

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РОЗРОБКА ANDROID-ДОДАТКУ ДЛЯ ІНВЕНТАРИЗАЦІЇ

DEVELOPMENT OF AN ANDROID APPLICATION FOR INVENTORY

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗ-42

Кочура Артем Володимирович


(підпис)

Керівник:

Потейчук Михайло Іванович


(підпис)

Кваліфікаційну роботу
допущено до захисту

«8» 06 2024р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна



Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

М. В. Мисирко

« 2 » 02 202__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Кочурі Артему Володимировичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка Android-додатку для інвентаризації»

Керівник роботи: Потейчук М. І.

затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня 2024 р. _____

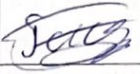
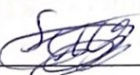
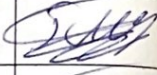
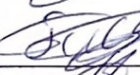
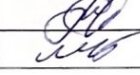
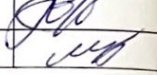
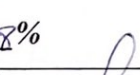
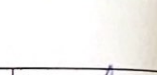
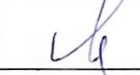
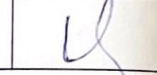
3. Вихідні дані до роботи:

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

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.

5. Перелік графічного матеріалу:

6. Консультанти розділів роботи

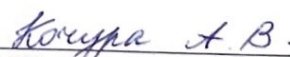
Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Потейчук М. І.		
Специфікація вимог до розробленої системи	Потейчук М. І.		
Розробка об'єкта проектування	Потейчук М. І.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту	4,8%		
Академічна доброчесність	Яцук А. А.		

7. Дата видачі завдання «2» лютого 2024 р.

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 07.02.2024 р.	Вик.
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 27.02.2024 р.	Вик.
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 12.03.2024 р.	Вик.
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 17.04.2024 р.	Вик.
5	Практична реалізація об'єкта проектування та розробка бази даних	до 18.05.2024 р.	Вик.
6	Тестування та налагодження об'єкта проектування	до 01.06.2024 р.	Вик.
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 05.06.2024 р.	Вик.

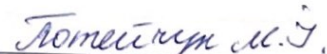
Здобувач вищої освіти


(підпис)


(прізвище, ініціали)

Керівник кваліфікаційної роботи


(підпис)


(прізвище, ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

Рецензія
на кваліфікаційну роботу

Здобувач вищої освіти: Кочура Артем Володимирович

Тема: «Розробка Android-додатку для інвентаризації»

Коротка характеристика кваліфікаційної роботи. Кваліфікаційна робота представляє собою добре структуровану, детально обгрунтовану та професійно виконану роботу, яка орієнтована на розробку Android-додатку для інвентаризації.

Самостійні розробки і пропозиції автора. Розроблений мобільний додаток для інвентаризації з використанням QR-кодів є ефективним інструментом, що дозволяє значно оптимізувати процеси інвентаризації в організаціях. Його використання забезпечує швидкість, точність та зручність проведення інвентаризації, зменшуючи кількість помилок та витрат часу. Практична значимість мобільного додатку для інвентаризації полягає в тому, що розроблений мобільний додаток для інвентаризації може використовуватися співробітниками будь-якої організації, які є матеріально відповідальними особами.

Недоліки. Істотних недоліків в роботі не виявлено.

Загальний висновок. Робота виконана на високому фаховому рівні, має логічну та послідовну структуру, відповідає встановленим вимогам, може бути представлена до захисту екзаменаційній комісії та заслуговує позитивної оцінки.

Рецензент: Здобичька Ніна Василівна, доцент каф. КН
(прізвище, ім'я, по батькові, посада)

Рецензент кваліфікаційної роботи _____
(підпис)

« 8 » 06 2024 р.

Міністерство освіти і науки України
Луцький національний технічний університет

Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
Здобувач вищої освіти Кочура Артем Володимирович
група ІПЗ-42

Тема кваліфікаційної роботи: Розробка Android-додатку для інвентаризації

Керівник: Потейчук Михайло Іванович

Актуальність теми. Актуальність теми мобільного додатку для інвентаризації зумовлена його здатністю значно покращити ефективність і точність інвентаризаційних процесів у будь-якій організації. Такий додаток може використовуватися матеріально відповідальними особами, полегшуючи їхню роботу та знижуючи ймовірність помилок. Завдяки цьому, організації можуть оптимізувати витрати часу і ресурсів, що підвищує загальну продуктивність і забезпечує більш точний облік матеріальних цінностей.

Об'єктом дослідження є процес інвентаризації в організації.

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи. Розроблений мобільний додаток для інвентаризації з використанням QR-кодів є ефективним інструментом, який значно оптимізує інвентаризаційні процеси в організаціях. Завдяки додатку, процеси інвентаризації стають швидшими, точнішими та зручнішими, що знижує кількість помилок та витрати часу. Практична користь додатку підтверджена результатами тестування та позитивними відгуками користувачів.

Зауваження та недоліки: істотних недоліків не виявлено.

Загальний висновок робота виконана на високому рівні та заслуговує позитивної оцінки _за умови успішного захисту.

Керівник


(підпис)

(Потейчук М. І.)
(прізвище, ім'я, по батькові)

« 7 » 06 2024 р.

АНОТАЦІЯ

Кочура А. В. Розробка Android-додатку для інвентаризації. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Метою роботи є теоретичне обґрунтування та розробка мобільного додатку для інвентаризації з використанням QR-кодів. Під час розробки мобільного додатку було вивчено основи мови програмування Kotlin. Детально вивчаються скрипти Google, а також взаємодія клієнта програми з Android API. За результатами кваліфікаційної роботи було розроблено мобільний додаток для інвентаризації.

Ключові слова: інформаційні технології, Android, мобільний додаток, мобільна розробка, Kotlin.

ABSTRACT

Kochura A. V. Development of an Android application for inventory. Manuscript. Qualification work of the bachelor's program "Software engineering" in the specialty "Software engineering". Lutsk National Technical University. Lutsk, 2024.

The purpose of the work is the theoretical justification and development of a mobile application for inventory using QR codes. During the development of the mobile application, the basics of the Kotlin programming language were studied. Google scripts are studied in detail, as well as the interaction of the application client with the Android API. Based on the results of the qualification work, a mobile application for inventory was developed.

Keywords: information technology, Android, mobile application, mobile development, Kotlin.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	12
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЕНОЇ СИСТЕМИ	14
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	14
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання ...	20
РОЗДІЛ 3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКА	26
3.1 Практична реалізація об'єкта проектування	26
3.2 Тестування мобільних додатків	32
3.3 Інструкція користувача.....	37
ВИСНОВКИ.....	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	42

ВСТУП

Актуальність теми. В даний час мобільні технології міцно увійшли у виробничий цикл будь-якого підприємства, дозволяють полегшити роботу, удосконалити процес обробки інформації, її накопичення і зберігання.

«Інвентаризація – це перевірка наявності майна організації і стану її фінансових зобов’язань на певну дату шляхом зіставлення фактичних даних з даними бухгалтерського обліку» [1].

Процес інвентаризації може бути полегшений впровадженням на підприємстві комп’ютерних програм. У наш час більшість повсякденних процесів можна автоматизувати. Інвентаризація також не є винятком у процесі автоматизації. Тому дуже важливо використовувати в процесі автоматизації мобільні інструменти (телефон або планшет). Це підвищить швидкість інвентаризації, точність, а також додасть можливість використовувати віддалений доступ до сховища (таблиці) з будь-якого пристрою.

Метою роботи є теоретичне обґрунтування та розробка мобільного додатку для інвентаризації з використанням QR-кодів.

Об’єктом дослідження є процес інвентаризації в організації.

Предметом дослідження є процес розробки мобільного додатку для інвентаризації з використанням QR-кодів.

Мобільний додаток дозволить користувачам сканувати QR-коди майна організації та проводити інвентаризацію.

Мета, об’єкт і предмет дослідження зумовили постановку і вирішення наступних завдань:

- провести аналіз домену та переглянути існуючі мобільні рішення;
- визначити функціональні вимоги до розроблюваного мобільного додатку;

- розробити мобільний додаток для інвентаризації для операційної системи Android;
- виконати апробацію та тестування.

Практична значимість мобільного додатку для інвентаризації полягає в тому, що розроблений мобільний додаток для інвентаризації може використовуватися співробітниками будь-якої організації, які є матеріально відповідальними особами.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз сучасного стану проблеми

Для того, щоб приступити до проектування мобільного додатку, необхідно визначитися з предметною областю. В якості предметної області був обраний процес інвентаризації в ЛНТУ.

Основним напрямком діяльності в ЛНТУ є підготовка студентів за багатьма спеціальностями та освітніми програмами. Щороку відбувається зарахування студентів на кожну спеціальність на основі конкурсного відбору. На кожного студента створюється особова справа, яка зберігається протягом усього періоду навчання.

