Міністерство освіти і науки України

Одеський національний політехнічний університет

Навчально-науковий інститут комп’ютерних систем

Кафедра системного програмного забезпечення

Паршин Ілля Андрійович,

студент групи АС-151

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

Програма для аналізу відгуків студентів щодо якості навчального процесу з використанням статистичних методів обробки даних

Спеціальність:

121 – Інженерія програмного забезпечення

Спеціалізація:

Інженерія програмного забезпечення

Керівник:

Тройніна Анастасія Сергіївна,

канд. техн. наук, доцент

Одеса - 2020

**ЗМІСТ**

Завдання [НА КВАЛІФІКАЦІЙНУ РОБОТУ 4](#_Toc58766139)

[ЗАВДАННЯ НА РОЗРОБКУ РОЗДІЛУ «ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ» 6](#_Toc58766140)

[Анотація 7](#_Toc58766141)

[Вступ 8](#_Toc58766142)

[1 АНАЛІЗ предметної області і аналогів 10](#_Toc58766143)

[1.1 Опис предметної області 10](#_Toc58766144)

[1.2 Опис аналогів системи 11](#_Toc58766145)

[2 Методи оцінювання якості навчального процесу 14](#_Toc58766146)

[2.1 Місце SET у навчальному процесі 14](#_Toc58766147)

[2.2 Метод оцінювання і аналізу результатів опитування 17](#_Toc58766148)

[3 Специфікація вимог до програмного продукту 20](#_Toc58766149)

[3.1 Функціональні вимоги до продукту 20](#_Toc58766150)

[3.2 Нефункціональні вимоги до продукту 31](#_Toc58766151)

[4. Проектування програмної системи 33](#_Toc58766152)

[4.1 Проектування архітектури системи 33](#_Toc58766153)

[4.2 Проектування структури і організації класів 35](#_Toc58766154)

[4.3 Проектування бази даних 40](#_Toc58766155)

[4.4 Проектування інтерфейсу користувача 42](#_Toc58766156)

[5 Програмна реалізація системи і тестування 51](#_Toc58766157)

[5.1 Набір інструментальних засобів розробки 51](#_Toc58766158)

[5.2 Розробка функціональних тестів 54](#_Toc58766159)

[6 приклад використання програмної системи і експеримент 62](#_Toc58766160)

[6.1 Приклад використання 62](#_Toc58766161)

[6.2 Тестування зміни рівня якості навчального процесу з використанням розробленої системи 67](#_Toc58766162)

[7 охорона праці 72](#_Toc58766163)

[Висновки 74](#_Toc58766164)

[Список використаних джерел 75](#_Toc58766165)

[Додаток А. МАТЕРІАЛИ ТЕЗИСІВ 77](#_Toc58766166)

[Додаток Б. ЛІСТИНГ ПРОГРАМИ 79](#_Toc58766167)

[Додаток В. ПИТАННЯ З ОХОРОНИ ПРАЦІ ТА БЕЗПЕКИ У НАДЗВИЧАЙНИХ СИТУАЦІЯХ 93](#_Toc58766168)

[В.1 Охорона праці. Організація та управління 93](#_Toc58766169)

[В.2 Визначення основних параметрів робочого місця користувача ЕВМ 96](#_Toc58766170)

[В.3Розрахунок занулення 98](#_Toc58766171)

[В.4 Безпека під час надзвичайних ситуації техногенного характеру. 101](#_Toc58766172)

[В.5 Заходи з охорони праці та безпеки у надзвичайних ситуаціях. 102](#_Toc58766173)

[В.6 Безпека у суспільстві в умовах загрози COVID-19. 104](#_Toc58766174)

Міністерство освіти і науки України

Одеський національний політехнічний університет

Навчально-науковий інститут комп’ютерних систем

Кафедра системного програмного забезпечення

Рівень вищої освіти: другий(магістерський)

Спеціальність: 121 – Інженерія програмного забезпечення

Спеціалізація: Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_ Крісілов В.А.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Паршина Іллі Андрійовича, група АС-151

1. Тема роботи: Програма для аналізу відгуків студентів щодо якості навчального процесу з використанням статистичних методів обробки даних.

Керівник роботи: Комлева Наталія Олегівна, канд. техн. наук, доцент

затверджені наказом ректора від «29»жовтня 2020 р. № 412-в.

2. Зміст роботи: аналіз предметної області і аналогів, методи оцінювання якості навчального процесу, специфікація вимог до програмного продукту, проектування програмної системи, програмні засоби реалізації системи, тестування і приклад використання програмної системи, охорона праці

3. Перелік ілюстративного матеріалу: Згідно слайдів презентації

4. Консультанти розділів роботи

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
| завдання видав | завдання прийняв |
| 7 | Москалюк А.Ю. | 02.09.20 | 24.11.2020 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

5. Дата видачі завдання: «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ р.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання | Примітка |
| 1 | Аналіз предметної області і існуючих рішень | 2.09.2020 – 14.09.2020 | вик. |
| 2 | Аналіз і розробка теоретичної частини програмної системи | 15.09.2020 –30.09.2020 | вик. |
| 3 | Специфікація вимог до програмної системи | 1.10.2020 – 7.10.2020 | вик. |
| 4 | Проектування програмної системи | 8.10.2020 – 14.10.2020 | вик. |
| 5 | Програмна реалізація та тестування системи | 15.10.2020 – 16.11.2020 | вик. |
| 6 | Завдання з охорони праці | 17.11.2020 – 24.11.2020 | вик. |
| 7 | Оформлення пояснювальної записки і супутнього графічного матеріалу | 25.11.2020 – 10.12.2020 | вик. |

Здобувач вищої освіти \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ І. А. Паршин

Керівник роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. С. Тройніна

ЗАВДАННЯ НА РОЗРОБКУ РОЗДІЛУ «ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ»

Паршина Іллі Андрійовича, група АС-151

Навчально-науковий інститут комп’ютерних систем

Кафедра системного програмного забезпечення

Тема роботи: Програма для аналізу відгуків студентів щодо якості навчального процесу з використанням статистичних методів обробки даних.

Зміст розділу:

1. Організація та управління охороною праці на робочому місці менеджера по роботі з клієнтами.
2. Визначення основних параметрів робочого місця користувача ЕВМ.
3. Розрахунок занулення.
4. Безпека під час надзвичайних ситуації техногенного характеру.
5. Заходи з охорони праці та безпеки у надзвичайних ситуаціях.
6. Безпека у суспільстві в умовах загрози COVID-19.

Керівник роботи Консультант з охорони праці та БНС

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.С. Тройніна \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Ю. Москалюк

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 р. «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 р.

Анотація

Метою роботи є підвищення якості навчального процесу шляхом розробки ефективного і зручного інструменту для збору і аналізу відгуків студентів щодо навчального процесу. Технологіями розробки є мова програмування Javascript і її бібліотеки - React і Redux для написання клієнтської частини системи, JavaScript-оточення Node.js і фреймворк Express.js для написання серверної частини системи. Для зберігання даних використовується реляційна система управління базами даних MySQL. Як результат роботи виконана програмна реалізація системи що надає гнучкі інструменти для створення і редагування анкет, проведення анкетування і подальшого аналізу отриманих даних з використанням статистичних методів обробки даних.

Ключові слова: навчальний процес, обробка даних, анкета, Javascript, React, Redux, Node.js, Express.js, MySQL.

**ABSTRACT**

The aim of the work is to improve the quality of the educational process by developing an effective and convenient tool for collecting and analyzing student feedback on the educational process. The development technologies are the Javascript programming language and its libraries - React and Redux for writing the client part of the system, JavaScript-environment Node.js and the Express.js framework for writing the server part of the system. MySQL database management system is used for data storage. As a result of the work the software implementation of the system was developed, which provides flexible tools for creating, editing and actually conducting surveys and further analysis of the obtained data using statistical methods of data processing.

Key words: educational process, data processing, survey, Javascript, React, Redux, Node.js, Express.js, MySQL.

Вступ

Протягом багатьох років найбільш поширеним процесом, що використовувався для покращення навчального процесу у вищих навчальних закладах було оцінювання його студентами (student evaluation of teaching - SET). Такий спосіб оцінювання навчального процесу дозволяє отримати відгуки від респондентів, що є об’єктами навчального процесу, тому інформація, отримана від них, є найбільш достовірною і актуальною, що в свою чергу дає змогу прийняти найефективніші методи і прийоми для покращення якості навчального процесу.

Найчастіше для SET використовується анкетування. Такий спосіб збору інформації є дешевим в реалізації, адже потребує лише складання самої анкети, друку бланків і власне проведення анкетування.

Однак такий спосіб має ряд недоліків, перелічених нижче.

1. Використання паперових носіїв може ускладнити проведення анкетування через необхідність поширення бланків серед студентів, їх подальший збір, можливу втрату деяких бланків через людський фактор.

2. Обробка даних, отриманих з паперових носіїв є трудомістким процесом, що може зайняти багато часу.

3. Корисність зібраних під час анкетування даних напряму залежить від якості її складання, адже питання, включені до неї, мають бути точними, простими, лаконічними, зрозумілими і мати однозначне трактування.

4. Питання анонімності. Перевагою анонімних анкет є вільне висловлення думок, але з анонімності випливають значні недоліки - можливість багаторазового заповнення анкети однією особою, труднощі збереження анонімності в малих групах. Неанонімне анкетування забезпечує таку додаткову перевагу, як кореляція з параметрами профілю студента, зокрема, успішністю, відвідуваністю, науково-дослідницькою роботою. Але недоліком неанонімного анкетування є людський фактор, адже студенти, знаючи, що анкетування неанонімне, можуть надати інформацію, що не відповідає дійсності, аби уникнути конфлікту з викладачем.

Отже, метою роботи є підвищення якості навчального процесу шляхом розробки ефективного і зручного інструменту для збору і аналізу відгуків студентів щодо навчального процесу.

Під час розробки даного програмного продукту будуть вирішені такі задачі:

1) розробка моделі підвищення якості навчального процесу у закладах вищої навчальної освіти;

2) розробка програмної системи, що використовує розроблену модель підвищення якості навчального процесу.

Використання створеного програмного продукту на основі розробленої моделі дозволить внести необхідні зміни до навчального процесу, що в подальшому позначиться на успішності студентів.

Був проведений аналіз моделі покращення якості навчального процесу, результати опубліковано у збірнику доповідей XIIІ міжнародної науково-практичної конференція «Інформаційні технології і автоматизація – 2020» [1].

1 АНАЛІЗ предметної області і аналогів

1.1 Опис предметної області

Предметна область – процес покращення якості навчального процесу у закладах вищої освіти, а саме збір і аналіз зворотнього зв’язку студентів щодо навчального процесу.

Якість навчального процесу значною мірою залежить від уміння викладацького складу узгодити інтереси учасників навчального процесу з навчальною програмою іцілями конкретного курсу і знайти найбільш прийнятні рішення. Принципова неможливість існування досконалих процедур попереднього узгодження рішень та доцільність випереджувального виявлення проблем обумовлюють необхідність існування відповідних інструментів зворотного зв’язку.

Такими інструментами є, у першу чергу, регулярні загальні і тематичні опитування, які дозволяють отримати максимально достовірну оцінку суб’єктивного бачення студентами різних аспектів якості навчального процесу на тому чи іншому курсі.

Проведення таких опитувань на постійній основі вимагає відповідного методичного (формування анкет, вибір оптимальних періодів опитувань, забезпечення валідності тощо), організаційного (виконавці, розподіл обов’язків, повноваження) та матеріального забезпечення.

Саме тому найкращим рішенням цієї проблеми буде розробка автоматизованої системи для збору і аналізу зворотнього зв’язку студентів щодо якості навчального процесу.

На момент розробки дипломного проекту існуючі програмні продукти, що використовуються для подібних цілей, не повністю задовольняють потреби користувачів, тому наразі організатори анкетування вимушені змиритися з їх недоліками або ж застосовувати технологічно застаріле і трудомістке анкетування з допомогою паперових носіїв. Рішенням цієї проблеми стане програмна система, що вирішить ключові проблеми обраної предметної області.

1.2 Опис аналогів системи

Більшість існуючих програмних рішень також є веб застосуваннями для створення і проходження анкет, що підтверджує твердження, що така реалізація подібних систем є найбільш зручною і доцільною.

Найбільш відомими і розповсюдженими аналогами даної системи є GoogleForms, SurveyMonkey, Survio і Typeform.

Необхідно виділити ключові ознаки, за якими буде вестися порівняння. Перша виділена ознака випливає з недоліків анонімного і неанонімного анкетування – можливість проведення анонімного анкетування і при цьому збереження можливості контролю за повторним проходженням анкети. Дана система призначена для використання у закладах вищої освіти, тому передбачається, що студент матиме проходити декілька анкет за короткий час (за різними дисциплінами або групами дисциплін, тощо), тому для зручності ці анкети мають бути агреговані в одному місці. Тому можливість доступу до анкет в одному місці буде однією з ключових ознак. Також важливою особливістю є наявність обмежень на створення і проходження анкет. Більшість існуючих програмних рішень орієнтовані на вирішення задач маркетологів, соціальних робітників та інших спеціалістів, які працюють з дуже широкою аудиторією клієнтів, тому очевидним є факт, що зберігання всіх даних про заповнені анкети є ресурсомістким, а отже і дорогим. Анкетування має застосовуватися в конкретній предметній області, тому в системі мають бути обчислення результатів анкетування з урахуванням параметрів, специфічних для даної предметної області, тому це також є важливою характеристикою програмного продукту. Останньою виділеною ознакою є ціна, адже існуючі сервіси є дорогими, якщо користуватися їх повним функціоналом. Заклади вищої освіти часто не мають таких ресурсів, тому ціна може стати визначальним фактором при остаточному виборі сервісу для проведення анкетування.

Нижче наведена порівняльна таблиця продукту, що розроблюється, з визначеними аналогами.

Таблиця 1.1 – Порівняння з аналогами

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Назва продукту | Ознаки | | | | |
| Анонімність з контролем повторного проходження | Агрегація анкет | Відсутність обмежень функціоналу в безкоштовній версії | Наявність специфічних для предметної області обчислень | Ціна |
| SurveyMonkey | - | - | до 10 питань в одній анкеті, до 100 респондентів на місяць в безкоштовній версії | - | безкоштовно, тарифи від 30 євро на місяць за розширений функціонал |
| Survio | - | - | до 100 респондентів на місяць | - | безкоштовно, тарифи від 16 доларів на місяць за розширений функціонал |
| Typeform | - | - | до 10 питань в одній анкеті, до 100 респондентів на місяць в безкоштовній версії | - | безкоштовно, тарифи від 35 доларів на місяць за розширений функціонал |
| GoogleForms | - | - | + | - | безкоштовно |
| OPU Surveys | + | + | + | + | Ціна за хостинг (від 4 доларів на місяць) |

Як бачимо з таблиці, продукт OPU Surveys є кращим за всіма визначеними характеристиками, що є визначальними для доцільності і успішної експлуатації цієї системи в обраній предметній області. Найближчим аналогом є GoogleForms, що наразі і використовується в більшості випадків, але продукт OPU Surveys є більш вузьконаправленим і надає функціонал, що є необхідним в конкретній предметній області.

**Висновок**

В даному розділі було виявлено ключові характеристики і ознаки для програмного продукту, що має використовуватися в даній предметній області. За цими ознаками було проведено порівняння з існуючими аналогами. В результаті аналізу було виявлено, що система, що розроблюється, має ряд переваг, а саме: можливість анонімного проходження анкет і контроль повторного проходження анкетування, можливість агрегації анкет, відсутність обмежень безкоштовної версії, що наявні в аналогічних системах, наявність специфічних для предметної області обчислень, що зумовлено її спеціалізацією для використання в даній предметній області, а також ціна, що є нижчою за аналоги. Таким чином дана система даватиме більше цінності і робитиме це дешевше за аналоги.

2 Методи оцінювання якості навчального процесу

2.1 Місце SET у навчальному процесі

Основна мета навчального процесу – перетворення навчальних цілей в навчальні результати в ході власне навчального процесу. Навчальні результати мають бути чітко і однозначно сформульовані у навчальному плані. Навчальний процес складається з трьох складових: зміст курсу, методи донесення інформації і матеріали курсу. Конкретна реалізація навчального процесу залежить від форми навчання (очна, заочна, дистанційне навчання тощо). Процес вивчення залежить від ступеня залучення студентів. В результаті оцінювання можна виявити рівень досягнень і успіхів що їх досягли студенти в результаті навчального процесу. Для оцінювання викладач використовує різні види завдань, а саме: теоретичне дослідження, вивчення і вирішення практичних задач, тестування, підготовка презентацій, тощо. В залежності від типів завдань вони можуть бути індивідуальними або груповими, мати різні часові обмеження та дозволяти або обмежувати вільний вибір інструментів для виконання. Розробка і критерії оцінювання виконаних завдань здійснюються викладачем згідно з навчальним планом.

