

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра комп'ютерних наук та програмної інженерії

**РОЗРОБКА УНІВЕРСАЛЬНОГО ДОДАТКУ ДЛЯ АБІТУРІЄНТІВ
ТА СТУДЕНТІВ НА МОБІЛЬНІ ПРИСТРОЇ**
Кваліфікаційна робота (проект)
на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 4 курсу 441 групи

Спеціальності: 121 Інженерія

програмного забезпечення

Освітньо-професійної програми:

Інженерія програмного забезпечення

першого

Гриценко Денис Сергійович

Керівник: доцент кафедри

комп'ютерних наук та програмної

інженерії

Шишко Людмила Станіславівна

Рецензент: Тарасіч Ю. Г.

директор ТОВ "ГАРУДА.АІ"

Херсон – Івано-Франківськ – 2023

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1: ОСНОВНІ ПОНЯТТЯ	5
Основні поняття Android розробки.....	5
Мови програмування, що підтримуються платформою Android.....	6
ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ	11
РОЗДІЛ 2: АНАЛІЗ ТА ВИМОГИ	12
Аналіз ринку додатків	12
Формування вимог до проекту	15
ВИСНОВКИ ДО ДРУГОГО РОЗДІЛУ	18
РОЗДІЛ 3: ПРОЕКТУВАННЯ ТА РОЗРОБКА	19
Опис обраної технології	19
Розробка додатку	20
Сценарії використання додатку.....	22
ВИСНОВКИ ДО ТРЕТЬОГО РОЗДІЛУ	29
ВИСНОВКИ	30
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	31
ДОДАТКИ	33
ДОДАТОК А. КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ	33

ВСТУП

Актуальність дослідження: додаток з такою тематикою є актуальним та корисним вирішенням для багатьох студентів з наступних причин:

- Зручність та доступність: подібний мобільний додаток надає швидкий та доступний доступ до різних сервісів та ресурсів університету. Воно буде доступним в будь-який час і в будь-якому місці, що дозволить користувачам залишатися в курсі всіх подій та змін у житті університету.
- Покращення навчального процесу: додаток допоможе студентам оптимізувати та поліпшити свій навчальний процес. Гнучка система розкладу в купі з відстеженням поточних завдань з кожного предмета допоможе студенту стати більш організованим і продуктивним на заняттях.
- Конкурентна перевага: мобільний додаток, в очах абітурієнтів, може стати додатковою конкурентною перевагою для університету та дозволить залучити більше нових студентів.

Таким чином, мобільний додаток для абітурієнтів та студентів є актуальним рішенням, яке може допомогти студентам та абітурієнтам краще орієнтуватися у житті університету.

Об'єкт роботи: мобільний додаток

Предмет роботи: мобільний додаток-посібник для студентів та абітурієнтів

Мета роботи: Метою роботи є проектування універсального додатку для абітурієнтів та студентів на мобільні пристрої, який буде надавати корисний функціонал та інформацію

Для досягнення мети були поставлені наступні завдання:

1. Провести аналіз ринку додатків, схожою за тематикою, або з функціоналом, котрий може гарно вписатися в додаток, що знаходиться в розробці
2. На базі аналізу ринку виокремити основні особливості та, якщо можливо, реалізації функціоналу
3. Визначити вимоги до додатку
4. Створення макету додатку і його фінальної реалізації

В першому розділі будуть наведені теоретичні відомості щодо платформи Android, історії її розробки та технології й мови програмування, що використовуються для створення мобільних додатків. Кожен варіант буде проаналізовано, будуть виділені плюси та мінуси використання.

Другий розділ присвячений аналізу мобільних додатків. Будуть знайдені додатки, що підходять за тематикою, і розглянуті. Акцент буде поставлений на фактичні можливості додатку та його інтерфейс. Після цього аналізу будуть створені вимоги до проекту.

В третьому розділі, на базі проаналізованої інформації з першого розділу, буде описані фінальні технології, що будуть використовуватись під час створення мобільного додатку. Буде наведений сценарій розробки з описом розташування елементів на екрані та його функціоналу. Будуть наведені сценарії використання додатку, що висвітлить взаємозв'язок його окремих модулів.

РОЗДІЛ 1: ОСНОВНІ ПОНЯТТЯ

Основні поняття Android розробки

Для початку необхідно зазначити, що таке Android. Android є операційною системою, що базується на ядрі Linux[1]. Дана операційна система розроблялася для таких мобільних пристроїв, як планшети і смартфони. Торгівельна марка Android належала компанії «Android Inc» до 2005 року. Але це не лише операційна система для смартфонів і планшетів, існують також:

- Android Wear для носяться пристроїв (переважно розумних годин)
- Android Auto для автомобілів.
- Android TV для розумних телевізорів.

Тобто всі пристрої на платформі Android утворюють цілу екосистему девайсів, які контактують і доповнюють один одного.

Початкову компанію-розробника Android, «Android Inc». Було засновано в Пало Алто, Каліфорнія, в жовтні 2003 року Енді Рубіном, Річчем Мінером, Ніком Сіресом та Крісом Уайтом. Рубін описував, що проект Android має «величезний потенціал у розвитку мобільних девайсів що в курсі про локацію та уподобання користувача»[2].

