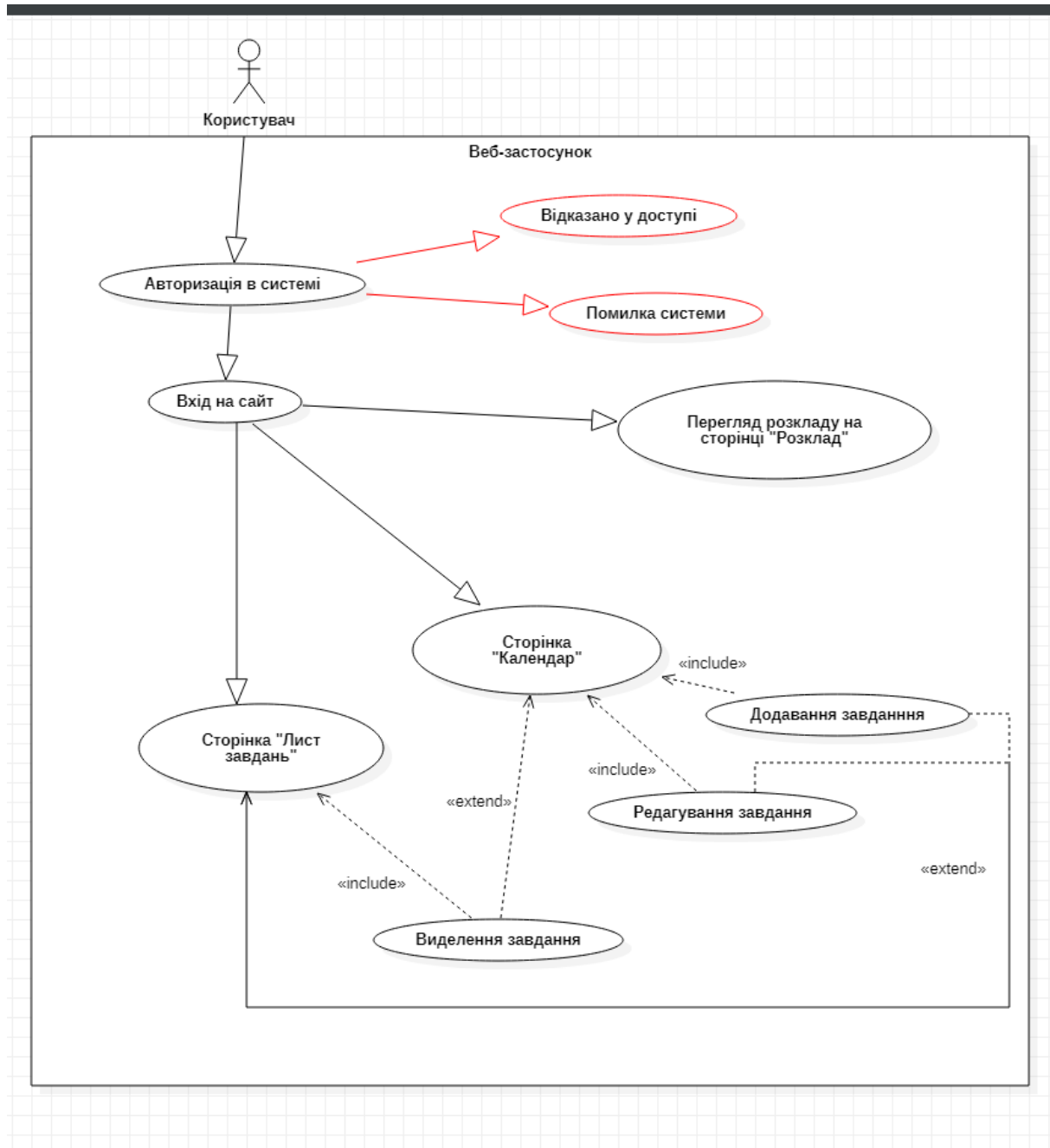


Рисунок 3.5 – Перегляд розкладу та створення завдання на сторінці «Календар»

3.2 Діаграма класів

Діаграма класів займає центральне місце в проектуванні об'єктно-орієнтованої системи. Нотація класів використовується на різних етапах проектування та будівництва з різним ступенем деталізації.

Вебзастосунок, що створюється, складається з 5 класів. Діаграму класів застосунку можна побачити на рисунку 3.6.

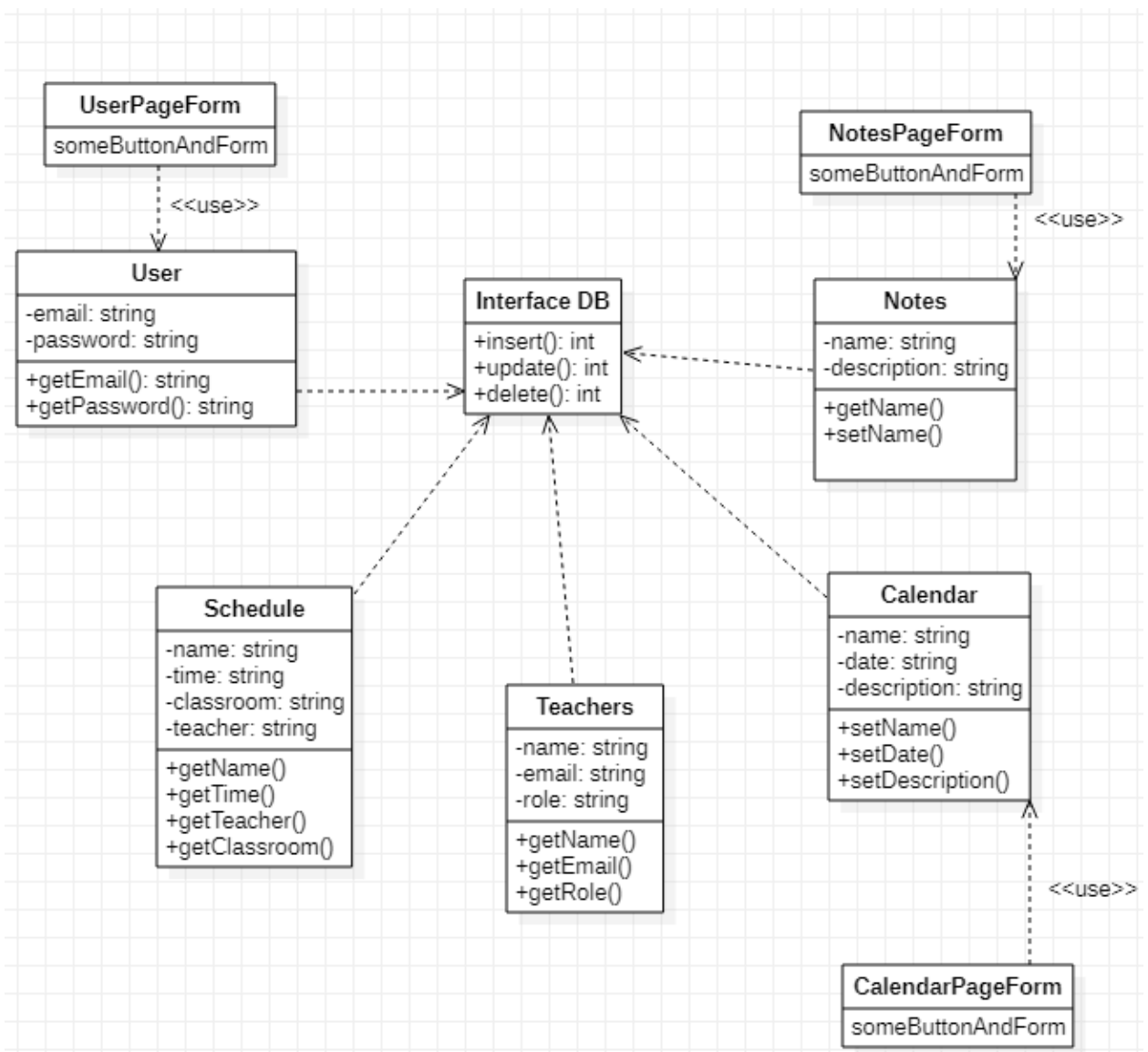


Рисунок 3.6 – Діаграма класів

Пояснення до діаграми класів клієнтської частини застосунку.

Опис класу User:

User – клас, який відповідає за представлення користувача;

- string email – поле, містить email користувача;
- string password – поле, містить пароль користувача;

Опис класу Calendar:

Calendar – клас, який відповідає за створення завдань;

- string name – поле, містить назву дисципліни;
- new Date – поле, містить дату заняття по дисципліні;
- string description – поле, містить опис дисципліни;

Опис класу Notes:

- Notes – клас, який відповідає за створення примітки;
- string name – поле, містить назву примітки;

Опис класу Teachers:

Teachers – клас, який відповідає за відображення викладачів;

- string name – поле, містить ім'я викладача;
- string email – поле, містить пошту викладача;
- string role – поле, містить посаду викладача;

Опис класу Schedule:

Schedule – клас, який відповідає за відображення розкладу;

- string name – поле, містить назву дисципліни;
- string time – поле, містить час заняття по цій дисципліні;
- string classroom – поле, містить номер кабінету;
- string teachers – поле, містить ПІБ вчителя

Опис інтерфейсу DB:

DB – інтерфейс, класи які його реалізують мають можливість взаємодіяти з БД;

- int insert() – метод, за допомогою якого виконується запис даних до

БД;

- int update() – метод, за допомогою якого виконується зміна даних у

БД;

– int delete() – метод, за допомогою якого виконується видалення даних з БД.

3.4 Діаграма послідовності

Діаграми послідовності описують взаємодію між класами в термінах обміну повідомленнями в часі. Їх також називають діаграмами подій. Діаграма послідовності є хорошим способом візуалізації та перевірки різних сценаріїв виконання.

