

**Національний університет кораблебудування
імені адмірала Макарова**

Навчально-науковий інститут комп'ютерних наук та управління проектами

(повне найменування інституту, назва факультету)

Кафедра інформаційних управляючих систем та технологій
(повна назва кафедри)

Пояснювальна записка

до дипломної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему «Розробка технологій доповненої реальності для автоматичної системи
«візитна картка»»

Виконав: студент 6 курсу, групи 6144
напряму підготовки (спеціальності)

126 Інформаційні системи та технології

(шифр і назва напрямку підготовки, спеціальності)

Король В. К.

(прізвище та ініціали)

Керівник Михелєв І. Л.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

м. Миколаїв – 2021 року

АНОТАЦІЯ

Дипломна робота "Розробка підсистеми автоматичної обробки візитних карток з використанням елементів доповненої реальності" присвячена розробці мобільного додатку розширення інформативності візитних карток.

Додаток написаний з використанням мови програмування Java, мови розмітки XML, та СУБД SQLite.

Дипломна робота містить інформацію про створення мобільного додатку, постановку задачі, проект програмного забезпечення, результати розробки.

Дипломна робота виконана на 41 сторінках, містить 19 рисунки, 6 таблиць, 1 додаток та список використаної літератури з 12 найменувань.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

АННОТАЦИЯ

Дипломная работа "Разработка подсистемы автоматической обработки визитных карточек с использованием элементов дополненной реальности" посвящена разработке мобильного приложения расширения информативности визитных карточек.

Приложение написано с использованием языка программирования Java, языка разметки XML, и СУБД SQLite.

Дипломная работа содержит информацию о создании мобильного приложения, постановку задачи, проект программного обеспечения, разработки.

Дипломная работа выполнена на 41 страницах, содержит 19 рисунка, 6 таблиц, 1 приложение и список использованной литературы из 12 наименований.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

ABSTRACT

Thesis "Development of a subsystem for automatic processing of business cards using elements of augmented reality" is devoted to the development of a mobile application to expand the information content of business cards.

The application is written using the Java programming language, XML markup language, and SQLite.

The thesis contains information on the creation of a mobile application, task setting, software project, development.

The thesis was completed on 41 pages, contains 19 figures, 6 tables, 1 appendix and a list of used literature from 12 items.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ, РОЗРОБКА ЇЇ КОНЦЕПЦІЇ І ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

1.2 Аналіз існуючих рішень

1.2.1 AR business card connect

1.2.2 QR scanner

1.2.3 AR Business Card

1.3 Аналіз існуючих програмних продуктів

1.4 Розробка концепції на основі аналізу продуктів

1.5 Постановка задачі

РОЗДІЛ 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Опис мов програмування, програмних середовищ та систем управління

2.1.1 Java

2.1.2 XML

2.1.4 SQLite

2.1.5 Android Studio

РОЗДІЛ 3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Проектування бази даних

3.1.1 Опис основних сутностей

3.1.2 Концептуальне проектування і ER-модель бази даних

3.1.3 Логічна модель бази даних

3.1.4 Фізична модель бази даних

3.2 Структура мобільного додатку

3.3 Розробка дизайну графічного інтерфейсу

3.4 Розробка діаграми функціональності додатку

3.5 Структура мобільного додатку

					ДР.122.4141.10.ПЗ.Р4	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

РОЗДІЛ 4 АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Реалізований функціонал

4.2 Опис розробленого мобільного додатку

4.2.1 Головна сторінка

4.2.2 Сторінка оголошень

4.2.3 Сторінка входу

4.2.4 Сторінка реєстрації

4.2.5 Сторінка додавання нового об'єкта сканування

РОЗДІЛ 5 ОХОРОНА ПРАЦІ

5.1 Вступ

5.2 Хімічні речовини у повітрі робочої зони

5.3 Шум

5.4 Мікроклімат у приміщенні

5.5 Робоча поза

5.6 Загальні вимоги безпеки праці з комп'ютерною технікою

Висновки

Список використаних джерел

ДОДАТКИ

Додаток А. Дизайн проекту

Додаток Б. Повне технічне задання

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

ВСТУП

Для людини завжди була дуже важливою можливість обміну послугами або товарами між собою.

Способів знайти працю або продати(обміняти) товар з кожним роком стає все більше. Але один завжди залишиться та буде актуальним - це домова однієї людини з іншої. Раніше потребувалось, читати газети або слухати радіо чи йти до магазинів, щоб знайти потрібний товар. Однак з розвитком цифрових технологій такі методи над застарілі. Зараз найпоширенішим методом пошуки та надання роботи або продаж товарів є оголошення в інтернет-ресурсах. Завдяки розвитку технологій одна людина може запропонувати товар(роботу) іншій, яка знаходиться у іншому кутку країни або світу.

Метою даної дипломної роботи є розробка веб-дошці, що зможуть надати людям зручну площадку для розміщення своїх оголошень.

Об'єктом розробки є веб-дошка.

Предметом розробки є розробка додатку веб-дошки.

Для досягнення поставленої мети треба розв'язати наступні задачі:

- 1) Проаналізувати дані, описані в предметної області. Визначити таблиці, необхідні для подання предметної області в рамках реляційної моделі даних;
- 2) Виконати аналіз існуючих рішень;
- 3) Спроекувати базу даних веб-дошки;
- 4) Розробити структуру і дизайн інтерфейсу клієнтської частини веб-додатку;
- 5) Провести програмну реалізацію серверної частини веб-додатку;
- 6) Провести тестування функціональності веб-додатку.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ, РОЗРОБКА ЇЇ КОНЦЕПЦІЇ І ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Можливості сучасних комп'ютерних технологій можливо використовувати для підвищення інформативності візиток, рекламних буклетів.

На даний момент великого поширення має концепція розподілених автоматизованих систем, спрямованих на обробку інформації. Це дозволяє розробити програму яка підвищить інформативність візитних карток. Для реалізації даної ідеї необхідно створити для кожної предметної області автоматизовані робочі місця на базі персональних комп'ютерів.

Скануючий пристрій (далі по тексту СП) – сукупність апаратних і програмних засобів, які знаходяться безпосередньо у користувача і призначений для зчитування даних з візитної картки та демонстрації необхідної інформації. СП мають проблеми неоднакових інтерфейсів взаємодії програмного додатку та операційної системи.

Мобільний додаток забезпечує:

- простоту, зручність і дружнє ставлення до користувача;
- легка адаптація до конкретних функцій користувача;
- підвищення інформативності.

Ефективним режимом роботи СП є його функціонування в рамках всесвітньої обчислювальної мережі. Створені СП дають можливість користувачеві працювати в діалоговому режимі, оперативно вирішувати поточні завдання, зручно вводити дані, вести контроль, обробку інформації, виводити і передавати каналами зв'язку.

Скануючий пристрій з додатком повинен бути в кожного користувача який хоче використовувати дану функціональність. Проте, загальні принципи залишаються незмінними:

- системність;
- ефективність.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Під системністю варто розуміти наступне: робоча станція повинна являти собою систему взаємопов'язаних компонентів. Ця структура повинна чітко виконувати функції, для виконання яких створюється ця робоча станція.

Принцип гнучкості має першорядне значення в створенні сучасного та ефективного функціонування робочих місць. Цей принцип означає, що СП може бути адаптований до пропонованого оновленню програмного забезпечення. В даний час, швидкість старіння обладнання та програмного забезпечення продовжує рости, дотримання цього правила стає одним з найбільш важливих умов для створення оновлень нашого мобільного додатку.

Принцип ефективності означає, що вартість створення системи і експлуатації не повинні перевищувати економічні вигоди від її реалізації.

Крім того, при створенні скануючого пристрою повинно враховуватися, що його ефективність багато в чому буде залежати від адекватного поділу функцій і навантаження на інтернет і засобами обробки інформації, ядром якої є смартфон. Тільки в таких умовах СП стає не тільки засобом підвищення продуктивності і ефективності управління, а також соціальної комфортності користувачів.

В даній дипломній роботі буде розроблено програмний комплекс по розпізнаванню та підвищення інформативності візитних карток. Саме такого підвищення інформативності візитних карток. Головні проблеми будуть розглянуті нижче.

1.2 Аналіз існуючих рішень

Існує чотири варіанти підвищення інформативності візитних карток, банерів, буклетів, тощо. Варіант перший – використання усієї площі поверхні для розміщення тексту, фотографій, логотипів, та іншої інформації. Варіант другий – використати QR код для переходу на веб сторінку з усією необхідною інформацією. Варіант третій – придбати можливість використати сторонні додатки з функцією доповненої реальності. Варіант четвертий – розробити свою систему для розпізнавання об'єктів, зберігання даних, мобільний додаток для користувачів.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Системи розпізнавання об'єктів існують двох видів: з використанням штучного інтелекту и використання математичних моделей.

Переваги використання математичної моделі у додатку в тому, що можливо отримати не тільки інформацію про сам відсканований об'єкт, а ще і положення об'єкту у просторі. Також для систем з використанням штучного інтелекту потрібно використати більше ніж одне зображення, для збільшення шансу розпізнавання, ніж для математичної моделі. Що до переваг використання мобільного додатку з доповненою реальністю від використання QR коду в тому що процес взаємодії людини зі змішаною реальністю біль запам'ятовуються. У випадку використання лише візитних карток, проблеми з'являються такі як: неможливість актуалізації даних, відсутність інтерактивності, неможливість помістити усю необхідну інформацію.

1.2.1 AR business card connect

Мобільний додаток AR business card connect – використовує доповнену реальність для відображення доданої інформації відносно положення карки у просторі. У системі присутні: централізована база, сервера обробки інформації, та клієнти що використовують мобільний додаток.