Інвентаризація проводиться з метою точного виявлення та обліку всіх наявних ресурсів, таких як будівлі, обладнання, комп'ютери, меблі, навчальні посібники та інші матеріальні цінності. Це дозволяє адміністрації університету мати повне уявлення про свої активи та ефективно планувати їх використання.

Інвентаризація – це процес обліку основних засобів підприємства і товарно-матеріальних цінностей. З метою проведення інвентаризації збирається інвентаризаційна комісія, яку очолює головний бухгалтер організації.

Метою інвентаризації є фактичне виявлення наявності або відсутності того чи іншого майна в організації відповідно до реєстрів бухгалтерського обліку. Порядок проведення інвентаризації, її терміни визначаються самим підприємством.

Інвентаризація є обов'язковою в наступних випадках:

- якщо майно підприємства підлягає оренді або підлягає продажу, викупу та реорганізації підприємства;
- до складання річної фінансової звітності;
- у разі зміни матеріально відповідальних осіб;
- у разі крадіжки, зловживання або псування майна;

- у разі стихійного лиха, пожежі або інших надзвичайних ситуацій, спричинених екстремальними умовами;

- у разі ліквідації (реорганізації) організації до складання ліквідаційного (відокремлювального) балансу.

На даний момент процес інвентаризації протікає наступним чином:

- ручний друк всього переліку номенклатури на папері;
- безпосередньо на робочих місцях здійснюється візуальний пошук у друкованому списку та робиться відмітка про наявність номенклатури;
- ручне створення та заповнення інвентаризаційної документації, подальша звірка отриманих результатів.

Розглянемо додатки, пов'язані з предметною областю.

Сьогодні існує багато різних програм для Android, які дозволяють сканувати QR-коди. Одним з таких додатків є «Сканер штрих-кодів». Особливістю цієї програми є можливість перегляду відсканованого QR-коду. Крім того, в додатку передбачений пошук в інтернеті об'єкта, що сканується, а також можливість поділитися вмістом QR-коду. Недоліками цієї програми є велика кількість реклами, повільне фокусування камери. Також ця програма не має бази даних, де зберігаються відскановані дані. Скріншот головного екрана програми показаний на рисунку 1.1.

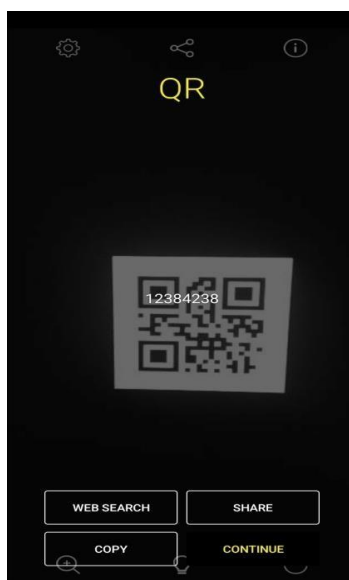


Рисунок 1.1 – Екран програми «Сканер штрих-кодів» [2]

Ще одна програма для сканування штрих-кодів і QR-кодів – «QR і сканер штрих-кодів». Після сканування результат сканування, а саме номер посилання або штрих-коду, автоматично копіюється в буфер обміну, після чого користувачеві надається можливість знайти сканований об'єкт в інтернеті [2].

Перевагами цього додатка є можливість використання спалаху пристрою для підсвічування штрих-коду і пошуку сканованого об'єкта в інтернеті. До недоліків можна віднести велику кількість реклами та відносно повільне сканування штрих-кодів та QR-кодів. Робоче середовище сканера QR та штрих-кодів показано на рисунку 1.2.



Рисунок 1.2 – Екран програми QR і Barcode Scanner [2]

Тепер, коли ми розглянули програми, які реалізують використання QR-кодів, перейдемо до другого типу програм, пов'язаних із процесом інвентаризації.

Перейдемо до розбору програми «Штрих-коди та інвентаризація та Excel», яка вже використовується для завдання інвентаризації. Після сканування QR-код, можна вибрати деякі параметри: скільки товарів є в наявності, розташування, ціна. У додатку є можливість пошуку даних в електронній таблиці Excel. В цілому цей додаток хороший, але головним недоліком є те, що він

платний (безкоштовний пробний період становить 60 хвилин, далі доведеться заплатити) [2]. Екрани «Штрих-коди та інвентаризація» та «Excel» показані на рисунку 1.3.

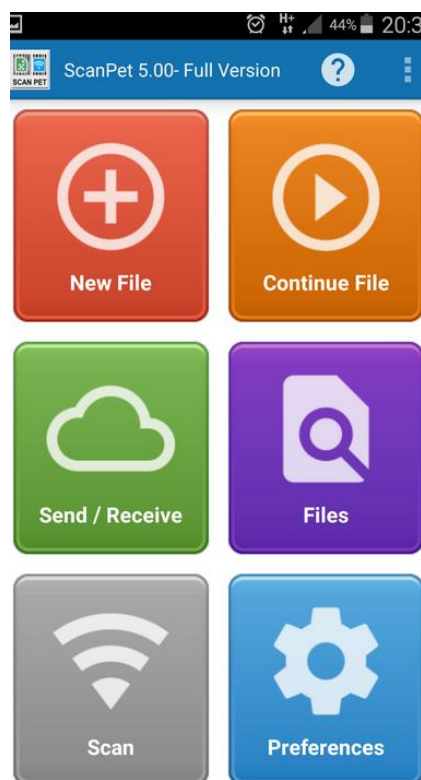


Рисунок 1.3 – Екрани штрих-кодів та інвентаризації та Excel [3]

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Виходячи із мети роботи сформулюємо такі завдання дослідження:

- аналіз предметної області та огляд сучасних методів інвентаризації. Провести дослідження існуючих методів інвентаризації, включаючи ручні методи, використання штрих-кодів та RFID-технологій, проаналізувати переваги та недоліки кожного методу з точки зору точності, швидкості та витрат;
- аналіз існуючих мобільних додатків для інвентаризації. Знайти та вивчити популярні мобільні додатки для інвентаризації, визначити основні функції, які вони надають, та їх користувацький інтерфейс;

- визначення функціональних вимог основні функціональні можливості, інтерфейс користувача;
- розробка мобільного додатку для Android. Архітектура додатку, реалізація функціоналу сканування QR-кодів, інтеграція бази даних;
- апробація та тестування додатку.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Для аналізу наявних мобільних додатків для інвентаризації виділимо найбільш конкретні вимоги до мобільного додатку для інвентаризації.

Вагомою перевагою мобільного додатку є його доступність для необмеженої кількості користувачів та економія коштів при його використанні.

Ще однією перевагою мобільного додатку є те, що для перевірки не потрібні сторонні інвентаризаційні пристрої (сканери тощо).

Крім того, додаток повинен містити широкий спектр функцій.

Отже, сформулюємо такі вимоги до мобільного додатку:

- програмне забезпечення має вільно розповсюджуватися;
- в якості бази даних слід використовувати Google Таблиці (Google Sheet).
- у додатку є можливість пошуку даних в Google Sheets (Google table);
- як сканер QR-коду використовується камера мобільного телефону;
- програмний продукт повинен мати можливість переглядати раніше відскановані дані.

Аналіз існуючих рішень показав, що розглянуті дані заявки не відповідають обраним вимогам. Після проведеного аналізу було прийнято рішення про створення власного мобільного додатку, в якому враховані перераховані вище вимоги (рис. 2.1).

На основі аналізу літературних джерел опишемо функціональні можливості програми «Інвентаризація», що розробляється:

а) основний функціонал:

- можливість сканування QR-коду;
- можливість додавати дані вручну;
- можливість перегляду даних;

б) додатковий функціонал (не обов'язковий для впровадження):

- експорт даних у файл;
- можливість пошуку відсканованих даних.

