МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської та_____

аспірантської підготовки

Кафедра Автоматизованих систем
моніторингу навколишнього середовища

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Розробка оптимального додатку для складання розкладу занять навчальних закладів

Виконав студент 2 курсу, групи МК- 62
спеціальності 122 Компютерні науки
Гіржев Віталій Сергійович

Керівник к.т.н, доц.
Лавріненко Юліан Володимирович

Консультант _____

Резидент д.ф. – м.н, проф.
Ковальчук Володимир Володимирович

Одеса 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської та аспірантської підготовки
Кафедра Автоматизованих систем моніторингу навколишнього середовища

Рівень вищої освіти магістр
Спеціальність 122 Комп'ютерні науки
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри автоматизованих систем моніторингу навколишнього середовища
Лавріненко Ю.В.
" 29 " жовтня 2018 року

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ГІРЖЕВ ВІТАЛІЙ СЕРГІЙОВИЧ
(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка оптимального додатку для складання розкладу занять в навчальних закладах

керівник роботи: Лавріненко Юліан Володимирович, к.т.н., доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджений наказом вищого навчального закладу від „05" жовтня 2018року № 271 «С»

2. Строк подання студентом роботи: 10. 12. 2018 р.
3. Вихідні дані до роботи: Зразок розкладу ВУЗ, додаток Android

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз вимог до розкладу занять
2. Обгрунтовування структури,алгоритму та програми
3. Створення програмного коду

4. Управління проектом_____
5. Перелік графічного матеріалу:_____
Презентація роботи_____

6. Консультанти розділів роботи:

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

7. Дата видачі завдання:____„29” жовтня 2018 року_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Одержання завдання на виконання магістерської роботи	29.10.2018	100	відмінно
2	Пошук та підбір літератури та інших джерел інформації	31.10.2018	100	відмінно
3	Проведення аналізу предметної області і написання першого розділу пояснювальної записки до магістерської роботи	05.11.2018	90	добре
4	Проведення аналізу предметної області і написання другого розділу пояснювальної записки до магістерської роботи	08.11.2018	100	відмінно
5	Проведення аналізу предметної області і написання третього розділу пояснювальної записки до магістерської роботи	14.11.2018	100	відмінно
6	Рубіжна атестація	19-24.11.2018	95	добре
7	Проведення аналізу предметної області і написання четвертого розділу пояснювальної записки до магістерської роботи	30.11.2018	100	відмінно
8	Виготовлення презентації	03.12.2018	90	добре
9	Друкування пояснювальної записки	04.12.2018	100	відмінно
10	Одержання висновку керівника магістерської роботи	05.12.2018	100	відмінно
11	Проходження нормативного контролю	06.12.2018	100	відмінно
12	Переплетіння пояснювальної записки	07.12.2018	100	відмінно
13	Здача роботи ка кафедру та одержання висновку кафедри про допуск магістерської роботи до захисту	10.12.2018	100	відмінно
14	Перевірка на оригінальність тексту	13.12.2018	100	відмінно
15	Одержання рецензії	20.12.2018	100	відмінно
16	Здача готової магістерської роботи і документів секретарю АК	20.12.2018	100	відмінно
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)		90	відмінно

Студент _____ ____Гіржев В.С.___
(підпис) (прізвище та ініціали)

Керівник роботи _____ __Лавріненко Ю.В.___
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Представлена робота Гіржева Віталія Сергійовича на тему «Розробка оптимального додатку для складання розкладу занять навчальних закладів».

Мета магістерської роботи – це ознайомлення з аналогами програмного забезпечення, розробка та дослідження сучасного додатку і серверної частини, та тестування на локальному сервері .

Для досягнення зазначеної мети необхідно вирішення наступних завдань:

– ознайомитись з аналогами програмного забезпечення;

– виявити слабкі місця в існуючих аналогах;

– підібрати необхідні матеріали та інструменти для реалізації завдання;

– розробити та протестувати на хостингу.

У своїй магістерській роботі я розглядав різні додатки для перегляду розкладу навчальних закладів, як вони працюють, з’ясував які в них переваги та недоліки,також були описані існуючі аналогічні додатки.

Розроблений додаток який призначений для перегляду розкладу навчальних закладів, дозволяє переглядати, редагувати, а також добавляти інформацію про розклад та вчителів.

Магістерська робота містить: 76 с., рис. 33, табл. 0, додатки 4, використаних літературних джерел 26.

Ключові слова: додаток, проектування, сервер, мобільний пристрій, клієнт, архітектура.

SUMMARY

The work of Vitaliy Sergeyevich Girceva on the topic "Development of an Optimum application for Scheduling Classes at the Educational Institutions" is presented.

The purpose of the master's work - is Introduction to the analogues of the application, the development and research of the modern application and the server part, as well as testing on the local server.

To Achieve This Purpose and Patents for Solving The following task:

– get acquainted with analogues of software;
– visually impaired places in existing analogues;
– the necessary materials and tools for the task are selected;
– develop and test on the hosting.

In my life without master's thesis, I have reviewed the Different Appendices for viewing the Schedule of Educational Institutions, as they seem to work, find out what are the Advantages and Disadvantages in them, and have similar existing Annexes.

Application development What is the purpose of the School Schedule View, allows you to view, edit, and add information about the timetable and the teachers.

Master's thesis: 76 p., Fig. 33, tab. 0, Annexes 4, used literary sources 26.

Keywords: application, project, server, mobile attachment, key, architecture.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ,УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

JQUERY–бібліотека для JAVASCRIPT,за допомогою якої легко отриматі доступ до DOM

HTML– HyperText Markup Language – Мова розмітки гіпертекстових документів) – стандартна мова розмітки веб-сторінок в Інтернеті

PHP– гіпертекстовий препроцесор – скриптова мова програмування

JSON–JavaScriptObjectNotation– це текстовий формат обміну даними між комп'ютерами

MVC–Model-view-controller– архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення

OPENSERVER– це портативний локальний WAMP / WNMP сервер

AndroidStudio– інтегроване середовище розробки (IDE) для платформи Android

HTTP – протокол прикладного рівня

POST – метод запиту,який приймає данні з текстовіх полів

GET- метод запиту,який отримує інформацію із сервера

AJAX – підхід до побудови призначених для користувача інтерфейсів веб-додатків, що полягає в «фоновому» обміні даними браузера з веб-сервером

JQUERY – бібліотека JavaScript, що фокусується на взаємодії JavaScript і HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елементу DOM, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними

Denwer - набір дистрибутивів (локальний сервер WAMP) [2] і програмна оболонка, призначені для створення і налагодження сайтів (веб-додатків, іншого динамічного вмісту інтернет-сторінок) на локальному ПК (без необхідності підключення до мережі Інтернет) під керуванням ОС Windows

phpMyAdmin – веб - додаток з відкритим кодом, написаний на мові PHP і представляє собою веб-інтерфейс для адміністрування СУБД MySQL

MySQL – вільна реляційна система управління базами даних

XAMPP – кроссплатформенная збірка веб-сервера (розвиток LAMP), що містить Apache, MySQL, інтерпретатор скриптів PHP, мова програмування Perl і велика кількість додаткових бібліотек, що дозволяють запустити повноцінний веб-сервер

PHPExcel – бібліотека для роботи з exceldокументами на PHP

ZendEngine – віртуальна машина з відкритим кодом, широко відома як основна частина інтерпретатора PHP

Delphi – інтегроване середовище розробки ПО для MicrosoftWindows, MacOS, iOS і Android на мові Delphi (раніше мав назву ObjectPascal)

Github – найбільший веб-сервіс для хостингу IT-проектів і їх спільної розробки

Gradle – система автоматичного складання, побудована на принципах Apache Ant і Apache Maven

FTP – протокол передачі файлів по мережі

SSL – криптографічний протокол, який має на увазі більш безпечний зв'язок

API – опис способів (набір класів, процедур, функцій, структур або констант), якими одна комп'ютерна програма може взаємодіяти з іншою програмою

CMS – система управління вмістом сайту з відкритим вихідним кодом; написана на PHP

PostgreSQL – вільна об'єктно-реляційна система управління базами даних

JAVA – сильно типізований об'єктно-орієнтована мова програмування

CGI – стандарт інтерфейсу, використовуваного для зв'язку зовнішньої програми з веб-сервером

URL – однаковий локатор (визначник місцезнаходження) ресурсу

ВСТУП

В даний час інформаційні системи використовуються повсюдно. Вони стали невід'ємною частиною більшості робочих процесів. Їх використання призводить до поліпшення якості продукції, прискорення обробки інформації, зменшення трудо і економічних витрат.

Інформаційна система – являє собою сукупність організаційних, технічних, програмних і інформаційних засобів, об'єднаних в єдину систему з метою збору, зберігання, обробки і видачі необхідної інформації, призначена для виконання заданих функцій.

Автоматизація – один із напрямів науково-технічного прогресу, застосування саморегулюючих технічних засобів, економіко-математичних методів і систем управління, які звільняють людину від участі в процесах отримання, перетворення, передачі і використання енергії, матеріалів або інформації, істотно зменшують ступінь цієї участі або трудомісткість виконуваних операцій.

Один із способів автоматизації, це впровадження автоматизованих інформаційних систем в діяльність виробництва. Це дозволяє автоматизувати або частина дій виробництва, або все виробництво цілком. Все залежить від масштабів системи і масштабів підприємства.

Поєднанням інформаційних і автоматизованих систем є автоматизована інформаційна система (АІС) – сукупність програмно-апаратних засобів, призначених для автоматизації діяльності, пов'язаної зі зберіганням, передачею та обробкою інформації.

Інформаційні системи єфактором розвитку багатьох компаній. Подібні системи можуть застосовуватися в різних за розмірами, за видами діяльності організаціям. Але скрізь вони переслідують наступні завдання:

–підвищувати продуктивність роботи персоналу;

– покращувати якість обслуговування клієнтів;

– знижувати трудомісткості і напруженість праці персоналу, мінімізувати помилки в його діях.

Швидко зростаючий ринок мобільних додатків продовжує завойовувати найрізноманітніші сфери нашого життя. Люди активно

використовують мобільні девайси в повсякденному житті, що, безсумнівно, збільшує попит на самі різні додатки для мобільних телефонів. Саме цим пояснюється той факт, що більшість стартаперів вибирають розробку мобільних додатків в якості відправної точки для свого бізнесу. Мобільні пристрої – це та технологія, яку люди весь час тримають під рукою, так як з їх допомогою можна вкрай швидко отримати достовірні відомості.

Мета магістерської роботи: Створення програми для Одеського Державного Екологічного Університету за допомогою якого можна було б переглядати розклад навчального процесу, список пар і викладачів,а також надавати права користувачам для редагування та створення інформації.

Ця програма має бути розроблена для мобільних пристроїв на платформі Android тим самим не займаючи багато телефонної пам'яті.

Зручність його повинна бути в тому, що користувачеві не потрібно було шукати по інституту або питати у одногрупників розклад пар, а всього лише завантажити додаток, встановити на пристрій і мати доступ до інтернету.

Завдання магістерської роботи:

1. Вибір програмних засобів для реалізації додатку,його шаблон та архітектура.
2. Написання додатка розкладу пар в середовищі розробки Android-Studio.
3. Створення Віджету для додатку.
4. Створення серверної частини веб-додатки на PHP і JAvaScript.
5. Оформлення дизайну за допомогою мови гіпертекстової розмітки HTML і CSS.
6. Створення Адмін-панелі.
7. Надання сайту дінамачності та сучасності за допомогою бібілотекиJQuery та технології AJAX.
8. Після чого провести тестування на локальному сервері Open Server з подальшим розміщенням його на веб-хостинг.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.2 Аналіз існуючих аналогів

1.2.1Опис додатоку УНІВЕР

Якщо пошукати в інтернеті додатки, або сайти для навчальних закладів які розблені спеціальнодля переглядату розкладу то можна зробити висновок, що їх дуже мало і майже ніхто не займається їх розробкою. А якщо і займаються, то всі скаржаться на його неактуальність.Один з додатків «Універ», який створено для надання студентам розкладу предметів в різних університетах стан Росії і України.

При установці цього додатка розробник пропонує простий і зручний продукт для поліпшення навчального процесу від лідера ринка.Додаток, заряджений розкладом для студентів країн: Більше 440 університетів і інститутів, понад 35 000 груп з розкладом (рис. 1.1).

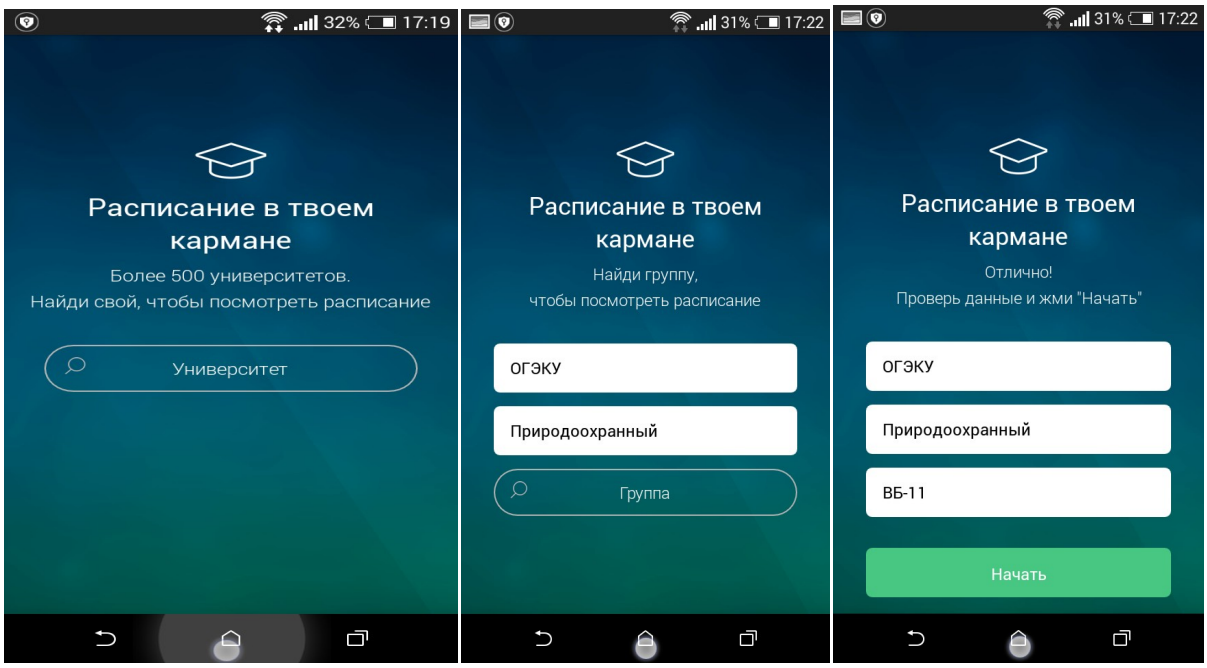

Рисунок 1.1– Інтерфейс програми Універ. Вибір навчального закладу, факультету та групи

Так само в цьому додатку є такі опції як досконале відображення розкладу:

- нумерація тижнів;
- потижневий перегляд розкладу;
- картка деталей пари;
- маркування типів пар;
- авторизація через соціальну мережу;
- сучасний інтерфейс;

Управління розкладом, тобто можна вносити або ж редагувати зміни прямо в додаток, Розклад викладачів, Журнал відвідуваності групи, який можна редагувати за наявності адмінки. «Універ» пропонує вам додати завдання з предметів, з можливістю зберігати все в одному місці, виставляючи термін виконання завдань, і прикріплюючи фото і предмети, щоб все встигнути і нічого не забути (рис. 1.2).

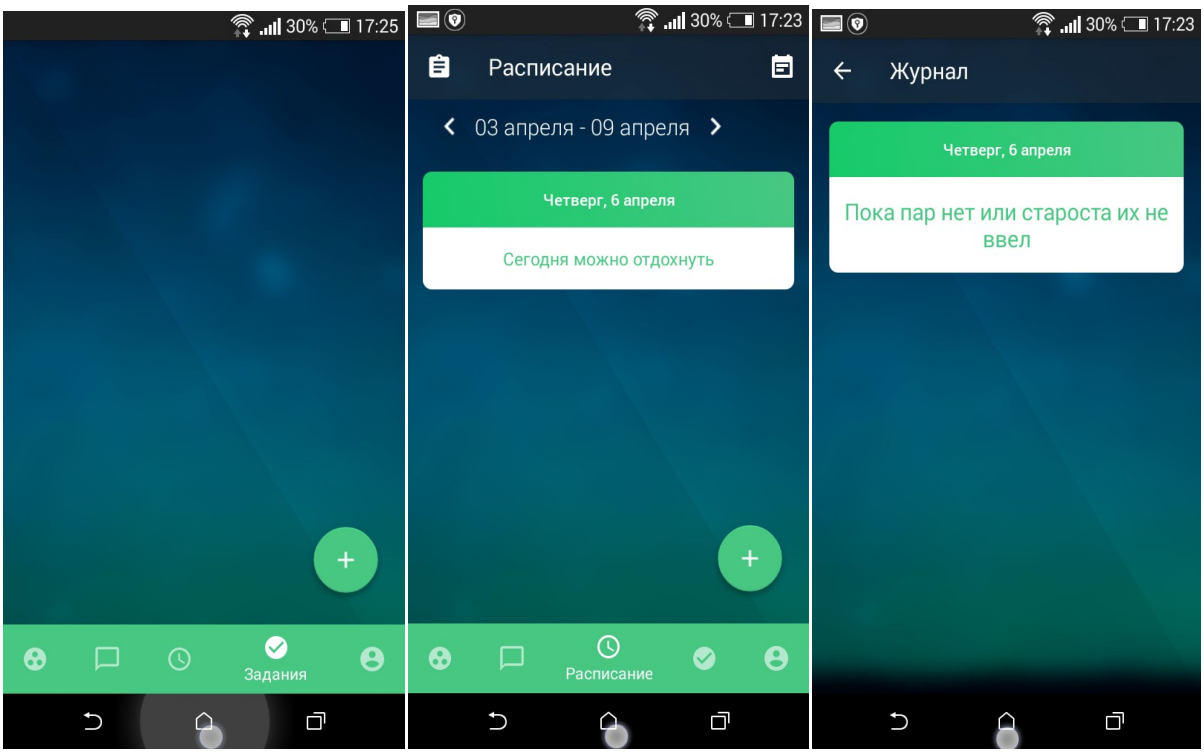

Рисунок 1.2– Інтерфейс редагування завдання і перегляду розкладу

1.2.2 Опис додатоку РОЗКЛАД МГТУ

Для прикладу можна взяти ще один додаток який створено конкретно для одного інституту МГТУ ім. Н. Е. Баумана. Воно дозволяє отримати актуальний розклад для будь-якої групи МГТУ ім. Н. Е. Баумана. Після першого завантаження з розкладом можна працювати автономно. Якщо буде обраний режим авто-оновлень, то додаток буде відображати всі знову з'явилися зміни. Поки що ця перша версія програми (рис. 1.3).

Надалі розробники обіцяють поліпшитидодаток, але є один мінус.Безкоштовна версія діє тільки місяць, після чого блокує доступ до розкладу.Якщо вірити відгукам,то додаток більш популярний в Россії,в Україні майже ним не користуються.

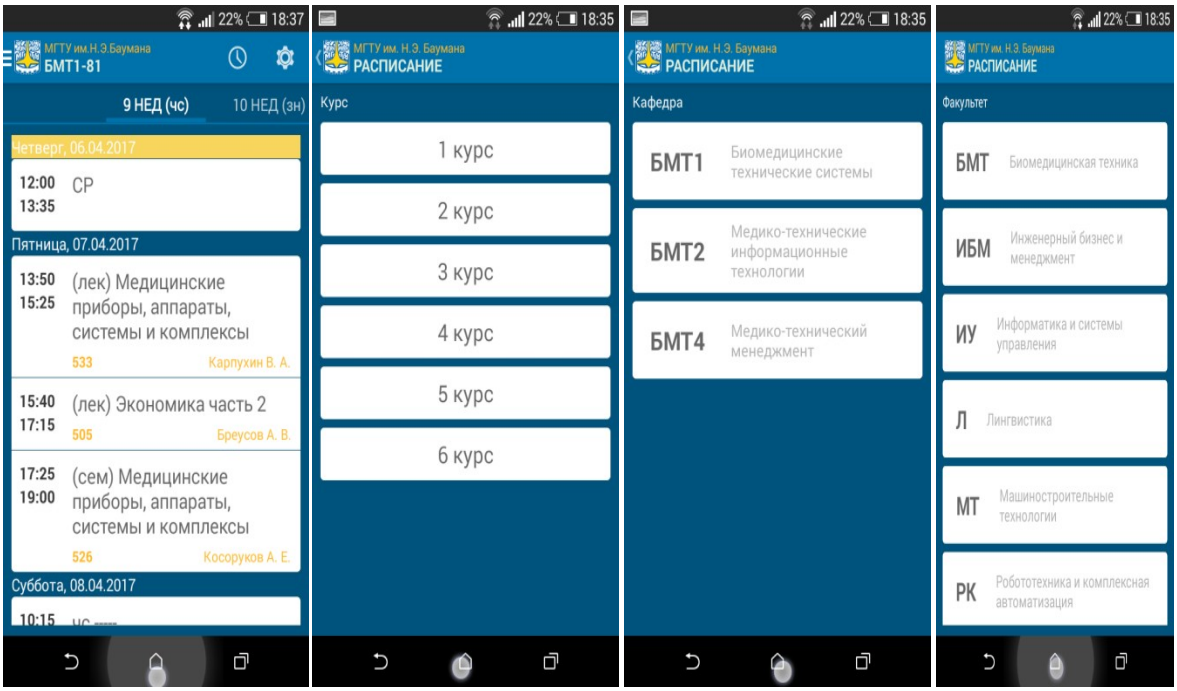

Рисунок 1.3 – Інтерфейс додатку МГТУ

1.2.3 Опис додатоку «Расписание занятий СумГУ»

Наступний проект, який я хочу привести для прикладу є СумГУ. Додаток дозволяє отримувати розкладом навчального процесу, список пар і викладачів (рис. 1.4).

Його створено спеціальноСтворено для студентів Сумського Державного Університету.

Функціональність і спосіб роботи максимально наближений до зручного для вас увазі:

- зручний, інтуїтивно-зрозумілий інтерфейс програми для взаємодії користувача і системи розкладу;
- можливість здійснення вибірки розкладу за кількома параметрами: групам, викладачам, аудиторіям;
- надійність роботи при найменшій витраті апаратних ресурсів пристрою і трафіку;
- можливість перегляду розкладу за минулі дні;

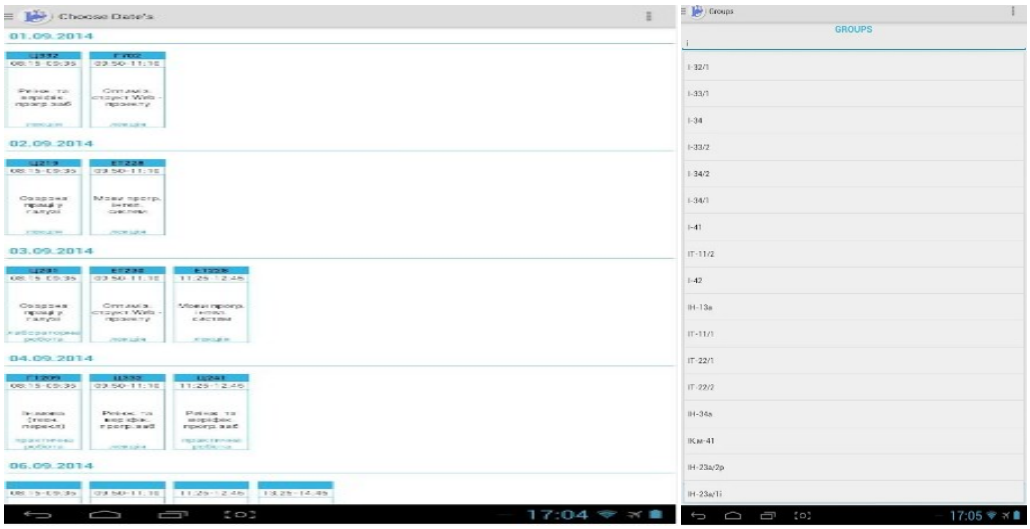

Рисунок 1.4–Інтерфейс додатку СумГУ

2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Діаграмадекомпозицій

Декомпозиція –це поділ складного об'єкта, системи, завдання на складові частини, елементи.

За допомогою діаграми декомпозиції першого рівнями можемо подивитись, з яких більш дрібних робіт складається робота "Получить расписание".У даній роботі були виділені наступні дочірні роботи: Ввести данні,відправити данні,отримати(Сервер),обробити вхідну інформацію таотримати результат.Методология IDEF0 предписывает построение иерархической системы диаграмм - единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция - система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности.

Кожна IDEF0 –діаграм а містить блоки і дуги. Блоки зображують функції модельованої системи. Дуги пов'язують блоки разом і відображають взаємодії і взаємозв'язку між ними (рис. 2.1).

Функціональні блоки (роботи) на діаграмах зображуються прямокутниками, які дають зрозуміти зазначені процеси, функції або завдання, які відбуваються протягом певного часу і мають розпізнавані результати. Ім'я роботи повинне бути виражене віддієслівним іменником, що позначає дію.

Після з'єднання граничних стрілок з роботами наступним кроком з'єднаємо роботи між собою.

Робота "Ввести данні" отримує на вході значення які введені в полях (тобто группа, факультет та курс.

