

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук
Кафедра інформаційних технологій

ДИПЛОМНА РОБОТА

Рівень вищої освіти бакалавр

на тему: Розробка веб-орієнтованої системи розкладу занять ВНЗ

Виконав студент 4 курсу групи К-45
Напрямок підготовки 6.050101
комп'ютерні науки
Гіржев Віталій Сергійович

Керівник к.гегр.н., доц.
Коваленко Людмила Борисівна

Консультант _____

Рецензент д.т.н., проф.
Мещеряков Володимир Іванович

Одеса 2017

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

Скорочення

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

PHP – гіпертекстовий препроцесор – скриптова мова програмування

JSON – JavaScript Object Notation – це текстовий формат обміну даними між комп'ютерами

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

OPEN SERVER – це портативний локальний WAMP / WNMP сервер

Android Studio – інтегроване середовище розробки (IDE) для платформи Android

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.2 Аналіз існуючих аналогів.....	8
1.2.1 Опис додатку УНІВЕР.....	8
1.2.2 Опис додатку РОЗКЛАД МГТУ.....	9
1.2.3 Опис додатку «Расписание занятий СумГУ».....	10
2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
2.1 Діаграма декомпозицій.....	11
2.2 Діаграма дерева вузлів.....	12
2.3 Контекстна діаграма.....	13
3 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ.....	14
3.1 Особливості використання HTML.....	14
3.2 Використання JSON для створення системи.....	15
3.3 Тестування на локальному сервері OPEN SERVER.....	16
3.4 Обробка та оформлення звіту в EXCEL на PHP.....	19
3.5 Використання PHP.....	19
3.6 Опис середовища розробки ANDROID STUDIO.....	21
3.7 Використання MS Excel.....	23
3.8 Вибір шаблону проектування MVC.....	25
4 ПРОЕКТУВАННЯ ІНФОРМАЦІОННОЇ СИСТЕМИ.....	27
4.1 Створення панелі адміністратора.....	27
4.2 Відображення інформації у файлах MS Excel.....	28
4.3 Завантаження сайту на хостинг.....	31
4.4 Розробка клієнтської частини.....	33
4.4.1 Створення дизайну додатка.....	33
4.4.2 Розробка програмної реалізації додатку.....	35
4.6 Створення MVC шаблону проектування.....	36
4.7 Застосування JQuery.....	38
ВИСНОВКИ.....	41
ПЕРЕЛІК ПОСИЛАНЬ.....	42

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 трудно 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 є фактором 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 дипломної роботи: Створення програми для Одеського Державного Екологічного Університету за допомогою якого можна було б переглядати розклад навчального процесу, список пар і викладачів.

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 дипломного проекту:

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

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТИ

1.2 Аналіз існуючих аналогів

1.2.1 Опис додатку УНІВЕР

На мою думку, так мало проектів, які займаються будь-яким розкладом. А якщо і займаються, то все скаржаться на його неактуальність. Один з додатків «Універ», яке 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).

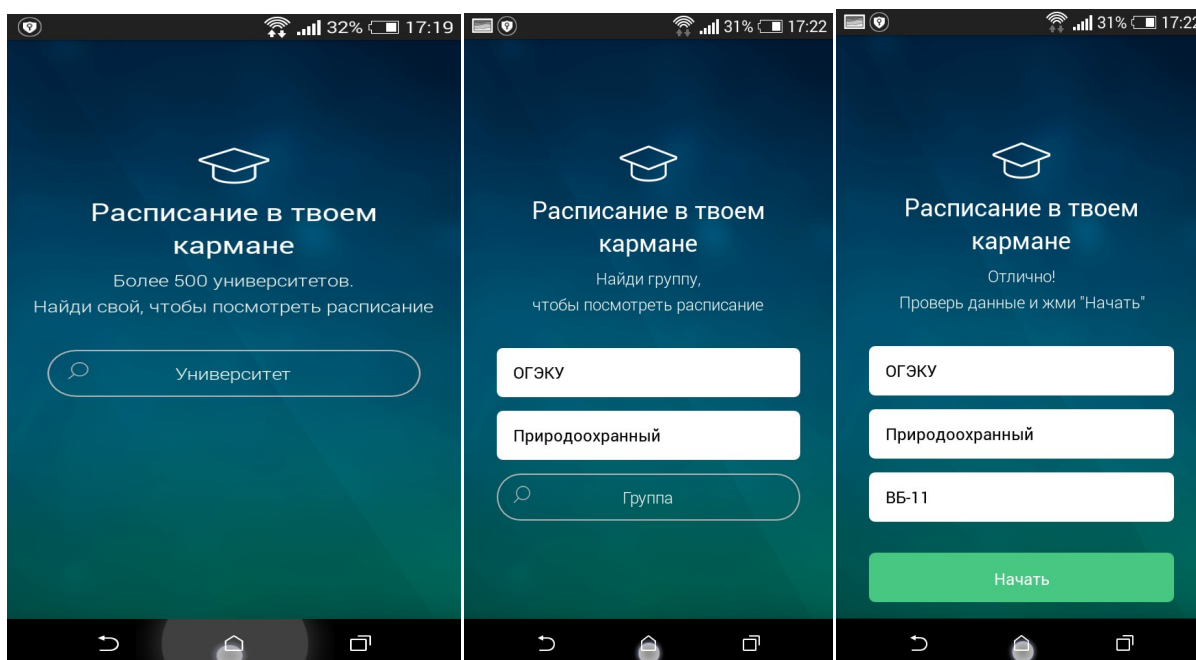


Рисунок 1.1 – Інтерфейс програми Універ. Вибір навчального закладу, факультету та групи.

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).

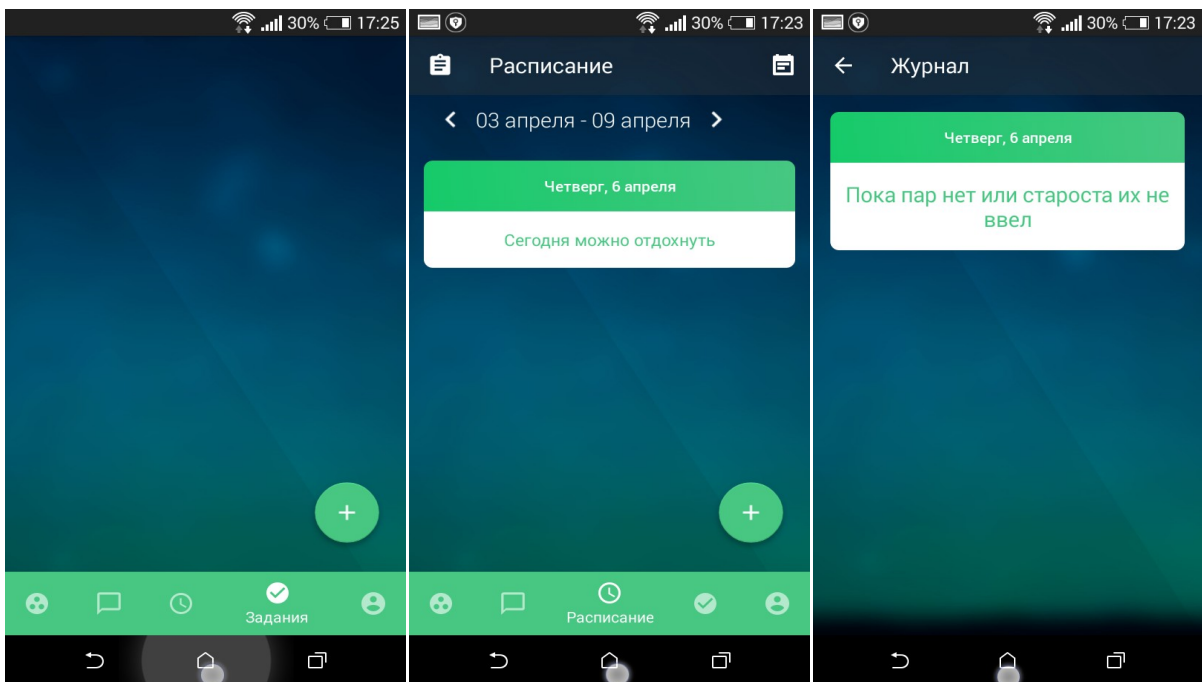


Рисунок 1.2 – Інтерфейс редагування завдання і перегляду розкладу.

1.2.2 Опис додатку РОЗКЛАД МГТУ

Як приклад я наведу ще один додаток яке 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 поліпшити додаток, але є один мінус. Безкоштовна версія діє тільки місяць, після чого блокує доступ до розкладу. Якщо вірити відгукам, то додаток більш популярний в Росії, в Україні майже ним не користуються.