Спочатку проект був націлений на розробку розвинутої операційної системи для цифрових камер, але вже через 5 місяців ідеї були зсунуті на ринок мобільних телефонів і метою було перевершити популярні на той час операційні системи Symbian та Microsoft Windows Mobile.

В 2005 році Рубін проводив переговори з компаніями Samsung та HTC. Але невдовзі після них компанію придбали Google в липні 2005, і через декілька років ця покупка була названа «найкращою угодою коли-небудь». Ключові співробітники, що стояли за створенням компанії Android Inc увійшли у нову команду по розробці операційної системи.

За ядро системи було взято Linux Kernel а Google просувало операційну систему на умовах гнучкої системи з постійними оновленнями.

Перший прототип системи з'явився в грудні 2006 року – планувалося, що телефон майбутнього буде схожий на телефони від компанії BlackBerry – з повною клавіатурою, а не звичайною компактною. Але вихід нового смартфона від компанії Apple – iPhone вимусило компанію включити в документацію специфікацій підтримку тач-скрінів. Першим комерційним смартфоном став HTC Dream, вихід якого було анонсовано на 23 вересня 2008 року.

Починаючи з 2008 року, після виходу першої версії, Android отримав велику кількість оновлень, що значно покращували роботу самої операційної системи, додавали нові функції та чинили старі. Варто зазначити, що кожне велике оновлення отримувало назву в алфавітному порядку на честь десертів: Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean[3].

На даний момент Android представляє собою найпопулярнішу операційну систему, котра пропонує низку можливостей для розробників та звичайних користувачів.

Мови програмування, що підтримуються платформою Android

В даному розділі мова піде про мови програмування, на яких можна створювати додатки на платформу Android.

Почати необхідно з тих мов, на яких написаний сам Android, а це Java, C, C++. C та C++ відповідають за основу та більшість функцій системи. На мові Java був написаний вся графічна частина системи.

Через свою структуру, а саме одну з ключових концепцій системи Android – відкритий початковий код, Android включає в себе велику кількість мов, і майже кожен з них можна адаптувати під Android. Але

проблема в тому, що це має сенс лише в деяких випадках. Наприклад, коли необхідно випустити аналог вже існуючої програми на смартфоні а часу в обмаль. Тоді можна адаптувати і не торкатися оригінального коду та лише створити оболонку для взаємодії оригінального коду та смартфона. Подібні реалізації не мають сенсу при написанні додатку з нуля, тому нижче будуть перелічені лише ті мови, використання котрих має сенс.

Java

Java – строго типізована об’єктно орієнтована мова програмування спільного призначення, був розроблений компанією Sun Microsystems (пізніше їх придбали Oracle). На даний момент розробка ведеться спільноту через платформу Java Community Process[4].

Додатки на мові Java зазвичай транслюються в спеціальний байт-код, через це вони можуть працювати на майжелюбій комп’ютерній архітектурі, для котрих існує реалізація віртуальної Java-машини.

Мову активно використовують і для написання додатків під платформу Android. Однак, код транслюється в інший тип байт-коду, так як оригінальна віртуальна Java-машина не працює на Android. Замість неї фігурує віртуальна машина Dalvik (одна, починаючи з Android 5 «Lollipop» віртуальна машина була замінена на нову ART). Для такої компіляції використовується додатковий інструмент Android SDK (Software Development Kit), котрий був розроблений компанією Google.

Плюси використання Java:

1. Перша офіційна мова програмування для розроблення додатків на Android, тому має великий набір інструментів, плагінів та зовнішній бібліотек
2. Велика стандартна бібліотека, що включає в себе все необхідне для роботи

3. Простий синтаксис в купі з об'єктно орієнтованою структурою робить Java гарним вибором для написання Android додатків

Мінуси:

1. В Java досить велика кількість вимог та обмежень в плані структури коду, в тому числі і через об'єктно орієнтовану структуру
2. Велика кількість коду може суттєво вплинути на швидкість роботи додатку

Java являється дуже гарним варіантом для написання мобільних додатків, в основному через офіційну підтримку на Android та велике покриття функцій у стандартній бібліотеці.

Kotlin

Kotlin – крос-платформна, статистично типізована об'єктно орієнтована мова програмування. Була розроблена з оглядом на повну взаємодію з мовою програмування Java та віртуальною машиною Java відповідно. Окрім цього стандартні бібліотеки Kotlin повністю залежать від бібліотек класів Java (Java Class Library), але вслід цього синтаксис мови Kotlin вдалося зробити більш лаконічним і через це компіляція займає менше часу[5]. Окрім можливості компілюватися на віртуальній машині Java, Kotlin має можливість компілюватися у JavaScript[6]. Мова була розроблена компанією JetBrains[7].

У травні 2017 року на заході Google I/O було анонсовано, що Kotlin став третьою мовою програмування, що має повну програмну і апаратну підтримку на платформі Android, після Java та C++[8]. На даний момент Kotlin має велику перевагу як мова програмування, приблизно 70% всіх додатків, що поширюються через сервіс Google Play і знаходяться в топ 1000 всіх додатків використовують саме Kotlin, враховуючи і компанії Google, яка має вже понад 60 додатків, що написані або перешли на цю мову.

