МIНIСТЕРСТВО ОСВIТИ I НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНIВЕРСИТЕТ

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП’ЮТЕРНИХ НАУК

СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

**ДИПЛОМНИЙ ПРОЕКТ**

**на тему:** «Створення мобільного додатку кафедри комп’ютерних наук для платформи Windows Phone 8.1»

за напрямом підготовки 6.050101 «Комп’ютерні науки»

**Виконавець роботи**  *Пракопчик Роман Ігорович*

**(**прізвище, ім’я, по батькові)

(підпис, дата)

**Дипломний проект бакалавра**

**захищений на засіданні ЕК**

**з оцінкою** « » 2017 р.

Консультант з управління

проектами к.т.н. Гайдабрус Б. В.

(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Науковий керівник к. т. н. Парфененко Ю.В. .

(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії к.т.н. Дорошенко С. О.

(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Суми-2017

ЗМІСТ

[ВСТУП 3](#_Toc483581878)

[1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 5](#_Toc483581879)

[1.1 Аналіз стану ринку мобільних сервісів 5](#_Toc483581880)

[1.2 Аналіз програмних засобів створення мобільних додатків 9](#_Toc483581881)

[1.3 Опис мобільного додатку кафедри комп’ютерних наук СумДУ 12](#_Toc483581882)

[2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ 13](#_Toc483581883)

[2.1 Мета та задачі 13](#_Toc483581884)

[2.2 Вибір методів 13](#_Toc483581885)

[2.3 Планування робіт 14](#_Toc483581886)

[3 РЕАЛІЗАЦІЯ ПРОЕКТУ 25](#_Toc483581887)

[3.1 Структура мобільного додатку 25](#_Toc483581888)

[2.2 Створення головного меню 26](#_Toc483581889)

[2.3 Створення розкладу 28](#_Toc483581890)

[2.4 Створення новин 31](#_Toc483581891)

[4 РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ 32](#_Toc483581892)

[4.1 UI інтерфейс мобільного додатку 32](#_Toc483581893)

[4.2 UX інтерфейс мобільного додатку 33](#_Toc483581894)

[ВИСНОВКИ 35](#_Toc483581895)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 36](#_Toc483581896)

[ДОДАТОК А. Технічне завдання 38](#_Toc483581897)

[ДОДАТОК Б 46](#_Toc483581913)

[ДОДАТОК В 51](#_Toc483581914)

[ДОДАТОК Г 52](#_Toc483581915)

# ВСТУП

Наука і техніка як напрямки людської діяльності існують віддавна. Однак, якщо в минулих сторіччях проблеми цих областей цікавили лише вузьке коло причетних до них інтелектуалів, а відношення до техніки було лише прикладним, то наш час висунув обидва ці явища в центр суспільної уваги, залучило до них погляди мільйонів людей.

Стрімкий розвиток техніки обумовив її всеосяжний вплив на сучасний світ. Популярність портативної електроніки та мобільних гаджетів у наш час важко заперечити. За новинами сучасного ринку мобільних телефонів дуже тяжко слідкувати, тому що швидкість створення нових технологій та розробок досягла рекордних показників.

У таких умовах дуже популярними стають мобільні додатки та програмне забезпечення і саме популярність тієї чи іншої мобільної платформи чи операційної системи визначає майбутнє розробників. Сьогодні сфера створення мобільних додатків знаходиться на високому рівні розвитку і лише динаміка розвитку популярності розумних гаджетів допоможе перейти їй на новий рівень.

Прикладні програми для систем мобільного зв’язку поділяються на:

* персональні додатки, що забезпечують доступ до інформаційних сервісів;
* системи моніторингу та системи віддаленого управління, браузери, дозволяють завантажувати сторінки у мережі Інтернет;
* мобільні ігри, мережеві багатокористувацькі гри;
* бізнес-додатки, клієнти інформаційних систем масштабу підприємства;
* комерційні веб-додатки (Інтернет-банкінг та інші види віддаленого доступу).

Розробка мобільних додатків – це довготривалий та складний процес у якому спеціалізовані додатки розробляються для невеликих портативних пристроїв, таких як сучасні смартфони або мобільні телефони. Мобільні додатки входять у комплект нового мобільного телефону або ж є адаптованими для швидкісного завантаження за допомогою спеціальних платформ розповсюдження ПО.

Метою дипломного проекту є створення мобільного додатку кафедри комп’ютерних наук для платформи Windows Phone 8.1. Додаток як об’єкт проекту повинен задовольняти всім поставленим функціональним і нефункціональним вимогам. Предметом проекту є задоволення всіх цих вимог, тобто розроблення необхідних сервісів, таких як: розклад, новини тощо.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Мобільний додаток створено спеціально для кафедри комп’ютерних наук факультету електроніки та інформаційних технологій Сумського державного університету. Розділи додатку розташовані у вірному, інтуїтивному порядку, тобто спочатку користувач знайомиться з кафедрою та спеціальностями, знайомиться з інститутом аспірантури та докторантури.

Додано розділ з розкладом СумДУ. Розклад додатку дозволяє отримати необхідну інформацію щодо пар університету швидше, з меншим використанням інтернет-трафіку, за рахунок вже деякої завантаженої інформації, та в зручному вигляді. Останніні результати розкладу тимчасово зберігаються, що дозволяє їх переглянути без інтернет з’єднання. Ця функція буде корисною як студентам так і викладачам.

Також додано розділ з новинами університету, що дозволяє бути в курсі останніх подій СумДУ.

## 1.1 Аналіз стану ринку мобільних сервісів

У наш час мобільні пристрої сильно проникли у повсякденне життя кожної людини. Кожна друга людина на Землі користується смартфоном, а кожна третя використовує його, як кишеньковий помічник.

Смартфони – це окрема категорія телефонів, які мають власну оперативну пам'ять і власний процесор. Більшість смартфонів працюють під операційними системами Windows Phone, Android, IOS.

Для того щоб смартфоном було комфортніше користуватись, розробники створюють спеціальні мобільні додатки. Вони можуть виконувати безліч функцій починаючі з різноманітних музикальних плеєрів до серйозного корпоративного програмного забезпечення. У бізнесі – мобільний додаток дозволяє налагодити зв’язок між клієнтами. А це надає можливість постійного інформування клієнтів що до нових пропозицій, тощо. Клієнт також має змогу ознайомитись з вашим бізнесом сидячі у себе вдома зі своїм смартфоном.

Користувач смартфону має можливість завантажити будь-який мобільний додаток у спеціальному магазині Windows Phone Store (рис. 1.1), AppStore або Google Play відповідно до використовуваної операційної системи, але для цього тільки потрібно створити спеціальний аккаунт, який створює вільний доступ до перегляду та завантаження контенту.

Загалом визначають чотири основні типи мобільних додатків:

* Корпоративні. Їх мета – спрощення роботи компанії, швидка передача даних між працівниками або отримання корпоративної інформації. Цільовою аудиторією таких мобільних додатків є, насамперед, працівники вашої компанії, а також реальні та потенційні клієнти та партнери. Розробка мобільних додатків такого типу простіше, для їх дизайну використовуються кольори та елементи корпоративного стилю компанії.

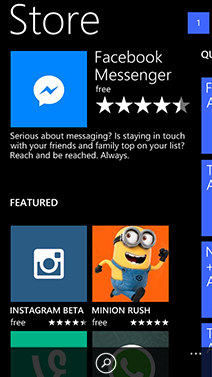


Рисунок 1.1. – Windows Phone Store

* Контентні. Це – мобільні додатки основна мета яких - надавати різного роду інформацію. Текстову або у відео або аудіоформаті. В основному, така розробка мобільних додатків здійснюється для засобів масової інформації, радіо, телеканалів чи порталів.
* Сервісні . Їх мета випливає з назви – надавати певні сервісні послуги, тобто виконувати завдання в реальному часі, які ставить користувач. Розробка мобільних додатків такого типу - складна, адже вони повинні працювати, як годинник: починаючи від калькулятора або будильника і закінчуючи програмами для роботи з великими обсягами тексту або графіки.
* Ігрові. Основне завдання таких мобільних додатків – розважати, але це не означає, що вона єдина. Розробка мобільних додатків такого типу ускладнюється необхідністю вдало розмістити контекстну або пряму рекламу.

На сьогоднішній час ринок мобільних послуг зростає великими темпами, але щоб такий прогрес був і на далі треба виділити основні передумови для його зростання, а саме: використання соціальних мереж, зростання рівня використання планшетних пристроїв, розвиток мобільного банкінгу, електронної комерції і платіжних систем, геопозиціювання, впровадження LTE або 3G тощо.

Перешкодами зростання ринку мобільних додатків є: неінформованість користувачів, труднощі сплати, низький рівень ІТ готовності населення та інші.

Додатки, які вже стали популярними і займають топові місця в “Windows Phone Store” мають явну перевагу над новими додатками, але крім грошей розробникам цікаво знати детальну інформацію про завантаження, коли, де, на якому пристрої та якість обслуговування додатком користувача, для цього існує спеціальна система аналізу для збору та обробки такої інформації, перед тим як назвати такі сервіси, треба розуміти, які базові функції вона повинна мати, а саме:

• Визначення джерел трафіку (хоча б відслідковування рекламних систем).

• Воронки.

• Повноцінний когортний аналіз.

Когортний аналіз – це виділення певної групи користувачів (когорти) та аналіз її поведінки в часі. Користувачів в когорті об'єднує вчинення дії на певному проміжку часу.

Ідеальний інструмент окрім базових функцій повинен мати наступний функціонал:

• Можливість А/Б тестування в мобільному додатку.

• Можливість відправки таргетувальні пуш-повідомлення.

• Повноцінне API з можливістю вивантажувати сирі данні.

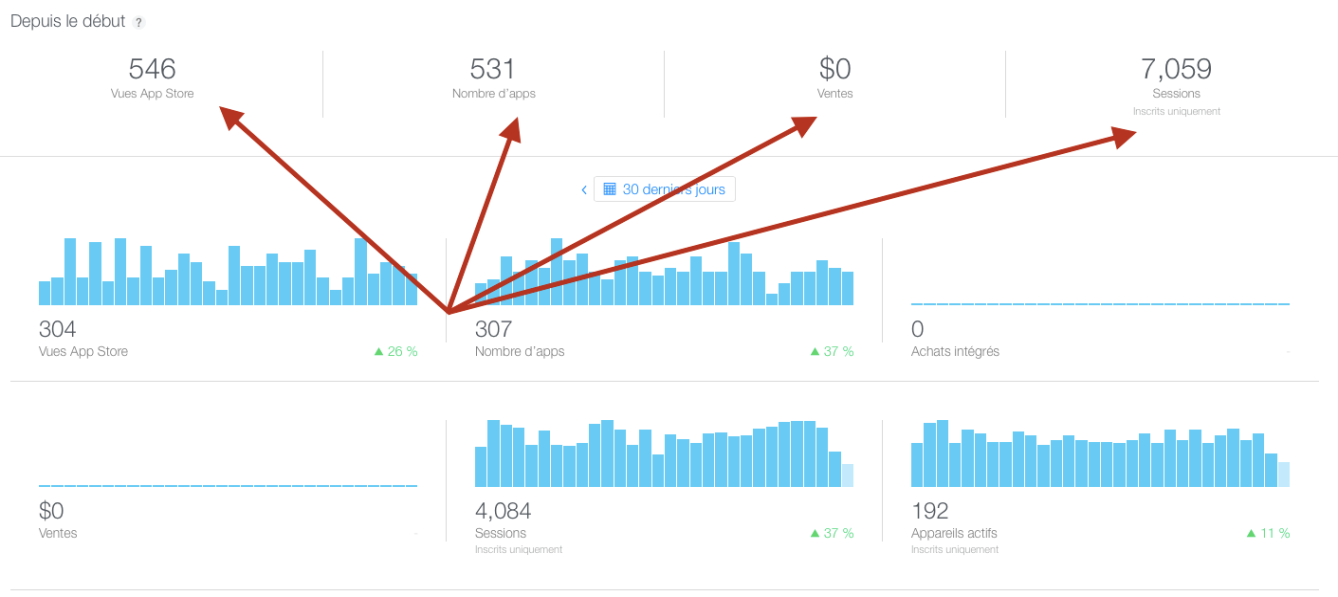
****

Рисунок 1.2. – Детальна система аналізу

На жаль, жоден з існуючих на ринку інструментів не вміє робити все це відразу (хоча деякі близькі, але дуже дорогі), що створює ряд складнощів в аналітиці мобільних додатків, тим паче сервіс для збору аналітики існує і самі сучасніші із них це – Яндекс метрика, Kissmetrics, Kontagent тощо.