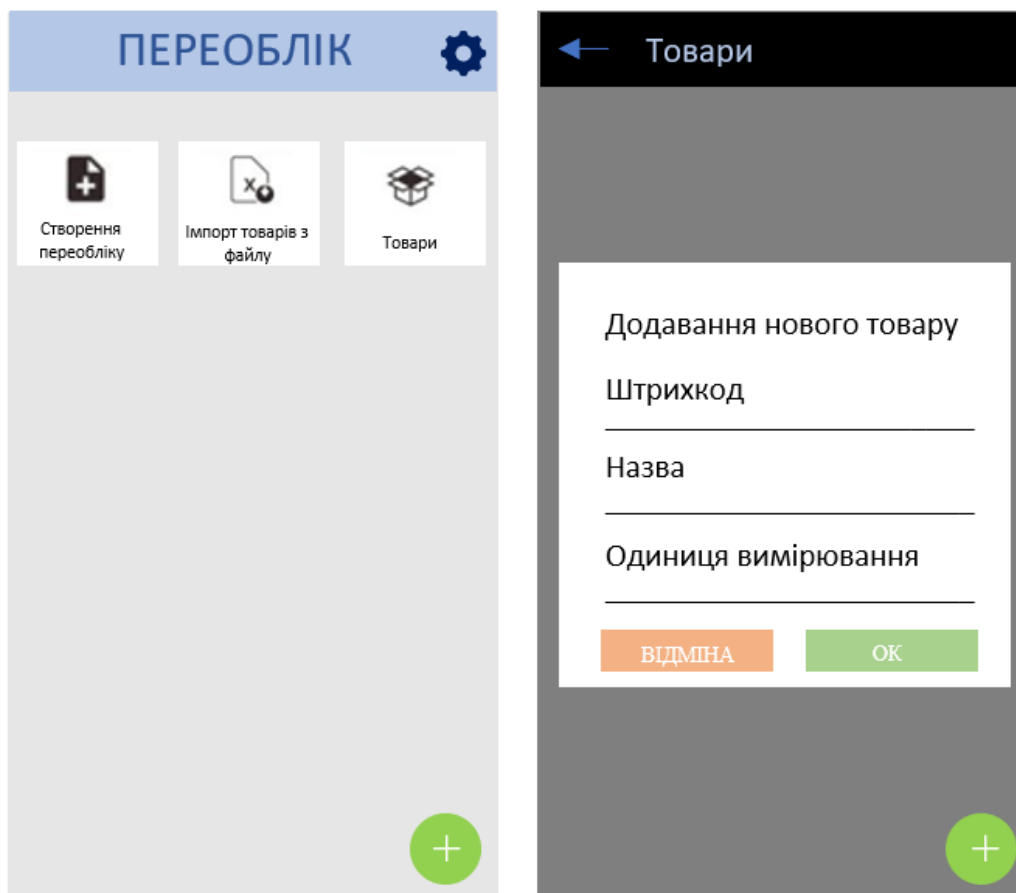


Рисунок 2.1 – Екран програми інвентаризації

Також необхідно розглядати наступні вимоги до інтерфейсу користувача:

- елементи повинні відображатися коректно;
- інтерфейс користувача повинен адаптуватися під різну роздільну здатність екрана.

Програмне рішення має коректно працювати на різних пристроях під операційною системою Android, починаючи з версії 5.0 і вище.

Процес інвентаризації в університеті представлений у вигляді діаграми IDEF0.

IDF0, або Integration Definition for Function Modeling – це техніка моделювання, яка використовується для картографування процесів, процедур і діяльності в організації. Він включає в себе графічну мову, яка служить для передачі конкретної інформації і забезпечує розуміння і аналіз процесів. Діаграма запасів IDEF0 показана на рисунку 2.2 нижче.

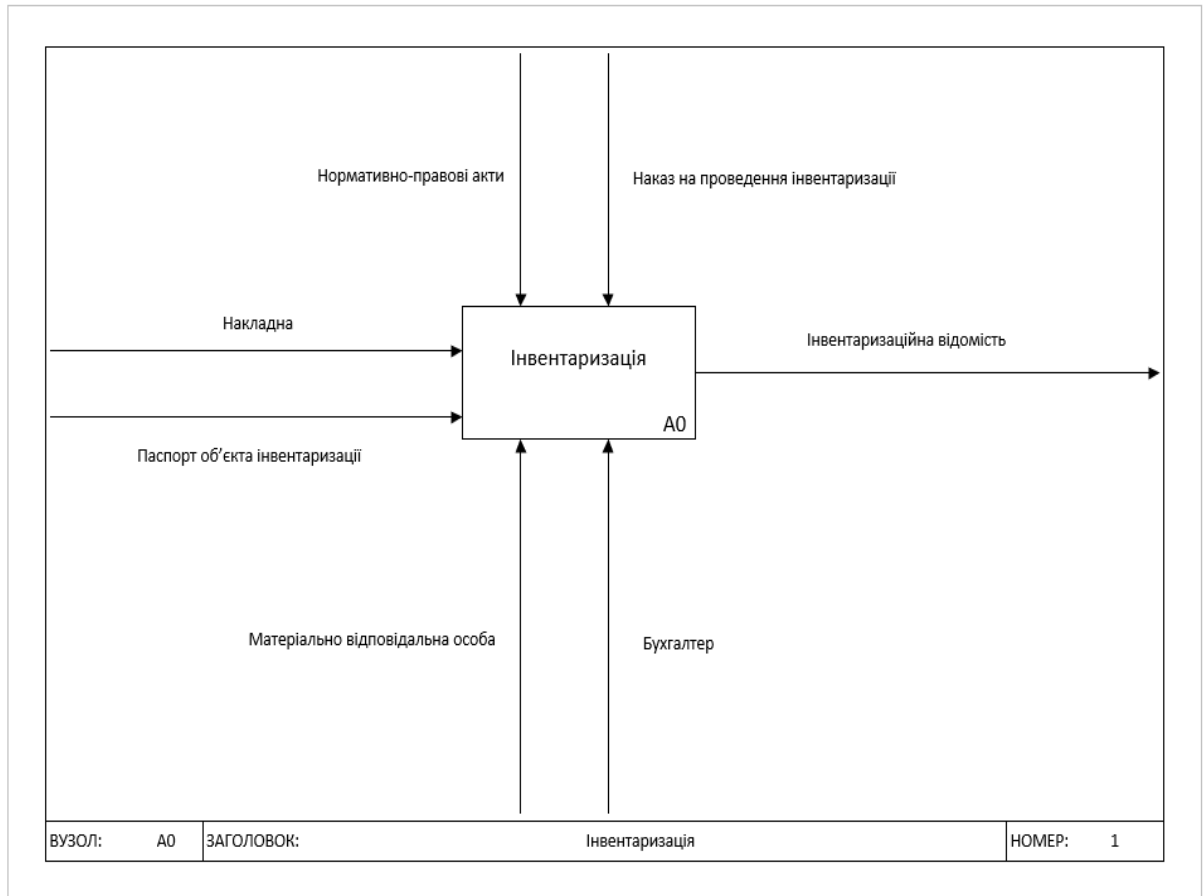


Рисунок 2.2 – Бізнес-процес інвентаризації (базовий рівень)

Проведемо декомпозицію процесу інвентаризації. Розкладання показано на рисунку 2.3.

Таким чином, розіб'ємо процес інвентаризації на блоки, що дозволить нам визначити процеси, які ми хочемо оптимізувати. Для нашого проекту нас цікавлять лише блоки «Пошук інвентарного номера на об'єкті та у формі» та блок «Внесення інвентарного номера у форму».

Також необхідно вибрати модель дизайну мобільного додатку.

Ми обрали покрокову модель дизайну для розробки нашого мобільного додатка для інвентаризації. Цей підхід передбачає розбиття процесу розробки на невеликі кроки або ітерації.

Кожен крок включає повний цикл розробки, від аналізу вимог і проектування до розробки, тестування та впровадження певної частини функціоналу програми.