Роботі "Відправити данні" для свого функціонування заповнені поля, які вона вона отримує від попередньої роботи (вихідна стрілка "Заповнені поля"). Отриману інформацію вона передає вона також передає роботі "Отримати(Сервер)" .

Далі робота "Отримати данні" збирає інформацію та передає наступній “Обробити вхідні данні”

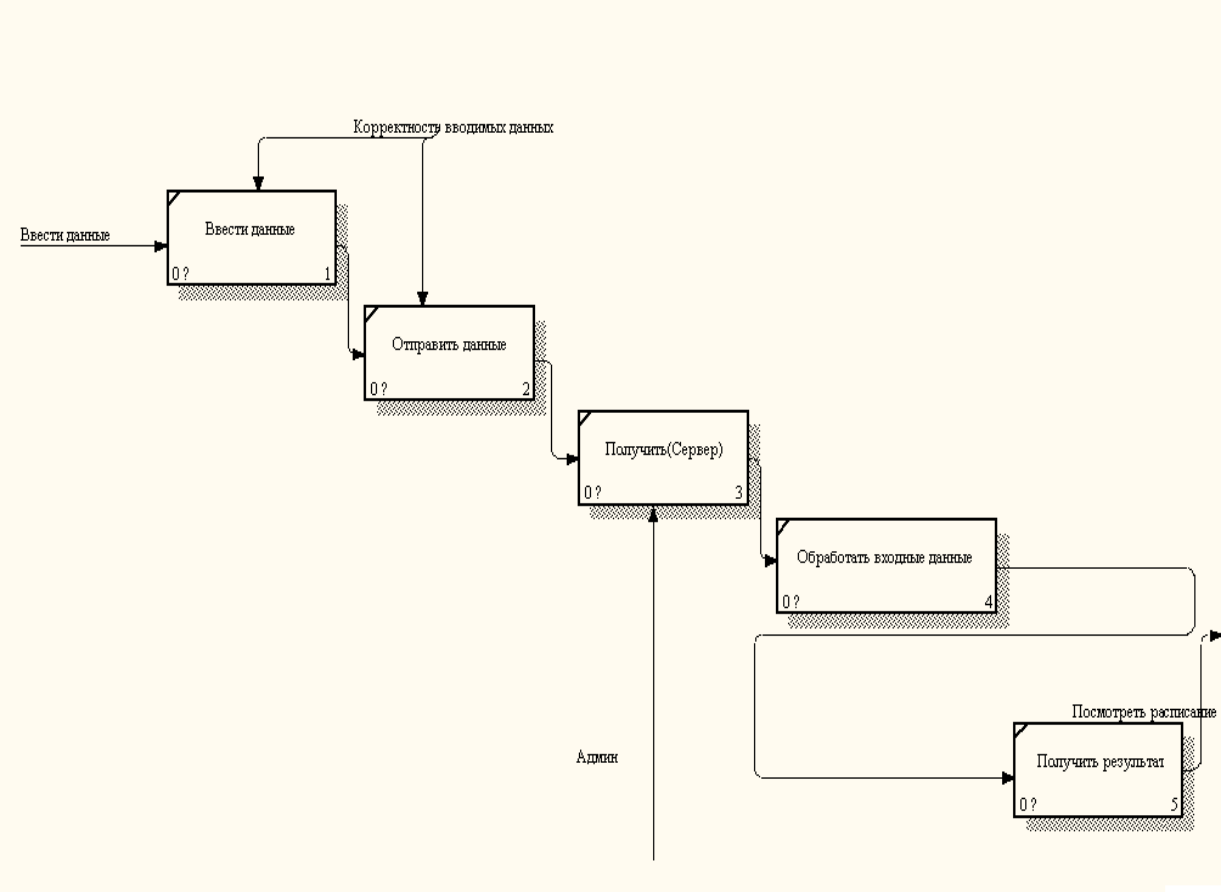

Рисунок 2.1 –Діаграма декомпозиці

2.2 Діаграма дерева вузлів

Діаграма дерева вузлів показує ієрархію робіт в моделі і дозволяє розглянути всю модель цілком, але не показує взаємозв'язку між роботами.

Процес створення моделі робіт є ітераційним, отже, роботи можуть змінювати своє розташування в дереві вузлів багаторазово. Щоб не заплутатися і перевірити спосіб декомпозиції, слід після кожної зміни створювати діаграму дерева вузлів. Втім, BPwin має потужний інструмент навігації по моделі - Model Explorer, який дозволяє представити ієрархію робіт і діаграм в зручному і компактному вигляді, проте цей інструмент не є складовою стандарту IDEF0.Оскільки дерево вузлів не обов'язково в якості верхнього рівня повинна мати контекстну роботу і мати довільну глибину. В одній моделі можна створювати безліч діаграм дерев вузлів. Ім'я дерева вузлів за замовчуванням збігається з ім'ям роботи верхнього рівня, а номер діаграми автоматично генерується як номер вузла верхнього уровя плюс літера "N", наприклад A0N. Якщо в моделі створюється два дерева вузлів, що мають в якості верхнього рівня одну і ту ж роботу, то за

замовчуванням діаграми отримають ідентичні номер і ім'я. Тому рекомендується при створенні діаграми дерева вузлів внести ім'я діаграми, відмінне від значення за замовчуванням (рис. 2.2).

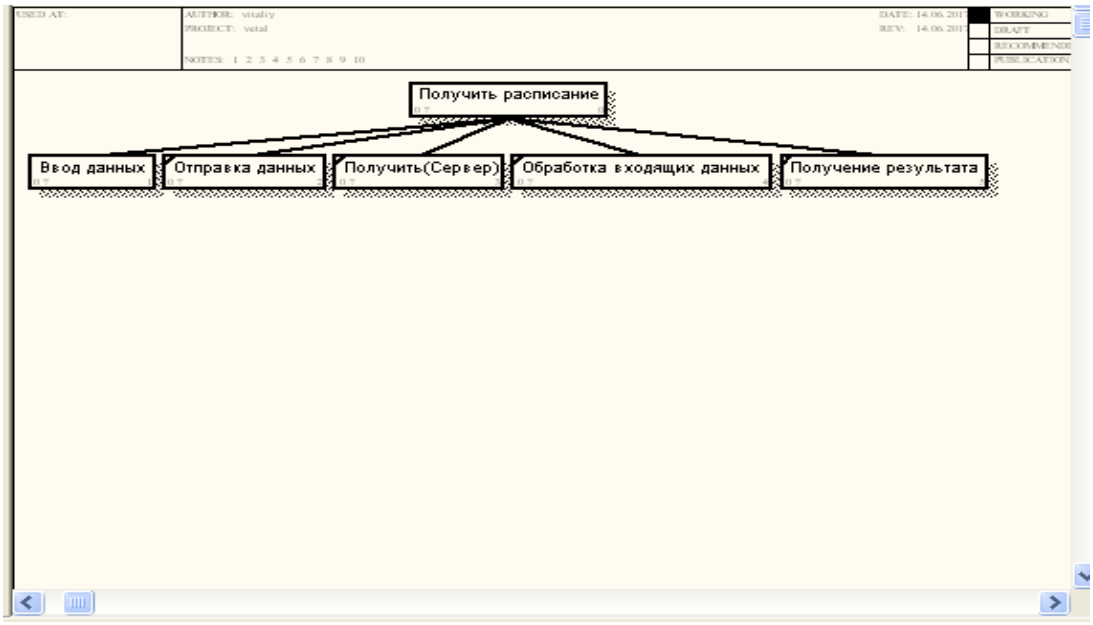

Рисунок 2.2– Діаграма дерева вузлів

2.3Контекстна діаграма

Контекстна – діаграма аналітична модель, яка описує абстрактну систему високого рівня. Вона визначає зовнішні для системи об'єкти, кото риє взаємодіють з нею, але нічого не відображає внутрішньої структури або поведінки системи (рис. 2.3).

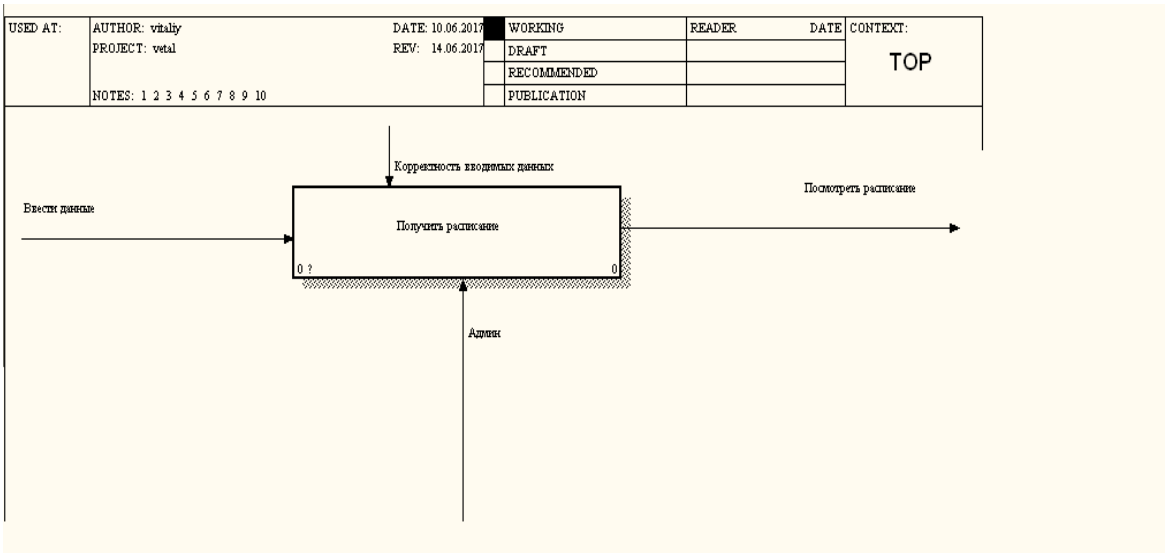

Рисунок 2.3–Контекстна діаграма

3 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ

3.1 Особливості використання HTML

HTML–стандартизований мову розмітки документів у Всесвітній павутині. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML). Мова HTML інтерпретується браузерами; отриманий в результаті інтерпретації форматований текст відображається на екрані монітора комп'ютера або мобільного пристрою.

У всесвітній павутині HTMLсторінки, як правило, передаються браузерам від сервера по протоколах HTTP або HTTPS, у вигляді простого тексту або з використанням шифрування.

Текстові документи, що містять розмітку на мові HTML (такі документи зазвичай мають розширення .html або .htm), обробляються спеціальними додатками, які відображають документ в його форматованому вигляді. Такі додатки, звані «браузерами» або «інтернет-оглядачами», зазвичай надають користувачеві зручний інтерфейс для запиту веб-сторінок, їх перегляду (і виведення на інші зовнішні пристрої) і, при необхідності, відправки введених користувачем даних на сервер. Найбільш популярними на сьогоднішній день браузерами є GoogleChrome, MozillaFirefox, Opera, InternetExplorer і Safari.[1].

При верстці (грубо кажучи, створення сторінок) HTMLсторінок це необхідно пам'ятати, і переглядати їх в найпопулярніших браузерах типу MozillaFirefox, InternetExplorer і Opera.

Код розмітки в HTML складається з так званих «тегів». Теги надають інформацію браузерам про форматування і розмітки сторінки. Назва тега полягає в кутові дужки «<» і «>». Деякі теги бувають відкриваються і закриваються, наприклад, тег b, який робить текст жирним (рис. 3.1).

Щоб писати на HTML треба його просто знати. Однак, з огляду на величезну популярність Інтернету, і величезна кількість користувачів, що бажають зробити свої сторінки, було створено безліч редакторів, що спрощують роботу з кодом. Один з найпростіших з них - це FrontPage, що входить в пакет MicrosoftOffice. Це програмне забезпечення (ПО) досить просто в зверненні. І все ж FrontPage має ряд істотних мінусів: засмічує

сторінки (до 70% зайвого коду), знижує швидкодію, має обмежені можливості дизайну. [2].

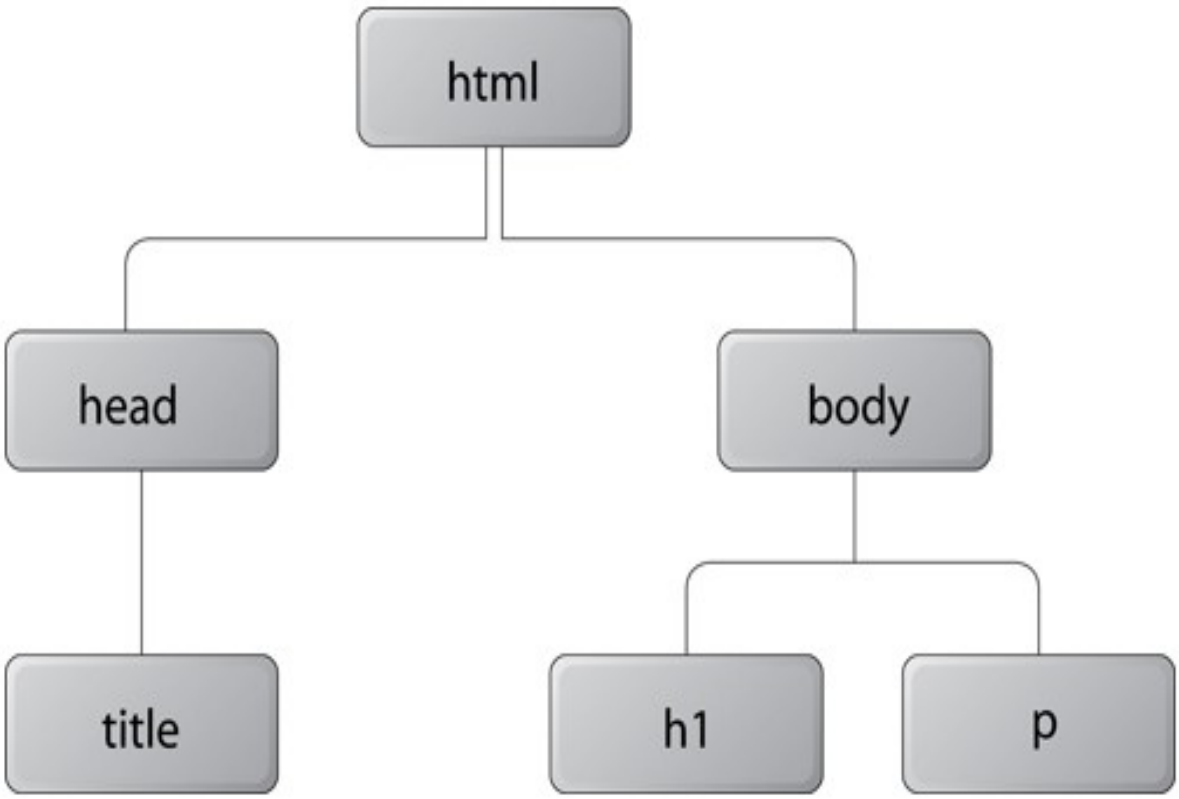

Рисунок 3.1 – СтруктураHTMLрозмітки

3.2 Використання JSON для створення системи

JSON– текстовий формат обміну даними, заснований на JavaScript. Формат JSON був розроблений Дугласом Крокфорд (рис. 3.2).

Незважаючи на походження від JavaScript (точніше, від підмножини мови стандарту ECMA-262 1999 роки), формат вважається незалежним від мови і може використовуватися практично з будь-якою мовою програмування. Для багатьох мов існує готовий код для створення і обробки даних в форматі JSON. [3].

За рахунок своєї лаконічності в порівнянні з XML, формат JSON може бути більш підходящим для сериализации складних структур. Якщо говорити про веб-додатках, в такому ключі він доречний в задачах обміну даними як між браузером і сервером (AJAX), так і між самими серверами (програмні HTTP-сполучення).

Оскільки формат JSON є підмножиною синтаксису мови JavaScript, то він може бути швидко десеріалізован вбудованою функцією eval ().

Крім того, можлива вставка цілком працездатних JavaScript-функцій. У мові PHP, починаючи з версії 5.2.0, підтримка JSON включена в ядро у вигляді функцій json_decode () і json_encode (), які самі перетворять типи даних JSON в відповідні типи PHP і навпаки.

У JSON використовуються їх наступні форми:

Об'єкт–це неврегульована безліч пар ім'я / значення. Об'єкт починається з символу {і закінчується символом}. Кожне значення слід за: і пари ім'я / значення відділяються комами.Масив – це впорядкована множина значень. Масив починається символом і . Значення відокремлюються комами.Значення може бути рядком в подвійних лапках, або числом, або true, або false, або null, або об'єктом, або масивом. Ці структури можуть бути вкладені одна в одну.Рядок - це впорядкована множина з нуля або більше символів юникода, укладені в подвійні лапки, з використанням escape-послідовностей починаються з зворотної косої межі (backslash). Символи представляються простий рядком. [4].

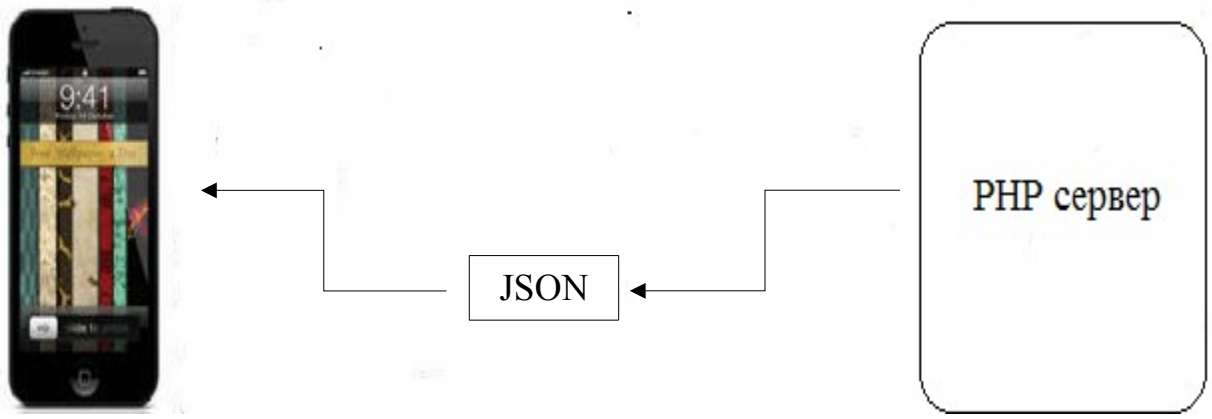

Рисунок 3.2 – Отримання даних у форматі JSON

Хоча JSON простий для розуміння і використання, він є дуже корисним і гнучким інструментом для передачі даних між додатками і комп'ютерами, особливо при використанні AJAX. Якщо ви плануєте розробляти AJAX додаток, то немає сумнівів, що JSON стане найважливішим інструментом у вашій майстерні.

3.3 Тестування на локальному сервері OPEN SERVER

Це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників з урахуванням їх рекомендацій і побажань.

Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості з адміністрування та налаштування компонентів. Платформа широко використовується з метою розробки, налагодження і тестування веб-проектів, а так само для надання веб-сервісів в локальних сетях.Хотя спочатку програмні продукти, що входять до складу комплексу, не розроблялись спеціально для роботи один з одним, така зв'язка стала дуже популярною серед користувачів Windows, в першу чергу через те, що вони отримували безкоштовний комплекс програм з надійністю на рівні Linux серверів. [5].

Основні можливості OpenServer:

- швидкий запуск і завершення роботи;
- відсутність прив'язки до конкретного ПК;
- автозапуск сервера при запуску керуючого ПО;
- управління доменами в декількох режимах;
- можливість монтування віртуального диска;
- управлння через командний рядок;
- демонстрація логів компонентів;
- функція перемикання між модулями HTTP, MySQL, PHP;
- робота комплексу програм на локальному, мережевому і зовнішньому IP адресу;
- підтримка SSL без додаткових налаштувань;
- створення домену за допомогою створення простої папки;
- конвертація доменних імен;
- підтримка доменів на кирилиці, доменних покажчиків;
- забезпечення захисту сервера від доступу ззовні;
- можливість створення локального поддомена і забезпечення одночасної видимості основного домену в мережі.

Локальний сервер Open Server призначений для роботи на ОС Windows і включає в себе як набір програмних модулів WAMP, так і WNMP. Сервер може застосовуватися і на платформі Linux. Open Server - це відносно новий, швидко набирає обертів, проект, в якому використовуються останні версії web-серверів, СУБД та інших необхідних елементів для web-розробки. Платформу Open Server відрізняє наявність графічного інтерфейсу, значно підвищує зручність роботи з цим локальним сервером. [6].

Зручність і простота управління безумовно не залишать вас байдужими, за час свого існування OpenServer зарекомендував себе як першокласний і надійний інструмент необхідний кожному веб-

майстру.Для того щоб запустити сайт на локальному сервері його потрібно завантажитив папку domains.

Для реального тестування веб-додатків використовуються популярні локальні сервера, такі як Denwer і Open Server. Далі розглянемо кожен з них.

Денвер називають ще джентльменським набором розробника, так як він складається з усіх необхідних компонентів. Даний локальний сервер рекомендують для початківців, тому що він легкий і простий в налаштуванні. Популярний Денвер має підтримку і базу знань. Включає в себе: веб-сервер Apache; інтерпретатор мови PHP; інтерпретатор мови PERL; базу даних MySQL; імітація сервера Email пошти; движок phpMyAdmin для керування MySQL; емулятор sendmail і сервера SMTP з підтримкою роботи спільно з PHP, Perl і ін (рис. 3.3).

Модульність, розширюваність, компактність. Немає необхідності викачувати многомегабайтние дистрибутиви окремих компонентів. Базова версія Денвера, що включає Apache + SSL + PHP5 + MySQL5 + phpMyAdmin, має розмір всього близько 5.4мб і при цьому повністю функціональна.

Open Server містить в собі більш пізні версії ПО web-розробки, що видно при відвідуванні офіційних сайтів розробників цих локальних серверів. [7].

Використання Open Server не припускав тотального відмови від Денвера. Або будь-якого іншого інструменту (наприклад популярного XAMPP, який я теж подивлюся). Вони портативні, вони можуть існувати паралельно, не заважаючи один одному. Кожен інструмент має свою нішу. Open Server, можливо, більш зручний для стаціонарної розробки, з безліччю модулів і програмного забезпечення. А Денвер більш мобільний і легкий, що уміщається на флешку в 250 мб або заповнений мультимедійним контентом смартфон(рис. 3.3).

Разом з тим, OpenServer досить вимогливий до системних налаштувань. Для коректної роботи необхідно мати права адміністратора, доступ до файлу c: \ windows \ system32 \ drivers \ etc \ hosts, вільний порт 80. Цілком можлива ситуація, що комплект не буде працювати спільно з Skype, який також використовує восьмидесятих порт для роботи, а також з браундмауером або антивірусом. З цими проблемами стикаються багато веб-сервери під Windows, тому вони досить відомі і рішення давно знайдені. У довідці OpenServer (http://open-server.ru/help.html) всі типові питання розглянуті і дані відповіді на них, так що якщо веб-сервер не завантажується, слід подивитися логи сервера і заглянути в довідку.

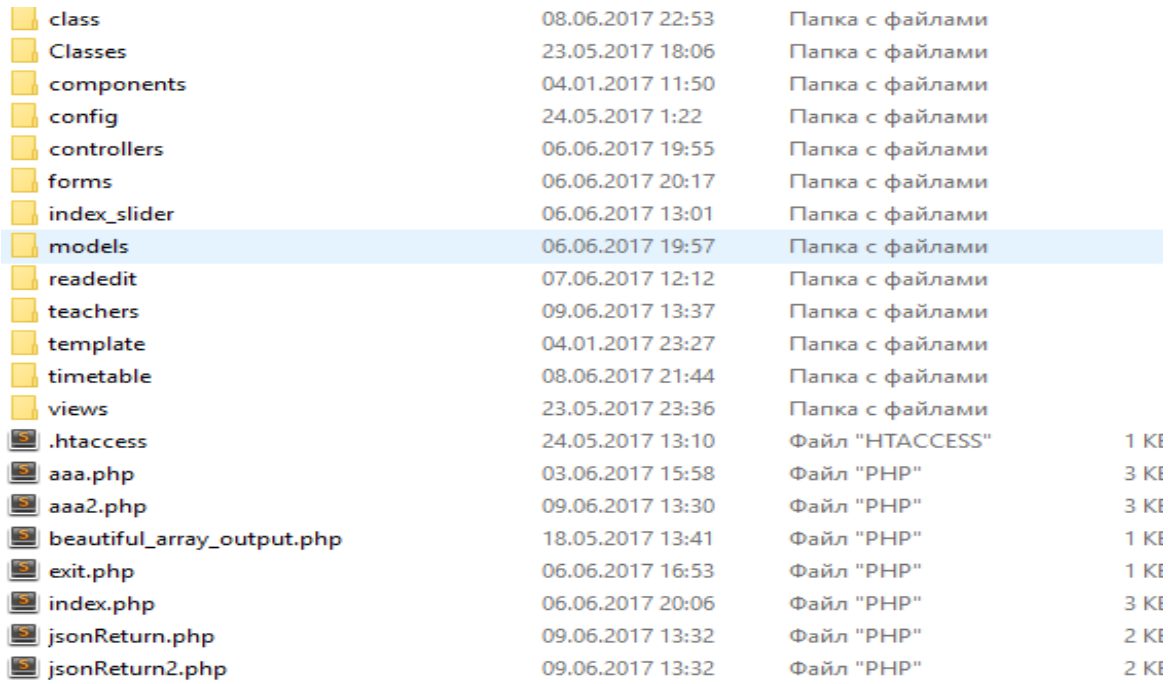

Рисунок 3.3 – Завантаження сайту на локальний сервер

Спільними характеристиками з Денвером є їх безкоштовний доступ і широкий функціонал програмного забезпечення для вже безпосередньо роботи з сайтами.