## 1.2 Аналіз програмних засобів створення мобільних додатків

Для розробки додатків для Windows Phone компанія створила свій пакет інструментів для розробки – Windows Phone Software Development Kit (SDK). Корпорація “Microsoft” до останнього тримала SDK для “Windows Phone 8” під замком, але тепер його можна вільно завантажити на офіційному сайті компанії. Відсутність доступу до SDK призвело розробників в деяке здивування, адже в “Windows Phone” завжди була проблема з вибором якісних додатків. В рамках Build Conference компанія “Microsoft” зняла покрови з SDK, і розробники нарешті можуть приступити до створення нових ігор і додатків для “Windows Phone 8.1”.

Для розробки додатку було використане середовище розробки Microsoft Visual Studio 2013 Update 5, яке розповсюджується та підтримується компанією Microsoft. Пакет програм Visual Studio є пропрієтарним продуктом, проте існує безкоштовна версія Community.

Для створення макету сторінок використовувалась декларативна мова розмітки XAML. З точки зору моделі програмування .NET Framework ця мова спрощує створення користувацького інтерфейсу для програми .NET Framework. Можна створити видимі елементи інтерфейсу користувача в декларативній розмітці XAML, а потім відокремити визначення користувача інтерфейсу від логіки часу виконання, використовуючи файли коду програмної частини, приєднані до розмітки за допомогою визначень поділюваних класів. Мова XAML безпосередньо представляє створення екземплярів об'єктів в конкретному наборі резервних типів, визначених у збірках. У цьому полягає її відмінність від більшості інших мов розмітки, які, як правило, являють собою інтерпретовані мови без прямого зв'язку з системою резервних типів. Мова XAML забезпечує робочий процес, що дозволяє декільком учасникам розробляти користувацький інтерфейс і логіку програми, використовуючи потенційно різні засоби.

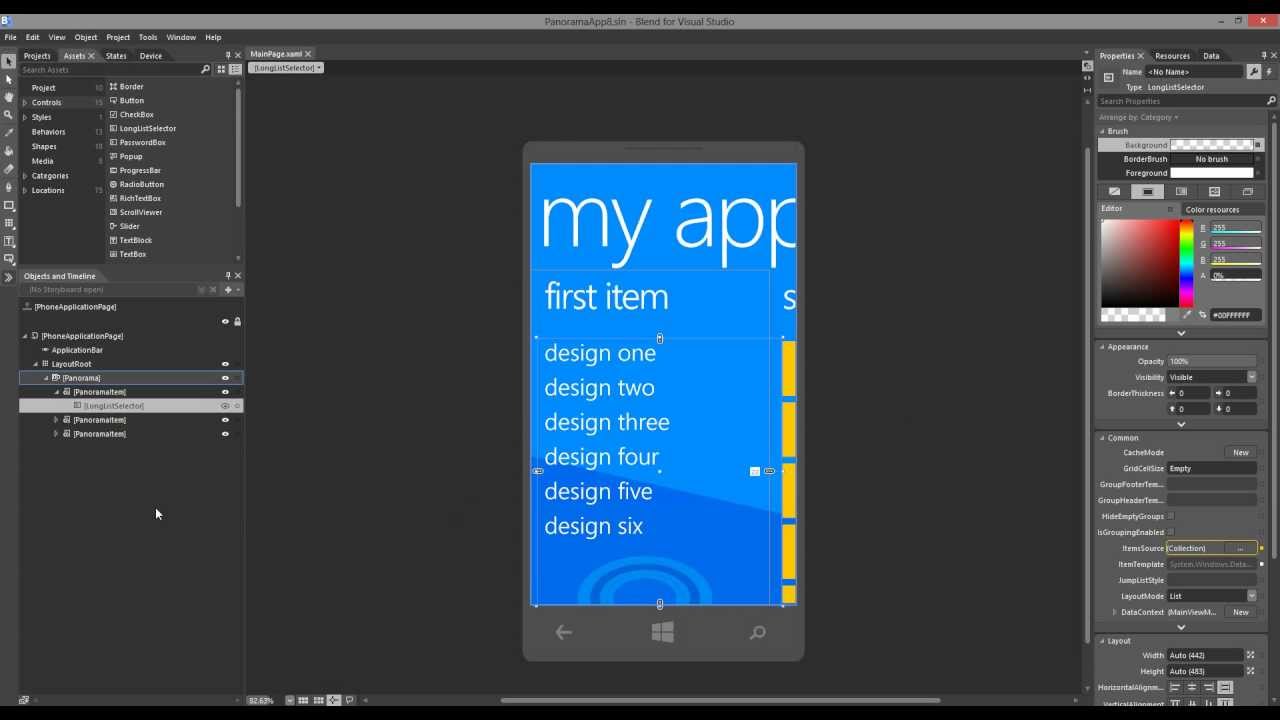


Рисунок 1.3. – Скріншот Expression Blend 2013

Для проектування інтерфейсу був використаний Expression Blend, призначений для розробки дизайну додатків, який входить до складу пакету Microsoft Expression Studio. Цей WYSIWYG - редактор XAML-сторінок дозволяє реалізувати найсміливіші дизайнерські ідеї.

Для створення логіки мобільного додатку була використана об'єктно-орієнтована мова програмування C#. Головною особливістю мови є її орієнтованість на платформу .NET. C# приховує від розробника багато незначних технічних деталей, включаючи операції пов’язані з упаковкою і розпаковкою типів, ініціалізації змінних, збиранням сміття. Це дозволяє C# програмісту приділяти більше уваги змістовій частині коду.

Однією з найбільш значних проблем проектування є складність. Чим більша і складніша програмна система, тим важливіше розбити її на невеликі, чітко окреслені частини. Щоб впоратися зі складністю, необхідно абстрагуватися від деталей. У цьому сенсі ООП являє собою дуже зручну методику проектування.

Класи дозволяють проводити конструювання з корисних компонентів, що володіють простими інструментами, що дозволяє абстрагуватися від деталей реалізації.

Дані та операції над ними утворюють певну сутність, і вони не розносяться по всій програмі, як нерідко буває у випадку процедурного програмування, а описуються разом. Локалізація коду і даних покращує наочність і зручність супроводу програмного забезпечення.

Інкапсуляція дозволяє привнести властивість модульності, що полегшує розпаралелювання виконання завдання між кількома виконавцями і оновлення версій окремих компонентів.

ООП дає можливість створювати системи, що розгортаються. Це одна з основних переваг ООП, і саме вона відрізняє даний підхід від традиційних методів програмування. Розширюваність означає, що існуючу систему можна змусити працювати з новими компонентами, причому без внесення до неї будь-яких змін. Компоненти можуть бути додані на етапі виконання програми.

Часто багаторазового використання програмного забезпечення не вдається домогтися через те, що існуючі компоненти вже не відповідають новим вимогам. ООП допомагає цього досягти без порушення роботи вже наявних компонентів, що дозволяє витягти максимум з багаторазового використання компонентів.

* Скорочується час на розробку, який може бути віддано іншим завданням.
* Компоненти багаторазового використання звичайно містять набагато менше помилок, ніж знову розроблені, адже вони вже не раз піддавалися перевірці.
* Коли якийсь компонент використовується відразу декількома клієнтами, поліпшення, що вносяться до його код, одночасно надають позитивний вплив і на безліч працюючих з ним програм.
* Якщо програма спирається на стандартні компоненти, її структура і призначений для користувача інтерфейс стають більш уніфікованими, що полегшує її розуміння і спрощує використання.

Використання об’єктно-орієнтованого підходу дозволяє значно зменшити кількість помилок за рахунок інкапсуляції та прискорити налагодження програми за рахунок приналежності методів до свого класу. Набагато прискорюється модифікація програми.

## 1.3 Опис мобільного додатку кафедри комп’ютерних наук СумДУ

Мобільний додаток створений спеціально для додатку кафедри комп’ютерних наук СумДУ факультету електроніки та інформаційних технологій Сумського державного університету. Розділи додатку розташовані у вірному, інтуїтивному порядку, тобто спочатку абітурієнт знайомиться з факультетом, кафедрами та спеціальностями, знайомиться з інститутом аспірантури та докторантури.

Додано розділ з розкладом СумДУ. Розклад додатку дозволяє отримати необхідну інформацію щодо пар університету швидше, з меншим використанням інтернет-трафіку, за рахунок вже деякої завантаженої інформації, та в зручному вигляді. Останніні результати розкладу тимчасово зберігаються, що дозволяє їх переглянути без інтернет з’єднання. Ця функція буде корисною як студентам так і викладачам.

Також додано розділ з новинами університету, що дозволяє бути в курсі останніх подій СумДУ.

# 2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

## 2.1 Мета та задачі

Метою виконання даної роботи є розробка готового до використання мобільного додатку для Windows Phone. Для реалізації поставленого завдання були виділені наступні задачі дослідження:

* аналіз аналогів;
* розроблення моделі програмного продукту ;
* розробка додатку;
  + розробка інтерфейсу;
  + розробка модулю новин університету;
  + розробка модулю розкладу;
  + розробка інших модулів;
* відладка та тестування;
* оформлення супроводжуючої документації.

## 2.2 Вибір методів

Для вирішення поставленої задачі будуть використовуватися методи аналізу, моделювання та розробки.

Під час аналізу були проаналізовані існуючі програмні аналоги: функціонал, дизайн інтерфейсу та основні технології, які використовувалися під час їх створення.

Розроблюваний програмний модулю буде програмуватися в середовищі програмування Microsoft Visual Studio 2013. Для розробки дизайну використаний Expression Blend 2013.

## 2.3 Планування робіт

2.3.1 Ідентифікація мети ІТ-проекту. Мета проекту: розробити мобільний додаток для мобільної платформи Windows Phone 8.1, для того щоб полегшити доступ до корисної інформації, максимально ознайомити користувача з важливою інформацією, щодо кафедри комп’ютерних наук, також ознайомити з навчальним закладом СумДУ, з персоналом кафедри. Для створення проекту є всі необхідні дані та інструментарій для розробки програмних продуктів.

Мобільний додаток має можливість переглядати потрібну документацію через спеціальні розділи. Для того, щоб користувач не мав проблем з використанням мобільного додатка, всі розділи, кнопки керування, створені таким чином, щоб їх було легко використовувати навідь самим звичайним користувачам. Кожна кнопка має коментарі.

Подібний проект є унікальним тому, що не кожен навчальний заклад має подібний мобільний додаток. Досліджуючі IT ринок в Україні можна зробити висновок, що подібні мобільні додатки існують але ти, які направлені на наукову діальяність - дуже мало. Прикладом є мобільний проект КПІ у м. Київ де розробники створили мобільний додаток з невеликими функціями на операційніий системі Android., але проект фактично «загинув» тому, що розробники не оновлювали контент. Проту проект додатку постійно оновлюється, проект є якісним та корисним. І подальшому проект буде розвиватись завдякі не тільки розробникам, а й кафедрі комп’ютерних наук та СумДУ загалом.