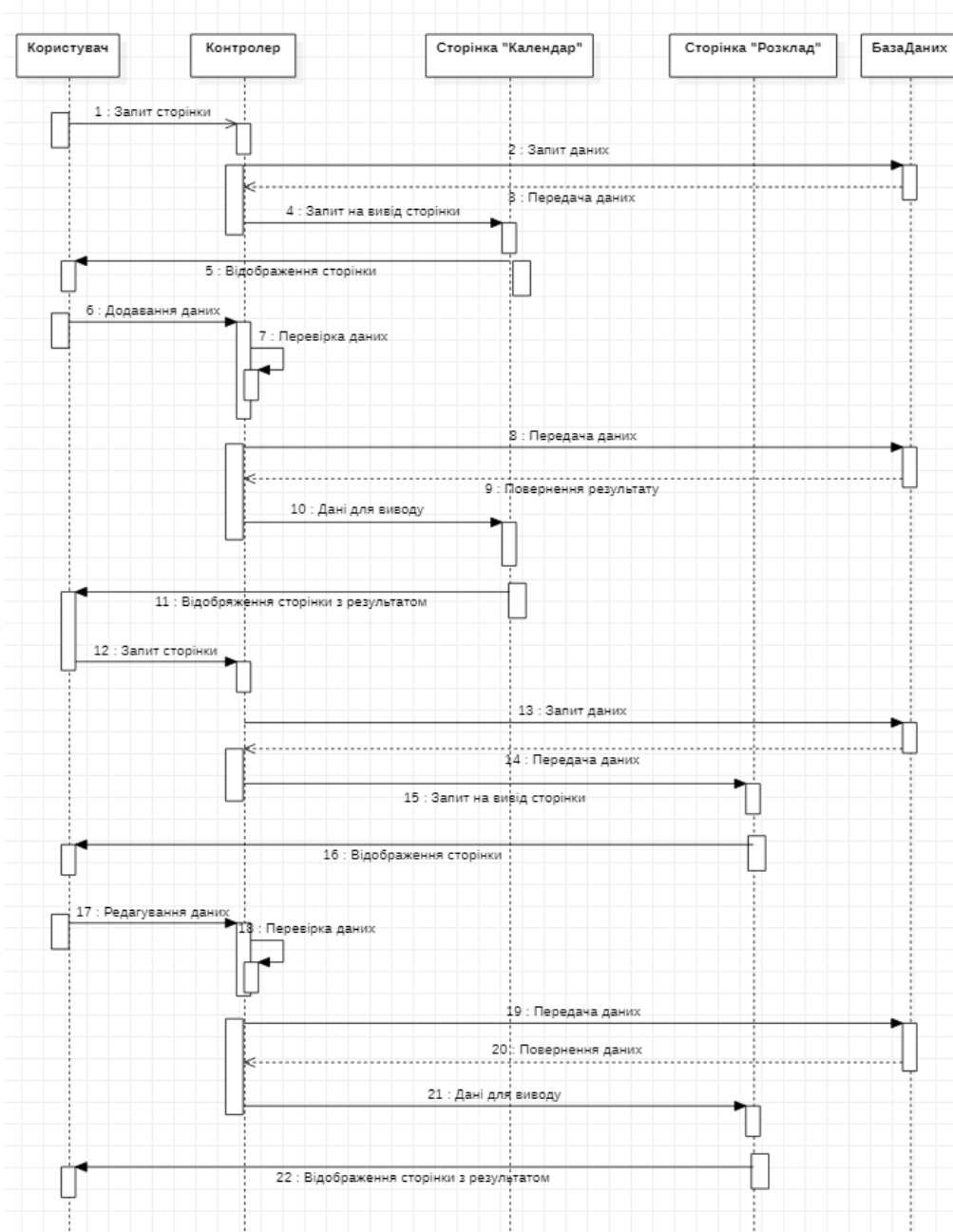


Рисунок 3.7 – Діаграма послідовності

На діаграмі послідовності зображено додавання та редагування даних. Користувач вже увійшов до системи, та хоче відкрити сторінку «Календар». Контролер робить запит до бази даних, щоб отримати необхідні дані. База даних в свою чергу, повертає дані. Контролер робить запит до сторінки «Календар» і після цього користувач бачить відображення сторінки. Користувач змінює необхідне завдання. Контролер перевіряє введені дані, відправляє до бази даних, отримує результат. Після цього, контролер передає дані до сторінки, а та відображає нові дані. Користувач вирішив додати новий предмет до розкладу. Для цього необхідно перейти до сторінки «Розклад», та обрати із розкладу необхідний день та додати дані. Логіка роботи буде аналогічною, окрім редагування даних. В цей раз вони будуть додаватися.

3.4 Технології

3.4.1 Webstorm

WebStorm [10] є інструментом для розробки вебсайтів і редагування HTML, CSS і JavaScript коду. Рішення забезпечує швидку навігацію по файлах і генерує повідомлення про проблеми в коді в режимі реального часу. JetBrains WebStorm дозволяє додавати розмітку HTML-документів або елементів SQL безпосередньо до JavaScript. JetBrains WebStorm здійснює розгортання та синхронізацію проектів через протокол FTP.

Використовуючи можливості коду HTML/XHTML та XML, WebStorm забезпечує автоматичне завершення стилів, посилань, атрибутів та інших елементів коду. Під час роботи з CSS здійснюється завершення коду класів, HTML-номерів, ключових слів тощо. WebStorm пропонує автоматичне вирішення таких проблем, як вибір формату, властивостей, класів, посилань на файли та інших атрибутів CSS.

Інтерфейс редактора можна побачити на рис. 3.8.

Кафедра інженерії програмного забезпечення
Вебзастосунк ведення розкладу занять

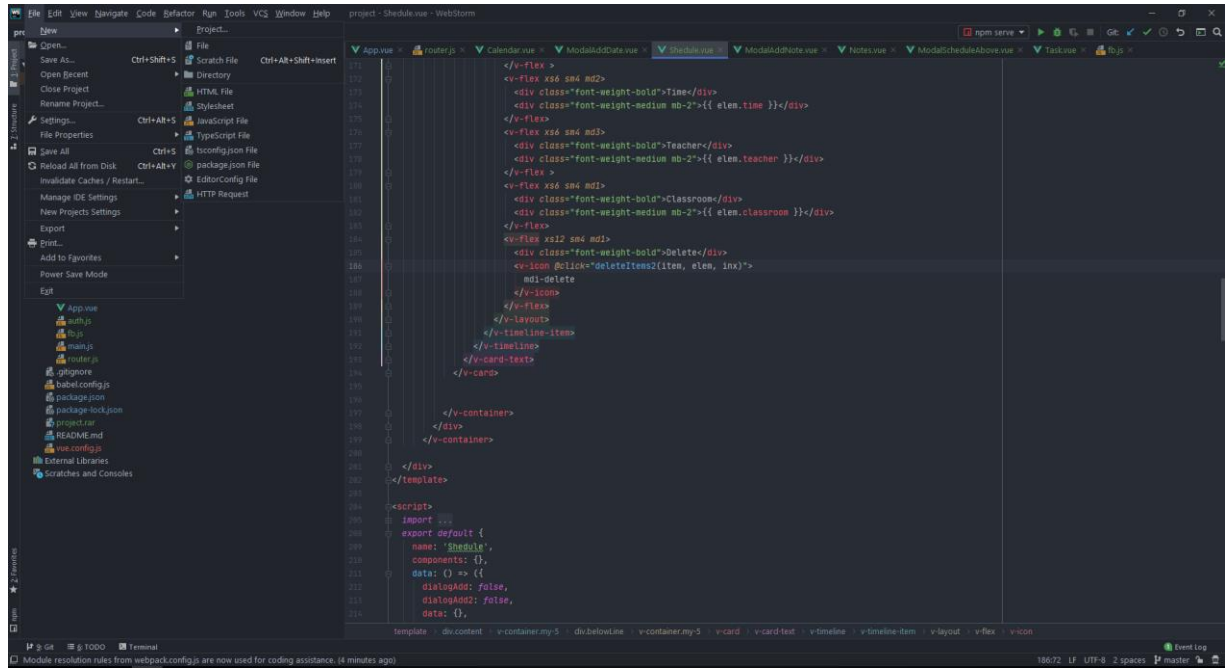


Рисунок 3.8 – Інтерфейс WebStorm

3.4.2 Vue

Vue.js [11] – це JavaScript бібліотека для створення вебінтерфейсів з використанням шаблону архітектури MVVM (Model-View-ViewModel). Оскільки Vue працює тільки на «рівні представлення» і не використовується для проміжного програмного забезпечення та бекенду, він може легко інтегруватися з іншими проектами та бібліотеками. Vue.js містить широкую функціональність для рівня представлент і може використовуватися для створення потужних односторінкових вебзастосунків.

Головну сторінку сайту можна побачити на рис. 3.9.



Рисунок 3.9 – Головна сторінка сайту Vue

Функції Vue.js:

- реактивні інтерфейси;
- декларативний рендеринг;
- зв'язування даних;
- директиви (усі директиви мають префікс «V-»). У директиву передається значення стану, а як аргументи використовуються html атрибути або Vue JS події); логіка шаблонів;
- компоненти;
- опрацювання подій;
- властивості;
- переходи та анімація CSS;
- фільтри.

Основна бібліотека Vue.js 2 дуже мала (всього 17 кБ). Це гарантує, що навантаження на проєкт, реалізований за допомогою Vue.js, мінімальне, а сайт швидко завантажуватиметься.

Vue підходить для невеликих проєктів, яким необхідно додати трохи реактивності, представити форму за допомогою AJAX, відобразити значення під час введення даних користувачем, авторизація або інші аналогічні завдання. Vue легко масштабується та добре підходить для об'ємних проєктів, тому його називають прогресивним фреймворком.

Vue також чудово підходить для великих односторінкових застосунків завдяки своїм основним компонентам, таким як Router та Vuex. З Vue можна використовувати загальнодоступні API для створення програм, так і реалізовувати виконувани сервером програми. Але Vue найкраще підходить для розробки рішень, які використовують зовнішні API для обробки даних.

За допомогою Vue також можна створювати frontend-блог на популярних CMS. Vue.js відмінно підходить і для розробки динамічних інтерфейсів, що адаптуються до користувача

Vue-Router [12] – це пакет JavaScript, який дозволяє налаштувати маршрутизацію для односторінкових програм (SPA).

SPA відноситься до вебзастосунку, який обслуговує тільки одну сторінку index.html і динамічно відображає контент, що, швидше за все, є способом налаштування сучасних JavaScript-фреймворків, таких як Vue.js.

Використання SPA має безліч переваг, але одним із основних недоліків є те, що всі компоненти вебсторінки доставляються, додаються або видаляються за допомогою JavaScript без завантаження додаткових HTML-сторінок із сервера. У цьому суть SPA, але головна проблема полягає в можливості пересуватися по сторінках, до яких користувачі звикли на більшості вебсайтів. Ось тут і допомагає Vue-Router.

3.4.3 Vuetify

В даний час вебзастосунки вже важко уявити без складних елементів керування, таких як інтерактивні таблиці з даними, діалогові вікна, випадаючі меню та ієрархічні дерева. Писати весь цей набір компонентів самостійно, щоб вирішити бізнес-завдання — довго. Але, на щастя, існують уже готові рішення, які беруть на себе створення такої інфраструктури. Одне з них – Vuetify [16].