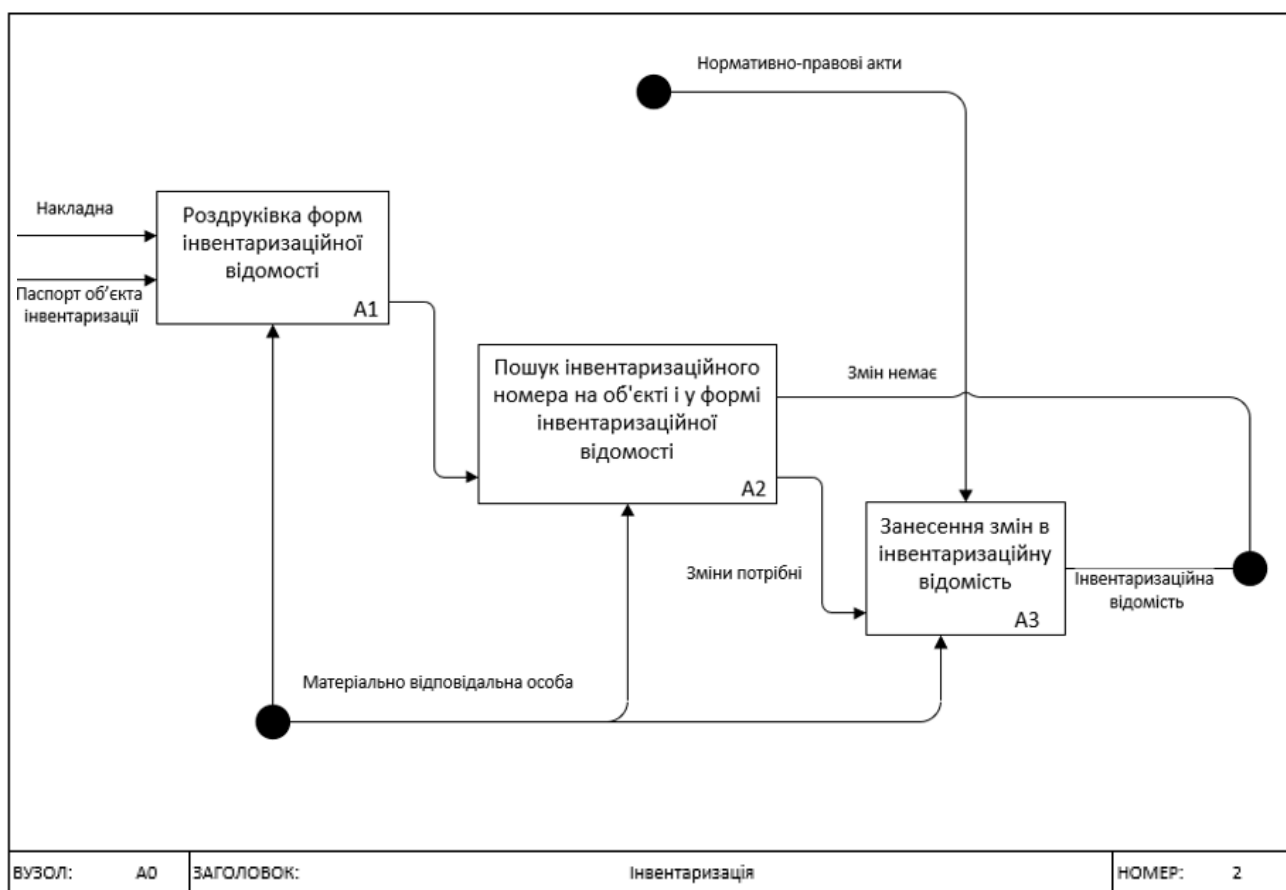


Рисунок 2.3 – Декомпозиція процесу інвентаризації

Інкрементальна модель має ряд переваг:

- швидке застосування. Розділивши розробку на менші частини, кожен крок можна реалізувати швидше. Це дозволяє отримувати зворотний зв'язок від користувачів раніше і швидко вносити зміни;

- гнучкість. Інкрементальна модель дозволяє гнучко реагувати на зміну вимог або побажань користувача. Якщо є необхідність змінити функціональність, це можна легко додати до наступного приросту;

- краща простежуваність. кожен крок має свої вимоги та функціональність, що полегшує відстеження та управління процесом розробки. Кожен етап розробки фокусується на певній частині програми;

- раннє виявлення проблем. Завдяки частим ітераціям та тестуванню кожного інкременту можна швидко виявити та виправити проблеми або помилки у функціональності.

У випадку з мобільним додатком для інвентаризації, кожен приріст може бути призначений для розробки конкретних модулів або функцій.

Блок-схема мобільного додатку. Блок-схема – це тип графічної моделі, яка зображує набір основних компонентів і взаємозв'язки між ними.

Блок-схема необхідна в мобільному додатку для інвентаризації з наступних причин:

- розуміння структури даних – блок-схема допомагає розробникам додатків і користувачам зрозуміти, як організовані дані інвентаризації та як вони пов'язані один з одним. Це дозволяє ефективно працювати з даними та забезпечує наочність для користувачів програми.

- ідентифікація компонентів – структурна схема допомагає ідентифікувати основні компоненти інвентаризації та зрозуміти їх взаємозв'язки. Наприклад, можна визначити категорії товарів, окремі товари запасів і взаємозв'язки між ними. Це полегшує навігацію та керування даними.

Основні функції, які повинні бути реалізовані в додатку, впливають з вимог до розроблюваного додатку.

На рисунку 2.4 представлена структурна схема програми, зображена у вигляді ключових модулів і взаємозв'язків між ними. Ця діаграма дає чітке уявлення про те, як вищезгадані вимоги реалізовані в програмі.

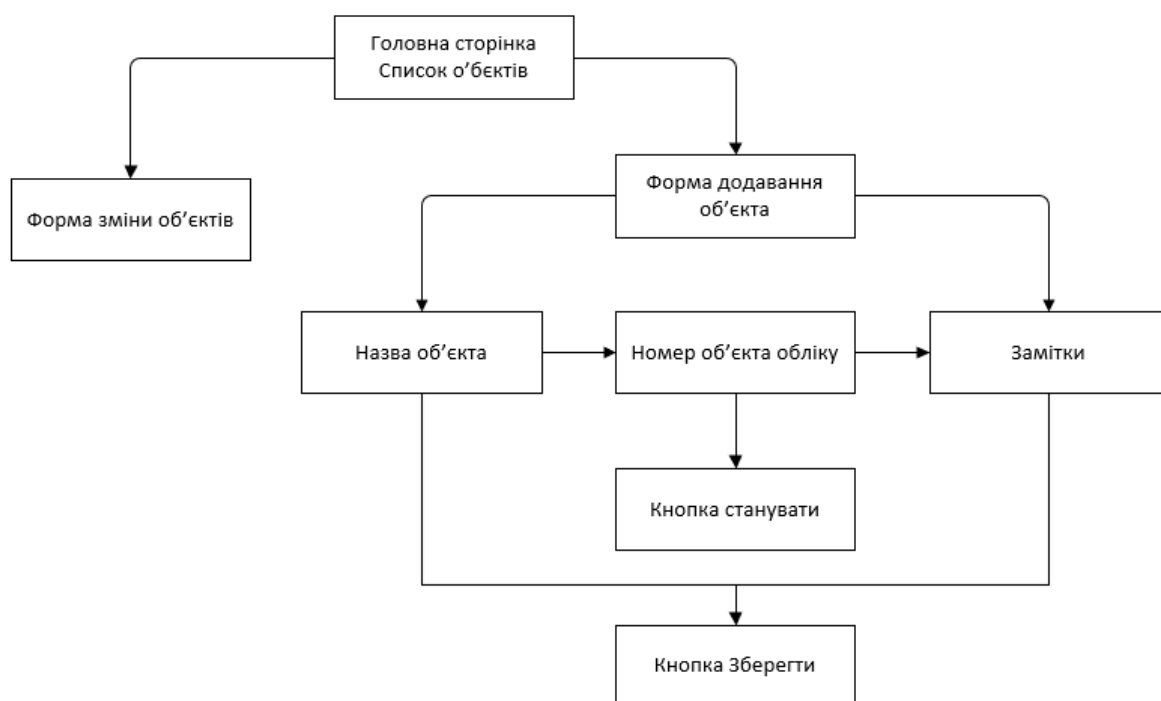


Рисунок 2.4 – Блок-схема застосування

Для подальшого розвитку потрібно побудувати компонентну діаграму.

Схема компонентів мобільного додатку наведена на рисунку 2.5 і складається з наступних компонентів:

- мобільний додаток;
- API;
- Google Таблиці.

База даних представлена у вигляді таблиці Google Sheets, яка встановлює зв'язок з мобільним додатком через API.

У мобільній розробці API (Application Programming Interface) – це набір функцій, методів і протоколів, що надаються операційною системою (iOS, Android тощо) або сторонніми сервісами, які дозволяють розробникам створювати додатки для мобільних пристроїв і взаємодіяти з різними функціональними можливостями пристрою або зовнішніми сервісами.

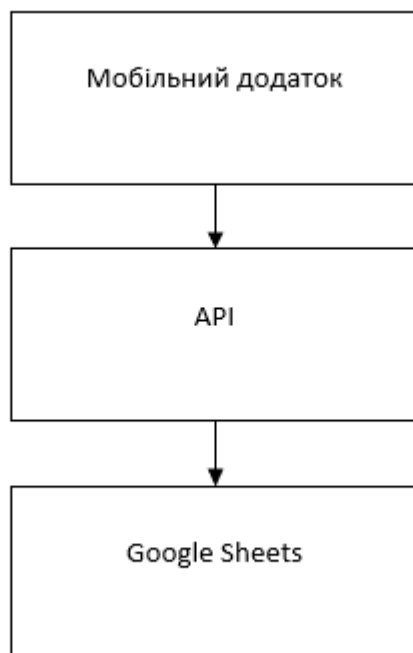


Рисунок 2.5 – Діаграма компонентів

API Google Таблиць заснований на протоколі HTTP і дозволяє взаємодіяти з таблицями Google Таблиць і їх даними. Програма може надсилати HTTP-запити до API для виконання різних операцій, таких як читання, запис або оновлення даних у таблицях.