2.3.2 Управління вимогами. мобільний додаток повинен відповідати поставленим критеріям, які були висунуто у якості функціональних та не функціональних вимог:

* розмір додатку не повинен перевищувати 35 мегабайт;
* мобільний додаток повинен бути безкоштовним;
* мобільний додаток повинен бути без реклами, яка б псувала або заважали комфортній роботі;
* інтерфейс повинен бути інтуїтивно-зрозумілим, повинен підходити для кожного користувача не залежно від полу, року та здібностей;
* усі тексти повинні бути читабельними, кольори - не навантажувати очі як при денній так і вечірній годині;
* надати користувачу доступ до інформації у зручному та правильному контексті до всіх розділів;
* надати користувачу детальний опис дистанційного, заочного та денного форм навчання в СумДУ, ознайомити користувачів з методом навчання на дистанційній формі;
* надавати технічну підтримку усім тим хто її потребує, у мобільному додатку створити усі благо приємні умови для контакту з технічною підтримкою;
* надати можливість перегляду розкладу СумДУ.
* відображення новин університету;
* надання доступу до корисної інформації;
* для моделювання використовувати методологію UML та IDEF0;
* для програмування використовувати Microsoft Visual Studio;
* використовувати мову c#;
* надати можливість користування додатком під сучасною операційною системою Windows Phone 8.1 (RT).

2.3.3 Планування змісту структури робіт ІТ-проекту (WBS). WBS — це ієрархічна структура, побудована з метою логічного розподілу усіх робіт з виконання проекту і подана у графічному вигляді. Це сукупність декількох рівнів, кожний з яких формується в результаті розподілу роботи попереднього рівня на її складові. Елементом найнижчого рівня є група робіт, або так званий робочий пакет (work package)[10].

Схема WBS представлена в додатку Б.(Рис.Б.1)

2.3.4 Планування структури організації, для впровадження готового проекту (OBS). Організаційна структура проекту (ОBS) стосується тільки внутрішньої організаційної структури проекту і не зачіпає відносин проектних груп чи учасників з батьківськими організаціями. Будується ОBS аналогічно робочій структурі, а саме:

* на першому рівні відображається організаційна структура як єдиний елемент;
* на другому і нижчих рівнях триває поділ структури на основні організаційні елементи.

Цей процес повторюється до найнижчого рівня — базових робочих груп (змішаних цільових або функціональних), а при реалізації малих проектів — до окремих виконавців. Обсяг робіт для цих найнижчих організаційних рівнів являє собою найнижчі елементи WBS, кожний з яких можна планувати і контролювати як окремі одиниці. Таке саме правило чинне для створення OBS. Кількість рівнів залежить від розміру проекту[11].

Схема OBS представлена в додатку Б.(Рис.Б.2)

2.3.5 Побудова матриці відповідальності (виконавців пакетів робіт). Матриця відповідальності — це графічне визначення того, хто що робить по проекту, або закріплення відповідальності членів проектної команди за виконання окремих елементів проекту і відносини із залученими партнерами. [12]

Матриця відповідальності наведена у додатку (табл. В) та будується на підставі WBS (рис. Б.1) та OBS (рис. Б.2).

2.3.6 Розробка PDM-мережі (розгорнутий вигляд мережевих діаграм Ганта). PDM-мережі складаються з двох типів елементів: робіт, які розташовані у вузлах, та стрілок, які вказують логічні взаємозв’язки між роботами проекту. PDM-мережі наведені в додатках Б.3 та Б.4

2.3.7 Побудова календарного графіку виконання ІТ-проекту (включаючи побудову часткових мережевих моделей у вигляді діаграм Ганта). Діаграма Ганта ( англ. Gantt chart, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами. [13]. Діаграма наведена в додатку Б.5.

Таблиця 2.2 – Детальний перелік робіт

|  |  |
| --- | --- |
| **№** | **Розробка ПП** |
| 1 | Аналіз існуючого ПП для інших платформ |
| 1.1 | Аналіз функціоналу |
| 1.2 | Аналіз інтерфейсу |
| 2 | Розробка ТЗ |
| 2.1 | Опис вимог до ПП |
| 2.1.1 | Функціональні вимоги |
| 2.1.2 | Вимоги до надійності |
| 2.1.3 | Опис умов експлуатації |
| 2.2 | Опис розроблюваного ПП |
| 2.3 | Опис вимог до документації |
| 3 | Розробка моделі |
| 3.1 | Побудова use case diagrams |
| 3.1.1 | Визначення акторів |
| 3.1.2 | Визначення варіантів використання |
| 3.1.3 | Встановлення взаємодії акторів та ВВ |
| 3.2 | Побудова sequence diagrams |
| 4 | Розробка ПП |
| 4.1 | Аналіз мов програмування |
| 4.2 | Розробка інтерфейсу |
| 4.2.1 | Створення pages |
| 4.2.2 | Додавання елементів інтерфейсу |
| 4.3 | Програмування |
| 5 | Тестування |
| 5.1 | Аналіз результатів роботи ПП |
| 5.2 | Перевірка та виправлення помилок |
| 5.2.1 | Виявлення помилок |
| 5.2.2 | Виправлення помилок |
| 6 | Розробка необхідної документації |
| 6.1 | Розробка керівництво користувача |

2.3.8 Розрахунок бюджету проекту. Бюджет проекту повинен враховувати:

* собівартість виконання робіт;
* резерв на відомі ризики (виявлення на етапі планування ризиків;
* базову вартість;
* резерв на ризики, що не були виявлені на етапі планування.

Першочергові витрати – це витрати на закупівлю робочого місця, а саме програмне та апаратне забезпечення для розробки. Ці витрати мають одноразовий характер.

Загальна вартість апаратного забезпечення становить 8000 грн.

Вартість ліцензійного Visual Studio 2013 – 2000$.

Включити у витрати можна також вартість електроенергії.

Згідно актуальної ціни за електроенергію в Україні на роботу апаратного забезпечення буде затрачено близько 50 грн, враховуючи зовнішнє освітлення.

Такий фактор як орендна плата за приміщення для роботи не є оплачуваним, адже розробник працює у себе вдома.

Через не закладені ризики, які можуть вплинути на розробку проекту, бюджет може змінитися.

2.3.9 Планування можливих відхилень (графіки поставок, укладання контрактів тощо). Фактори, які можуть призвести до відхилень поділяються на декілька групи:

* часові (недотримання календарного плану);
* технічні (відмова техніки);
* фінансові (перевищення бюджетних розтрат проекту).

Якщо дані фактори проявляться у подальшій розробці проекту, то вони будуть подолані наступним чином:

* часові – за рахунок найняття нового персоналу та використання часового резерву;
* технічні – за рахунок залучення резервних коштів на покупку потрібного технічного забезпечення;
* фінансові – за рахунок залучення нових інвестицій.

2.3.10 Проведення оцінки ефективності проектних дій (NPV - чиста поточна вартість, IRR – ставка внутрішньої прибутковості). Для того щоб визначити ефективність проектних дій необхідно розрахувати [14]:

* чисту поточну вартість (NPV);
* ставку внутрішньої прибутковості (IRR).

Для початку розрахуємо дисконтний грошовий потік за формулою:

(2.1)

потік платежів після провадження системи: за перший рік прибуток становитиме , за другий рік , за третій ;

– ставка дисконтування, яка дорівнює – 0.2;

Тепер розрахуємо чисту поточну вартість:

(2.2)

розмір інвестиції, дорівнює –

Тепер розрахуємо для ставки дисконтування :

;

;

Тепер розрахуємо ставку внутрішньої прибутковості :

(2.3)

Оскільки , то проект можна вважати прийнятним.

2.3.11 Управління ризиками ІТ-проекту. Класифікація ризиків проекту зазвичай проводиться на етапі їх аналізу за такими критеріями як імовірність виникнення та ступінь впливу в рамках на проект. Але досить доцільним є створення початкової класифікації ризиків ІТ-проектів, яка слугуватиме вхідною інформацією для процесу їх ідентифікації. В даному напрямку вже проведено немало аналітичної роботи вітчизняними та зарубіжними спеціалістами. Серед всіх опублікованих варіантів класифікації ризиків як в загальному плані так і зокрема в ІТ-індустрії є досить багато спільного. Перш за все всі ризики прийнято розділяти за джерелом виникнення на внутрішні та зовнішні. Внутрішні ризики залежать безпосередньо від внутрішнього середовища проекту та мають відносно високий рівень керованості. Зовнішні ризики виникають незалежно від проектної діяльності, можуть враховуватися учасниками проекту та бути частково керованими. Зовнішні ризики можна якісно розділити на дві підгрупи – передбачувані та непередбачувані.

Таблиця 2.3 – Ризики проекту

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Ризик** | **Ефект** | **Вирішення** |
| 1. | Несправність обладнання. Під час роботи над проектом може відмовити технічне забезпечення (ноутбук або його компоненти). | Витрата часу | Залучення (резервування) обладнання або грошей, збереження проміжних результатів проекту |
| 2. | Відмова програмного забезпечення. | Витрата часу | Збереження проміжних результатів проекту |
| 3. | Проблеми з електропостачанням. | Витрата часу та якості | Забезпечення електрогенераторами |
| 4. | Нестача часу для реалізації всього запланованого функціоналу. | Невиконання ТЗ | Резервування часу |
| 5. | Нечітка постановка ТЗ. | Невиконання ТЗ | Детальний опис ТЗ, обговорення «проблемних» компонентів. |
| 6. | Зміна ТЗ | Витрата часу та якості | Укладання договору щодо штрафів при зміні ТЗ. |
| 7. | Хвороба розробника | Витрата часу та якості | Залучення (резервування) додаткового розробника |
| 8. | Інші (непроектні) проблеми персоналу, залученого до проекту | Витрата часу та якості | Залучення (резервування) додаткового персоналу |
| 9. | Критичні баги, які унеможливлюють подальшу роботу над проектом | Витрата часу | Детальне планування робіт проекту |
| 10. | Державна нестабільність (початок революції, державного перевороту ,війни тощо) | Неможливість персоналу приймати подальшу участь у проекті | Залучення (резервування) обладнання та персоналу на територіально віддалених регіонах. |
| 11. | Природні катастрофи (урагани, повені тощо) | Неможливість персоналу приймати подальшу участь у проекті | Залучення (резервування) обладнання та персоналу на територіально віддалених регіонах. |

**Розрахунок індексів ризиків**

Ймовірність виникнення:

1. Слабоймовірні
2. Малоймовірні
3. Ймовірні
4. Досить ймовірні
5. Майже можливі

Величина втрат:

1. Мінімальні
2. Низькі
3. Середні
4. Високі
5. Максимальні

Ступінь впливу:

1. Ігноровані (R=(1:4))
2. Незначні (R=(5:8))
3. Помірні (R=(9:10))
4. Суттєві (R=(12:16))
5. Критичні (R=(20:25))

Рівень ризику:

1. Прийнятні (R=(1:4))
2. Виправдані (R=(5:10))
3. Недопустимі (R=(12:25))

Оцінка індексу ризику:

де R — індекс ризику (бали); — вірогідність виникнення ризиків згідно класифікації (бали); — величина втрат відповідно до класифікації ризику (бали).