Немає більше необхідності додавати таблиці, форми, меню та діалогові вікна у вигляді окремих пакетів у проєкт, а потім приводити їх усіх до

загального стилю оформлення. Компоненти Vuetify вже виконані в одноманітному стилі та дуже добре налаштовуються та розширюються під ваші потреби.

Сторінка сайту продемонстровано на рис. 3.10.

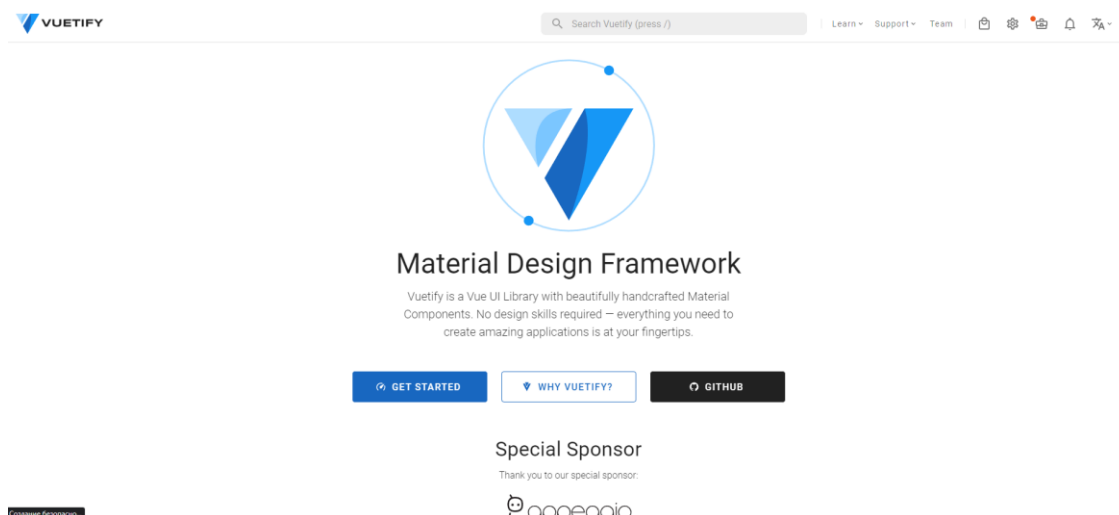


Рисунок 3.10 – Головна сторінка сайту Vuetify

Vuetify розроблений точно відповідно до специфікації Material Design. Кожен компонент розроблений вручну, щоб надати найкращі інструменти інтерфейсу користувача для майбутнього застосунку. Розробка не зупиняється на основних компонентах, описаних у специфікації Google. Завдяки підтримці членів спільноти та спонсорів, додаткові компоненти будуть розроблені та доступні для всіх.

3.4.4 Firebase

Firebase [17] - це платформа для розробки мобільних застосунків від компанії Google, в якій є найсучасніші функції для розробки, перекомпонування та покращення застосунків.

Firebase – це, по суті, набір інструментів, які розробники можуть використовувати, створюючи та змінюючи програми залежно від своєї потреби.

Мета Firebase полягає у вирішенні трьох основних проблем розробників:

- Швидко створити програму.
- Випустити та забезпечити надійний моніторинг працездатності.
- Залучити користувачів.

Головну сторінку можна переглянути на рис. 3.11.

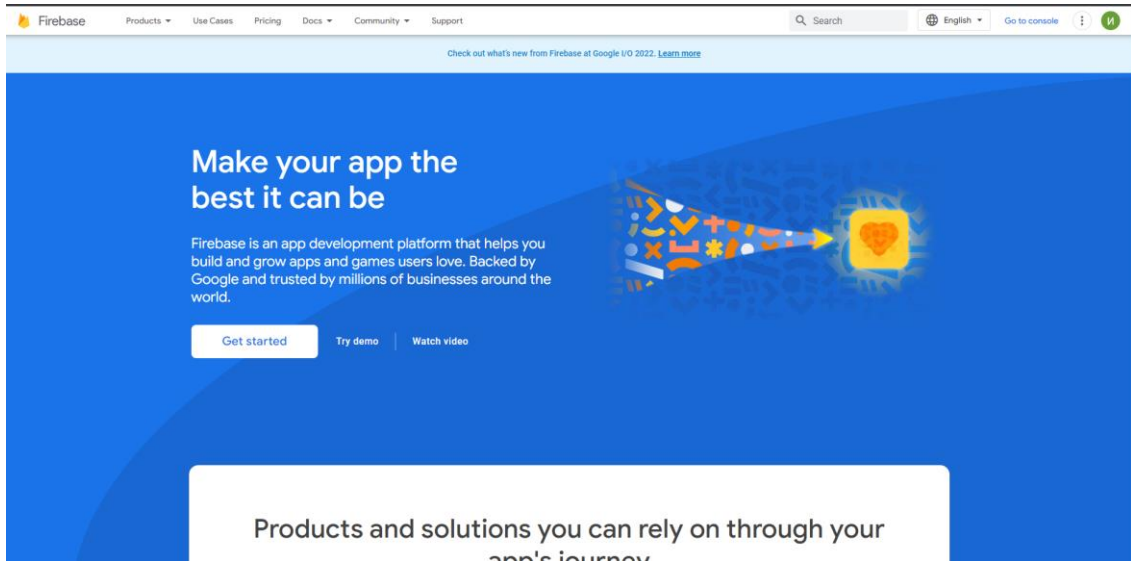


Рисунок 3.11 – Головна сторінка сайту Firebase

Розробники, що використовують цю платформу, отримують доступ до сервісів, за допомогою яких вони можуть розробляти свої продукти, і це дозволяє їм зосередитися безпосередньо на наданні якісного продукту.

Деякі з найпопулярніших функцій платформи Google Firebase включають бази даних, автентифікацію, push-сповіщення, аналітику, зберігання файлів і багато іншого.

Висновки до розділу 3

У третьому розділі був розглянутий такий етапи розробки застосунку як проектування. В результаті проектування спочатку розроблено діаграму варіантів використання за допомогою сценарної техніки опису взаємодії Use Case, та описано варіант використання. Також було розроблено та описано діаграму класів застосунку.

У кінці розділу було наведено основні технології для розробки клієнтської та серверної частини вебзастосунку.

4 ПРОЄКТУВАННЯ І ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ

4.1 Налаштування середовища для розробки

Спочатку перед розробкою потрібно налаштувати середовище для програмування. Як було описано раніше, в якості середовища для розробки буде використано WebStorm. Налаштування у даному застосунку досить просте, все починається з першого запуску, де відкривається головне вікно редактора з підказками до гагачих клавіш (рис. 4.1). Далі середовище запропонує завантажити додаткові плагіни для більш зручного програмування, так як будуть з'являтися підказки чи спрацьовувати автопідстановку тексту.

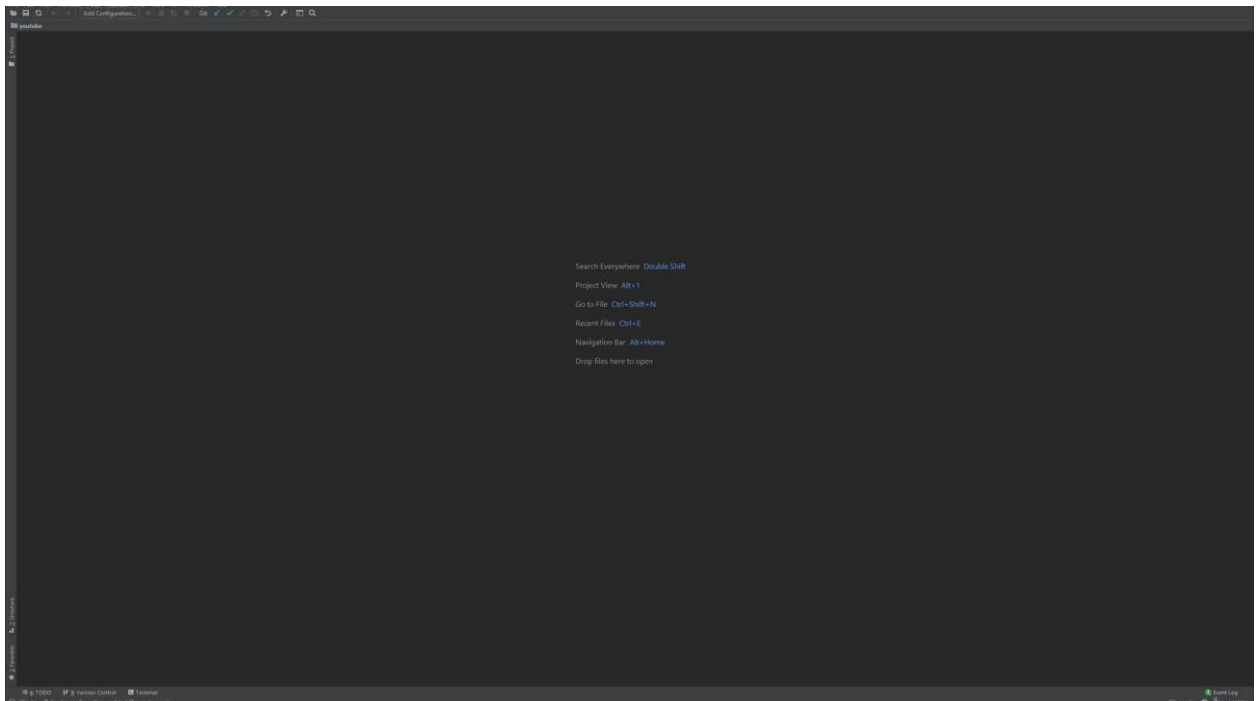


Рисунок 4.1 – Головна сторінка редактору

Другим кроком налаштування є створення проєкту у редакторі. Завдяки плагінам середовище буде зразу запропоновувати вебфреймворки. Обираємо потрібний нам це VueJS та натискаємо кнопку «Create». Зі створенням нового проєкту і якщо було обрано потрібний фреймворк,

останній в свою чергу почне завантажуватись та встановлюватись до проекту.

Обираємо місце знаходження, де буде створюватися проект та натискаємо кнопку.