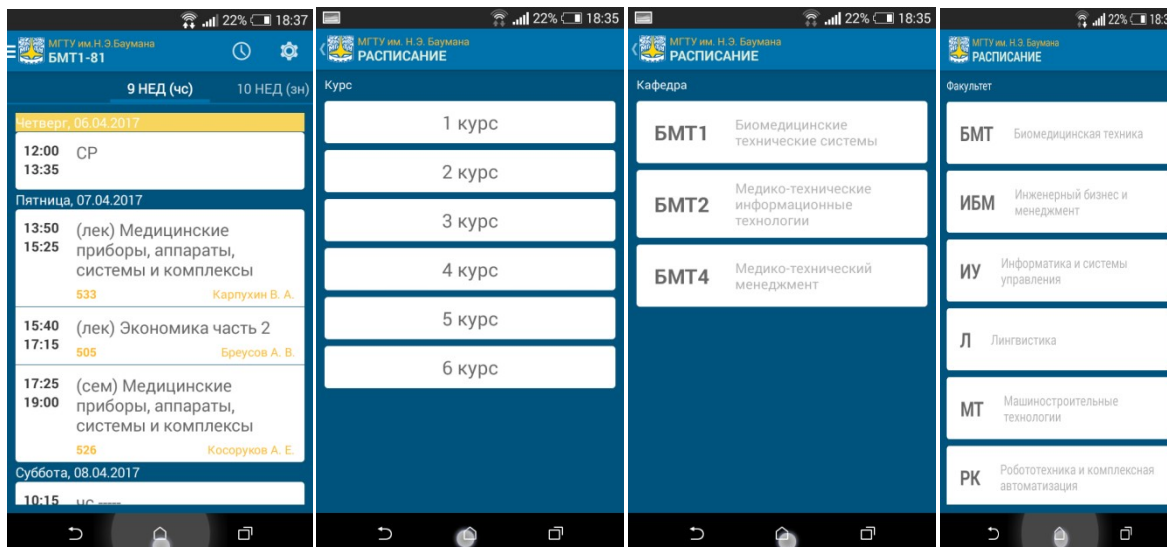


Рисунок 1.3 – Інтерфейс додатку МГТУ

1.2.3 Опис додатку «Расписание занятий СумГУ»

Додаток дозволяє отримувати розкладом навчального процесу, список пар і викладачів (рис. 1.4).

Створено 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.

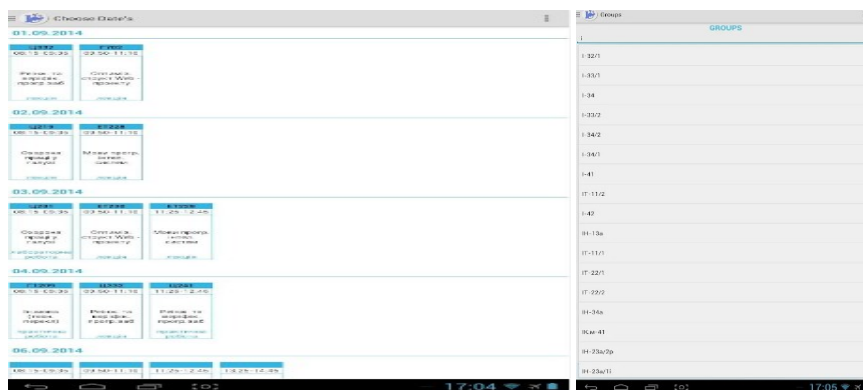


Рисунок 1.4 – Інтерфейс додатку СумГУ
2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Діаграма декомпозицій

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

За допомогою діаграми декомпозиції першого рівня я покажу, 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 та отримати результат. Методологія IDEF0 предписує побудову ієрархічної системи діаграм - єдиничних описаній фрагментів системи. Спочатку проводиться описання системи в цілому та її взаємодія з оточуючим світом (контекстна діаграма), після чого проводиться функціональна декомпозиція - система розбивається на підсистеми та кожна підсистема описується окремо (діаграми декомпозиції). Далі кожна підсистема розбивається на більш дрібні та так далі до досягнення потрібного рівня деталізації.

Кожна IDEF0 – діаграма містить блоки та дуги. Блоки зображують функції моделюваної системи. Дуги пов'язують блоки разом та відображають взаємодії та зв'язки 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 та мають розпізнавані результати. Ім'я роботи повинне бути виражене віддієслівним іменником, що позначає дію.