«Незважаючи на широкий вибір видів завдань, з їх допомогою можна визначити лише спостережувані характеристики якості навчального процесу (*observed teaching quality characteristics* - *OTQC*). Однак неспостережувані характеристики навчального процесу (*unobserved teaching quality characteristics* - *UTQC*) також мають істотний вплив на навчальний процес. Таким чином набір характеристик навчального процесу (*teaching quality characteristics* - *TQC*) можна представити у наступному вигляді» [2]:

*TQC* = *OTQC*⋃*UTQC* (2.1)

Одним з найважливіших компонентів навчального процесу є зворотній зв’язок. Він використовується для впливу студентів на якість навчання, а також на оцінювання діяльності викладачів вищим керівництвом. Навчальний процес може розглядатися, як процес з негативним зворотнім зв’язком. Відхилення реальних значень елементів *TQC* (*TQCfact*) від рекомендованих (*TQCrec*) мають призводити до формування заходів з підвищення якості освіти (*quality improvement activities* - *QIA*), що мають на меті мінімізувати виявлені відхилення. Загалом *QIA* можуть бути представлені як інтегральний показник впливу на освітній процес у рамках обраної навчальної програми:

*QIA* = *F* (diff (*TQCfact*, *TQCstd*)) → min (2.2)

Таким чином навчальний процес можна представити у вигляді схеми, представленої нижче (рис. 2.1.1).

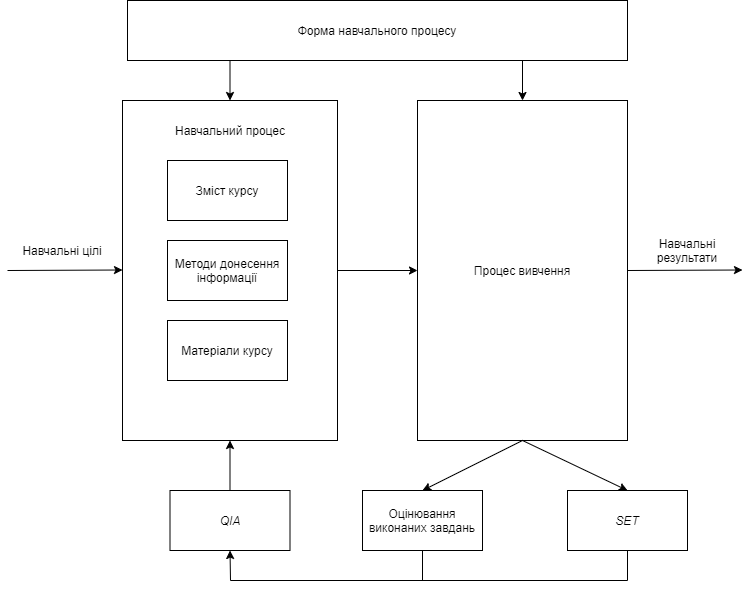


Рисунок 2.1.1 – Схема навчального процесу з відображенням циклу його покращення

Метод оцінювання *OTQC* є цілком зрозумілим: виконані студентами завдання оцінюються за кількісною шкалою за існуючими рекомендаціями.

«Оцінювання *UTQC* є більш складним. *UTQC* можна визначити за допомогою *SET* з використанням різноманітних технік, таких як опитування (анкетування), фокус групи, тощо. В ході цих заходів студенти надають зворотній зв’язок щодо змісту курсу, методів донесення інформації і матеріалів курсу» [2].

Зазвичай цільовими характеристиками *SET* є:

1) осмисленість – обізнаність, що курс дотримується навчального плану, розуміння ролі курсу в навчальному плані, задоволення власними успіхами протягом курсу, чітке і зрозуміле подання матеріалу, що допомагає розумінню зв’язків між частинами курсу, між даним курсом і іншими курсами в навчальному плані, а також розуміння методів оцінювання;

2) адаптивність – відповідність учбового навантаження вимогам курсу, прийняття швидкості подачі матеріалу;

3) ефективність – слідування розкладу, втрати часу, що викликані проблемами з доступом до навчальних матеріалів, своєчасність інформації про зміни в розкладі, допомога студентам у вивченні матеріалу, створення сприятливого для навчання середовища, своєчасне надання додаткових матеріалів і інструкцій, спонукання до виконання більш складних завдань в рамках курсу, бажання і можливість викладача зрозуміло і в повній мірі відповідати на поставлені запитання, своєчасне виконання завдань, своєчасне надання зворотнього зв’язку;

4) функціональність – узгодженість вимог викладача і його асистентів, вплив матеріалів на поліпшення знань і навичок студентів з предмету, доступність безкоштовних і зрозумілих матеріалів в бібліотеці університету і електронних бібліотеках, своєчасне оновлення електронних ресурсів курсу;

5) взаємодія – попередні знання про предмет вивчення, регулярність відвідування занять, ефективність організації і сприяння навчальним заходам викладачем;

6) активність – підготовка до занять, зусилля, спрямовані на виконання поставлених завдань.

Таким чином маємо набір неспостережуваних характеристик якості навчального процесу. Для того, щоб отримати найбільш повну і всебічну інформацію щодо якості навчального процесу, необхідно, щоб питання з анкети покривали всі ці характеристики. Єдиним обмеженням є час, адже рекомендована тривалість заповнення анкети – 10-15 хвилин. В разі якщо анкета потребує більше часу на заповнення,суттєво падає зацікавленість респондента, що впливає на якість зібраних даних.

2.2 Метод оцінювання і аналізу результатів опитування

Необхідно визначити шкалу оцінювання, за якою необхідно буде надати відповідь на кожне питання. У настановах Державного університету Айова (*Iowa State University*) щодо вибору інструментів *SET* зазначається, що найкращим підходом буде використання одного типу шкали, наприклад від 1 до 5, де 1 – «погано», 5 – «відмінно». Такий підхід допоможе при обробці отриманих даних, адже такі числові дані простіше обробляти за допомогою статистичних методів, а також для кожного питання будуть виконані аналогічні статистичні обчислення [3].

Конкретизуємо, які саме обчислення мають бути проведені для аналізу результатів анкетування. Як ми вже визначили вище, шкала, що буде використовуватися для надання відповідей матиме наступний вигляд: 1 – поганий результат, 2 – незадовільний результат, 3 – нейтральний результат, 4 – задовільний результат і 5 – гарний результат (семантичне значення оцінок може бути уточнене в тексті питання, тут наведена узагальнена інтерпретація оцінок системою) [4].

Для початку застосуємо міри центральної тенденції, а саме середнє арифметичне, моду і медіану.

Середнє арифметичне допоможе виявити найбільш типовий для вибірки результат і обчислюється за формулою:

, (2.3)

де *M* – середнє арифметичне,

*xi* – оцінка, виставлена студентом,

*n* – загальна кількість відповідей на питання.

«Мода (*Мо*) – значення, що найбільш часто зустрічається у вибірці. Якщо виникає ситуація, коли два сусідніх значення мають однакову частоту і їх частота більша за частоту будь-яких інших значень, мода обчислюється, як середнє арифметичне цих значень.

Медіана (*Ме*) – значення, що знаходиться в середині послідовності значень, якщо їх розмістити в порядку зростання чи спадання. Якщо ряд значень має парну кількість елементів, медіана обчислюється, як середнє арифметичне двох центральних значень ряду.

Значення моди і медіани корисні для того, щоб визначити, чи є розподіл значень ознаки, що вивчається, симетричним і наближується до нормального розподілу. Значення середнього арифметичного, моди і медіани зазвичай співпадають, або мало відрізняються одне від одного.» [5]

Для більш повного аналізу результатів анкетування необхідно використати міри розкиду даних, що характеризують ступінь індивідуальних відхилень від центральної тенденції. До таких мір відносяться дисперсія і стандартне відхилення.

Дисперсія характеризує, наскільки окремі значення відрізняються від середньої величини у даній вибірці. Чим більшою є дисперсія, тим більше відхилення чи розкид даних [5]. Дисперсія обчислюється за наступною формулою:

, (2.4)

де - дисперсія,

*xi* – оцінка, виставлена студентом,

*M* – середнє арифметичне,

*n* – загальна кількість відповідей на питання.

Стандартне відхилення є похідною величиною від дисперсії і дорівнює квадратному кореню дисперсії. Ця величина часто є більш зручною характеристикою варіювання, ніж дисперсія, адже виражається в тих самих одиницях вимірювання, що і дисперсія. Стандартне відхилення показує, що більшість відповідей на дане питання знаходиться в деяких межах від середнього [5].

Всі обчислені дані можна використовувати для аналізу результатів анкетування, щоб в подальшому використати їх для організації заходів з підвищення якості навчального процесу.

Тепер необхідно визначити проблеми, що можуть бути автоматично виявлені системою. Як бачимо зі шкали, значення 1 і 2 інтерпретуються, як негативні. Саме тому всі питання, середнє арифметичне значень яких буде менше 3, автоматично можна вважати проблемними і звертати увагу викладача на них при виведенні результатів. Також можлива ситуація, коли середнє арифметичне, мода і медіана не є приблизно однаковими. В такому випадку можна зробити висновок, що розподіл не є нормальним, тобто найбільш ймовірно, що оцінки студентів розділилися на дві чи більше груп на оціночній шкалі, що не є задовільним в рамках оцінювання якості навчального процесу і має розглядатися, як проблемна ситуація.

**Висновок**

В даному розділі було визначено місце SET в циклі покращення якості навчального процесу. Було виявлено, що метою SET є оцінювання неспостережуваних характеристик навчального процесу, що не можуть оцінюватися напряму через успішність студентів протягом курсу.Було виявлено і описано основні неспостережувані характеристики.

Був розроблений метод оцінки і аналізу результатів опитування, а саме: була обрана шкала, за якою респонденти будуть надавати відповіді на запитання, обрані статистичні розрахунки, що допоможуть проаналізувати отримані дані, а також описані проблемні ситуації, що можуть бути автоматично виявлені в ході розрахунку результатів анкетування.

3 Специфікація вимог до програмного продукту

Основна мета створення будь-якої програмної системи це створення програмного продукту, який допомагає користувачу виконувати свої повсякденні завдання. Для створення таких програм насамперед визначаються вимоги, яким повинна задовольняти система. Ці вимоги записуються в специфікацію.

«Специфікація вимог до програмного продукту (software requirements specification - SRS) – повний опис поведінки системи, що розробляється. Вона включає функціональні і нефункціональні вимоги до системи. Функціональні вимоги представляють собою множину прецедентів (варіантів використання)» [6]. Варіанти використання це - опис послідовності дій, які може здійснювати система у відповідь на зовнішні дії користувачів або інших програмних систем.Нефункціональні вимоги представляють собою набір обмежень, що накладаються на реалізацію програмного продукту. Нефункціональні вимоги стосуються таких аспектів програмного продукту як практичність, надійність, супроводжуваність, продуктивність та технічні обмеження [7].

3.1 Функціональні вимоги до продукту

Для даної системи було виділено два види користувачів (акторів): студенти (позначені, як «Студент» на діаграмі вище) і всі працівники кафедри, що мають відношення до організації навчального процесу. Для зручності будемо називати другу групу Викладачами.

Студенти можуть авторизуватися в системі за допомогою логіна і пароля, пройти доступні їм анкети і вийти з системи.

Викладачі мають широкий функціонал для створення і редагування анкет, створення питань, які будуть включені до цих анкет, відкриття і закриття доступу до анкети, щоб студенти могли проходити анкету лише деякий час. Після закриття анкети Викладач отримує результати анкетування, які зможе переглядати в майбутньому для перевірки наслідків прийнятих рішень щодо покращення навчального процесу.

В системі не передбачено самостійної реєстрації студентів. Таке рішення зумовлене тим, що для отримання достовірної і коректної інформації напряму залежить від респондентів, що проходять анкетування. При вільній реєстрації значно ускладнюється контроль за тим, хто зареєструвався в системі, чи дійсно він є студентом даного вузу і чи дійсно введені ним ім’я, прізвище і група є реальними. Саме тому реєстрація студентів в системі має проводитися викладачами або працівниками кафедри. Така схема наразі практикується в системі el.opu.ua і є цілком виправданою.

Нерідко відбувається ситуація, коли того чи іншого студента відраховують з університету. В такому разі необхідно видалити його з системи, щоб позбавити його можливості давати відгук на навчальний процес після відрахування. Ця функція доступна викладачу.

Результатом проведення анкетування і аналізу отриманих даних є заходи з підвищення якості навчального процесу, що мають на меті виправити недоліки, виявлені в ході опитування. Результати цих заходів мають бути перевірені для виявлення, чи дійсно обрані заходи виправили недоліки. Для цього в статистиці опитувань використовується регресійний аналіз. Він передбачає, що анкету без змін, або з мінімальними змінами студенти будуть проходити декілька разів. Таким чином будуть виявлені результати проведених заходів з підвищення якості навчального процесу. Саме тому Викладачу доступна функція «Створити копію існуючої анкети».

Для формалізації функціональних вимог до продукту, що розроблюється, було розроблено діаграму варіантів використання. Діаграма варіантів використання – це граф спеціального вигляду, який є графічною нотацією для представлення конкретних варіантів використання, акторів, можливо деяких інтерфейсів, і відносин між цими елементами [8]. При цьому окремі компоненти діаграми можуть бути поміщені в прямокутник, який позначає проектовану систему в цілому. Слід зазначити, що відносинами даного графа можуть бути тільки деякі фіксовані типи взаємозв'язків між акторами і варіантами використання, які в сукупності описують функціональні вимоги до модельованої системи. Діаграма використання для продукту, що розроблюється, представлена на рисунку 3.1.

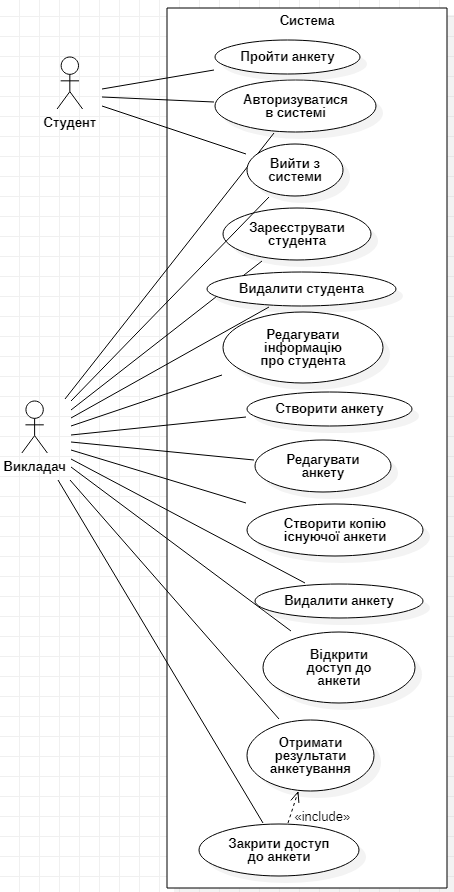


Рисунок 3.1 – Діаграма варіантів використання продукту

Для кожного варіанта використання з діаграми були прописані сценарії за шаблоном RUP [9], представлені нижче. У сценаріях «Авторизуватися в системі» і «Вийти з системи» семантично не є важливим, Студент це чи Викладач, адже цей функціонал у них спільний. Тому для зручності в цих сценаріях використовується загальний термін «Користувач».

1. Авторизуватися в системі.

Передумова: Користувач заходить на сторінку авторизації.

Мінімальні гарантії: Користувач не авторизований в системі.

Гарантії успіху: Користувач авторизований в системі.

Ініціатор: Користувач.

Основний успішний сценарій:

1. Користувач вводить свої аутентифікаційні дані, натискає кнопку «Авторизуватися».

2. Система перевіряє коректність введених даних.

3. Дані коректні. Система виводить користувачу функціонал, що відповідає його типу (Студент чи Викладач).

Альтернативні сценарії:

3.1 Дані некоректні. Система виводить повідомлення про некоректні дані користувачу.

2. Вийти з системи.

Передумова: Користувач авторизований в системі.

Мінімальні гарантії: Користувач вийшов з системи.

Гарантії успіху: Користувач вийшов з системи.

Ініціатор: Користувач.

Основний успішний сценарій:

1. Користувач натискає кнопку «Вийти з системи».

2. Система переводить користувача на екран авторизації.

Альтернативні сценарії: немає.

3. Пройти анкету.

Передумова: Користувач авторизований в системі як Студент.

Мінімальні гарантії: Студент не пройшов анкету.

Гарантії успіху: Студент пройшов анкету.

Ініціатор: Студент.

Основний успішний сценарій:

1. Студент обирає анкету зі списку доступних йому анкет.

2. Система переводить користувача на екран проходження анкети, виводить йому список питань, що входять в дану анкету.

3. Студент обирає відповіді, що на його думку є правильними і відображають дійсне положення речей.

4. Студент натискає кнопку «Зберегти відповіді».

5. Система успішно зберігає відповіді, виводить відповідне повідомлення Студенту.

6. Система переводить Студента на головний екран.

Альтернативні сценарії:

1.1 Студент вже пройшов дану анкету. Система не дає доступ до повторного проходження.

5.1 Помилка збереження. Система виводить відповідне повідомлення Студенту, переводить його на головний екран.

4. Створити анкету.

Передумова: Викладач авторизований в системі.

Мінімальні гарантії: Викладач не створив анкету.