Переваги:

- Можливість додавати об'єкти сканування без використання оновлення додатку
- Зручний та зрозумілий функціонал
- Можливість редагувати усю додану інформацію

Недоліки:

- Цей додаток працює лише на операційній системі IOS
- Великий розмір додатку (460 Мб)

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

1.2.2 QR scanner

Мобільний додаток QR scanner не використовує доповнену реальність. Основна мета цього додатку в тому що він розшифровує інформацію закодовану у QR код, також може запропонувати перейти за веб адресом що був закодований.

Переваги:

- Можливість додавати об'єкти сканування без використання оновлення додатку
- Зручний та зрозумілий функціонал
- Не можливість відредагувати інформацію що була закодована та знаходиться у користувача
- Додаток для управління є для різних операційних систем смартфона
- Може отримати інформацію з пошкодженого коду (до 25% пошкодження)
- Не потребує бази зі збереженою інформацією

Недоліки:

- Не можливо відредагувати закодовану інформацію що знаходиться у клієнта
- Можливо закодувати до 2953 байт

1.2.3 AR Business Card

Мобільний додаток AR business card – використовує доповнену реальність для відображення доданої інформації відносно положення карти у просторі. У системі присутні: централізована база, сервера обробки інформації, та клієнти що використовують мобільний додаток.

Переваги:

- Можливість додавати об'єкти сканування без використання оновлення додатку

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- Зручний та зрозумілий функціонал
- Можливість редагувати усю додану інформацію

Недоліки:

- Цей додаток працює лише на операційній системі Android

1.3 Аналіз існуючих програмних продуктів

Star Walk 2 Free: Карта зоряного неба і Астрономія - ваш інтерактивний гід по зірках і для планет, зоряний атлас космосу, додаток для пошуку і відстеження зірок, сузір'їв, планет, супутників та інших небесних тіл на карті зоряного неба. Досить навести пристрій до неба і на екрані з'явиться зоряне небо - точно що відображається карта зоряного неба з усіма об'єктами в їх правильному положенні.

Інтерактивний планетарій - астрономія і космос у вашому телефоні!

Карта зоряного неба оновлюється автоматично, слідуючи за вашим рухом. Направляючи пристрій на різні ділянки неба, на екрані ви побачите точне розташування небесних тіл (зірки і сузір'я на небі, планети і супутники, комети, метеори і т.д.) в реальному часі. (Ця функція працює тільки на пристроях з цифровим компасом.)

SketchAR: почни малювати з AR - Додаток розміщує віртуальне зображення за допомогою доповненої реальності, щоб обвести його на реальній поверхні.

Додаток корисно тим, хто давно хотів почати малювати, але не міг зробити перший крок. Також SketchAR допоможе професійним художникам як простий спосіб перенесення ескізу на поверхню.

Для коректної роботи дотримуйтесь рекомендацій:

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- Для стабільної роботи необхідно досить світле простір. У темній кімнаті камері буде складно визначити поверхню.

- Не забувайте використовувати тримач для паперу, щоб вона не рухалася під час малювання.

- Намагайтеся не трясти смартфон.

Дотримуйтесь цих правил і все буде працювати справно;

Wikitude дозволяє бачити більше, ніж можуть захопити ваші очі. Наша технологія «комп'ютерного зору» трохи схожа на магію і сполучає вас із світом абсолютно по-новому. Використовуйте додаток як інструмент для випробування змісту розширеної реальності, кампаній, проектів, акцій, ігор та багато іншого, просто скануючи те, що ви бачите!

В основі програми Wikitude - це Пошук електроенергії, ваш шлюз для розблокування світу цифрового розширеного вмісту навколо вас. Наприклад, компанії можуть надати вам код пошуку, який безпосередньо призведе до конкретного досвіду AR. Просто введіть його в поле пошуку вгорі і дозвольте Wikitude доповнити ваш світ.

Holo - Holograms for Videos in Augmented Reality - дозволяє додавати голограми реальних людей і тварин в ваш світ і зробити фотографії & відео, щоб поділитися з друзями. Веселіться холод-змішування! :)

- Створення забавний і унікальний контент, додавши символ Холо в камеру і клацаючи фото або відео.

- Виберіть з різних голограм, включаючи герой фільму, знаменитість, спортсмен, музикант і комедіант, а також забавні оригінальні персонаж і тварина. Нові Голографические персонажі випустили щотижня!

- Відрегулюйте голограму зміна розміру, переміщення, або обертаючи його, навіть під час запису Холо.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- Збереження відео Холо і фотографії в галерею телефону.

1.4 Розробка концепції на основі аналізу продуктів

СП - програмно-технічний комплекс, призначений для автоматизації діяльності певного виду. При розробці СП для управління технологічним обладнанням. СП об'єднує програмно-апаратні засоби, що забезпечують взаємодію людини з комп'ютером, надає можливість введення інформації (через клавіатуру, тачскрін, камеру та ін.) І її висновок на екран смартфона, динаміки смартфона або інші пристрої виведення. Як правило, СП є частиною ПК (програмного комплексу).

Програма має зручний призначений для користувача інтерфейс, основне меню, і працює на платформі Android SDK. Для роботи програми необхідна наявність функціонуючого смартфона з операційною системою Android (5.0 – 10). Надійність програми залежить від роботи всієї системи і від умов її зберігання. Якщо програма не встановлена на смартфоні, то її можливо скачати з “Play market”.

Основні цілі, для досягнення яких створений мобільний додаток:

- Забезпечення користувачам більш комфортного отримання інформації про конкретне підприємство;
- Забезпечення підприємств-клієнтів додатковою рекламою, новими клієнтами;
- Ретельне відстеження змін даних;
- Демонстрація можливостей команді-розробників.

Вибір фреймворків для будування доповненої реальності складається з таких як:

- AR Core (google)
- Wikitude

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- Maxst
- SDK DeepAR Augmented Reality EasyAR
- ARKit
- Vuforia

Якщо пройтись по кожному к цих фреймворків:

AR Core (google) - це набір для розробки програмного забезпечення, розроблений Google, який дозволяє будувати додатки розширеної реальності ARCore використовує три ключові технології для інтеграції віртуального контенту з реальним світом, що спостерігається через камеру вашого телефону:

- Шість ступенів свободи дозволяють телефону зрозуміти та відслідковувати своє становище відносно світу.
- Екологічне розуміння дозволяє телефону визначати розміри та розташування плоских горизонтальних поверхонь, таких як земля або журнальний столик.
- Оцінка світла дозволяє телефону оцінити поточні умови освітлення навколишнього середовища.

Wikitude – Wikitude - постачальник мобільних технологій доповненої реальності (AR), що базується в Зальцбурзі, Австрія. Заснований у 2008 році, Wikitude спочатку зосередився на наданні на основі локації досвіду розширеної реальності за допомогою програми веб-переглядів Wikitude World. У 2012 році компанія реструктуризувала свою пропозицію, запустивши SDK Wikitude, розроблену структуру, що використовує функції розпізнавання та відстеження зображень та геолокаційні технології. Базується на AR Core та ARKit.

Maxst – MAXST AR SDK - це двигун AR для розвитку послуг розширеної реальності. Він включає шість основних функцій SDK для розпізнавання та відстеження 2D-зображень та 3D-просторів. Вона також включає функцію розпізнавання зображень на основі хмарного сервера та AR Fusion Tracker з потужним відстеженням ARKit / ARCore.

SDK DeepAR Augmented Reality – створений для розробників, які хочуть додати найвищу якість, оптимізовану для мобільних пристроїв, стилі Snapchat та

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Facebook, 3D-реалістичні обличчя, маски та спеціальні FX для додатків iOS, Android, HTML5 та Unity. Найкраще - DeepAR SDK може бути інтегрований за лічені хвилини та підтримує сотні масок, лінз, ефектів та фільтрів. Наша команда також готова допомогти вам створити дивовижний новий контент для вашої платформи.

ARKit - розробка для платформ Apple надає передовій технології macOS, iOS, watchOS та tvOS під рукою, надаючи безмежні способи залучення неймовірних додатків до користувачів усього світу. Ці потужні платформи пропонують унікальні можливості та досвід користувачів, але тісно інтегруються та утворюють справжню екосистему. Апаратне, програмне забезпечення та сервіси розроблені з самого початку, щоб працювати разом, щоб ви могли будувати інтуїтивно зрозумілий багатогранний досвід, який справді є безпроблемним.

Vuforia Engine - це найпоширеніша платформа для розробки AR, яка підтримує провідні телефони, планшети та окуляри. Розробники можуть легко додати розширені функціональні можливості комп'ютерного зору для додатків Android, iOS та UWP, щоб створити досвід AR, який реально взаємодіє з об'єктами та оточенням. Бібліотека двигунів Vuforia містить технічну документацію, яка допомагає розробникам з нашим SDK та створенням програм AR. Ліворуч ви побачите, що бібліотека двигунів Vuforia організована усіма платформами, які підтримує Vuforia Engine. Vuforia Engine підтримує як Unity Engine, так і три основні платформи: iOS, Android та UWP.

Бібліотека двигунів Vuforia також організована за основними категоріями функцій. Основними категоріями зображень є зображення, об'єкти та оточення.

Має посилання на документацію API Vuforia Engine, яка включає C # API для Unity, API C ++ для iOS та UWP та Java для Android.

Java - об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код,

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Висновок: із двох представлених концепцій я обрав другу, та із представлених фреймворків я обрав Vuforia SDK тому що:

- Є необхідність кожного разу, коли користувач встановлює мобільний додаток мати можливість сканувати одразу, не викачуючи базу даних візиток;
- Має віддалене розпізнавання;
- Швидко розвивається;
- Має підтримку як Android так і IOS;
- Проводити складні математичні обчислення на мобільному процесорі ARM буде дуже довго;
- Необхідна підтримка мобільних платформ починаючи з Android 5.0.