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

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

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

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

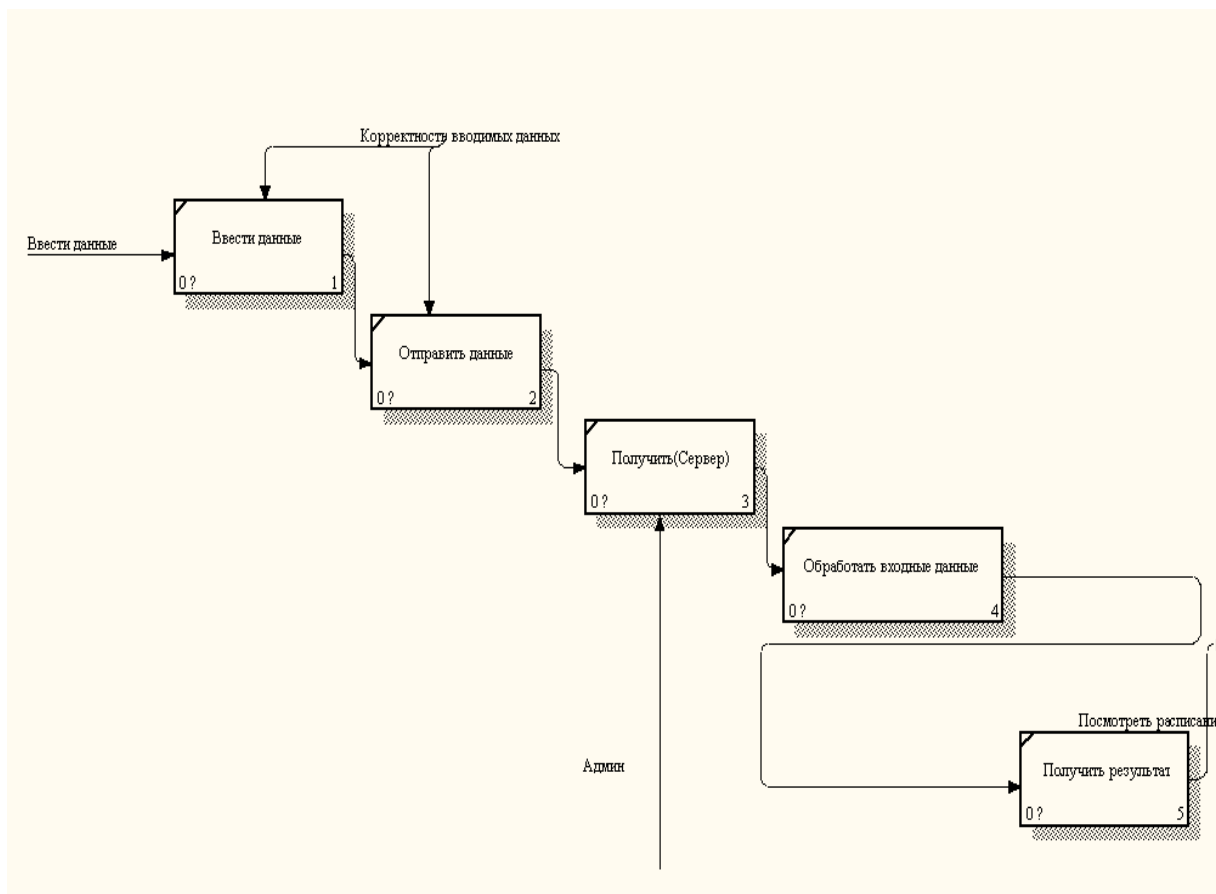


Рисунок 2.1 – Діаграма декомпозиції

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

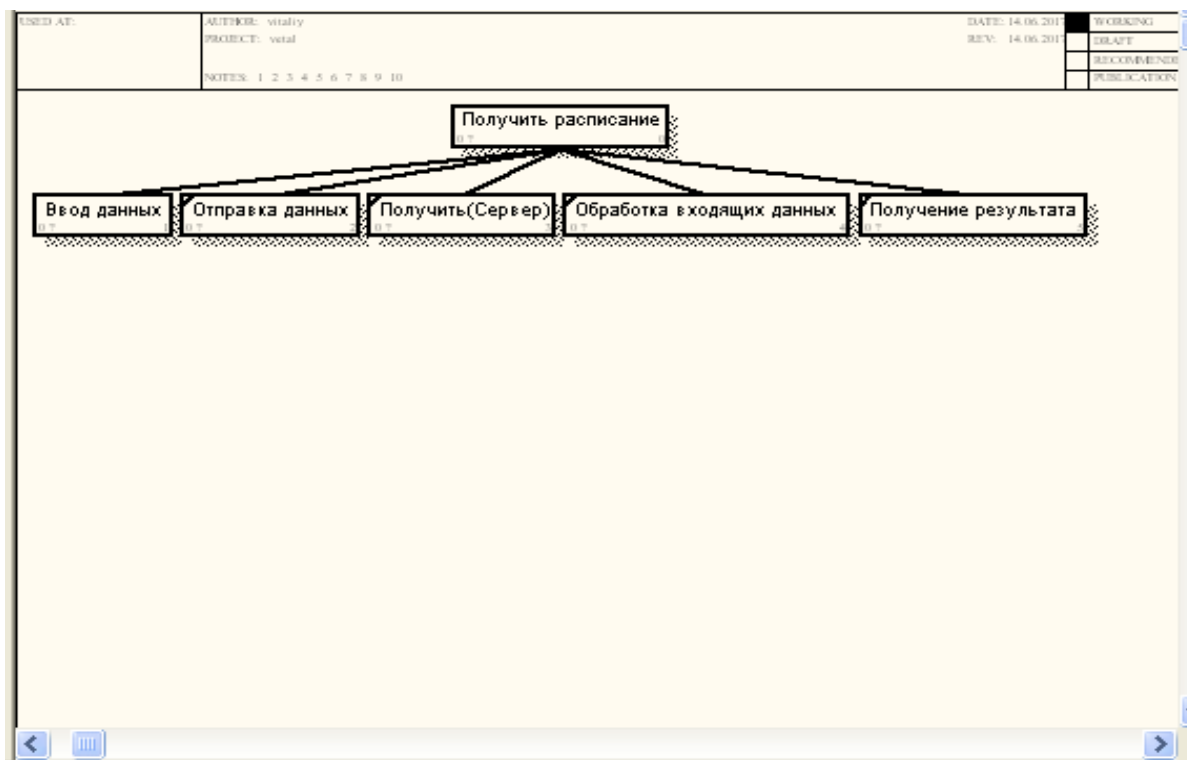


Рисунок 2.2 – Діаграма дерева вузлів

2.3 Контекстна діаграма

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

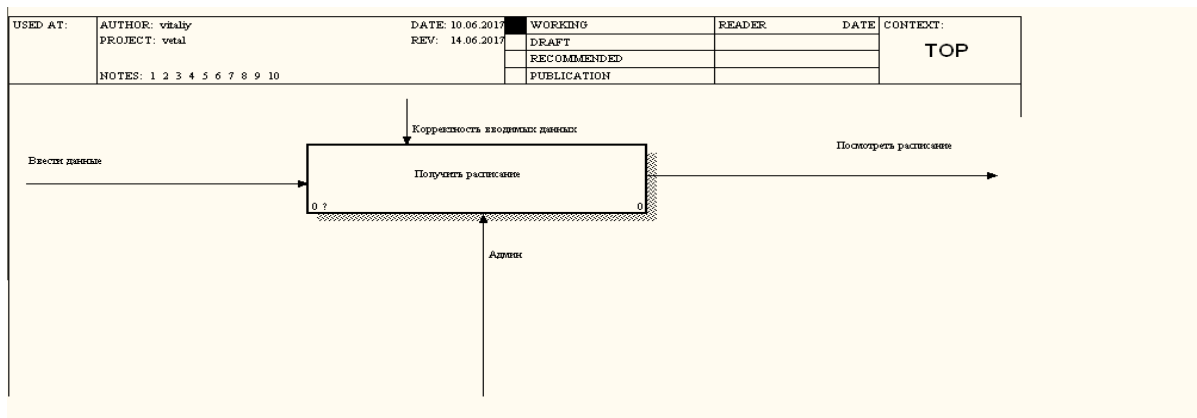


Рисунок 2.3 – Контекстна діаграма
з ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ

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 HTML сторінки, 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 форматovanому 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 Google Chrome, Mozilla Firefox, Opera, Internet Explorer і Safari.

При верстці (грубо кажучи, створення сторінок) HTML сторінок це необхідно пам'ятати, і переглядати їх в найпопулярніших браузерах типу Mozilla Firefox, Internet Explorer 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 Microsoft Office. 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.

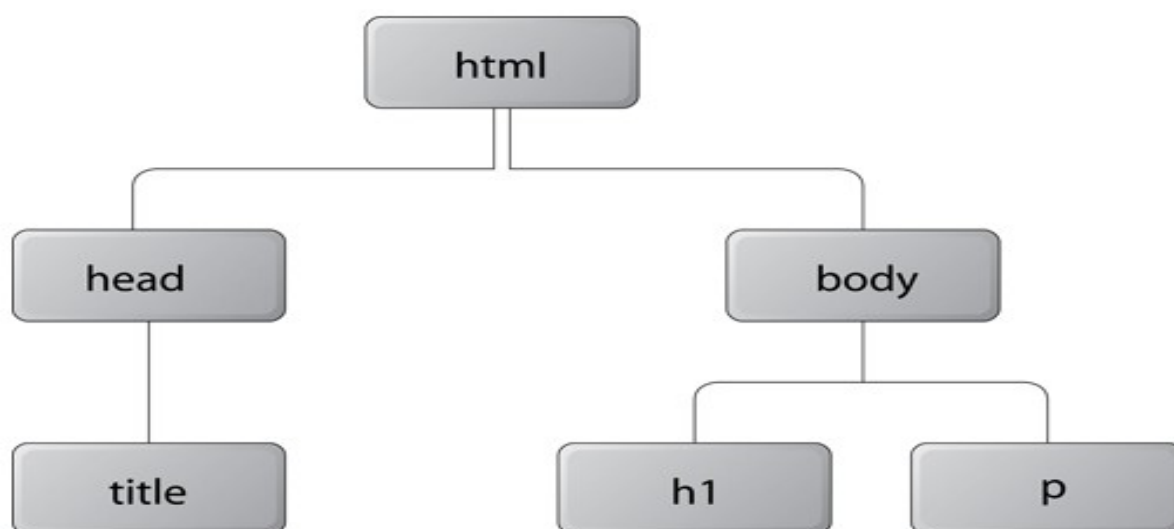


Рисунок 3.1 – Структура HTML розмітки

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 року), 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.

За рахунок своєї лаконічності в порівнянні з XML, формат JSON може бути більш підходящим для серіалізації 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 десеріалізований 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 перетворюють типи даних JSON в відповідні типи PHP і навпаки.

У JSON використовуються їх наступні форми:

Об'єкт – це нерегульована 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 закінчується символом `]`. 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 юнікода, укладені в подвійні лапки, з використанням escape-послідовностей починаються з зворотної косої межі (backslash). Символи представляються простий рядком.

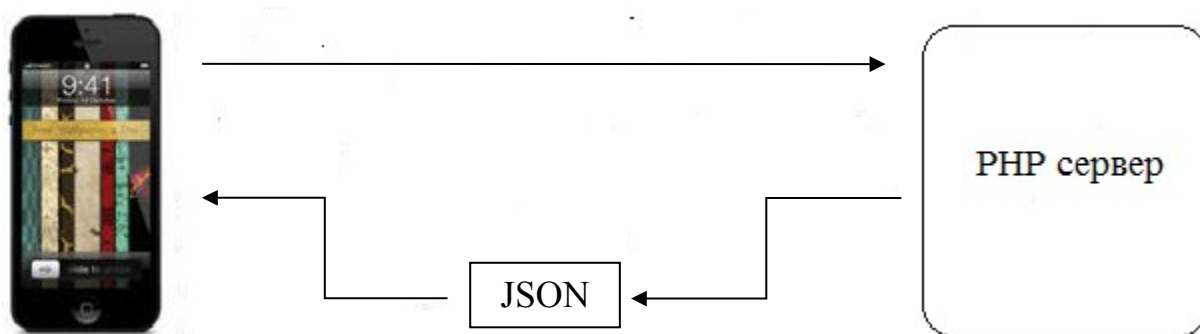


Рисунок 3.2 – Отримання даних у форматі JSON

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. Хоча 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.

Основні можливості Open Server:

- 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.
- Управління 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 піддомена 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.

Зручність і простота управління безумовно не залишать вас байдужими, за час свого існування Open Server зарекомендував себе як першокласний і надійний інструмент необхідний кожному веб-майстру. Для того щоб запустити сайт на локальному сервері його потрібно завантажити папку domains(див.рис.3.1).