Плюс, згаданий вище графічний інтерфейс. Дистрибутив Denwer'a важить значно менше пакета інсталяції Open Server'a (8,2 Мб проти 135 Мб (або 424 Мб для повної версії)). До того ж, почати працювати з Denwer значно простіше.

Але робота в Open Server зручніше в цілому. У ньому є можливість задавати більш гнучкі налаштування і використовувати як Apache, так і Nginx. [6].

3.4 Обробка та оформлення звіту в EXCEL на PHP

Не рідко при розробці якогось проекту, виникає необхідність у формуванні звітної статистики. Якщо проект розробляється на Delphi, C # або наприклад, на C ++ і під Windows, то тут проблем немає. Всього лише необхідно скористатися COM об'єктом. Але справи йдуть інакше, якщо необхідно сформувати звіт в форматі excel на PHP. І щоб це творіння функціонувало на UNIX-подібних системах. Але, на щастя, не так все погано. І бібліотек для цього вистачає. Я свій вибір зупинив на PHPExcel.

Я вже пару років працюю з цією бібліотекою, і залишаюся задоволений. Оскільки вона є кроссплатформенной, то не виникає проблем з перенесенням.

PHPExcel дозволяє робити імпорт і експорт даних в excel. Застосовувати різні стилі оформлення до звітів. Загалом, все на висоті. Навіть є можливість роботи з формулами (сам я не пробував). Тільки пам'ятайте, що вся робота (читання і запис) повинна вестися в кодуванні utf-8.

Дуже часто виникає необхідність виділити в звіті деякі дані. Зробити виділення шрифту або застосувати рамку з заливкою фону для деяких осередків і т.д. Що дозволяє сконцентруватися на найбільш важливих речей (правда може і навпаки відволікти). Для цих цілей в бібліотеці PHPExcel є цілий набір стилів, які можна застосовувати до осередків в excel. Є звичайно в цій бібліотеці невеликий "мінус" - не можна застосувати стиль до кількох осередкам одночасно, а тільки до кожної індивідуально. Але це не створює дискомфорту при розробці web-додатків.

3.5 Використання PHP

Приблизно 85% веб-сайтів написано на мові програмування PHP. PHP- «Інструмент для створення персональних веб-сторінок» – скриптова мовно бщего призначення, інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів.

PHP - це в кінцевому рахунку текст, одержуваний веб-сервером і перетворюваний в набір команд і інформації для веб-браузера. Оскільки робота ведеться з простим текстом, щоб приступити до програмування на PHP, потрібно буде всього лише ознайомитися з самим мовою PHP, і краще за все цей зробити, встановивши PHP на свій комп'ютер, навіть якщо більшість програм буде запускатися на веб-сервері.

Мова PHP мені симпатичний як дуже поширений, легкий в установці інструмент. При правильному використанні дозволяє отримати дуже хороші результати, як, втім, і будь-який інший інструмент. Це один з основних мов, на яких я програмую.

Мова і його інтерпретатор (Zend Engine) розробляються групою ентузіастів в рамках проекту з відкритим кодомПроект поширюється під власною ліцензією, несумісною.

PHP-код програми виконується на стороні сервера. Після того, як користувач здійснив на сайті якусь дію, наприклад клік по посиланню в меню, з метою перейти на іншу сторінку сайту, браузер посилає запит серверу на відповідну сторінку з PHP-кодом. Далі, PHP-код обробляється інтерпретатором PHP і генерується HTML-код, який повертається сервера. Сервер в свою чергу, передає цей HTML-код назад браузеру. В результаті користувач бачить відображення в браузері нової сторінки, що має свій HTML код. При перегляді ж вихідного коду цієї сторінки видно буде тільки HTML код, а PHP-код залишається недоступним для перегляду.

Великий плюс мови PHP полягає в тому, що PHP-код можна впроваджувати безпосередньо в HTML-файли. PHP-код вбудовується в HTML сторінки за допомогою кутових дужок і знака питання (рис. 3.4).

PHP відрізняється від JavaScript тим, що PHP-скрипти виконуються на сервері і генерують HTML, який надсилається клієнту. Якби у вас на сервері був розміщений скрипт, подібний вищенаведеного, клієнт отримав би тільки результат його виконання, але не зміг би з'ясувати, який саме код його справив. Ви навіть можете налаштувати свій сервер таким чином, щоб звичайні HTML-файли оброблялися процесором PHP, так що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипта.

PHP вкрай простий для освоєння, але разом з тим здатний задовольнити запити професійних програмістів. Не лякайтеся довгого списку можливостей PHP. Ви можете швидко почати, і вже протягом декількох годин зможете створювати прості PHP-скрипти. [7].

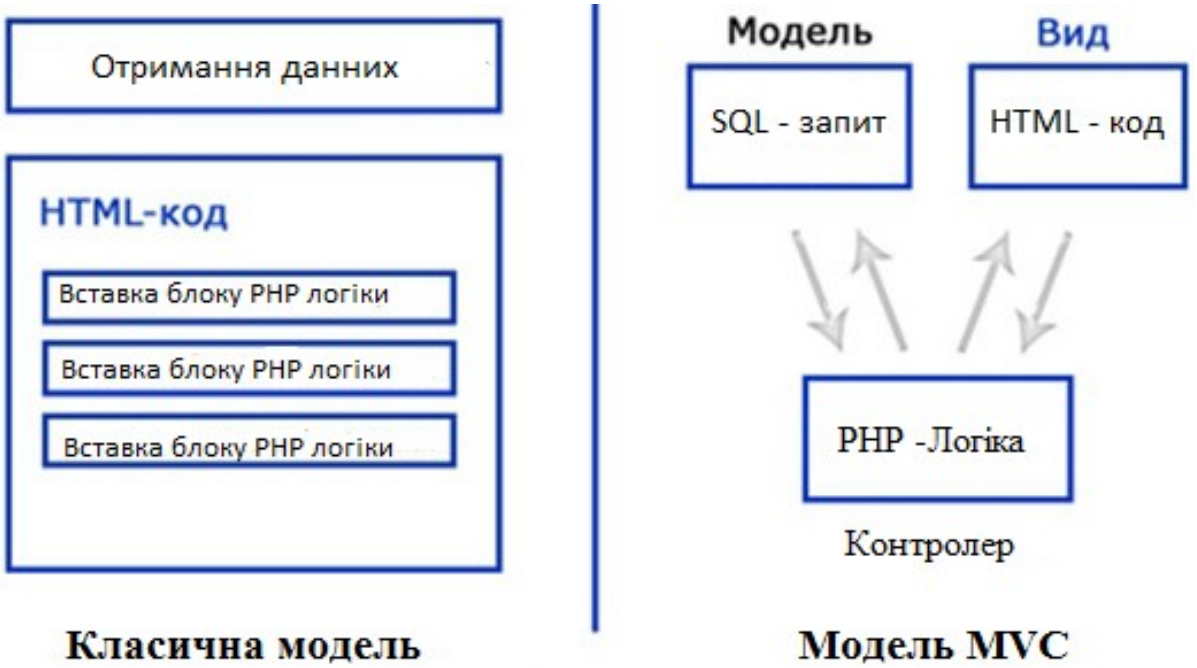

Рисунок 3.4 – Зв’язок між PHPта HTML

3.6 Опис середовища розробки ANDROIDSTUDIO

AndroidStudio – це інтегроване середовище розробки (IDE) для роботи з платформою Android, анонсована 16 травня 2013 року на конференції GoogleI / O.

IDE перебувала у вільному доступі починаючи з версії 0.1, опублікованій в травні 2013, а потім перейшла в стадію бета-тестування, починаючи з версії 0.8, яка була випущена в червні 2014 року. Перша стабільна версія 1.0 була випущена в грудні 2014 року, тоді ж припинилася підтримка плагіна AndroidDevelopmentTools (ADT) для Eclipse.AndroidStudio, заснована на програмному забезпеченні IntelliJIDEA від компанії JetBrains, офіційне засіб розробки Android додатків. Дане середовище розробки доступна для Windows, OS X і Linux.

Певні особливості пізніше розгорнуті для користувачів так як програмне забезпечення розвивається; поки, передбачені наступні функції:

- Живі макети (layout): редагувальнік WYSIWYG – живе кодування подання (rendering) програми в реальному часі.
- Консоль розробника: підказки щодо оптимізації, допомога по переводу, стеження за напрямком, агітації та акції – метрики Google аналітики.
- Резерви бета релізів і покрокові релізи;
- базування на Gradle;
- Android – орієнтований рефакторинг і швидкі виправлення;
- Lint утиліти для охоплення продуктивності, юзабіліті, сумісності версій і інших проблем;
- Використання можливостей ProGuard і підписів до програм;
- Шаблони для створення поширених Android дизайнів і компонентів.

Багатий редактор макетів (layouts) що дозволяє користувачам перетягнути і покласти (drag-and-drop) компоненти користувальницького інтерфейсу, як варіант, подивитися одночасно макети (layouts) на різних конфігураціях екранів.

Android Studio прийшло на зміну плагіна ADT для платформи Eclipse. Середа побудовано на базі вихідних текстів продукту IntelliJ IDEA Community Edition розвивається компанією JetBrains. Android Studio

розвивається в рамках відкритої моделі розробки і розповсюджується під ліцензією Apache 2.0.

Бінарні збірки підготовлені для Linux (для тестування використаний Ubuntu), Mac OS X і Windows. Середовище надає кошти для розробки додатків не тільки для смартфонів і планшетів, але і для носяться пристроїв на базі Android Wear, телевізорів (Android TV), очок Google Glass і автомобільних інформаційно-розважальних систем (Android Auto). Для додатків, спочатку розроблених з використанням Eclipse і ADT Plugin, підготовлений інструмент для автоматичного імпорту існуючого проекту в Android Studio.

Середовище розробки адаптовано для виконання типових завдань, що вирішуються в процесі розробки додатків для платформи Android. У тому числі в середу включені кошти для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування додатків, що працюють на пристроях з екранами різного дозволу (планшети, смартфони, ноутбуки, годинники, окуляри і т.д.). Крім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій, таких як нова уніфікована підсистема складання, тестування і розгортання додатків, заснована на складальному інструментарії Gradle і підтримуюча використання коштів безперервної інтеграції.

Для прискорення розробки додатків представлена колекція типових елементів інтерфейсу і візуальний редактор для їх перегляду, надає зручний попередній перегляд різних станів інтерфейсу додатку (наприклад, можна подивитися як інтерфейс буде виглядати для різних версій Android і для різних розмірів екрану(рис. 3.5).

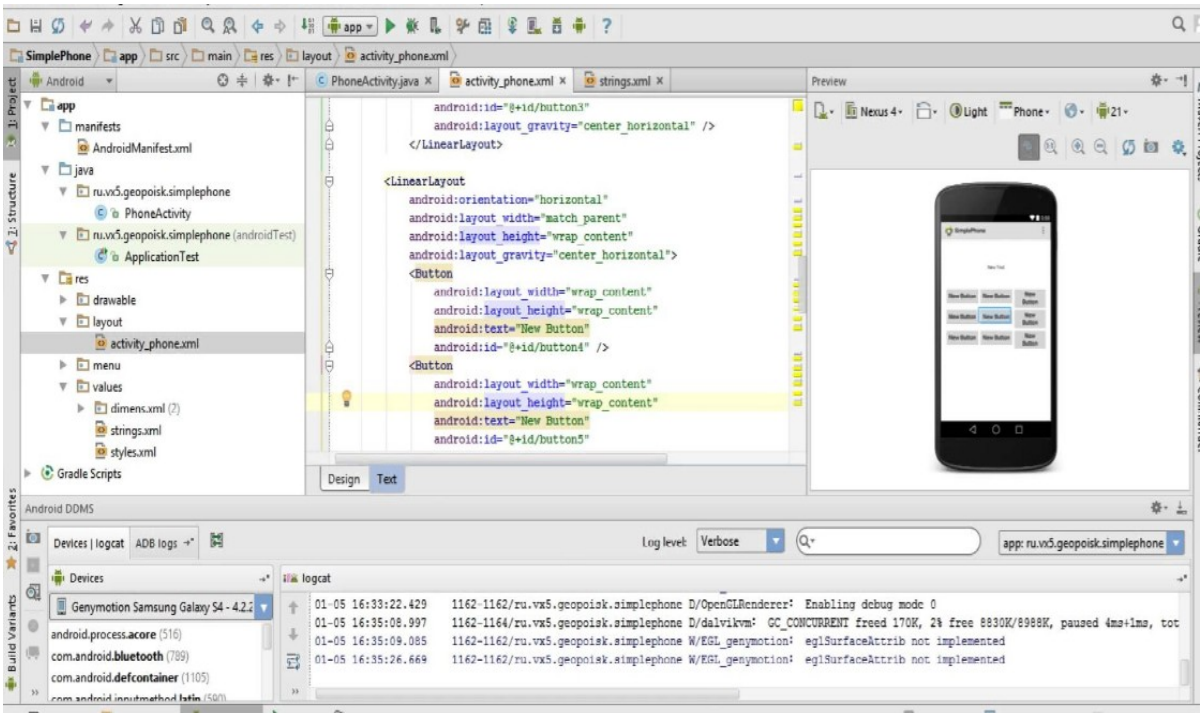

Рисунок 3.5 – Інтерфейс середовища розробкиAndroidStudio

Длястворенняінестандартнихінтерфейсівприсутніймайстерстворенняв ласнихелементівоформлення, підтримуєвикористанняшаблонів. У середу вбудовані функції завантаження типових прикладів коду з GitHub.

До складу також включені пристосовані під особливості платформи Android розширені інструменти рефакторинга, перевірки сумісності з минулими випусками, виявлення проблем з продуктивністю, моніторингу споживання пам'яті та оцінки зручності використання. У редактор доданий режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API. Інтегрована підтримка оптимізатора коду ProGuard. Засоби генерації цифрових підписів. Надано інтерфейс для управління перекладами на інші мови. [8].

3.7 Використання MS Excel

При згадці баз даних (БД) в першу чергу, звичайно, в голову приходять всякі розумні слова типу SQL, Oracle, 1C або хоча б Access. Безумовно, це дуже потужні (і недешеві в більшості своїй) програми, здатні автоматизувати роботу великої і складної компанії з купою даних. Біда в тому, що іноді така міць просто не потрібна. Програма яку створює розробникможе бути невеликою і з відносно нескладними бізнес-

процесами, але автоматизувати його теж хочеться. Причому саме для маленьких компаній це, найчастіше, питання виживання.

Microsoft Excel– табличний процесор, програма для роботи з електронними таблицями, створена корпорацією Microsoft для Microsoft Windows, Windows NT і Mac OS. Програма входить до складу офісного пакету Microsoft Office (рис. 3.6).

Excel– програмованій табличний калькулятор. Всі розрахунки в Excel виконують формули. Excel считает формулою все, что починається Із знаку «=». Если в комірці Написати просто «1 +1», то Excel немає буде обчислюваті цею віслів. Однако, если Написати «= 1 +1» і натіснуті клавішу Enter, в комірці з'явиться результат обчислення вирази – число 2. После натіскання клавіші Enter формула не пропадає, ее можна Побачити в панелі ІНСТРУМЕНТІВ «Рядок формул».

У Формулі можна використовуват Різні типи Операторів (аріфметічні и т.п.), текст, посилання на агентство комірку або діапазон комірок, круглі дужки, іменовані діапазоні. Природно, в формулах дотрімується ПРІОРИТЕТ Виконання операцій (множення віконується Ранее Додавання и т.п.). Для Зміни порядку Виконання операцій Використовують круглі дужки.

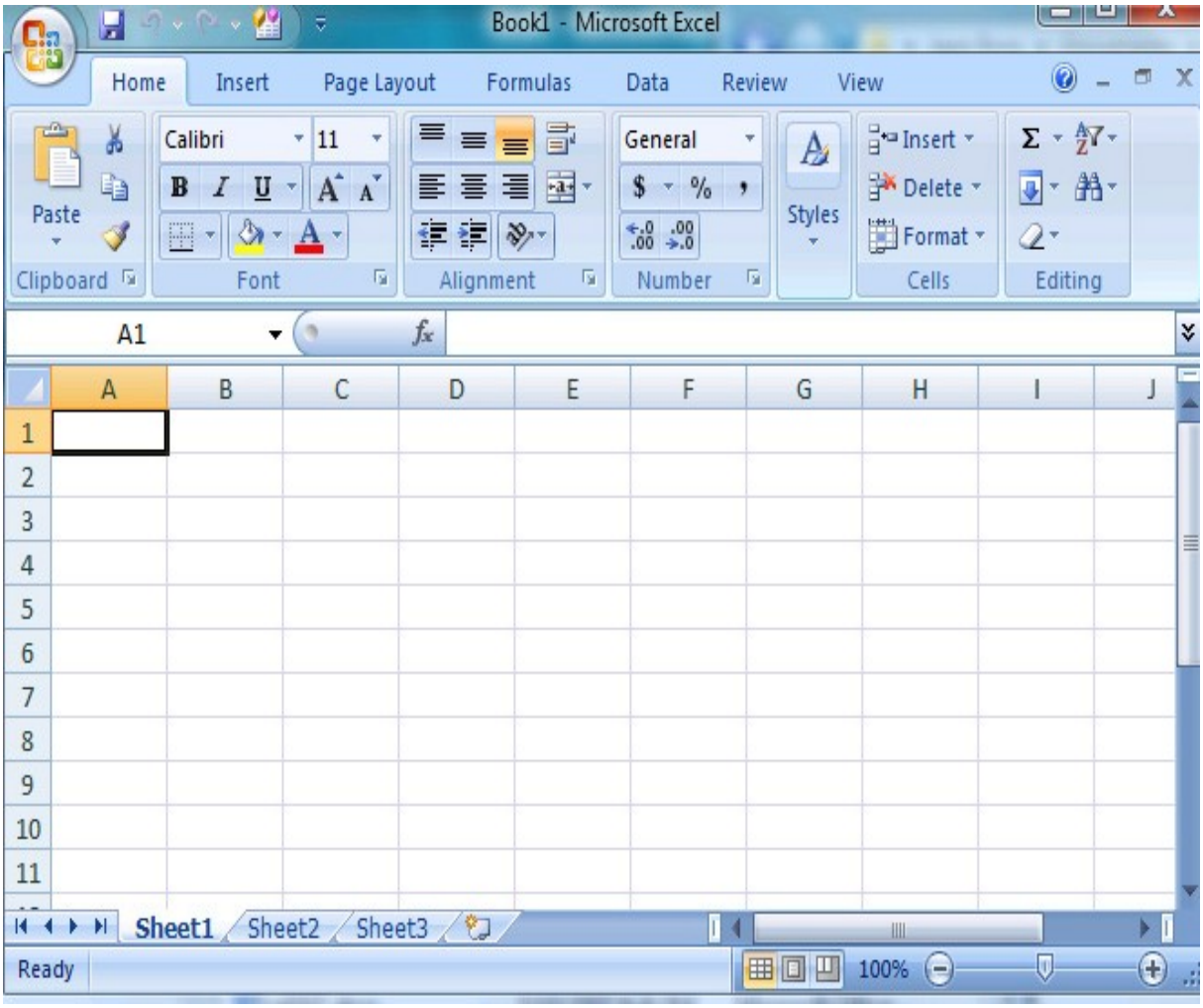

Рисунок 3.6 – Інтерфейс программи MSExcel

3.8 Вибір шаблона проектування MVC

Model-View-Controller –схема поділу Даних програми, призначення для користувача інтерфейсу и керуючої логіки на три Окремо компоненти: модель, уявлення и контролер таким чином , що модіфікація шкірного компонента может здійснюватіся Незалежності.

Основна мета застосування цієї концепції полягає в відділенні бізнес-логіки (моделі) від її візуалізації (уявлення, виду). За рахунок такого поділу підвищується можливість повторного використання коду. Найбільш корисне застосування даної концепції в тих випадках, коли користувач повинен бачити ті ж самі дані одночасно в різних контекстах і або з різних точок зору.

Контролер керує запитами користувача (одержувані у вигляді запитів HTTPGET або POST, коли користувач натискає на елементи інтерфейсу для виконання різних дій). Його основна функція – викликати і

координувати дію необхідних ресурсів і об’єктів, потрібних для виконання дій, що задаються користувачем. Зазвичай контролер викликає відповідну модель для задачі і вибирає відповідний вид.Модель – це дані і правила, які використовуються для роботи з даними, які представляють концепцію управління додатком. У будь-якому додатку вся структура моделюється як дані, які обробляються певним чином. Що таке користувач для додатка – повідомлення або книга? Тільки дані, які повинні бути оброблені відповідно до правил (дата не може вказувати в майбутнє, e-mail повинен бути в певному форматі, ім'я не може бути довшим X символів, і так далі).Контролер – блок, який отримує дані від користувача, обробляє, нормалізує їх, також виконує перевірку правильності введення і передає ці оброблені дані в потрібну модель. Також він приймає дані від моделі, потім вибирає потрібну уявлення, наповнює його даними і відображає на екрані браузера. Але при цьому, ще раз уточню, контролер не повинен містити в собі ніякої інформації про зовнішній вигляд веб-додатки. Контролер можна розглянути як сполучна ланка між поданням і моделлю.

Модель – це основа логіки нашого веб-додатки – вона відповідає за розрахунки, вибірку інформації з бази даних, зміна інформації в БД і т.д. Модель можна представити як бібліотеку різних функцій, що дозволяють реалізовувати функціонал нашого застосування. Тобто – це блок, який отримує дані від контролера, далі на основі цих даних робить необхідні перетворення, або знову ж вибирає дані з БД або змінює їх, а потім передає результат своєї роботи назад контролеру.

Модель дає контролеру уявлення даних, які запросив користувач (повідомлення, сторінку книги, фотоальбом, тощо). Модель даних буде однаковою, незалежно від того, як ми хочемо представляти їх користувачеві. Тому ми вибираємо будь-який доступний вид для відображення даних.

Модель містить найбільш важливу частину логіки нашого застосування, логіки, яка вирішує завдання, з якою ми маємо справу (форум, магазин, банк, тощо). Контролер містить в основному організаційну логіку для самого додатка (дуже схоже на ведення домашнього господарства).

Вид забезпечує різні способи представлення даних, які отримані з моделі. Він може бути шаблоном, який заповнюється даними. Може бути кілька різних видів, і контролер вибирає, який підходить якнайкраще для поточної ситуації.Веб додаток зазвичай складається з набору контролерів, моделей і видів. Контролер може бути влаштований як основний, який

отримує всі запити і викликає інші контролери для виконання дій в залежності від ситуації(рис. 3.7).

Вид – компонента перетворює дані, отримані від Моделі, в форму, зручну для інтерфейсу з користувачем. У Web-додатках Вид – головним чином HTML сторінка. «Вид» отримує дані від «Моделі» (через «Контролер») і направляє їх в інтерфейс користувача. Дана компонента ні за яких обставин не може стати причиною зміни бази даних, вона служить тільки для виведення даних, отриманих від Моделі. [9].

Вид або подання – ця частина відповідає за виведення інформації на екран – це вже дизайнерська частина нашого веб – додатки з мінімальною кількістю логіки. Тобто попросту кажучи, цей блок відповідає за зовнішній вигляд нашого застосування. Завдання уявлення зберігати дизайн скрипта.

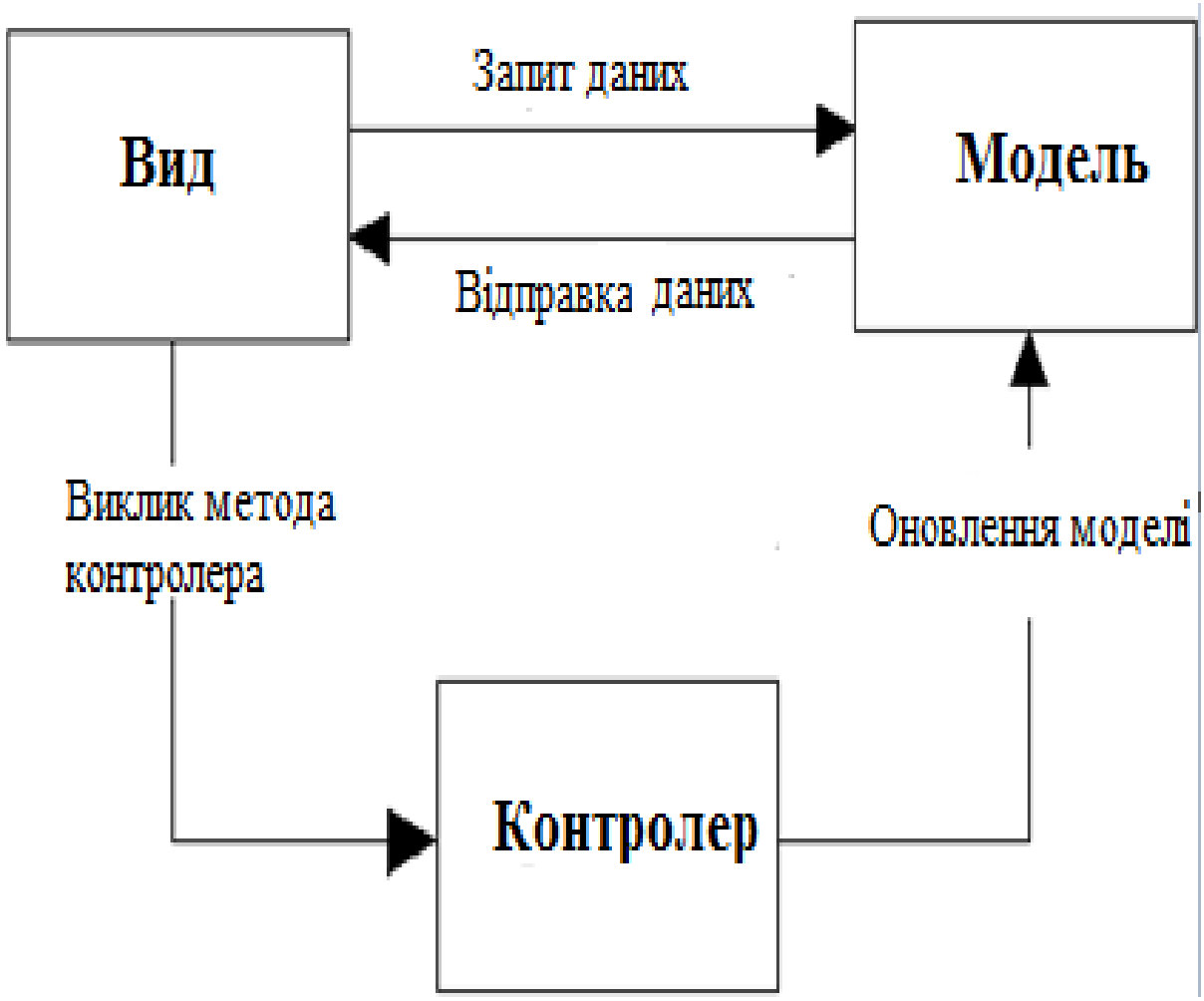

Рисунок 3.7– Архітектура MVC

3.9 Відправка файлів FILEZILLA

FTP клієнт є програмою, яка робить спробу з'єднається з серверним комп'ютером, як правило до порту номер 21. Після успішного підключення до FTP-сервера, можна здійснювати різноманітні операції над даними розташовуються на ньому, зокрема, переглянути вміст каталогів, завантажувати і викачувати файли з FTP сервера, перейменовувати, призначати права доступу, видаляти файли з сервера та інше.