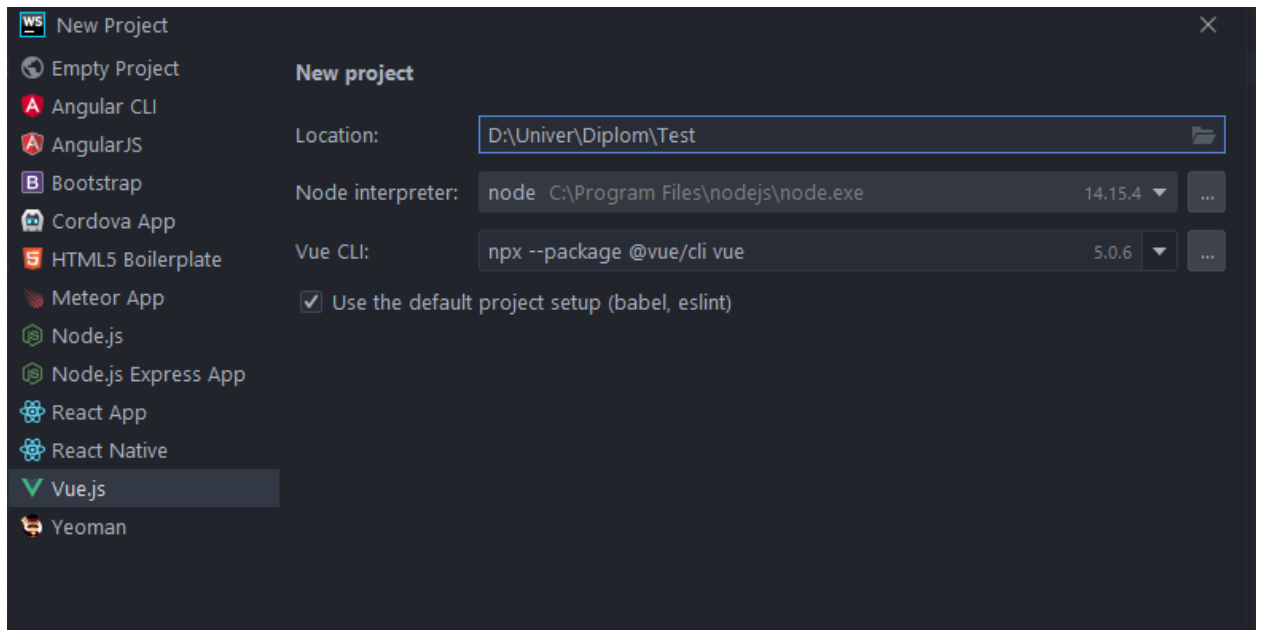


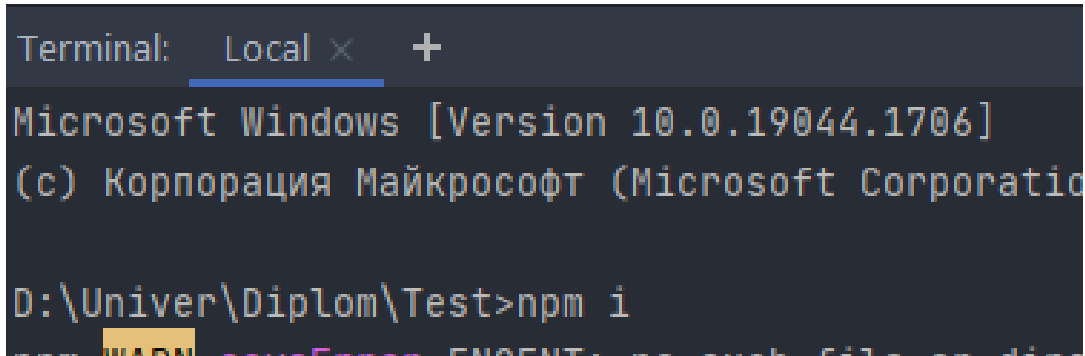
Рисунок 4.2 – Створення нового проекту з фреймворком Vue

Також перед розробкою вебзастосунку потрібно налаштувати середовище та встановити всі необхідні залежності, бібліотеки та пакети. Все це робиться за допомогою командного рядка яка має безліч альтернатив.

В якості командного рядка можна використовувати як і звичайну консоль так і PowerShell, термінал PhpStorm та консоль OpenServer. Отже першим кроком для розробки клієнтської частини потрібно завантажити та встановити фреймворк. Для цього використовується пакетний менеджер npm, який однією командою завантажує усі потрібні залежності та встановлює їх.

NPM – менеджер який використовується багатьма JS фреймворками для завантаження додаткових модулів та компонентів.

У відкритій консолі у редакторі вводимо необхідне.



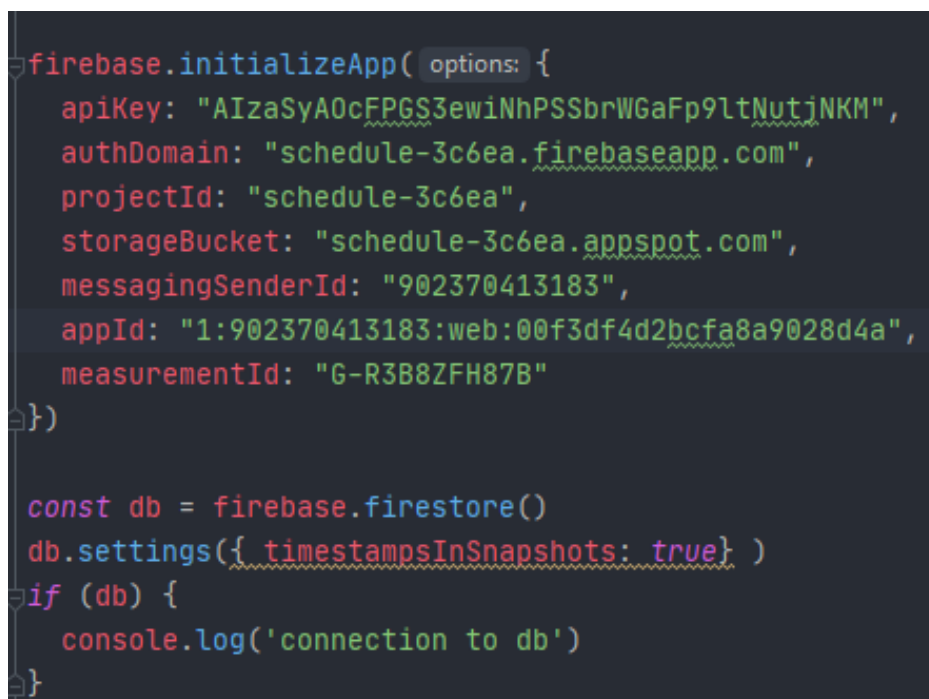
```
Terminal: Local x +
Microsoft Windows [Version 10.0.19044.1706]
(c) Корпорация Майкрософт (Microsoft Corporation)

D:\Univer\Diplom\Test>npm i
```

Рисунок 4.3 – Встановлення додаткових бібліотек та оновлення старих. Тепер, встановивши фреймворк та всі потрібні залежності, проект готовий до написання власного коду.

4.3 Розробка бази даних

Першим кроком необхідно перейти на головну сторінку Firebase, після на Firestore Database. Другим кроком треба перейти до налаштувань і скопіювати конфігурацію для підключення бази даних до проекту. У створеному js-файлі, вставимо конфігурацію та виведемо у лог-змінну, щоб перевірити підключення бази даних (рис. 4.4 – 4.5).



```
firebase.initializeApp( options: {
  apiKey: "AIzaSyA0cFP6S3ewiNhPSSbrWGaFp9ltNutjNKM",
  authDomain: "schedule-3c6ea.firebaseio.com",
  projectId: "schedule-3c6ea",
  storageBucket: "schedule-3c6ea.appspot.com",
  messagingSenderId: "902370413183",
  appId: "1:902370413183:web:00f3df4d2bcfa8a9028d4a",
  measurementId: "G-R3B8ZFH87B"
})

const db = firebase.firestore()
db.settings({ timestampsInSnapshots: true })
if (db) {
  console.log('connection to db')
}
```

Рисунок 4.4 – Ініціалізація бази даних

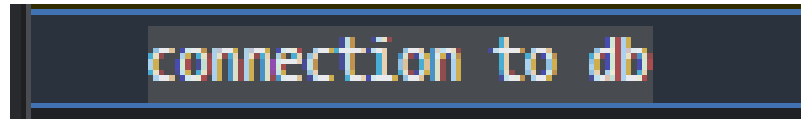


Рисунок 4.5 – Перевірка підключення

База даних підключена та працює.

Тепер у вкладці Firestore Database необхідно створити колекції, де буде зберігатися інформація сторінок вебзастосунку. Усі створені колекції представлено на рис. 4.6.

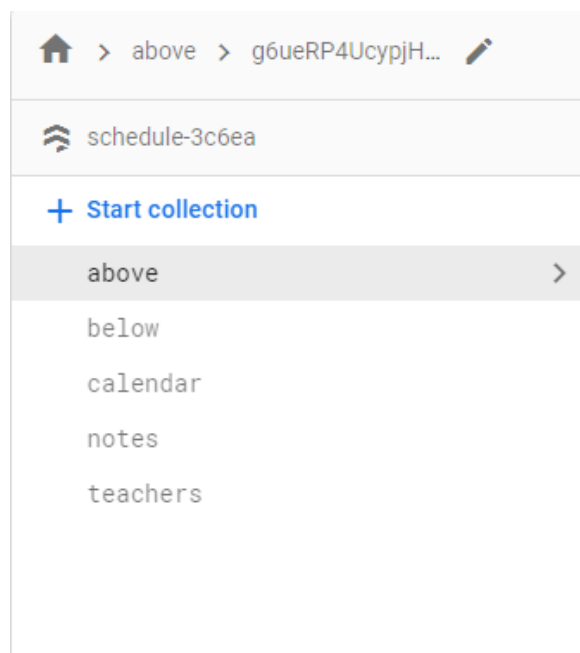


Рисунок 4.6 – Створені колекції

На сторінці authentication створимо користувача для входу до вебзастосунку.

Identifier	Providers	Created ↓	Signed In	User UID
igor@gmail.com		Apr 15, 2022	Jun 16, 2022	6QoMkMASr6hZ79AQNpCvnraHV...