В результаті вивчення існуючих мобільних платформ для впровадження була обрана система Android. Крім того, проаналізувавши аналоги, були визначені основні вимоги до функціоналу мобільного додатку. Також був ідентифікований і декомпонований бізнес-процес «Інвентаризація». Крім того, була побудована структурна схема додатку і схема компонентів, які допоможуть в подальшому розвитку мобільного додатку.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Тепер перейдемо до вибору мобільної платформи для майбутнього додатку.

На сьогоднішній день у світі існує величезний вибір мов програмування для розробки мобільного програмного забезпечення. Це пов'язано з тим, що для різних мобільних пристроїв необхідно використовувати різні SDK (SDK). Як правило, це залежить від операційної системи пристрою. Найпопулярнішими зараз є IOS, Android.

Ці платформи користуються попитом у всьому світі. Вони зручні та надають користувачеві низку функцій для роботи.

У плані надійності платформа iOS замкнена. Мобільний пристрій поставляється з усіма необхідними заводськими налаштуваннями і позбавляє користувача від необхідності налаштовувати свій смартфон. Однак, з іншого боку, це є недоліком, оскільки немає можливості розширити пам'ять смартфона. Платформа Android відкрита. Мобільний пристрій вимагає ретельного налаштування перед використанням, що, з одного боку, займає час, а з іншого боку, дозволяє користувачеві врахувати всі свої вимоги до смартфона і налаштувати його так, як потрібно користувачеві. З точки зору безпеки, iOS має вбудований захист від вірусів, тому для цієї дії їй не потрібні офлайн-програми. Але з Android користувачеві доводиться стикатися з величезною кількістю вірусів, тому користувачеві не обійтися без установки додаткових додатків, що відповідають за безпеку [4].

Для реалізації задуманого мобільного додатку необхідно визначитися з середовищем розробки.

Android Studio – це офіційне інтегроване середовище розробки для створення додатків Android, розроблене компанією Google. Він заснований на популярній IntelliJ IDEA Java IDE і надає повний набір інструментів і функцій, які допомагають розробникам створювати високоякісні, продуктивні та багатофункціональні програми для Android [5-7].

Android Studio має зручний інтерфейс із функціями редагування макетів і перетягування, що допомагає розробникам швидко створювати та тестувати свої програми. Він також підтримує різні мови програмування, включаючи Java та

Kotlin. Це дозволяє розробникам інтегруватися з іншими інструментами, такими як GitHub для контролю версій [8-9].

Однією з головних переваг Android Studio є те, що вона має велику та активну спільноту розробників, яка надає велику документацію, навчальні посібники та підтримку розробників. Це полегшує розробникам вивчення та використання платформи, а також отримання допомоги за потреби.

Загалом, Android Studio – це потужне та гнучке інтегроване середовище розробки (IDE), яке ідеально підходить для розробки додатків для Android будь-якого розміру та складності, від простих прототипів до великомасштабних корпоративних додатків [9-10].

Xamarin – це кросплатформне середовище розробки додатків, яке дозволяє розробникам використовувати єдину кодову базу для створення власних мобільних додатків для платформ Android, iOS та Windows. Він був придбаний Microsoft у 2016 році і з тих пір інтегрований у середовище розробки Visual Studio.

Однією з ключових переваг Xamarin є те, що він дозволяє розробникам писати програми на C#, популярній мові програмування, яка широко використовується в корпоративній розробці. Це дозволяє розробникам використовувати наявні знання бібліотек C# та .NET для створення мобільних додатків, що може призвести до скорочення часу розробки та зниження витрат.

Xamarin має додаткову перевагу у вигляді доступу до різноманітних API та функцій для конкретної платформи. Це дозволяє розробникам створювати мобільні додатки, які повністю відповідають стандартам і виглядають і відчуються для кожної платформи. Xamarin також надає надійний набір інструментів для налагодження, тестування та розгортання програм.

Одним з потенційних недоліків Xamarin є те, що його може бути складніше налаштувати та використовувати, ніж інші кросплатформні середовища, а деякі проекти можуть вимагати додаткового налаштування. Крім того, запуск коду C# на мобільних пристроях може призвести до низької продуктивності, що в деяких випадках може вплинути на продуктивність вашої програми.

Загалом, Xamarin – це потужна та гнучка платформа, яка добре підходить для розробки кросплатформних мобільних додатків [11].

Appcelerator Titanium – це кросплатформне середовище розробки додатків, яке дозволяє розробникам створювати нативні мобільні додатки для платформ Android та iOS за допомогою технологій веб-розробки, таких як HTML, CSS та JavaScript. Він надає широкий спектр інструментів і функцій, які допомагають розробникам швидко та ефективно створювати високоякісні, багатофункціональні мобільні додатки.

Однією з ключових переваг Appcelerator Titanium є те, що він дозволяє розробникам створювати кросплатформні мобільні додатки з єдиною кодовою базою, що може заощадити час і знизити витрати на розробку. Titanium також надає доступ до великої бібліотеки готових модулів і плагінів, які розробники можуть використовувати для додавання функціональності до своїх додатків.

Ще однією перевагою Titanium є те, що він надає візуальний інтерфейс для створення інтерфейсів користувача, дозволяючи розробникам швидко та легко створювати власні елементи інтерфейсу користувача. Він також надає широкий спектр інструментів налагодження та тестування, які допомагають розробникам забезпечувати якість і надійність своїх програм.

Одним з потенційних недоліків Appcelerator Titanium є те, що він може бути складнішим у використанні, ніж інші кросплатформні фреймворки, особливо для розробників, які не знайомі з технологіями веб-розробки, такими як HTML і CSS. Крім того, деякі розробники можуть виявити, що продуктивність додатків Titanium не така висока або чуйна, як у повністю нативних додатків.

Загалом, Appcelerator Titanium – це потужна та гнучка платформа, яка добре підходить для розробки кросплатформних мобільних додатків для різних сценаріїв використання, особливо для розробників, які знайомі з технологіями веб-розробки та хочуть використовувати ці навички для створення мобільних додатків.

Unity – популярний кросплатформний ігровий движок, який також можна використовувати для розробки мобільних додатків. Він забезпечує потужне та гнучке середовище розробки, яке дозволяє розробникам створювати високоякісні мобільні додатки з покращеною 3D-графікою та фізикою.

Однією з ключових переваг використання Unity для розробки мобільних додатків є те, що він підтримує широкий спектр платформ, включаючи iOS, Android і Windows. Це дозволяє розробникам створювати програми, які можуть працювати на кількох пристроях з єдиною кодовою базою, що може заощадити час і зменшити витрати на розробку.

Ще однією перевагою Unity є велика та активна спільнота розробників, яка надає велику документацію, навчальні посібники та підтримку. Це полегшує розробникам вивчення та використання платформи, а також отримання допомоги, коли це необхідно.

Крім того, Unity надає низку функцій та інструментів, спеціально розроблених для мобільних програм, таких як підтримка сенсорного керування, введення акселерометра, а також оптимізація графіки та продуктивності для мобільних пристроїв. Це дозволяє створювати мобільні додатки, оптимізовані під унікальні функції та вимоги мобільних пристроїв.

Одним із потенційних недоліків використання Unity для розробки мобільних додатків є те, що його може бути складніше налаштувати та використовувати, ніж інші платформи, особливо для розробників, які не знайомі з концепціями розробки ігор. Крім того, програми Unity можуть мати більший розмір файлу, ніж інші мобільні програми, що може вплинути на час завантаження та вимоги до сховища.

Загалом, Unity є потужною та гнучкою платформою для розробки мобільних додатків, особливо для розробників, які хочуть створювати захоплюючі 3D-додатки або вже знайомі з концепціями розробки ігор.

Підводячи підсумки, варто відзначити, що Android Studio є універсальним середовищем розробки, так як дозволяє оптимізувати роботу майбутніх додатків

для роботи не тільки на смартфонах, але і на планшетах, портативних персональних комп'ютерах, які працюють на базі даної операційної системи.

Kotlin – це сучасна мова програмування, розроблена компанією JetBrains. Основною метою створення Kotlin було поєднання простоти та лаконічності сучасних мов програмування з потужними інструментами для розробки великих і складних застосунків.

Kotlin повністю сумісний з Java, що дозволяє використовувати всі існуючі Java-бібліотеки та фреймворки. Це особливо важливо для розробників, які працюють з великою кількістю вже написаного на Java коду. Мова забезпечує безпеку типів, усуваючи багато поширених помилок, пов'язаних із використанням null-значень, що дозволяє уникати NullPointerException.