З'єднуючись з FTP сервером допустимо пройти авторизацію надаючи дані для входу без використання шифрування, а також можна підключитися анонімно, якщо це дозволяє FTP сервер. Для захищеної передачі, яка приховує (шифрує) логін, пароль і дані, FTP з'єднання може бути зашифровано за допомогою SSL / TLS (FTPS). SSH протокол передачі файлів (SFTP) іноді також використовується, але між ним і FTPS є істотна різниця.

FTP є СКОРОЧЕННЯ від англ. File Transfer Protocol - протокол передачі файлів, Який застосовується для обміну файлами по TCP / IP мереж между двома комп'ютерами (клієнт и сервер). Протоколу передачі файлів более 40 років, ВІН БУВ розроблення Перш чем з'явився TCP / IP и тім более HTTP, проти ВІН досі актуальний и вікорістовується для Підключення до віддаленіх серверів и обміну файлами.

Даний протокол застосовує різні мережеві з'єднання для передачі команд и файлів между Клієнтом и сервером. FTP сервер, представляет собою комп'ютер з встановленим на него спеціальнім Програмне забезпечення и очікують зовнішнього Підключення від других комп'ютерів.

Основне призначення FTP протоколу - це завантаження файлів і їх скачування з віддаленого сервера. Для передачі файлів в пасивному режимі ініціюється з'єднання FTP клієнтом з обумовленого діапазону портів до порту сервера. В активному режимі FTP сервер підключається до клієнта з порту 20 до певного порту, який повідомив йому клієнт. Основна відмінність між даними режимами, полягає в тому, з якого боку відкривається з'єднання для передачі файлів.

Для ефективної роботи з FTP ми рекомендуємо скористатися FTP менеджером FileZilla. Програма відмінно підійде як для новачків, так і для досвідчених користувачів. Легкість освоєння, кроссплатформенность, безліч підтримуваних мов, велика кількість налаштувань і можливостей роблять її однією з кращих FTP клієнтів.

FTP є скороченням від англ. File Transfer Protocol - протокол передачі файлів, який застосовується для обміну файлами по TCP / IP мереж між двома комп'ютерами (клієнт і сервер). Протоколу передачі

файлів більше 40 років, він був розроблений перш ніж з'явився TCP / IP і тим більше HTTP, проте він досі актуальний і використовується для підключення до віддалених серверів і обміну файлами.

Даний протокол застосовує різні мережеві з'єднання для передачі команд і файлів між клієнтом і сервером. FTP сервер, являє собою комп'ютер з встановленим на нього спеціальним програмним забезпеченням і очікують зовнішнього підключення від інших комп'ютерів. [10].

3.10 Використання БД MYSQL

У житті ми часто стикаємося з необхідністю зберігати будь-яку інформацію, а тому часто маємо справу і з базами даних.База даних - це сукупність пов'язаних даних, організованих за певними правилами, що передбачають загальні принципи опису, зберігання і маніпулювання, незалежна від прикладних програм. База даних є інформаційною моделлю предметної області. Звернення до баз даних здійснюється за допомогою системи управління базами даних (СКБД). СУБД забезпечує підтримку створення баз даних, централізованого управління та організації доступу до них різних користувачів.

Отже, ми прийшли до висновку, що зберігати дані незалежно від програм, так, що вони пов'язані між собою і організовані за певними правилами, доцільно. Але питання, як зберігати дані, за якими правилами вони повинні бути організовані, залишилося відкритим. Способів існує безліч (до речі, називаються вони моделями подання або зберігання даних). Найбільш популярні - об'єктна і реляційна моделі даних.

Автором реляційної моделі вважається Е. Кодд, який першим запропонував використовувати для обробки даних апарат теорії множин (об'єднання, перетин, різниця, декартовій твір) і показав, що будь-яке представлення даних зводиться до сукупності двовимірних таблиць особливого виду, відомого в математиці як відношення.

Таким чином, реляційна база даних являє собою набір таблиць (точно таких же, як наведена вище), пов'язаних між собою. Рядок в таблиці відповідає сутності реального світу (в наведеному вище прикладі це інформація про людину).

База даних відіграє важливу роль для кожного сучасного веб-додатки. Завдяки динамічній природі веб-додатків зараз, навіть найпростіші додатки вимагають деяких механізмів зберігання, доступу та

зміни даних (ось чому в Hostinger ми пропонуємо необмежені Бази даних MySQL для наших клієнтів з преміум і бізнес акаунтами). Природно, оскільки важливість баз даних стрімко росте, реляційні системи управління базами даних або реляційні СУБД набирають свою популярність (Relational Database Management Systems - RDBMS)

Двізних MySQL і SQL Server. Обидві виконують однакову функцію, хоча маю різні варіанти використання. Вони розрізняються деякими особливостями, але обидві системи базуються на SQL або Structured Query Language (структурована мова запитів). У зв'язку з цим, розробники можуть виявити кілька схожості між MySQL і SQL сервер, таких як використання таблиць для збереження даних, посилання на первинні та зовнішні ключі, також як кілька баз даних в одному середовищі або на одному сервері.

Не буде помилкою сказати, що MySQL і SQL сервер - це дві найбільш популярні реляційні СУБД серед існуючих, хоча Oracle і Postgres знайдеться, що сказати з цього приводу. Не дивлячись на те, що ми поступово стаємо свідками переходу з SQL на NoSQL, перші все ж продовжують домінувати. Це означає, що зараз все ще актуально вивчити як MySQL, так і SQL сервер.

У цьому керівництві ми докладно роз'яснимо, що таке MySQL і SQL сервер. Ми знайдемо відмінності між MySQL і SQL сервером і допоможемо вам вибрати найбільш підходящу для ваших потреб.

Уявіть собі найпростішу таблицю імен, номерів телефонів, адрес і т.д. Саме так і зберігаються дані реляційних БД - в таблиці, організовуються посредствам стовпців і рядків. Кожному стовпцю присвоєно ім'я, яке відображається в назві, всі значення в цьому стовпці належать до змінних тільки одного типу. Стовпці розташовані в певному строгому порядку, в той час як рядки невпорядковані. Найчастіше дані деяких осередків в одній таблиці пов'язані зі значенням осередків іншої таблиці і так далі. Запити до БД повертають результат у вигляді таблиці.

Дані в БД діляться на унікальні або неунікальні. Неунікальні - це ім'я, рік народження, час і т.д., в той час, як унікальні - номер кредитки, договору хостинг-послуг. Унікальні значення присутні в списках так званого «унікального індексу»

Великою перевагою MySQL є можливість роботи з інтерфейсом програмного додатка API (Application Program Interface). API може забезпечити простий доступ з програми користувача до СУБД. Нехай навіть ці програми будуть написані на на Perl, C і т.д.

Найпопулярнішою «зв'язкою» для управління сайтами вважається MySQL з мовою PHP. Багато CMS написані на PHP в зв'язці з БД MySQL. Одним з найяскравіших прикладів цього «союзу» може служити движок для сайтів і блогів WordPress, який завоював величезну популярність у світі. Взаємодія з MySQL в даному випадку ведеться за допомогою сукупності функцій. Прикладом такої функції може служити «mysql_connect», яка з'єднується з сервером БД і повертає дескриптор з'єднання з нею.

Існує безліч СУБД підтримують SQL мова запитів: MySQL, mSQL, PostgreSQL, MSSQL і багато інших. Кожна з них має переваги в певній сфері. І все ж саме MySQL завоювала широке визнання і популярність в Інтернеті завдяки своїй гнучкості та універсальності.

Сервер баз даних MySQL - дуже швидкий, надійний і простий в експлуатаціїсервер.

Якщо це якраз те, що ви шукаєте, варто з ним попрацювати. Сервер MySQL включає в себе практичний набір засобів, розроблених в тісній кооперації з спільнотою користувачів. Результати порівняльних тестів продуктивності MySQL та інших СУБД доступні за адресою http://dev.mysql.com/tech-resources/crash-me.php. Спочатку сервер MySQL був розроблений для більш швидкого управління великими базами даних, ніж існуючі рішення вцій галузі, і протягом ряду років успішно експлуатувався в середовищах, до яких пред'являлися досить високі вимоги. Незважаючи на те що MySQL перебуває в безперервному процесі розробки, на сьогоднішній день він надає багатий набір зручних в експлуатації засобів і функцій. Притаманні сервера MySQL можливості мережевої взаємодії, продуктивність і безпеку роблять його вдалим варіантом для роботи з базами даних в Internet.

Сервер MySQL працює в клієнт-серверних і вбудованих системах.СУБД MySQL є клієнт-серверної системою, що включає багато-SQL-сервер, що підтримує різні платформи, кілька клієнтських програм і бібліотек, інструменти адміністрування і широкий діапазон програмних інтерфейсів додатків (API-інтерфейсів).

Сервер MySQL існує також і в формі вбудовується багатопотокової бібліотеки, яку можна пов'язувати з розробляються додатками, щоб отримати більш компактні, швидкі і легкокеровані продукти.

Завдання тривалого зберігання і обробки інформації з'явилася практично відразу з появою перших комп'ютерів. Для вирішення цього завдання в кінці 60-х років були розроблені спеціалізовані програми, що отримали назву систем управління базами даних (СКБД). СУБД виконали

тривалий шлях еволюції від системи управління файлами, через ієрархічні і мережні бази даних. В кінці 80-х років домінуючою стала система керування базами даних (СУРБД). З цього часу такі СУБД стали стандартом де-факто, і для того, щоб уніфікувати роботу з ними, був розроблений структурована мова запитів (SQL), який представляє собою мову управління саме реляційними базами даних.

Взаємодія з базою даних відбувається за допомогою Системи Управління Базою Даних (СУБД), яка розшифровує запити і проводить операції з інформацією в базі даних. Тому більш правильно було б говорити про запит до СУБД і про взаємодію з СУБД з Web-додатки. Але так як це дещо ускладнює сприйняття, далі скрізь ми будемо говорити "база даних", маючи на увазі при цьому СУБД.

Ієрархічна база даних заснована на структурі дерева зберігання інформації. У цьому сенсі ієрархічні бази даних дуже нагадують файлову систему комп'ютера.

У реляційних базах даних дані зібрані в таблиці, які в свою чергу складаються з стовпців і рядків, на перетині яких розташовані осередки. Запити до таких баз даних повертає таблицю, яка повторно може брати участь в наступному запиті. Дані в одних таблицях, як правило, пов'язані з даними інших таблиць, звідки і пішла назва "реляційні".

В об'єктно-орієнтованих базах даних дані зберігаються у вигляді об'єктів. З об'єктно-орієнтованими базами даних зручно працювати, застосовуючи об'єктно-орієнтоване програмування. Однак, на сьогоднішній день такі бази дан-них ще не досягли популярності реляційних, оскільки поки значно поступаються їм в продуктивності.

Гібридні СУБД поєднують в собі можливості реляційних і об'єктно-орієнтованих баз даних. [11].

3.11 Вибір середи розробки для серверної частини додатку

Для розробки на мові PHPє безліч різних інструментів такі як TextSublime,PHPDesigner,Netbeans і навіть стандартний блокнот,але для полегшення виконання задачі,я вибрав найпопулярнішу та єфективну середу розробки PHPStorm. Мені здається вона найбільше всього підходить для реалізації моєї задачі.

Реалізований в PhpStorm графічний PHP-відладчик підтримує умовні точки зупину, відстеження значень і автоматизований вхід в налагодження окремих процедур. Для тестування додатків підтримується каркас тестових модулів PhpUnit і графічний інтерфейс для запуску тестів. При редагуванні

коду виділяються конструкції синтаксису, здійснюється розширене форматування конфігурації, виявлення помилок в режимі реального часу і завершення коду. PhpStorm-редактор враховує коментарі до коду при його завершенні, автоматично вибираючи оптимальне рішення проблеми. PHP-рефакторинг і редагування шаблонів гарантує зміна проекту в найкоротші терміни. PhpStorm дозволяє візуалізувати код в ієрархічності вигляді і забезпечує швидку навігацію по всіх елементах.

Завдяки застосуванню PHPUnit-тестів можна швидко переглядати результати генерації коду окремих блоків або всього програми. Якщо тест був проведений невдало, продукт дозволяє переглядати окремі кодові рядки, в яких балу виявлена помилка. PhpStorm забезпечує налагодження коду JavaScript і надає широкий діапазон можливостей: знаходження точки зупину в HTML і JavaScript, настройка параметрів точки зупину, тестування синтаксису коду в режимі реального часу і т. Д.

Сотні інспекцій піклуються про верифікації коду, аналізуючи проект цілком під час розробки. Підтримка PHPDoc, code (re) arranger, форматування коду з конфігурацією стилю коду і інші можливості допомагають розробникам писати охайний і легко-підтримуваний код.

Підтримуються передові технології веб-розробки, включаючи HTML5, CSS, Sass, SCSS, Less, Stylus, Compass, CoffeeScript, TypeScript, ECMAScript Harmony, шаблони Jade, Zen Coding, Emmet, і, звичайно ж, JavaScript.

PhpStorm включає в себе всю функціональність WebStorm (HTML / CSS редактор, JavaScript редактор) і додає повнофункціональну підтримку PHP і баз даних / SQL (рис.3.8).

Перевагами данної середи розробкиє:

– Автодоповнення коду фіналізують класи, методи, імена змінних, ключові слова PHP, а також широко використовуються імена полів і змінних в залежності від їх типу. Ця функція значно полегшує розробку додатку, навіть якщо користувач є досвідченим розробником;

– MVC подання для Symfony2 і Yii фреймворків;

– детектор дубльованого коду;

– підтримує PHP 7.0, 5.6, 5.5, 5.4 і 5.3, генератори, співпрограми і все синтаксичні поліпшення;

–інструменти роботи з базами даних, SQL редактор;

– HTML, CSS, JavaScript редактор. Налагодження і модульне тестування для JS. Підтримка HTML5, CSS, Sass, SCSS, Less, Stylus, Compass, CoffeeScript, TypeScript, ECMAScript Harmony, Emmet і інших передових технологій веб-розробки.[12].

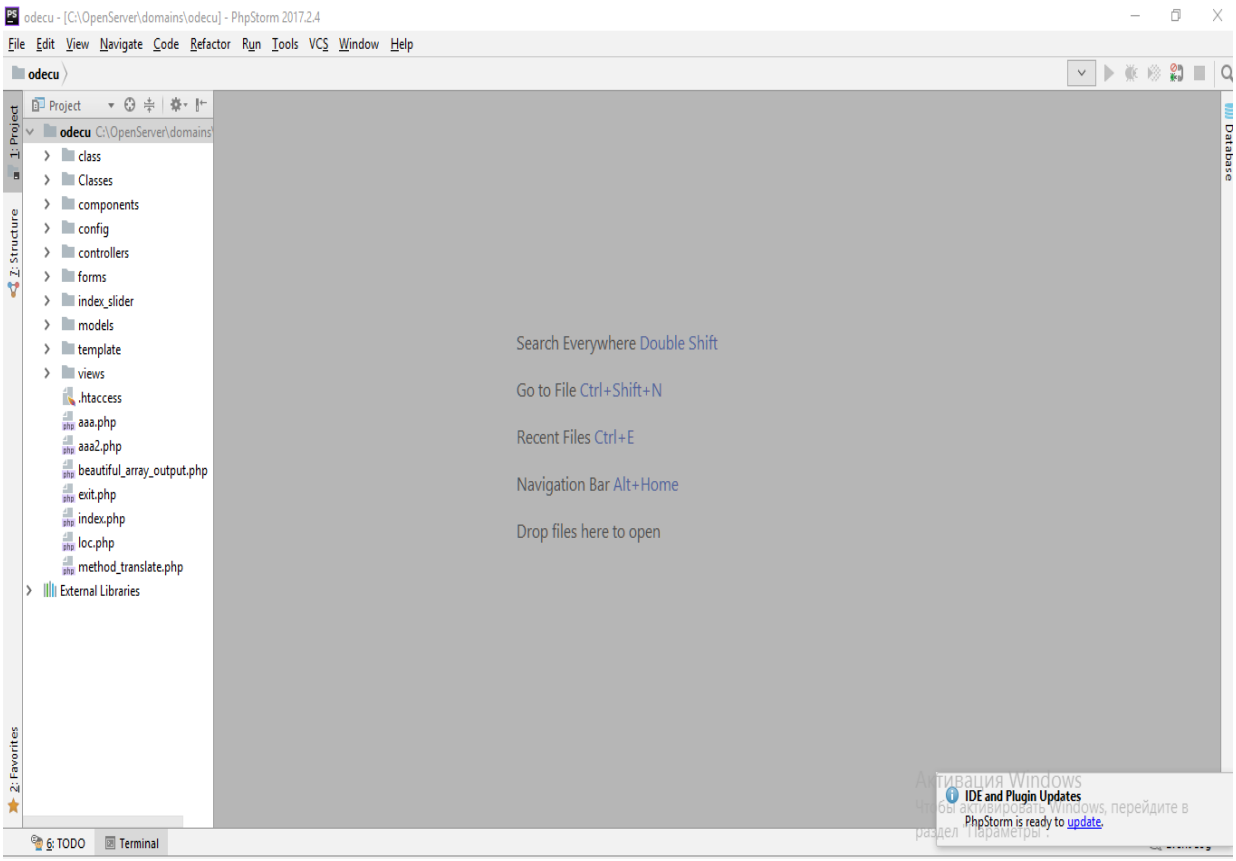

Рисунок 3.8 – Інтерфейс середовища розробки PHPStorm

4 ПРОЕКТУВАННЯ ІНФОРМАЦІОННОЇ СИСТЕМИ

4.1 Архітектурасистеми

Архітектура програмного забезпечення системи або набору систем складається з усіх важливих проектних рішень з приводу структур програми і взаємодій між цими структурами, які складають системи. Проектні рішення забезпечують бажаний набір властивостей, які повинна підтримувати система, щоб бути успішною. Проектні рішення надають концептуальну основу для розробки системи, її підтримки та обслуговування.

Архітектура програмної системи охоплює не тільки її структурні і поведінкові аспекти, а й правила її використання та інтеграції з іншими системами, функціональність, продуктивність, гнучкість, надійність, можливість повторного застосування, повноту, економічні та технологічні обмеження, а також питання призначеного для користувача інтерфейсу.

У міру розвитку програмних систем все більшого значення набуває їх інтеграція один з одним з метою побудови єдиного інформаційного простору підприємства. Як можна бачити з вищенаведених визначень інтеграція є найважливішим елементом архітектури.

Для того щоб побудувати правильну і надійну архітектуру і грамотно спроектувати інтеграцію програмних систем необхідно чітко слідувати сучасним стандартам в цих областях. Без цього велика ймовірність створити архітектуру, яка нездатна розвиватися і задовольняти зростаючі потреби користувачів IT. Як законодавців стандартів в цій галузі виступають такі міжнародні організації як SEI (Software Engineering Institute), WWW (консорціум World Wide Web), OMG (Object Management Group), організація розробників Java - JCP (Java Community Process), IEEE (Institute of Electrical and Electronics Engineers) та інші.

Хоча визначення дещо відрізняються, можна помітити чималу ступінь подібності. Наприклад, більшість визначень вказують на те, що архітектура пов'язана зі структурою і поведінкою, а також тільки зі значущими рішеннями, може відповідати деякому архітектурному стилю, на неї впливають зацікавлені в ній особи і її оточення, вона втілює рішення на основі логічного обгрунтування.

Клієнт-сервер (англ. Client-server) - мережева архітектура, в якій пристрої є або клієнтами, або серверами. Клієнтом (front end) є запитуюча машина (зазвичай ПК), сервером (back end) - машина, яка відповідає на запит. Обидва терміни (клієнт і сервер) можуть застосовуватися як до фізичних пристроїв, так і до програмного забезпечення.

Як зрозуміло з назви, в даній концепції беруть участь дві сторони: клієнт і сервер. Тут все як у житті: клієнт - це замовник тієї чи іншої послуги, а сервер - постачальник послуг. Клієнт і сервер фізично представляють собою програми, наприклад, типовим клієнтом є браузер. Як сервер можна навести такі приклади: всі HTTP сервера (зокрема Apache), MySQL сервер, локальний веб-сервер або готова збірка Denwer (останніх два приклади - це не проста сервера, а цілий набір серверів).

Основний принцип технології "клієнт-сервер" полягає в поділі функцій додатка на три групи:

– введення і відображення даних (взаємодія з користувачем);

– прикладні функції, характерні для даної предметної області;

– функції управління ресурсами (файловою системою, базою даних і т.д.).

Тому, в будь-якому додатку виділяються наступні компоненти:

– компонент представлення даних;

– прикладної компонент;

– компонент управління ресурсом.

Зв'язок між компонентами здійснюється за певними правилами, які називають "протокол взаємодії".

Сервер - це сама СУБД. Він підтримує всі основні функції СУБД: визначення даних, обробку даних, захист даних, підтримка цілісності даних і т.д.

Клієнт – це додаток користувача. Для отримання даних клієнт формує і відсилає запит віддаленого сервера, на якому розміщена БД. Запит формується на мові SQL, який є стандартним засобом доступу до сервера при використанні реляційних моделей даних. Після отримання запиту віддалений сервер направляє його SQL-сервера (сервера баз даних) - спеціальною програмою, що управляє віддаленої БД і забезпечує виконання запиту і видачу його результатів клієнту. [14].

Клієнт і сервер взаємодію один з одним в мережі Інтернет або в будь-який інший комп'ютерної мережі за допомогою різних мережевих протоколів, наприклад, IP протокол, HTTP протокол, FTP та інші. Протоколів насправді дуже багато і кожен протокол дозволяє надавати ту чи іншу послугу. Наприклад, за допомогою HTTP протоколу браузер

відправляє спеціальне HTTP повідомлення, в якому зазначено яку інформацію і в якому вигляді він хоче отримати від сервера, сервер, отримавши таке повідомлення, відсилає браузеру у відповідь схоже по структурі повідомлення (або кілька повідомлень), в якому міститься потрібна інформація, як правило це HTML документ.

Переваги:

– відсутність дублювання коду програми-сервера програмами-клієнтами;

– так як всі обчислення виконуються на сервері, то вимоги до комп'ютерів, на яких встановлено клієнт, знижуються;

– всі дані зберігаються на сервері, який, як правило, захищений набагато краще за більшість клієнтів. На сервері простіше організувати контроль повноважень, щоб вирішувати доступ до даних тільки клієнтам з відповідними правами доступу;

Недоліки:

– непрацездатність сервера може зробити непрацездатною всю обчислювальну мережу. Непрацездатним сервером слід вважати сервер, продуктивності якого не вистачає на обслуговування всіх клієнтів, а також сервер, що знаходиться на ремонті, профілактиці і т. П;

– підтримка роботи даної системи вимагає окремого фахівця - системного адміністратора;

– висока вартість обладнання.

Вартість серверного обладнання значно вище клієнтського. Сервер повинен обслуговувати спеціально навчений і підготовлений людина. Якщо в локальній мережі лягає сервер, то і клієнти не зможуть працювати (в якості окремого випадку можна привести приклад: потужності сервера не завжди вистачає, щоб задовольнить запити клієнтів, якщо ви хоч раз працювали з білінговими системами, то розумієте про що я: час очікування відповіді від сервера може бути дуже великим).

Повідомлення, які посилають клієнти отримали назви HTTP запити. Запити мають спеціальні методи, які говорять сервера про те, як обробляти повідомлення. А повідомлення, які посилає сервер отримали назву HTTP відповіді, вони містять крім корисної інформації ще й спеціальні коди стану, які дозволяють браузеру дізнатися те, як сервер зрозумів його запит.