1.5 Постановка задачі

Основним завданням впровадження даної підсистеми є підвищення інформативності та полегшення роботи відділів продажу компаній-клієнтів. Таким чином буде збільшена інформаційна розповсюдженість інформації про компанію-клієнта, отже фінансові втрати будуть збільшені.

Таким чином ефективність роботи реклами зросте, а навантаження на деякі відділи зменшиться.

Проаналізувавши побажання клієнтів і виявивши всі мінуси, постараємося створити таку систему, яка б мала такий функціонал:

- Розпізнавання візитної картки;
- Накладання біля картки відео яке клієнт вважатиме доречним;
- Демонстрація 3Д моделі поверх візитної картки;
- Відображення деякої кількості кнопок під візитною карткою.
- При нахилі картки відносно камери, усі вище описані пункти мають переходити у перспективу.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

В результаті розробка програмного комплексу зводиться до підвищення популярності компаній-клієнтів за допомогою підвищення інформативності візитних карток за допомогою технології доповненої реальності, та математичної моделі розпізнавання конкретної візитної картки серед великої кількості інших карток.

Головне призначення програмного комплексу в даному випадку - підвищити ефективність виконання основних функцій реклами та розповсюдженості інформації серед оточення.

Критеріями вибору математичних моделей розпізнавання є:

- Точність відповіді;
- Функціональна повнота системи, швидкодія

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

2.1 Опис мов програмування, програмних середовищ та систем управління

2.1.1 Java

Java — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

«Oracle» надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

Java вплинула на розвиток J++, що розроблялась компанією «Microsoft». Роботу над J++ було зупинено через судовий позов «Sun Microsystems», оскільки ця мова програмування була модифікацією Java. Пізніше в новій платформі «Microsoft» .NET випустили J#, щоб полегшити міграцію програмістів J++ або Java на нову платформу. З часом нова мова програмування C# стала основною мовою

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

платформи, перейнявши багато чого з Java. J# востаннє включався в версію Microsoft Visual Studio 2005. Мова сценаріїв JavaScript має схожу із Java назву і синтаксис, але не пов'язана із Java.

2.1.2 XML

Розширювана мова розмітки (англ. Extensible Markup Language, скорочено XML) — запропонований консорціумом World Wide Web Consortium (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет.[1] Є спрощеною підмножиною мови розмітки SGML. XML-документ складається із текстових знаків, і придатний до читання людиною.

Стандарт XML (англ. Recommendation, перше видання від 10 лютого 1998, останнє, четверте видання 29 вересня 2006) визначає набір базових лексичних та синтаксичних правил для побудови мови описання інформації шляхом застосування простих тегів. Цей формат достатньо гнучкий для того, аби бути придатним для застосування в різних галузях[2]. Іншими словами, запропонований стандарт визначає метамову, на основі якої шляхом запровадження обмежень на структуру та зміст документів визначаються специфічні, предметно-орієнтовані мови розмітки даних. Ці обмеження описуються мовами схем (англ. Schema), такими як XML Schema (XSD), DTD або RELAX NG. Прикладами мов, заснованих на XML, є: XSLT, XAML, XUL, RSS, MathML, GraphML, XHTML, SVG, а також XML Schema.

2.1.3 SQLite

SQLite — полегшена реляційна система керування базами даних. Втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92. Сирцевий код SQLite поширюється як суспільне надбання (англ. public domain), тобто може

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

використовуватися без обмежень та безоплатно з будь-якою метою. Фінансову підтримку розробників SQLite здійснює спеціально створений консорціум, до якого входять такі компанії, як Adobe, Oracle, Mozilla, Nokia, Bentley і Bloomberg.

З 2018 року SQLite, як й JSON та CSV, рекомендований Бібліотекою Конгресу США формат зберігання структурованого набору даних.

У 2005 році проект отримав нагороду Google-O'Reilly Open Source Awards.

2.1.4 Android Studio

Android Studio — интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O.

Данная IDE находилась в свободном доступе начиная с версии 0.1, опубликованной в мае 2013, а затем перешла в стадию бета-тестирования, начиная с версии 0.8, которая была выпущена в июне 2014 года. Первая стабильная версия 1.0 была выпущена в декабре 2014 года, тогда же прекратилась поддержка плагина Android Development Tools (ADT) для Eclipse.

Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, — официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, macOS и Linux. 17 мая 2017, на ежегодной конференции Google I/O, Google анонсировал поддержку языка Kotlin, используемого в Android Studio, как официального языка программирования для платформы Android в дополнение к Java и C++.

3.1 Проектування бази даних

3.1.1 Опис основних сутностей

Після аналізу предметної області основними сутностями бази даних мобільного додатку було виділено:

- Візитка картка
- Користувачі

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- Компанії
- Новини

Опис даних сутностей:

1. Візитна картка

Дана сутність зберігає інформацію про візитку картку. Цю інформацію вносить адміністратор при додаванні нової візитної картки.

Атрибути:

- Картинка (шлях до картинки);
- Назва візитної картки;
- Опис іншої картки;

2. Користувачі

Дана сутність зберігає інформацію про користувачів, інформацію про яких дають самі користувачі.

Атрибути:

- Ім'я;
- Логін;
- Пароль;
- Email;
- Id компанії до якої він належить;

3. Компанія

Дана сутність зберігає інформацію про компанію.

Атрибути:

- Назва компанії;
- Логотип компанії(шлях до картинки);

3.1.2 Концептуальне проектування і ER-модель бази даних

Модель «сутність-зв'язок» (ER-модель) (англ. Entity-relationship model або entity-relationship diagram) — модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. ER-модель — це мета-

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

модель даних, тобто засіб опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення, що його реалізує, є модель «сутність-зв'язок». Важливим є той факт, що з моделі «сутність-зв'язок» можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна, об'єктна), тому вона є найзагальнішою.

Модель сутність-зв'язок є результатом систематичного процесу, який описує та визначає деяку предметну область. Вона не визначає сам процес, а лише візуалізує його. Дані представлені у вигляді компонентів (сутностей), які пов'язані між собою певними зв'язками, які виражають залежності і вимоги між ними, такі як: одна будівля може бути розділена на нуль або більше квартир, але одна квартира може бути розташована лише в одній будівлі. Сутності можуть мати різні властивості (атрибути), які характеризують їх. Діаграми, створені для представлення цих сутностей, атрибутів і зв'язків графічно, називають сутність-зв'язок діаграмами.

ER-модель зазвичай реалізується в вигляді баз даних. У разі реляційної бази даних, в якій зберігаються дані в таблицях, кожен рядок кожної таблиці являє собою один екземпляр сутності. Деякі поля даних в цих таблицях вказують на індекси в інших таблицях. Такі поля є показниками фізичної реалізації зв'язків між сутностями.

Клас приналежності є обов'язковим, якщо всі об'єкти одного класу беруть участь в зв'язку з об'єктами іншого класу або необов'язковим, якщо об'єкти одного класу не беруть участі в зв'язку з об'єктами іншого класу.

3.1.3 Логічна модель бази даних

Основою на технічному завданні була створена логічна модель бази даних, що представлена на рис. 2.2.2:

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

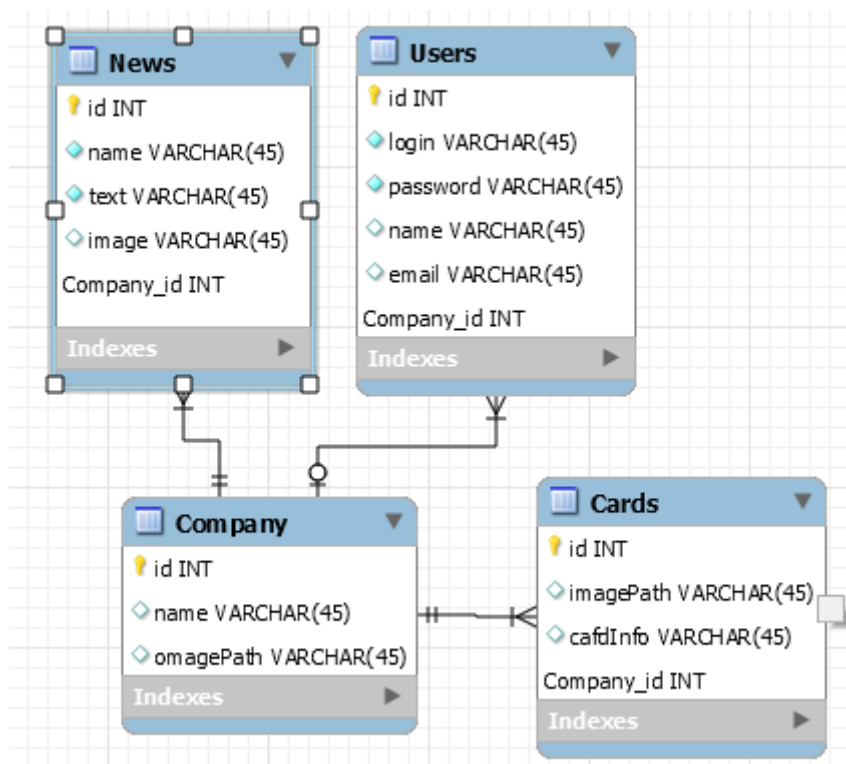


Рисунок 2.2.2 – Логічна модель бази даних

3.1.4 Фізична модель бази даних

Фізична модель (СУБД орієнтована модель) містить всю інформацію, необхідну для реалізації конкретної БД.

На основі логічної моделі бази даних була створена фізична модель бази даних. Дана модель містить безпосередні назви таблиць та атрибутів, які будуть використовуватися в подальшому, типи атрибутів, їх розмір та первинні і зовнішні ключі.