Код на Kotlin зазвичай більш компактний у порівнянні з аналогічним кодом на Java. Це досягається за рахунок таких можливостей, як виведення типів, лямбда-вирази, а також використання ключового слова `data` для автоматичного генерування допоміжних методів (наприклад, `equals()`, `hashCode()`, `toString()`). Kotlin підтримує функціональний стиль програмування, включаючи функції вищого порядку, лямбда-вирази та незмінні колекції [12-13].

Розробники можуть створювати розширення для існуючих класів без необхідності наслідування або використання шаблонів.

Таким чином, Android Studio була обрана для написання мобільного додатку з кількох причин:

- невеликий досвід написання мобільних додатків на мовах Java/Kotlin;
- гнучкість середовища розробки;
- велика кількість навчальних ресурсів;
- служба підтримки Google.

РОЗДІЛ 3

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

3.1 Практична реалізація об'єкта проектування

Додаток складається з декількох сторінок, написаних на Kotlin за допомогою Google Apps Script. Приклад структури сторінки застосунку «Інвентаризація» наведена на рисунку 3.1. Основними файлами є ReadActivity і WriteActivity.

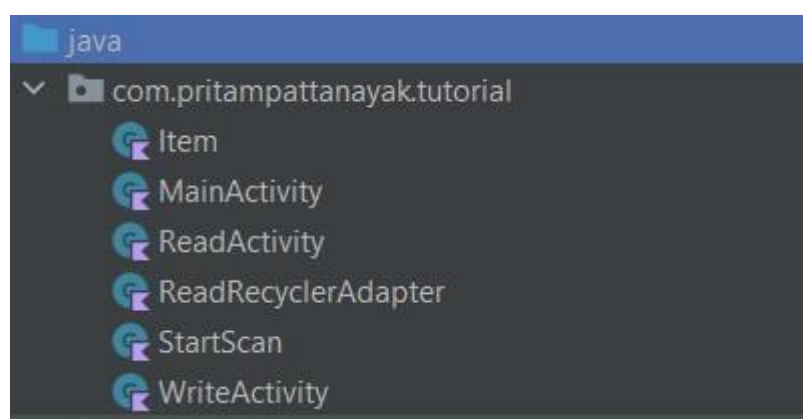


Рисунок 3.1 – Головна сторінка

Розглянемо докладніше кожен модуль.

Модуль збереження об'єктів. Модуль призначений для збереження даних про об'єкт інвентаризації за допомогою скриптів Google Apps.

Google Apps Script Object Saver – це функція, яка дозволяє користувачам зберігати та отримувати структуровані дані у своїх Google Таблицях за допомогою редактора сценаріїв у Google Таблицях або будь-якому іншому продукті Google Apps. Цей модуль забезпечує зручний спосіб збереження та завантаження об'єктів або складних структур даних без необхідності ручного аналізу та форматування.

Модуль використовує вбудовані методи, надані сценарієм Google App, такі як PropertiesService і SpreadsheetApp, для зберігання та пошуку даних. Він дотримується парного підходу ключ-значення, коли об'єкт або структура даних

серіалізується у форматі JSON, а потім зберігається як значення, пов'язане з певним ключем.

Щоб використовувати модуль «Зберегти об'єкти», необхідно включити сценарії в проект Apps Script. Опишемо основні скрипти, необхідні для нашого проекту, в розглянутому модулі.

Функція `doGet` in Google Apps Script – це обробник подій, який автоматично виконується, коли GET-запит надсилається на URL-адресу веб-додатку. Ця функція дозволяє обробляти вхідний GET-запит і генерувати відповідь для відправки назад запитувачу.

Функція `doGet` приймає один параметр `request`, який є HTTP-запитом, зробленим до веб-додатку. Цей параметр є об'єктом, який містить різні властивості та методи доступу до інформації про запит, такі як заголовки запитів, параметри запиту та IP-адреса запитувача.

У цьому сценарії можна виконувати такі завдання, як вилучення даних із зовнішніх джерел, маніпулювання даними, створення динамічного вмісту або взаємодія з іншими службами Google. Також є можливість налаштувати логіку відповідно до вашого конкретного випадку використання та вимог.

Структура лістингу функції `doGet` для нашого проекту виглядає наступним чином (лістинг 3.1).

Лістинг 3.1 – Функція `doGet`

```
function doGet(e){ post_value(e);  
}
```

Кінець лістингу 3.1

Отже, функція, що надається `post_value`, не має будь-якого конкретного функціоналу або реалізації. Його призначення та поведінка залежатимуть від

того, як ви збираєтеся його використовувати, та конкретної логіки, яку ви реалізуєте у функції.

Як правило, іменована функція `post_value`, яка приймає параметр запиту, зазвичай використовується для обробки запиту POST у веб-додатку або кінцевій точці API. Як правило, дані, надіслані в тілі запиту, обробляються, і на основі цих даних виконуються відповідні операції.

При цьому функція виконує такі дії:

- відкриває електронну таблицю Google Sheets на основі наданої URL-адреси;
- отримує аркуш із назвою «штрих-код» з електронної таблиці;
- отримує поточну дату і час і розбиває їх на окремі змінні;
- отримує значення штрих-коду з параметрів запиту;
- додає новий рядок на аркуш «штрих-коду» з поточною датою, часом, штрих-кодом та іншими параметрами;
- готує об'єкт відповіді з рядком JSON;
- повертає відповідь за допомогою `ContentService.createTextOutput`.

Таким чином, цей код отримує POST-запит зі штрих-кодом і параметрами країни. Потім він додає ці значення, а також поточну дату та час у вказану електронну таблицю Google Таблиць. Нарешті, він відповідає об'єктом JSON, що вказує на успіх операції. Код Function `post_value` для нашого проекту наведений в лістингу 3.2.

Лістинг 3.2 – Функція `doGet`

```
function post_value(request){ //Edit this link to your google sheets const
ss=SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1QGqd6s
Qg
skzkjav54m2Wouulsad9ATMDKJ8/edit?USP=Shering");
const sheet = ss.getSheetByName("штрих-код");
const d = new Date(); const currentTimeArray = d.toLocaleString().split(", ");
```

```

const date = currentTimeArray[0] const time = currentTimeArray[1]
const штрих-код = request.parameter.barcode;
sheet.appendRow([дата,час,штрих-код,країна]); result = JSON.stringify({
"result": "Добре"
});
повернути ContentService
createTextOutput("(" + результат + ")")
setMimeType(ContentService.MimeType.JAVASCRIPT);
}

```

Кінець лістингу 3.2

Зчитувач об'єктів. Модуль призначений для зчитування даних про об'єкт інвентаризації за допомогою скриптів Google Apps Script.

Програма Object Reader, що використовує скрипт Google Apps з Google Таблиць, дозволяє отримувати дані з цієї оболонки або зовнішньої комірки в Google Таблиці і вставляти ці дані в додаток. Для цього потрібно використовувати методи, доступні в об'єкті Range класу Sheet в Google Apps Script.

Функція doGet. HTTP GET запит – це один з методів протоколу HTTP, який використовується для отримання даних з сервера. У контексті Google Apps Script і Google Sheets, функція doGet() є обробником GET-запитів і дозволяє взаємодіяти з даними в таблиці Google Sheets при отриманні GET-запиту.

При цьому функція виконує такі дії:

- код оголошує змінну date та описує її дату та час;
- код створює змінну id, яка є реалізуючим елементом ідентифікатора. Це наслідок включення значення item та останнього рядка до робочого аркуша за допомогою методу getLastRow();
- далі код отримує значення параметра Name від об'єкта e.parameter і зберігає його в налаштуваннях itemName;

- таким же чином, код отримує значення ID, а Prices зберігає їх у `itemid` та `itemprices` відповідно;

- код використовує метод `appendRow()` об'єкта аркуша для додавання нового рядка до таблиці;

- в кінці створюється і повертається значення об'єкта `ContentService.createTextOutput()`.

Таким чином, функція `doPost()` бере POST-запит з набору параметрів запиту і зберігає їх в таблиці Google Sheets.

Функція `doPost` виконує такі дії:

- код оголошує змінну запис для зберігання записів (даних) як об'єкта;
- потім код отримує діапазон значень з аркуша. Він використовує `getRange()` для визначення діапазону комірок і `getValues()` для отримання значень цього діапазону. Отримані значення зберігаються в змінній `rows`;

- далі код перебирає кожен рядок змінної `rows` і генерує об'єкт запису для кожного рядка. Значення з певних стовпців зберігаються у властивостях об'єкта запису з відповідними іменами, такими як `itemName`, `itemid`, `itemprimech`;

- сформовані об'єкти запису додаються до масиву даних;
- потім код створює властивість `bookList` в об'єкті запису та присвоює йому значення масиву даних, який містить записи;

- результуюча змінна містить рядок JSON, отриманий з об'єкта запису за допомогою методу `JSON.stringify()`;

- в кінці створюється і повертається значення об'єкта `ContentService.createTextOutput()` содєржачий JSON-строку `result`.