Зараз ми схематично описали, як взаємодіють клієнт і сервер на сьомому рівні моделі OSI, але, насправді це взаємодія відбувається на всіх семи рівнях. Коли клієнт відправляє запит, повідомлення упаковується, можна уявити, що повідомлення загортається в сім обгорток (хоча їх може

бути набагато більше або ж менше), а коли повідомлення отримує сервер, він починає ці обгортки розгортати.

Також варто зауважити, що в основі взаємодії клієнт-сервер лежить принцип того, що така взаємодія починає клієнт, сервер лише відповідає клієнту і повідомляє про те чи може він надати послугу клієнтові і якщо може, то на яких умовах. Клієнтське програмне забезпечення та серверне програмне забезпечення зазвичай встановлено на різних машинах, але також вони можуть працювати і на одному комп'ютері.

Дана концепція взаємодії була розроблена в першу чергу для того, щоб розділити навантаження між учасниками процесу обміну інформацією, а також для того, щоб розділити програмний код постачальника і замовника. Нижче ви можете побачити спрощену схему взаємодії клієнт-сервер (рис. 4.1).

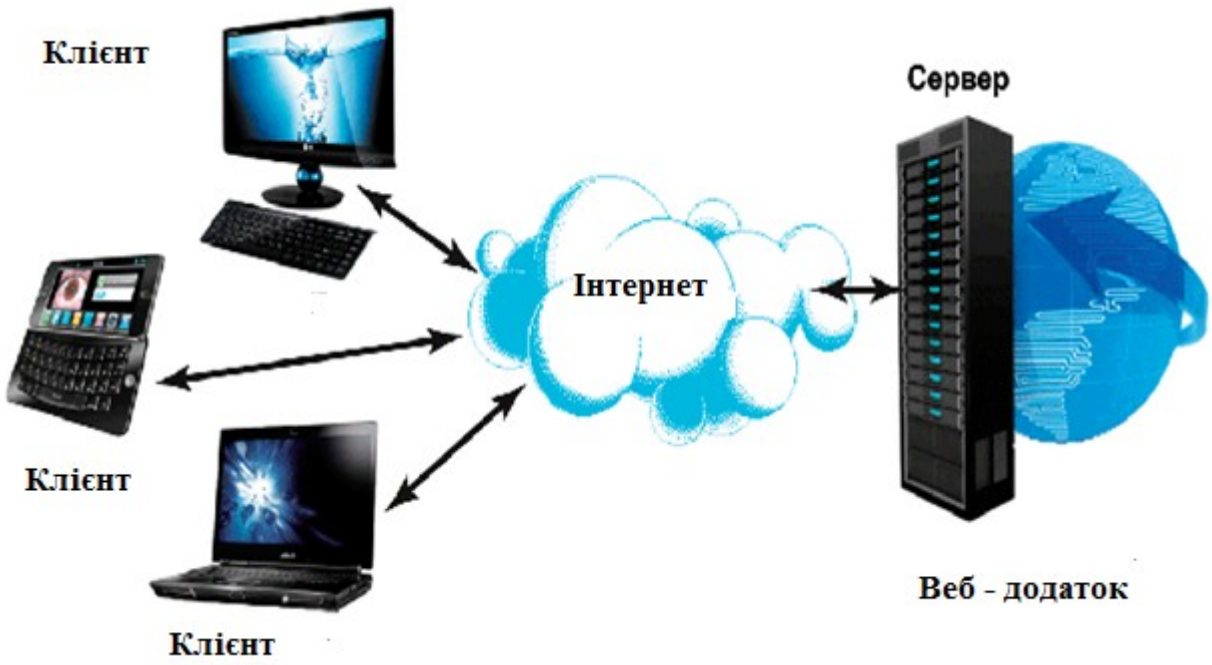

Рисунок 4.1 – Принцип роботи клієнт-серверного додатку

Клієнт – це інтерфейсний (зазвичай графічний) компонент, який представляє перший рівень, власне додаток для кінцевого користувача. Перший рівень не повинен мати прямих зв'язків з базою даних (за вимогами безпеки), бути навантаженим основний бізнес-логікою (за вимогами масштабованості) і зберігати стан додатки (за вимогами надійності).

На перший рівень може бути винесена і зазвичай виноситься найпростіша бізнес-логіка: інтерфейс авторизації, алгоритми шифрування,

перевірка вводятьсязначень на допустимість і відповідність формату, нескладні операції (сортування, угруповання, підрахунок значень) з даними, вже завантаженими на термінал.

Сервер додатків розташовується на другому рівні. На другому рівні зосереджена велика частина бізнес-логіки. Поза його залишаються фрагменти, що експортуються на термінали (див.вище), а також занурені в третій рівень збережені процедури і тригери.

Сервер бази даних забезпечує зберігання даних і виноситься на третій рівень. Зазвичай це стандартна реляційна або об'єктно-орієнтована СУБД. Якщо третій рівень являє собою базу даних разом з збереженими процедурами, тригерами та схемою, яка описує додаток в термінах реляційної моделі, то другий рівень будується як програмний інтерфейс, що зв'язує клієнтські компоненти з прикладної логікою бази даних.

Багаторівневі архітектури клієнт-серверархітектури більш розумно розподіляють модулі обробки даних, які в цьому випадку виконуються на одному або декількох окремих серверах. Ці програмні модулі виконують функції сервера для інтерфейсів з користувачами і клієнта - для серверів баз даних. Крім того, різні сервери додатків можуть взаємодіяти між собою для більш точного поділу системи на функціональні блоки, які виконують певні ролі. Наприклад, можна виділити сервер управління персоналом, який буде виконувати всі необхідні для управління персоналом функції. Зв'язавши з ним окрему базу даних, можна приховати від користувачів всі деталі реалізації цього сервера, дозволивши їм звертатися тільки до його загальнодоступним функцій. Крім того, таку систему дуже просто адаптувати до Web, оскільки простіше розробити html-форми для доступу користувачів до певних функцій бази даних, ніж до всіх даних.

У трирівневої архітектурі "тонкий" клієнт не перевантажений функціями обробки даних, а виконує свою основну роль системи подання інформації, що надходить з сервера додатків. Такий інтерфейс можна реалізувати за допомогою стандартних засобів Web-технології - браузера, CGI і Java. Це зменшує обсяг даних, переданих між клієнтом і сервером додатків, що дозволяє підключати клієнтські комп'ютери навіть по повільним лініях типу телефонних каналів. Крім того, клієнтська частина може бути настільки простий, що в більшості випадків її реалізують за допомогою універсального браузера. Але якщо міняти її все-таки доведеться, то цю процедуру можна здійснити швидко і безболісно. Трирівнева архітектура клієнт-сервер дозволяє більш точно призначати повноваження користувачів, так як вони отримують права доступу не до самої бази даних, а до певних функцій сервера додатків. Це підвищує

захищеність системи (у порівнянні з звичайно архітектурою) не тільки від навмисного нападу, а й від помилкових дій персоналу.

Багаторівневі клієнт-серверні системи досить легко можна перевести на Web-технологію – для цього достатньо замінити клієнтську частину універсальним або спеціалізованим браузером, а сервер додатків доповнити Web-сервером і невеликими програмами виклику процедур сервера. Для розробки цих програм можна використовувати як Common Gateway Interface (CGI), так і більш сучасну технологію Java.

Слід зазначити і той факт, що в трирівневої системи по каналу зв'язку між сервером додатків і базою даних передається досить багато інформації. Однак це не уповільнює обчислень, так як для зв'язку зазначених елементів можна використовувати більш швидкісні лінії. Це зажадає мінімальних витрат, оскільки обидва сервера зазвичай знаходяться в одному приміщенні. Таким чином, збільшується сумарна продуктивність системи - над одним завданням тепер працюють два різних сервера, а зв'язок між ними можна здійснювати за найбільш швидкісним лініях з мінімальними витратами коштів. Правда, виникає проблема узгодженості спільних обчислень, яку покликані вирішувати менеджери транзакцій - нові елементи багаторівневих систем.

У порівнянні з двухзвенной клієнт-серверної архітектурою або файл-серверної архітектурою трирівнева архітектура забезпечує, як правило, більшу масштабованість (за рахунок горизонтальної масштабованості сервера додатків і мультиплексування з'єднань), велику конфігурованість (за рахунок ізольованості рівнів один від одного). Реалізація програм, доступних з веб-браузера або з тонкого клієнта, як правило, має на увазі розгортання програмного комплексу в трирівневої архітектурі. При цьому зазвичай розробка трехзвенной програмних комплексів складніше, ніж для двухзвенних, також наявність додаткового сполучного програмного забезпечення може накладати додаткові витрати в адмініструванні таких комплексів. [15].

4.2Розробка серверної частини

Мова PHP підтримує поліморфізм в тому сенсі, що дозволяє використовувати замість примірників батьківського класу екземпляри підкласу. Введення в дію необхідного методу здійснюється на етапі прогону. Підтримка перевантаження методів, при якій введення методу в дію здійснюється з урахуванням сигнатури методу, відсутня. Справа в

тому, що в кожному класі може бути присутнім тільки один метод з певним ім'ям. Але завдяки тому, що в мові PHP застосовується слабка типізація і підтримується змінну кількість параметрів, з'являється можливість обійти це обмеження. Якщо вам цікаво, можу поділитися знаннями.Поліморфізм (многоформенность) є наслідком ідеї наслідування. У загальних словах, полиморфность класу - це властивість базового класу використовувати функції похідних класів, навіть якщо на момент визначення ще невідомо, який саме клас буде включати його в якості базового і, тим самим, ставати від нього похідним.оліморфізм - довге слово для дуже простий концепції.

Поліморфізм описує шаблон в об'єктно орієнтованому програмуванні, в якому класи мають різну функціональність при використанні загального інтерфейсу.Принадність поліморфізму полягає в тому, що можна працювати в коді з різними класами, і при цьому не потрібно знати, що за клас використовується, тому що вони мають один і той же інтерфейс. Аналогія поліморфізму в реальному світі - кнопка. Кожен знає як використовувати кнопку - потрібно просто натиснути на неї. Але те, що "робить" кнопка насправді, залежить від її з'єднань і контексту використання. Якщо хтось каже, що потрібно натиснути кнопку, то вже відомо, що потрібно зробити, щоб вирішити задачу.

У світі програмування поліморфізм використовується для створення модульних структур додатки і спрощення процедури розширення функціоналу. Замість того, щоб використовувати мішанину умовних виразів, що описують різні варіанти дій, можна створити взаємозамінні об'єкти, які будуть вибиратися в залежності від умов використання. Ось в чому полягає основна мета використання поліморфізму.

Папка Classesмістить у собі класи та бібліотеки які завантажуються із інтернету,а потім підключаються до всього проекту. В нашому випадку – це бібліотека PHPExcel.

Папка componentsвміщує два класи Router та DB.Перший із них є головним классом у всьому додатку з якого й починається завантаження серверної частини. Задача класу Db – це пдіключеня до бази даних.

Каталог configяк ви вже зрозуміли вміщує всі налаштування додатку,а саме паролі та логіни адмін панелі,бази данних.Також в ньому знаходятся всі шляхи посилання.

Дерикторія template вміщує в собі cssстилі та картинки які потрібні для оформлення сайту і надання йому більш барвистого виду.

Каталог Controllerмістить контроллери які потрібні для звязку між системою та користувачем.Контролює і направляє дані від користувача до

системи і навпаки. Використовує модель і уявлення для реалізації необхідного дії.

Папка modelsмає моделі які потрібні для надання даних і методів роботи з ними:

– запити до бази даних;

–перевірка на коректність.

Модель не залежить від уявлення (не знає як дані візуалізувати) і контролера (не має точок взаємодії з користувачем) просто надаючи доступ до даних і управління ними.

Модель будується таким чином, щоб відповідати на запити, змінюючи свій стан, при цьому може бути вбудовано повідомлення «спостерігачів».

Модель, за рахунок незалежності від візуального представлення, може мати кілька різних уявлень для однієї «моделі».

При розробці серверної частини я використав поліморфізм,де створювались класи-контроллери наприклад (ContactsController) з двома однаковими методами.ActionIndexпризначений для представлення всіх данних у вігляді списку або таблиці.При виклику методаactionView(id) та завдяки заданним йому параметрам,він повертає в результаті дані які мають однаковий idіз параметром. В нашому випадку створено три контролери TimetableController, TeacherController, ContactsController, як знаходяться в папці controllers.Ці перераховані класи ініціалізуються в класі Router.php, який зчитує данні з адресного рядка і підключає потрібний нам контролер. В кожному з них реалізується потрібний обєкт класа моделі. Всі моделі розміщені в каталозі models і виконують підключення та запроси до потрібної бази даних. Також в них виконується перевірка вхідних параметрів та сесій на правильніть,які отримуються з форм. Сама серверна частина закладається з панелі адміністратора,яка дозволяє вносити та редагувати інформацію. Також є сторінка для пошуку інформації вчителя,предметів,факультетів та груп. Але основна задача серверної частини додатку,це отриманння та повна обробка данних яка надходить із клієнта(Додаток).В подальшому повертає результат клієнту вв форматі JSON. [16].

4.3База даних для додатку

Для того щоб десь зберігати інформацію, можна використовувати файли різних форматів,але при подальшому користуванні буде не зручно

витягувати потрібні нам дані. Саме для того,щоб не виникало таких незручностей існують Бази даних.Вони створювались у веб-додатку для адміністрування СУБД MYSQL, який називається phpMyAdmin.Переважна кількість провайдерів використовують цю програму в якості панелі управління для того, щоб надати своїм клієнтам можливість адміністрування виділених їм баз даних. За допомогою нього набагато легше керувати базами даних. Для нашого додатку ми створимо три таблиці (рис. 4.2).

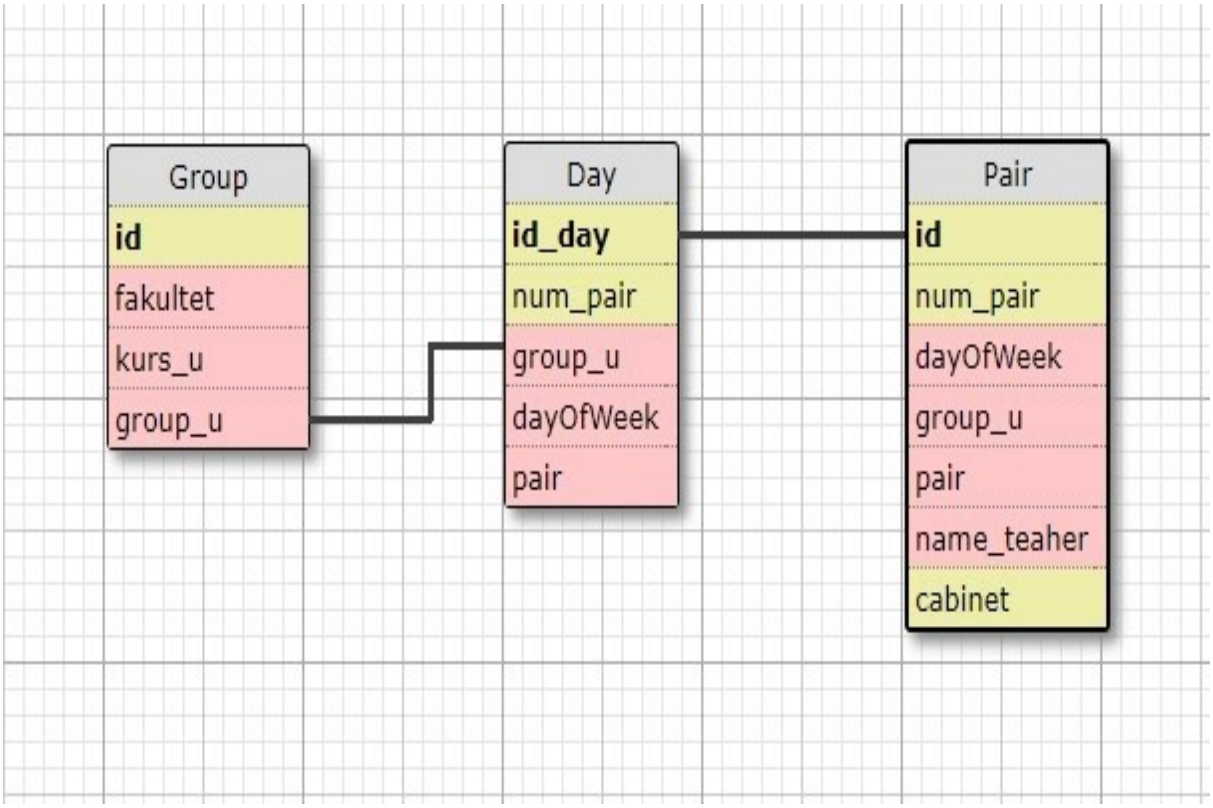

Рисунок 4.2 – Таблиці в Базі даних

Створюємо першу таблицю Група,яка буде вміщувати чотири поляid, fakultet, kurs_u, group_u. Вона буде зв’язана с таблицею День по полю group_u.Сама таблицяDaуматиме п’ять полів. Ключове поле id_day,номер пари num_pair, назва групи group_u,день тижня dayOfWeekта назва пари Pair. [17].

4.4Створення панелі адміністратора

Адмін це «серце» сайту або блогу. З адмін-панелікористувач робить всі необхідні дії: налаштовує дизайн сайту,редагує,створює та видаляє

розклад. Такожзмнюєрізні категорії з інформацією про викладачів. Загалом, в адмін-панелі укладені всі налаштування і елементи, які забезпечують стабільну роботу сайту, а також його оновлення. Цеприватна, невидима для відвідувачів динамічного сайту частина ресурсу, що дозволяє редагувати вміст сайту (текстів і картинок), а також додавати або видаляти розділи та вносити деякі зміни в дизайн сайту. Робота з адмінкой не вимагає спеціальних професійних навичок.Існують і сайти без можливості редагування. Вартість їх до 50% нижче, але якщо захочеться внести зміни, користувач без знань HTML змушений буде звернутися в веб-студію.Для доступу до адмін-панелі користувач повинен ввести відповідний логін та пароль(рис. 4.3).

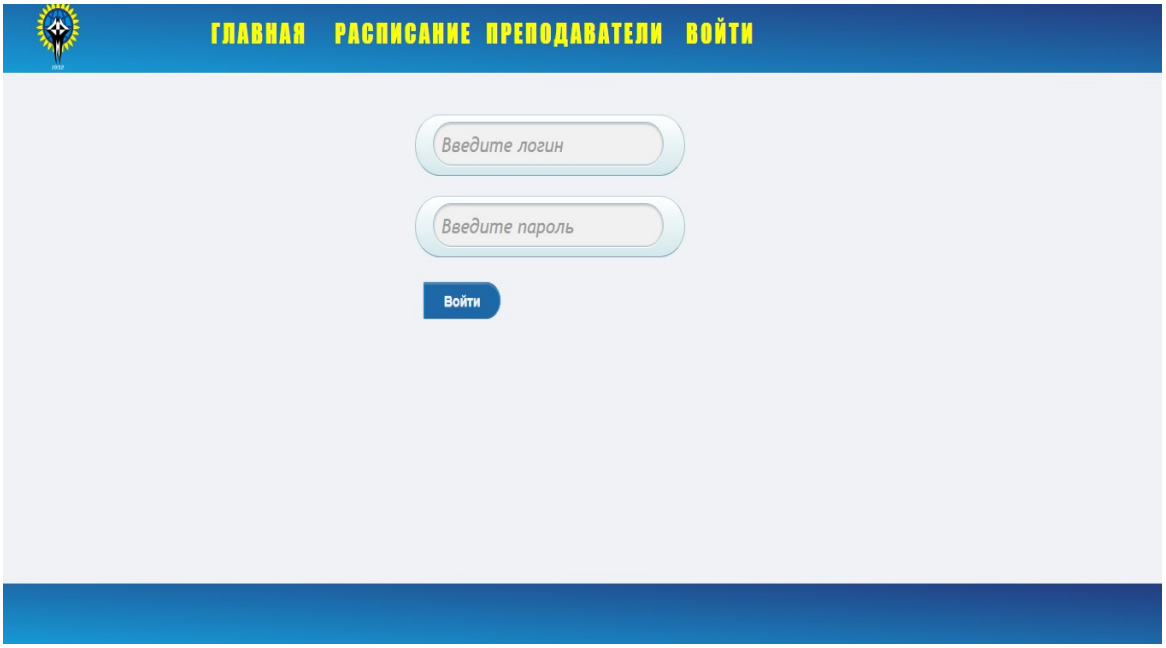

Рисунок 4.3–Сторінка доступу до адмін-панелі

В разі успішної авторизації,користувач потрапляє на сторінку де надається право змінювати, створювати,а також видаляти інформацію про викладача ,або ж розклад(рис.4.4). [18].

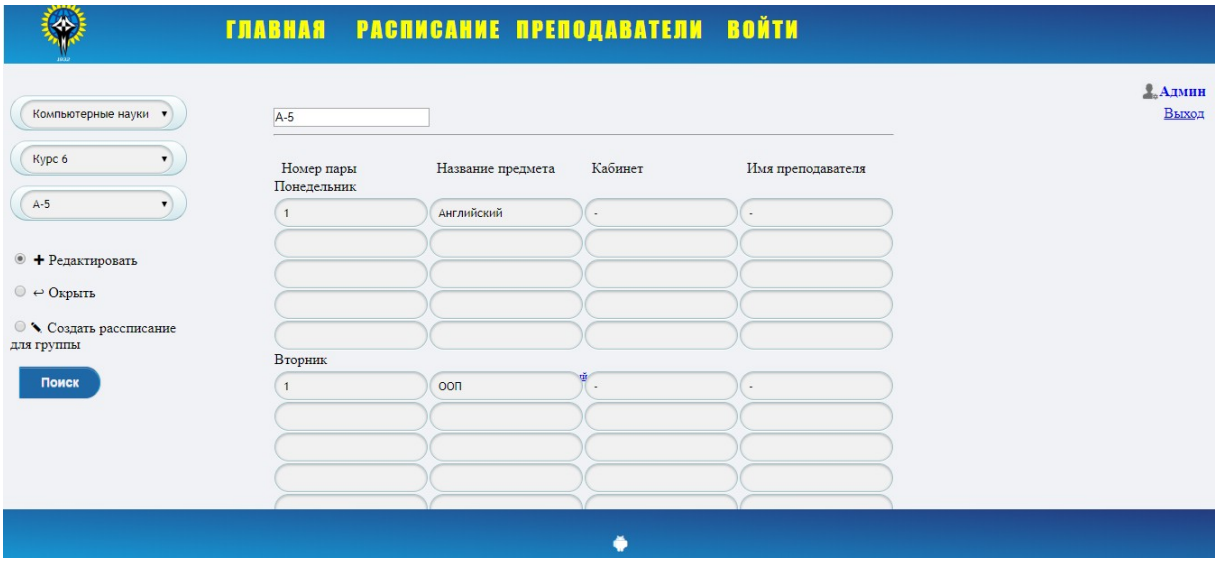

Рисунок 4.4– Сторінка редагування та створення розкладу

4.5Відображення інформації у файлах MSExcel

Часто буває, що база даних має просту структуру і невеликий розмір. Як приклад можна привести список викладачів невеликого вузла, список предметі та розкладу. Всі ці речі можна реалізувати за допомогою простих форм баз даних, для зберігання яких цілком підійдуть звичайні текстові файли. Тому давайте розглянемо всі плюси і мінуси технології, описаної в даному розділі.

Почнемо з хорошого. Використання текстових файлів для зберігання баз даних має кілька незаперечних переваг перед більш складними альтернативами, такими як DBM-файли або системи управління великими базами даних типу Oracle і Sybase. Нижче я перерахував деякі з основних переваг текстових файлів.

Бази даних, що зберігаються в текстових файлах, є стерпним. Їх можна без всяких проблем використовувати практично на будь-який комп'ютерної платформі.

Текстові файли можна редагувати за допомогою звичайного текстового редактора, а також роздрукувати на папері без залучення будь-яких спеціальних засобів.

Текстові файли баз даних дуже просто створювати, а також вносити в них початкові дані.Текстові файли баз даних можуть бути легко імпортовано в програми електронних таблиці, текстові процесори або СУБД. Практично всі відомі програми можуть імпортувати дані, що зберігаються в текстових файлах(рис. 4.5).

Безсумнівно великим плюсом є те що файл з данними можна скачати,або передати на сервер за допомогою протокола FTP,що даэ змогу редагувати та створювати файл без зєднання через інтернет.

Думаю у багатьох виникала необхідність зберігати файли, пов'язані із записом в таблиці. Це може бути картинка до новини, аватар, завантажений користувачем файл - так все, що завгодно. Зазвичай в цьому випадку надходять просто - файл лягає в файлову систему, а посилання на нього - в запис БД.

Але у такого класичного походу безліч недоліків:

–файли не видаляються при видаленні відповідного запису БД;

–проблеми при одночасній спробі оновлення файлу;

– порушення синхронізації між БД і файлової системою при відкат транзакції

– при резервного копіювання та відновлення інформації в БД може виникнути рассинхронизация з файлової системою;

– файли не підкоряються обмеженням доступу, накладеним за допомогою БД.

Саме тому крім збереження інформації в базі даних, було прийнято підключити до серверної частини додатку, бібліотеку PHPExcel, якака може створювати файли з розширенням “xlsx”і дозволятиме редагувати, свторювати та зберігати інформацію.За допомогою такого рішення можна легко відкривати файли маючи офісну програму MSExcel, та в подальшому переглядати розклад і данні про вчителів.Для цього використовуються файли двух різних типів.Це «преподаватели» та «расписание» яким присвоюються відповідні поля(рис. 4.6).