У травні 2019 року, Google анонсували, що Kotlin вважається рекомендованою мовою програмування для додатків на Android[9].

Плюси:

1. Kotlin нова мова програмування, яка заснована на Java. Ціллю було виправлення недоліків та збереження функцій й можливостей Java. Окрім цього Kotlin може використовувати бібліотеки та частини коду, написані на Java
2. Компактна структура дозволяє писати короткий та читабельний код на відміну від Java
3. Друга офіційна мова розробки додатків на систему Android

Мінуси:

1. Kotlin нова мова програмування, тому частина інструментів та плагінів, які були присутні з Java, так і не з'явилася
2. Не такий популярний, як Java

Як і Java, Kotlin – це дуже гарний варіант для написання додатку, в основному через його компактність, можливість використання лямбда-виразів, безпечними типами даних та покращеннями оригінальних функцій Java.

C++

C++ - високорівнева об'єктно орієнтована мова програмування, була розроблена Б'орном Струострупом як розширення мови програмування C[10].

C++ широко використовується для розробки програмних засобів, є одним з найпопулярніших мов програмування. Областю її використання є створення операційних систем, прикладних додатків, драйверів девайсів, серверів, створення комп'ютерних ігор.

На мові програмування C++ також можна писати додатки на платформу Android але лише у тандемі з іншою мовою програмування – Kotlin або Java. Це відбувається через те, що повністю компілювати код C++ на Android не ефективно, тому використовується обгортка с одних

із цих двох рідних для Android мовах. Але крім цього код на C++ можна повністю використовувати при написанні додатків на Kotlin або Java – він автоматично буде спочатку компілюватися під «рідну» мову і тільки після цього використовуватись[11].

Плюси використання:

1. Багата стандартна бібліотека
2. C++ - це один із найпопулярніших мов програмування сьогодні

Мінуси:

1. Офіційно не підтримується середовищем розробки Android Studio
 2. Не такий високий рівень абстракції, на відміну від Kotlin або Java, веде до більш громіздкого та менш читабельного коду
- C++, як і Java та Kotlin, являється гарним вибором для розробки Android додатків для тих, хто непогано володіє цією мовою програмування. Однак, відсутність підтримки з боку Android Studio та постійні проблеми з контролем пам'яті теж потрібно мати на увазі.

ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ

В першому розділі було визначено поняття Android, історія розробки даної платформи.

Було зазначено 3 найпопулярніші мови програмування, на яких рекомендують створювати мобільні додатки на платформу Android. Звісно, їх набагато більше, із популярних не зазначений мов програмування – це Python з технологіями Kivy та BeeWare, C# на фреймворкі Xamarin, JavaScript на ReactNative та NativeScript. Це можливо через політику Android – відкритий код, через це абсолютно на кожній мові програмування можна розробити технології для роботи на мобільних приладах. C++, Java та Kotlin були відокремлені і розглянуті через те, що вони признані платформою Android і працювати з ними це найкращій вихід. Були розглянуті переваги та недоліки кожної з мов програмування. Вибір мови буде та інших технологій буде розглянуто в наступному розділі.

РОЗДІЛ 2: АНАЛІЗ ТА ВИМОГИ

Аналіз ринку додатків

My Study Life[12]

Це безкоштовний додаток, що доступний на платформах Android, iOS та у браузері, яка допомагає студентам організувати свій навчальний розклад. Він дозволяє створювати розклад занять та іспитів, налаштовувати оповіщення для нагадування про заняття та завдання, а також переглядати календар заходів та створювати список завдань. Додаток має інтуїтивно зрозумілий інтерфейс і простий у використанні, тому він користується популярністю у студентів.

Розклад занять вводиться власноруч студентом спеціальну таблицю, що повністю повторює собою звичайний календар. Кожному предмету призначений свій колір для швидкої навігації, професор, котрий викладає даний предмет, аудиторія з вказанням корпусу або будівлі, де знаходиться ця аудиторія і час проведення заняття.

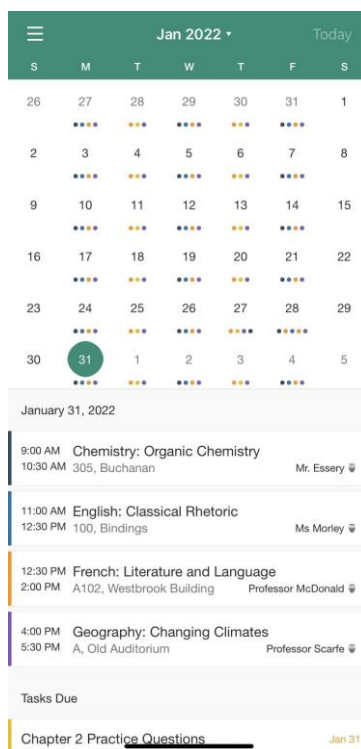


Рисунок 2.1 – Розклад занять в додатку My Study Life

На кожне заняття можна власноруч додати завдання по цьому предмету і дати йому коротку назву. Після цього створене завдання буде відображатись у іншій вкладці з вказанням предмета, назви завдання, дати на яку завдання потрібно виконати та спеціальна шкала, в якій можна вказати на який відсоток завдання виконано.