Search by email address, phone number, or user UID Add user ↻ ⋮

Rows per page: 50 1 - 1 of 1 < >

Рисунок 4.7 – Таблиця користувача

4.3 Розробка клієнтської частини

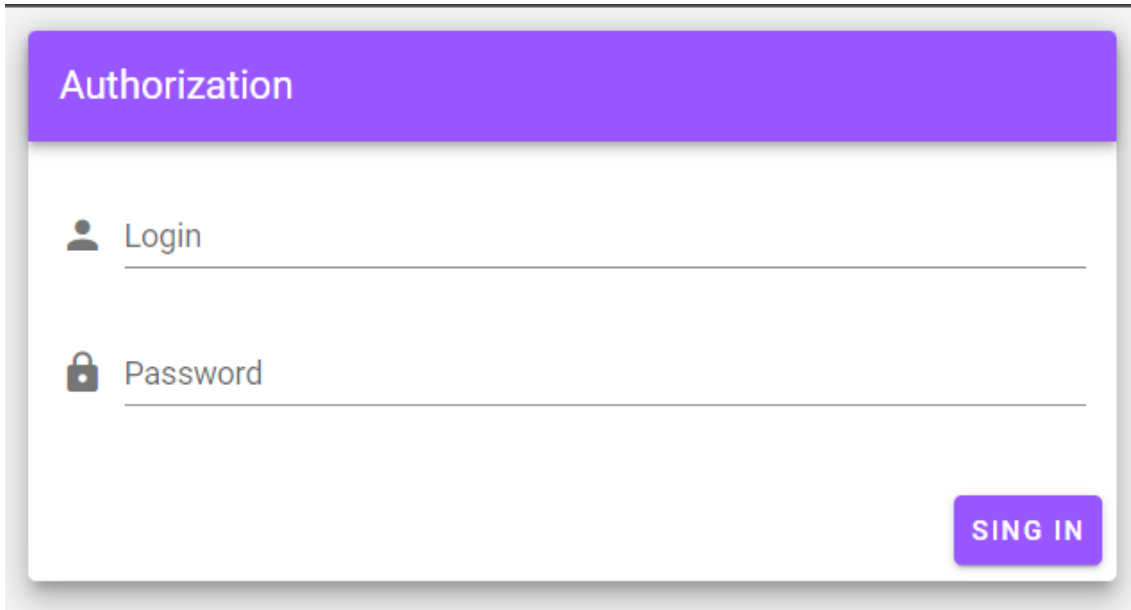
На початку розробки розглянемо файл проєкту «package.json». Даний файл є конфігураційним документом проєкту, тобто в ньому зберігається вся основна інформація про проєкт, а саме назва проєкту, версія, команди для виконання тих чи інших функцій, залежності, додаткові конфігурації.

```
1 {
2   "name": "project",
3   "version": "0.1.0",
4   "private": true,
5   "scripts": {
6     "serve": "vue-cli-service serve",
7     "build": "vue-cli-service build",
8     "lint": "vue-cli-service lint"
9   },
10  "dependencies": {
11    "core-js": "^3.6.5",
12    "date-fns": "^2.28.0",
13    "firebase": "^9.6.10",
14    "vue": "^2.6.11",
15    "vue-router": "^3.4.4",
16    "vuetify": "^2.4.0"
17  },
18 }
```

Рисунок 4.4 – Зміст частини файлу конфігурації клієнтської частини

Клієнтська частина, як було обговорено у попередньому розділі, буде виконана на базі фреймворку VueJS та основною мовою програмування використано JavaScript. А для прискорення розробки використано бібліотеку готових елементів Vuetify.

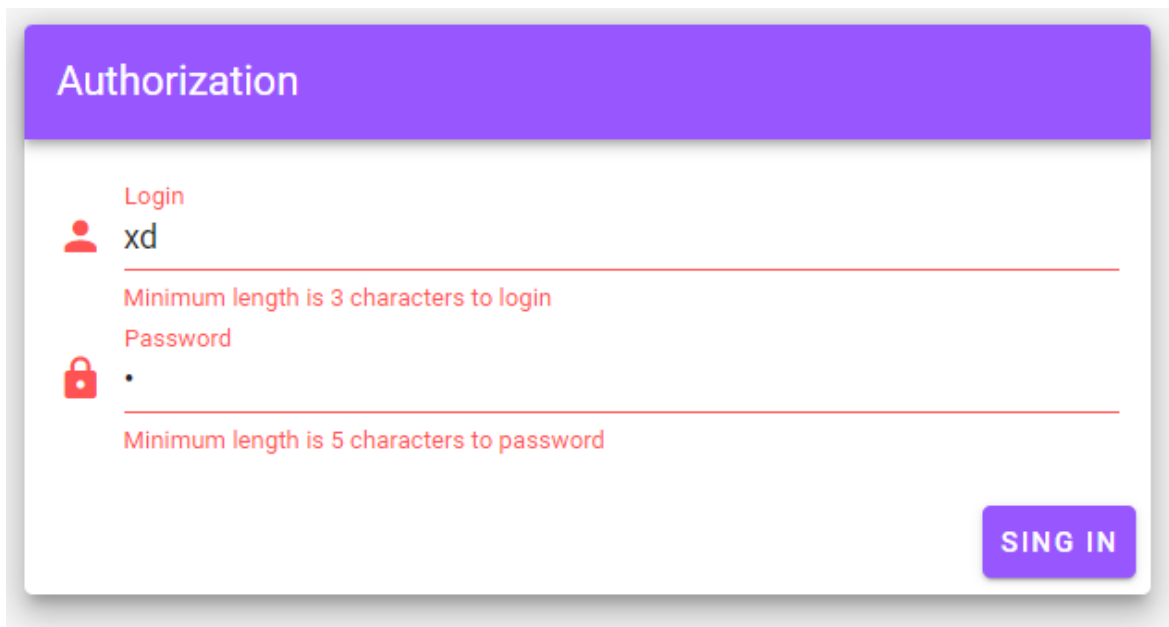
Під час потрапляння користувача на сторінку, його одразу переводить до сторінки авторизації.



The image shows a web form titled "Authorization" with a purple header. It contains two input fields: "Login" with a person icon and "Password" with a lock icon. A purple "SIGN IN" button is located at the bottom right.

Рисунок 4.5 – Готова сторінка авторизації

Також для введених даних була додана валідація цих самих даних. Це сама звичайна перевірка на наповненість полів та перевірка на скільки заповнені ці поля. Переглянути на ефект валідації, можна побачити на рисунку 4.6.



The image shows the same "Authorization" form as in Figure 4.5, but with validation errors. The "Login" field contains "xd" and has a red error message: "Minimum length is 3 characters to login". The "Password" field contains a single dot and has a red error message: "Minimum length is 5 characters to password". The "SIGN IN" button is still present at the bottom right.

Рисунок 4.6 – Повідомлення про помилку

В кодї валідація задається правилами за якими і валідуються оголошені поля. Також з функціоналу додані методи для авторизації.

```
57 <script>
58   import firebase from "firebase/compat";
59   export default {
60     name: 'AuthorizationPage',
61     data: () => ({
62       login: '',
63       pass: '',
64       alertModel: false,
65       inputRulesForLogin: [
66         v => v.length >= 3 || 'Minimum length is 3 characters to login'
67       ],
68       inputRulesForPass: [
69         v => v.length >= 5 || 'Minimum length is 5 characters to password'
70       ]
71     }),
72     methods: {
73       async pressed() {
74         try {
75           await firebase.auth().signInWithEmailAndPassword(this.login, this.pass)
76           this.$router.replace({location: {name: 'schedule'}})
77           console.log(this.login, this.pass)
78         } catch (e) {
79           this.alertModel = true
80         }
81       }
82     }
83   }
84 </script>
```

Рисунок 4.7 – Програмний код повідомлення про помилку

Для зручності користувачів було додано простий, але в той час і важливий елемент – це модальне вікно помилки, при ведення невірних даних.

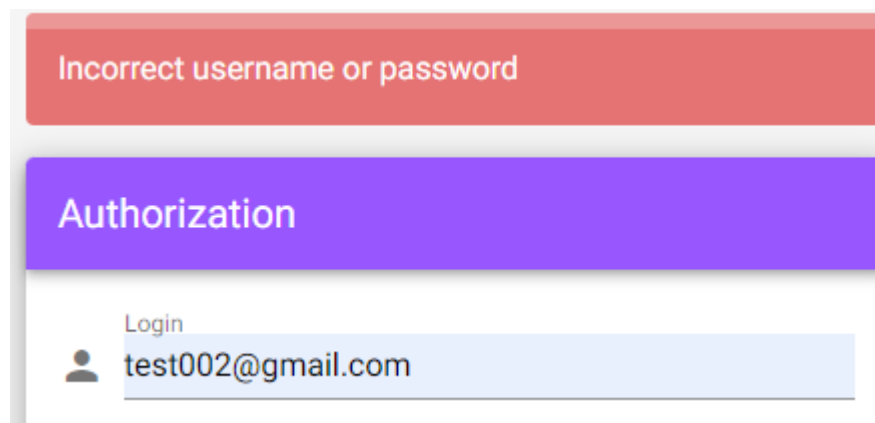


Рисунок 4.8 – Модальне вікно

Функціоналом даної сторінки є авторизація користувача. В ньому відсилається запит на авторизацію користувача і в залежності від стану

аккаунту користувач входить у систему чи отримує повідомлення з причиною чому його не авторизувало.

```

<v-card class="elevation-10">
  <v-toolbar dark color="#9856ff">
    <v-toolbar-title>Authorization</v-toolbar-title>
  </v-toolbar>
  <v-card-text>
    <v-form>
      <v-text-field
        prepend-icon="mdi-account"
        name="login"
        label="Login"
        type="text"
        color="#9856ff"
        v-model="login"
        :rules="inputRulesForLogin"
      ></v-text-field>
      <v-text-field
        id="password"
        prepend-icon="mdi-lock"
        name="password"
        label="Password"
        type="password"
        color="#9856ff"
        v-model="pass"
        :rules="inputRulesForPass"
      ></v-text-field>
    </v-form>
  </v-card-text>
  <v-card-actions>
    <v-spacer></v-spacer>
    <v-btn color="#9856ff" class="white--text" @click="pressed">Sing in</v-btn>
  </v-card-actions>
</v-card>
</v-flex>
</v-layout>
</v-container>

```

Рисунок 4.9 – Частина коду представлення сторінки авторизації

На даному рисунку можна побачити, як використовуються теги Vue та Vuetify. Та також можна використовувати класичні HTML-теги за необхідністю, але це потрібно так як обрані бібліотеки можуть повністю замінити класичні теги розмітки. Але після компіляції frontend-частини всі

теги фреймворку перетворюються та компілюються у класичні теги розмітки.

4.3.1 Головна сторінка

Наступною реалізованою сторінкою стає головна сторінка застосунку, тобто сторінка де відображаються розклад.

Як можна побачити, по ліву сторону вікна застосунку знаходиться меню з переходом на інші сторінки. Основну частину займає список розкладу, який поділений на 2 частини, під межею, та над межею. Зверху знаходиться панель, де видно сьогоднішню дату, назву спеціальності та кнопку щоб вийти з сайту.