Таблиця 2.4 – Розрахунок ступеню впливу ризиків проекту

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Ризик** | **Ймовірність** | **Втрати** | **Ступінь впливу (R)** |
| 1. | Несправність обладнання. Під час роботи над проектом може відмовити технічне забезпечення (ноутбук або його компоненти). | 1 | 4 | 4 |
| 2. | Відмова програмного забезпечення. | 2 | 4 | 8 |
| 3. | Проблеми з електропостачанням. | 2 | 4 | 8 |
| 4. | Нестача часу для реалізації всього запланованого функціоналу. | 3 | 4 | 12 |
| 5. | Нечітка постановка ТЗ. | 2 | 3 | 6 |
| 6. | Зміна ТЗ | 1 | 4 | 4 |
| 7. | Хвороба розробника | 2 | 2 | 4 |
| 8. | Інші (непроектні) проблеми персоналу, залученого до проекту | 3 | 3 | 9 |
| 9. | Критичні баги, які унеможливлюють подальшу роботу над проектом | 3 | 3 | 9 |
| 10. | Державна нестабільність (початок революції, державного перевороту ,війни тощо) | 1 | 5 | 5 |
| 11. | Природні катастрофи (урагани, повені тощо) | 1 | 5 | 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ймовірність | 5 | 5 | 10 | 15 | 20 | 25 |
| 4 | 4 | 8 | 12 | 16 | 20 |
| 3 | 3 | 6 | 9 | 12 | 15 |
| 2 | 2 | 4 | 6 | 8 | 10 |
| 1 | 1 | 2 | 3 | 4 | 5 |
|  | 1 | 2 | 3 | 4 | 5 |
|  | Втрати | | | | | |

Рисунок 2.4 – Матриця «Ймовірність – Втрати»

Таблиця 2.5 – Класифікація за рівнем ризику

|  |  |
| --- | --- |
| **R** | **Опис ідентифікованого ризику** |
| **Ігноровані (R=(1:4))** | |
| 4 | Несправність обладнання |
| 4 | Хвороба розробника |
| 4 | Зміна ТЗ |
| **Виправдані (R=(5:16))** | |
| 5 | Державна нестабільність |
| 5 | Природні катастрофи |
| 6 | Нечітка постановка ТЗ |
| 8 | Відмова програмного забезпечення |
| 8 | Проблеми з електропостачанням |
| 9 | Інші (непроектні) проблеми персоналу, залученого до проекту |
| 9 | Критичні баги, які унеможливлюють подальшу роботу над проектом |
| 12 | Нестача часу для реалізації всього запланованого функціоналу |
| **Недопустимі (R=(17:25))** | |
| – | – |

Таблиця 2.6 – Класифікація за ступенем впливу

|  |  |
| --- | --- |
| **R** | **Опис ідентифікованого ризику** |
| **Прийнятні (R=(1:4))** | |
| 4 | Несправність обладнання |
| 4 | Хвороба розробника |
| 4 | Зміна ТЗ |
| **Незначні (R=(5:8))** | |
| 5 | Державна нестабільність |
| 5 | Природні катастрофи |
| 6 | Нечітка постановка ТЗ |
| 8 | Відмова програмного забезпечення |
| 8 | Проблеми з електропостачанням |
| **Помірні (R=(9:10))** | |
| 9 | Інші (непроектні) проблеми персоналу, залученого до проекту |
| 9 | Критичні баги, які унеможливлюють подальшу роботу над проектом |
| **Суттєві (R=(12:16))** | |
| 12 | Нестача часу для реалізації всього запланованого функціоналу |
| **Критичні (R=(20:25))** | |
| – | – |

# 3 РЕАЛІЗАЦІЯ ПРОЕКТУ

## 3.1 Структура мобільного додатку

Версію додатку для платформи Windows Phone створено з використанням нативних інструментів розробки. Структура додатку складається з власної архітектури відповідно сайту кафедри комп’ютерних наук та кількох додаткових розділів.

Для сворення додатку було використано архітектуру MVVM (рис. 3.1). MVVM полегшує відокремлення розробки графічного інтерфейсу від

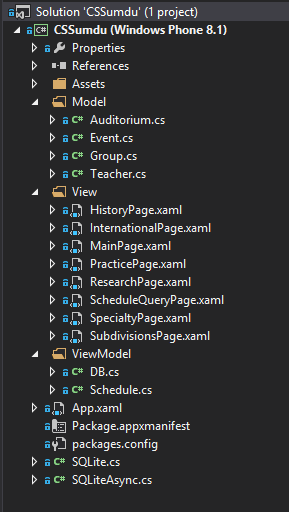


Рисунок 3.1. – Структура додатку

розробки бізнес логіки, відомої як модель (можна також сказати, що це відокремлення представлення від моделі). Модель представлення є частиною, яка відповідає за перетворення даних для їх подальшої підтримки і використання. З цієї точки зору, модель представлення більше схожа на модель, ніж на представлення і оброблює більшість, якщо не всю, логіку відображення даних.

Мобільний додаток складається з п’яти інформаційних розділів та двох сервісних (рис. 3.2), які розміщені на основному екрані.

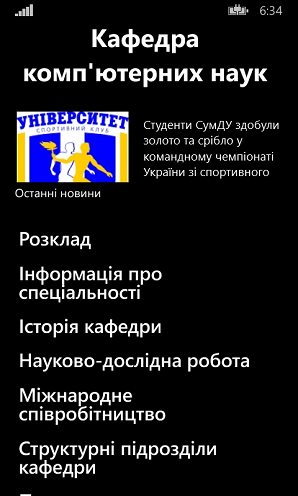


Рисунок 3.2. – Головне меню додатку

## 2.2 Створення головного меню

Головне меню – це доступ до контенту, який знаходиться у мобільному додатку. Головне меню додатку складається як вже це раніше було сказано з семи основних розділів, але це лише структура, проте як же розроблене головне меню?

Спочатку був створений XAML-макет сторінки головного меню. Головним елементом є ListView, розділи є ListView елементами. Кожна сторінка (page) Windows Phone додатку має асоційований з нею .cs файл з кодом на C#, для створення логіки. Отже для створення пунктів головного меню створюємо ObserverableCollection, наповнюємо колекцію необхідними даними та «зв’язуємо» з створеним раніше ListView у XAML. Далі створюємо обробник події при натисканні на будь-який елемент ListView. При спрацьовуванні він визначає, який з пунктів меню був обраний та викликає необхідну нову сторінку.

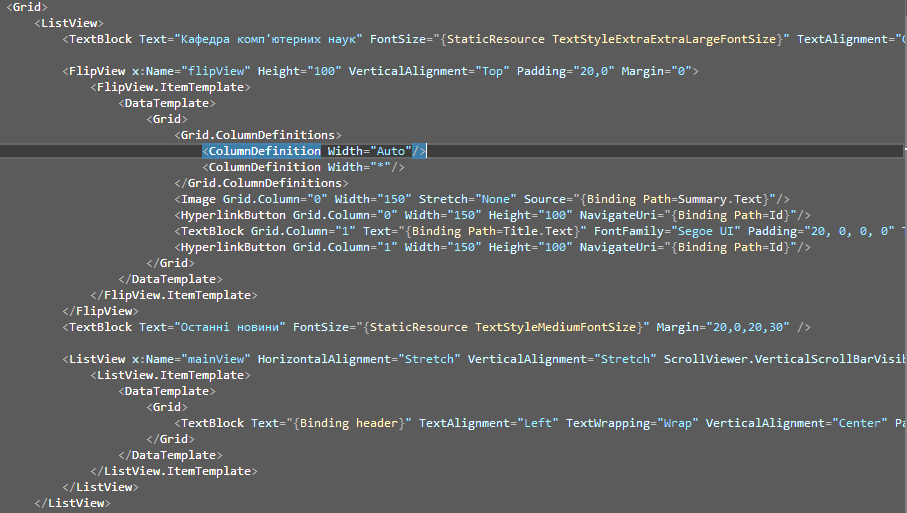


Рисунок 3.3 – XAML-макет сторінки головного меню

Крім того при навігації до головного меню відбувається спроба завантажити списки викладачів, груп і аудиторій, а також спроба завантаження та відображення новин. (рис. 3.5)

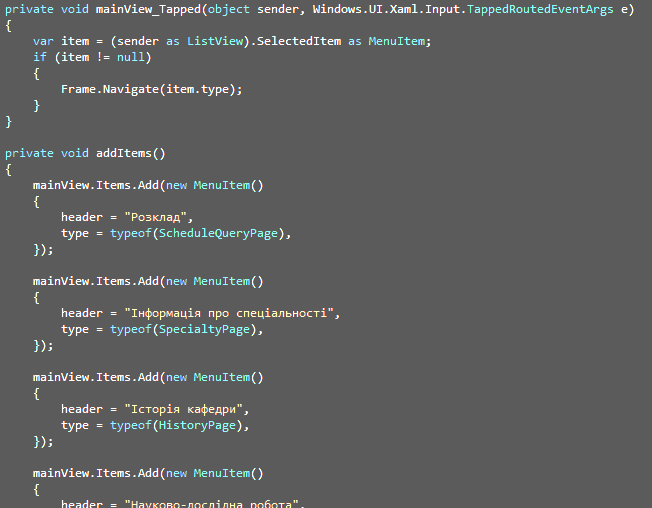


Рисунок 3.4 – Створення пунктів меню та оброблювача події

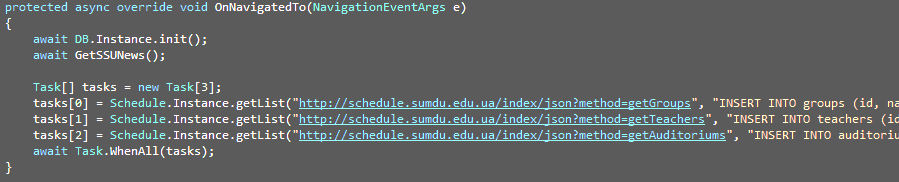


Рисунок 3.5 – Завантаження новин та списків

## 2.3 Створення розкладу

Однією з функціональних вимог проекту є створення розкладу СумДУ. Сервер розкладу має API (Application Programming Interface), за допомогою якого можна отримати інформацію про пари у форматі JSON.



Рисунок 3.6 – Скріншот розкладу

XAML-розмітка цієї сторінки має два DatePicker, для вибору дат початку і закінчення розкладу та три AutoSuggestBox для вибору групи або/та викладача або/та аудиторії. Списки викладачів, груп та аудиторій напередодні завантажуються з сервера, створюється масиви з необхідними даними та записується до локальної бази данних SQLite.

Особливістю SQLite є те, що вона не використовує парадигму клієнт-сервер, тобто рушій SQLite не є окремим процесом, з яким взаємодіє застосунок, а надає бібліотеку, з якою програма компілюється і рушій стає складовою частиною програми. Таким чином, як протокол обміну використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується застосунок. Простота реалізації досягається за рахунок того, що перед початком виконання транзакції весь файл, що зберігає базу даних, блокується; ACID-функції досягаються зокрема за рахунок створення файлу-журналу.

Елементи AutoSuggestBox-ів та дані останнього запиту зчитуються з бази даних, що дозволяє скоротити використання Інтернет-трафіку та збільшити швидкодію, що в свою чергу є корисним для смартфонів.

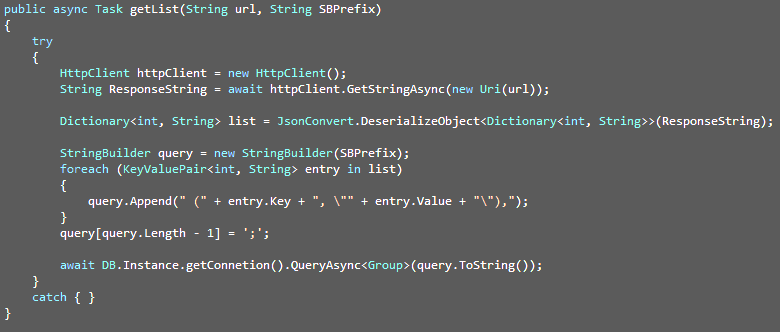


Рисунок 3.7 – Код для завантаження списків

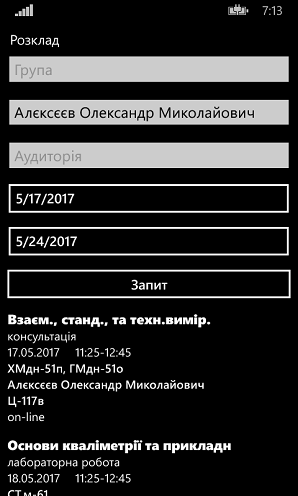


Рисунок 3.8 – Приклад відображення розкладу

Після обирання користувачем цікавих йому параметрів розкладу та натиснення кнопки «Запит» обробник події завантажує з сервера файл у форматі JSON, а синтаксичний аналізатор (парсер) перетворює дані. Ці дані починають асинхронно зберігатися у SQLite-файл та відображаються.