Гарантії успіху: Викладач створив анкету.

Ініціатор: Викладач.

Основний успішний сценарій:

1. Викладач натискає кнопку «Створити анкету».

2. Система переводить користувача на екран створення анкети.

3. Викладач додає необхідну кількість питань, вводить коректний текст питань.

4. Викладач натискає кнопку «Зберегти анкету».

5. Система успішно зберігає анкету, виводить відповідне повідомлення Викладачу.

6. Система переводить Викладача на головний екран.

Альтернативні сценарії:

3.1 Викладач намагається ввести текст питання, що є довшим за дозволений. Система попереджає про некоректність введених даних.

5.1 Помилка збереження. Система виводить відповідне повідомлення Викладачу, переводить його на головний екран.

5. Редагувати анкету.

Передумова: Викладач авторизований в системі.

Мінімальні гарантії: Викладач не відредагував анкету.

Гарантії успіху: Викладач відредагував анкету.

Ініціатор: Викладач.

Основний успішний сценарій:

1. Викладач натискає кнопку «Список анкет».

2. Система переводить користувача на екран зі списком анкет.

3. Викладач обирає необхідну анкету, натискає кнопку «Редагувати анкету».

4. Система переводить користувача на екран редагування анкети.

5. Викладач редагує анкету, натискає кнопку «Зберегти».

6. Система успішно зберігає відредаговану анкету, виводить відповідне повідомлення Викладачу.

7. Система переводить Викладача на екран зі списком анкет.

Альтернативні сценарії:

3.1 Анкета є активною (проводиться на момент спроби редагування) або є закритою (тобто для неї вже виконані всі статистичні розрахунки, передбачені системою). Система виводить повідомлення про заборону редагування, залишає Викладача на сторінці зі списком анкет.

6.1 Помилка збереження. Система виводить відповідне повідомлення Викладачу, переводить його на екран зі списком анкет.

6. Створити копію існуючої анкети.

Передумова: Викладач авторизований в системі.

Мінімальні гарантії: Викладач створив копію анкети.

Гарантії успіху: Викладач створив копію анкети.

Ініціатор: Викладач.

Основний успішний сценарій:

1. Викладач натискає кнопку «Список анкет».

2. Система переводить користувача на екран зі списком анкет.

3. Викладач обирає необхідну анкету, натискає кнопку «Створити копію анкети».

4. Система створює копію анкети з позначкою, що це копія.

Альтернативні сценарії:

4.1 Помилка збереження. Система виводить відповідне повідомлення Викладачу, переводить його на список анкет.

7. Видалити анкету.

Передумова: Викладач авторизований в системі.

Мінімальні гарантії: Викладач видалив анкету.

Гарантії успіху: Викладач видалив анкету.

Ініціатор: Викладач.

Основний успішний сценарій:

1. Викладач натискає кнопку «Список анкет».

2. Система переводить користувача на екран зі списком анкет.

3. Викладач обирає необхідну анкету, натискає кнопку «Видалити».

4. Система видаляє обрану анкету, виводить відповідне повідомлення.

Альтернативні сценарії:

4.1 Помилка видалення. Система виводить відповідне повідомлення Викладачу, переводить його на список анкет.

8. Відкрити доступ до анкети.

Передумова: Викладач авторизований в системі.

Мінімальні гарантії: Викладач відкрив доступ до проходження анкети студентами.

Гарантії успіху: Викладач відкрив доступ до проходження анкети студентами.

Ініціатор: Викладач.

Основний успішний сценарій:

1. Викладач натискає кнопку «Список анкет».

2. Система переводить користувача на екран зі списком анкет.

3. Викладач обирає необхідну анкету, натискає кнопку «Відкрити доступ».

4. Система виводить список груп.

5. Викладач обирає групи, студентам яких буде доступна обрана анкета.

6. Система відкриває доступ студентам для проходження обраної анкети, виводить відповідне повідомлення.

Альтернативні сценарії:

3.1 Анкета вже є активною (проводиться на момент спроби відкриття) або є закритою (тобто для неї вже виконані всі статистичні розрахунки, передбачені системою). Система виводить повідомлення про неможливість відкриття, залишає Викладача на сторінці зі списком анкет.

4.1 Викладач залишив список груп порожнім. Система виводить відповідне повідомлення.

6.1 Помилка збереження. Система виводить відповідне повідомлення Викладачу, переводить його на список анкет.

9. Закрити доступ до анкети.

Передумова: Викладач авторизований в системі.

Мінімальні гарантії: Викладач закрив доступ до проходження анкети студентами.

Гарантії успіху: Викладач закрив доступ до проходження анкети студентами.

Ініціатор: Викладач.

Основний успішний сценарій:

1. Викладач натискає кнопку «Список анкет».

2. Система переводить користувача на екран зі списком анкет.

3. Викладач обирає необхідну анкету, натискає кнопку «Закрити доступ».

4. Система закриває доступ студентам для проходження обраної анкети, розраховує всі необхідні показники за зібраними даними, виводить відповідне повідомлення Викладачу.

Альтернативні сценарії:

3.1 Анкета не є активною (не проводиться на момент спроби закриття анкети) або вже є закритою (тобто для неї вже виконані всі статистичні розрахунки, передбачені системою). Система виводить повідомлення про неможливістьзакриття доступу, залишає Викладача на сторінці зі списком анкет.

4.1 Помилка збереження. Система виводить відповідне повідомлення Викладачу, переводить його на список анкет.

10. Отримати результати анкетування.

Передумова: Викладач авторизований в системі, в системі є закриті анкети.

Мінімальні гарантії: Викладач отримав результати анкетування.

Гарантії успіху: Викладач отримав результати анкетування.

Ініціатор: Викладач.

Основний успішний сценарій:

1. Викладач натискає кнопку «Список анкет».

2. Система переводить користувача на екран зі списком анкет.

3. Викладач обирає необхідну закриту анкету, натискає кнопку «Переглянути результати».

4. Система виводить результати анкетування за обраною анкетою.

Альтернативні сценарії:

4.1 Анкета не є активною (не проводиться на момент спроби отримання результатів анкетування) або є активною (проводиться на момент спроби отримання результатів анкетування). Система виводить повідомлення про неможливість показу результатів, залишає Викладача на сторінці зі списком анкет.

11. Зареєструвати студента.

Передумова: Викладач авторизований в системі.

Мінімальні гарантії: Викладач не зареєстрував студента.

Гарантії успіху: Викладач зареєстрував студента.

Ініціатор: Викладач.

Основний успішний сценарій:

1. Викладач натискає кнопку «Список студентів».

2. Система виводить список студентів Викладачу.

3. Викладач натискає кнопку «Зареєструвати студента».

4. Система переводить користувача на форму для заповнення даних студента (ФІО, група, середній бал).

5. Викладач заповнює дані студента, натискає кнопку «Зберегти».

6. Система успішно зберігає дані про студента, виводить відповідне повідомлення Викладачу.

7. Система генерує логін і пароль для авторизації студента, зберігає їх.

Альтернативні сценарії:

4.1 Викладач вводить некоректні дані. Система виводить відповідне повідомлення Викладачу, залишає його на сторінці введення інформації про студента.

6.1 Помилка збереження. Система виводить відповідне повідомлення Викладачу, переводить його на список студентів.

12. Редагувати інформацію про студента.

Передумова: Викладач авторизований в системі, в системі є хоча б один зареєстрований студент.

Мінімальні гарантії: Викладач не відредагував інформацію про студента.

Гарантії успіху: Викладач відредагував інформацію про студента.

Ініціатор: Викладач.

Основний успішний сценарій:

1. Викладач натискає кнопку «Список студентів».

2. Система виводить список студентів Викладачу.

3. Викладач обирає необхідного студента, натискає кнопку «Редагувати».

4. Система переводить користувача на форму для редагування даних студента.

5. Викладач редагує дані студента, натискає кнопку «Зберегти».

6. Система успішно зберігає дані про студента, виводить відповідне повідомлення Викладачу.

Альтернативні сценарії:

5.1 Викладач вводить некоректні дані. Система виводить відповідне повідомлення Викладачу, залишає його на сторінці редагування інформації про студента.

6.1 Помилка збереження. Система виводить відповідне повідомлення Викладачу, переводить його на список студентів.

13. Видалити інформацію про студента.

Передумова: Викладач авторизований в системі, в системі є хоча б один зареєстрований студент.

Мінімальні гарантії: Викладач не відредагував інформацію про студента.

Гарантії успіху: Викладач відредагував інформацію про студента.

Ініціатор: Викладач.

Основний успішний сценарій:

1. Викладач натискає кнопку «Список студентів».

2. Система виводить список студентів Викладачу.

3. Викладач обирає необхідного студента, натискає кнопку «Видалити».

4. Система видаляє обраного студента, виводить відповідне повідомлення.

Альтернативні сценарії:

4.1 Помилка видалення. Система виводить відповідне повідомлення Викладачу, переводить його на список студентів.

3.2 Нефункціональні вимоги до продукту

За результатами аналізу даної системи були виявлені і задокументовані нефункціональні вимоги до системи, представлені нижче.

1. Практичність (Usability).
   1. Система має бути інтуїтивно зрозумілою користувачу.
   2. Система не має потребувати, або потребувати мінімум первинної підготовки для досягнення користувачем мінімальної продуктивності.
   3. Система має мати стриманий і простий інтерфейс користувача, адже вона в першу чергу спрямована на виконання своєї конкретної функції.
2. Надійність (Realiability).
   1. Система має бути доступною 95% часу.
   2. 90% збоїв мають бути ліквідовані протягом 2 годин.
3. Продуктивність (Performance).

Система не є сильно навантаженою, тому вимоги до її продуктивності є порівнянними з відповідними показниками будь-якого веб застосування.

* 1. В середньому час відповіді для транзакції має становити 2-3 секунди, максимальний час – 7 секунд.
  2. Система має підтримувати пропускну здатність не менше 100 транзакцій в секунду.

1. Можливість обслуговування (Supportability).

Дана система має бути легко модифікованою для додавання нових функцій, розвитку і покращення існуючого функціоналу.

5. Обмеження (Constraints).

Веб складова даної системи є веб застосуванням, тому вона може відкриватися у будь-якому браузері. Серверна частина має бути розгорнута на серверах, що підтримують запуск Node.js.

**Висновок**

В даному розділі за результатами аналізу системи були сформульовані функціональні і нефункціональні вимоги. Функціональні вимоги представлені у вигляді діаграми і опису сценаріїв варіантів використання.

4 Проектування програмної системи

4.1 Проектування архітектури системи

Дана програмна система являє собою веб-застосування і серверну частину, що взаємодіє з базою даних, що в сукупності надають гнучкі інструменти для створення і редагування анкет, проведення анкетування і подальшого аналізу отриманих даних з використанням статистичних методів обробки даних.

Високорівнева структура системи складається з трьох частин, а саме: WebApplication, Server і Database. Зв’язки між ними показані на рисунку 4.1.



Рисунок 4.1 – Високорівнева архітектура системи

Як видно з діаграми, структура мого продукту є клієнт-серверною, в якій клієнтом є веб-застосування.

Частина WebApplication має засоби управління всім функціоналом користувача, містить інтерфейс, з яким взаємодіє користувач, виводить необхідні дані користувачу і реалізує зв’язок з сервером.

Частина Server реалізує обмін даними між базою даних і веб-застосуванням, а також виконує розрахунки результатів анкетування після закриття анкети.

Частина Database є базою даних, в якій зберігаються всі дані системи: інформація про користувачів, анкети і результати минулих анкетувань.

Веб застосування побудоване за технологією SPA (односторінкове застосування).

На відміну від традиційного підходу, коли на кожну дію користувача відбувається перезавантаження і чергове виведення сторінки, в односторінковому застосуванні завантаження програми в браузер відбувається один раз і далі все взаємодія з сервером йде у фоновому режимі, у даному випадку за допомогою технології Virtual DOM. Тобто, коли користувач працює з елементами управління, не відбувається повного перезавантаження сторінки. Необхідна порція інформації підвантажується і оновлюються конкретні елементи сторінки (вузли DOM-дерева). Різниця в підходах зображена на рисунках 4.2, 4.3.

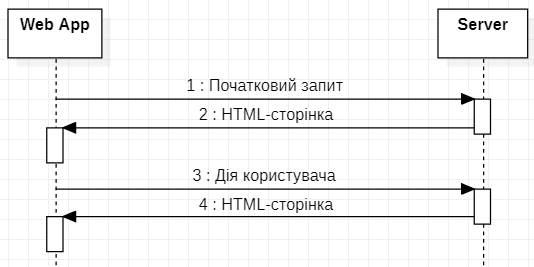


Рисунок 4.2 – Традиційний підхід до взаємодії веб-застосувань з сервером

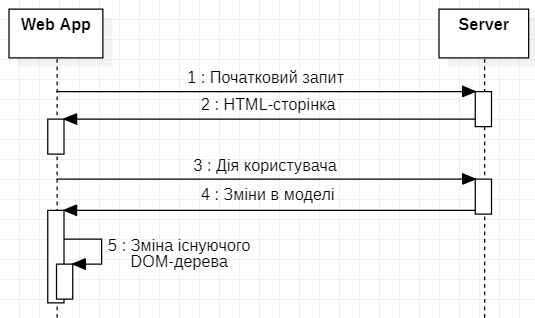


Рисунок 4.3 – Підхід SPA

4.2 Проектування структури і організації класів

Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. Діаграма класів може зображати, зокрема, різні взаємозв'язки між окремими сутностями предметної області, такими як об'єкти і підсистеми, а також описує їх внутрішню структуру і типи відношень. На даній діаграмі не вказується інформація про часові аспекти функціонування системи. З цієї точки зору діаграма класів є подальшим розвитком концептуальної моделі проектованої системи [10].

Діаграма класів представляє собою деякий граф, вершинами якого є елементи типу «клас», які зв'язані різними типами структурних відношень. Варто зауважити, що діаграма класів може також містити інтерфейси, пакети, відношення і навіть окремі екземпляри, такі як об'єкти і зв'язки. Коли говорять про дану діаграму, мають на увазі статичну структурну модель проектованої системи. Тому діаграму класів прийнято вважати графічним представленням таких структурних взаємозв'язків логічної моделі системи, які не залежать від часу.

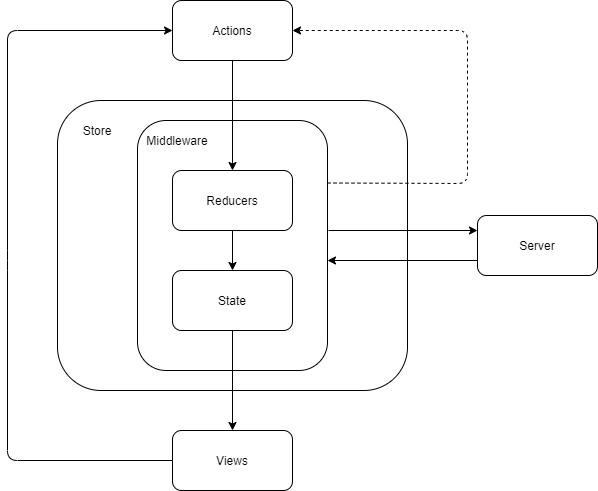
Діаграма класів складається з множини елементів, які в сукупності зображають декларативні знання про предметну область. Ці знання інтерпретуються в базових поняттях мови UML, таких як класи, інтерфейси і відношення між ними і їх складовими компонентами. При цьому окремі компоненти цієї діаграми можуть утворювати пакети для представлення більш загальної моделі системи. Якщо діаграма класів є частиною деякого пакету, то її компоненти повинні відповідати елементам цього пакету[10].

В результаті аналізу вимог було прийнято рішення розбити програмну систему на компоненти з визначеними залежностями між ними. Залежність між двома компонентами означає, що один компонент надає сервіс іншому компоненту.

Як зазначалося раніше, система розділена на дві частини – веб-застосування і сервер.

Веб застосування побудоване за архітектурою Flux.

Модель розробленої програмної системи показано на рис. 4.4.

Рисунок 4.4 – Модель функціонуваннявеб застосування

Клієнтська частина веб-застосування розділена на такі частини:

1) Views – відображення даних в системі. Це всі компоненти React, що створені в програмній системі і будуть відображати дані на інтерфейсі користувача, наприкладформа логіну, форма створення запису про студента, форма редагування питання в анкеті тощо. Вони повторно рендеряться, коли Store змінює стан.

2) Store – сховище даних, з якого Views отримують дані для виведення.

3) Reducers – визначають, як стан застосування зміниться у відповідь на Action, тобто є тимчасовим сховищем для змін в даних. Об’єкт Reducer неможливо змінити, лише створити знову. В ньому зберігаються локальні дані інтерфейсу. Наприклад, інформація про користувача, що авторизувався в системі.

4) Actions – це об’єкти, що містять ідентифікатор елемента, який ми хочемо змінити і інформацію, яка буде оновлена в Store. Створює нове значення в об’єкті reducer.

5) Middleware – надають сторонню точку розширення між відправкою Action і моментом, коли Action досягає об’єкту Reducer. Middlewareобробляє тіAction, які відправляють запит на сервер та передає результати запиту в Action. Таким чином в Reducer отримується і сам Action, і дані, необхідні для його обробки.