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, Parser і ін (рис. 3.3).

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

Використання Open Server не припускає тотального відмови від Денвера. Або будь-якого іншого інструменту (наприклад популярного ХАМРР, який я теж подивлюся). Вони портативні, Портабельная, 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 .

class	08.06.2017 22:53	Папка с файлами	
Classes	23.05.2017 18:06	Папка с файлами	
components	04.01.2017 11:50	Папка с файлами	
config	24.05.2017 1:22	Папка с файлами	
controllers	06.06.2017 19:55	Папка с файлами	
forms	06.06.2017 20:17	Папка с файлами	
index_slider	06.06.2017 13:01	Папка с файлами	
models	06.06.2017 19:57	Папка с файлами	
readedit	07.06.2017 12:12	Папка с файлами	
teachers	09.06.2017 13:37	Папка с файлами	
template	04.01.2017 23:27	Папка с файлами	
timetable	08.06.2017 21:44	Папка с файлами	
views	23.05.2017 23:36	Папка с файлами	
.htaccess	24.05.2017 13:10	Файл "HTACCESS"	1 Кі
aaa.php	03.06.2017 15:58	Файл "PHP"	3 Кі
aaa2.php	09.06.2017 13:30	Файл "PHP"	3 Кі
beautiful_array_output.php	18.05.2017 13:41	Файл "PHP"	1 Кі
exit.php	06.06.2017 16:53	Файл "PHP"	1 Кі
index.php	06.06.2017 20:06	Файл "PHP"	3 Кі
jsonReturn.php	09.06.2017 13:32	Файл "PHP"	2 Кі
jsonReturn2.php	09.06.2017 13:32	Файл "PHP"	2 Кі

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.

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 кроссплатформенною, 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 бщеого 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.

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 кодом. Проект 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.

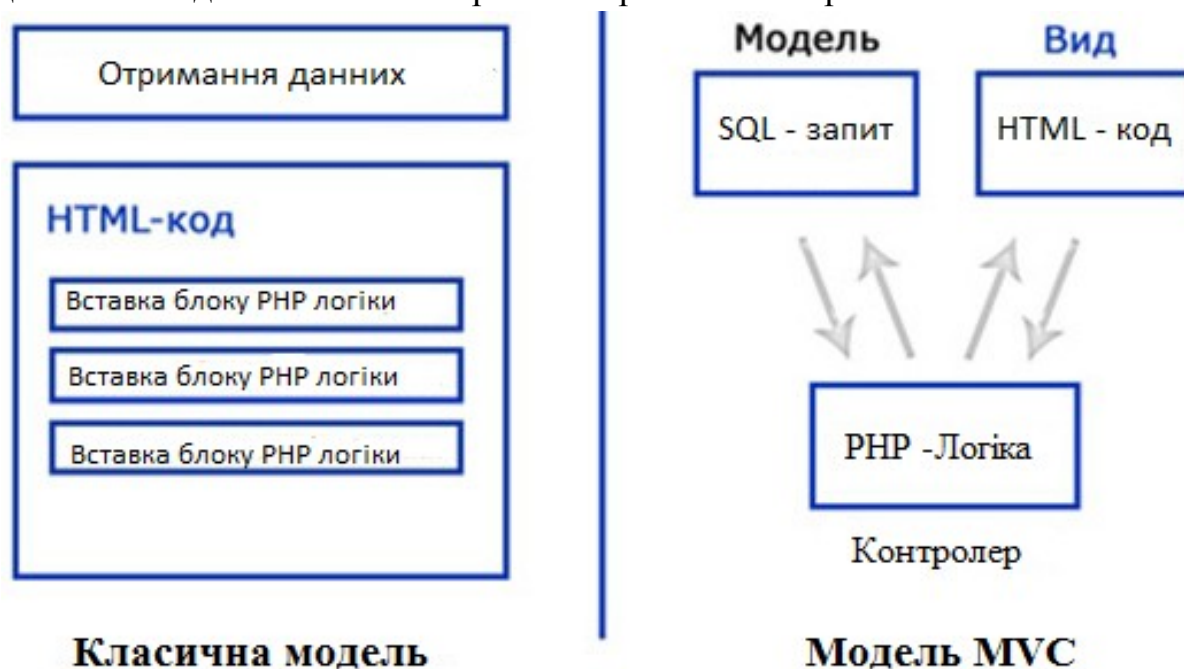


Рисунок 3.4 – Зв'язок між PHP та HTML

3.6 Опис середовища розробки ANDROID STUDIO

Android Studio – це інтегроване середовище розробки (IDE) для роботи з платформою Android, анонсована 16 травня 2013 року на конференції Google I / 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 Android Development Tools (ADT) для Eclipse. Android Studio, заснована на програмному забезпеченні IntelliJ IDEA 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): редагувальник 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).

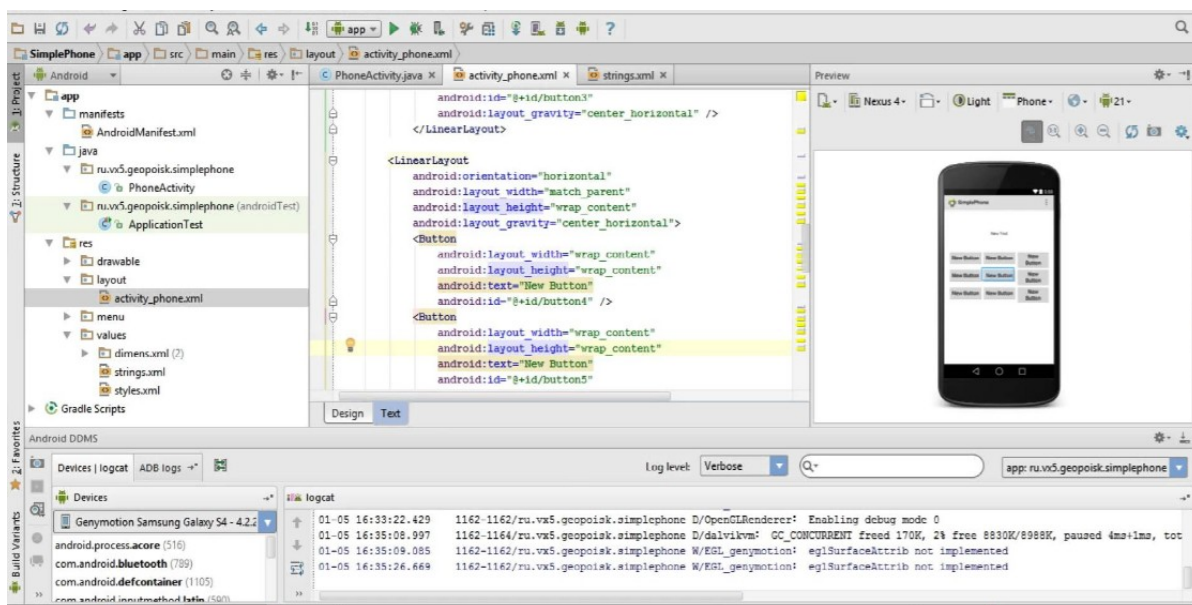


Рисунок 3.5 – Інтерфейс середовища розробки Android Studio

Для створення нестандартних інтерфейсів присутній майстер створення власних елементів оформлення, підтримує використання шаблонів. 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.

3.7 Використання MS Excel

При згадці баз даних (БД) в першу чергу, звичайно, в голову приходять всякі розумні слова типу SQL, Oracle, 1С 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. Ваша програма може 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 считає формулою все, що 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 (арифметичні и т.п.), текст, посилання на агентство комірку або діапазон комірок, круглі дужки, іменовані діапазони. Природно, в формулах дотримується ПРІОРИТЕТ Виконання операцій (множення виконується Ранее Додавання и т.п.). Для Зміни порядку Виконання операцій Використовують круглі дужки.