## 2.4 Створення новин

Спеціально для відображення новин у додатку було створено синтаксичний аналізатор (парсер), для отримання новин. Новини університету можна отримати використовуючи RSS. Синтаксично проаналізувавши файл отримуємо масив з новинами, який використовуємо для їх відображення на сторінці.

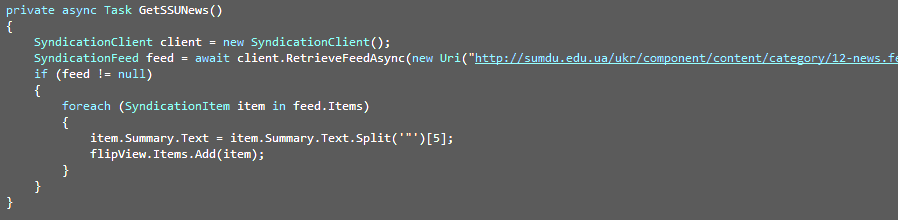


Рисунок 3.9– Фругмент коду парсеру новин університету



Рисунок 3.14–Приклад вигляду новин

# 4 РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ

## 4.1 UI інтерфейс мобільного додатку

У третьому розділі дипломного проекту було розглянуто програмну концепцію створення мобільного додатку, тепер розглянемо основну оболонку, саме те, що користувач бачить має можливість натиснути або потягнути. Як правило оболонка майбутнього мобільного додатку створюється завжди першою, тому що розробникам потрібно бачити, як буде виглядати додаток, які елементи повинні бути щоб їх у подальшому реалізувати програмно.

UI (скорочено) або (User Interface) – сукупність засобів для обробки та відображення інформації, максимально пристосованих для зручності користувача, у графічних системах інтерфейс користувача реалізовується багато віконним режимом, змінами кольору, розміру, видимості (прозорість, напівпрозорість, невидимість) вікон, їхнім розташуванням, сортуванням елементів вікон, гнучкими налаштовуваннями як самих вікон, так і окремих їхніх елементів (файли, папки, ярлики, шрифти тощо), доступністю багатокористувацьких налаштувань. Користувацький інтерфейс створюють через спеціальні програмні продукти такі як: “CorelDRAW” та Expression Blend тощо. Дизайн мобільного додатку було створено за допомогою програмного продукту “ CorelDRAW Graphics Suite X8” (рис. 3.1). Створення такого дизайну як правило починається з технічного завдання для дизайнера, якщо ж технічне завдання вже існує – можна починати працювати.

Дизайн додатку кафедри для Widows Phone 8.1 має на меті представити додаток як нативний. Саме тому він наслідував UI та UX платформи. Це дозволило спростити розробку дизайну.

Були створені іконки додатку. Іконки були створені з використанням CorelDRAW Graphics Suite X8.

Залишився останній момент – вибір шрифту. Враховуючи різні можливості людей бачити текст, треба підібрати правильний та доступний для всіх умов комфорту читання шрифт. За основу був взятий новий читабельний шрифт “Roboto” (рис. 4.1)



Рисунок 4.1 – Шрифт “Roboto”

Цей шрифт використовуються у різних варіаціях, наприклад перша варіація шрифту – “Roboto-Medium” використовується у заголовках кожного екрану і не більше.

Друга варіація - “Roboto-Regular”, використовується у додатковому тексті, наприклад: текст опису розділу у головному меню або опис завантажування файлів. Отже практично на всіх цих елементах тримається дизайн мобільного додатку.

## 4.2 UX інтерфейс мобільного додатку

Досвід користувача (user experience, скорочено UX) – це відчуття людини при взаємодії з системою. Системою може виступати мобільний додаток, веб-сайт, програмне забезпечення, тощо.

Ті, хто працює над створенням UX (так звані UX дизайнери) проводять дослідження і оцінюють взаємодію користувачів з системою, беручи до уваги такі речі, як простота використання, сприйняття вартості системи, корисності, ефективності тощо.

За весь період створення мобільного додатку було використано більше двох варіацій створення мобільного дизайну. Була проведена велика робота по визначенню шрифтів та кольорів, які б не навантажували очі та користувачу було б комфортніше читати або переглядати текстову інформацію. UX може існувати і працювати дуже ефективно з досить простим для користувача інтерфейсом. Наприклад, ви можете зустріти додаток з приголомшливим дизайном, яке складно використовувати (хороший UI, поганий UX). Також бувають додатки, які виглядають погано, але виявляються дуже зручними у використанні (поганий UI, хороший UX). Якість UX визначають спеціальні заходи тестування мобільного додатку, іншими словами це людина – тестер, саме вона перевіряє на зрозумілість для різних користувачів інтерфейс та перевіряє функціонал додатку, а після перевірки віддає звітність що до нюансів, які потрібно виправити для того, щоб пустити продукт на ринок для загального користування.

# ВИСНОВКИ

Дипломна робота орієнтована не тільки на абітурієнтів, які бажають навчатись на кафедрі комп’ютерних наук, а і для студентів, які вже навчаються або закінчили університет але мріють вчитись на аспірантурі або докторантурі.

Проект реалізовано з використанням мови програмування С#, XAML, що забезпечує достатню швидкодію додатку. Були використані методологія ООП, патерни проектування та особливості мов для запезпечення якості коду і легкої подальшої підтримки проекту.

Було ретельно описано основні функції додатку, архітектуру головного меню, UI та UX дизайн. У ході роботи можна було побачити, як змінювався дизайн проекту за часи його розробки починаючі від шрифтів та закінчуючи графікою. Також не забули про детальний аналіз ринку мобільних технологій, оцінили всі його переваги та недоліки, ознайомились з системою аналітики, розглянули типи систем аналізу.

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Microsoft Developer Network [Електронний ресурс] – Режим доступу до ресурсу: https://msdn.microsoft.com.
2. Stack Overflow [Електронний ресурс] – Режим доступу до ресурсу: http://stackoverflow.com/.
3. Habrahabr [Електронний ресурс] – Режим доступу до ресурсу: https://habrahabr.ru/.
4. Github [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/.
5. Нейгел К. C# 5.0 и платформа .NET 4.5 для профессионалов / К. Нейгел, Б. Ивьен. – Москва: Вильямс, 2014. – 1440 с.
6. Шевчук А., Касьянов А. Patterns via C#. Приемы объектно-ориентированного проектирования: ITVDN / А. Шевчук, Д. Охрименко, 2015 – 288с.
7. Иванов Д., Ф. Новиков Моделирование на UML / Д. Иванов, Ф. Новиков – Санкт-Петербург: СПбГУ ИТМО, 2010. - 200 с. - экз.
8. Proft [Електронний ресурс] – Режим доступу до ресурсу: 8. http://proft.me/2013/05/26/uml-2-tipy-diagramm.
9. Mymanager [Електронний ресурс] – Режим доступу до ресурсу: http://mymanager.com.ua/bp/bs/overview/notation\_idef0.php.
10. Батенко Л. П., Загородніх О. А., Ліщинська В. В. Б 28 Управління проектами: Навч. посібник. — К.: КНЕУ, 2003. — 231 с.
11. Ubooks [Електронний ресурс] – Режим доступу до ресурсу: http://ubooks.com.ua/books/000245/inx18.php.
12. Ubooks [Електронний ресурс] – Режим доступу до ресурсу: http://ubooks.com.ua/books/000245/inx14.php.
13. Wikipedia [Електронний ресурс] – Режим доступу до ресурсу: 13. https://uk.wikipedia.org/wiki/Діаграма\_Ганта.
14. Studfiles [Електронний ресурс] – Режим доступу до ресурсу: http://www.studfiles.ru/preview/5534709/page:9/.
15. Джозеф А. C# 5.0. Карманный справочник / А. Джозеф, А. Бен., 2013. – 288 с.
16. Симан М. Dependency Injection in .NET / М. Симан, А. Барышнев, Е. Зазноба., 2014. – 364 с.
17. Millett S. Patterns, Principles, and Practices of Domain-Driven Design / S. Millett, N. Tune., 2015. – 795 с.
18. Натан А. WPF 4: Unleashed / Адам Натан., 2011. – 880 с.
19. Троелсен Э. Pro Expression Blend 4 / Эндрю Троелсен., 2012. – 368 с. – (Вильямс). – (Expert's Voice).
20. Garofalo R. pic Building Enterprise Applications with Windows® Presentation Foundation and the Model View ViewModel Pattern / Raffaele Garofalo., 2011. – 225 с.

**ДОДАТОК А. Технічне завдання**

Міністерство освіти і науки України

Сумський державний університет

Кафедра комп’ютерних наук

Секція інформаційних технологій проектування

Технічне завдання

на розроблення програмного продукту

**Модуль перевірки знань студентів**

**Шифр: ПП-МД-МЕ**

**Аркушів 6**

2017

Додаток А.Технічне завдання

ЗМІСТ

[1 ВСТУП 40](#_Toc483581564)

[1.2 Область застосування 41](#_Toc483581565)

[1.3 Об’єкт, в якому використовують програму 41](#_Toc483581566)

[2 ПІДСТАВА ДЛЯ РОЗРОБКИ 41](#_Toc483581567)

[2.1 Документи на основі яких ведеться проектування 41](#_Toc483581568)

[2.2 Шифр теми 42](#_Toc483581569)

[3 ВИМОГИ ДО ПРОГРАМНОГО ВИРОБУ 42](#_Toc483581570)

[3.1 Вимоги до функціональних характеристик 42](#_Toc483581571)

[3.2 Вимоги до надійності 42](#_Toc483581572)

[3.3 Умови експлуатації 43](#_Toc483581573)

[3.4 Вимоги до складу і параметрами технічних засобів 43](#_Toc483581574)

[3.5 Вимоги до інформаційної та програмної сумісності 44](#_Toc483581575)

[4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ 44](#_Toc483581576)

[5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ 44](#_Toc483581577)

[6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ 44](#_Toc483581578)

**1 ВСТУП**

Дана програма використовується викладачами для перевірки набутих знань студентів, шляхом проходження тесту по вибраному предмету.

**1.1 Найменування програми**

Додаток для Windows Phone. повинен відповідати поставленим критеріям:

* мобільний додаток повинен бути безкоштовним;
* мобільний додаток повинен бути без реклами, яка б псувала або заважали комфортній роботі;
* інтерфейс повинен бути інтуїтивно-зрозумілим, повинен підходити для кожного користувача не залежно від полу, року та здібностей;
* усі тексти повинні бути читабельними, кольори - не навантажувати очі як при денній так і вечірній годині;
* надати можливість перегляду розкладу СумДУ.
* відображення новин університету;
* надання доступу до корисної інформації;
* надати користувачу детальний опис дистанційного, заочного та денного форм навчання в СумДУ, ознайомити користувачів з методом навчання на дистанційній формі;

**1.2 Область застосування**

Полегшення доступу до корисної інформації, максимальне ознайомлення абітурієнта з важливою інформацією, що стосується вступу, також з навчальним закладом СумДУ.

**1.3 Об’єкт, в якому використовують програму**

Додаток можуть використовувати в студенти, абітурієнти, викладачі.

**2** **ПІДСТАВА ДЛЯ РОЗРОБКИ**

Розробка ведеться на основі дипломної роботи.

Найменування організації: група ІТдн-21, кафедра Комп’ютерних наук, секція ІТП. Тема проекту: «Мобільний додаток кафедри комп’ютерних наук»

**2.1 Документи на основі яких ведеться проектування**

1. Техническоезадание. Требования к содержаниюи оформлению. ГОСТ19.201-78.  
2. Руководство системного программиста. Требования к содержанию и оформлению. ГОСТ19.503-79

 3. Руководство оператора. Требования к содержаниюи оформлению. ГОСТ19.505-79

4. C# 2005 и платформа .Net 3.0 для профессионалов. Нейгел К., Ивьен Б. и др. –М.: ООО “И.Д. Вильямс", 2008. –1376 с.

5. ЭнрюТроелсен. Язык программирования C# 5.0 и платформа .Net 4.5, 6-е изд: Пер. с англ. –М.: ООО “И.Д. Вильямс", 2013. –1312 с.