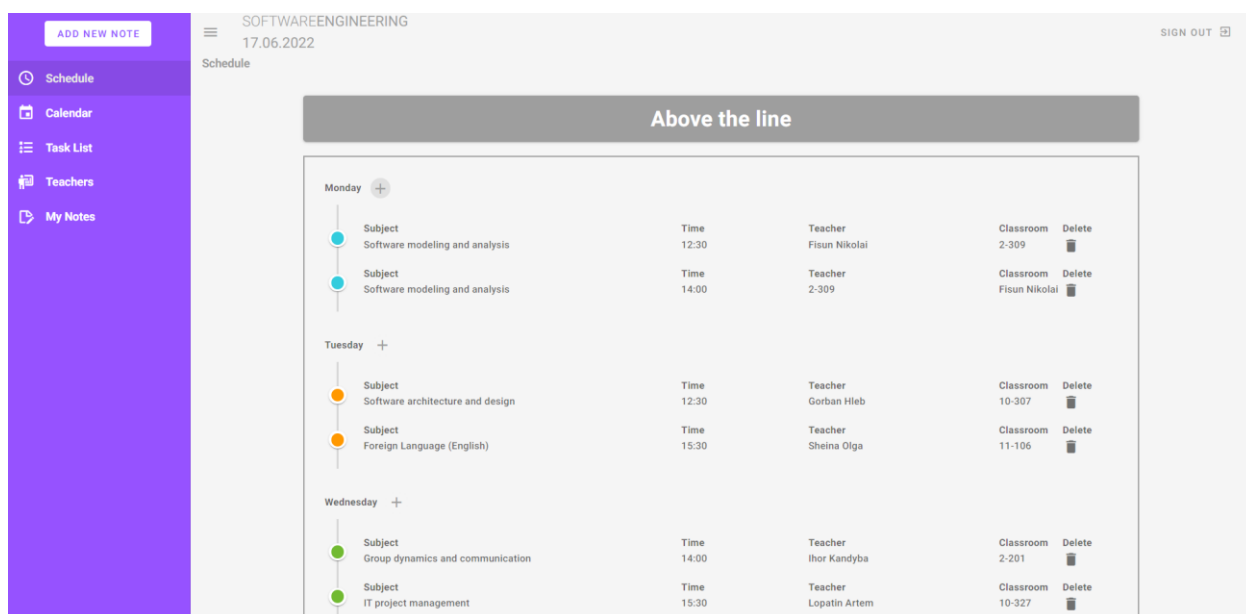


Рисунок 4.10 – Головна сторінка сайту

Біля кожного дня тижня присутня кнопка у вигляді плюса. При її натисканні, відкривається модальне вікно для додавання нової дисципліни.

Рисунок 4.11 – Частина коду представлення сторінки авторизації
Після натиску кнопки, предмет додається до обраного дня.

Subject	Time	Teacher	Classroom	Delete
Software modeling and analysis	12:30	Fisun Nikolai	2-309	
Software modeling and analysis	14:00	2-309	Fisun Nikolai	
Тест1	Тест1	Тест1	Тест1	

Рисунок 4.12 – Додавання дисципліни до розкладу

У кінці кожного предмету є кнопка для видалення його видалення.

4.3.2 Сторінка «Календар»

Наступною сторінкою для розробки є сторінка «Календар». Ця сторінка відповідає за створення завдань, наприклад лабораторна робота. Інформація буде виводитися у компонент `v-calander`, котрий відразу створює шаблон календаря.

Кафедра інженерії програмного забезпечення
Вебзастосунок ведення розкладу занять

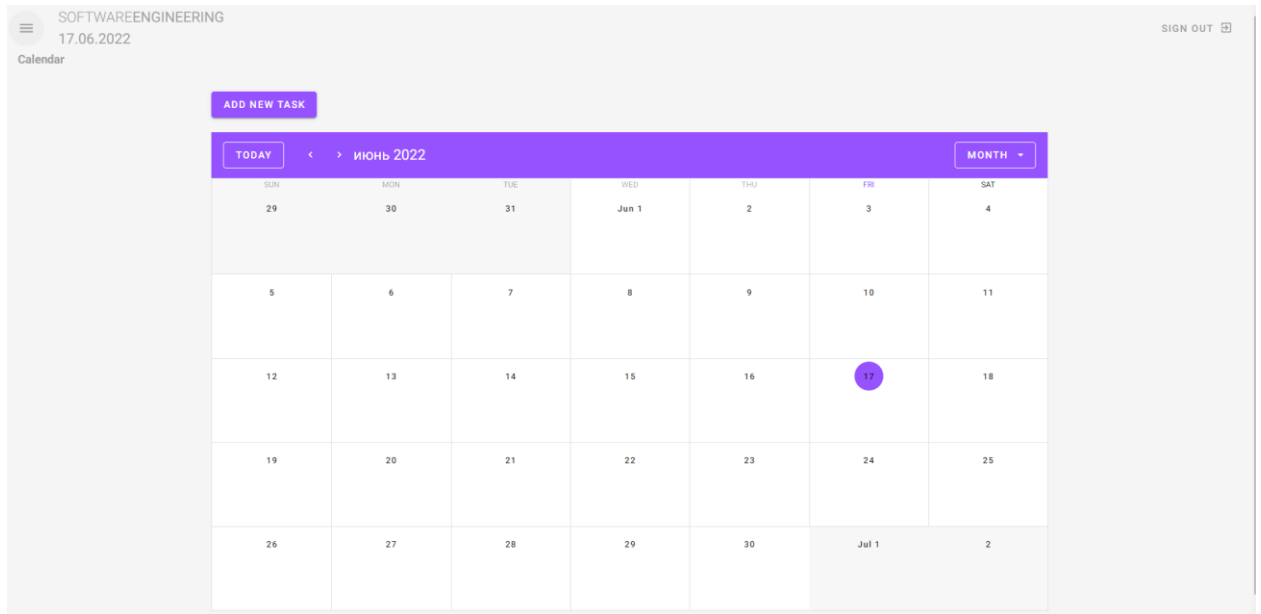


Рисунок 4.13 – Сторінка «Календар»

Над самим календарем, є кнопка для створення у модальному вікні самого завдання, у який входить назва, описання, дата и колір, яким буде відображатися у календарі.

Рисунок 4.14 – Вікно для створення завдання

Тепер завдання відображаються у самому календарі.

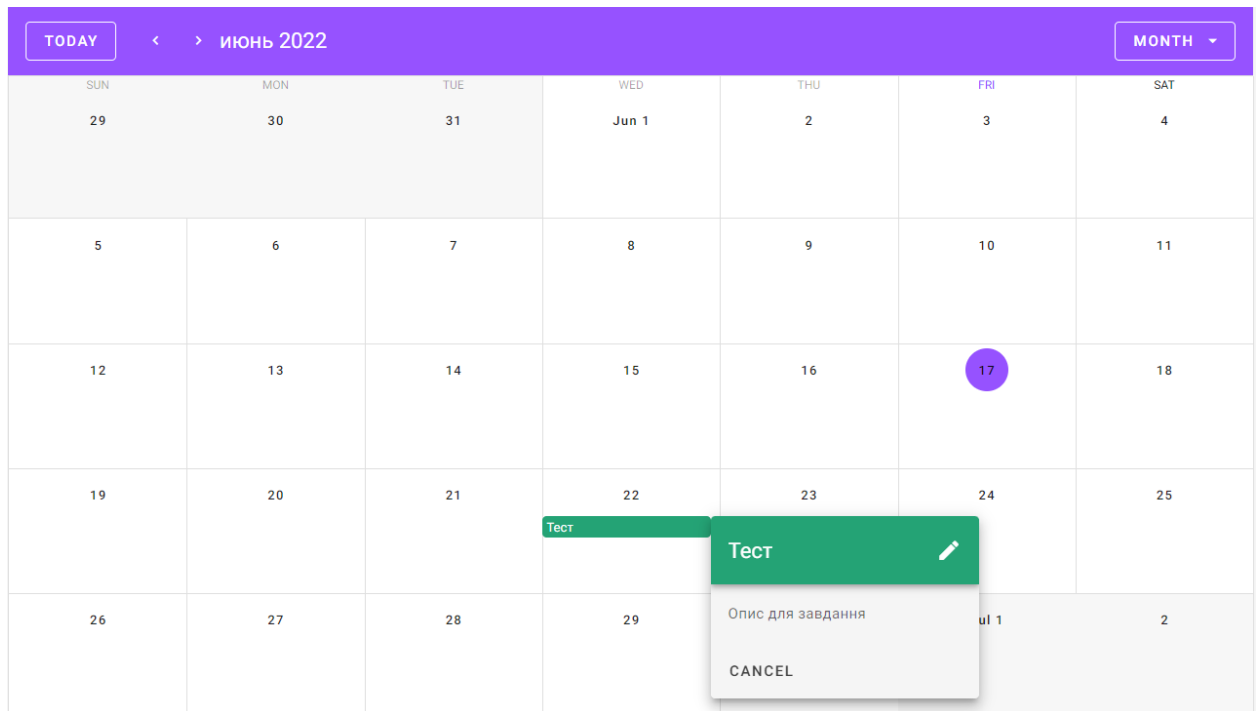


Рисунок 4.15 – Перевірка додання завдання до календаря

При необхідності, завдання можна змінити натиснувши на кнопку «олівця». Додавання до бази даних відбувається за допомогою запитів.

```

async add(){
  let newTask = {
    name: this.title,
    start: this.due,
    description: this.descrip,
    color: this.showColor,
  }
  if (this.$refs.form.validate()){
    await db.collection( collectionPath: 'calendar').add(newTask)
    // this.events.push(newTask)
    this.title = ''
    this.descrip = ''
    this.due = ''
    this.color = ''
    this.dialog = false
  }
},

```

Рисунок 4.16 – Запит до бази даних

Щоб відправити дані до бази даних, вони спочатку проходять перевірку на валідацію, а потім потрібно обрати необхідну колекцію, в котру передати об'єкт зі створеними даними.

4.3.3 Сторінка «Лист завдань»

Третьою сторінкою для розробки буде «Лист завдань». Ця сторінка є зв'язаною зі сторінкою календаря. Тут виводяться у вигляді списку завдання, які створювалися на попередній сторінці.