Сервер побудований за архітектурою MVC.

«Модель–вигляд–контролер (або Модель–представлення–контролер, англ. Model-view-controller, MVC) - архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення» [10].

Модель (Model) - надає собою об'єктну модель якоїсь предметної області, включає в себе дані і методи роботи з цими даними, реагує на запити з контролера, повертаючи дані і/або змінюючи свій стан, при цьому модель не містить в собі інформації, як дані можна візуалізувати, а також не "спілкується" з користувачем безпосередньо.

Подання (View) - відповідає за відображення інформації (візуалізацію), одні і ті ж дані можуть представлятися різними способами, наприклад, колекцію об'єктів за допомогою різних видів можна уявити як в табличному вигляді, так і списком.

Контролер (Controller) - забезпечує зв'язок між користувачем і системою, використовує модель і уявлення для реалізації необхідної реакції на дії користувача, як правило, на рівні контролера здійснюється фільтрація отриманих даних і авторизація (перевіряються права користувача на виконання дій або отримання інформації).

В сервері Моделлю є класи, об’єкти яких є відображенням таблиць бази даних. Це реалізовано за допомогою технології ORM (Object-RelationalMapping). Це технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних». ORM використовується для спрощення процесу збереження об'єктів в реляційну базу даних і їх вилучення, при цьому ORM сама піклується про перетворення даних між двома несумісними станами. Більшість ORM-інструментів значною мірою покладаються на метадані бази даних і об'єктів, так що об'єктам нічого не потрібно знати про структуру бази даних, а базі даних - нічого про те, як дані організовані в програмі. ORM забезпечує повне розділення завдань в добре спроектованих програмах, при якому і база даних, і додаток можуть працювати з даними кожен у своїй вихідній формі.

Контролерами в сервері є класи, що здійснюють доступ і управління даними моделі. В даному випадку це класи AuthController, UserController і SurveyController.

Видом в даному випадку є відображення даних у веб застосуванні.

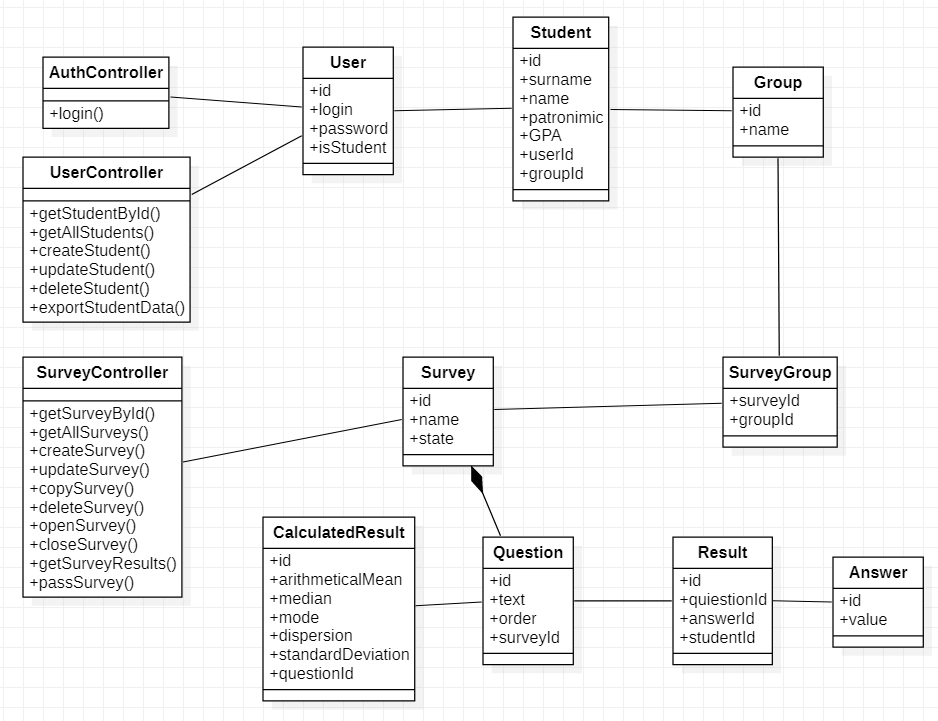


Рисунок 4.5 – Діаграма класів для серверної частини

На рисунку 4.5 представлена діаграма класів для серверної частини системи.

Опишемо отриману діаграму класів.

Клас AuthController – реалізує функції авторизації на сервері.

Методи класу:

login() – метод для авторизації користувача в системі.

Клас UserController – реалізує функції управління даними про студентів в базі даних.

Методи класу:

1) getUserById – отримання даних конкретного студента з бази даних за його унікальним ідетифікатором.

2) getAllStudents – отримати список всіх студентів з бази даних.

3) createStudent – створити запис про нового студента в базі даних.

4) updateStudent – оновити запис про існуючого студента в базі даних.

5) deleteStudent – видалити запис про існуючого студента з бази даних.

6) exportStudentData – експорт даних студентів у документ формату .csv.

Клас SurveyController реалізує функції управління даними про анкети в системі.

Методи класу:

1) getSurveyById – отримання даних про конкретну анкети з бази даних за її унікальним ідентифікатором.

2) getAllSurveys – отримати список всіх анкет з бази даних.

3) createSurvey – створити запис про нову анкету в базі даних.

4)updateSurvey – оновити запис про існуючу анкету в базі даних.

5) copySurvey – створити копію існуючої анкети в базі даних.

6) deleteSurvey –видалити запис про існуючу анкету з бази даних.

7) openSurvey – відкрити анкету для проходження студентам.

8) closeSurvey – закрити анкету для проходження студентам.

9) getSurveyResults –отримати результати анкетування.

10) passSurvey – пройти анкету.

4.3 Проектування бази даних

Нижче представлена схема бази даних, розроблена для даної системи.

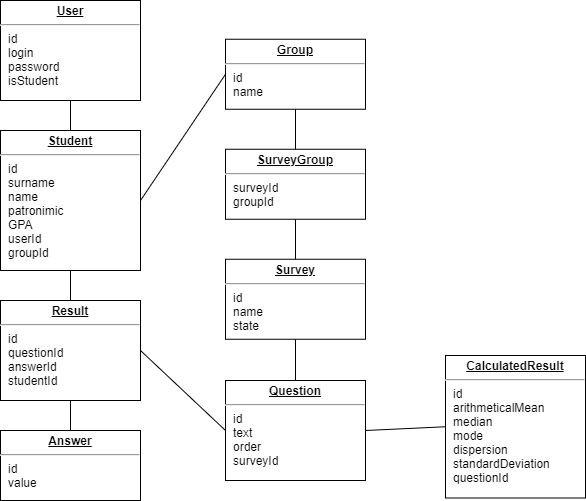


Рисунок 4.6– Структура бази даних

Таблиця User:

1) id – PRIMARY KEY, INT(10), унікальний ідентифікатор користувача;

2) login – VARCHAR(25), логін користувача;

3) password – VARCHAR(32), пароль користувача;

4) isStudent – BOOL, відображає, чи є користувач студентом.

Таблиця Student:

1) id – PRIMARY KEY, INT(10), унікальний ідентифікатор інформації про студента;

2) surname – VARCHAR(30), прізвище студента;

3) name – VARCHAR(20), ім’я студента;

4) patronymic - VARCHAR(20), по-батькові студента;

5) GPA – FLOAT, середній бал студента;

6)userId - FOREIGN KEY, INT(10), ідентифікатор користувача з isStudent = 1;

7)groupId – FOREIGN KEY, INT(10), ідентифікатор групи, до якої належить студент.

Таблиця Group:

1) id – PRIMARY KEY, унікальний ідентифікатор групи;

2) name - VARCHAR(7), ідентифікатор групи типу «АС-151».

Таблиця Survey:

1) id – PRIMARY KEY, унікальний ідентифікатор анкети;

2) name – VARCHAR(50), назва анкети.

3) state– ENUM(“new”, “open”, “closed”), стан анкети, приймає значення «нова», «відкрита», «закрита».

Таблиця SurveyGroup:

1) surveyId – FOREIGN KEY, INT(10), ідентифікатор анкети, яка доступна даній групі;

2) groupId – FOREIGN KEY, INT(10), ідентифікатор групи, студентам якої доступна дана анкета.

Таблиця Question:

1) id – PRIMARY KEY, унікальний ідентифікатор питання;

2) text – VARCHAR(150), текст питання;

3) order – INT(5), порядковий номер питання у анкеті;

4) surveyId – FOREIGN KEY, INT(10), ідентифікатор анкети, до якої входить дане питання

Таблиця Answer:

1) id – PRIMARY KEY, унікальний ідентифікатор відповіді;

2) value – INT, вага відповіді, може приймати цілі значення від 1 до 5.

Таблиця Result:

1) id – PRIMARY KEY, унікальний ідентифікатор відповіді студента;

2) questionId – FOREIGN KEY, INT(10), ідентифікатор питання, на яке була дана відповідь;

3) answerId – FOREIGN KEY, INT(10), ідентифікатор відповіді;

4) studentId – FOREIGN KEY, INT(10), ідентифікатор студента, який залишив відповідь.

Таблиця CalculatedResult:

1) id – PRIMARY KEY, унікальний ідентифікатор результатів, обчислених для питання за разультатами анкетування;

2) arithmeticalMean – FLOAT, середнє арифметичне відповідей на дане питання;

3) median – FLOAT, медіана даного набору даних;

4) mode – FLOAT, мода даного набору даних;

5) dispersion – FLOAT, дисперсія даного набору даних;

6) standardDeviation – FLOAT, стандартне відхилення даного набору даних;

7) questionId – FOREIGN KEY, INT(10), ідентифікатор питання.

4.4 Проектування інтерфейсу користувача

Цільовою аудиторією даної програмної системи є працівники кафедри, викладачі, а також студенти. Сама система не буде в постійному використанні, адже проведення заходів для покращення навчального процесу, а саме збір відгуків студентів, зазвичай проводяться наприкінці навчального семестру. Беручи до уваги цей фактор можна зробити висновок, що інтерфейс даної програмної системи має бути простим, інтуїтивно зрозумілим, не має потребувати додаткового навчання для використання взагалі або необхідний обсяг такого навчання має бути мінімальним.

Для спрощення проектування і розробки було виконано прототипування інтерфейсу користувача.

«Прототипування - створення прототипу інтерфейсу, що відображає розташування інтерфейсних об'єктів і описує їх взаємодію. На відміну від технічного завдання, текст якого кожна із зацікавлених сторін може трактувати на свою сторону, прототип однозначний, дозволяє побачити і усвідомити прообраз майбутнього продукту і внести необхідні корективи на ранніх стадіях розробки. Вносити зміни в прототип дешевше і швидше, ніж в продукт на стадії реалізації» [11].

Для прототипування були обрані wireframe через те, що такий метод прототипування дозволяє швидко створювати макети інтерфейсів користувача, що передають загальне уявлення про нього, не заглиблюючись в деталі.

Wireframe (Вайрфрейм) - це образ дизайну низькою точності. Він повинен чітко показувати: основну групу контенту, структуру інформації, опис та базову візуалізацію взаємодії між інтерфейсом і користувачем. Вайрфрейми виглядають як набір сірих блоків. Wireframe - скелет дизайну, який повинен зображати кожну деталь фінального продукту. Вайрфрейми зазвичай використовуються як документація до проекту, тому що вони є статичним і фіксованим способом відображення інтерфейсу. їх потрібно відповідно описати (від короткого пояснення до складної технічної документації, якщо це необхідно). Однак вони можуть бути використані не за настільки формальному призначенням. Так як вони зручні і прості у використанні, вони корисні як швидкі начерки для внутрішньої комунікації в команді. Вайрфрейми рідко використовуються як матеріал для тестування, хоча вони можуть допомогти отримати зворотній зв'язок на початковому етапі дослідження.

Існує безліч сервісів для швидкого і зручного створення вайрфреймів. Для розробки інтерфейсу даної системи був використаний сервіс Moqups.

Для того, щоб увійти в систему, в ній необхідно авторизуватися. Це можна зробити на сторінці авторизації. На цій сторінці має бути форма для введення аутентифікаційних даних, а також кнопка для відправлення даних з цієї форми. Приблизний дизайн сторінки авторизації представлений на рисунку нижче.

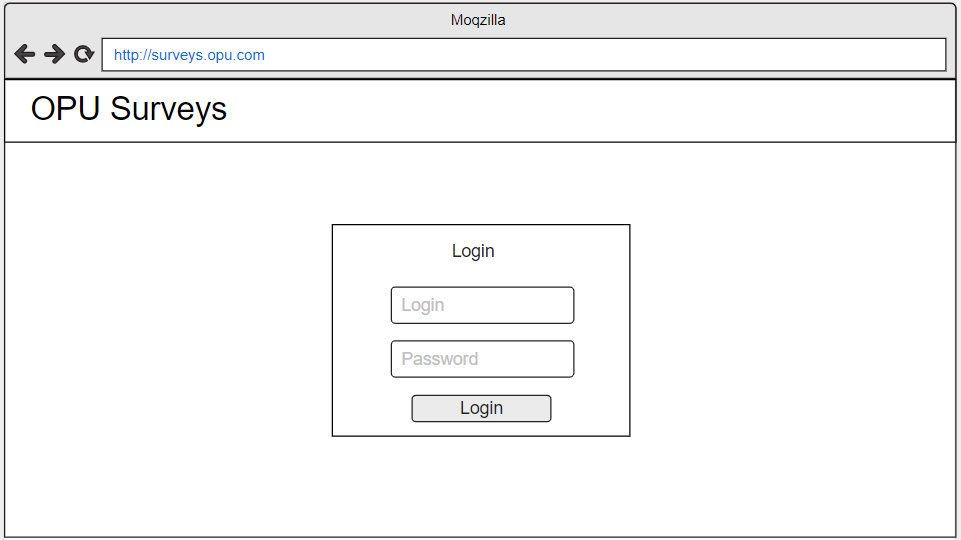


Рисунок 4.7 – Макет сторінки авторизації

Якщо користувач, що авторизувався, є студентом, то система виводить йому сторінку зі списком доступних йому анкет. Макет цієї сторінки представлено нижче.

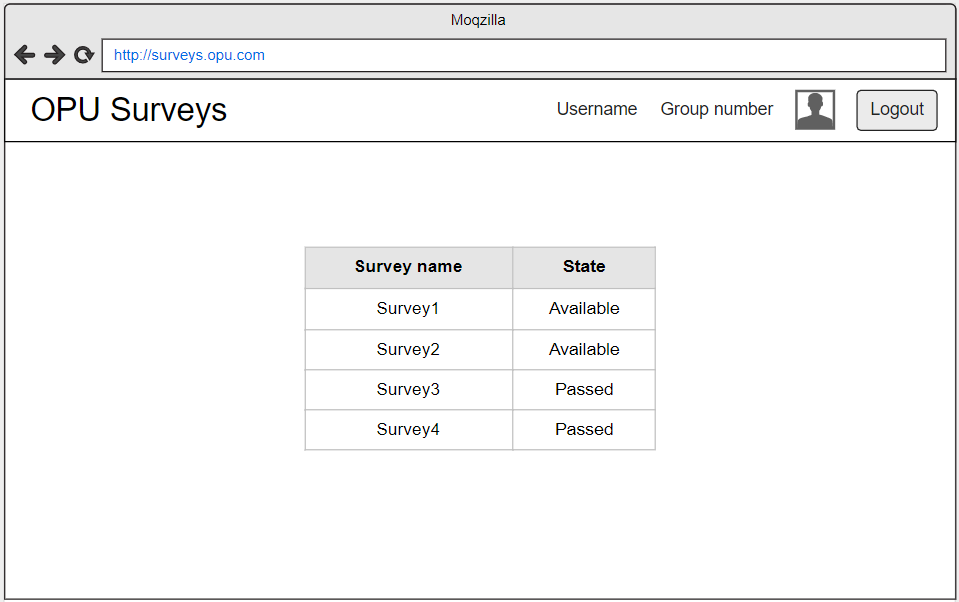


Рисунок 4.8 – Макет сторінки студента з інформацією про анкети

Список анкет подається у вигляді таблиці, в якій наведені дані про доступні анкети – назва анкети і її статус (пройдена анкета студентом чи ще ні). Також на цій сторінці виведене ФІО і номер групи студента, що авторизувався, і кнопка для виходу з системи, що переведе студента на сторінку авторизації.

Студент обирає доступну непройдену анкету. Система виводить йому сторінку з питаннями даної анкети. На цій сторінці є назва обраної анкети і кнопка для збереження обраних відповідей. Макет цієї сторінки представлений нижче.