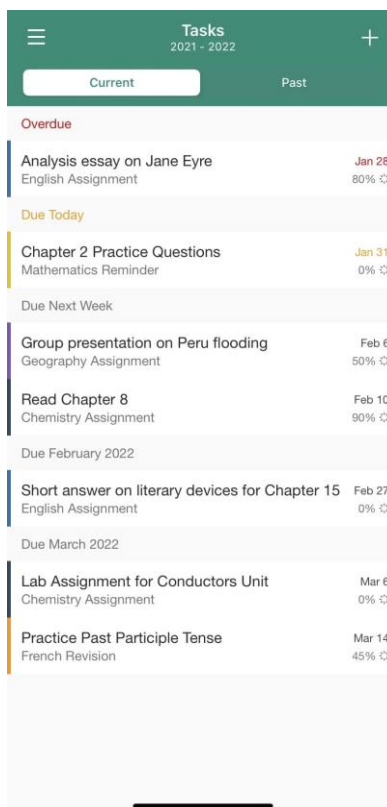


Рисунок 2.2 – Сторінка завдань

Третя і остання частина додатку – система сповіщень. Додаток буде сповіщати студента о наступній парі або тесту за певний проміжок часу до його початку. Так само можна включити сповіщення на нагадування та відображення завдань.

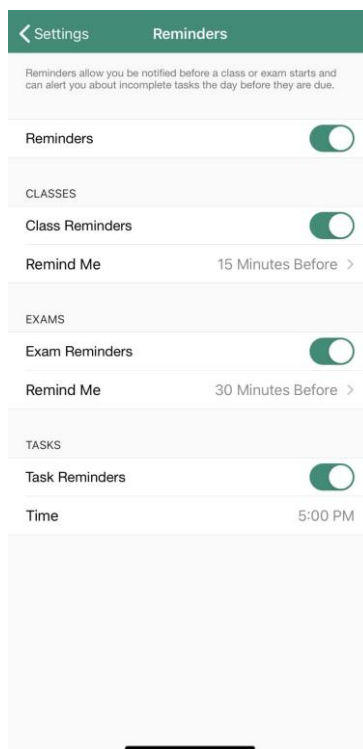


Рисунок 2.3 – Сторінка налаштування нагадувань

Timetable[13]

Не зважаючи на велику кількість подібних додатків, аналіз буде проводитись лише Timetable, так як він виграє у всіх інших додатків по функціоналу та відгукам, що доказує певний попит на ринку подібних додатків.

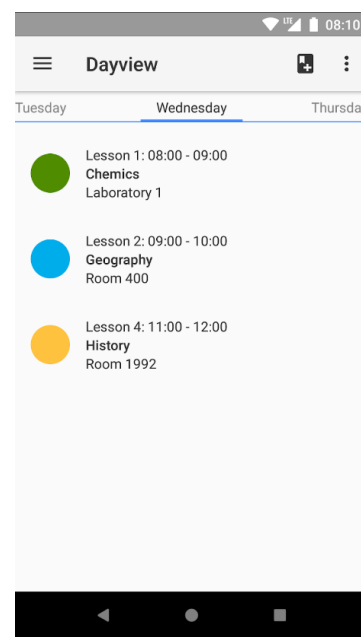
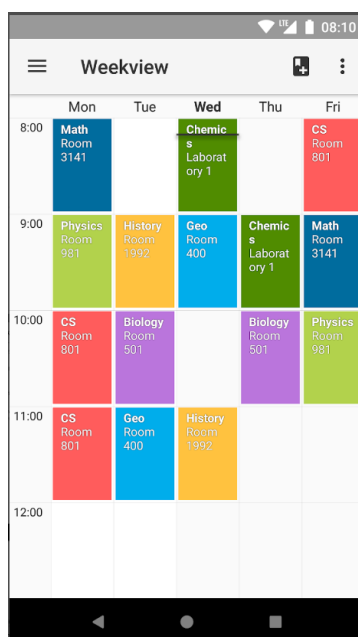


Рисунок 2.4 та 2.5 – Сторінки розкладу занять додатку Timetable[14]

Аналізуючи цей та інші незазначені додатки, наприклад Studios, можна зробити висновок, що всі вони по функціоналу походять на My Study Life.

Формування вимог до проекту

Тема проекту достатньо амбіційна і, цілком очікувано, одним розробником не може повністю бути покрита. Найчастіше, абітурієнтів та студентів умовно розділяють на дві групи, котрі не перетинаються одна з одною. Тим не менш, для початку потрібно визначитись з усіма можливими функціями, що будуть органічно вписуватись в тему додатку.