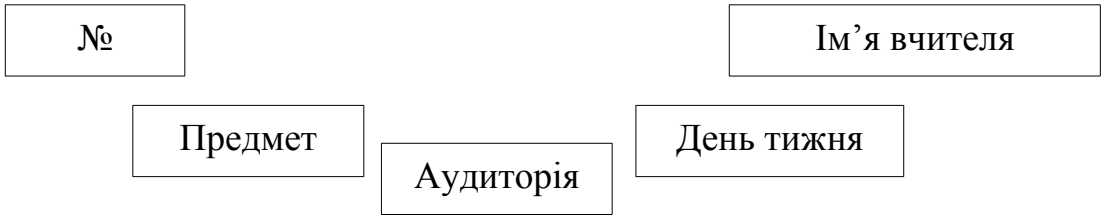

Рисунок 4.5–Схема рядків та колонок файла «Расписание»

№ - номер пари;

Расписание – ця колонка позначає назву предмета по знаменнику ічисельнику;

Аудитория – номер аудиторії де буде проводитися дана пара.

День недели – день тижня;

Имя преподавателя – вчитель який проводитиме пару(рис. 4.7).

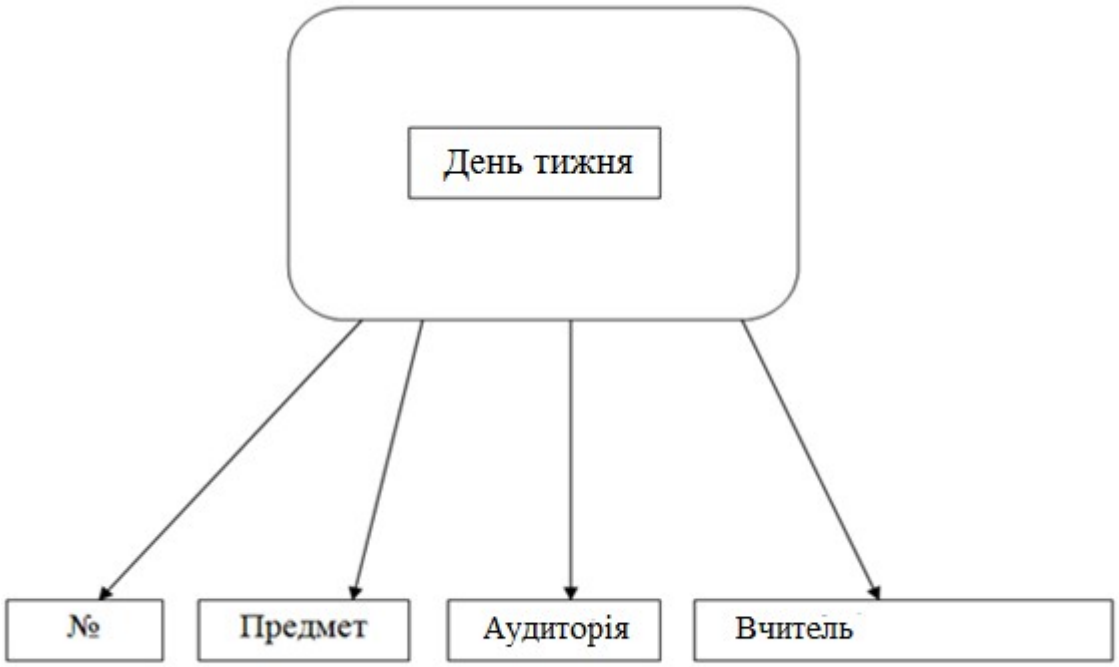

Рисунок 4.6– Схема рядків та колонок файла «Преподаватель»

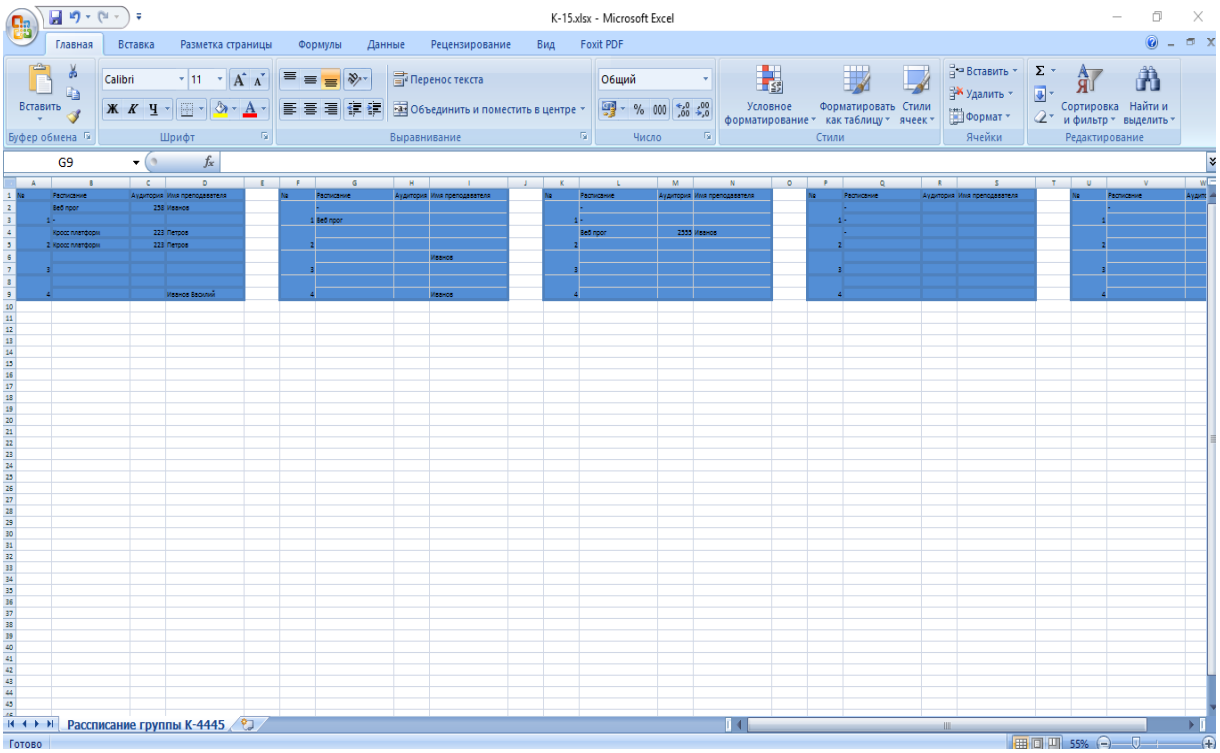

Рисунок 4.7– Документ MSExcel для зберігання данних

№ - номер пари.

Расписание – ця колонка позначає назву предмета по знаменнику ічисельнику.

Аудитория – номер аудиторії де буде проводитися дана пара.

Информация –інформація щодо пари.

Сам файл для збереження данних про розклад та вчителів створюється програмно разом з розміткою в якому формуються пять навчальних днів,рядки та колонки.

4.6Завантаження сайту на хостинг

Хостинг – послуга з надання місця для фізичного розміщення інформації на сервері, що постійно підключенний до інтернету.Крім того, хостингом можна назвати послугу з розміщення обладнання клієнта на території провайдера із забезпеченням підключення до мережі.Є й друга назва цієї послуги – collocation.

Під поняттям хостингу зазвичай мають на увазі як мінімум розміщення файлів сайту на сервері, на якому запущене ПЗ, необхідне для обробки запитів інтернет-користувачів. Додатково до самої послуги надається місце під поштові акаунти, бази даних, резервні копії сайту.

Крім того, до датково клієнт може придбати послуги файлового сховища, адміністрування сайту. Всі компанії надають технічну підтримку веб-проекту.

Чому виникає необхідність в хостингу?

Припустимо, ви написали в html свою сторінку і захотіли, щоб до неї отримали доступ і інші користувачі інтернету. Розмістити сайт або серверну частину сайта на своєму комп'ютері значить встановити складне програмне забезпечення, підтримувати гарну швидкість інтернету і тримати персональний комп'ютер постійно включеним. Ось для чого потрібні хостинг-провайдери: фірми зі спеціальним обладнанням і програмним забезпеченням.

Для розміщення серверної частини було вибрано український найпопулярніший хостингZZZ,який найбільш простий в користуванні.Більше десяти років ZZZ надає професійні хостингові послуги. Ми - лідери на ринку безкоштовних веб-рішень.

Головна місія фірми - надання професійних хостингових послуг і розширення доступу до сучасних технологій. Вони пропонуть безкоштовні і платні пакети послуг, а також домени і сервери VPS. Їх міжнародна команда забезпечує надання послуг у широкому колі країн, в який на даний момент входять Польща, Україна, Росія, Південна Америка, Африка та країни Близького Сходу.

Постійно зростаюча кількість користувачів є доказом того, що наша робота приносить плоди. На даний момент ми є найпопулярнішим хостингом в цьому сегменті на європейському ринку. Ми кожен день працюємо над тим, щоб досягти такого ж успіху в інших країнах.

Ця фірма надає свої послуги на ринку хостингових послуг вже більше 11 років. Серед їх клієнтів – кілька поколінь користувачів Інтернету. Багато з них залишаються з нами з самого початку існування цього хостингу. [20].

Потрібно вибирати якогось провайдера, і зареєструватися. У цьому прикладі, виключно в ілюстративних цілях, ми використовуємо безкоштовний хостинг zzz.com.ua. Для інших хостингів процес буде схожий. Суть цього етапу - отримати реквізити доступу до сервера, на якому розмістимо наш сайт. Процедура стандартна: вказати ім'я, пошту і пароль, підтвердити реєстрацію пройшовши за посиланням, яку відправлять на пошту, і зайти в панель адміністрування.Після авторизації потрібно замовити бесплатний хостинг.

Після реєстрації в розділі "Хостинг" з'являється домен. Потрібно зайти і вибирати його. Далі на панелі буде знаходитиь кнопка "FTP

Доступ" (знаходиться в розділі "Файли") на яку потрібно буде натинути. Дані, які н знадобляться для підключення до серверу через FTP: "FTP сервер", "FTP порт", "FTP користувач" і пароль, якиі були введені трохи раніше, при реєстрації піддомену (рис. 4.8).

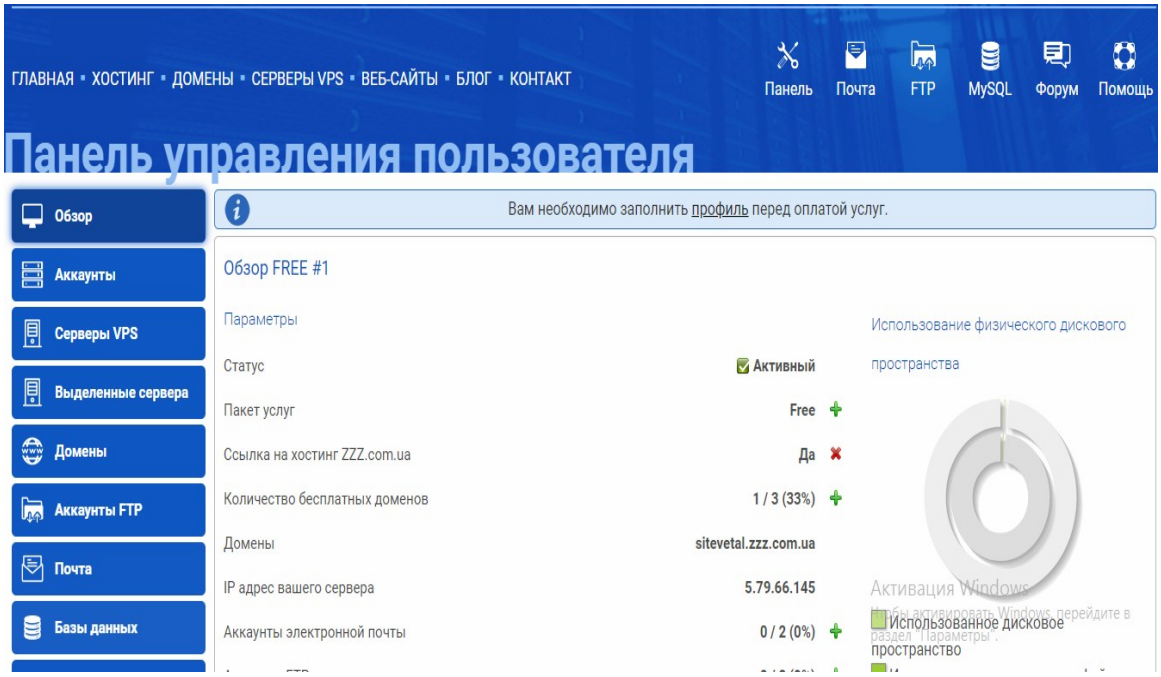

Рисунок 4.8– Розміщення серверної частини додатка

Щоб підключатися до сайту,потрібно використовувати программу FileZilla,яка надає змогу віддалено через FTPпротокол керуватисайтом,завантажувати,видаляти та створювати файли на сервері (рис. 4.9).

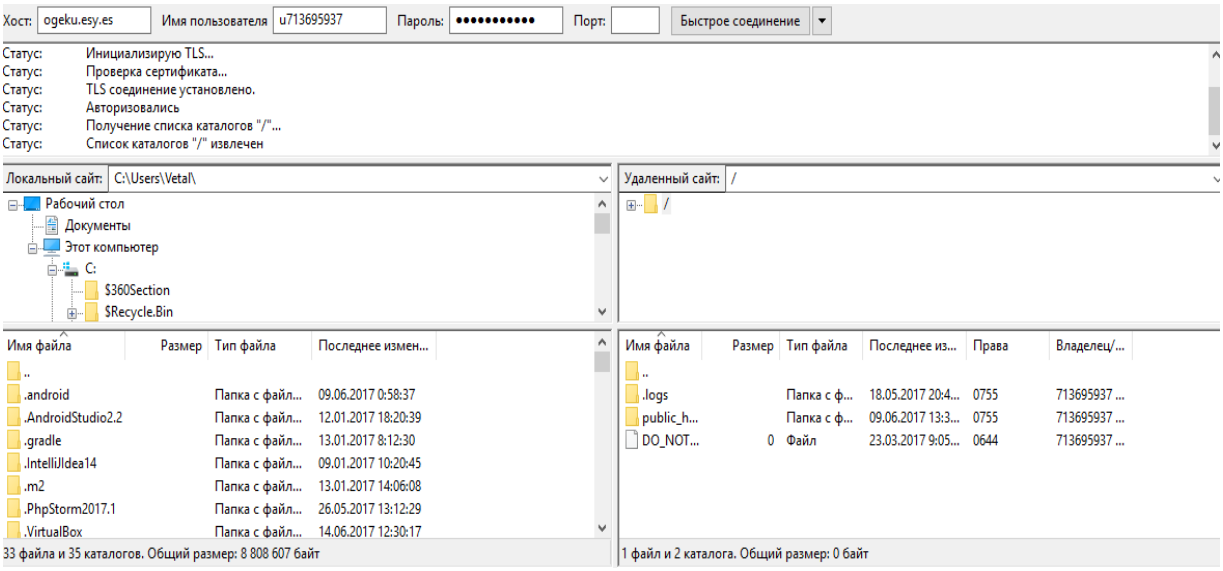

Рисунок 4.9– Доступ до хостингу через FileZilla

Для того щоб потрапити в базу данних потрібно натиснути в верхній частині сайту на силку MYSQL,ввести пароль та логін Бази Даних, після чого користувач потрапдяє в саму БД в якійзнаходиться три таблиціDay, Group, Pair (рис. 4.10). [19].

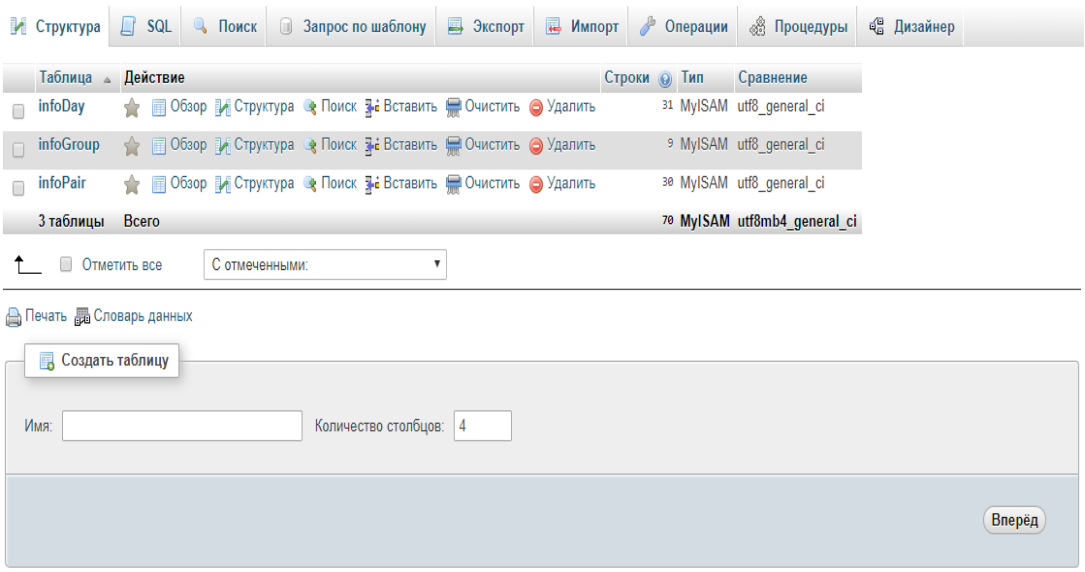

Рисунок 4.10– База Даних серверної частини додатку

4.7 Розробка клієнтської частини

4.7.1Створення дизайну додатка

Для простоти можна вважати, що веб-додаток і веб-сайт - один і той же. Не завжди веб-додаток є сайтом, і навпаки, але поки немає завдання вдаватися в подробиці, оскільки розглядати будемо найпростіший варіант.

Найчастіше веб-додатки складаються як мінімум з трьох основних компонентів:

Клієнтська частина веб додатки– це графічний інтерфейс. Це те, що ви бачите на сторінці. Графічний інтерфейс відображається в браузері,а бо як в нашому випадку на мобільному пристрої . Користувач взаємодіє з веб-додатком саме через браузер, клацаючи по посиланнях і кнопках.

Серверна частина веб-додатку– це програма або скрипт на сервері, обробна запити користувача (точніше, запити браузера). Найчастіше серверна частина веб-додатки програмується на PHP. При кожному переході користувача по посиланню браузер відправляє запит до сервера.

Сервер обробляє цей запит, викликаючи деякий PHP-скрипт, який формує веб-сторінку, описану мовою HTML, і відсилає клієнтові по мережі. Браузер тут же відображає отриманий результат у вигляді чергової веб-сторінки.

Файлова база даних – програмне забезпечення на сервері, що займається зберіганням даних і їх видачею в потрібний момент. Ці файли з даними розташовується на сервері. Серверна частина веб-додатку (тобто, PHP скрипт) звертається до файлових даних, витягуючи дані, які необхідні для формування сторінки, запитаної користувачем.

Клієнт – в нашому випадку, програма на мобільному пристрої, яка вміє формувати зрозумілий серверу запит і читати отриману відповідь. Додаток, який є клієнтом, взаємодіє з сервером, використовуючи певний протокол,в даному випадку я використав HTTP. Додаток може запитувати з сервера будь-які дані, маніпулювати даними безпосередньо на сервері, запускати на сервері нові процеси і т. д. Отримані від сервера дані клієнтська програма може надавати користувачеві або використовувати як-небудь інакше, в залежності від призначення програми. Програма-клієнт і програма-сервер можуть працювати як на одному і тому ж комп'ютері, так і на різних. У другому випадку для обміну інформацією між ними використовується мережеве з'єднання. Для розробки клієнтського додатку я вибрав інтегроване середовище розробки (IDE) для платформи Android(рис.4.11).

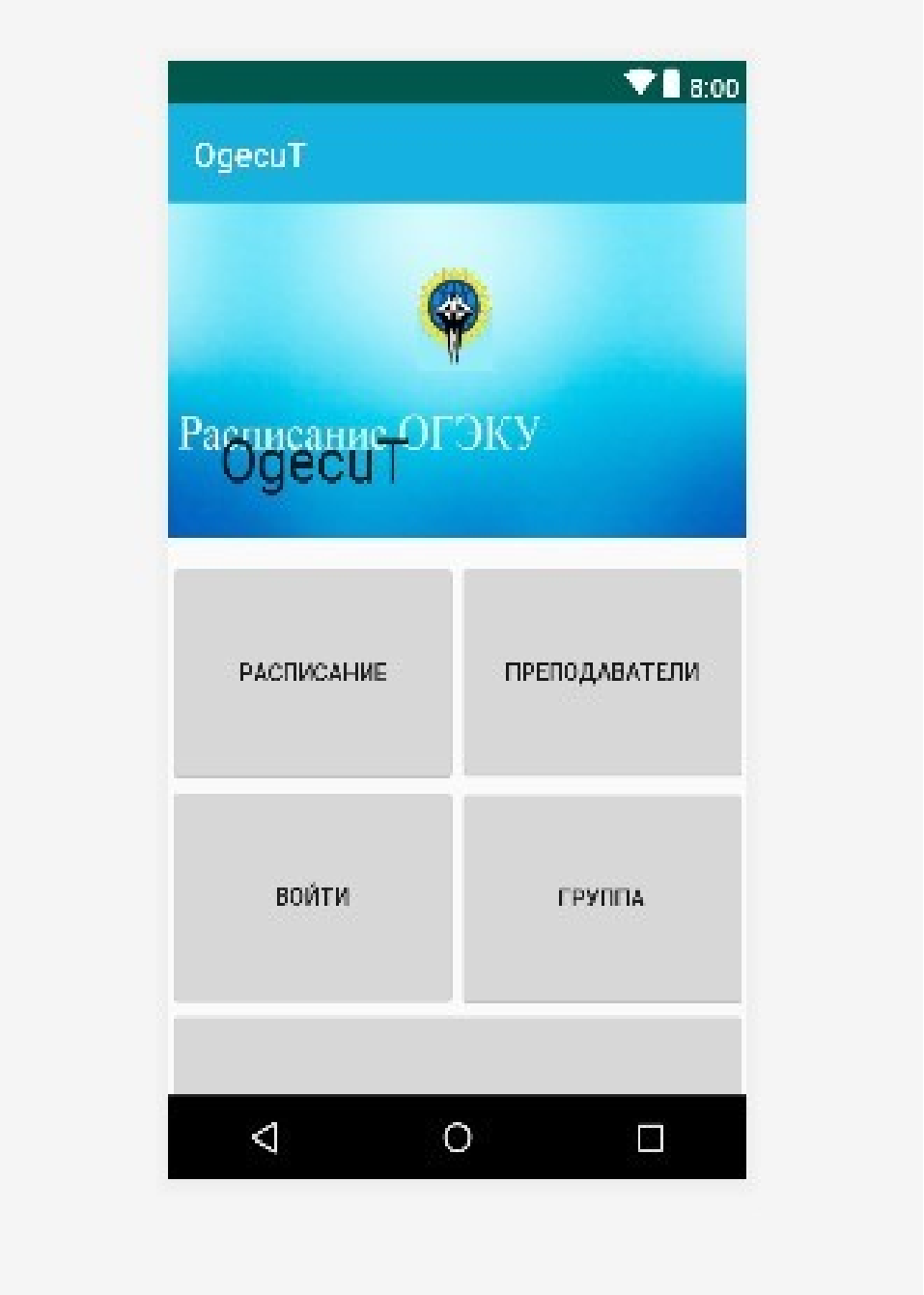

Рисунок 4.11– Інтерфейс клієнтського додатку в процесі розробки

На главному екрані мобільного пристрою розташовані три компоненти Button. При натисканні на одну з кнопок, користувач потрапляє на відповідну активність.

На формах буде розташовано однакова кількість компонентів,але з різними значеннями та ідентифікаторами. Один «Button», «editText» та два «Spinner». Якщо користувач заповне всі значення полів і натисне кнопку «Поиск»,тоді дані з полів відправляться на сервер де будуть оброблятись(рис. 4.12).[20].