**2.2 Шифр теми**

Шифр теми: ПП-МД-ККП.

**3 ВИМОГИ ДО ПРОГРАМНОГО ВИРОБУ**

При реалізації та використанні додатку повинні бути враховані вимоги до функціональних характеристик, надійності проекту, параметрами технічних засобів, інформаційної і програмної сумісності.

**3.1 Вимоги до функціональних характеристик**

ПП повинен виконувати наступні функції:

* розмір додатку не повинен перевищувати 35 мегабайт;
* мобільний додаток повинен бути безкоштовним;
* надати користувачу доступ до інформації у зручному та правильному контексті до всіх розділів;
* надати можливість перегляду розкладу СумДУ.
* відображення новин університету;
* надання доступу до корисної інформації;

**3.2 Вимоги до надійності**

Розроблюваний програмний продукт для тестування студентів не повинен вимагати від користувача спеціальної підготовки і так само має мати захист від помилкових дій користувача, а саме:

* повинна бути розроблена інструкція користувача;
* контроль вводу даних користувачем;

**3.3 Умови експлуатації**

Програма може бути перенесена на такі види носіїв даних як: лазерний диск, флеш-накопичувач, жорсткий диск. Не повинно бути впливів води і механічних ударів на носій. Умови експлуатації повинні відповідати санітарним та технічним нормам експлуатації ЕОМ і визначаються відповідно до СанПіН2.2.2545-96 «Гігієнічні вимоги до відео-дисплейних терміналів, персональних обчислювальних машин і організації роботи». Для роботи з ПК допускаються працівники, які мають достатній рівень знань в предметній області.

**3.4 Вимоги до складу і параметрами технічних засобів**

* Мобільний телефон з операційною системою Windows Phone 8.1;

**3.5 Вимоги до інформаційної та програмної сумісності**

Для роботи програмного продукту автоматичного формування специфікацій на машині повинні бути встановлені:

* операційна система Windows Phone 8.1;

**4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ**

Попередній склад програмної документації встановлено у відповідності з ГОСТ19.101-77. Нижче наведений список програмних документів та їх зміст.  
Технічне завдання - даний документ.

Експлуатаційні документи - керівництво користувача до програмного продукту та керівництво системного адміністратора.

**5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ**

1. до 15.03.2017 р. Аналіз існуючих аналогів.
2. до 20.03.2017 р. Розробка ТЗ.
3. до 29.03.2017 р. Розробка моделі ПП.
4. до 10.05.2017 р. Розробка програмного продукту в Visual Studio.
5. до 28.05.2017 р. Відладка та тестування.
6. до 02.06.2017 р. Розробка супроводжуючої документації
7. до 04.06.2017 р. Здача в експлуатацію

**6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ**

Контроль коректності функціонування і придатність мобільного додатку для Windows Phone до використання в робочих умовах виконується спільно розробниками і споживачами даного програмного комплексу на підставі його тестування. Рішення про приймання в експлуатацію приймається на підставі результатів тестування .

Порядок контролю:

1.Контроль плану роботи (до 15.03.2017 р.)

2.Контроль проекту (до 10.05.2017 р.)

3.Тестування додатку (до 28.05.2017 р.)

4. Здача супроводжуючої документації (до 02.06.2017 р.)

6. Презентація роботи (до 10.06.2017 р.)

**ДОДАТОК Б**

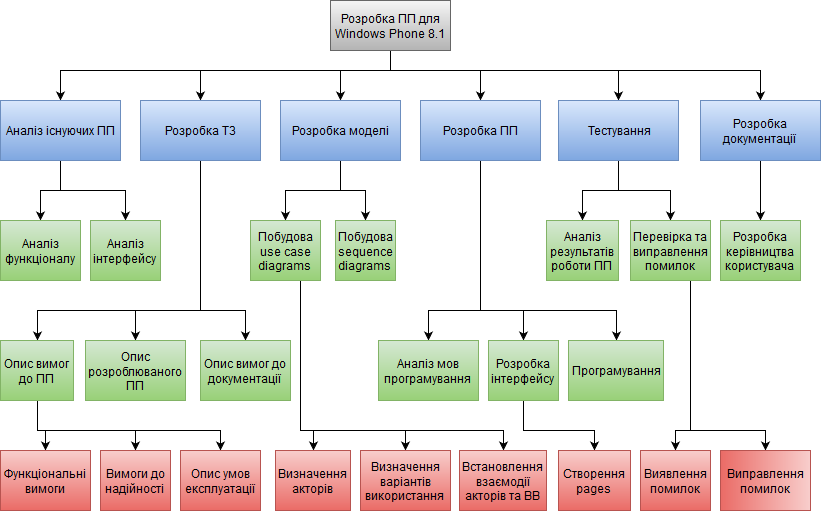


Рисунок Б.1 – Ієрархічна структура робіт (WBS)

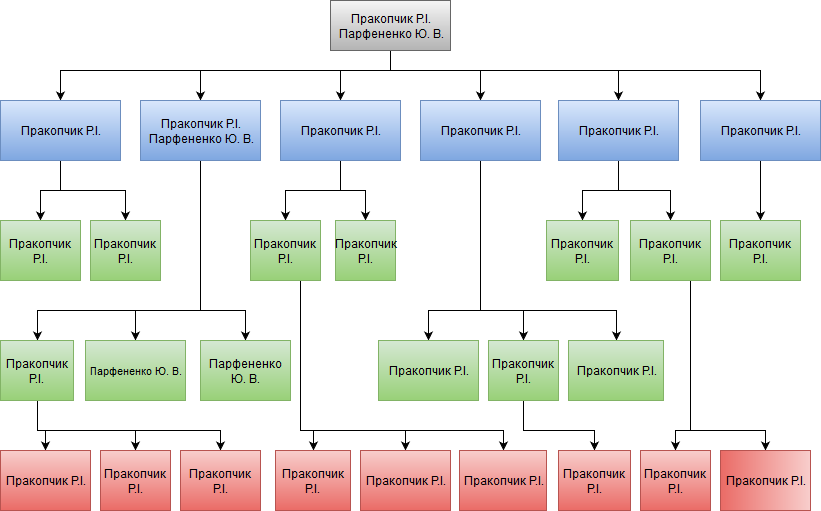


Рисунок Б.2 – Організаційна структура виконавців проекту (OBS)