1. Розділ з необхідною інформацією. Назва говорить сама за себе – спеціальний розділ, де абітурієнт та студент зможуть дізнатися необхідну інформації. Для абітурієнта це, найважливіше, спеціальності та кафедри, на які він може подати документи для вступу
2. Розклад занять - розділ у додатку, де студенти матимуть змогу бачити свій розклад занять на поточний тиждень. Розклад має бути інтуїтивно зрозумілим та простим у використанні. Також можна додати функцію оповіщення, яка нагадуватиме студентам про майбутні заняття.
3. У доповнення розкладу – система відстеження завдань студента. Як і розклад, ця система повинна бути зрозумілою і простою у використанні.
4. Календар університетських заходів, де можна дізнатися про майбутні заходи та події, що відбуваються в університеті. Таким чином, студенти не пропустять важливих заходів, наприклад, конференції, семінари, зустрічі з роботодавцями тощо.

5. Система оповіщень та повідомлень. Розділ програми, де можна дізнатися про важливі новини, зміни в розкладі занять, дату іспитів тощо. Також можна додати функцію сповіщення у випадку, якщо викладач змінив розклад або скасував заняття.
6. Бібліотека університету - розділ, де можна дізнатися про наявність та доступність книг у бібліотеці університету. Можна додати функцію пошуку книг та резервування потрібної літератури, а також сповіщення про повернення книги до бібліотеки.
7. Карта університету - розділ, де можна дізнатися про розташування корпусів університету. Це допоможе студентам швидше орієнтуватися на території університету та не гаяти часу на пошук потрібних об'єктів.

Нажаль, як одиночному розробнику, реалізувати деякі функції буде попросту неможливо через їх складність або потреба у виділенні сервера для збереження даних. Тому, технічні вимоги до проекту виглядають наступним чином:

- Інформаційний розділ
- Розклад занять
- Завдання і система нагадування по термінам виконання
- Карта корпусів

До інших не функціональних вимог можна винести:

- Ефективність програми – додаток повинен швидко відкриватись і працювати без затримок
- Зрозумілий, не перевантажений інтерфейс програми
- Доступність – якомога більше студентів та абітурієнтів повинна мати можливість запустити додаток на своєму мобільному пристрої

- Компактність – додаток повинен не займати багато місця на мобільному пристрої
- Самостійність – навіть без доступу до мережі Інтернет додаток повинен працювати в базовому варіанті

ВИСНОВКИ ДО ДРУГОГО РОЗДІЛУ

В другому розділі було проаналізовано декілька додатків з теми створення та відстеження розкладу занять. Хоча подібних додатків досить багато на ринку, згадки заслуховує лише My Study Life як додаток, що запровадив найбільшу кількість корисних функцій, але вони можуть бути покращені

Також були розглянуті можливі функції додатку, але, через нестачу ресурсів, певних необхідних навичок, і найголовніше робочої сили, було зроблено рішення відмовитись від деякого функціоналу та зупинитися на найголовнішому – інформаційні сторінки з необхідною інформацією для студентів і абітурієнтів, розкладом занять та картою всіх корпусів університету.

.

РОЗДІЛ 3: ПРОЕКТУВАННЯ ТА РОЗРОБКА

Опис обраної технології

В першому розділі ми коротко розглянули можливі мови програмування, на яких можна писати додатки на систему Android.

Нижче наведені технології та програми, які будуть використовуватись для створення мобільного додатку.

- Мова програмування. З перерахованих раніше мов програмування було вирішено зупинитись на Kotlin з включенням Java. Рішення було прийняте через більш компактний та читабельний синтаксис мови, швидкість виконання та офіційну підтримку. Без Java, скоріш за все, не обійтись, бо можливість відсутності деякого функціоналу у Kotlin не можна ігнорувати, а так як Kotlin сумісний з бібліотеками та кодом написаними на Java, ця проблема відпадає. Досвіду у використанні Kotlin в мене значно менше, ніж у Java, але переваг у використанні Kotlin, на мою думку, більше.
- Середовище розробки. В якості варіантів середовища розробки виступають Eclipse[15], Unity[16] та Android Studio[17]. Eclipse не підходить через те, що він не підтримує Kotlin на базовому рівні а лише з підключенням додаткової надбудови, і те, що розробляти додатки на Android без іншого надбудови неможливо. Unity – це ігровий рушій, що був створений з прицілом на комп'ютерні системи, а саме Windows. Тим не менш він підтримує розробку додатків на Android за умови підключення офіційної надбудови. Більша частина моїх минулих проектів на Android я розробляв саме на Unity, і це була необхідність, ніж мій вибір. Все це через технологію, що на той час погано працювала з Android Studio, Vuforia

Engine[18], тому доводилось використовувати ігровий рушій. Інший нюанс використання Unity – це мова програмування C#. І це не є проблемою, так як досвіду з цією мовою в мене достатньо. Головна проблема Unity – це постійні критичні проблеми у побудові додатків на Android з використанням останніх версій рушія, і в придачу – великі розміри навіть найпростіших додатків. Тому, якості інтегрованого середовища розробки було обрано Android Studio. Це офіційне інтегроване середовище розробки мобільних додатків на Android з повною підтримкою як Java так і Kotlin, через те, що воно базується на іншому середовищі, рідному і розробленому спеціально під ці мови програмування - IntelliJ IDEA[19].