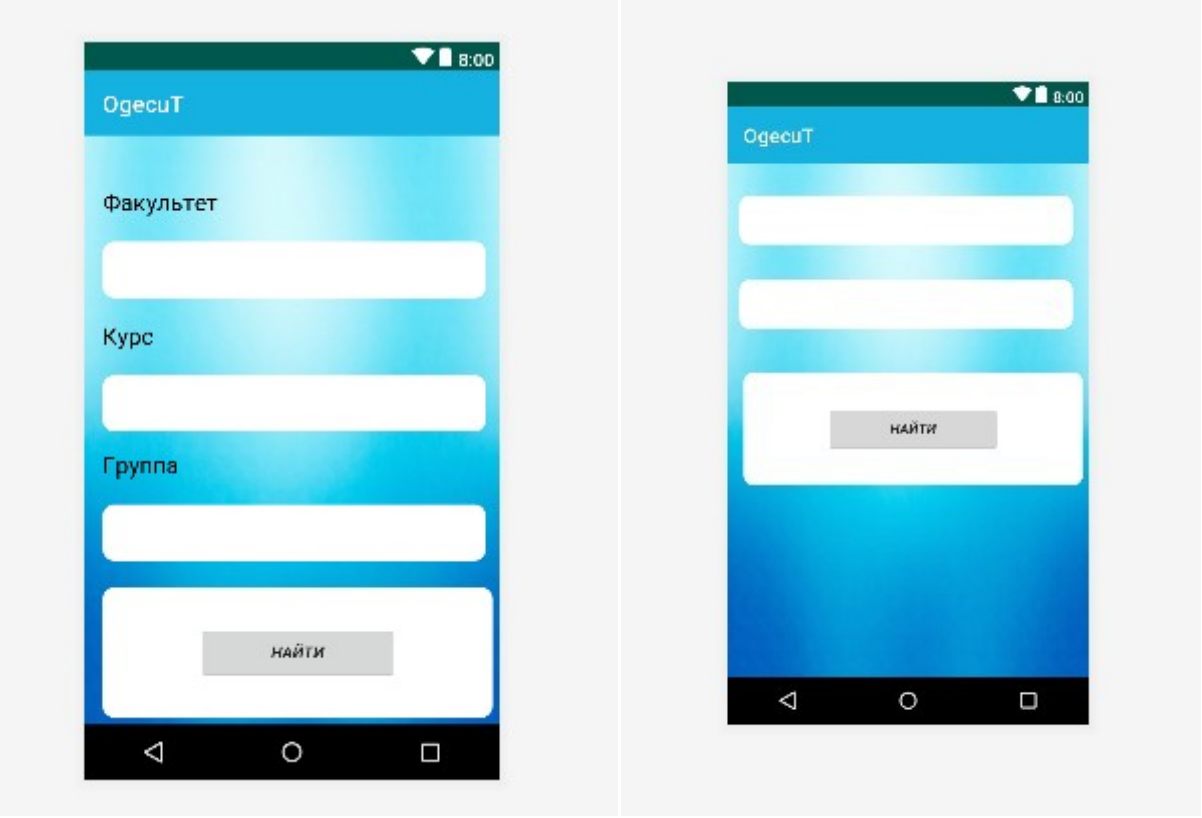

Рисунок 4.12– Інтерфейс двух активностей в процесі розробки

4.7.2 Розробка програмної реалізації додатку

Для розробки програмної реалізації додатку ,було використано різні класи та бібліотеки для AndroidStudio,такі як AsyncTask,Intent, BasicNameValuePair,DefaultHttpClient,HttpPost,ProgressDialog,JSONArray,ArrayList,HashMap,JSONObject,SimpleAdapter та інші.

Клас AsyncTask пропонує простий і зручний механізм для переміщення трудомістких операцій в фоновий потік. Завдяки йому ви отримуєте просто синхронізувати свою обробників подій з графічним потоком, що дозволяє оновлювати елементи призначеного для користувача інтерфейсу для звіту про хід виконання завдання або для виведення результатів, коли задача завершена.

Слід пам'ятати, що AsyncTask не є універсальним рішенням для всіх випадків життя. Його слід використовувати для не надто тривалих операцій - завантаження невеликих зображень, файлові операції, операції з базою даних.

Безпосередньо з класом AsyncTask працювати не можна, вам потрібно успадковуватися від нього (extends).Реалізація повинна

передбачати класи для об'єктів, які будуть передані в якості параметрів методу execute (), для змінних, що будуть застосовуватися для оповіщення про хід виконання, а також для змінних, де буде зберігатися результат. В даному випадку цей клас використовується для передачіпараметрів на сервер через POST запит,де при оголошенні класів DefaultHttpClient та HttpPost ми вказуємо посилання та тип запиту (рис. 4.13).В той час сервер приймає данні уявлення заповненої веб-форми(Додаток А).

Якщо данні будуть переданні успішно,тоді прийде відповідь від сервера в форматі JSON,але його синтаксис який складається з об’єктів та массивів важко сприймається людиною.Саме тому краще застосувати класи JSONObject та JSONArray які дозволяють нам парсити кожен рядок та виводити в окремі блоки або текстові поля (Додаток Б).

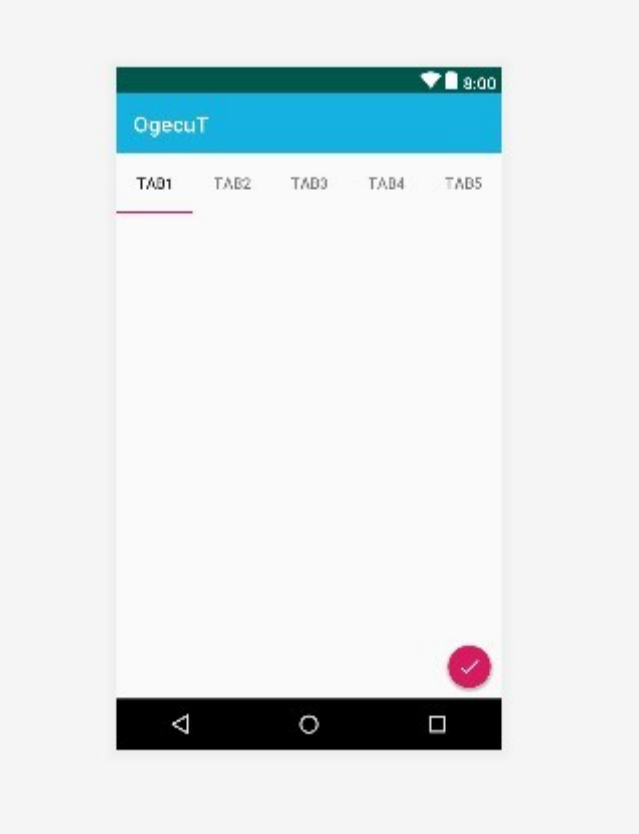

Рисунок 4.13–Форма виведення результату

4.8Класи та бібліотеки для розробки додатку

AsyncTask – клас який дозволяє використовувати поток інтерфейса користувача.Він дозволяє виконувати операції у фоновому режимі та публікувати результат в інтерфейсі для користувача.Цей клас слід використовувати для коротких операцій.При виклику данного

класса,потрібно перевизначити три методи,один з яких є обов’язковим doInBackground.Останні два onProgressUpdate и onPostExecute використовуються за бажанням.Найчастіше метод Asyn-Taskвикористовують для запита на сервер,через HTTP протокол.В моєму випадку,данний клас використовувався для відправки данних на сервер в окремому потоці,щоб зменшити навантаження на додаток та не заблокувати основний поток.

ArrayAdapter – клас який дозволяє повернути данні в вигляді колекції, яку можна інтегрувати в такі компоненти як Grid-View,ListView,TableLayout,TableRowта ін.

SharedPreferences – це інтерфейс для збереження та отримання данних в вигляді Integer,String та ін.Викликати SharedPreferences можна одним із трох методів:

– getPreferences () – всередині активності, щоб звернутися до певного класа;

– getSharedPreferences ()– всередині активності, щоб звернутися до класа на рівні програми;

– getDefaultSharedPreferences ()– з об'єкта PreferencesManager, щоб отримати загальнодоступну настройку, що надається Android;

Для прикладу,данний клас можна використати при збереженні налаштувань у своєму додатку.

Intent є об'єктом обміну повідомленнями, за допомогою якого можна запросити виконання дії у компонента іншої програми. Незважаючи на те, що об'єкти Intent спрощують обмін даними між компонентами у декількох аспектах, в основному вони використовуються в трьох ситуаціях:

Для запуску операції:

Компонент Activity є один екран в додатку. Для запуску нового екземпляра компонента Activity необхідно передати об'єкт Intent методуs-tartActivity (). Об'єкт Intent описує, для чого потрібно запустити, а також містить всі інші необхідні дані.

Якщопіслязавершенняопераціївіднеївимагаєтьсяотриматирезультат, викличтеметодstartActivityForResult(). Ваша операція отримає результат у вигляді окремого об'єкта Intent в зворотному виклику методу onActivityRe-sult () операції. Докладну інформацію див. На сторінці Інструкції Операції.

Для запуску службиService є компонентом, який виконує дії у фоновому режимі без користувальницького інтерфейсу. Службу можна запустити для виконання одноразової дії (наприклад, щоб завантажити файл), передавши об'єкт Intent методу startService(). Об'єкт Intent описує службу, яку потрібно запустити, а також містить всі інші необхідні

дані.Якщо служба сконструйована з інтерфейсом клієнт-сервер, до неї можна встановити прив'язку з іншого компонента, передавши об'єктIntent методу bindService(). Докладну інформацію див. В керівництві Служби.

Для розсилки широкомовних повідомленьШироковещательное повідомлення - це повідомлення, яке може прийняти будь-який додаток. Система видає різні широкомовні повідомлення про системні події, наприклад, коли система завантажується або пристрій починає заряджатися. Для видачі широкомовних повідомлень іншим додаткам необхідно передати об'єкт Intent методу sendBroadcast (), sendOrdered-Broadcast () або sendStickyBroadcast ().

Для того щоб потрапити на іншу активність,потібностворитиоб’єкт Intent,передавши йому в конструктор такі параметри як Con-texttаназвакласу.Накінець,викликаємометод StartActivity() [21].

4.9 Розробка Віджету для додатка

Сьогодні майже у кожного є комп'ютер або ноутбук, телефон або планшет. І тут навіть не важливо для чого людина використовує свій комп'ютер або девайс, а важливо те, що ці пристрої приносять користь. Полегшують роботу з документами, надають доступ до корисної інформації в мережі, та й зрештою, просто дозволяють розслабитися у вільний час і пограти в ігри. При цьому, кожне з них підтримує віджети. Що ж це таке і які бувають віджети? Яке їхнє призначення і сфера застосування?Сучасні віджети можуть повідомляти будь-яку інформацію, доповнювати робочий стіл, забезпечувати обмін сообщеніямі.Чтоби зрозуміти, що таке віджет, згадайте про сигнал, який показує, що вам прийшло чергове лист. Це найпоширеніший приклад віджета.

Віджет це – (Елемент інтерфейсу)примітив графічного інтерфейсу користувача, який має стандартний зовнішній вигляд і виконує стандартні дії.А якщо простіше,то віджетом може бути: Значок, список, кнопка, поле для пошуку інформації, панель та ін.

Віджети класифікують за зовнішнім виглядом, функціоналу і місця установки. За зовнішнім виглядом виділяють наступні види цих програм:

– топпери, розташовані у верхній частині екрану і виглядають як смужка з полем і кнопкою. Виконують інформаційну функцію, повідомляючи про акції і знижки, можуть уживати для збору контактів;

– флор виконує ті ж функції, що і топпер, але знаходиться в нижній частині екрана;

– вспливаючі вікна, що перекривають сторінку сайту. Їх використовують для повідомлень про різного роду акціях або для пропозицій залишити контактні дані;

– ярлички розташовуються збоку екрану і застосовуються для отримання зворотного зв'язку від відвідувачів сайту;

– вбудовувані віджети.

Найпопулярніші віджети якими користуються більшість людей це – GoogleПошук,Карты ,ПовідомленняGmail.

Для свого додатку я також вирішив створити віджет, який автоматично буде визначати день тижня і зчитувати інформацію,яка в подальшому розташовуватиметься в виді таблиці GridView.Для того щоб додати віджет,потрібно натиснути на головний екрани,та вибрати елемент зі списку (рис.4.14).

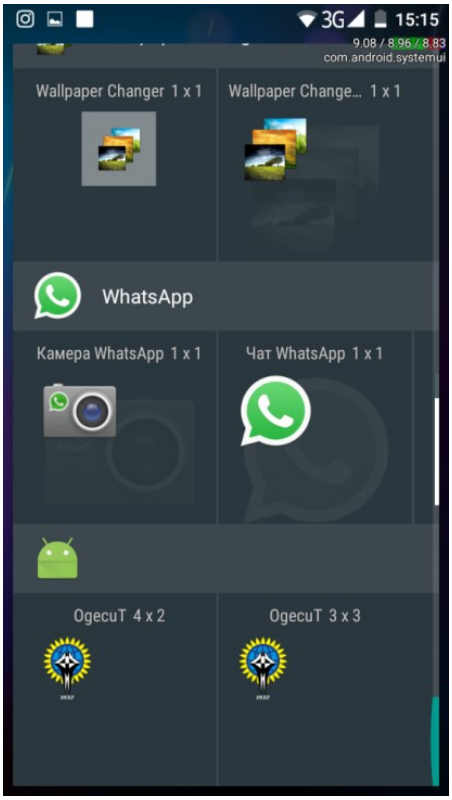

Рисунок 4.14– Установка віджету на головний екран

Для того щоб в додатку був присутній віджет, потрібно створити свій клас, який повинен успадкувати клас AppWidgetProviderз чотирма перевизначеними методами onEnabled, onDisabled, onUpdate, updateAppWidget. В іншому файлі створюємо клас-фабрику який буде успадковуватися від інтерфейсу RemoteViewsService.RemoteViewsFactory

та разом з тим перевизначити методи onDataSetChanged(), getCount(), getViewAt(), getViewTypeCount(), getItemId():

– onDataSetChanged()– метод призначений для привласнення данних массиву (ArrayList);
– getCount() –кількість данних в масиві;
– getViewAt–метод для відображення данних в компоненті.

Щоб запустити клас-фабрику,його потрібно викликати із Сервісу. Тому створюємо свій Сервіс,який також як і попередні класи буде успадковуватись,але від RemoteViewsService. Разом і з ним перевизначається метод onGetViewFactory. Також додаток має можливість автоматично визначати день тижня, та відображати у віджеті саме ту інформацію(предмети) яка відповідає теперішньому часу. Принцип роботи віджета зрозуміти не важко [22].Для того щоб його оновити, потрібно просто нажати на віджет. Це створено за для того щоб з економити батарею та не відправляти зайвий раз запит на сервер (рис. 4.15).

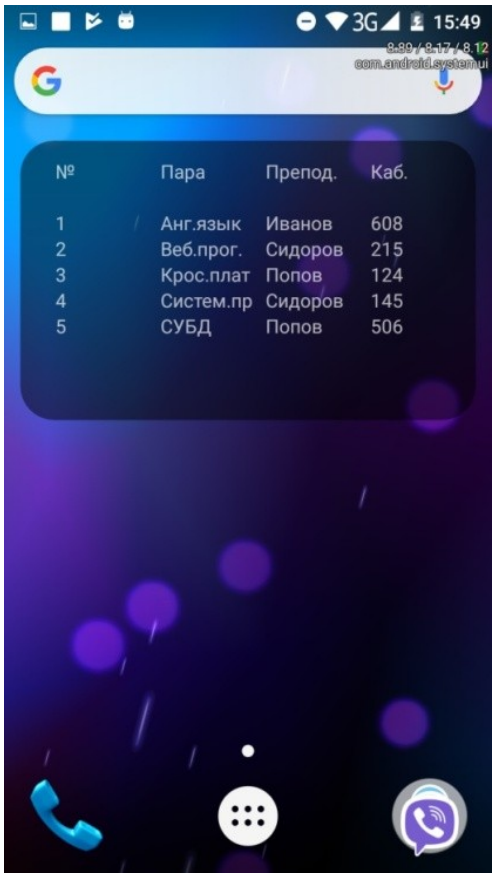

Рисунок 4.15– Інтерфейс віджету який розташований
Наголовному екрані

4.10Створення MVCшаблона проектування

В знайденої мною технічної документації використовуються досить незрозумілі формулювання (типу «використовує модель і уявлення для реалізації необхідної реакції»), вважаю за необхідне написати своє розуміння концепції MVC (щоб не виникало непорозумінь).

Отже, згідно з концепцією MVC, додаток повинен складатися з 3-х фундаментальних логічних частин: controller (контролер), model (модель), view (подання / відображення). Блок controller – перетворює дії користувача (в даному контексті, користувач – не обов'язково чоловік) у вхідні параметри для Model і передає управління в Model. Блок model – реалізує всю логіку роботи програми і готує дані для відображення. Блок view - візуалізує результати роботи програми. Кожна дія користувача завжди запускає ланцюжок controller-> model-> view (рис. 4.16).

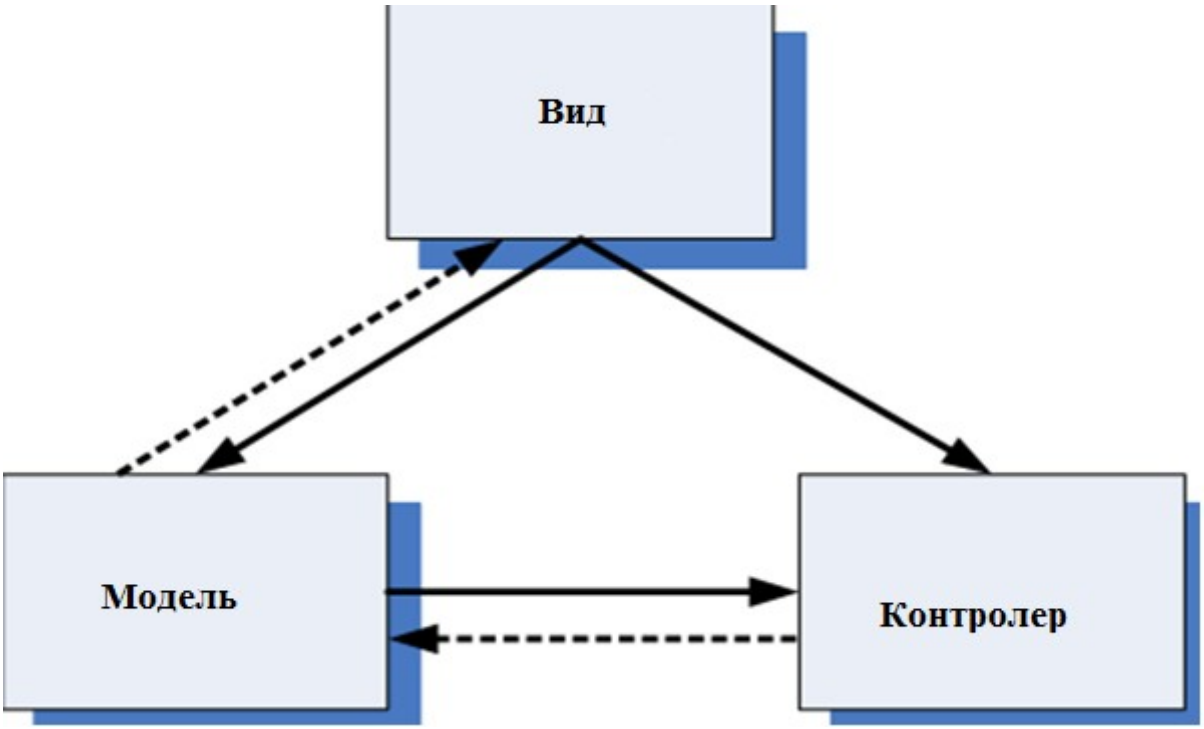

Рисунок 4.16– МодельMVS

Розпишемо функції кожного блоку більш детально, controller:

− завантажує змінні оточення (POST / GET змінні, параметри командного рядка, URL параметри і т. д.);

− виконує первинну обробку змінних оточення (перевірка типів змінних, їх наявність, установка значень за замовчуванням і т. д.);

− реалізує механізми контролю за позаштатними ситуаціями;

− реалізує механізми логгірованія (НЕ аутентифікації, а ведення журналів);

− виконує кінцеву перевірку входять параметрів (допустимість значень, діапазонів і т. д.);

− реалізує взаємодію з системами зберігання даних (бази даних, файли, SOAP і т. д.);

− реалізує логіку роботи програми;

–готує дані для візуалізації;

− організовує механізми візуалізації результатів роботи програми.

Необхідність використання більшої кількості ресурсів. Складність обумовлена тим, що всі три фундаментальних блоку є абсолютно незалежними і взаємодіють між собою виключно шляхом передачі даних. Controller повинен завжди завантажити (і при необхідності створити) всі можливі комбінації змінних і передати їх в Model. Model, в свою чергу, повинен завантажити всі дані для візуалізації і передати їх у View. Наприклад, в модульному підході, модуль може безпосередньо обробляти змінні оточення і візуалізувати дані без завантаження їх в окремі секції пам'яті.

Ускладнений механізм поділу програми на модулі. У концепції MVC наявність трьох блоків (Model, View, Controller) прописано жорстко. Відповідно кожен функціональний модуль повинен складатися з трьох блоків, що в свою чергу, дещо ускладнює архітектуру функціональних модулів програми.

Ускладнений процес розширення функціоналу. Проблема дуже схожа з вищеописаної. Недостатньо просто написати функціональний модуль і підключити його в одному місці програми. Кожен функціональний модуль повинен складатися з трьох частин, і кожна з цих частин повинна бути підключена в відповідному блоці.

Єдина концепція системи. Безсумнівним плюсом MVC є єдина глобальна архітектура програми. Навіть в складних системах, розробники (як ті, які розробляли систему, так і знову приєдналися) можуть легко орієнтуватися в програмних блоках. Наприклад, якщо виникла помилка в логіці обробки даних, розробник відразу відкидає 2 блоку програми (controller і view) і займається дослідженням 3-го (model). Я не раз дивувався, наскільки сильно спростилася локалізація проблем.

Спрощено механізм налагодження програми. Т. к. Весь механізм візуалізації тепер сконцентрований в одному програмному блоці, спростилися механізми опционального виведення графічних елементів. Я не можу оцінити наскільки це твердження може бути застосовано в

програмуванні класичних додатків, але в Web програмуванні ця архітектурна особливість стала безсумнівним плюсом. [23].

4.11Застосування JQuery

Більшість PHPрозробників освоювали цю мову по-старому, почавши з визначення і побудови простих PHPсторінок та їх сполуки з простими ж таблицями MySQL. Надалі, набуваючи різноманітного досвіду, вони вчилися створювати на PHP все більш складну функціональність, об'єднувати таблиці в MySQL і вирішувати інші складні завдання.

При виконанні дипломної роботи я покажу, як використовувати бібліотеку jQuery, щоб легко додавати функціональність Ajax в будь-який PHP-додаток. Ми створюємо Web-сервер із застосуванням PHP і Excel. У ній буде все необхідне розклад інституту з інформацією про викладачів.т.д. Потім ми додамо jQuery, щоб, серверна частина додатку була динамічною та більш сучасною.Тим самим оновлюючи інформацію без перезавантаження сторінок.

jQuery дуже легка бібліотека Javascript (деякі називають її фреймворком), яка позбавляє від головного болю при написанні Javascript коду. У неї багато дуже потужних можливостей, як наприклад: відстеження DOM, додавання красивих ефектів і анімацій до елементів, супер прості Ajax техніки і методи. На головній сторінці сайту jQuery найбільш точне, на мій погляд, опис:

jQuery швидка і лаконічна бібліотека, яка спрощує обробку подій, анімацію та взаємодія з Ajax для більш швидкої веб розробки. jQuery розроблений для того, щоб змінити методи написання JavaScript коду.

В фреймворку JQueryутакі переваги.Давайте коротко пройдемося по деяким із них та властивостями даного фреймворка:

− істотно зменшується кількість коду (необхідного для роботи скрипта) в порівнянні з JavaScript, що в свою чергу означає менше витрат часу і більш читабельний код. Далі в статті будуть розглянуті деякі приклади;

− набагато простіше зрозуміти код (на відміну від JavaScript). У нашому світі, чим швидше Ви закінчите процес програмування, тим більше часу зможете приділити іншим цілям;

− дуже зручна документація та активне ком'юніті, готове завжди надати допомогу при необхідності;

– використання Ajax стає набагато простіше. Вам буде потрібно всього 5 рядків коду (іноді менше) для створення простого Ajax запиту;

– величезна кількість плагінів, за допомогою яких можна зробити практично все що завгодно.

Для того щоб почати працювати з бібліотекою насамперед нам необхідно відвідати головну сторінку офіційного сайту jQuery і завантажити найсвіжішу версію даного фреймворка. Після завантаження необхідно закачати цей файл до себе на хостинг, і в шапці документа прописати посилання на цей файл (рис. 4.17).

```
<script type="text/javascript" src="/jquery.js"></script>
```

Рисунок 4.17– Підключення JQuery

Для виконання нашого jQuery скрипта, нам необхідно помістити весь наш скрипт в функцію. Ця функція буде виконана при повній готовності DOM (коли "документ буде готовий" – дослівний переклад з англ.). Зауважте, що це дуже схоже на популярний подія 'onload', але не є тим же самим(рис.4.18). [24].

```
$(document).ready(function(){
  //Code here
});
```

Рисунок 4.18–Виконання JQueryскрипту

ВИСНОВКИ

В результаті виконання магістерської роботи проведено проектування і виконана програмна реалізація серверної і клієнтської частин додатку розклад пар для вищих навчальних закліадів, яка надає можливість управління системою Адміністратору.У користувача є можливість перегляду розкладу пар та інформації про викладачів.

Швидко зростаючий ринок мобільних додатків продовжує завойовувати найрізноманітніші сфери нашого життя. Люди активно використовують мобільні девайси в повсякденному житті, що безсумнівно збільшує попит на самі різні додатки для мобільних телефонів. Саме цим пояснюється той факт, що більшість користувачів переходять на мобільні смартфони. Мобільні пристрої - це та технологія, яку люди весь час тримають під рукою, так як з їх допомогою можна вкрай швидко отримати достовірні відомості. [26].

Отже, зробивши всі висновки, можна вважати,що зроблений додаток використовуватиметься для будь-якого навчального закладу, томущо завдяки якому можна набагато легше отримати доступ до інформації студенту або викладачеві. Великим плюсом цього додатку є те, що користувачу не подрібно заходити на сайт,а досить всього один раз завантажити його на пристрій.