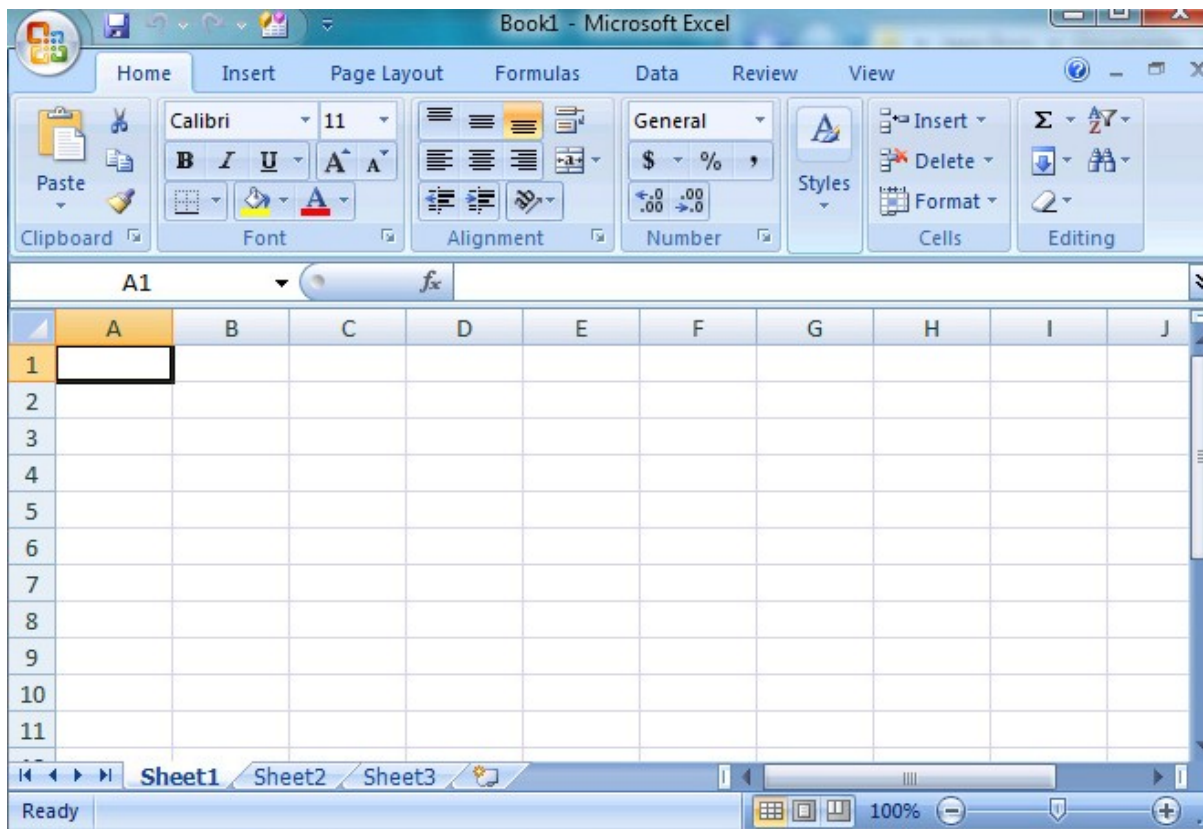


Рисунок 3.6 – Інтерфейс програми MS Excel

3.8 Вибір шаблону 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 модифікація шкірного компонента может здійснюватися 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 HTTP GET 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).

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.

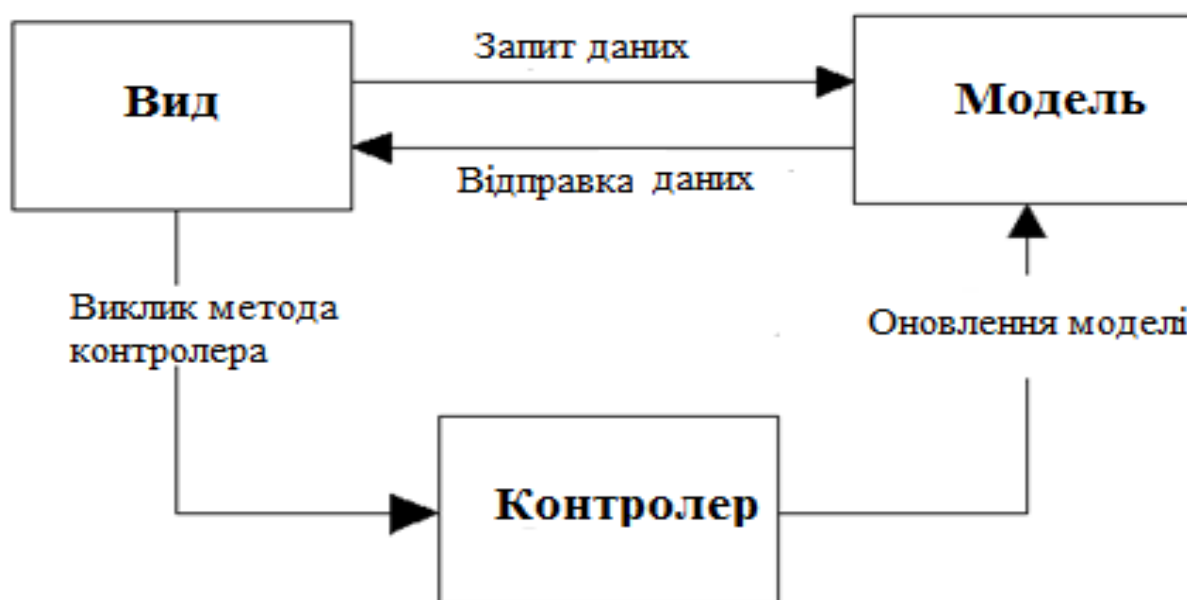


Рисунок 3.7 – Архітектура MVC

4 ПРОЕКТУВАННЯ ІНФОРМАЦІОННОЇ СИСТЕМИ

4.1 Створення панелі адміністратора

Адмін це «серце» сайту або блогу. З адмін-панелі користувач робить всі необхідні дії: налаштовує дизайн сайту, редагує, створює та видаляє розклад. Також змінює різні категорії з інформацією про викладачів. Загалом, в адмін-панелі укладені всі налаштування і елементи, які

забезпечують стабільну роботу сайту, а також його оновлення. Це приватна, 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 адмінкою 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. 4.1).

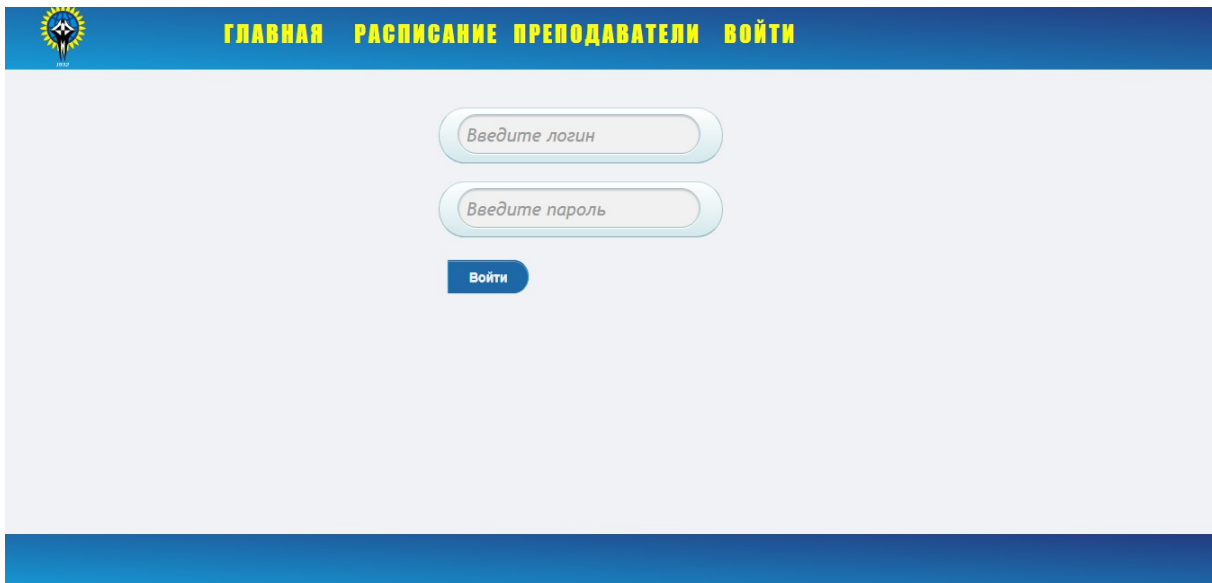


Рисунок 4.1 – Сторінка доступу до адмін-панелі

В разі успішної авторизації, користувач потрапляє на сторінку де надається право змінювати, створювати, а також видаляти інформацію про викладача, або ж розклад (рис.4.2).

ГЛАВНАЯ РАСПИСАНИЕ ПРЕПОДАВАТЕЛИ ВОЙТИ

Факультет ИИ

Курс 4

К-45

+ Редактировать

← Открыть

Создать расписание для группы

Поиск

Админ Выход

ПОНЕДЕЛЬНИК

Предмет	Аудитория	Имя преподавателя
1) Веб прог	288	Иванов
Веб прог	257	Иванов
2) .	.	.
3) Спектральный ан-з	201	.
4) .	.	.

ВТОРНИК

Предмет	Аудитория	Имя преподавателя
.	.	.