Таким чином, функція `doGet()` отримує GET-запит, зчитує дані з таблиці Google Sheets, генерує об'єкт JSON, що містить дані, і повертає його у вигляді рядка JSON.

Модуль сканування об'єктів – це частина програмного забезпечення, яка дозволяє сканувати та отримувати інформацію про об'єкти за допомогою спеціалізованих пристроїв або технологій. У контексті Google Apps Script

механізм сканування об'єктів можна використовувати для сканування та зчитування інформації за допомогою штрих-кодів, QR-кодів або інших типів кодованих тегів.

Модуль сканування об'єктів на основі Google Apps Script може включати в себе наступний функціонал:

- інтеграція з пристроями сканування. Модуль може взаємодіяти з фізичними сканерами або камерами пристроїв для захоплення зображень або даних з об'єктів;

- закодована обробка етикеток. Модуль може аналізувати отримані дані та розпізнавати закодовані мітки, такі як штрих-коди або QR-коди. Це може включати розшифровку інформації, що міститься в закодованих етикетках;

- інтеграція з базою даних. Модуль може використовувати дані, отримані в результаті сканування об'єкта, і пов'язувати їх з відповідними записами в базі даних або таблиці Google Sheets. Наприклад, можна зберігати інформацію про скановані об'єкти, їх характеристики або місцезнаходження;

- обробка та аналіз даних. Модуль може надавати функціональність для обробки та аналізу отриманих даних. Наприклад, можна шукати об'єкти за певними критеріями або зводити інформацію про відскановані об'єкти;

- інтеграція з іншими програмами. Механізм сканування об'єктів може інтегруватися з іншими програмами або службами, такими як Google Таблиці, щоб автоматично оновлювати дані або виконувати певні дії на основі результатів сканування.

Загальний функціонал модуля сканування об'єктів може сильно відрізнятися в залежності від конкретних вимог і можливостей розробленого рішення на базі Google Apps Script.

Таким чином, ми розробили мобільний додаток для полегшення інвентаризації, який дозволяє користувачеві сканувати QR-коди для обліку та відстеження запасів. Додаток надає можливість використовувати камеру мобільного пристрою для сканування QR-кодів, прив'язаних до кожного товару інвентарю. Відзначимо, що додаток працює не тільки з QR-кодами, а й з будь-

якою текстовою інформацією, яку можна ввести в поле «Назва об'єкту». Після сканування QR-коду програма зберігає інформацію про товар, як-от його ідентифікатор, назву або інші додаткові дані, пов'язані з цим предметом.

Отож, розроблений додаток, який зручний для використання на підприємствах, або в будь-якій ситуації, коли потрібна інвентаризація та відстеження товарів. Додаток дозволяє автоматизувати та спростити процес інвентаризації, скоротити час та виключити можливість помилок у ручному обліку.

3.2 Тестування мобільних додатків

Тестування мобільного додатку для інвентаризації за допомогою сканування QR-коду є важливою частиною розробки. Тестування дозволяє переконатися в функціональності, надійності і відповідності вимогам додатку. Ось кілька кроків, які ви можете виконати під час тестування нашого додатку.

Функціональне тестування. Потрібно переконатися, що основні функції вашого додатка працюють належним чином. Слід перевірити такі аспекти:

- сканування QR-коду з використання камери мобільного пристрою;
- зчитування та розпізнавання інформації з QR-коду;
- збереження інформації про запаси, отриманої зі сканованого QR-коду;
- відображення збереженої інформації про товарно-матеріальні цінності.

Тестування різних типів QR-кодів. Перевіримо, як програма обробляє різні типи QR-кодів. Потрібно переконатися, що вона може обробляти різні формати QR-кодів і правильно розпізнавати інформацію з них.

Тестування введення даних. Потрібно перевірити, як програма обробляє різні варіанти введення. Потрібно перевірити введення неприпустимих символів, довгих рядків, спеціальних символів та інших можливих ситуацій, щоб переконатися, що ваша програма обробляє їх правильно і не викликає збоїв.

Тестування стабільності. Треба перевірити стабільність і продуктивність програми. Потрібно запустити програму в різних мережевих умовах і перевірити

її реакцію на низьку швидкість Інтернету або відсутність мережі. Також перевірити, як програма працює з великим обсягом даних і багаторазовим скануванням.

Тестування на сумісність. Треба переконатися, що програма правильно працює на різних мобільних пристроях та операційних системах.

Протестуємо її на різних версіях Android і на різних роздільних здатностях екрана, щоб переконатися, що користувачі з різними пристроями можуть використовувати його без проблем.

Тестування інтерфейсу користувача. Оцінимо інтерфейс програми, переконаємося, що він інтуїтивно зрозумілий і простий у використанні. Потрібно перевірити навігацію, кнопки, візуальні ефекти та чуйність інтерфейсу під час взаємодії з додатком.

У таблиці 3.1 наведено результати функціональних тестів мобільного додатку «Інвентаризація».

Таблиця 3.1 – Результати функціональних тестів

Тест	Опис	Результат
Тест сканування QR-коду	Додаток правильно сканує QR-коди за допомогою камери мобільного пристрою. Потрібно запустити програму, навести камеру на QR-код і переконатися, що вміст QR-код успішно розпізнається та відобразиться на екрані.	Тест пройшов без помилок
Тест на збереження інформації про запаси	Перевірити, чи правильно збережена інформація про товарно-матеріальний об'єкт після сканування QR-коду. Відсканувати QR-код, що містить інформацію про товар, і переконатися, що відповідні дані, такі як назва, опис, ідентифікатор, збережені в базі даних.	Тест пройшов без помилок

Продовження таблиці 3.1

Тест	Опис	Результат
Тест на редагування інформації про об'єкт інвентаризації	Перевірити можливість редагування інформації про об'єкт інвентаризації. Запустити програму, відсканувати QR-код, а потім змінити деякі деталі, назву чи опис, і зберегти зміни. Переконайтеся, що змінена інформація збережена правильно та відображається під час повторного сканування того самого QR-коду.	Тест пройшов без помилок
Недійсний тест на обробку QR-коду	Переконайтеся, що програма обробляє неправильні або пошкоджені QR-коди. Спробувати відсканувати QR-код з помилкою або спотвореним зображенням і переконайтеся, що програма показує відповідну помилку або повідомлення про невдале сканування.	Тест пройшов без помилок
Тест на інтеграцію з базою даних	Додаток використовує базу даних для зберігання інформації про об'єкти інвентаризації, тестування інтеграції з базою даних. Переконайтеся, що дані правильно збережені, отримані та оновлені в базі даних після сканування QR-коду.	Тест пройшов без помилок

У таблиці 3.2 наведено результати тестів, проведених для зчитування різних типів QR-кодів.

Таблиця 3.2 – Результати тестів для зчитування різних типів QR-кодів

Тест	Результат
Стандартний тест тексту	Тест пройшов без помилок
Тест URL	Тест пройшов без помилок
Перевірка серійного номера	Тест пройшов без помилок
Тест інформації про об'єкт інвентаризації	Тест пройшов без помилок

У таблиці 3.3 наведені результати проведених тестів введення даних.

Таблиця 3.3 – Результати тестування введення даних

Тест	Опис	Результат
Тест на ввід коректних даних	Переконатися, що програма правильно обробляє правильні вхідні дані. Ввести правильні дані, такі як назва товару, опис, ідентифікатор та іншу необхідну інформацію. Переконатися, що додаток зберігає та відображає введені дані без помилок.	Тест пройшов без помилок
Тест введення обов'язкових даних	Переконатися, що додаток перевіряє та обробляє необхідні поля даних. Спробувати ввести лише частину необхідних даних і переконатися, що додаток попереджає про порожні поля і не дозволяє зберегти товар без усіх необхідних даних.	Тест пройшов без помилок
Тест введення спеціальних символів	Перевірити, як додаток обробляє спеціальні символи, як-от лапки, косі риски або розділові знаки. Переконатися, що додаток екранує або обробляє спеціальні символи правильно, щоб уникнути проблем зі збереженням і відображенням даних	Тест пройшов без помилок
Тест введення повторюваних даних	Перевірити, як додаток обробляє введення повторюваних даних, таких як ідентифікатори предметів інвентарю. Спробувати зберегти кілька об'єктів з однаковими ідентифікаторами та переконатися, що програма попереджає про конфлікт і не дозволяє зберегти повторювані дані.	Тест пройшов без помилок

Продовження таблиці 3.3

Тест	Опис	Результат
Тест введення некоректних даних	Переконатися, що програма обробляє неправильне введення даних. Вводити дані з помилками, наприклад неправильним форматом ідентифікатора або неприпустимими символами в полях даних. Переконатися, що програма відображає відповідні повідомлення про помилки та не дозволяє зберігати неправильні дані.	Тест пройшов без помилок

У таблиці 3.4 наведено результати тестів, проведених для перевірки стабільності роботи мобільного додатку.