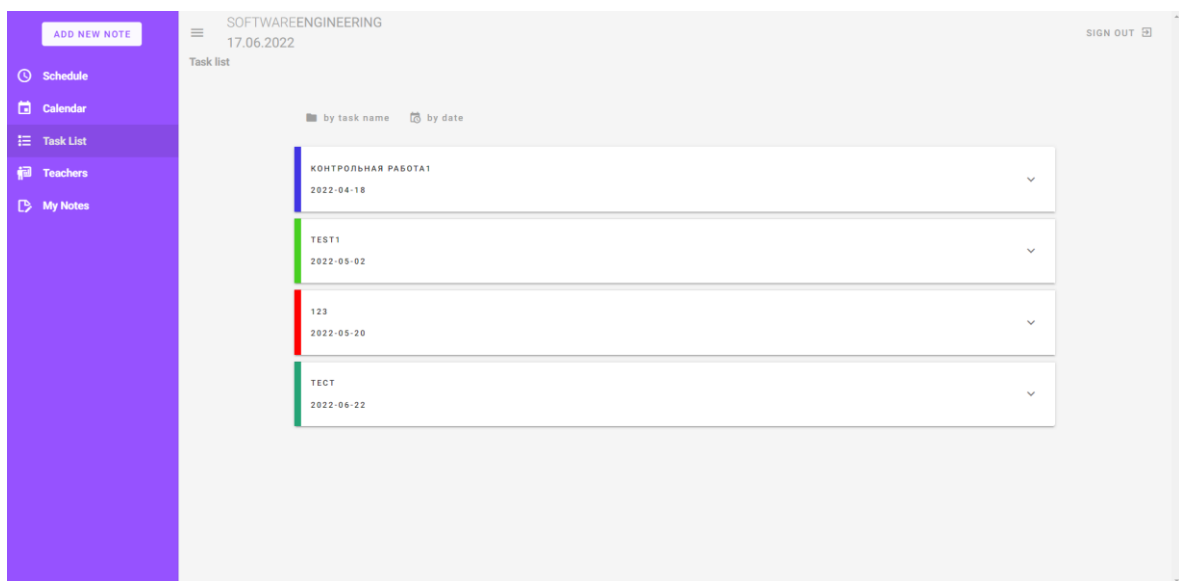


Рисунок 4.17 – Сторінка «Лист завдань»

При натисненні на необхідне завдання, воно відкриється, щоб відобразити описання та кнопку видалення.

Кафедра інженерії програмного забезпечення
Вебзастосунок ведення розкладу занять

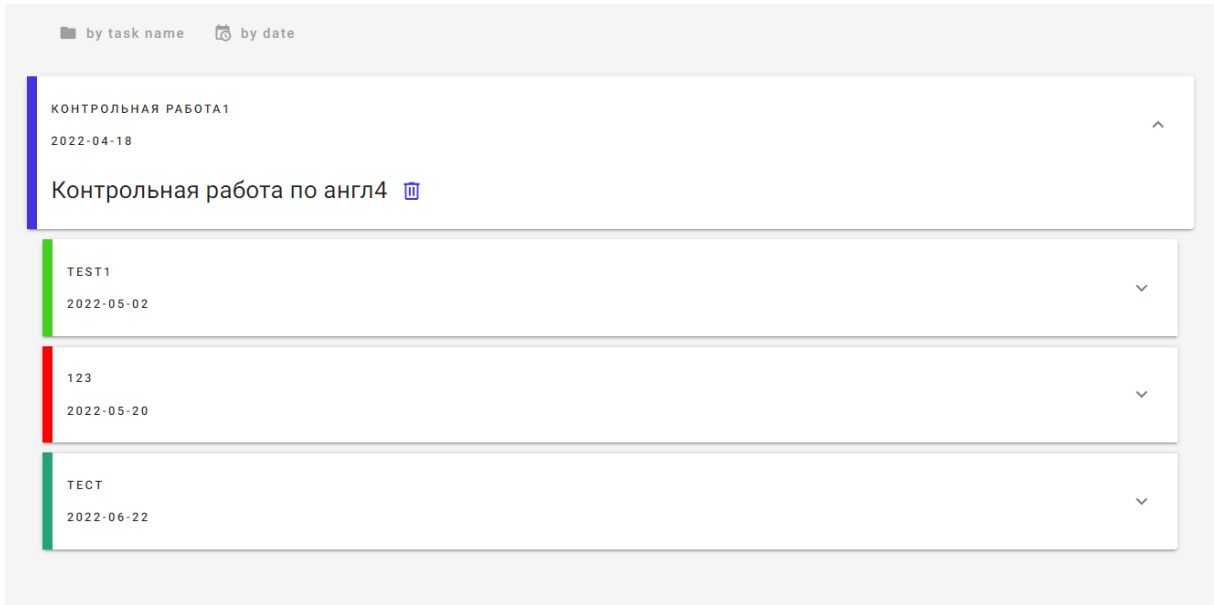


Рисунок 4.18 – Завдання після натиску

На цій сторінці над списком присутні дві кнопки, для фільтрації списку. Відфільтрувати можна за назвою або датою.



Рисунок 4.19 – Сортування за назвою

```

methods: {
  sortBy(prop) {
    this.events.sort( compareFn: (a,b) => a[prop] < b[prop] ? -1 : 1)
  },

```

Рисунок 4.20 – Метод для сортування

Сортування відбувається за допомогою функції `sort()` та передається до кнопки, в яку передається необхідний для фільтрації елемент.

```

<v-btn depressed class="grey--text" @click="sortBy( prop: 'name')">
  <v-icon left >mdi-folder</v-icon>
  <span class="text-lowercase">By task name</span>
</v-btn>
<v-btn depressed class="grey--text" @click="sortBy( prop: 'start')">
  <v-icon left >mdi-calendar-clock</v-icon>
  <span class="text-lowercase">By date</span>
</v-btn>

```

Рисунок 4.21 – Метод сортування на кнопках

4.3.4 Сторінка «Викладачі»

Четверта за рахунком сторінка має назву «Викладачі». Дана сторінка має собою структуру списку, але у вигляді карток.

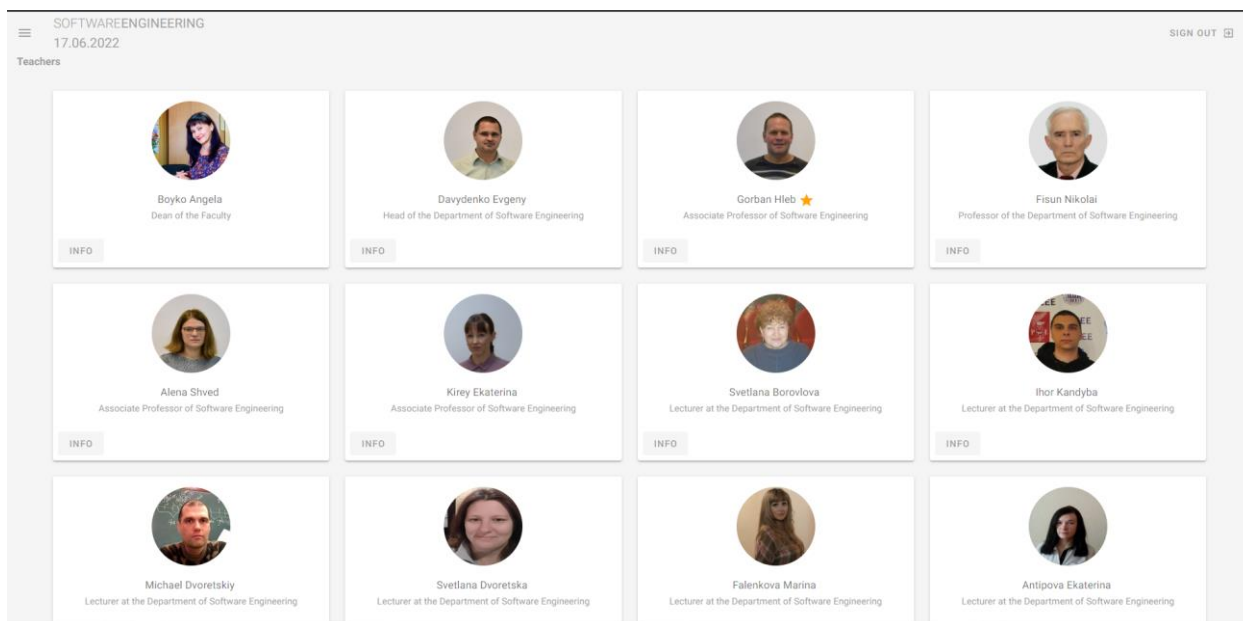
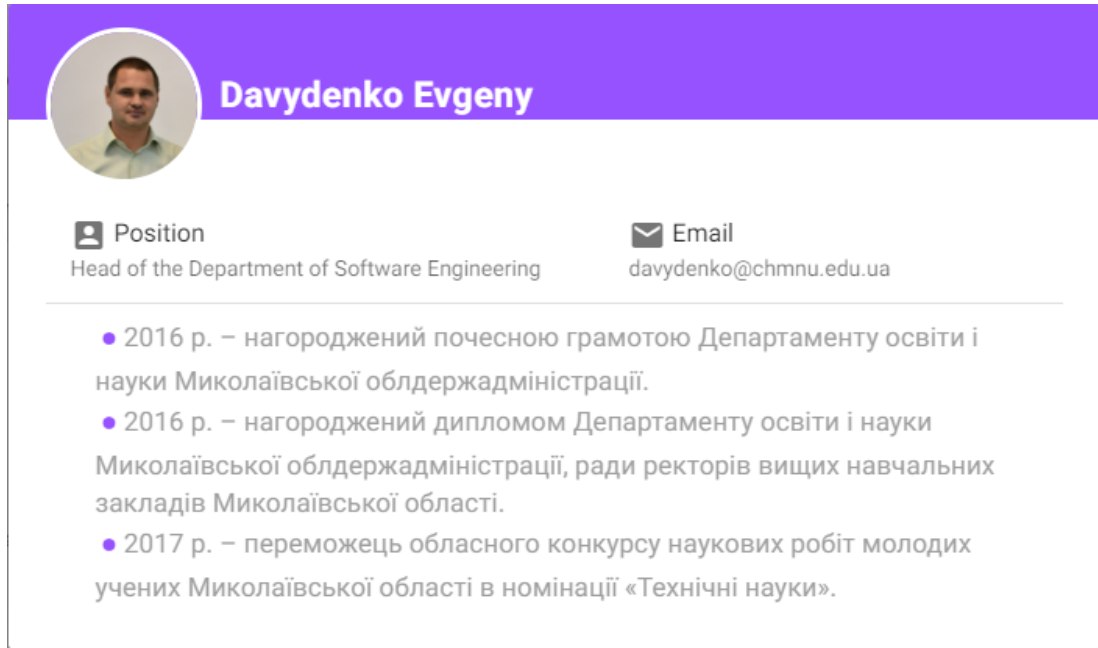




Рисунок 4.22 – Сторінка «Вчителя»

На кожній картці відображається фото викладача, його ім'я, прізвище та посада. У нижньому лівому куті знаходиться кнопка, при натисканні якої буде виводитися більше інформації про обраного викладача.