Рисунок 4.2 – Сторінка редагування та створення розкладу

4.2 Відображення інформації у файлах MS Excel

Часто буває, що база даних має просту структуру і невеликий розмір. Як приклад можна привести список викладачів невеликого вузла, список предметів 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-якій комп'ютерній 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 даними можна скачати, або передати на сервер за допомогою протокола FTP, що дає змогу редагувати та створювати файл без з'єднання через інтернет.

Думаю у многих возникла необходимость хранить файлы, связанные с записью в таблице. Это может быть картинка к новости, аватар, загруженный пользователем файл — да все, что угодно. Обычно в этом случае поступают просто — файл ложится в файловую систему, а ссылка на него — в запись БД.

Но у такого классического подхода множество недостатков:

- файлы не удаляются при удалении соответствующей записи БД
- проблемы при одновременной попытке обновления файла
- нарушение синхронизации между БД и файловой системой при откате транзакции
- при резервном копировании и восстановлении информации в БД может возникнуть рассинхронизация с файловой системой
- файлы не подчиняются ограничениям доступа, наложенным с помощью БД

Саме тому я вирішив замінити базу даних на файли з розширенням «xlsx». Для цього використовуються файли двох різних типів. Це «преподаватели» та «расписание» яким присвоюються відповідні поля (рис. 4.3).

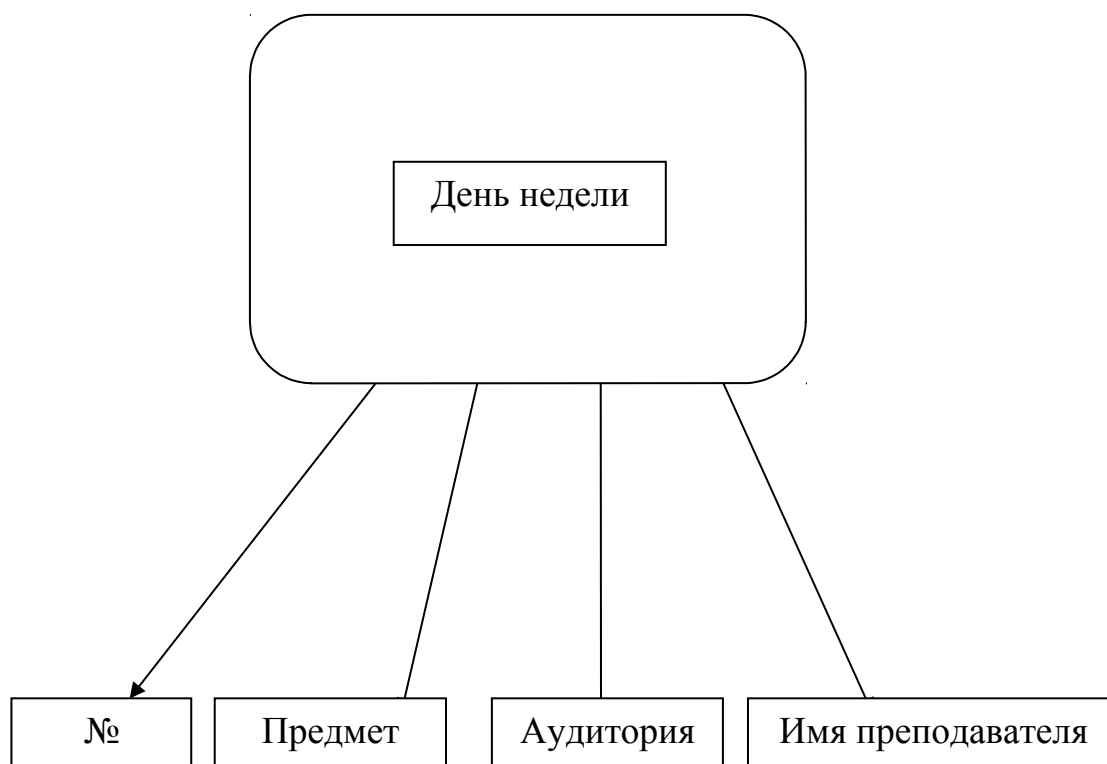


Рисунок 4.3 – Схема рядків та колонок файла «Расписание»

№ - номер пари.

Расписание – ця колонка позначає назву предмета по знаменнику і чисельнику.

Аудитория – номер аудиторії де буде проводитися дана пара.

Имя преподавателя – вчитель який проводитиме пару (рис. 4.4).

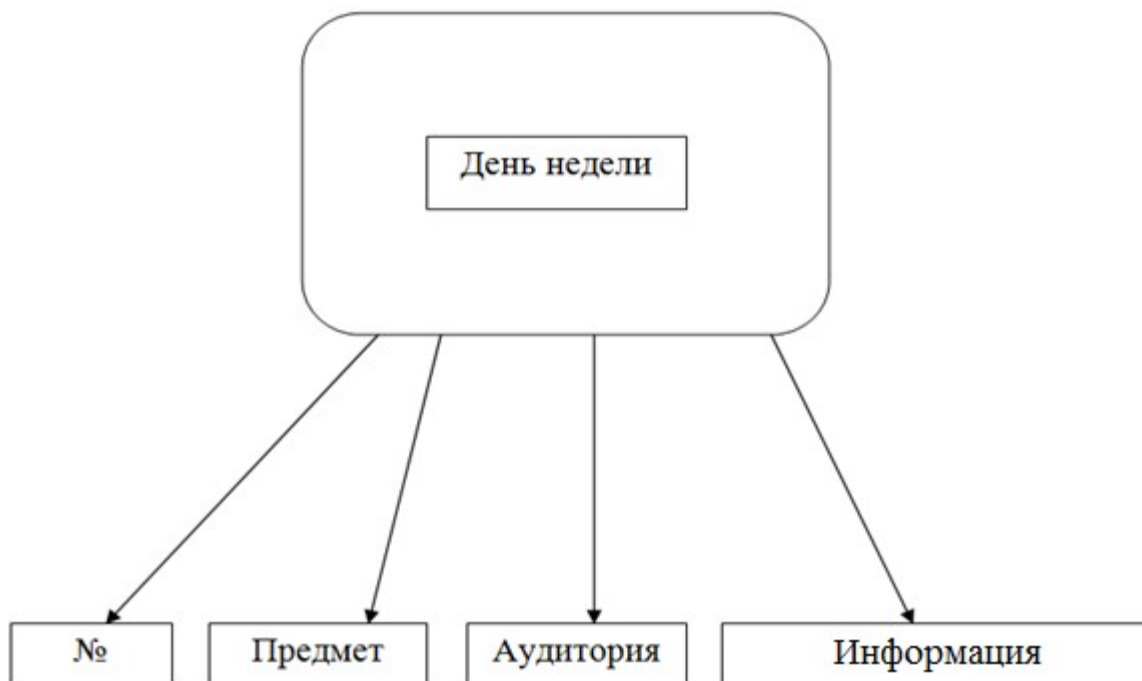


Рисунок 4.4 – Схема рядків та колонок файла «Преподаватель»

№	Расписание	Аудитория	Имя преподавателя	№	Расписание	Аудитория	Имя преподавателя	№	Расписание	Аудитория	Имя преподавателя
1	Веб-прог	120	Иванов	1	Веб-прог			1	Веб-прог	120	Иванов
2	Иванов			2	Иванов			2	Иванов		
3	Иванов			3	Иванов			3	Иванов		
4	Иванов			4	Иванов			4	Иванов		

Рисунок 4.5 – Документ MS Excel для зберігання даних

№ - номер пари.

Расписание – ця колонка позначає назву предмета по знаменнику і чисельнику.

Аудитория – номер аудиторії де буде проводитися дана пара.

Информация – інформація щодо пари.

Сам файл для збереження даних про розклад та вчителів створюється програмно разом з розміткою в якому формуються п'ять навчальних днів, рядки та колонки (рис.4.5).

4.3 Завантаження сайту на хостинг

Хостинг – послуга з надання місця для фізичного розміщення інформації на сервері, що постійно підключений 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, додатково 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 сайту 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 хостинг Hostinger, який найбільш простий в користуванні.

вибираємо якогось провайдера, і реєструємося. У цьому прикладі, виключно в ілюстративних цілях, ми використовуємо безкоштовний хостинг hostinger.ru. 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 безплатний хостинг

Після реєстрації в розділі "Хостинг" з'являється наш домен. Заходимо і вибираємо його. Далі в панелі шукаємо кнопку "FTP Доступ" (знаходиться в розділі "Файли") і натискаємо на неї. Дані, які нам знадобляться: "FTP сервер", "FTP порт", "FTP користувач" і пароль, який вводили трохи раніше, при реєстрації піддомена (рис. 4.6).