Фізична модель БД для SQLite відображена в таблицях 2.1-2.14

Таблиця 2.1

Таблиця «users»

Поле	Тип	NN	Ключ
id	INT	Так	ПК
name	VARCHAR(255)	Так	

email	VARCHAR(255)	Так	
password	VARCHAR(255)		
login	VARCHAR(255)		
compant_id	INT		

Таблиця 2.2

Таблиця «News»

Поле	Тип	NN	Ключ
id	INT	Так	РК
name	VARCHAR(255)	Hi	
text	TEXT	Hi	
imagePath	TIMESTAMP	Hi	
company_id	VARCHAR(255)	Hi	

Таблиця 2.3

Таблиця «card»

Поле	Тип	NN	Ключ
id	INT	Так	РК
name	VARCHAR(255)		
image_path	VARCHAR(255)	Так	FK

Таблиця 2.4

3.2 Структура мобільного додатку

Основна задача мобільного додатку – презентація візитних карток. Беручи до уваги те, що додаток має бути зручним, містити мінімальну кількість інформації щоб користувач не міг заплутатися, і мати зручну навігацію, було вирішено використовувати павутину структуру.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Структура додатку представлена на рис. 2.3:

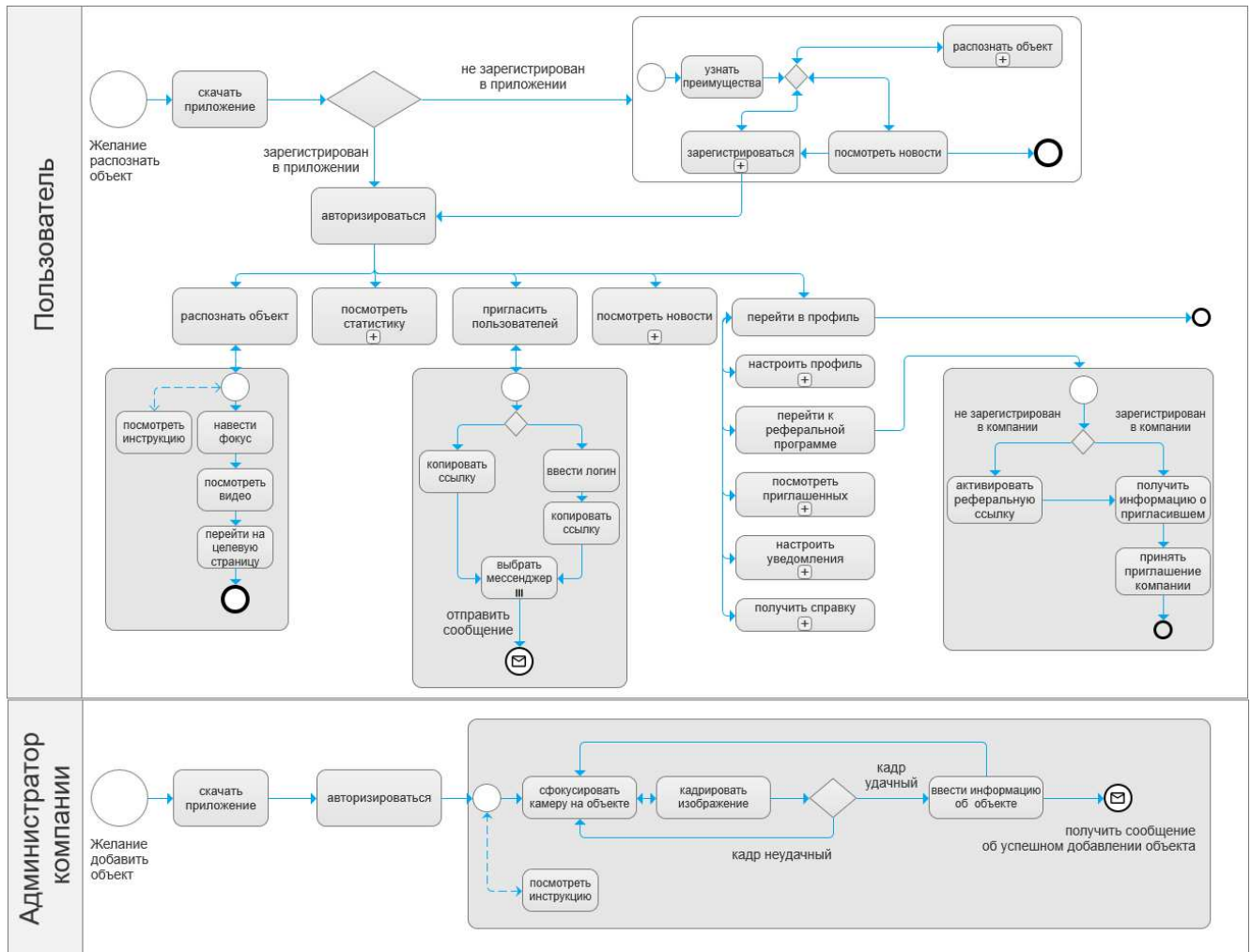


Рисунок 2.3 – Структура мобільного додатку

3.3 Розробка дизайну графічного інтерфейсу

В процесі створення дизайну були обрані світлі кольори для додатку.

На всіх сторінках буде відображена панель з кнопками та бокове меню.

Бокове меню буде містити в собі наступні пункти:

- Логотип компанії;
- «Розпізнати»- блок:
 - картинка;
 - текст;
- «Мої запрошені»- блок:
 - картинка;
 - текст;
- «Запросити»- блок:
 - картинка;

- текст;
- «Новини»- блок:
 - картинка;
 - текст;
- «Профиль»- блок:
 - картинка;
 - текст;
- «Істория»- блок:
 - картинка
 - текст;
- «Вийти» - блок:
 - картинка;
 - текст.

Дизайн буде прикладений в додатку А.

3.4 Розробка діаграми функціональності додатку

UML (Unified Modeling Language) - уніфікована мова моделювання - це мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, який використовує графічні позначення для створення абстрактної моделі системи, що має назву «UML модель». UML був створений для визначення, візуалізації, проектування та документування програмних систем.

Використання UML не обмежується моделюванням програмного забезпечення. Його також використовують для моделювання бізнес-процесів, системного проектування та відображення організаційних структур.

На даний момент додаток веб-дошки не потребує адміністрації, так як основний контент завантажують самі користувачі. На рис. 2.5 відображена діаграма варіантів використання додатку.

Функціонал додатку розділений на дві частини:

1) Гість - може користуватися навігацією по додатку, пошуком, переглядом оголошень та коментарів.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

2) Користувач – після реєстрації та авторизації відвідувач сайту отримує свою сторінку та має можливість створювати, редагувати та видаляти оголошення, та коментарі і змінювати інформацію свого профілю.



Рисунок 2.5 – Діаграма варіантів використання веб-додатку

3.5 Структура мобільного додатку

Проект даного додатку складається з клієнтської та серверної частини. Клієнтська частина містить наступні компоненти:

- **build.gradle** - відповідає за компоновку майбутнього мобільного додатку
- ***.class** – файли з таким розширенням містять реалізацію класів
- **manifest** – відповідає за взаємодію майбутнього додатку з системою
- **res** – папка яка містить в собі усі елементи візуальної частини мобільного додатку
- **java** – папка яка містить в собі весь програмний код майбутнього додатку
- ***.xml** – файл який містить у собі елементи візуального інтерфейсу

РОЗДІЛ 4 АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Реалізований функціонал

В процесі написання даної роботи була розроблена бета версія мобільного додатку, в якому біли реалізовані наступні функції:

- Розпізнавання об'єктів з камери.
- Демонстрація відео біля візитної картки.
- Демонстрація кнопок під візитною карткою.
- Додавання об'єктів для розпізнавання.

Нереалізовані функції знаходяться в стані розробки.

4.2 Опис розробленого мобільного додатку

4.2.1 Головна сторінка :

На рис. 3.2.1 – 3.2.2 зображений макет головної сторінки мобільного додатку. На ній відображаються 4 візитки чи будь які інші предмети для розпізнавання. Навігація по додатку здійснюється за допомогою бокового меню та невеликого меню знизу екрана.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