Davydenko Evgeny

 **Position**
 Head of the Department of Software Engineering

 **Email**
 davydenko@chmnu.edu.ua

- 2016 р. – нагороджений почесною грамотою Департаменту освіти і науки Миколаївської облдержадміністрації.
- 2016 р. – нагороджений дипломом Департаменту освіти і науки Миколаївської облдержадміністрації, ради ректорів вищих навчальних закладів Миколаївської області.
- 2017 р. – переможець обласного конкурсу наукових робіт молодих учених Миколаївської області в номінації «Технічні науки».

Рисунок 4.23 – Детальна інформація про викладача

У куратора групи, біля прізвища є спеціальна іконка у вигляді зірки.

Vuetify має спеціальні теги, які допомагають адаптувати сторінку та її елементи під мобільний пристрій.

```
<v-flex xs12 sm6 md4 lg3
```

Рисунок 4.24 – Теги розмірів сторінки

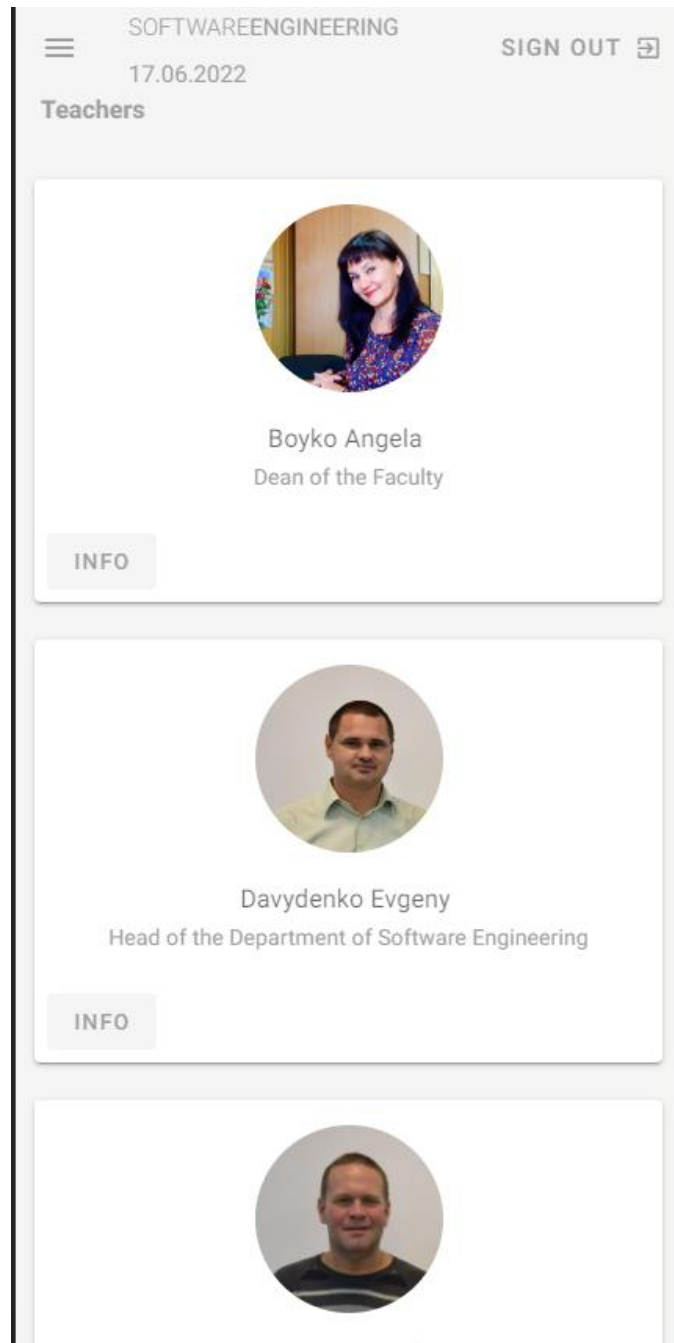


Рисунок 4.25 – Сторінка на мобільному пристрої

Як бачимо, теги дуже прискорює розробку адаптації сторінок.

4.3.5 Сторінка «Нотатки»

Перейшовши на сторінку з нотатками, користувач одразу бачить панель для введення тексту. Після занесення тексту, користувач може видалити його або поміти та викреслити.

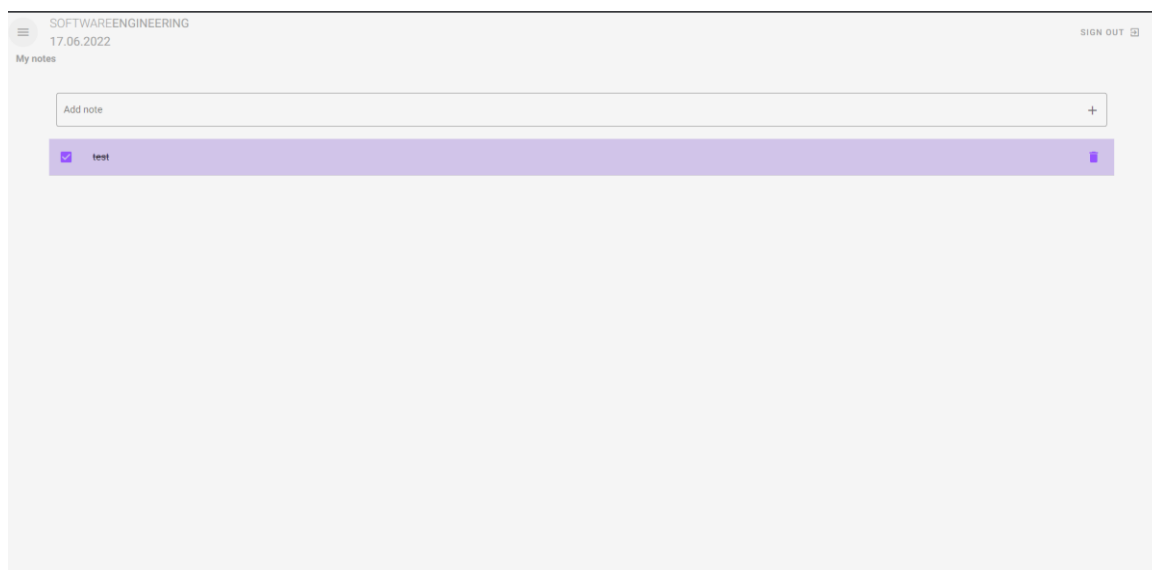


Рисунок 4.26 – Сторінка на мобільному пристрої

Оскільки замітки іноді необхідно додати дуже швидко, для зручності створено кнопку в меню, щоб можна було додати нотатки з будь-якої іншої сторінки.

Висновки до розділу 4

У четвертому розділі кваліфікаційної роботи бакалавра розглянуто процес розробки вебзастосунку для ведення розкладу занять.

Розглянуто технології, які використано під час розробки застосунку для виконання поставлених задач.

В розділі розглянуто процеси налаштування проєкту та середовища для розробки, процес розробки функції з демонстрацією важливих модулів коду вебзастосунку.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи бакалавра здійснена розробка вебзастосунку для ведення розкладу занять. Для виконання роботи відповідно до мети визначено завдання, які успішно виконані.

Під час виконання кваліфікаційної роботи бакалавра проаналізовано предметну область і поставлено задачу. В ході проектування системи застосунку для ведення розкладу виконано детальний аналіз аналогічних систем. На основі отриманої інформації сформовано вимоги до програмного забезпечення, що розроблюється. Сформовано специфікацію вимог до програмного забезпечення, що представляє собою опис поведінки програмного забезпечення, що розроблюється. Проведено моделювання об'єкту та предмету дослідження, під час якого визначені дійову особу застосунку, побудовано діаграму варіантів використання, описано сценарії варіантів використання застосунку. Під час виконання моделювання системи проведено моделювання взаємодії між об'єктами завдяки за допомогою діаграми послідовності. Розроблено діаграму класів.

Детально розглянуто технології, які використовувалися для успішного виконання кваліфікаційної роботи бакалавра. Детально описано процес виконання роботи та створення застосунку.

Отриманий у результаті реалізації вебзастосунок має достатньо зручний інтерфейс та функціонал.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. iSpring: вебсайт. URL: <https://www.ispringsolutions.com> (дата звернення 18.05.2022).
2. Moodle: вебсайт. URL: <https://moodle.org/> (дата звернення 18.05.2022).
3. Teachbase: вебсайт. URL: <https://teachbase.ru/> (дата звернення 18.05.2022).
4. Waterfall model: вебсайт. URL: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm (дата звернення 18.05.2022).
5. V-model: вебсайт. URL: <https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/> (дата звернення 18.05.2022).
6. Incremental Model: вебсайт. URL: <https://www.javatpoint.com/software-engineering-incremental-model> (дата звернення 18.05.2022).
7. Agile Model: вебсайт. URL: https://www.tutorialspoint.com/sdlc/sdlc_agile (дата звернення 18.05.2022).
8. BPwin: вебсайт. URL: <http://bourabai.ru/cm/bpwin.htm> (дата звернення 08.06.2022).
9. StartUML: вебсайт. URL: <https://staruml.io/> (дата звернення 10.06.2022).
10. WebStorm: вебсайт. URL: <https://www.jetbrains.com/webstorm/> (дата звернення 10.06.2022).
11. VueJS: вебсайт. URL: <https://stfalcon.com/ru/blog/post/vue-js-guide-to-tech> (дата звернення 10.06.2022).
12. Vue-Router: вебсайт. URL: https://rukovodstvo.net/posts/id_1178/ (дата звернення 10.06.2022).
13. Chess: вебсайт. URL: <https://www.chess.com/> (дата звернення 10.06.2022).

14. Codeship: вебсайт. URL: <https://app.codeship.com/> (дата звернення 10.06.2022).

15. Livestorm: вебсайт. URL: <https://livestorm.co/> (дата звернення 11.06.2022).

16. Vuetify: вебсайт. URL: <https://vuetifyjs.com/en/> (дата звернення 11.06.2022).

17. Firebase: вебсайт. URL: <https://firebase.google.com/> (дата звернення 11.06.2022).