Таблиця 3.4 – Результати тесту стабільності мобільних додатків

Тест	Опис	Результат
Випробування безперервної роботи	Запустити програму та відсканувати кілька QR-кодів поспіль, не перезапускаючи програму. Переконатися, що програма залишається стабільною та продовжує коректно обробляти сканування QR-коду навіть через тривалий період часу використання.	Тест пройшов без помилок
Тест на перемикання між мережами	Перевіряти стабільність програми під час перемикання між різними мережами, такими як Wi-Fi і мобільна мережа. Перемикатися між мережами під час сканування QR-кодів і переконатися, що додаток підтримує стабільне з'єднання та продовжує працювати надійно.	Тест пройшов без помилок

Продовження таблиці 3.4

Тест	Опис	Результат
Тест на багатопоточність	Перевіряти стабільність роботи програми при сканування кількох QR-кодів одночасно. Запускати кілька екземплярів програми на різних пристроях або емуляторах і сканувати QR-коди одночасно. Переконалися, що всі екземпляри програми стабільні та правильні сканування процесу.	Тест пройшов без помилок

В таблиці 3.5 представлені результати тесту на сумісність.

Таблиця 3.5 – Результати тесту на сумісність мобільних додатків

Тест	Результат
Тестування на різних версіях операційної системи, починаючи з Android 5.0	Тест пройшов без помилок
Тестування на різних роздільних здатностях екрану від 640×1136	Тест пройшов без помилок
Тестування на різних пристроях	Тест пройшов без помилок

У таблиці 3.6 наведено результати тестів валідації інтерфейсу мобільного додатку.

Таблиця 3.6 – Результати тестування валідації інтерфейсу мобільного додатка

Тест	Результат
Тест відображення елементів і верстки	Тест пройшов без помилок
Тест адаптивності та роздільної здатності екрана	Тест пройшов без помилок
Тест локалізації	Тест пройшов без помилок

3.3 Інструкція користувача

Встановлення та запуск. Щоб використовувати програму, вам потрібно встановити та запустити арк-файл на своєму мобільному пристрої. Слід зазначити, що версія операційної системи повинна бути Android 5.0 або вище.

Для того, щоб запустити програму на телефоні, потрібно просто натиснути на встановлену іконку.

Робота з додатком. Стартова сторінка, яка показана на рисунку 3.2, показує об'єкти інвентаризації, які вже були внесені в Google Sheets.



Рисунок 3.2 – Домашня сторінка програми

На рисунку 3.2 використані такі позначення: 1 – кнопка пошуку об'єктів інвентаризації; 2 – кнопка оновлення переліку об'єктів інвентаризації; 3 – кнопка додавання об'єкта інвентаризації.

Для того, щоб додати новий об'єкт інвентаризації, потрібно натиснути на кнопку «+». Далі відкриється форма для внесення нового об'єкта інвентаризації в нашу базу даних. На рисунку 3.4 показана форма нового об'єкта.

Дата

14.05.2024.,18:44:51

Назва об'єкта обліку

Номер (код) об'єкта обліку

Примітки

Відмінити Зберегти

Рисунок 3.4 – Верхнє меню програми

Далі вводимо дані і зберігаємо об'єкт інвентаризації. Потрібно переконатися, що наш об'єкт збережений і в додатку, і в базі даних. На рисунку 3.5 зображений скріншот об'єкта інвентаризації, який щойно додали в додаток.

← 🗑️ ↺ ⋮

Дата
14.05.2024 18:44:51

Назва об'єкта
Монітор HP

Номер (код) об'єкта
388622876

Примітка
110

Інвентаризація

Рисунок 3.5 – Відображення нового об'єкта інвентаризації у додатку

На рисунку 3.6 показаний фрагмент таблиці з доданим об'єктом інвентаризації в базі даних Google Sheets.

Дата	Назва об'єкта	Номер (код) об'єкта обліку	Примітка
09.05.2024 17.22	Відеореєстратор цифровий триплесний	4103400003471	212
09.05.2024 17.23	Джерело безперебійного живлення	21013400007717	101
13.05.2024 14.21	Джерело безперебійного живлення	21013400007719	102
13.05.2024 14.22	Джерело безперебійного живлення	21013400007710	103
13.05.2024 14.23	Джерело безперебійного живлення	21013400007715	104
13.05.2024 14.24	Джерело безперебійного живлення	21013400007716	105
13.05.2024 14.25	Джерело безперебійного живлення	21013400007711	106

Рисунок 3.6 – Відображення нового об'єкта інвентаризації в електронній таблиці Google

Реалізований додаток має наступний функціонал:

- можливість сканування QR-коду;
- можливість додавати дані вручну;
- можливість видалення даних;
- можливість пошуку даних;
- можливість перегляду даних.

ВИСНОВКИ

В ході кваліфікаційної роботи були реалізовані всі поставлені завдання, а саме:

- проведено ґрунтовний аналіз сучасних методів інвентаризації в організаціях, що дозволило визначити основні проблеми та недоліки традиційних підходів;

- вивчено та проаналізовано аналоги заявок на інвентаризацію, виявлено їх недоліки, у зв'язку з чим сформульовано вимоги до розробленого продукту. Проаналізовано наявні мобільні додатки для інвентаризації з використанням QR-кодів. Виявлено, що більшість з них мають обмежену функціональність або високу вартість впровадження;

- створено архітектуру додатку, що включає базові компоненти для забезпечення його стабільної та ефективної роботи. Реалізовано функціонал для сканування QR-кодів, введення даних та їх зберігання у внутрішній базі даних додатку;

- розроблено програму для Android, був обраний оптимальний набір інструментів для цього випадку, в який входили: мова програмування Kotlin, Google App Script, база даних Google Sheets, а також сторонні бібліотеки для Android;

- підготовлено додаток до впровадження в робочі процеси організації, що дозволить значно підвищити ефективність процесу інвентаризації.

Розроблений мобільний додаток для інвентаризації з використанням QR-кодів є ефективним інструментом, що дозволяє значно оптимізувати процеси інвентаризації в організаціях. Його використання забезпечує швидкість, точність та зручність проведення інвентаризації, зменшуючи кількість помилок та витрат часу. Практична значимість додатку підтверджена результатами апробації та позитивними відгуками користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інвентаризація: сутність, класифікація, принципи, функції та завдання. URL: http://www.visnyk-econom.uzhnu.uz.ua/archive/22_1_2018ua/12.pdf (дата звернення: 10.05.2024).
2. Сканер QR і штрих-коду. URL: <https://play.google.com/store/apps/details?id=com.gamma.scan&hl=uk> (дата звернення: 10.05.2024).
3. Інвентаризації + штрих-код. URL: <https://play.google.com/store/apps/details?id=com.maiko.scanpet&hl=uk> (дата звернення: 10.05.2024).
4. Типи мобільних додатків. URL: <https://smile-ukraine.com/ua/mobile-apps/mobile-apps-types> (дата звернення: 10.05.2024).
5. Офіційна документація Android. URL: <https://developer.android.com/docs> (дата звернення: 10.05.2024р.).
6. Android Mobile Application Development. URL: https://docwiki.embarcadero.com/RADStudio/Sydney/en/Android_Mobile_Application_Development (дата звернення: 10.05.2024).
7. Android Application Development. A Brief Overview of Android Platforms and Evolution of Security Systems. A. Sarkar et al. Palladam: IEEE, 2019. 79 p.
8. Shevtsiv N. A., Striuk A. M. Cross platform development vs native development. Kryvyi Rih: Kryvyi Rih National University, 2020. 83 p.
9. Як розробити додаток на Android. URL: <https://foxminded.ua/rozrobyty-dodatok-na-android/> (дата звернення: 20.05.2024).
10. Android Studio: переваги та особливості. URL: <https://qagroup.com.ua/publications/android-studio-perevagy-ta-osoblyvosti/> (дата звернення: 20.05.2024).
11. Microsoft Xamarin. URL: <https://visualstudio.microsoft.com/xamarin/> (дата звернення: 20.05.2024).
12. Розробка на Kotlin. URL: <https://wezom.com.ua/ua/kotlin> (дата звернення: 20.05.2024).

13. Mastering Kotlin: A Beginner's Guide. (Mastering Computer Science)
Paperback, 22 Nov. 2022. 334 p.