ПЕРЕЛІК ДЖЕРЕЛПОСИЛАНЬ

1. ДэвидМакфарланд.БольшаякнигаCSS3 –CSS3. СПБ.: Питер,2016. 251 с.
2. Бер Бибо, Иегуда Кац.jQuery. ПодробноеруководствопопродвинутомуJavaScript – СПБ:Питер.2011. 258 с.
3. Бретт Маклафлин.Объектно – ориентированный анализ и проектирование–С.:O'Reilly. 2018. 258 с.
4. Лиза Гарднер, Джейсон Григсби. Разработка веб-сайтов для мобильных устройств– М.:Вильямс. 2014. 268с.
5. Дэвид Сойер Макфарланд.Разработка сайтов на JavaScript и jQuery. Исчерпывающее руководство– С.:O'Reilly,2012.587с.
6. ЭрикФримен, ЭлизабетРобсон. Head First JavaScript Programming – С.: O'Reilly, 2010.267 с.
7. Кэти Сиерра, Берт Бейтс.Head First Java – O'Reilly. 2014.155 с.
8. Роберт Лафоре.Структуры данных и алгоритмы в Java –С.: O'Reilly,2010. 55 с.
9. А.Аллан.Программирование для мобильных устройств – СПБ.: O'Reilly.2012. 188с.
10. Скляр Д., Трахтенберг А..PHP. Рецепты программирования. 3-е изд.– СПБ.:Питер.2012. 335с.
11. А.Аллан.Клиентская разработка для профессионалов. Node.js– СПБ.: Питер.2017. 220с.
12. Роберт Мартин.Чистый код. Создание, анализ и рефакторинг.СПБ.:Питер.2018. 380с.
13. Герберт Шилдт.Java. Полное руководство. 10-е издание.– М.:Вильямс,2018. 725с.
14. Этан Браун.Изучаем JavaScript: руководство по созданию современных. – С.:Диалектика – Вильямс,2016. 550с.
15. Джордж Хайнеман, Гэри Поллис, Стэнли Селков.Алгоритмы. Справочник с примерами на С. С++. Java и Python.С.: Диалектика – Вильямс.2016. 375с.
16. Ян Ф. ДарвинAndroid. Сборник рецептов: задачи и решения для разработчиков приложений.С.:Диалектика-Вильямс.2018. 458с.

17. Фримен Э., Сьерра К., Бейтс Б., Робсон Э.Паттерны проектирования. Обновленное юбилейное издание.СПБ.:Питер.2018. 755с.

18. Майк МакГрат.PHP7 для начинающих с пошаговыми инструкциями.ЭКСМО.2018. 785с.

19. Шварц Б., Зайцев П., Ткаченко В.MySQL по максимуму.СПБ.:Питер.2018. 758с.

20. Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель.SQL: полное руководство.С.:Диалектика.2017. 475с.

21. Брюс Эккель. Философия Java. 4-е полное изд.СПБ.:Питер.2018. 458с.

22. Clean architecture from Uncle Bob. Series. Robert C. Martin. 2017. 745p.

23. Systems Analysis and Design. Bookboon. Howard Gould . 2015.147p.

24. Object Oriented Programming using Java. Bookboon. Simon Kendal. 2014. 216p.

25. Excel 2010 Introduction: Part I. Bookboon. Stephen Moffat. 2014.541p.

26. Java 16: Mobile phones and Android. Bookboon. Poul Klausen. 2017. 209p.

ДОДАТКИ

Додаток А

Лістинг клієнтської частини

```
class RequestTask extends AsyncTask<String, String, String> {
@Override
protected String doInBackground(String... params) {
try {
//создаем запрос на сервер
DefaultHttpClient hc = new DefaultHttpClient();
ResponseHandler<String> res = new BasicResponseHandler();
//он у нас будет посылать post запрос
HttpPost postMethod = new HttpPost(params[0]);
//будем передавать 4 параметра
List<NameValuePair> nameValuePairs = new
ArrayList<NameValuePair>(4);
//передаем параметры из наших текстбоксов
//логин
String f =  new
String(spinner3.getSelectedItem().toString().getBytes("UTF-
8"), "ISO-8859-1");
String k =  new
String(spinner.getSelectedItem().toString().getBytes("UTF-8"),
"ISO-8859-1");
String g =  new
String(group.getText().toString().getBytes("UTF-8"), "ISO-
8859-1");
BasicNameValuePair bass = new
BasicNameValuePair("fakultet",f );
nameValuePairs.add(bass);
nameValuePairs.add(new BasicNameValuePair("kurs", k));
nameValuePairs.add(new BasicNameValuePair("group",  g));
nameValuePairs.add(new BasicNameValuePair("dzen", open));
//собераем их вместе и посылаем на сервер
postMethod.setEntity(new
UrlEncodedFormEntity(nameValuePairs));
//получаем ответ от сервера
String response = hc.execute(postMethod, res);
//посылаем на вторую активность полученные параметры
Intent intent = new Intent(MainActivity.this,
SecondActivity.class);
```

```
//то что куда мы будем передавать и что, putExtra(куда, что);
intent.putExtra(SecondActivity.JsonURL, response.toString());
startActivity(intent);
} catch (Exception e) {
System.out.println("Exp=" + e);
}
return null;
}
@Override
protected void onPostExecute(String result) {
dialog.dismiss();
super.onPostExecute(result);
}
@Override
protected void onPreExecute() {
dialog = new ProgressDialog(MainActivity.this);
dialog.setMessage("Загружаюсь...");
dialog.setIndeterminate(true);
dialog.setCancelable(true);
dialog.show();
super.onPreExecute();
}
}
```

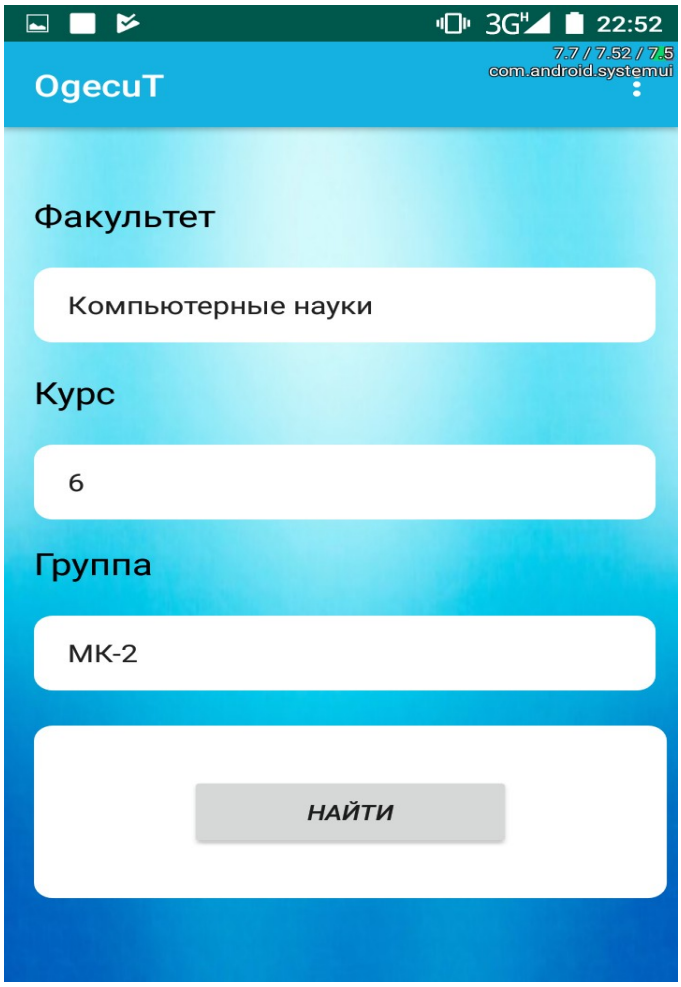

Рисунок А.1 – Інтерфейс додатку для мобільного пристрою

Додаток Б

Лістинг парсингу даних

```
JSONObject json = new JSONObject(result);
JSONArray urls = json.getJSONArray("week");
JSONObject monday = urls.getJSONObject(0);
JSONArray arrs = monday.getJSONArray("Monday");
JSONObject tuesday = urls.getJSONObject(1);
JSONArray arrs2 = tuesday.getJSONArray("Tuesday");
JSONObject wednesday = urls.getJSONObject(2);
JSONArray arrs3 = wednesday.getJSONArray("Wednesday");
JSONObject thursday = urls.getJSONObject(3);
JSONArray arrs4 = thursday.getJSONArray("Thursday");
JSONObject friday = urls.getJSONObject(4);
JSONArray arrs5 = friday.getJSONArray("Friday");
final int numberOfItemsInResp = urls.length();
for (int i = 0; i <= numberOfItemsInResp; i++) {
HashMap<String, Object> hm;
```

```
hm = new HashMap<String, Object>();
hm.put(Monday, Monday);
hm.put(Tuesday, Tuesday);
hm.put(Wednesday, Wednesday);
hm.put(Thursday, Thursday);
hm.put(Friday, Friday);
hm.put(PAIR11,
arrs.getJSONObject(i).getString("num1").toString());
hm.put(PAIR12,
arrs.getJSONObject(i).getString("num2").toString());
myBooks.add(hm);
SimpleAdapter adapter = new SimpleAdapter(SecondActivity.this,
myBooks, R.layout.list,
NewString[]
{ Monday,PAIR11,PAIR12,PAIR21,PAIR22,PAIR31,PAIR32,PAIR41,PAIR
42,
},
Newint[]
{ R.id.text1,R.id.text2,R.id.text3,R.id.textView3,R.id.textVie
w4,R.id.textView5,R.id.textView6,R.id.textView7,R.id.textView8
}
listView.setAdapter(adapter);
listView.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
}
} catch (JSONException e) {
Log.e("log_tag", "Error parsing data " + e.toString());
}
}
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.url);
listView = (ListView) findViewById(R.id.list);
myBooks = new ArrayList<HashMap<String, Object>>();
//принимаем параметр который мы послылали в manActivity
Bundle extras = getIntent().getExtras();
//превращаем в тип стринг для парсинга
String json = extras.getString(JsonURL);
//передаем в метод парсинга
JSONURL(json);
}
}
```

Додаток В

Лістинг серверної частини додатку

```
class Router{
private $routes;
public function __construct(){
$routesPath = ROOT.'/config/routes.php';
$this->routes = include($routesPath)}
private function getURI(){
// Получить строку запроса
if(!empty($_SERVER['REQUEST_URI']))
{
return trim($_SERVER['REQUEST_URI'],'/');
}
}
public function run(){
$uri = $this->getURI();

// Проверить наличие такого запроса в routes.php
```

```
foreach ($this->routes as $uriPattern => $path) {
// Сравниваем $uri и $uripattern
if (preg_match("~$uriPattern~", $uri))
{
// Если есть совпадение,проверить какой контроллер и action
обрабатывают запрос
$internalRoute = preg_replace("~$uriPattern~",$path,$uri);
$segments = explode("/", $internalRoute);
$controllerName = array_shift($segments).'Controller';
//print_r($controllerName);
$controllerName = ucfirst($controllerName); // Делает первую
букву большой
$actionName = 'action'.ucfirst(array_shift($segments));
$parameters = $segments; // Например число ид 1 из адрессной
строки
// Подключить файл класса контроллер
$cotrollerFile = ROOT.'/controllers/'.$controllerName.'.php';
if (file_exists($cotrollerFile)) {
include_once($cotrollerFile);
# code...
}
// Создать обьект,вызвать метод т.е action
$controllerObject = new $controllerName;
$result = call_user_func_array(array($controllerObject,
$actionName),$parameters);
if ($result != null) {
break;
}
}
}
}
}
```

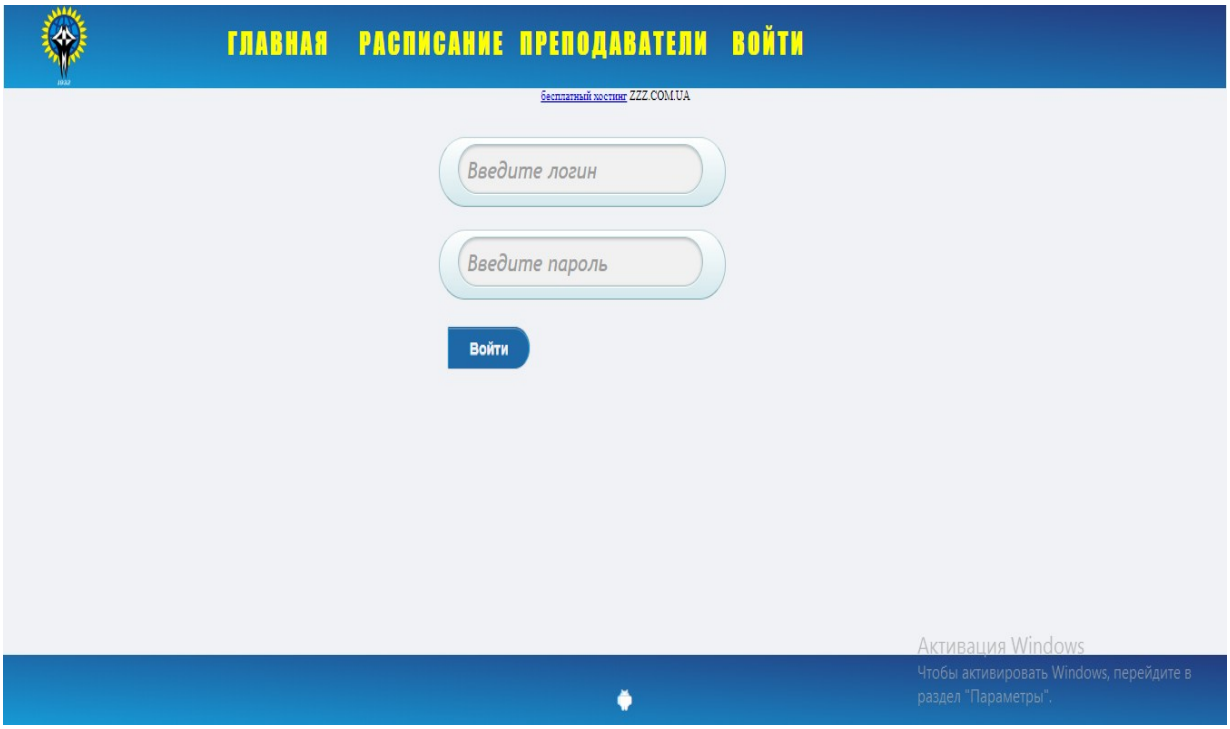

Рисунок В.1 – Інтерфейс серверної частини

Додаток Г

Інтерфейс додатку серверної та клієнтьскої частини

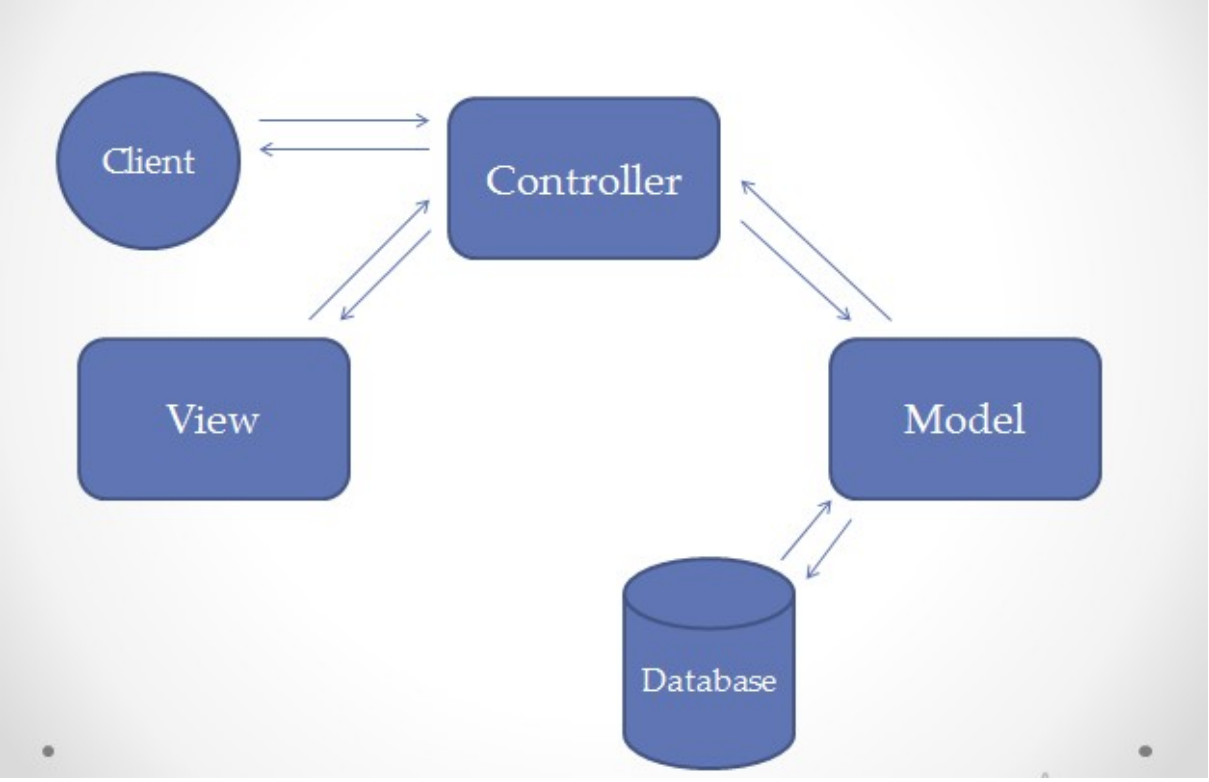

Рисунок Г.1 –Архітектура додатку

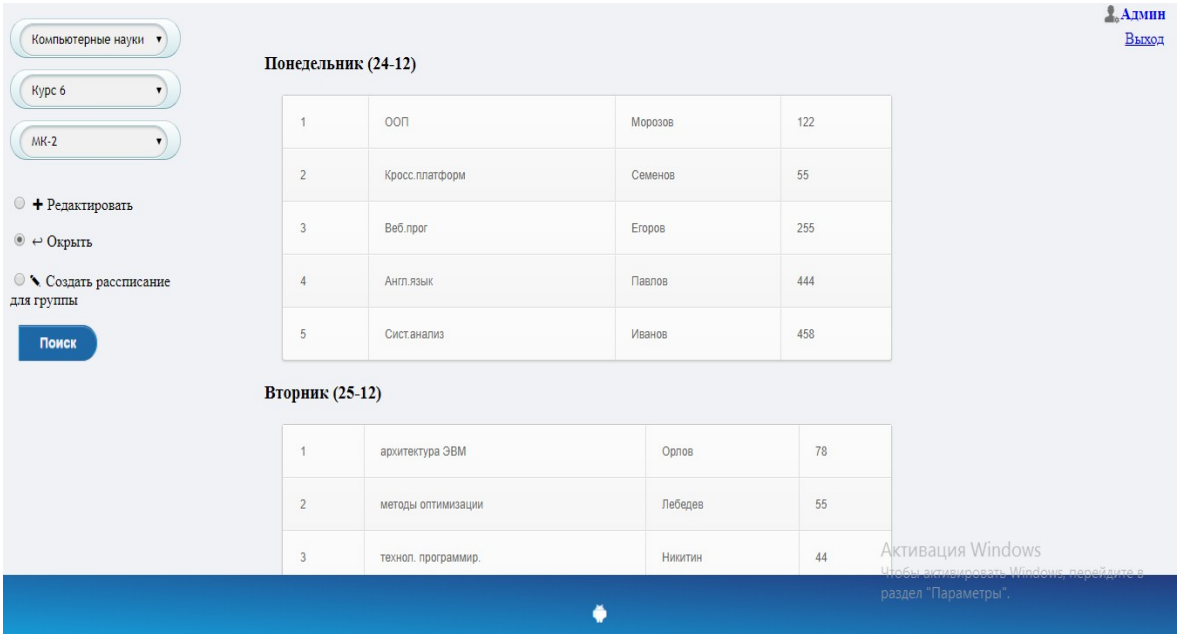

Рисунок Г.2 – Перегляд розкладу на сайті

Рисунок Г.3 – Панель редагування розкладу

Рисунок Г.4 – Інтерфейс додатку

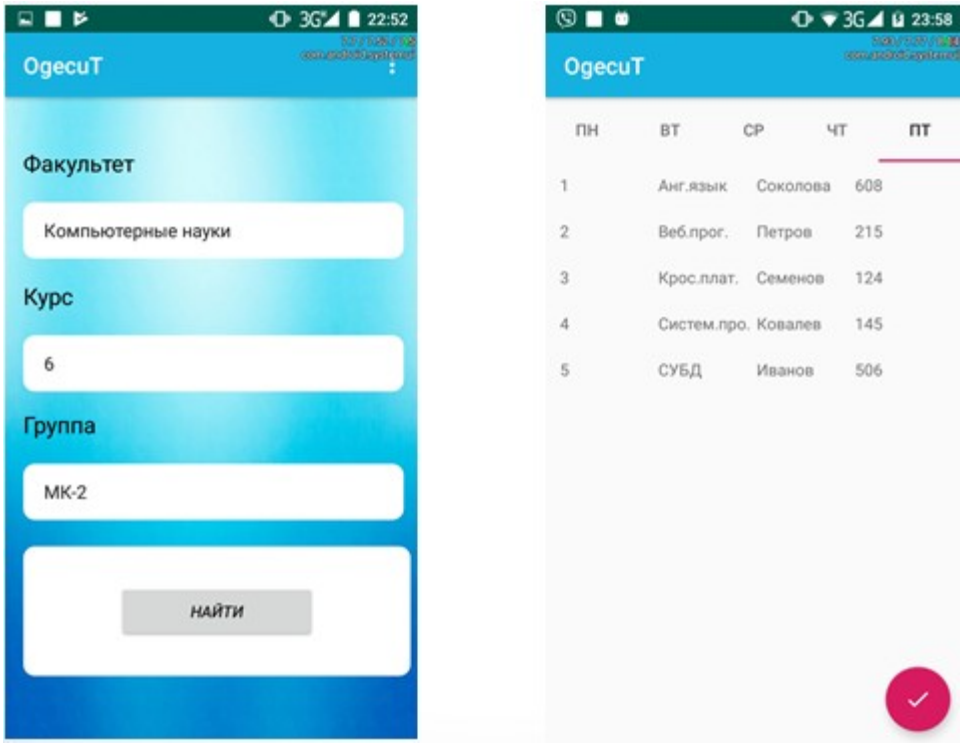

Рисунок Г.5 – Пошук розкладу

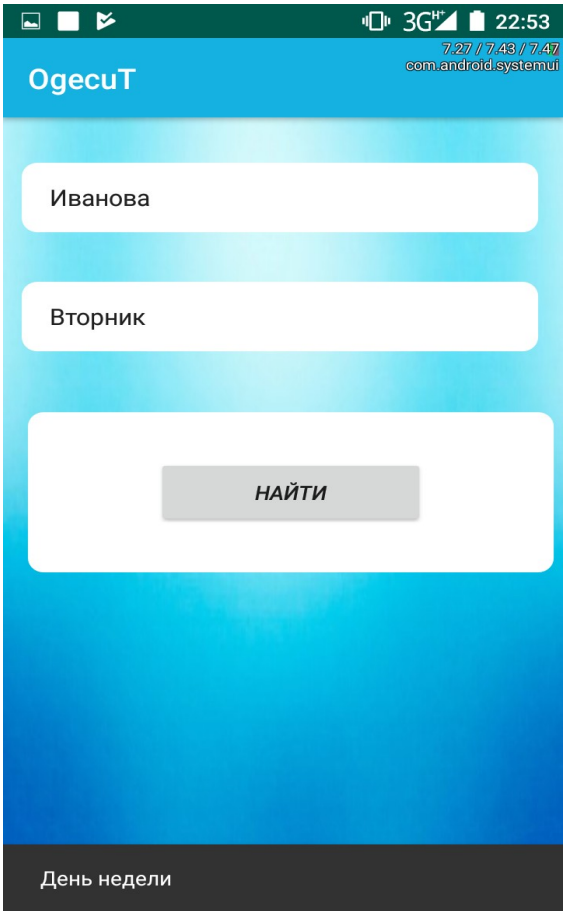

Рисунок Г.6 – Пошук вчителів у клієнтському додатку

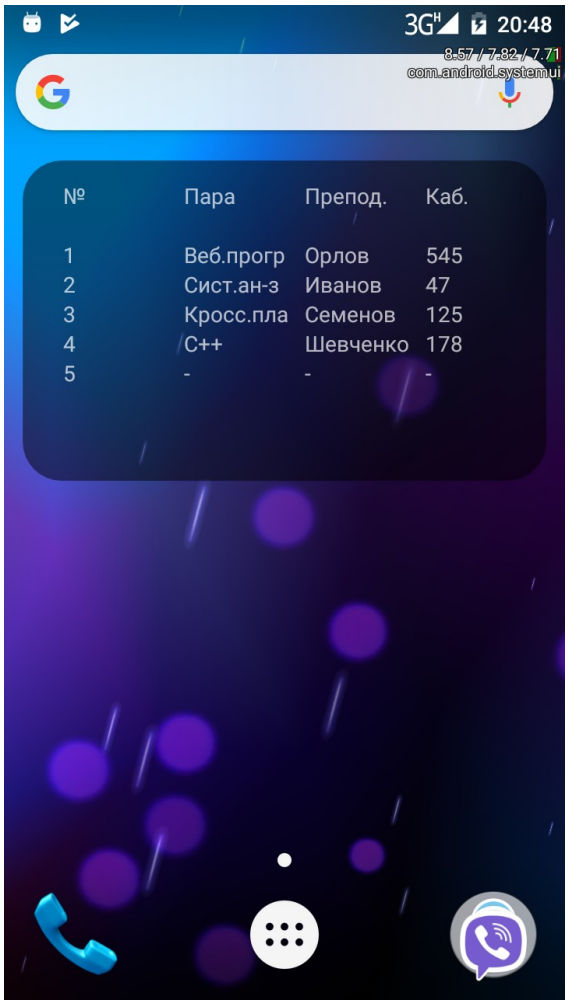

Рисунок Г.7 – Віджет додатку