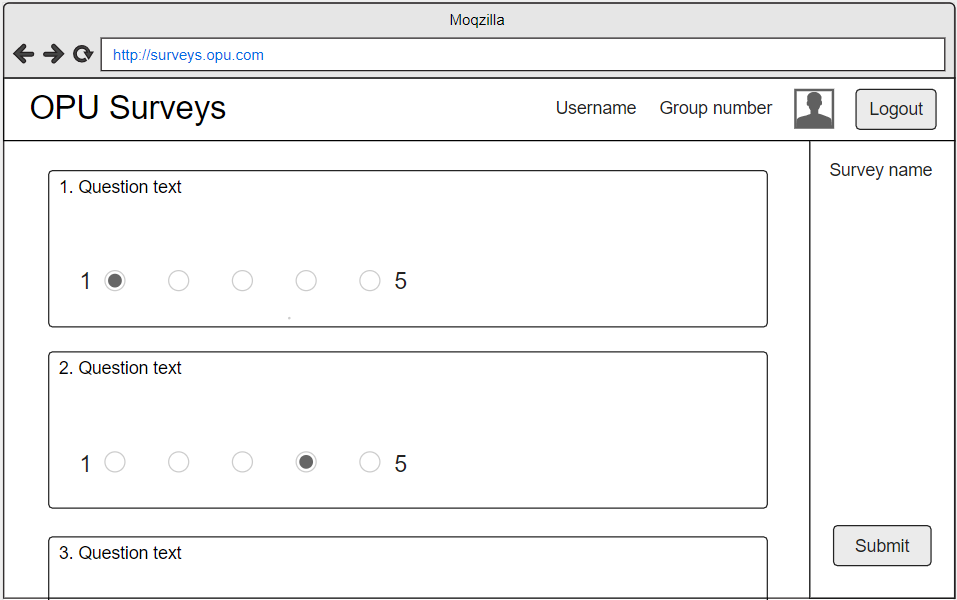


Рисунок 4.9 – Макет сторінки проходження анкети

Якщо користувач, що авторизувався, є викладачем, то система також виводить йому список анкет, але вона відрізняється від аналогічної у студента. Викладачу доступні такі функції як створення нової анкети, редагування і видалення існуючої анкети, відкриття анкети для проходження, закриття анкети. Дещо відрізняється і інформація про анкети в таблиці. У стовбці «State» мають відображатися такі статуси анкет:

1. New – тільки створена анкета.

2. Open – анкета, що була відкрита для проходження.

3. Closed – закрита анкета, для якої вже були обчислені всі результати.

На сторінці наявні елементи навігації, адже крім сторінки для управління анкетами викладачу доступна сторінка для управління інформацією про студентів. Макет описаної сторінки представлений нижче.

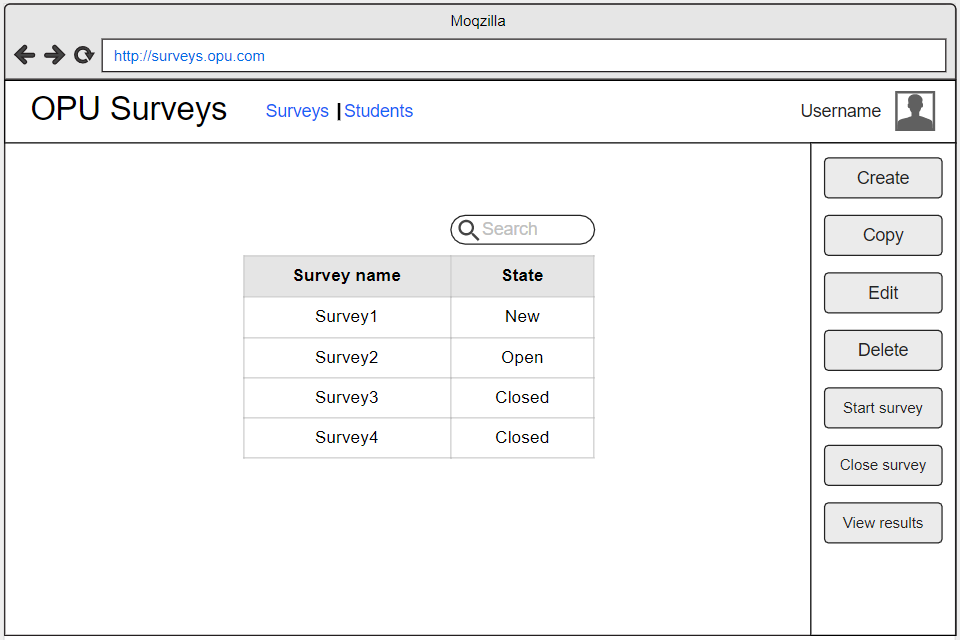


Рисунок 4.10 – Макет сторінки викладача з інформацією про анкети

Сторінка для управління інформацією про студентів буде мати подібний вигляд.

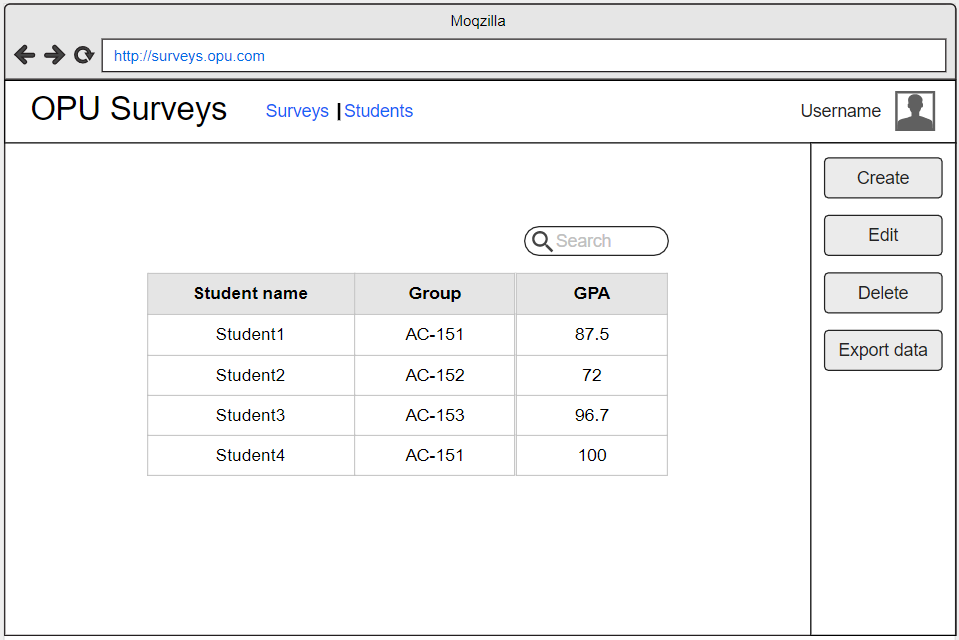


Рисунок 4.11 – Макет сторінки викладача з інформацією про студентів

Сторінка створення студента містить форму для заповнення всіх даних студента, що є важливими для функціонування системи, а саме: прізвище, ім’я, по батькові, ідентифікатор групи і середній бал студента. Також на сторінці наявні кнопка збереження даних, що створить запис про студента в системі, а також кнопка скасування, яка поверне викладача на сторінку перегляду списку студентів. Макет описаної сторінки представлений нижче.

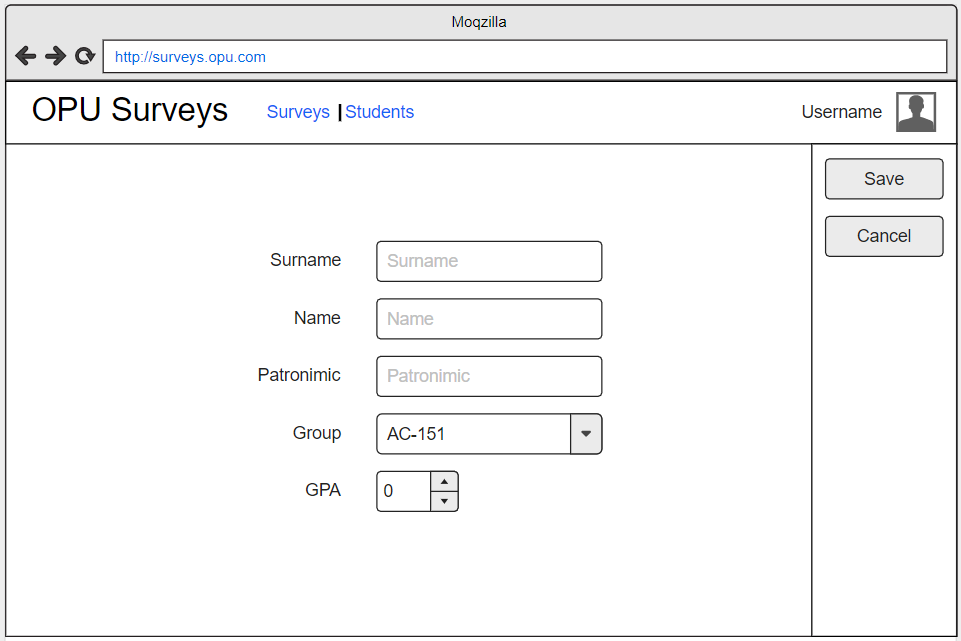


Рисунок 4.12 – Макет сторінки викладача для створення запису студента

Сторінка редагування інформації про студента матиме такий самий вигляд, але при її відкритті поля введення інформації про студента будуть вже заповнені його даними. Викладач зможе редагувати інформацію в цих полях, натиснути кнопку «Зберегти» для оновлення інформації про студента. Натискання на кнопку скасування переведе викладача на сторінку перегляду списку студентів, ігноруючи всі зміни, зроблені ним в полях даних студента.

Сторінка створення анкети містить поле для вводу її назви, кнопки додавання питання, збереження анкети, скасування її створення. Після натискання на кнопку додавання питання на сторінці з’являється елемент з полем для введення тексту питання.. Макет описаної сторінки представлений нижче.

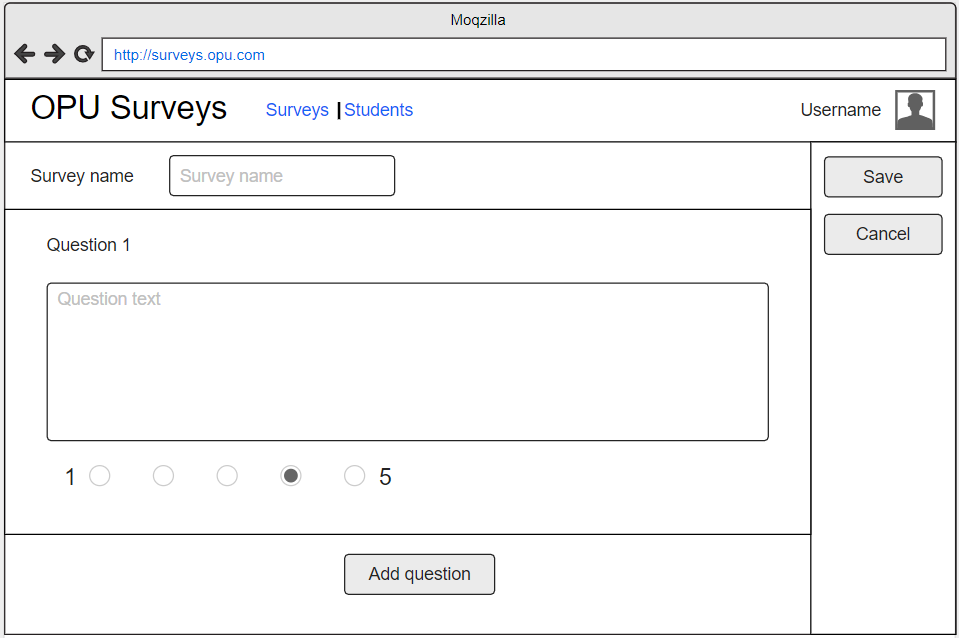


Рисунок 4.13 – Макет сторінки викладача для створення анкети

Сторінка редагування анкети буде містити ті самі елементи, але на ній будуть наявні всі питання, що були попередньо додані до анкети. Так само, як і на сторінці редагування інформації про студента, викладач має змогу редагувати всю інформацію в анкеті, змінювати питання, додавати нові і видаляти вже створені.

Після вибору закритої анкети на екрані перегляду списку анкет викладача і натискання кнопки «Viewresults» система виводить екран з результатами анкетування. На цьому екрані виведені всі питання анкети зі статистичними даними, що були розраховані за відповідями студентів. в разі якщо системою була виявлена проблема, передбачена в розрахунках, в елементі відповідного питання буде виведене відповідне повідомлення. Макет сторінки з результатами анкетування представлений на рисунку 4.14.

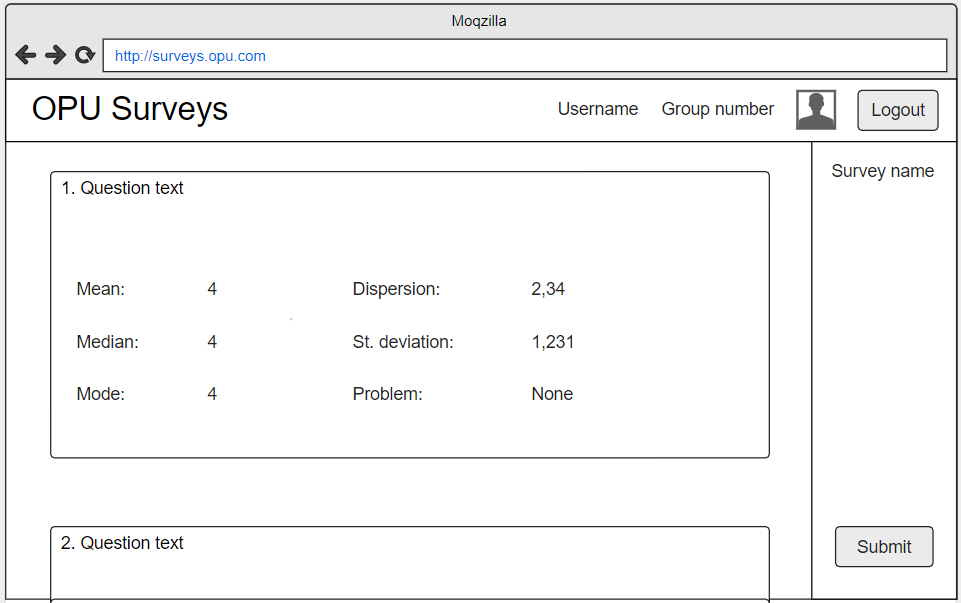


Рисунок 4.14 – Макет сторінки викладача для перегляду результатів анкетування

**Висновок**

В даному розділі було виконане проектування програмного продукту, що розроблюється, а саме: описана загальна архітектура програмної системи, розроблені і описані схема функціонування веб застосування і діаграма класів серверної частини. Також буда розроблена структура реляційної бази даних для програмної системи. Було виконано проектування інтерфейсу користувача за допомогою вайрфреймів, результати проектування наведені в розділі.

5Програмнареалізація системи і тестування

5.1 Набір інструментальних засобів розробки

Система передбачає розробку веб-застосування і серверної частини. Для розробки веб-застосування була обрана мова Javascript і її бібліотеки - React і Redux, а для серверної частини - середовище Node.js іфреймворк Express.js.

«JavaScript - динамічна, об'єктно-орієнтована прототипна мова програмування реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки» [12]. JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

«React - JavaScript-бібліотека для роботи з користувача інтерфейсами (UI), яку створили розробники Facebook. Бібліотеку почали використовувати на сайті цієї соціальної мережі в 2011 році, а в 2013 році Facebook відкрив вихідний код React. За допомогою React розробники створюють веб-застосування, які змінюють відображення без перезавантаження сторінки» [13]. Завдяки цьому застосування швидко реагують на дії користувача, наприклад, заповнення форм, застосування фільтрів, додавання товарів в корзину і так далі. React застосовують для відтворення компонентів для користувача інтерфейсу. Також бібліотека може повністю управляти клієнтською частиною застосування. В цьому випадку React використовують з бібліотеками для управління станом і роутингу, наприклад, Redux і ReactRouter. Одна з ключових особливостей React - універсальність. Цю бібліотеку можна використовувати на сервері і на мобільних платформах за допомогою ReactNative. Це принцип LearnOnce, WriteAnywhere або «Навчіться один раз, пишіть де завгодно». Ще одна важлива особливість бібліотеки - декларативність. За допомогою React розробник описує, як компоненти інтерфейсу виглядають в різних станах. Декларативний підхід скорочує код і робить його зрозумілим. React заснований на компонентах, це ще одна ключова особливість бібліотеки. Кожен компонент повертає частину призначеного для користувача інтерфейсу зі своїм станом. Об'єднуючи компоненти, програміст створює завершений інтерфейс веб-додатки. Важлива особливість React - використання JSX. Це розширення синтаксису JavaScript, яке зручно використовувати для опису інтерфейсу. JSX дозволяє писати JavaScript-код за допомогою готових компонентів, які практично повністю повторюють HTML. Це спрощує розробку. До важливих особливостей React відноситься використання віртуального DOM (Virtual DOM). Віртуальний DOM - об'єкт, в якому зберігається інформація про стан інтерфейсу. При зміні стану, наприклад, після відправки форми або натискання кнопки, React розраховує різницю між старим і новим станом. Після цього бібліотека рендерить нове стан. Використання віртуального DOM дозволяє бібліотеці ефективно оновлювати реальний DOM.