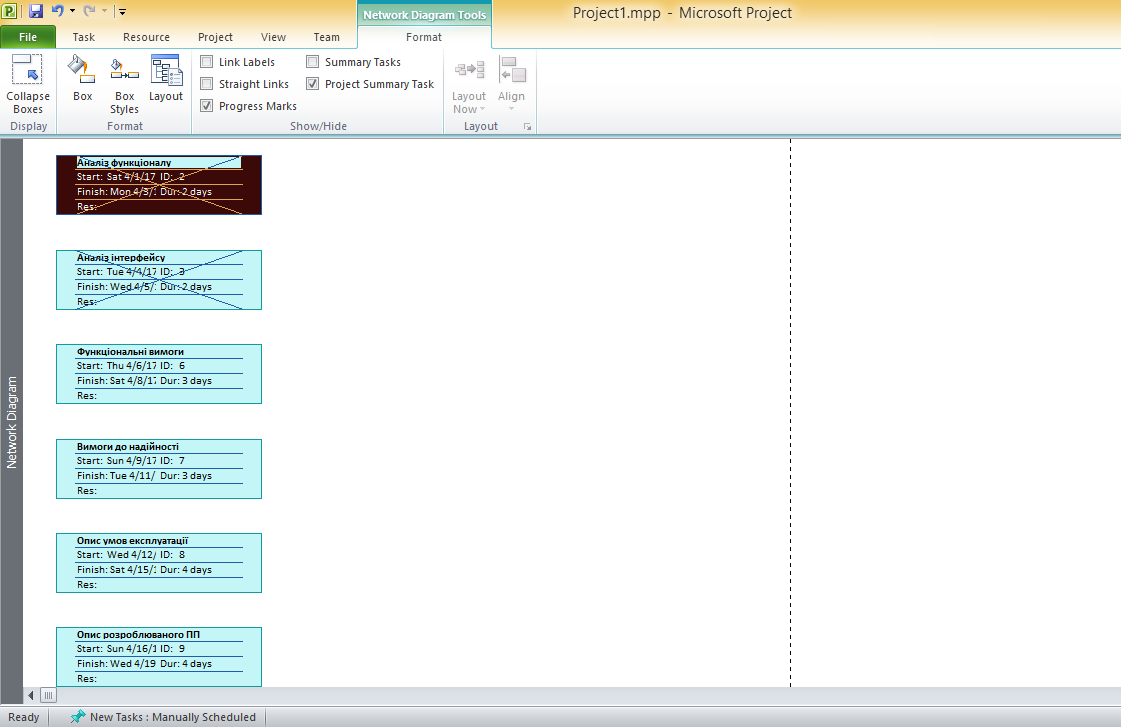


Рисунок Б.3 – PDM-мережа всіх робіт

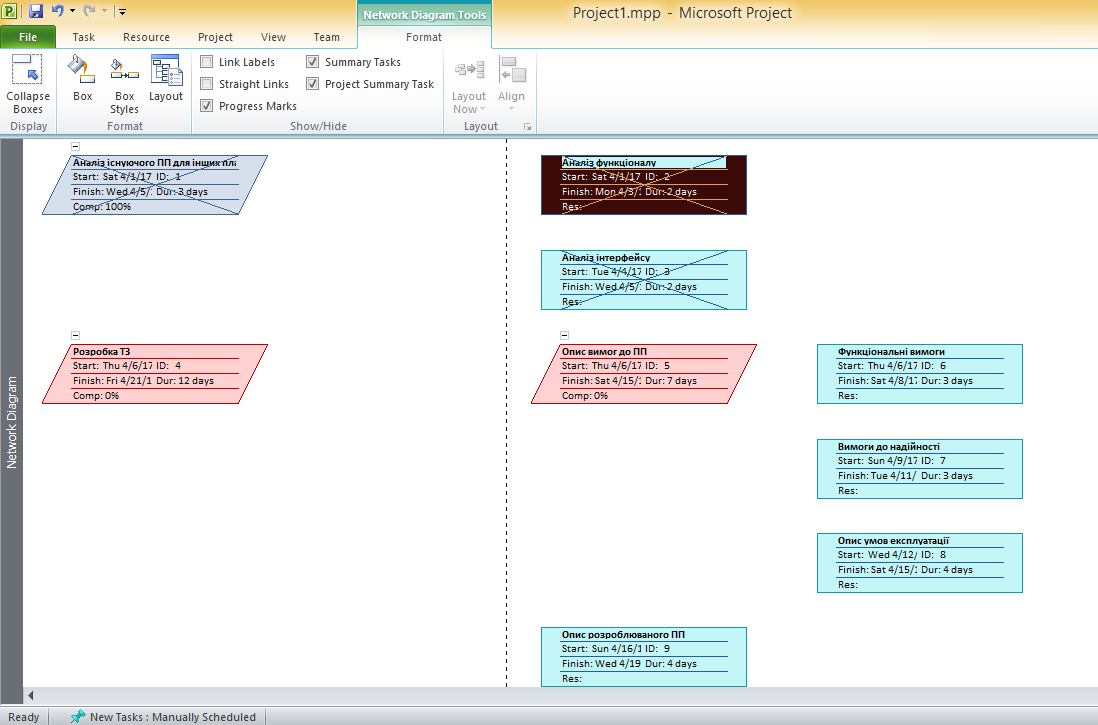


Рисунок Б.4 – PDM-мережа загальних робіт

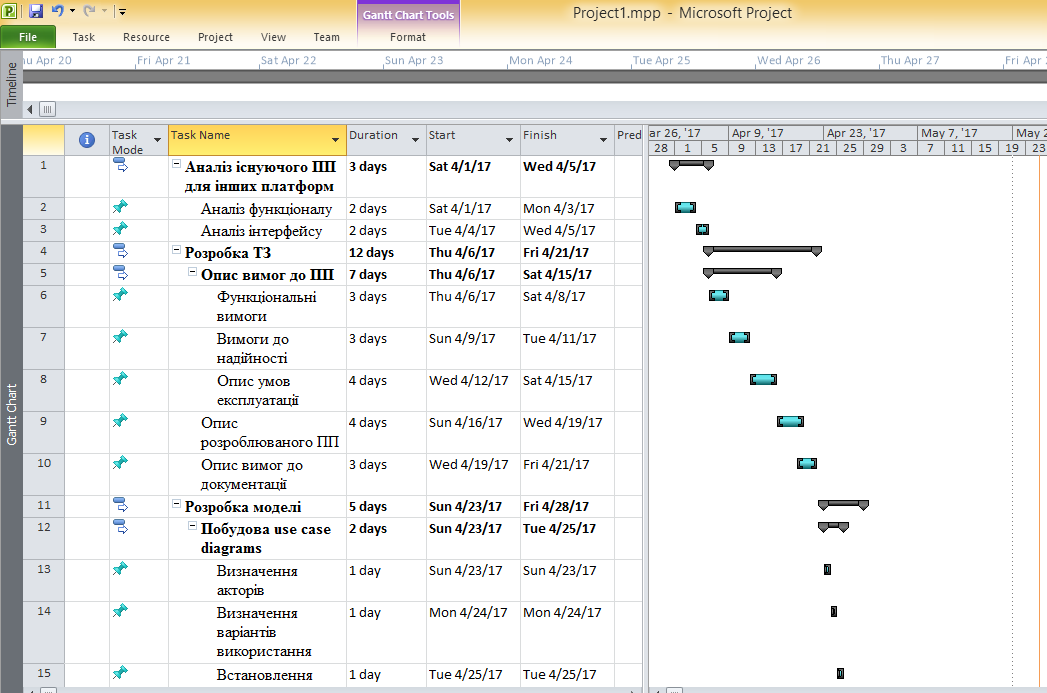


Рисунок Б.5 – Діаграма Ганта

**ДОДАТОК В**

Таблиця В.1 – Матриця відповідальності

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Розробка ПП** | **Парфененко Ю. В.** | **Пракопчик Р.І.** |
| 1 | Аналіз існуючого ПП для інших платформ |  |  |
| 1.1 | Аналіз функціоналу |  |  |
| 1.2 | Аналіз інтерфейсу |  |  |
| 2 | Розробка ТЗ |  |  |
| 2.1 | Опис вимог до ПП |  |  |
| 2.1.1 | Функціональні вимоги |  |  |
| 2.1.2 | Вимоги до надійності |  |  |
| 2.1.3 | Опис умов експлуатації |  |  |
| 2.2 | Опис розроблюваного ПП |  |  |
| 2.3 | Опис вимог до документації |  |  |
| 3 | Розробка моделі |  |  |
| 3.1 | Побудова use case diagrams |  |  |
| 3.1.1 | Визначення акторів |  |  |
| 3.1.2 | Визначення варіантів використання |  |  |
| 3.1.3 | Встановлення взаємодії акторів та ВВ |  |  |
| 3.2 | Побудова sequence diagrams |  |  |
| 4 | Розробка ПП |  |  |
| 4.1 | Аналіз мов програмування |  |  |
| 4.2 | Розробка інтерфейсу |  |  |
| 4.2.1 | Створення pages |  |  |
| 4.3 | Програмування |  |  |
| 5 | Тестування |  |  |
| 5.1 | Аналіз результатів роботи ПП |  |  |
| 5.2 | Перевірка та виправлення помилок |  |  |
| 5.2.1 | Виявлення помилок |  |  |
| 5.2.2 | Виправлення помилок |  |  |
| 6 | Розробка необхідної документації |  |  |
| 6.1 | Розробка керівництво користувача |  |  |

**ДОДАТОК Г**

Розмітка головної сторінки:

<Page

x:Class="CSSumdu.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:CSSumdu"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

<Grid>

<ListView>

<TextBlock Text="Кафедра комп'ютерних наук" FontSize="{StaticResource TextStyleExtraExtraLargeFontSize}" TextAlignment="Center" TextWrapping="Wrap" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" Padding="30,0,30,30" Width="Auto" />

<FlipView x:Name="flipView" Height="100" VerticalAlignment="Top" Padding="20,0" Margin="0">

<FlipView.ItemTemplate>

<DataTemplate>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Image Grid.Column="0" Width="150" Stretch="None" Source="{Binding Path=Summary.Text}"/>

<HyperlinkButton Grid.Column="0" Width="150" Height="100" NavigateUri="{Binding Path=Id}"/>

<TextBlock Grid.Column="1" Text="{Binding Path=Title.Text}" FontFamily="Segoe UI" Padding="20, 0, 0, 0" TextWrapping="WrapWholeWords" FontSize="16" MaxLines="4" VerticalAlignment="Center"/>

<HyperlinkButton Grid.Column="1" Width="150" Height="100" NavigateUri="{Binding Path=Id}"/>

</Grid>

</DataTemplate>

</FlipView.ItemTemplate>

</FlipView>

<TextBlock Text="Останні новини" FontSize="{StaticResource TextStyleMediumFontSize}" Margin="20,0,20,30" />

<ListView x:Name="mainView" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" ScrollViewer.VerticalScrollBarVisibility="Hidden" Tapped="mainView\_Tapped">

<ListView.ItemTemplate>

<DataTemplate>

<Grid>

<TextBlock Text="{Binding header}" TextAlignment="Left" TextWrapping="Wrap" VerticalAlignment="Center" Padding="25,10" FontSize="{StaticResource TextStyleExtraLargeFontSize}" FontWeight="SemiBold" LineStackingStrategy="BlockLineHeight" LineHeight="{StaticResource TextStyleExtraLargeFontSize}"/>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</ListView>

</Grid>

</Page>

Код головної сторінки:

using CSSumdu.View;

using CSSumdu.ViewModel;

using System;

using System.Threading.Tasks;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Navigation;

using Windows.Web.Syndication;

namespace CSSumdu

{

public sealed partial class MainPage : Page

{

public MainPage()

{

this.InitializeComponent();

this.NavigationCacheMode = NavigationCacheMode.Required;

addItems();

}

protected async override void OnNavigatedTo(NavigationEventArgs e)

{

await DB.Instance.init();

await GetSSUNews();

Task[] tasks = new Task[3];

tasks[0] = Schedule.Instance.getList("http://schedule.sumdu.edu.ua/index/json?method=getGroups", "INSERT INTO groups (id, name) VALUES");

tasks[1] = Schedule.Instance.getList("http://schedule.sumdu.edu.ua/index/json?method=getTeachers", "INSERT INTO teachers (id, name) VALUES");

tasks[2] = Schedule.Instance.getList("http://schedule.sumdu.edu.ua/index/json?method=getAuditoriums", "INSERT INTO auditoriums (id, name) VALUES");

await Task.WhenAll(tasks);

}

private async Task GetSSUNews()

{

SyndicationClient client = new SyndicationClient();

SyndicationFeed feed = await client.RetrieveFeedAsync(new Uri("http://sumdu.edu.ua/ukr/component/content/category/12-news.feed"));

if (feed != null)

{

foreach (SyndicationItem item in feed.Items)

{

item.Summary.Text = item.Summary.Text.Split('"')[5];

flipView.Items.Add(item);

}

}

}

private class MenuItem

{

public String header { get; set; }

public Type type { get; set; }

}

private void mainView\_Tapped(object sender, Windows.UI.Xaml.Input.TappedRoutedEventArgs e)

{

var item = (sender as ListView).SelectedItem as MenuItem;

if (item != null)

{

Frame.Navigate(item.type);

}

}

private void addItems()

{

mainView.Items.Add(new MenuItem()

{

header = "Розклад",

type = typeof(ScheduleQueryPage),

});

mainView.Items.Add(new MenuItem()

{

header = "Інформація про спеціальності",

type = typeof(SpecialtyPage),

});

mainView.Items.Add(new MenuItem()

{

header = "Історія кафедри",

type = typeof(HistoryPage),

});

mainView.Items.Add(new MenuItem()

{

header = "Науково-дослідна робота",

type = typeof(ResearchPage),

});

mainView.Items.Add(new MenuItem()

{

header = "Міжнародне співробітництво",

type = typeof(InternationalPage),

});

mainView.Items.Add(new MenuItem()

{

header = "Структурні підрозділи кафедри",

type = typeof(SubdivisionsPage),

});

mainView.Items.Add(new MenuItem()

{

header = "Проходження практики",

type = typeof(PracticePage),

});

}

}

}

Класс для роботи з базою даних:

using CSSumdu.Model;

using SQLite;

using System.Threading.Tasks;

namespace CSSumdu.ViewModel

{

class DB

{

private static DB instance;

private DB() { }

public static DB Instance

{

get

{

if (instance == null)

{

instance = new DB();

}

return instance;

}

}

private static SQLiteAsyncConnection conn;

public async Task init()

{

conn = new SQLiteAsyncConnection("cssumdu.db");

await conn.CreateTableAsync<Event>();

await conn.CreateTableAsync<Group>();

await conn.CreateTableAsync<Teacher>();

await conn.CreateTableAsync<Auditorium>();

}

public SQLiteAsyncConnection getConnetion()

{

return conn;

}

}

}

Класс для роботи з розкладом:

using CSSumdu.Model;

using Newtonsoft.Json;

using System;

using System.Collections.Generic;

using System.Net.Http;

using System.Text;

using System.Threading.Tasks;

namespace CSSumdu.ViewModel

{

class Schedule

{

private static Schedule instance;

private Schedule() { }

public static Schedule Instance

{

get

{

if (instance == null)

{

instance = new Schedule();

}

return instance;

}

}

public async Task getList(String url, String SBPrefix)

{

try

{

HttpClient httpClient = new HttpClient();

String ResponseString = await httpClient.GetStringAsync(new Uri(url));

Dictionary<int, String> list = JsonConvert.DeserializeObject<Dictionary<int, String>>(ResponseString);

StringBuilder query = new StringBuilder(SBPrefix);

foreach (KeyValuePair<int, String> entry in list)

{

query.Append(" (" + entry.Key + ", \"" + entry.Value + "\"),");

}

query[query.Length - 1] = ';';

await DB.Instance.getConnetion().QueryAsync<Group>(query.ToString());

}

catch { }

}

// http://schedule.sumdu.edu.ua/index/json?method=getSchedules&id\_grp=0&id\_fio=0&id\_aud=0&date\_beg=00.00.0000&date\_end=00.00.0000

public async Task getSchedule(int groupId, int teacherId, int auditoriumId, DateTimeOffset begin, DateTimeOffset end)

{

StringBuilder url = new StringBuilder("http://schedule.sumdu.edu.ua/index/json?method=getSchedules");

url.Append("&id\_grp=" + groupId + "&id\_fio=" + teacherId + "&id\_aud=" + auditoriumId);

url.Append("&date\_beg=" + begin.Day + "." + begin.Month + "." + begin.Year);

url.Append("&date\_end=" + end.Day + "." + end.Month + "." + end.Year);

try

{

HttpClient httpClient = new HttpClient();

String ResponseString = await httpClient.GetStringAsync(new Uri(url.ToString()));

List<Event> list = JsonConvert.DeserializeObject<List<Event>>(ResponseString);

StringBuilder query = new StringBuilder("INSERT INTO Events (START\_TIME, DATE\_REG, TIME\_PAIR, NAME\_FIO, NAME\_AUD, NAME\_GROUP, ABBR\_DISC, NAME\_STUD, REASON) VALUES");

foreach (Event entry in list)

{

var date = entry.DATE\_REG.Split('.');

char[] array = { ':', '-' };

var time = entry.TIME\_PAIR.Split(array);