- Інше. Не виключно, що для проекту потрібно буде розробити додаткові матеріали, такі як іконки, кнопки, фонові зображення та інші графічні матеріали. В такому випадку їх необхідно буде створити. Для цього мною був обраний графічний редактор GIMP[20].

Розробка додатку

Додаток буде складатися з окремих вікон, котрі можна буде додатково додавати як під час розробки так і після її завершення.

Пересування по вікнам додатку буде відбуватися за допомогою бокового меню, що буде визиватись по натисканню кнопки, розташованої зверху зліва. В цьому меню повинні бути наступні пункти:

- Розклад занять
- Завдання
- Інформаційна сторінка
- Предмети
- Карта
- Налаштування

Перше, що побачить новий користувач при першому та наступних запусках додатку – це інформаційне вікно університету. На ньому буде знаходитись вся упорядкована інформація, що може знадобитись абітурієнту, що тільки почав своє знайомство з університетом, так і звичайному студенту. До неї можуть входити останні університетські новини, розділ з необхідною для абітурієнта інформацією – доступні факультети та кафедри, приблизний поріг балів для вступу на контрактну та бюджетну форми навчання, професорській состав та інша потенційно важлива інформація для вступника. Для студента там буде знаходитись актуальний розклад дзвінків та канікул, дата початку сесії і найважливіше – актуальна інформація щодо навчання за програмою подвійних дипломів, наукових виставок, курсів та лекційних занять від ІТ компаній.

Розклад занять представляє собою звичний всім календар на поточний місяць. Тиждень буде починатись з понеділка, поточний день виділяється не великою але помітною підводкою. При натисканні на будь який день знизу екрана буде відкриватись розклад на обраний день, котрий можна буде змінити або додати завдання при натисканні на відповідні кнопки. Якщо на цей день буде якесь завдання, студент це побачить нижче за розклад.

Вікно з завданнями буде представляти собою розділ, в якому студент зможе побачити завдання на поточну, наступну та інші неділі. Вони йдуть за порядком, від понеділка до неділі, у тій самій послідовності, в якій будуть проходити заняття. Якщо на певний день не буде завдання він ніяк не виділяється і просто пропускається. Окрім цього студент зможе власноруч додати завдання з цього вікна, вказавши предмет та день, в який він буде проходити. Якщо в цей день немає цієї пари, то при виборі вільного місця система автоматично зареєструє це заняття і додасть до нього вказане завдання. У кожного завдання є предмет, якому

воно належить та назва, котру обирає студент. Опціонально, можна додати короткий опис.

У вікні предметів відображаються всі дисципліни, котрі студент зможе додавати у розклад і завдання. Кожний предмет має назву, лекційного та практичного викладача, стандартну аудиторію проведення лекційних та практичних занять та можливість додати унікальний колір, щоб швидше орієнтуватись під час перегляду розкладу.

У вікно «Карта» буде вмонтований додаток Google Maps щоб студент, котрий не знає шляху зміг швидко прокласти маршрут до необхідної йому університетської будівлі, будь то інший корпус чи гуртожиток. Окрім цього на кожному будівлю буде короткий опис, як потрапити від найближчих зупинок так від головного корпусу університету.

У вікні налаштування буде варіанти роботи сповіщення о початках пар та нагадування о завданні на наступний день. Окрім цього тут можна змінити стандартні дзвінки на пари та з пар.

Сценарії використання додатку

Нижче будуть наведені деякі можливі сценарії використання додатку.

Назва сценарію	Перехід до розкладу занять
Діючі особи	Користувач, Система
Цілі	Відкрити розклад занять
Передумова	Немає
Пріоритет	Високий
Частота виконання	Часто
Успішний основний сценарій:	
1. Користувач знаходиться на будь якому екрані	

<ol style="list-style-type: none"> 2. Користувач натискає на кнопку виклику «Головного меню» додатку, що розташована зліва вверху 3. Система відкриває бокове «Головне меню» 4. Користувач обирає пункт «Розклад» в «Головному меню» 5. Система завантажує вікно «Розклад занять» 	
Результат	Користувач успішно перейшов на вікно з розкладом занять

Назва сценарію	Заповнити розклад
Діючі особи	Користувач, Система
Цілі	Занести або змінити інформацію щодо пар, що будуть проводитись на цьому тижні
Передумова	Поточний екран – Розклад занять
Пріоритет	Високий
Частота виконання	Раз на тиждень
Успішний основний сценарій:	
<ol style="list-style-type: none"> 1. Користувач знаходиться на екрані з розкладом занять 2. Користувач обирає необхідний йому день неділі 3. Система відкриває режим детального перегляду обраного дня 4. Користувач натискає кнопку «Змінити» 5. Система відкриває вікно внесення змін до розкладу 6. Користувач обирає необхідну йому час проведення заняття <ol style="list-style-type: none"> a. За необхідністю користувач може змінити час проведення заняття 7. Система відображає все предмети, що можуть бути внесені до розкладу на поточний день 8. Користувач обирає необхідний йому предмет 	