«Redux - це шаблон і бібліотека для управління та оновлення стану програми, використовуючи події, що називаються "дії". Він служить централізованим сховищем для стану, який потрібно використовувати в усьому вашому додатку, з правилами, що забезпечують оновлення стану лише передбачуваним чином.Redux допомагає вам керувати "глобальним" станом - станом, необхідним для багатьох частин вашої програми» [14].Шаблони та інструменти, надані Redux, полегшують розуміння того, коли, де, чому та як оновлюється стан у вашій програмі та як поводиться логіка застосування, коли відбудуться ці зміни. Redux спонукає до написання коду, який можна передбачити та перевірити, що допомагає надати розробнику впевненості в тому, що розроблена ним програма працюватиме належним чином.

Застосування, написане на Redux, складається з наступних частин: state, view і **actions. State – єдиний глобальний об’єкт, що відображає стан застосування. View–інтерфейс користувача, що рендериться в залежності від поточного state. Actions – події, що відбуваються в застосуванні в залежності від введення користувача чи триггерних оновленнях в стані.**

«Node.js - платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript. Засновником платформи є РаянДал (RyanDahl). Якщо раніше Javascript застосовувався для обробки даних в браузері користувача, то node.js надав можливість виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їхнього виконання. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників»[15]. Node.js має наступні властивості:

1) асинхронна однопоточна модель виконання запитів;

2) неблокуюче введення/виведення;

3) система модулів CommonJS;

4) рушій JavaScriptGoogle V8.

Для керування модулями використовується пакетний менеджер npm (nodepackagemanager).

«Express.js - це мінімалістичний і гнучкий веб-фреймворк для додатків Node.js, що надає великий набір функцій для мобільних і веб-застосувань. Він є програмним каркасом розробки серверної частини веб-застосунків для Node.js, реалізований як вільне і відкрите програмне забезпечення під ліцензією MIT. Він спроектований для створення веб-застосунків і API. Маючи в своєму розпорядженні безліч службових методів HTTP і проміжних оброблювачів, можна швидко і легко створити надійний REST-API для застосування» [16].

5.2 Розробка функціональних тестів

Функціональне тестування – метод тестування, який передбачає аналіз функціональних характеристик застосування та перевірку на невідповідності між реальною поведінкою реалізованих функцій і очікуваною поведінкою відповідно до специфікації і бізнес-вимог. Функціональне тестування імітує фактичне використання системи.

Функціональні тести можуть проводитися на усіх рівнях тестування (компонентному, інтеграційному, системному, приймальному). Як правило, ці функції описуються в вимогах, функціональних специфікаціях або у вигляді варіантів використання системи (Usecase).

Після проходження кожного тесту необхідно виставити його результат, щоб після проходження всіх тестів виявити функціонал, що не працює або працює некоректно.

Функціональні тести для цих сценаріїв варіантів використання представлено в наступних таблицях.

1. Авторизуватися в системі.

Таблиця 5.1 – функціональний тест для варіанту використання «Авторизуватися в системі»

|  |  |  |  |
| --- | --- | --- | --- |
| Умова | Користувач ввів коректні дані при авторизації | T | F |
| Дія | Надати доступ в систему | + |  |
|  | Вивести повідомлення про помилку: введені некоректні дані |  | + |
|  |  | Passed | Passed |

2. Вийти з системи.

Таблиця 5.2 – функціональний тест для варіанту використання «Вийти з системи»

|  |  |  |
| --- | --- | --- |
| Умова | Користувач натискає кнопку «Вийти з системи» | T |
| Дія | Перевести користувача на екран авторизації | + |
|  |  | Passed |

3. Пройти анкету.

Таблиця 5.3 – функціональний тест для варіанту використання «Пройти анкету»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Умова | Студент обирає непройдену анкету зі списку доступних йому анкет | T | F | T |
|  | Збереження даних успішне | T | T | F |
| Дія | Перевести користувача на екран проходження анкети | + |  |  |
|  | Зберегти дані | + |  |  |
|  | Вивести повідомлення про помилку: кожну анкету можна пройти не більшеодного разу |  | + |  |
|  | Вивести повідомлення про помилку: збій збереження |  |  | + |
|  |  | Passed | Passed | Passed |

4. Створити анкету.

Таблиця 5.4 – функціональний тест для варіанту використання «Створити анкету»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Умова | Викладач ввів коректні дані | T | F | T |
|  | Збереження даних успішне | T | T | F |
| Дія | Зберегти дані | + |  |  |
|  | Вивести повідомлення про помилку: допустима довжина питання перевищена |  | + |  |
|  | Вивести повідомлення про помилку: збій збереження |  |  | + |
|  |  | Passed | Passed | Passed |

5. Редагувати анкету.

Таблиця 5.5 – функціональний тест для варіанту використання «Редагувати анкету»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Умова | Викладач вибрав анкету зі станом “New” | T | F | T | T |
|  | Викладач ввів коректні дані | T | T | F | T |
|  | Збереження даних успішне | T | T | T | F |
| Дія | Зберегти дані | + |  |  |  |
|  | Вивести повідомлення про помилку: обрана анкета наразі відкрита для проходження, редагування заборонено |  | + |  |  |
|  | Вивести повідомлення про помилку: допустима довжина питання перевищена |  |  | + |  |
|  | Вивести повідомлення про помилку: збій збереження |  |  |  | + |
|  |  | Passed | Passed | Passed | Passed |

6. Створити копію існуючої анкети.

Таблиця 5.6 – функціональний тест для варіанту використання «Створити копію існуючої анкети»

|  |  |  |  |
| --- | --- | --- | --- |
| Умова | Викладач натискає кнопку «Створити копію анкети» | T | T |
|  | Збереження даних успішне | T | F |
| Дія | Створити копію анкети | + |  |
|  | Вивести повідомлення про помилку: збій збереження |  | + |
|  |  | Passed | Passed |

7. Видалити анкету.

Таблиця 5.7 – функціональний тест для варіанту використання «Створити копію існуючої анкети»

|  |  |  |  |
| --- | --- | --- | --- |
| Умова | Викладач натискає кнопку «Видалити анкету» | T | T |
|  | Видалення даних успішне | T | F |
| Дія | Створити копію анкети | + |  |
|  | Вивести повідомлення про помилку: збій видалення |  | + |
|  |  | Passed | Passed |

8. Відкрити доступ до анкети.

Таблиця 5.8 – функціональний тест для варіанту використання «Відкрити доступ до анкети»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Умова | Викладач вибрав анкету зі станом “New” | T | F | T | T |
|  | Викладач ввів коректні дані | T | T | F | T |
|  | Збереження даних успішне | T | T | T | F |
| Дія | Зберегти дані | + |  |  |  |
|  | Вивести повідомлення про помилку: обрана анкета наразі відкрита для проходження або вже закрита, відкриття неможливе |  | + |  |  |
|  | Вивести повідомлення про помилку: список груп не може бути порожнім |  |  | + |  |
|  | Вивести повідомлення про помилку: збій збереження |  |  |  | + |
|  |  | Passed | Passed | Passed | Passed |

9. Видалити інформацію про студента.

Таблиця 5.9 – функціональний тест для варіанту використання «Редагувати інформацію про студента»

|  |  |  |  |
| --- | --- | --- | --- |
| Умова | Збереження даних успішне | T | F |
| Дія | Зберегти дані | + |  |
|  | Вивести повідомлення про помилку: збій збереження |  | + |
|  |  | Passed | Passed |

10. Закрити доступ до анкети.

Таблиця 5.10 – функціональний тест для варіанту використання «Закрити доступ до анкети»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Умова | Викладач вибрав анкету зі станом “Open” | T | F | T |
|  | Збереження даних успішне | T | T | F |
| Дія | Зберегти дані | + |  |  |
|  | Вивести повідомлення про помилку: обрана анкета наразі не відкрита для проходження або вже є закритою, закриття неможливе |  | + |  |
|  | Вивести повідомлення про помилку: збій збереження |  |  | + |
|  |  | Passed | Passed | Passed |

11. Отримати результати анкетування.

Таблиця 5.11 – функціональний тест для варіанту використання «Отримати результати анкетування»

|  |  |  |  |
| --- | --- | --- | --- |
| Умова | Викладач вибрав анкету зі станом “New” | T | F |
| Дія | Вивести дані | + |  |
|  | Вивести повідомлення про помилку: обрана анкета ще не відкрита або ще не закрита, отримання результатів анкетування неможливе |  | + |
|  |  | Passed | Passed |

12. Зареєструвати студента.

Таблиця 5.12 – функціональний тест для варіанту використання «Відкрити доступ до анкети»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Умова | Викладач ввів коректні дані | T | F | T |
|  | Збереження даних успішне | T | T | F |
| Дія | Зберегти дані | + |  |  |
|  | Вивести повідомлення про помилку: введені некоректні дані |  | + |  |
|  | Вивести повідомлення про помилку: збій збереження |  |  | + |
|  |  | Passed | Passed | Passed |

13. Редагувати інформацію про студента.

Таблиця 5.13 – функціональний тест для варіанту використання «Редагувати інформацію про студента»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Умова | Викладач ввів коректні дані | T | F | T |
|  | Збереження даних успішне | T | T | F |
| Дія | Зберегти дані | + |  |  |
|  | Вивести повідомлення про помилку: введені некоректні дані |  | + |  |
|  | Вивести повідомлення про помилку: збій збереження |  |  | + |
|  |  | Passed | Passed | Passed |

**Висновок**

В даному розділі були обґрунтовані переваги обраних для розробки технологій і засобів реалізації програмного продукту. З описаного в даному розділі можна зробити висновок, що обрані технології дозволять в повній мірі реалізувати розроблений функціонал.

Був розроблений набір функціональних тестів для розробленої програмної системидля виявлення помилок і дефектів, що могли виникнути в процесі реалізації. За результатами тестування дефектів у програмному продукті не виявлено, система повністю відповідає розробленим вимогам.

6 приклад використання програмної системи і експеримент

6.1 Приклад використання

Для початку роботи необхідно відкрити веб застосування в браузері. У ньому відобразиться сторінка авторизації користувача. Вона єдина для всіх типів користувачів. В застосуванні бачимо форму логіна, кнопку “Login” для відправки даних форми. Сторінка авторизації користувача представлена на рисунку 6.1.

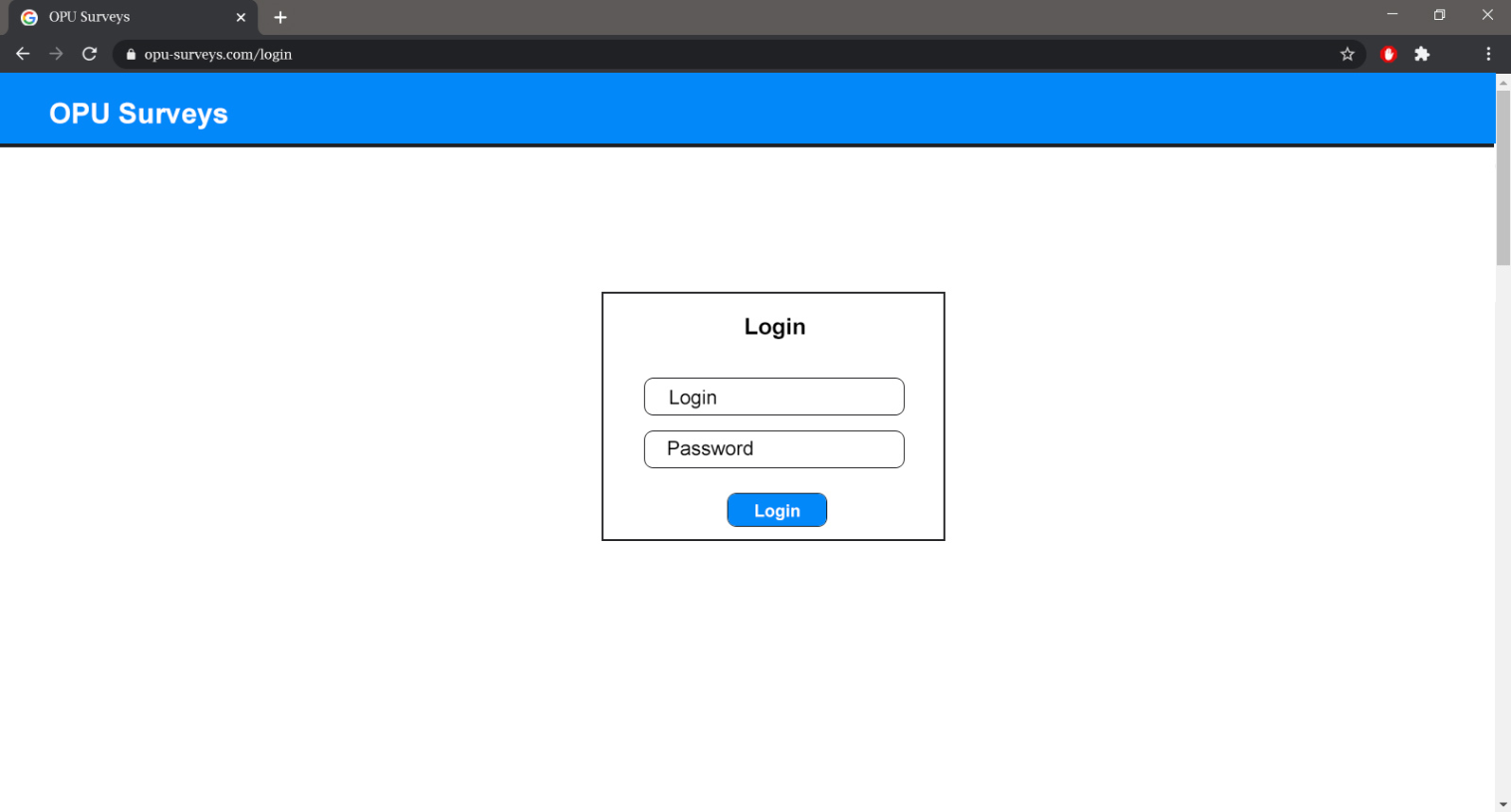


Рисунок 6.1 – Сторінка авторизації

Якщо користувач авторизувався, як студент, то йому виведеться сторінка зі списком доступних йому анкет. Агрегування анкет на сторінці користувача є значною перевагою системи, адже таким чином користувач не має переходити за безліччю посилань, щоб пройти декілька анкет. Скріншот даної сторінки представлений на рисунку 6.2.

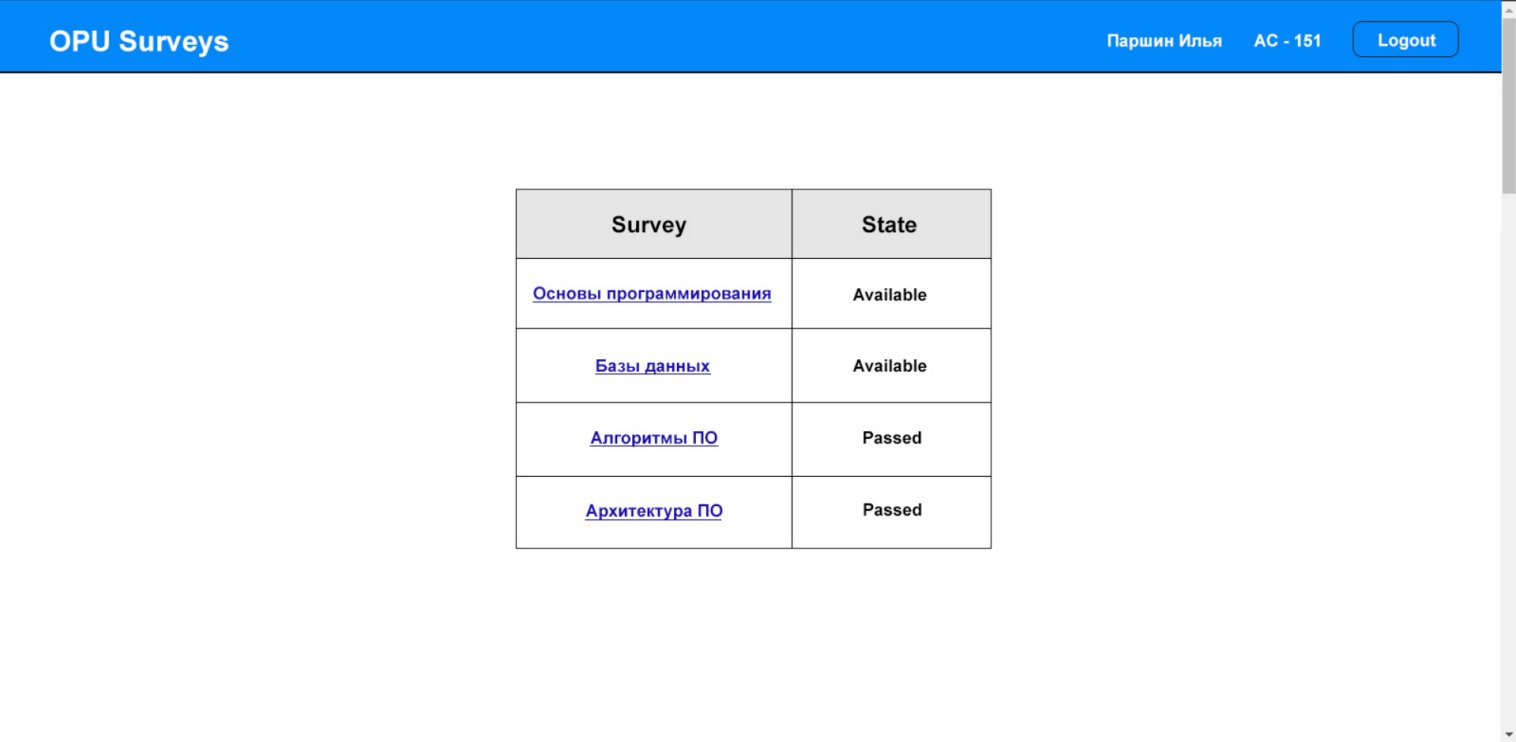


Рисунок 6.2 – Список анкет, доступних студенту

Після того, як студент обирає зі списку анкету, що ще не буда пройдена ним, система переводить його на сторінку для проходження анкети. На ній він може заповнити відповіді на питання, натиснувши на кнопку Submit, він відправить їх на сервер, дана анкета стане недоступною для проходження. Сторінка проходження анкети зображена на рисунку 6.3.