DateTimeOffset d = new DateTimeOffset(Int32.Parse(date[2]), Int32.Parse(date[1]),

Int32.Parse(date[0]), Int32.Parse(time[0]), Int32.Parse(time[1]), 0, new TimeSpan());

query.Append(" (" + d.Ticks +

", \"" + entry.DATE\_REG +

"\", \"" + entry.TIME\_PAIR +

"\", \"" + entry.NAME\_FIO +

"\", \"" + entry.NAME\_AUD +

"\", \"" + entry.NAME\_GROUP +

"\", \"" + entry.ABBR\_DISC +

"\", \"" + entry.NAME\_STUD +

"\", \"" + entry.REASON +

"\"),");

}

query[query.Length - 1] = ';';

await DB.Instance.getConnetion().QueryAsync<Event>("DELETE FROM events");

await DB.Instance.getConnetion().QueryAsync<Event>(query.ToString());

}

catch { }

}

}

}

Класс таблиці Event у БД:

using SQLite;

using System;

namespace CSSumdu.Model

{

[Table("Events")]

class Event

{

[PrimaryKey]

public long START\_TIME { get; set; }

public String DATE\_REG { get; set; }

public String TIME\_PAIR { get; set; }

[Indexed]

public String NAME\_FIO { get; set; }

[Indexed]

public String NAME\_AUD { get; set; }

[Indexed]

public String NAME\_GROUP { get; set; }

public String ABBR\_DISC { get; set; }

public String NAME\_STUD { get; set; }

public String REASON { get; set; }

}

}

Класс таблиці Group у БД:

using SQLite;

using System;

namespace CSSumdu.Model

{

[Table("Groups")]

class Group

{

[PrimaryKey, Indexed]

public int id { get; set; }

[Indexed]

public String name { get; set; }

}

}

Класс таблиці Teacher у БД:

using SQLite;

using System;

namespace CSSumdu.Model

{

[Table("Teachers")]

class Teacher

{

[PrimaryKey, Indexed]

public int id { get; set; }

[Indexed]

public String name { get; set; }

}

}

Класс таблиці Auditorium у БД:

using SQLite;

using System;

namespace CSSumdu.Model

{

[Table("Auditoriums")]

class Auditorium

{

[PrimaryKey, Indexed]

public int id { get; set; }

[Indexed]

public String name { get; set; }

}

}

Класс для роботи з розкладом:

using CSSumdu.Model;

using CSSumdu.ViewModel;

using System;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Threading.Tasks;

using Windows.UI;

using Windows.UI.ViewManagement;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

namespace CSSumdu.View

{

public sealed partial class ScheduleQueryPage : Page

{

public ScheduleQueryPage()

{

this.InitializeComponent();

}

protected async override void OnNavigatedTo(NavigationEventArgs e)

{

end.Date = start.Date.AddDays(7);

var eventList = await DB.Instance.getConnetion().Table<Event>().ToListAsync();

eventList.Sort((x, y) => x.START\_TIME.CompareTo(y.START\_TIME));

ObservableCollection<Event> bindedList = new ObservableCollection<Event>(eventList);

ListBox.DataContext = bindedList;

}

private async void AutoSuggestBox\_TextChanged(AutoSuggestBox sender, AutoSuggestBoxTextChangedEventArgs args)

{

List<String> myList = new List<String>();

var groupList = await DB.Instance.getConnetion().QueryAsync<Pair>("SELECT \* FROM " + sender.Tag + " WHERE name LIKE \"%" + sender.Text + "%\"");

foreach (var group in groupList)

{

myList.Add(group.name);

}

sender.ItemsSource = myList;

}

private void AutoSuggestBox\_SuggestionChosen(AutoSuggestBox sender, AutoSuggestBoxSuggestionChosenEventArgs args)

{

b.Focus(FocusState.Programmatic);

}

private void AutoSuggestBox\_GotFocus(object sender, RoutedEventArgs e)

{

if ((sender as AutoSuggestBox).Text == "")

{

AutoSuggestBox\_TextChanged(sender as AutoSuggestBox, null);

}

}

private class Pair

{

public int id { get; set; }

public String name { get; set; }

}

private async void b\_Click(object sender, RoutedEventArgs e)

{

StatusBarProgressIndicator progressbar = StatusBar.GetForCurrentView().ProgressIndicator;

int group = (gr.Text == "") ? 0 : (await DB.Instance.getConnetion().QueryAsync<Pair>("SELECT id FROM groups WHERE name=\"" + gr.Text + "\" LIMIT 1;"))[0].id;

int teacher = (te.Text == "") ? 0 : (await DB.Instance.getConnetion().QueryAsync<Pair>("SELECT id FROM teachers WHERE name=\"" + te.Text + "\" LIMIT 1;"))[0].id;

int auditor = (au.Text == "") ? 0 : (await DB.Instance.getConnetion().QueryAsync<Pair>("SELECT id FROM auditoriums WHERE name=\"" + au.Text + "\" LIMIT 1;"))[0].id;

if (group != 0 | teacher != 0 | auditor != 0)

{

await progressbar.ShowAsync();

await Schedule.Instance.getSchedule(group, teacher, auditor, start.Date, end.Date);

var eventList = await DB.Instance.getConnetion().Table<Event>().ToListAsync();

eventList.Sort((x, y) => x.START\_TIME.CompareTo(y.START\_TIME));

ObservableCollection<Event> bindedList = new ObservableCollection<Event>(eventList);

ListBox.DataContext = bindedList;

await progressbar.HideAsync();

}

}

}

}

Розмітка ScheduleQueryPage:

<Page

x:Class="CSSumdu.View.ScheduleQueryPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:CSSumdu.View"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

<Page.Resources>

<Thickness x:Key="TextControlMarginThemeThickness">0,9.5,0,9.5</Thickness>

<Thickness x:Key="AutoSuggestListViewItemMargin">6.5,9.5,0,0</Thickness>

<x:Double x:Key="ContentControlFontSize">20.26</x:Double>

<Thickness x:Key="TextControlBorderThemeThickness">2.5</Thickness>

<Style TargetType="AutoSuggestBox">

<Setter Property="Margin" Value="{ThemeResource TextControlMarginThemeThickness}"/>

<Setter Property="VerticalAlignment" Value="Top"/>

<Setter Property="IsTabStop" Value="False"/>

<Setter Property="ItemContainerStyle">

<Setter.Value>

<Style TargetType="ListViewItem">

<Setter Property="Margin" Value="{ThemeResource AutoSuggestListViewItemMargin}"/>

<Setter Property="FontSize" Value="{ThemeResource ContentControlFontSize}"/>

<Setter Property="Foreground" Value="{ThemeResource TextBoxForegroundThemeBrush}"/>

</Style>

</Setter.Value>

</Setter>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="AutoSuggestBox">

<Grid>

<VisualStateManager.VisualStateGroups>

<VisualStateGroup x:Name="Orientation">

<VisualState x:Name="Landscape"/>

<VisualState x:Name="Portrait"/>

</VisualStateGroup>

</VisualStateManager.VisualStateGroups>

<TextBox x:Name="TextBox" ScrollViewer.BringIntoViewOnFocusChange="False" Header="{Binding Header, RelativeSource={RelativeSource Mode=TemplatedParent}}" Margin="0" PlaceholderText="{Binding PlaceholderText, RelativeSource={RelativeSource Mode=TemplatedParent}}" Width="{TemplateBinding Width}" Canvas.ZIndex="0" BorderBrush="{Binding BorderBrush, RelativeSource={RelativeSource Mode=TemplatedParent}}"/>

<Popup x:Name="SuggestionsPopup">

<Border x:Name="SuggestionsContainer" BorderBrush="{ThemeResource PhoneAccentBrush}" BorderThickness="{ThemeResource TextControlBorderThemeThickness}" Background="{ThemeResource AutoSuggestBackgroundThemeBrush}">

<Border.RenderTransform>

<TranslateTransform x:Name="UpwardTransform"/>

</Border.RenderTransform>

<ListView x:Name="SuggestionsList" ItemTemplate="{Binding ItemTemplate, RelativeSource={RelativeSource Mode=TemplatedParent}}" ItemsSource="{Binding ItemsSource, RelativeSource={RelativeSource Mode=TemplatedParent}}" ItemContainerStyle="{Binding ItemContainerStyle, RelativeSource={RelativeSource Mode=TemplatedParent}}" ItemTemplateSelector="{Binding ItemTemplateSelector, RelativeSource={RelativeSource Mode=TemplatedParent}}" RenderTransformOrigin=".5,.5">

<ListView.RenderTransform>

<ScaleTransform x:Name="ListItemOrderTransform"/>

</ListView.RenderTransform>

</ListView>

</Border>

</Popup>

</Grid>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Page.Resources>

<Grid>

<ListView>

<Grid Width="400">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<StackPanel Width="Auto">

<TextBlock Margin="13,13,13,0" Text="Розклад" FontSize="{StaticResource TextStyleLargeFontSize}"/>

<AutoSuggestBox x:Name="gr" TextChanged="AutoSuggestBox\_TextChanged" SuggestionChosen="AutoSuggestBox\_SuggestionChosen" GotFocus="AutoSuggestBox\_GotFocus" Margin="10,9.5" Tag="groups" PlaceholderText="Група"/>

<AutoSuggestBox x:Name="te" TextChanged="AutoSuggestBox\_TextChanged" SuggestionChosen="AutoSuggestBox\_SuggestionChosen" GotFocus="AutoSuggestBox\_GotFocus" Margin="10,9.5" Tag="teachers" PlaceholderText="Викладач"/>

<AutoSuggestBox x:Name="au" TextChanged="AutoSuggestBox\_TextChanged" SuggestionChosen="AutoSuggestBox\_SuggestionChosen" GotFocus="AutoSuggestBox\_GotFocus" Margin="10,9.5" Tag="Auditoriums" PlaceholderText="Аудиторiя"/>

<DatePicker x:Name="start" Margin="12,0"/>

<DatePicker x:Name="end" Margin="12,0"/>

<Button x:Name="b" Width="Auto" Content="Запит" Margin="10,0" Click="b\_Click" HorizontalAlignment="Stretch"/>

<ListView x:Name="ListBox" Width="Auto" HorizontalAlignment="Left" ItemsSource="{Binding}" >

<ListView.ItemTemplate>

<DataTemplate>

<Grid Width="390">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" FontSize="{StaticResource TextStyleLargeFontSize}" Text="{Binding ABBR\_DISC}" FontWeight="ExtraBold" Margin="10,5,10,0"/>

<TextBlock Grid.Row="1" Text="{Binding NAME\_STUD}" Margin="10,0" FontSize="{StaticResource TextStyleMediumFontSize}"/>

<StackPanel Grid.Row="2" Orientation="Horizontal">

<TextBlock Text="{Binding DATE\_REG}" Margin="10,0" FontSize="{StaticResource TextStyleMediumFontSize}"/>

<TextBlock Text="{Binding TIME\_PAIR}" Margin="10,0" FontSize="{StaticResource TextStyleMediumFontSize}"/>

</StackPanel>

<TextBlock Grid.Row="3" Text="{Binding NAME\_GROUP}" Margin="10,0" FontSize="{StaticResource TextStyleMediumFontSize}" FontWeight="SemiBold"/>

<TextBlock Grid.Row="4" Text="{Binding NAME\_FIO}" Margin="10,0" FontSize="{StaticResource TextStyleMediumFontSize}" FontWeight="SemiBold"/>

<TextBlock Grid.Row="5" Text="{Binding NAME\_AUD}" Margin="10,0" FontSize="{StaticResource TextStyleMediumFontSize}" FontWeight="SemiBold"/>

<TextBlock Grid.Row="6" Text="{Binding REASON}" FontSize="{StaticResource TextStyleMediumFontSize}" Margin="10,0,10,10"/>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</StackPanel>

</Grid>

</ListView>

</Grid>

</Page>