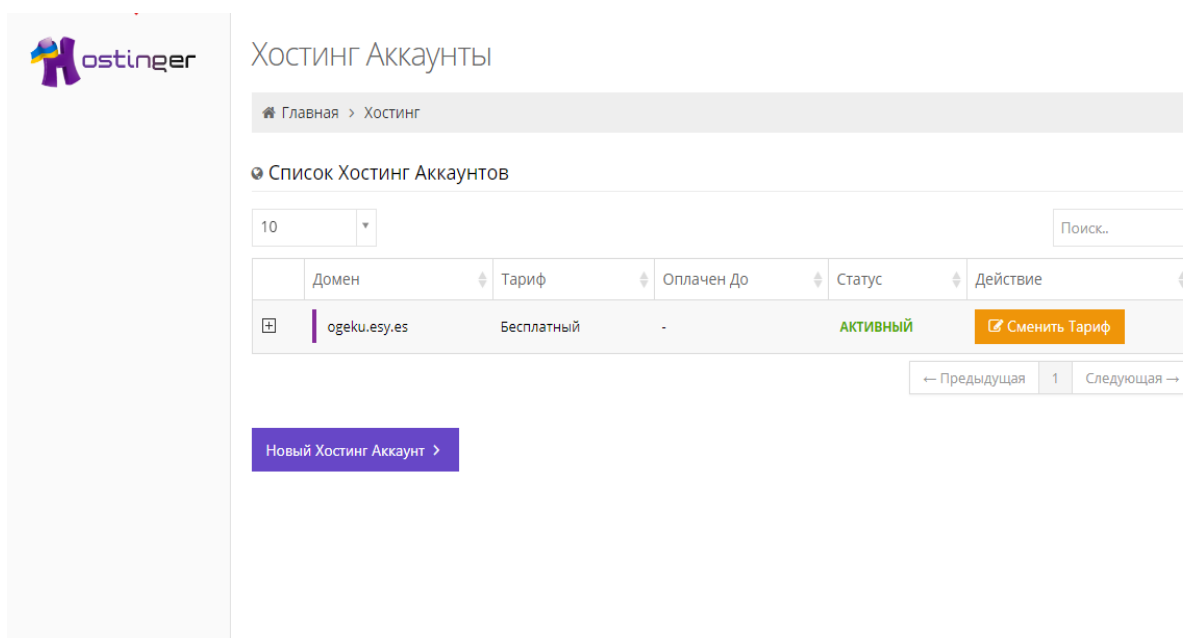


Рисунок 4.6 – Розміщення серверної частини додатка

Щоб підключитися до сайту, потрібно використовувати програму FileZilla, яка надає змогу віддалено через FTP протокол керувати сайтом, завантажувати, видаляти та створювати файли на сервері (рис. 4.7).

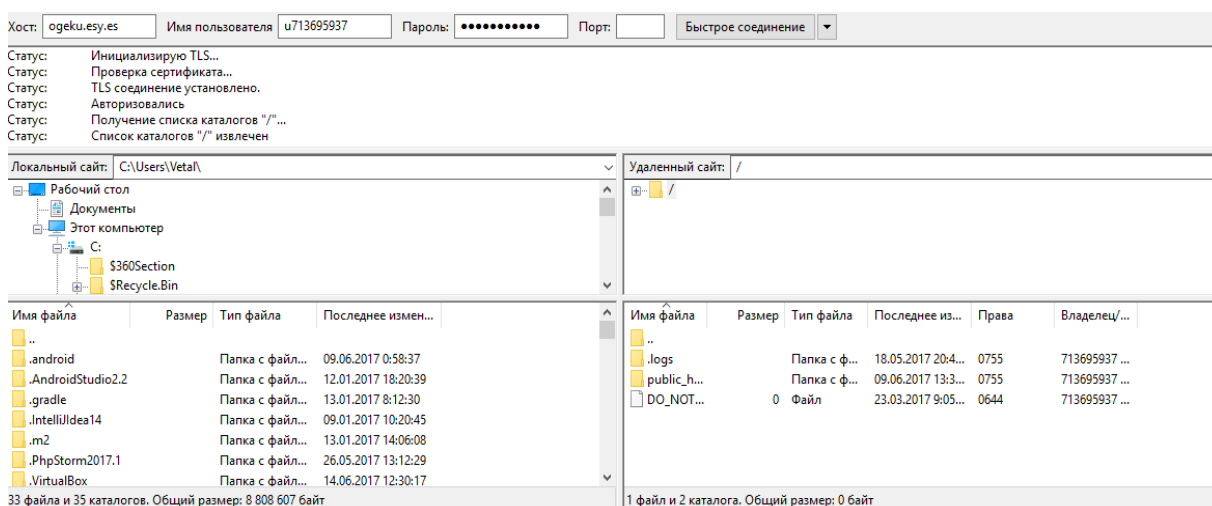


Рисунок 4.7 – Доступ до хостингу через FileZilla

4.4 Розробка клієнтської частини

4.4.1 Створення дизайну додатка

Давайте для простоти будемо 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.

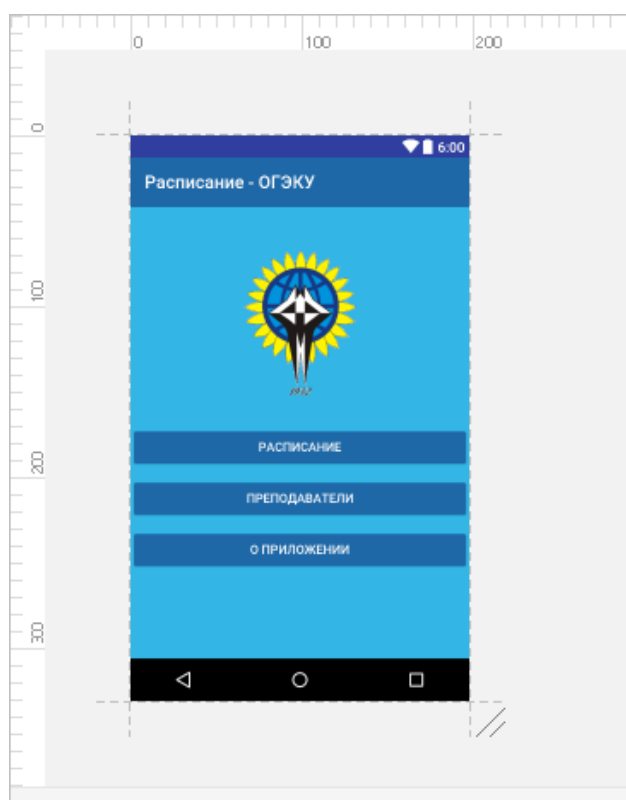


Рисунок 4.8 – Інтерфейс клієнтського додатку в процесі розробки

На главному екрані мобільного пристрою розташовані три компоненти «Button»(рис.4.8).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 відправляються на сервер де будуть обробляться (рис. 4.9).