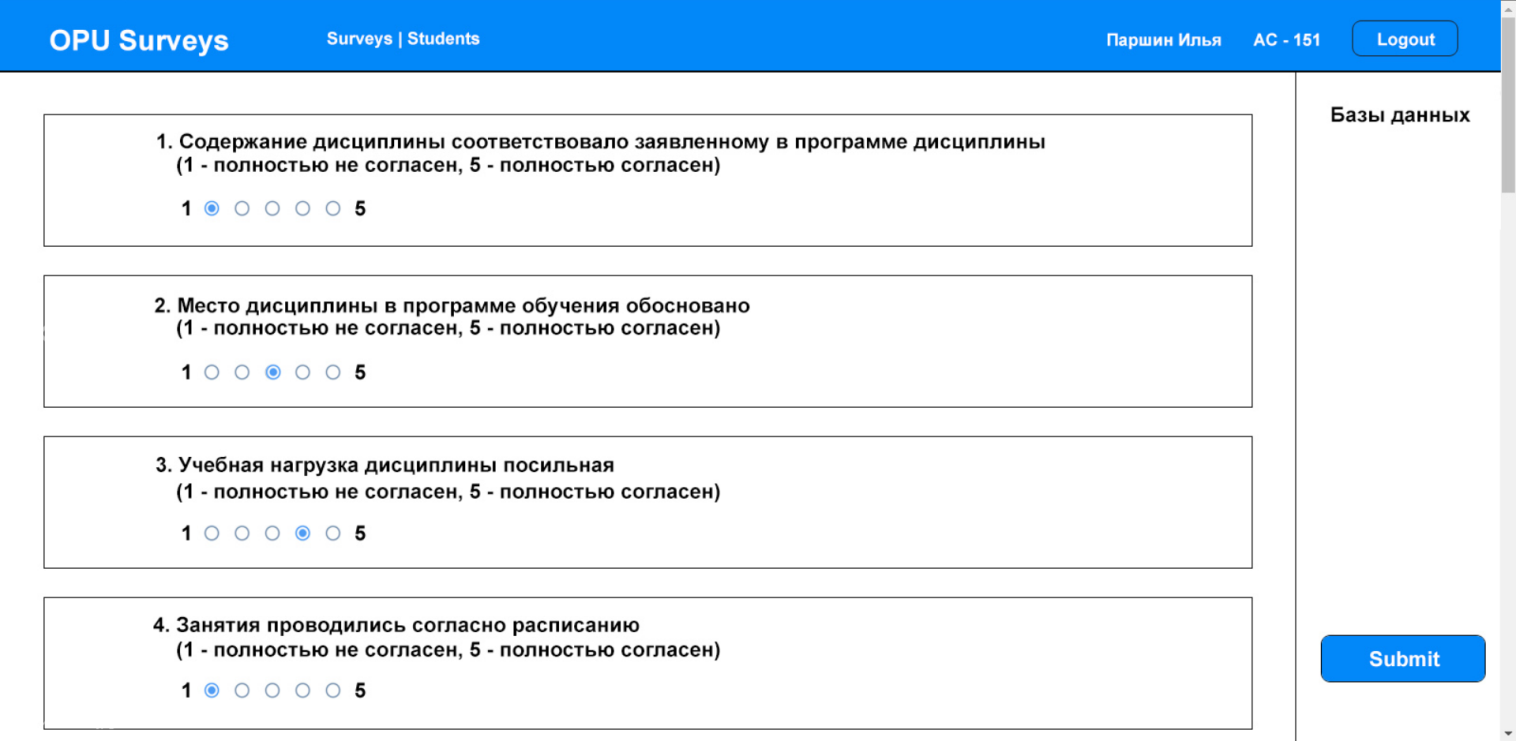


Рисунок 6.3 – Сторінка для проходження анкети

Якщо на сторінці авторизації користувач ввів автентифікаційні дані викладача, то система переводить його на сторінку з інформацією про всі наявні анкети. Викладачу доступні всі передбачені системою функції для управління даними про анкети, а саме: створення, редагування, копіювання, видалення, відкриття і закриття анкети для проходження, а також перегляд результатів закритих для проходження анкет. Скріншот сторінки викладача з інформацією про анкети представлений на рисунку 6.4.

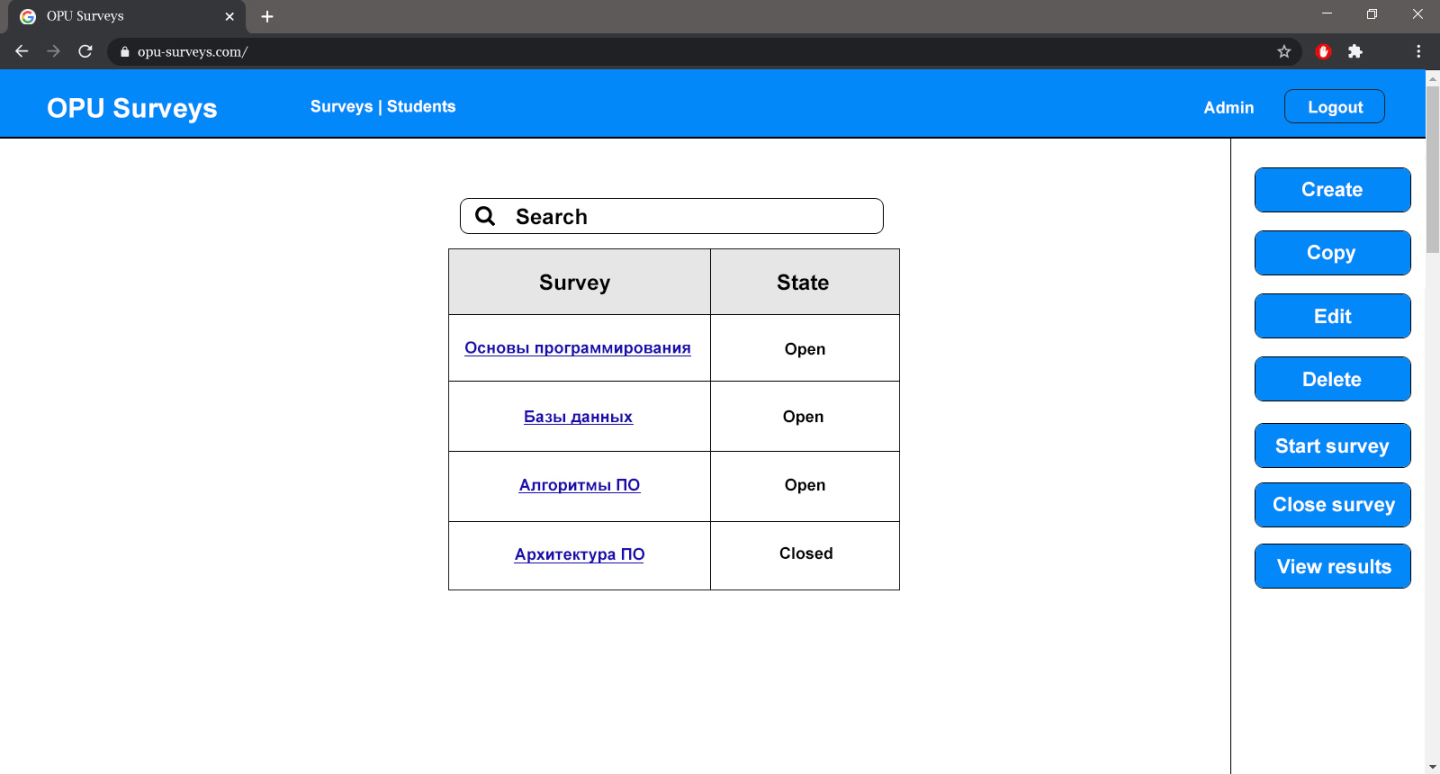


Рисунок 6.4 – Сторінка викладача з інформацією про анкети

Якщо Викладач натискає на посилання «Students» зверху екрана, система виводить викладачу екран зі студентами. Викладачу доступні функції управління інформацією про студента, такі як створити запис про студента, редагувати запис про студента, видалити запис про студента і експортувати дані про студента у файл csv. Скріншот сторінки викладача з інформацією про студентів представлений на рисунку 6.5.

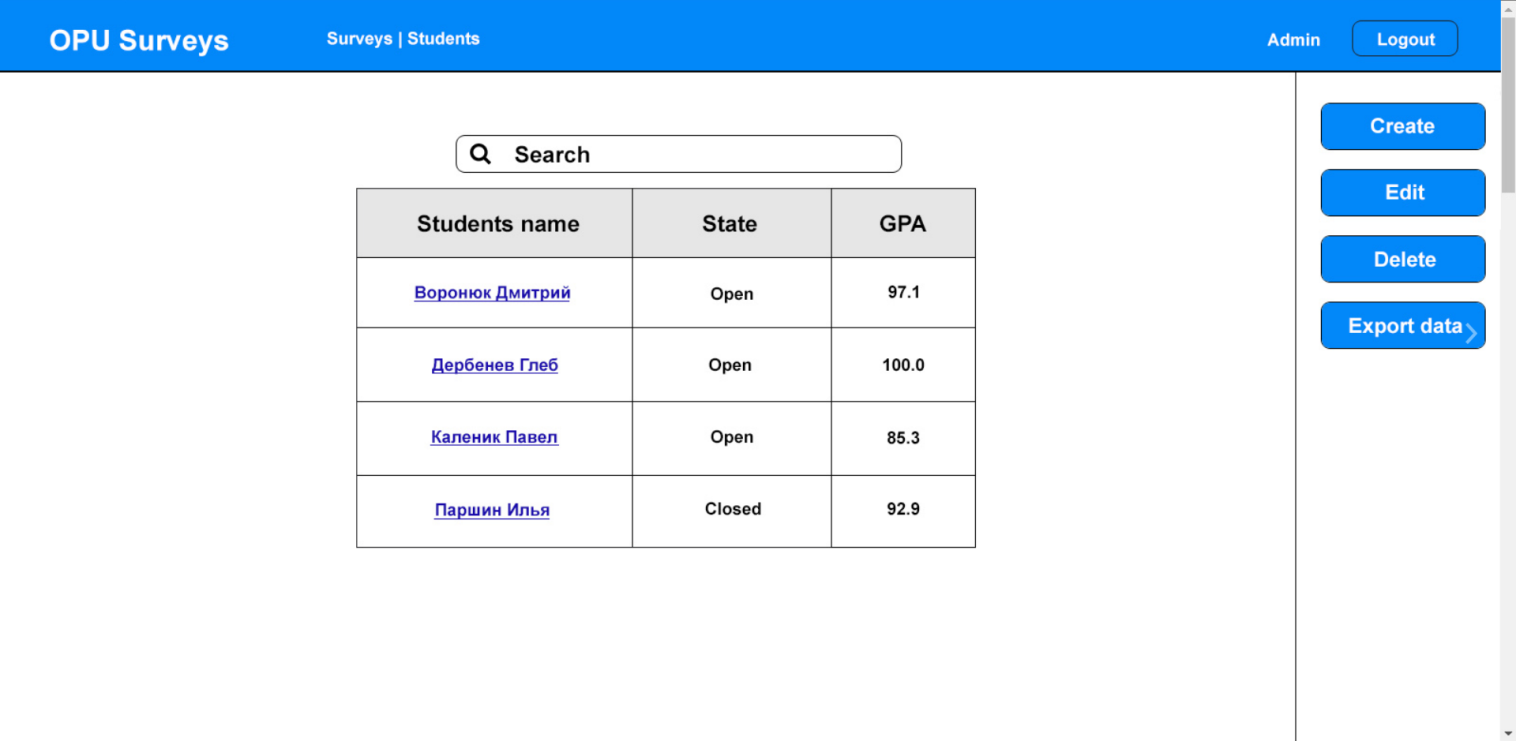


Рисунок 6.5 – Сторінка викладача з інформацією про студентів

Якщо викладач натискає кнопку «Create», система виводить йому сторінку Сторінка створення студента містить форму для заповнення всіх даних студента, що є важливими для функціонування системи, а саме: прізвище, ім’я, по батькові, ідентифікатор групи і середній бал студента. Також на сторінці наявні кнопка збереження даних, що створить запис про студента в системі, а також кнопка скасування, яка поверне викладача на сторінку перегляду списку студентів. Скріншот сторінки для створення запису про студента представлений на рисунку 6.6.

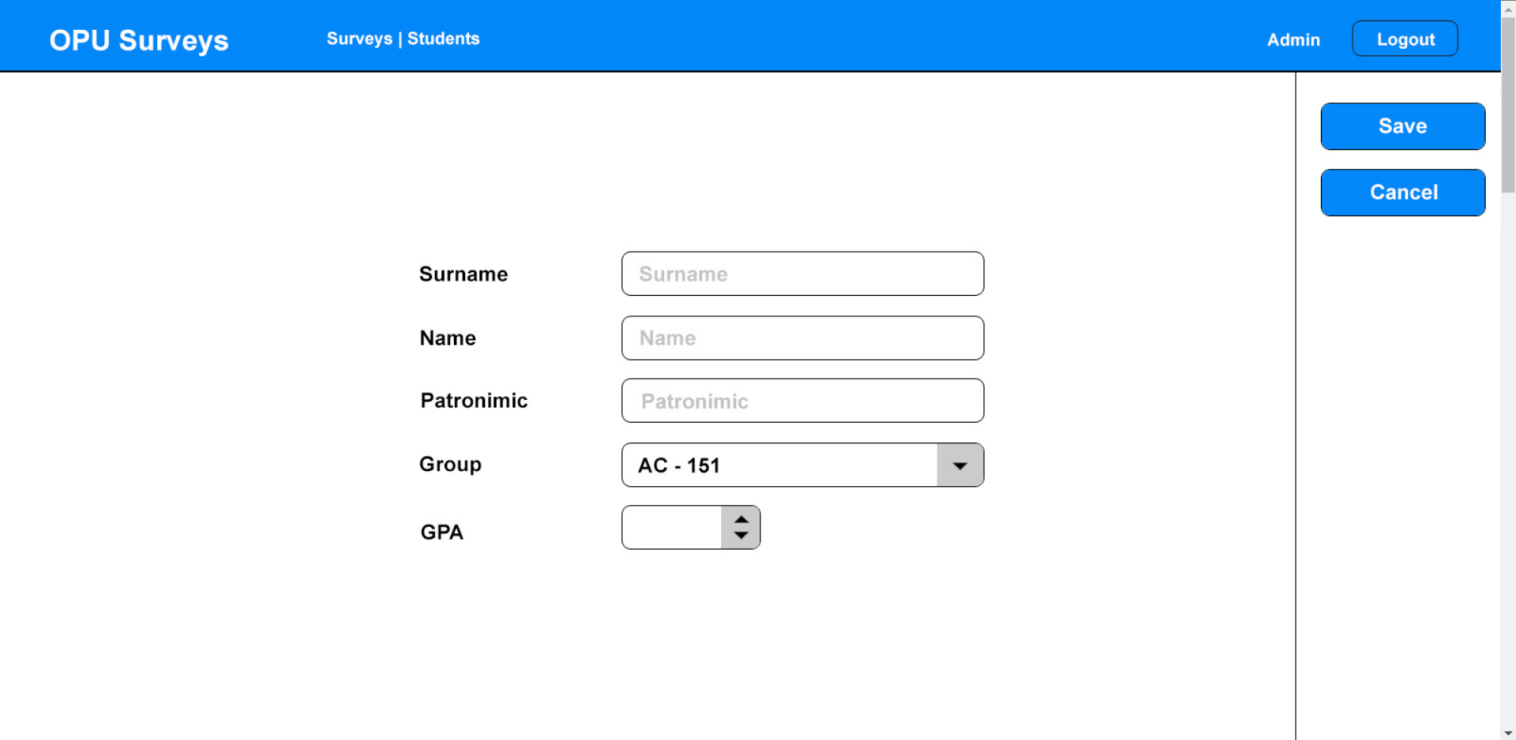


Рисунок 6.6 - Сторінка для створення запису про студента

Якщо викладач обирає студента і натискає кнопку “Edit”, система виводить користувачу сторінку, аналогічну за складом сторінці для створення запису про користувача, але в цьому разі поля вже заповнені даними і доступні для редагування.