<p>а. За необхідністю користувач може змінити аудиторію проведення або посилання на дистанційну конференцію</p> <p>9. Система заносить обраний предмет на раніше визначений час</p>	
Результат	Користувач успішно створив або вніс зміни до розкладу
Розширення	
7а	Якщо доступних предметів для занесення немає Викликається сценарій «Створення предмета» з переходом на крок 7

Назва сценарію	Створення предмета
Діючі особи	Користувач, Система
Цілі	Внесення інформації щодо нового предмета
Передумова	Немає
Пріоритет	Високий
Частота виконання	Рідко
Успішний основний сценарій:	
<ol style="list-style-type: none"> 1. Користувач знаходиться на будь якому екрані 2. Користувач натискає на кнопку виклику «Головного меню» додатку, що розташована зліва вверху 3. Система відкриває бокове «Головне меню» 4. Користувач обирає пункт «Предмети» в «Головному меню» 5. Система завантажує вікно «Предмети» 	

<p>6. Користувач обирає пункт «Додати новий предмет»</p> <p>7. Система відкриває вікно «Створення нового предмету»</p> <p>8. Користувач вводить:</p> <p style="padding-left: 40px;">а. Назву предмета</p> <p style="padding-left: 40px;">б. Викладача предмета (ПІБ)</p> <p style="padding-left: 40px;">с. Аудиторію в якій буде проводитись предмет, або посилання на дистанційне заняття</p> <p>9. Користувач натискає на кнопку «Зберегти»</p> <p>10. Система зберігає внесену інформацію</p>	
Результат	Користувач успішно створив новий предмет, котрий можна буде внести у розклад занять
Розширення	
ба	Користувач обирає вже існуючий предмет Відкривається вікно зміни обраного предмета Перехід на крок 8
9а	Поле назви предмету не заповнено Перехід на крок 8а з підказкою «Заповніть це поле»

Назва сценарію	Перехід до вікна завдань
Діючі особи	Користувач, Система
Цілі	Відкрити вікно з поточними завданнями
Передумова	Немає
Пріоритет	Середній
Частота виконання	Часто

Успішний основний сценарій:	
<ol style="list-style-type: none"> 1. Користувач знаходиться на будь якому екрані 2. Користувач натискає на кнопку виклику «Головного меню» додатку, що розташована зліва вверху 3. Система відкриває бокове «Головне меню» 4. Користувач обирає пункт «Завдання» в «Головному меню» 5. Система завантажує вікно «Завдання» 	
Результат	Користувач успішно перейшов на вікно з поточними завданнями

Назва сценарію	Створення завдання – Розклад
Діючі особи	Користувач, Система
Цілі	Занести інформацію щодо завдання по предмету через вікно розкладу
Передумова	Поточне вікно – Розклад занять
Пріоритет	Високий
Частота виконання	Часто
Успішний основний сценарій:	
<ol style="list-style-type: none"> 1. Користувач знаходиться на екрані з розкладом занять 2. Користувач обирає необхідний йому місяць, тиждень та день 3. Система відкриває режим детального перегляду обраного дня 4. Користувач натискає кнопку «Додати завдання» напроти необхідного предмета 5. Система відкриває вікно «Створення завдання» 6. Користувач вводить назву завдання 7. Користувач вводить короткий опис завдання 8. Система зберігає внесену інформацію 	

Результат	Користувач успішно вніс інформацію про поточне завдання
Розширення	
ба	Поле назви завдання не заповнено Перехід на крок 6 з підказкою «Заповніть це поле»

Назва сценарію	Створення завдання – Завдання
Діючі особи	Користувач, Система
Цілі	Занести інформацію щодо завдання по предмету через вікно завдань
Передумова	Поточне вікно – Завдання
Пріоритет	Низький
Частота виконання	Рідко
Успішний основний сценарій:	
<ol style="list-style-type: none"> 1. Користувач знаходиться на екрані з завданнями 2. Користувач натискає на кнопку «Додати нове завдання» 3. Система відкриває вікно створення завдання 4. Користувач обирає потрібний предмет 5. Користувач вводить назву нового завдання 6. Користувач вводить короткий опис завдання 7. Користувач вводить день здачі роботи 8. Система зберігає внесену інформацію 	
Результат	Користувач успішно вніс інформацію про поточне завдання
Розширення	
4а	Якщо доступних предметів для створення завдань немає

	Викликається сценарій «Створення предмета» з переходом на крок 7
5a	Поле назви завдання не заповнено Перехід на крок 5 з підказкою «Заповніть це поле»
7a	Поле дня здачі завдання не заповнено Перехід на крок 7 з підказкою «Заповніть це поле»

ВИСНОВКИ ДО ТРЕТЬОГО РОЗДІЛУ

В третьому розділі були описані мови програмування та інші технології, що використовувались для створення додатку. Були наведені причини їх вибору. Так, основною мовою програмування став Kotlin через компактність та швидкість роботи, та Java, як запасний варіант у випадку, якщо Kotlin буде недостатньо. В якості IDE виступає Android Studio з тієї причини, що це середовище є офіційним для розробки мобільних додатків під Android.