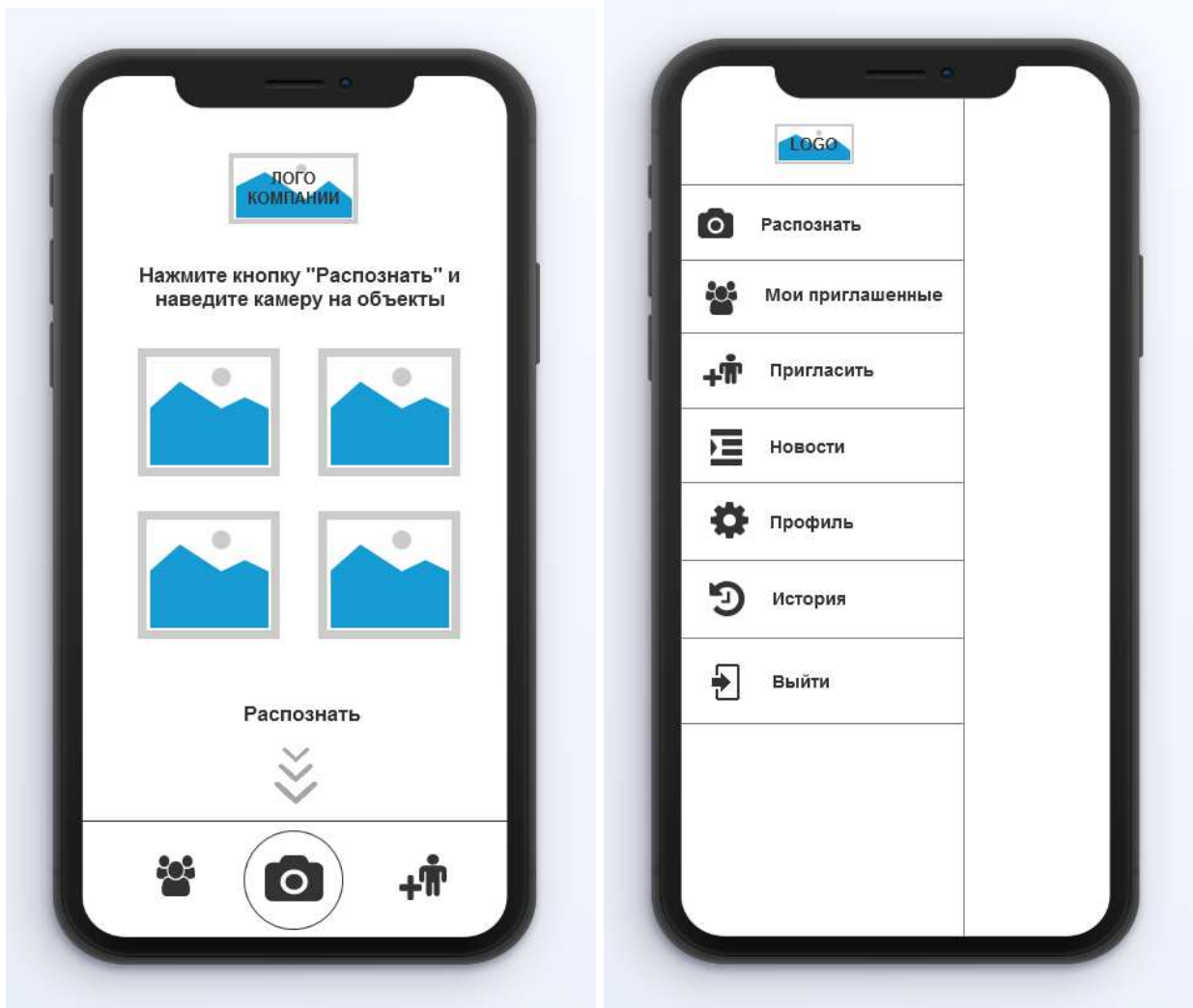


Рисунок 3.2.1, 3.2.2 – Головна сторінка

4.2.2 Сторінка оголошень:

Сторінка розпізнавання демонструє зображення з камери, та часки графічно накладених елементів доповненої реальності.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

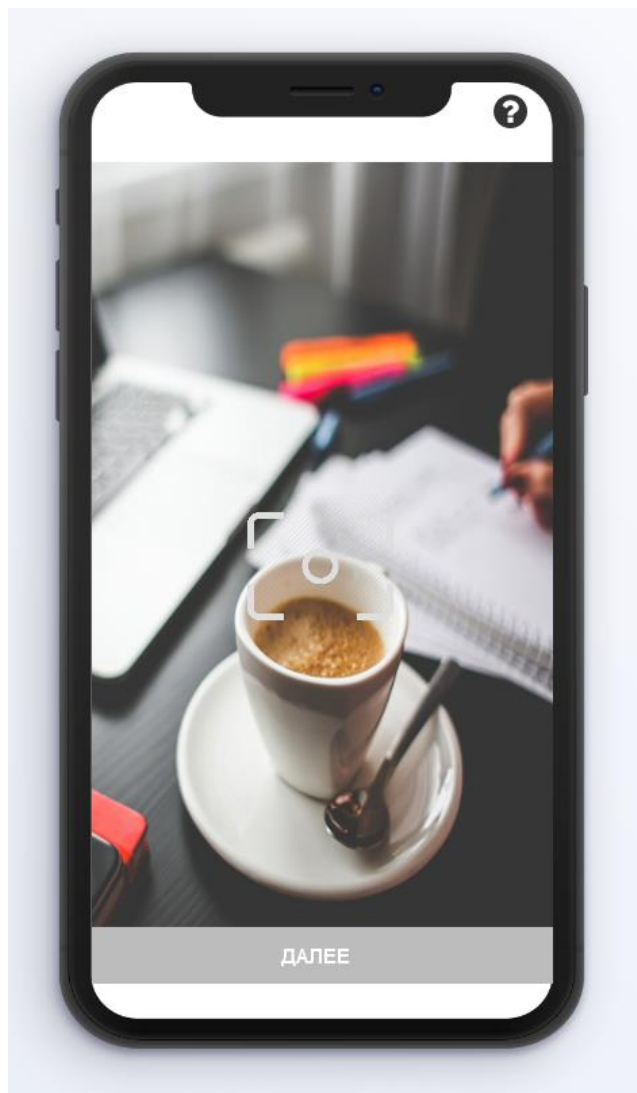


Рисунок 3.2.2 – Розпізнавання

4.2.3 Сторінка входу:

Для входу в аккаунт користувачеві необхідно ввести свої логін і пароль. Якщо користувач не зареєстрований на сайті, йому необхідно перейти за на сторінку реєстрація.

4.2.4 Сторінка реєстрації:

Сторінка реєстрації містить в собі поля для введення імені, телефону, email, інформації про запрошуючого користувача, та назва компанії до якої він належить.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

4.2.5 Сторінка додавання нового об'єкта сканування:

Сторінка додавання нового об'єкта має у собі поля для додавання інформації про об'єкт, такі як:

- Назва об'єкта
- Посилання на відео
- Інформацію про об'єкт
- Налаштування кнопок для переходу на різні ресурси

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

РОЗДІЛ 5 ОХОРОНА ПРАЦІ

5.1 Вступ

Зараз важко уявити собі підприємство, установу чи офіс, де б не використовувався у роботі персональний комп'ютер. З того часу, як комп'ютери увійшли в життя людей, проводяться дослідження, як саме вони впливають на здоров'я. Дізнайтеся, які фактори виробничого середовища і трудового процесу впливають на умови праці з персональним комп'ютером та яких заходів безпеки слід дотримуватися.

Діяльність більшості працівників сучасних професій у виробничій сфері пов'язана з використанням комп'ютерної техніки. Комп'ютер для сучасної людини є такою ж технічною необхідністю, як телевізор або холодильник. Побутові прилади ми використовуємо не задумуючись про їх шкідливість або нешкідливість, усвідомлюючи лише переваги їх наявності. Щодо комп'ютерів, то існує багато інформації як про їх безпечність, так і шкідливість.

Наказом Держгірпромнагляду від 26.03.2010 р. № 65 затверджено Правила охорони праці під час експлуатації електронно-обчислювальних машин (далі— Правила № 65). Вони поширюються на всіх суб'єктів господарювання незалежно від форм власності, які у своїй діяльності здійснюють роботу, пов'язану з електронно-обчислювальними машинами (ЕОМ) з відеодисплейними терміналами (ВДТ), у т. ч. на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і периферійними пристроями (ПП).

Персональні комп'ютери (ПК) — це теж ЕОМ, тож у статті термін «комп'ютер» і аббревіатура «ЕОМ» є синонімами.

Осіб, які працюють з комп'ютерами, поділяють на групи:

- розробники програм (інженери-програмісти) мають справу переважно з відео терміналами, їх робота характеризується інтенсивною розумовою творчою працею з підвищеним напруженням зору, концентрацією уваги на фоні нервово-

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

емоційного напруження, вимушеною робочою позою, загальною гіподинамією, періодичним навантаженням на кисті рук;

- оператори електронно-обчислювальних машин виконують роботу, пов'язану з обліком інформації, одержаної з візуального дисплейного терміналу за попереднім запитом, або тієї, що самостійно надходить з нього, яка супроводжується перервами різної тривалості, пов'язана з виконанням іншої роботи і характеризується як робота з напруженням зору, невеликими фізичними зусиллями, нервовим напруженням середнього ступеня та виконується у довільному темпі;

- оператори комп'ютерного набору займаються одноманітною за характером роботою з документацією та клавіатурою, під час якої нечасто та ненадовго переключають погляд на екран дисплея, вводять дані з високою швидкістю. Робота характеризується як фізична праця з підвищеним навантаженням на кисті рук на фоні загальної гіподинамії з напруженням зору (фіксація зору переважно на документи), нервово-емоційним напруженням.

Згідно з визначенням, наведеним у Правилах № 65, оператор ЕОМ з ВДТ і ПП — працівник, який використовує екранні пристрої під час своєї роботи.

Фактори виробничого середовища і трудового процесу під час роботи з комп'ютером

Працюючи з комп'ютером, працівник потрапляє під вплив різноманітних факторів виробничого середовища та трудового процесу. Розглянемо ці фактори та їх дію на організм людини.

5.2 Хімічні речовини у повітрі робочої зони

Хімічний фактор робочого простору не є характерним для працівників, які використовують у своїй роботі ПК. Так, внаслідок економії бюджету на меблі безпосередньо у робочій зоні користувачів ПК може бути присутнім феноли, толуол, ацетальдегід (хімічна речовина 1 класу небезпеки, гостроспрямованої дії на

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

організм людини, перш за все виділяється з матеріалу для меблі, ДСП). За результатами проведених гігієнічних досліджень вміст цих компонентів, як правило, не перевищує ГДК.

Джерелом хімічних речовин у повітрі робочої зони можуть бути матеріали, які використані для ремонту та оздоблення приміщень. Серед них синтетичні матеріали для підлоги (ворсоніт, ковролін, лінолеум), полімерні матеріали для оздоблення стін, тощо. Найпоширенішою речовиною, яка може виділятися з полімерних матеріалів, є формальдегід (хімічна речовина 2 класу небезпеки, гостроспрямованої та алергенної дії на організм людини, граничнодопустима концентрація 0,5 мг/м³). Вміст формальдегіду у повітрі робочої зони може бути на рівні 0,5 ГДК.