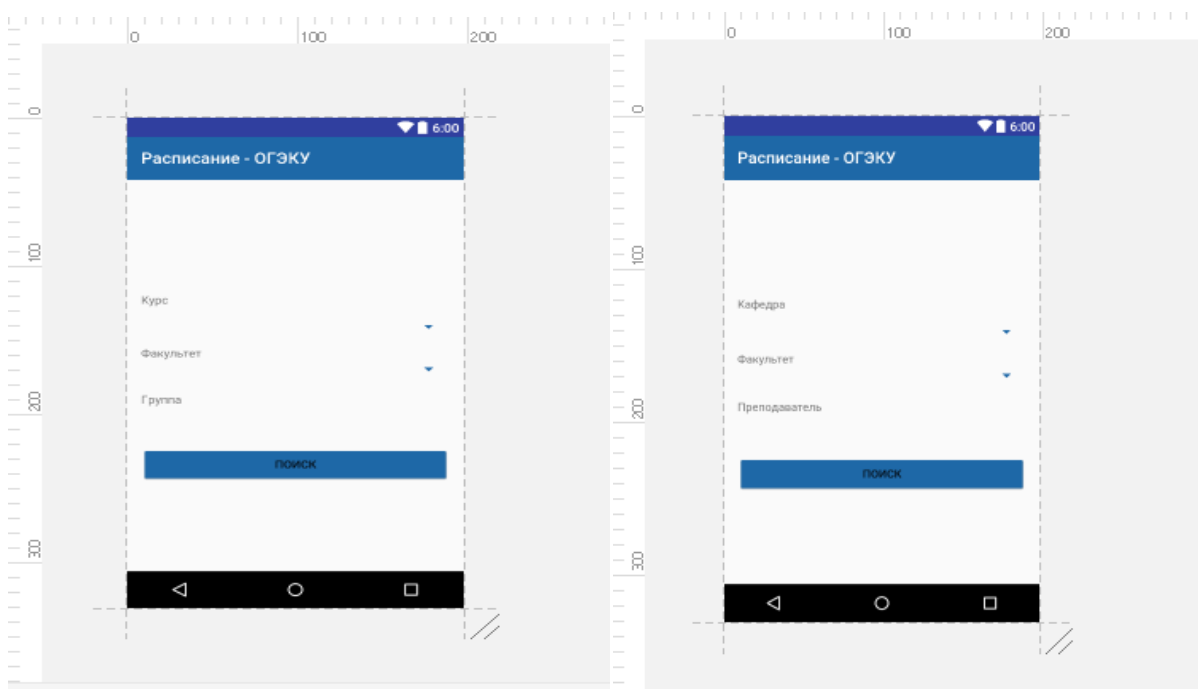


Рисунок 4.9 – Інтерфейс двох активностей в процесі розробки

4.4.2 Розробка програмної реалізації додатку

Для розробки програмної реалізації додатку, було використано різні класи та бібліотеки для Android Studio, 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 передачі параметрів на сервер через POST запит, де при оголошенні класів `DefaultHttpClient` та `HttpPost` ми вказуємо посилання та тип запиту. В той час сервер приймає дані уявлення заповненої веб-форми. (див. Додаток А)

Якщо дані будуть передані успішно, тоді прийде відповідь від сервера в форматі JSON, але його синтаксис який складається з об'єктів та масивів 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 (див. Додаток Б).

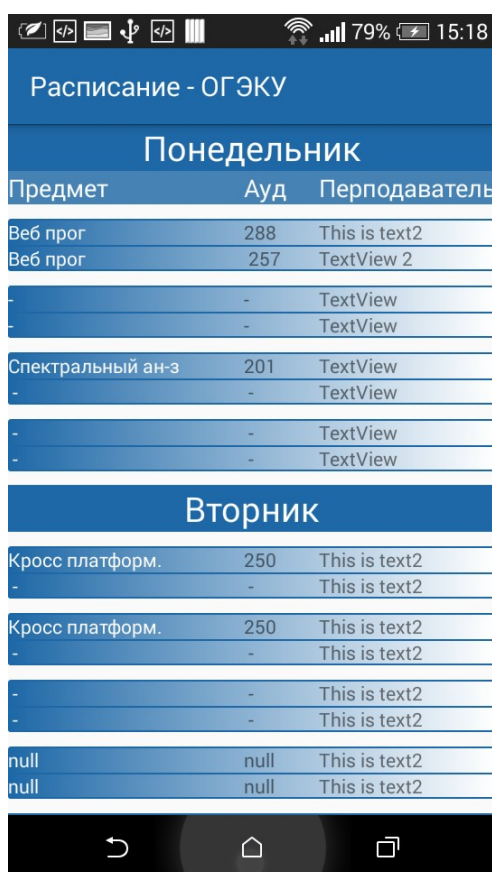


Рисунок 4.10 – Форма виведення результату

4.6 Створення MVC шаблону 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. 4.11).

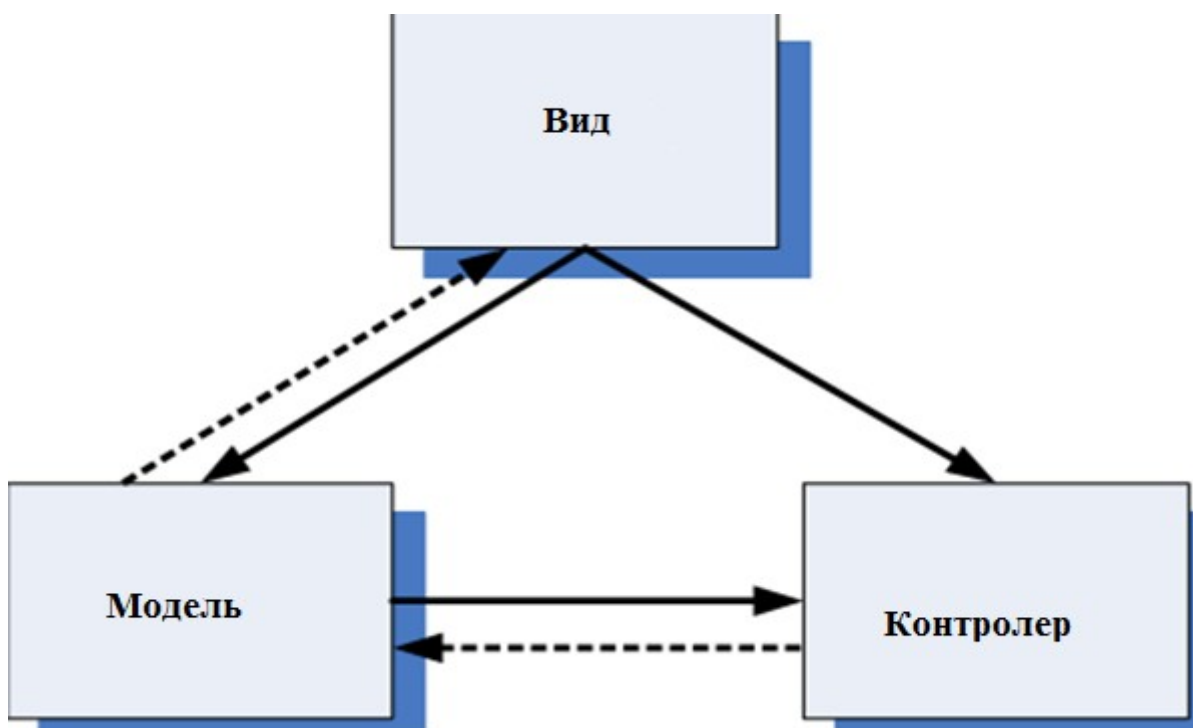


Рисунок 4.11 – Модель MVS

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 логірування (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 опціонального 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.

4.7 Застосування JQuery

Більшість PHP розробників освоювали цю мову по-старому, почавши з визначення і побудови простих PHP сторінок 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 точно, 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 JQuery такі 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 4.12).

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

Рисунок 4.12 – 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.4.13).

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

Рисунок 4.13 – Виконання JQuery скрипту

ВИСНОВКИ

В результаті дипломної роботи проведено проектування і виконана програмна реалізація серверної і клієнтської частин додатка розклад пар для Одеського Державного Екологічного Університета, 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.

Саме тому я вважаю, що зроблений мною додаток можна використати для будь-якого навчального закладу тому, що завдяки ньому можна набагато легше отримати доступ до інформації студенту або викладачеві. Великим плюсом цього додатку є те що користувачу не потрібно заходити на сайт, а досить всього один раз завантажити його на пристрій.

ПЕРЕЛІК ПОСИЛАНЬ

1. Дэвид Макфарланд. Большая книга CSS3 – CSS3. СПб.: Питер, 2016 – 251 с.
2. Бер Бибо, Иегуда Кац. jQuery. Подробное руководство по продвинутому JavaScript – СПб.: Питер, 2011. – 258 с.
3. Бретт Маклафлин. Объектно – ориентированный анализ и проектирование – С.: О’Reilly, 258 с.
4. Лиза Гарднер, Джейсон Григсби. Разработка веб-сайтов для мобильных устройств – М.: Вильямс, 2014. – 268 с.
5. Дэвид Сойер Макфарланд. Разработка сайтов на JavaScript и jQuery. Исчерпывающее руководство – С.: О’Reilly, 2012. – 587 с.
6. Эрик Фримен, Элизабет Робсон. Head First JavaScript Programming – С.: О’Reilly, 2010. – 267 с.
7. Кэти Сиерра, Берт Бейтс. Head First Java – О’Reilly, 2014. – 155 с.
8. Роберт Лафоре. Структуры данных и алгоритмы в Java – С.: О’Reilly, 2010. – 55 с.
9. А. Аллан. Программирование для мобильных устройств – СПб.: О’Reilly, 2012. – 188 с.
10. Скляр Д., Трахтенберг А.. PHP. Рецепты программирования. 3-е изд. – СПб.: Питер, 2012. – 335 с.
11. А. Аллан. Клиентская разработка для профессионалов. Node.js – СПб.: Питер, 2017. – 220 с.

Д О Д А Т К И

ДОДАТОК А

Листинг 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));
            //собираем 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();
}
}
```

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,
    },
    New int[]
    { 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 послали в MainActivity
Bundle extras = getIntent().getExtras();
//превращаем в тип строка для парсинга
String json = extras.getString(JsonURL);
//передаем в метод парсинга
JSONURL(json);
}
}
```