Був наведений фінальний вигляд додатку, з частковим описом всіх його екранів. Окрім цього були створенні Use-Case сценарії, що детально описують роботу деяких модулів додатку, варіанти подій при різних випадках та взаємодії між ними.

ВИСНОВКИ

Були надані визначення та висвітлена історія появи такої операційної системи як Android. Були зазначені 3 найголовніші мови програмування, на котрих можна побудувати мобільний додаток: Kotlin в середовищах Android Studio та Eclipse, Java, котра активно підтримується в тих самих інтегрованих середовищах, і C++, котра підтримується в Android Studio.

Були проаналізовані додатки, схожою за темою для позначення і виявлення потенційного функціоналу та його можливої реалізації. Окрім функціоналу і модулів зверталась увага на інтерфейс, з котрим користувач буде взаємодіяти. На основі цього аналізу було виділено низку можливих функцій, котрі можна буде розробити в майбутньому:

- Розділ з необхідною інформацією
- Розклад занять
- Система відстеження завдань студента
- Календар університетських заходів
- Система оповіщень та повідомлень якщо викладач змінив розклад або скасував заняття
- Бібліотека університету
- Карта університету

На основі аналізу можливих мов програмування була обрана основна, проведений невеликий аналіз інтегрованих середовищ розробки на платформу Android.

Був детально описаний вид додатку, розписаний функціонал мобільного додатку. Взаємодія модулів була описана завдяки сценаріям використання Use-Case. На основі цього був розроблений прототип додатку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is Android | Android. URL: <https://www.android.com/what-is-android/>
2. Ben Elgin, Google Buys Android for Its Mobile Arsenal. URL: https://web.archive.org/web/20110205190729/http://www.businessweek.com/technology/content/aug2005/tc20050817_0949_tc024.htm
3. Murali K Menon, Android Nougat: Here's why Google names the OS after sweets. URL: <https://indianexpress.com/article/lifestyle/food-wine/from-donut-to-nougat-why-are-android-versions-named-after-sweets-2887237/>
4. Java Software | Oracle. URL: <https://www.oracle.com/java/>
5. Kotlin for Android | Kotlin Documentation. URL: <https://kotlinlang.org/docs/android-overview.html>
6. Releases JetBrains/kotlin. URL: <https://github.com/JetBrains/kotlin/releases>
7. Kotlin Programming Language. URL: <https://kotlinlang.org/>
8. Frederic Lardinois, Google makes Kotlin a first-class language for writing Android apps. URL: <https://techcrunch.com/2017/05/17/google-makes-kotlin-a-first-class-language-for-writing-android-apps/>
9. Frederic Lardinois, Kotlin is now Google's preferred language for Android app development. URL: <https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/>
10. Bjarne Stroustrup, The C Programming language. URL: https://archive.org/details/cprogramminglang00stro_0

11. Cross-platform mobile development with C++ | Microsoft Learn.
URL: <https://learn.microsoft.com/en-gb/cpp/cross-platform/visual-cpp-for-cross-platform-mobile-development?view=msvc-170>
12. My Study Life – Online Student Planner. URL:
<https://mystudylife.com/>
13. Timetable. URL: <https://timetableapp.net/>
14. Timetable – App in Google Play. URL:
<https://play.google.com/store/apps/details?id=com.gabrielittner.timetable>
15. The Community for Open Innovation and Collaboration | The Eclipse Foundation. URL: <https://www.eclipse.org/>
16. Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine. URL: <https://unity.com/>
17. Download Android Studio & App Tools – Android Developers.
URL: <https://developer.android.com/studio>
18. Home | Vuforia EDP. URL: <https://developer.vuforia.com/#>
19. IntelliJ IDEA – the Leading Java and Kotlin IDE. URL:
<https://www.jetbrains.com/idea/>
20. GIMP – GNU Image Manipulator Program. URL:
<https://www.gimp.org/>

ДОДАТКИ

ДОДАТОК А. КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Гриценко Денис Сергійович, учасник освітнього процесу Херсонського державного університету, **УСВІДОМЛЮЮ**, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу.

ЗАЯВЛЯЮ, що у своїй освітній і науковій діяльності **ЗОБОВ'ЯЗУЮСЯ**:

- дотримуватися:
 - вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
 - принципів та правил академічної доброчесності;
 - нульової толерантності до академічного плагіату;
 - моральних норм та правил етичної поведінки;
 - толерантного ставлення до інших;
 - дотримуватися високого рівня культури спілкування;
 - надавати згоду на:
 - безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
 - оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
 - використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;
 - самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;
 - надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;
 - не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;
 - своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;
 - не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;
 - підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;
 - поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;
 - не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;
 - відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науководслідницькі завдання;
 - запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;
 - не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;
 - не підроблювати документи;
 - не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;
 - не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;
 - не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;
 - не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;
 - не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;
 - не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягати власних корисних цілей;
 - не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.
- УСВІДОМЛЮЮ**, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.

12.09.2019

(дата)



(підпис)

Гриценко Денис

(ім'я, прізвище)