5.3 Шум

Нормативним значенням еквівалентного рівня звуку для програмістів є 50 дБА, для операторів у залах оброблення інформації — 65 дБА, для операторів у приміщеннях, де розташовані гучні агрегати — 75 дБА.

За результатами проведених гігієнічних досліджень, еквівалентні рівні шуму на робочих місцях працівників офісних приміщень знаходяться у межах 48-59 дБАекв. Як правило, перевищення рівня шуму на робочих місцях офісних працівників є наслідком телефонних розмов співробітників (у приміщеннях з великою кількістю робочих місць).

Неіонізуюче випромінювання. Як правило, роботу працівників з ПК пов'язують зі шкідливим впливом електромагнітних полів. Це мало місце під час використання моніторів з електронно-променевою трубкою. У сучасних комп'ютерах використовуються рідкокристалічні монітори, тому вплив електромагнітного поля від монітора практично відсутній. Наявним залишається електростатичне поле на поверхні монітора та поверхні клавіатури.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Нормативне значення напруженості електростатичного поля становить 150 В/см. Напруженість електростатичного поля, визначена на поверхні монітора та поверхні клавіатури, як правило, не перевищує нормативного значення.

Рівень напруженості електростатичного поля залежить від вологості повітря, регулярного прибирання робочого місця (усунення запиленості).

5.4 Мікроклімат у приміщенні

У теплий період року температура повітря має бути у межах 22-25 °С, швидкість руху повітря — до 0,1 м/с, відносна вологість повітря — 40-60%.

У холодний період року температура повітря може коливатися у межах 21-24 °С, швидкість руху повітря — до 0,1 м/с, вологість повітря — 40-60%.

Для робочої зони працівників з ПК, які працюють в офісних центрах з централізованим повітрозабезпеченням, як влітку, так і взимку характерні значення відносної вологості повітря на рівні 20-40%, знижена або підвищена швидкість руху (рухливість) повітря робочої зони.

Як правило, температура повітря перевищує нормативні значення протягом року через постійне нагрівання деталей ПК, знижену вологість.

Саме температура та вологість повітря впливають на загальне самопочуття, стан слизових оболонок очей, верхніх дихальних шляхів та шкіри персоналу офісів. Низька вологість збільшує час «зависання» пилу в повітрі приміщення.

До характеристик повітря робочої зони користувачів ПК віднесені такі, як вміст позитивних та негативних іонів у повітрі.

Як правило, у приміщеннях з розвиненою системою припливно-витяжної вентиляції та/або з кондиціонерами виявляються порушення норм аероіонного складу повітря, при цьому концентрація корисних для організму негативно заряджених аероіонів може бути в 10-50 разів нижче норми, а концентрація шкідливих позитивних аероіонів значно перевищувати норму. Ось чому необхідно, щоб до робочого місця з комп'ютером надходило свіже повітря. Рекомендується робити вологе прибирання не рідше одного разу на день. Обов'язково слід

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

протирати поверхню столу серветкою з антистатиком, оскільки позитивні іони мають властивість з'єднуватися з частинкам пилу.

У сучасних офісних спорудах з центральною вентиляцією і відсутністю можливості провітрювання приміщень забезпечення відповідної аероіонізації є досить проблематичним.

Слід додати, що серед офісних працівників відмічається високий рівень захворюваності з тимчасовою втратою працездатності, з них захворювання органів дихання складають 45-60%. За даними медичних установ, серед офісних працівників значно поширені захворювання ЛОР-органів (до 25% працівників), при цьому переважають ураження верхніх дихальних шляхів, а саме: хронічні катаральні фарингіти (15,7%). Слід зазначити високу частоту розвитку алергічних ринітів (1,3%), що у декілька разів вища за середню по Україні (0,28%).

5.5 Робоча поза

Цей фактор трудового процесу залежить від правильної організації робочого місця для забезпечення зручної робочої пози, адже «закам'яніле» положення шкідливо впливає на хребтово-м'язову систему.

Стіл має бути просторим, із спеціальною підставкою для ніг, а робочий стілець мати відрегульовану висоту, певний кут нахилу сидіння і спинки.

Під час роботи за комп'ютером людина сидить кілька годин поспіль у незручному положенні. Це не тільки викликає загальну втому, а й може призвести до розвитку остеохондрозу різних ділянок хребта — шийного, грудного, попереково-крижового.

Неправильне положення рук під час друкування на клавіатурі призводить до хронічних захворювань кисті. Тому клавіатура має розташовуватися на відстані 10-15 см від краю столу.

Щоб робота за комп'ютером не шкодила здоров'ю, необхідно постійно стежити за своєю поставою. Правильна постава максимально розвантажує м'язи і дозволяє працювати довше, менше втомлюючись.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Напруженість праці. Напруженість праці працюючого з ПК пов'язана, перш за все, з напруженістю очей, тривалістю зосередження уваги, яка може становити понад 75% часу. Тому саме очі найбільш страждають під час роботи з комп'ютером.

Велике значення при роботі за комп'ютером мають такі речі як: відстань до екрана, шрифт, розмір тексту на моніторі, наявність або відсутність мерехтіння, яскравість екрана, освітлення робочого місця, наявність перерв у роботі. Саме ігнорування таких простих на перший погляд речей у значній мірі призводить до погіршення зору та хвороб очей.

5.6 Загальні вимоги безпеки праці з комп'ютерною технікою

Умови праці користувачів ПК мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 № 7 (далі — ДСанПіН 3.3.2.007-98), та Правил № 65.

ДСанПіН 3.3.2.007-98 поширюються на умови й організацію праці під час робіт з ВДТ усіх типів вітчизняного та зарубіжного виробництва на основі електронно-променевих трубок (ЕПТ), що використовуються в ЕОМ колективного та персонального використання.

Сьогодні ж у більшості офісних приміщень встановлено нові сучасні комп'ютери з рідкокристалічними моніторами, для яких не прописано гігієнічні вимоги у ДСанПіН 3.3.2.007-98. Водночас вимоги ДСанПіН 3.3.2.007-98 до виробничих приміщень, параметрів виробничого середовища приміщень, організації і обладнання робочих місць, режимів праці та відпочинку є актуальними під час робіт з будь-якими комп'ютерами та моніторами.

Відповідно до пунктів 4.1, 4.2 Правил № 65 ЕОМ з ВДТ і ПП під час експлуатації мають відповідати вимогам стандартів, вказаних Правил і пройти державну санітарно-епідеміологічну експертизу згідно з Порядком проведення

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

державної санітарно-епідеміологічної експертизи, затвердженим наказом МОЗ України від 09.10.2000 № 247 (у редакції наказу МОЗ України від 14.03.2006 № 120).

ЕОМ з ВДТ і ПП закордонного виробництва мають відповідати вимогам національних стандартів держав-виробників і мати відповідну позначку на корпусі, у паспорті або в іншій експлуатаційній документації.

Зазначені вище нормативні документи — обов'язкові для виконання Вони визначають критерії безпечного використання комп'ютерної техніки та призначені для запобігання несприятливої дії на працівників шкідливих факторів, що супроводжують роботу з візуальними дисплейними терміналами.

Керівники державних органів, підприємств, установ, організацій незалежно від форми власності й підпорядкування у порядку забезпечення контролю зобов'язані облаштувати робочі місця користувачів ЕОМ та персональних ЕОМ (ПЕОМ) з ВДТ відповідно до вимог Правил № 65.

Керуючись зазначеними вище нормативними документами можна дійти висновку, що для забезпечення комфортної роботи оператора ПК необхідно звернути увагу на такі основні вимоги:

- стан виробничих приміщень;
- характеристики та стан техніки (ПК, монітори, принтери, наявність заземлення тощо);
- організація робочого місця;
- дотримання режимів праці та відпочинку;
- моніторинг стану здоров'я працівника (стан органів зору, показники захворювань з тимчасовою втратою працездатності, ін.);
- профілактичні заходи.

Так, робоче місце оператора має відповідати вимогам ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования».

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Висновки

Результатом виконання дипломної роботи є розробка мобільного додатку автоматичної обробки візитних карток з використанням елементів доповненої реальності.

Був проведений аналіз предметної області, аналіз існуючих рішень та розробка концепції мобільного додатку, спроектована база даних, дизайн та інтерфейс клієнтської частини.

На даний момент проект продовжує розвиватися та розроблятися. Планується підвищення точності розпізнавання та підвищена точність позиціонування візитної картки відносно мобільного телефону.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація SQLite - <https://www.sqlite.org/index.html>
2. Документація Android - Android Programming: The Big Nerd Ranch Guide, 4th Edition / 2019
3. Стаття Вікіпедії XML - <https://uk.wikipedia.org/wiki/XML>
4. Документація AR Core - <https://developers.google.com/ar/discover>
5. Документіція Vuforia - <https://developer.vuforia.com/>
6. Сайт «google play» - <https://play.google.com/store>
7. Стаття Вікіпедії ER - https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%C2%AB%D1%81%D1%83%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C_%E2%80%94%D0%B7%D0%B2%27%D1%8F%D0%B7%D0%BE%D0%BA%C2%BB
8. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин /ДСанПіН 3.3.2.007-98. – К.: Постанова Головного державного санітарного лікаря України, 1998. - № 7.
9. Анісімов В. Н. Комп'ютер: вчені починають розслідування \\\ Інформатика та освіта. 2008 В даній статті йдеться про вплив комп'ютерної техніки на користувача, про шкоду впливу електричних і магнітних полів на організм людини.
10. Баловсяк Н.В. Комп'ютер і здоров'я. Пітер. 2008 У книзі йдеться про профілактику несприятливого впливу комп'ютера на здоров'я користувача.
11. Ергономіка сучасного офісу. Тріо-плюс 2009. У книзі йдеться про правильне розміщення користувача при роботі з комп'ютером.
12. Ананін В. короткозорості при тривалій роботі з комп'ютером можна уникнути \\\ Людина і комп'ютер. 2008 У книзі йдеться про запобігання захворювань очей при роботі з комп'ютером.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

ДОДАТКИ

Додаток А. Технічне завдання

1. Мета

1. презентація товарів та послуг
2. допомога з рекрутингом
3. допомога з навчанням

2. Модуль «Авторизація»

Мета: авторизація необхідна для доступу користувачів до системи і розширеним функціям додатку.