Якщо на сторінці зі списком анкет викладач натискає кнопку «Create», система виводить користувачу сторінку створення анкети. Вона містить поле для вводу її назви, кнопки додавання питання, видалення питання збереження анкети, скасування її створення. Після натискання на кнопку додавання питання на сторінці з’являється елемент з полем для введення тексту питання. Скріншот описаної сторінки представлений на рисунку 6.7.

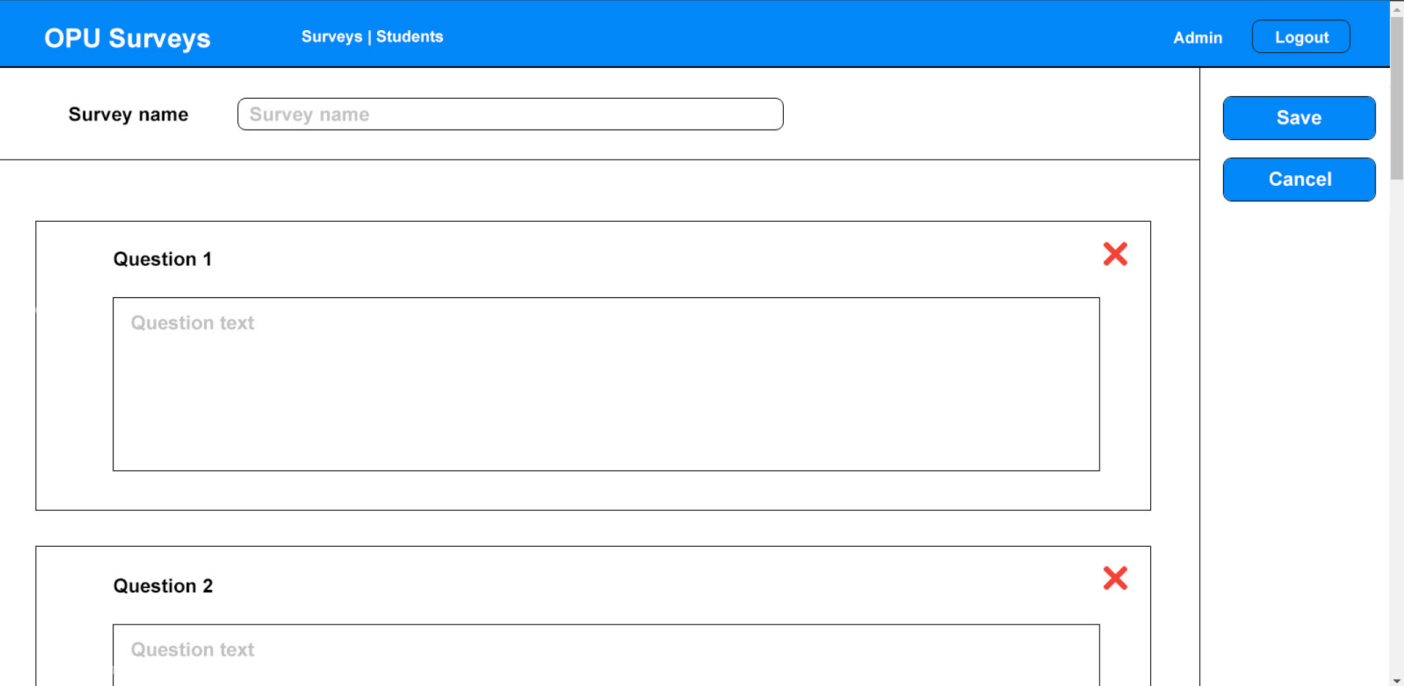


Рисунок 6.7 - Сторінка для створення анкети

6.2 Тестування зміни рівня якості навчального процесуз використанням розробленої системи

Метою даної роботи є підвищення якості навчального процесу шляхом розробки ефективного і зручного інструменту для збору і аналізу відгуків студентів щодо навчального процесу. Саме тому необхідно провести тестування системи в реальних умовах, аби з’ясувати, наскільки корисною єдана система для підвищення якості освіти.

Для початку необхідно визначити спосіб оцінювання. Як зазначалося в другому розділі роботи, навчальний процес складається з трьох складових частин, а саме: зміст курсу, методи донесення і матеріали курсу.

Для оцінки корисності доцільно використати умовний показник задоволеності кожною з визначених частин навчального процесу.Щоб розрахувати даний показник, необхідно отримати вихідні дані, що є результатами анкетування, проведеного з допомогою розробленої системи. Анкета для проведення експерименту розроблена таким чином, що кожне питання в ній стосується того чи іншого аспекту складових частин навчального процесу. В розробленій системі передбачено, що для кожного питання розраховується середнє арифметичне всіх відповідей на це питання, зібраних в процесі анкетування. Для розрахунку показника задоволеності для кожної складової навчального процесу використаємо формулу:

, (6.1)

де *M* – показник задоволеності однією зі складових навчального процесу,

–сумасередніх арифметичних відповідей на питання, що стосуються однієї зі складових частин навчального процесу,

*n* – загальна кількість питань, що стосуються однієї зі складових частин навчального процесу.

Таким чином отримуємо значення *MCC*, *MDM*, *MCM*, що є показниками задоволеності змістом курсу, методами донесення і матеріалами курсу відповідно.

Для того, щоб виявити, наскільки корисною є дана система, необхідно розрахувати приріст кожного з цих показників після проведення заходів з підвищення якості навчального процесу за формулою:

, (6.2)

де *L* – приріст показника у відсотках,

–значення показника за результатами первинного анкетування,

– значення показника за результатами анкетування після проведення заходів з підвищення якості навчального процесу.

Для проведення експерименту була обрана паралель десятих класів ЗОШ №86 міста Одеса, курс всесвітньої історії. Для цього курсу була створена анкета, що відповідає вимогам, описаним вище.Результати проведення первинного анкетування наведені в таблиці 6.1.

Таблиця 6.1 – Результати первинного анкетування

|  |  |
| --- | --- |
| Показник | Значення |
| *MCC* | 3,8 |
| *MDM* | 2,67 |
| *MCM* | 3,3 |

Результати анкетування було проаналізовано і в результаті були прийняті такі рішення:

1) Зміст курсу визначається МОН України, тому вплинути на зміст курсу в короткій перспективі не є можливим.

2) Методи донесення отримали низьку оцінку, що в розробленій системі інтерпретується, як проблема, тому дана частина навчального процесу потребує змін. Зроблено припущення, що застосування більшої кількості наочних матеріалів, зокрема фільмів і презентацій, зможе підвищити зацікавленість і залученість учнів у навчальний процес.

3) Матеріали курсу також отримали доволі низьку оцінку. Хоча дана оцінка і не є критичною, було прийняте рішення внести корективи в матеріали курсу. Так, наприклад,дуже корисним є роздавання списків найважливіших дат і термінів за кожним параграфом підручника, що полегшує їх пошук і запам’ятовування.

Описані заходи з підвищення якості проводилися протягом двох тижнів, після чого було проведене повторне анкетування за тим самим набором питань для порівняння результатів. Результати повторного анкетування представлені в таблиці 6.2.

Таблиця 6.2 – Результати первинного анкетування

|  |  |
| --- | --- |
| Показник | Значення |
| *MCC* | 4,3 |
| *MDM* | 3,93 |
| *MCM* | 3,7 |

Як бачимо, показники задоволеності навчальним процесом збільшилися, при чому застосовані заходи опосередковано вплинули на сприйняття змісту курсу, щодо якого не було застосовано ніяких специфічних заходів.

Тепер маємо всі дані для розрахунку відсоткового приросту для кожного показника. Також для зручності розрахуємо середній приріст серед всіх трьох показників. Результати розрахунку представлені в таблиці 6.3.

Таблиця 6.3 – Результати розрахунку

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Показник | Значення | Показник | Значення | Показник | Значення |
| *MCC1* | 3,8 | *MCC2* | 4,3 | *LCC* | 13,15% |
| *MDM1* | 2,67 | *MDM2* | 3,93 | *LDM* | 47,19% |
| *MCM1* | 3,3 | *MCM2* | 3,7 | *LCM* | 12,12% |
|  |  |  |  | *Lсер* | 24,15% |

Для більш наочного представлення отриманих результатів експерименту наведена лінійчатадіаграма. Вона представлена на рис. 6.8.

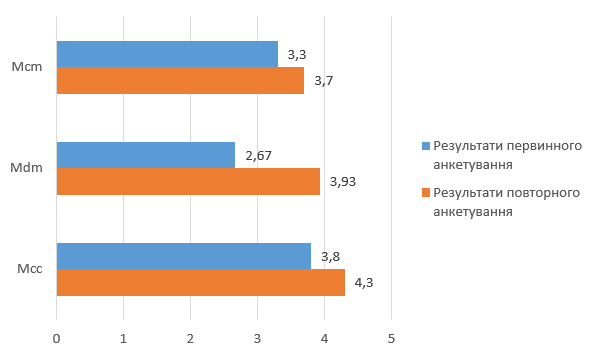


Рисунок 6.8 – Діаграма результатів експерименту

Таким чином в результаті експерименту було виявлено, що з використанням даної системи показники задоволеності навчальним процесом в середньому виросли на 24,15%. Використання розробленої системи допомогло швидко зібрати зворотній зв’язок учнів щодо навчального процесу і проаналізувати його. Це дало змогу прийняти вірні рішення щодо заходів для покращення якості навчального процесу. Вірність цих рішень підтвердилася в результаті повторного анкетування.

**Висновок**

Був наведений приклад використання системи зі скріншотами всіх найважливіших сторінок веб застосування.

Був проведений експеримент, що довів застосовність і корисність розробленої системи в сфері покращення якості навчального процесу.

7 охорона праці

«Поняття «охорона праці» визначено статтею 1 Закону України «Про охорону праці». Охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.»[17].

Головною метою охорони праці є створення на кожному робочому місці безпечних умов праці, безпечної експлуатації обладнання, зменшення або повна нейтралізація дії шкідливих і небезпечних виробничих факторів на організм людини і, як наслідок, зниження виробничого травматизму та професійних захворювань.

У дипломному проекті розглянуті питання підвищення якості навчального процесу у вищих навчальних закладах, вирішення яких здійснюєтьсяІТ відділом продуктової компанії Mindojo, а саме їх виконання здійснюється розробником програмного забезпечення.

Отже розглянемо питання охорони праці в надзвичайних ситуаціях на робочому місці розробника ПЗ, яке розташоване в двоповерховій будівлі на першому поверсі, на території навчального закладу.

Питання охорони праці в надзвичайних ситуаціях на робочому місці розробника ПЗ детально розглянуто в Додатку Б. В ході аналізу цих питань були розкриті наступні аспекти охорони праці:

1) Визначення основних параметрів робочого місця користувача ЕВМ.

2)Дослідження ефективності занулення.

3) Прогнозування наслідків аварії з викидом хімічно-небезпечних речовин.

4)Заходи з охорони праці для безпеки у надзвичайних ситуаціях.

5) Безпека у суспільстві в умовах COVID-19.

**Висновок**

В даному розділі розглянуто організацію, управління охороною праці на підприємстві, навчання охороні праці, проведення інструктажів та розслідування несчасних випадків на підприємстві «Mindojo».

Проаналізовано умови праці на робочому місці розробника програмного забезпечення. Розраховано занулення, перевірена правильність вибору елементів занулення заданої за варіантом установки. Оскільки індивідуальний розрахунок був спрямований на занулення, запропоновані заходи з охорони праці спрямовані на питання електробезпеки.

Виконано прогнозування наслідків аварії з викидом хімічно небезпечних речовин, а саме аміаку.

Описано базові правила безпеки в суспільстві в умовах пандемії COVID-19.

Висновки

В даній роботі була розглянута проблема підвищення якості навчального процесу у закладах вищої освіти. Рішенням цієї проблеми стала програмна система для збору і статистичного аналізу зворотнього зв’язку студентів. Розроблена система надає викладачам зручний інструмент для створення анкет, власне проведення анкетування і перегляду отриманих результатів. Студенти отримують доступ до анкет, що агреговані в одному місці, що значно спрощує їх заповнення.

Був проведений експеримент з використанням розробленої програмної системи, в результаті якого був отриманий середній приріст 24,15% до показників задоволеності начальним процесом.

У процесі розробки даної програмної системи були виконані наступні етапи: аналіз вимог до програмного продукту, аналіз і вибір методів оцінки навчального процесу, специфікація вимог до програмного продукту, проектування його архітектури і інтерфейсу користувача, вибір програмних засобів для реалізації, власне, розробка і тестування розробленої системи.

Обрана предметна область була проаналізована з теоретичної точки зору, а саме: проведений аналіз *SET*, виявлені ключові її аспекти, виявлені характеристики, що впливають на якість навчального процесу, обрана шкала оцінювання для відповідей на питання анкет, а також обрані розрахунки, необхідні для аналізу результатів анкетування.

Для розробки системи була обрана архітектура «клієнт-сервер», у якій клієнтом виступає веб застосування. Сервером виступає REST-API.

Всі вимоги були виконані в повному обсязі. Розроблена система має перспективу розвитку за рахунок розширення і покращення існуючого функціоналу. Наприклад, додавання нових функцій для управління акаунтами і автентифікаційними даними. Доцільним буде покращення адаптивності веб застосування для використання на мобільних пристроях, а обрана архітектура дозволить розробити застосування під мобільні операційні системи.

Список використаних джерел

1. Komleva N.O., Parshin I.A. Program for collection and analysis of students' responses on the quality of the educational process using statistical methods of data processing // XIII Міжнародна науково-практична конференція «Інформаційні технології і автоматизація – 2020», Одеса, 22-23 жовтня 2020. С. 193-195.
2. Nataliia Komleva, Vira Liubchenko and Svitlana Zinovatna. Improvement of Teaching Quality in the View of a Resource-Based Approach. 16th International Conferenceon ICT inEducation, Research, and Industrial Applications. October 06-10, 2020.
3. Iowa State University. Student Evaluation of Teaching (SET) Guidelines [Електронний ресурс] – Режим доступу: https://www.celt.iastate.edu/teaching/assessment-and-evaluation/student-evaluation-of-teaching-set-guidelines-and-recommendations-for-effective-practice.
4. National Research Council 2003. Evaluating and Improving Undergraduate Teaching in Science, Technology, Engineering, and Mathematics. Washington, DC: The National Academies Press. https://doi.org/10.17226/10024.
5. Міри розкиду даних. [Електронний ресурс] – Режим доступу: http://cito-web.yspu.org/link1/metod/met125/node26.html
6. Козак О.Л. Опорний конспект лекцій з курсу «Аналіз вимог до программного забезпечення» для студентів напрямку підготовки «Програмна інженерія» / О.Л. Козак. – Тернопіль, 2011. – 56 с.
7. Конспект лекцій з дисципліни «Аналіз вимог до программного забезпечення» для студентів інституту комп’ютерних систем з базової підготовки за напрямком «Програмна інженерія»/Авт. О.Б. Кунгурцев. - Одеса. 2016. – 101 с.
8. Бабич А. В. Введение в UML. ISBN 978-5-94774-878-9, 6. Лекция: Диаграммы прецедентов: крупным планом [Електронный ресурс] – Режим доступа: www.intuit.ru/studies/courses/1007/229/lecture/5962.
9. Розробка інформаційних ресурсів і систем: Електронне навчальне видання. Конспект лекцій / Л. С. Глоба, Т. М. Кот. – К.: НН ІТС НТУУ «КПІ», 2012. – 322 с.
10. Методичні вказівки до практичних занять з дисципліни «Архітектура та проектування программного забезпечення» для студентів напряму 6.050103 Програмна інженерія / Укл. В. В. Любченко. O Одеса: ОНПУ, 2018. - 17 с.
11. Конспект лекцій з дисципліни «Людино-машинна взаємодія» для студентів напряму підготовки 6.050403 «Програмна інженерія» / Оніщенко Т.В., Городнича Е.А. Одеса: ОНПУ, 2017. – 95 с.
12. Поллок, Дж. JavaScript. Руководство разработчика / Дж. Поллок. - М.: Питер, 2011. - 237 c.
13. Документація React [Електронний ресурс] – Режим доступу: URL: https://uk.reactjs.org/.
14. Redux Documentation [Електронний ресурс] - Режим доступу: URL: https://redux.js.org/.
15. Shelley Powers “Learning Node: Moving to the Server-Side. 2nd edition", St. Petersburg, 2017. – 288 p.
16. Express.js Documentation [Електронний ресурс] - Режим доступу: URL: https://expressjs.com/uk/.
17. Винокурова Л. Е., Васильчук М. В., Гаман М. В. В49 Основи охорони праці: Підручн. для проф.-техн. навч. закладів. — 2-ге вид., допов., перероб. — К.: Вікторія, 2001. - 192 с.