Модуль складається з наступних розділів:

- Авторизація – сторінка.

1. Сторінка «Авторизація»

Сторінка «Авторизація» призначена для введення даних користувача і авторизації згідно з ним у додатку.

Сторінка «Авторизація» повинна мати:

- логотип додатку;
- заголовок;
- «E-mail» - однострокове текстове поле;
- «Виберіть компанію» - випадаючий список;
- «Пароль» - однострокове текстове поле;
- «Показати пароль» - кнопка;
- «Увійти» - кнопка;
- «У вас ще немає аккаунта?» - текст;
- «Зареєструйтесь» - гіперсилка;
- інформація о авторских правах – текст;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- головне меню (див. ТЗ «Меню додатку»):
- бокове меню (див. ТЗ «Меню додатку»), за замовчуванням приховано.

При додаванні Email вміст випадаючого списку повинно бути змінено:

- У випадаючому списку повинно бути відображено компанії, в котрих приймає участь користувач з введеним Email.

Якщо по введеному Email не знайдено користувачів:

- поле повинно бути підсвічене червоним;
- повинно бути відображено спливаюче повідомлення «Введений Email не знайдений в системі. Спробуйте ще раз чи зареєструйтесь.».

Поле «Пароль» введені символи повинні бути скриті за спеціальний символ «*». При натисканні на кнопку «Показати пароль» символи повинні бути відображені.

При натисканні на кнопку «Увійти» повинні бути виконана перевірка, чи відповідає введений пароль Email-адресу користувача у вибраній компанії рекрутинга.

При успішному проходженні перевірки додаток повинен бути запущен у відповідності з обраною компанією та ролі користувача.

Якщо перевірка не пройдена, користувачу повинно бути відображено спливаюче повідомлення з вказаною помилкою: «Пароль не відповідає введенню Email чи Email вказан не вірно. Перевірте введені дані та спробуйте ще раз.».

3. Модуль «Реєстрація»

Мета: модуль необхідно для реєстрації нових користувачів у рекрутингу.

Модуль складається з наступних розділів:

- «Реєстрація» - сторінка;
- «Вкладка з основними даними» - сторінка;
- «Вкладка з даними про компанії» - сторінка.

1. Сторінка «Реєстрація»

Сторінка «Реєстрація» необхідна для реєстрації нових користувачів у рекрутингу.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Сторінка повинна мати:

- логотип додатку;
- «Реєстрація» - заголовок;
- «Вкладка з основними даними» - вкладка:
- «Ваше ім'я» - блок:
- «Номер телефона» - блок:
- «Ваш Email» - блок:
- логотип;
- однорядкове текстове поле;
- «Ваш пароль» - блок:
- «Показати пароль» - кнопка;
- «Далі» - кнопка;
- «© GlobalScan 2019» - текст;
- «Вкладка з даними про компанію» - вкладка:
- «Ваш спонсор» - блок:
- «Ваша компанія» - блок:
- «Я згоден з умовами» - флажок, де частина «умовами» є гіперпосиланням на сторінку з публічними умовами;
- «Реєстрація» - логотип;
- «© GlobalScan 2019» - текст;

При натисканні на кнопку «Далі» винна, чи заповнені всі поля. При успішному проходженні перевірки додаток повинен відобразити вкладку з даними компанії.

На вкладці з даними компанії в поле «Ваш спонсор» повинен бути відображений логін користувача, з чиєї реферального посилання було завантажено додаток. Якщо додаток було завантажено без реферального посилання, поле повинно бути порожнім.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

При введеному значенні в поле «Ваш спонсор» в випадіючому списку повинні бути відображені рекрутингу, членом яких є спонсор. Якщо збігів не знайдено, користувачеві повинна бути відображена помилка «На жаль, зазначений спонсор не знайдений в системі».

При натисканні на кнопку «Реєстрація» повинна бути виконана перевірка:

- чи всі поля заповнені (необхідний результат перевірки - все поля заповнені);
- обраний прапорець «Я згоден з умовами»;
- повинна бути виконана перевірка, чи існує користувач з введенням E-mail в обраному рекрутингу (необхідний результат перевірки - користувач не існує);
- повинна бути виконана перевірка, чи існує користувач з введенням E-mail в обраному рекрутингу (необхідний результат перевірки - користувач не існує);

При успішному проходженні перевірки:

- новий користувач повинен бути зареєстрований в рекрутингу;
- користувач повинен бути автоматично авторизований в додатку за вказаними при реєстрації даних.

Якщо перевірка не пройдена, має бути відображено повідомлення з відповідною помилкою:

- «Всі поля повинні бути заповнені»;
- «Ви не прийняли умови угоди»;
- «Користувач з вказаним E-mail вже існує в обраному рекрутингу»;
- «Спонсор із зазначеним користувач не існує знайдений в обраному рекрутингу».

4. Модуль «Переваги»

Мета: модуль призначений для первинного ознайомлення користувача з функціями програми.

Модуль доступний неавторизованого користувачеві:

- відразу при першому запуску програми;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- при натисканні на кнопку «Допомога».

Модуль складається з наступних розділів:

- «Вкладка 1» - сторінка;
- «Вкладка 2» - сторінка;
- «Вкладка 3» - сторінка;
- «Вкладка 4» - сторінка.

1. Сторінка «Вкладка 1»

Сторінка потрібна для відображення інформації про один з переваг використання програми.

Сторінка повинна містити:

- логотип програми;
- «Виділяйтеся на тлі конкурентів» - заголовок;
- «Використовуйте нові можливості для просування вашої компанії» - текст;
- зображення;
- «Пропустити» - кнопка;
- перемикач вкладок;
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

При натисканні на кнопку «Пропустити» повинна бути відображена головна сторінка додатка для неавторизованого користувача (див. ТЗ «Головна сторінка»).

Блок перемикачів повинен прокручуватися свайпом вліво і вправо.

У головному меню кнопка «Допомога» повинна бути активна.

2. Сторінка «Вкладка 2»

Сторінка потрібна для відображення інформації про один з переваг використання програми.

Сторінка повинна містити:

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- логотип програми;
- «Дивуйте своїх партнерів» - заголовок;
- «Проводьте презентації компанії тут і зараз просто навівши на об'єкт» - текст;
- зображення;
- «Пропустити» - кнопка;
- перемикач вкладок;
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

При натисканні на кнопку «Пропустити» повинна бути відображена головна сторінка додатка для неавторизованого користувача (див. ТЗ «Головна сторінка»).

Блок перемикачів повинен прокручуватися свайпом вліво і вправо.

У головному меню кнопка «Допомога» повинна бути активна.

3. Сторінка «Вкладка 3»

Сторінка потрібна для відображення інформації про один з переваг використання програми.

Сторінка повинна містити:

- логотип програми;
- «Партнерська програма» - заголовок;
- «Переглядайте вашу структуру і з легкістю діліться реферальною посиланням» - текст;
- зображення;
- «Пропустити» - кнопка;
- перемикач вкладок;
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

При натисканні на кнопку «Пропустити» повинна бути відображена головна сторінка додатка для неавторизованого користувача (див. ТЗ «Головна сторінка»).

Блок перемикачів повинен прокручуватися свайпом вліво і вправо.

У головному меню кнопка «Допомога» повинна бути активна.

4. Сторінка «Вкладка 4»

Сторінка потрібна для відображення інформації про один з переваг використання програми.

Сторінка повинна містити:

- логотип програми;
- «Будьте в курсі подій» - заголовок;
- «Отримуйте повідомлення про додавання нового учасника в вашу структуру і стежте за новинами компанії» - текст;
- зображення;
- «Розпізнати» - кнопка;
- перемикач вкладок
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

При натисканні на кнопку «Розпізнати» повинна бути відображена сторінка додатка «Розпізнати» (див. ТЗ «Розпізнавання»).

Блок перемикачів повинен прокручуватися свайпом вліво і вправо.

У головному меню кнопка «Допомога» повинна бути активна.

5. Головна сторінка

Мета: сторінка необхідна для відображення користувачеві основної інформації про програму.

Модуль складається з наступних розділів:

- головна сторінка для неавторизованого користувача;
- головна сторінка для авторизованого користувача;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- головна сторінка для адміністратора.

1. Головна сторінка для неавторизованого користувача

Сторінка призначена для відображення основної інформації про програму неавторизованого користувачеві.

Сторінка повинна містити:

- логотип програми;
- «Натисніть кнопку" Розпізнати "і наведіть камеру на об'єкти» - заголовок;
- «E-mail» - однорядкове текстове поле;
- блок зображень;
- «Розпізнати» - текст;
- зображення - анімація стрілки на кнопку «Розпізнати»;
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

Блок зображень містить превью стандартних об'єктів, які користувач може розпізнати. При розпізнаванні одного із запропонованих стандартних об'єктів користувачеві повинно бути відображено відео з інформацією про програму. Під відео повинна бути розташована кнопка «Увійти». При натисканні на кнопку браузер користувача повинен бути перенаправлений на

2. Головна сторінка для авторизованого користувача

Сторінка призначена для відображення основної інформації про програму авторизованому користувачеві.

Сторінка повинна містити:

- логотип компанії;
- «Натисніть кнопку" Розпізнати "і наведіть камеру на об'єкти» - заголовок;
- «E-mail» - однорядкове текстове поле;
- блок зображень;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- «Розпізнати» - текст;
- зображення - анімація стрілки на кнопку «Розпізнати»;
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

Блок зображень містить превью об'єктів компанії, які користувач може розпізнати.

3. Головна сторінка для адміністратора

Сторінка призначена для відображення основної інформації про програму адміністратору.

Сторінка повинна містити:

- логотип компанії;
- «Пожвавите свій бренд разом з GlobalScan:» - заголовок;
- «Додайте об'єкти, які зможуть розпізнати ваші користувачі» - текст;
- «Додайте відео, яке буде відтворюватися після наведення камери на об'єкт» - текст;
- «Збережіть об'єкт» - текст;
- «Дивуйте своїх партнерів» - текст;
- «Мої об'єкти» - кнопка;
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

При натисканні на кнопку «Мої об'єкти» додаток повинен бути перенаправлено на сторінку «Мої об'єкти» (див. ТЗ «Мої об'єкти»).

6. Модуль «Розпізнавання»

Мета: модуль призначений для розпізнавання об'єктів і відтворення їх вмісту.

Модуль складається з наступних розділів:

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- Сторінка розпізнавання;
- Сторінка відтворення;
- Сторінка з цільовим дією.

1. Сторінка розпізнавання

Сторінка призначена для введення даних користувача і авторизації по ним в додатку.

Сторінка «Авторизація» повинна містити:

- підказка за допомогою:
- «Закрити» - кнопка;
- «Наведіть камеру на об'єкт. Додаток автоматично відтворить вміст. » - текст;
- область для розпізнавання.

2. сторінка відтворення

Сторінка призначена для відтворення вмісту об'єкта.

Сторінка повинна містити:

- область, на якій розміщений об'єкт;
- матеріал для відтворення.

Після відтворення повинна бути відображена сторінка з цільовим дією.

3. Сторінка з цільовим дією

Сторінка призначена для повторного відтворення матеріалу і перенаправлення користувача на цільову сторінку.

Сторінка повинна містити:

- область, на якій розміщений об'єкт;
- блок матеріалу для відтворення;
- матеріал для відтворення;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- «Повторити» - кнопка;
- кнопка з цільовим дією.
- При натисканні на кнопку з цільовим дією:
- повинен бути відкритий браузер за замовчуванням;
- повинна бути завантажена посилання, яка прив'язана до об'єкта (див. ТЗ «Додаток. Адміністратор. Мої об'єкти»).

7. Модуль «Мої запрошені»

Мета: модуль призначений для перегляду списку користувачів структури та інформації про них і для запрошення нових користувачів.

Модуль складається з наступних розділів:

- «Мої запрошені» - сторінка;
- «Запросити» - сторінка.

1. Сторінка «Мої запрошені»

Сторінка призначена для виведення списку користувачів в структурі і основної інформації про них.

Сторінка повинна містити:

- «Мої запрошені» - заголовок;
- блок фільтра:
- іконка;
- «Фільтр» - однорядкове текстове поле;
- «Запросити користувача» - кнопка;
- блок користувача:
- фото;
- «[Ім'я] [Прізвище]» - текст;
- «[E-mail]» - текст;
- «[Телефон]» - текст;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- блок дублюється під кожного користувача;
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

При введенні значення в поле «Фільтр» список користувачів повинен бути відфільтрований відповідно до введеного значенням. Під фільтрацію потрапляють всі поля. Збігаються значення в списку користувачів повинні бути виділені.

При натисканні на блок користувача повинно бути відображено системне вікно з пропозицією зателефонувати або відправити E-mail.

При натисканні на кнопку «Запросити користувача» додаток повинен відобразити сторінку «Запросити».

2. Сторінка «Запросити»

Сторінка призначена для запрошення нового користувача в структуру.

Сторінка повинна містити:

- «Назад» - кнопка;
- «Запросити користувача» - заголовок;
- «Надішліть реферальну посилання зручним для вас способом» - текст;
- блок реферального посилання:
- «Копіювати посилання» - кнопка;
- «[Реферальная посилання]» - текст;
- блок «Поділитися»:
- «Поділитися в Telegram» - кнопка;
- «Поділитися в Facebook Messenger» - кнопка;
- «Поділитися в VK» - кнопка;
- «Поділитися в Instagram» - кнопка;
- «Ще» - кнопка.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

При натисканні на кнопку «Назад» додаток повинен завантажити сторінку «Мої запрошені».

При натисканні на кнопку «Копіювати посилання» реферальна посилання повинна бути скопійована в буфер обміну системи.

При натисканні на кнопки «Поділитися в Telegram», «Поділитися в Facebook Messenger», «Поділитися в VK», «Поділитися в Instagram» повинен бути запущений функціонал опрацювання повідомлення у вибраний месенджер.

При натисканні на кнопку «Ще» має бути відображена системне модальне вікно з вибором месенджера.

8. Модуль «Новини»

Мета: модуль необхідний для відображення останніх новин компанії.

Модуль складається з наступних розділів:

- «Новини» - сторінок;
- сторінка перегляду новини.

1. Сторінка «Новини»

Сторінка «Новини» призначена для відображення новин компанії.

Сторінка «Новини» повинна містити:

- «Новини компанії [Назва компанії]» - заголовок;
- блок новини:
- превью новини;
- заголовок новини - текст;
- дата публікації новини - текст;
- короткий опис новини;
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

Новини повинні бути відображені в хронологічному порядку, починаючи з останньої за датою публікації.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

При натисканні на блок новини повинні бути завантажена сторінка перегляду новини.

2. Сторінка перегляду новини

Страница необхідна для просмотра полного текста новости в приложении.

Страница должна содержать:

- «Назад» - кнопка;
- заголовок новини - текст;
- дата публікації новини - текст;
- превью новини - зображення;
- текст новини - текстовий блок;

При натисканні на кнопку «Назад» додаток повинен завантажити сторінку «Новини».

9. Модуль «Профіль»

Мета: модуль необхідно для перегляду інформації про користувача і редагуванні налаштувань повідомлень.

Модуль складається з наступних розділів:

- Профіль - сторінка;
- Налаштування повідомлень - сторінка.

1.1 Сторінка «Профіль»

Сторінка призначена для відображення особистої інформації користувача.

Сторінка повинна містити:

- «Налаштування повідомлень» - кнопка;
- «Профіль» - заголовок;
- фото користувача - зображення;
- прізвище та ім'я користувача - текст;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- блок Email:
 - Email користувача - текст;
 - «Редагувати» - кнопка;
- «Для зміни особистих даних перейдіть до свого кабінету на сайті компанії:" - текст;
- «Перейти» - кнопка;
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

При натисканні на кнопку «Налаштування повідомлень» додаток повинен бути переправлено на сторінку «Налаштування повідомлень».

При натисканні на кнопку «Редагувати» користувачу повинно бути відображено модальне вікно редагування Email з кнопками «ОК» і «Скасувати». При натисканні на кнопку «ОК», повинна бути виконана перевірка в рекрутингу, чи можливо змінити Email. При успішному проходженні перевірки:

- Email повинен бути змінений;
- інформація в профілі користувача повинна бути оновлена.

При зміні Email через додаток користувач не повинен підтверджувати зміну на старому Email-адресу.

При натисканні на кнопку «Перейти»:

- повинен бути відкритий браузер за замовчуванням в системі користувача;
- повинна бути завантажена сторінка з профілем користувача в рекрутингу.

1.2 Сторінка «Налаштування повідомлень»

Сторінка призначена для можливості налаштовувати, які повідомлення користувач буде отримувати від програми.

Сторінка повинна містити:

- «Назад» - кнопка;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- «Налаштування повідомлень» - заголовок;
- «Отримувати повідомлення» - блок прапорців:
 - «Новий користувач в структурі»;
 - «Нова стаття в новинному розділі».

Схематично зовнішній вигляд сторінки представлений на малюнку нижче.

2 Модуль «Історія переглядів»

Мета: модуль необхідний перегляду історії розпізнаних об'єктів і відтворення їх вмісту.

Модуль складається з наступних розділів:

- Історія переглядів - сторінка;
- Сторінка перегляду об'єкта.

2.1 Сторінка «Історія переглядів»

Сторінка призначена для перегляду раніше розпізнаних об'єктів і їх вмісту.

Сторінка повинна містити:

- кнопка зміни відображення;
- «Історія переглядів» - заголовок;
- при обраному відображенні «Плитка»:
 - блок переглянутого об'єкта:
 - **перев'ю – зображення;**
 - **дата перегляду – текст;**
 - блок дублюється для кожного переглянутого об'єкта;
- при обраному відображенні «Один об'єкт»:
 - «Дата: [значення]» - дата перегляду, текст;
 - назва об'єкта - текст;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- матеріал, який відображається при розпізнаванні - відео або зображення;
 - опис об'єкта;
 - «Перейти» - кнопка;
 - блок гортання об'єктів;
- головне меню (див. ТЗ «Меню додатка»);
 - бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

При натисканні на кнопку зміни відображення повинен бути змінений принцип відображення об'єктів на плитку або один об'єкт.

При типі відображення «Плитка» при натисканні на блок переглянутого об'єкта додаток повинен відкрити сторінку перегляду об'єкта.

При типі відображення «Один об'єкт»:

- кнопка «Перейти» відображається в разі, якщо до об'єкта прив'язана посилання (див. ТЗ «Додаток. Адміністратор. Додавання об'єкта»);
- при натисканні на кнопку «Перейти»:
 - повинен бути відкритий браузер за замовчуванням;
 - в браузері повинна бути завантажена посилання, прив'язана до об'єкта.

2.2 Сторінка перегляду об'єкта

Сторінка призначена для перегляду розпізнаного раніше об'єкта та прикріпленого до нього матеріалу.

Сторінка повинна містити:

- «Назад» - кнопка;
- «Історія переглядів» - заголовок;
- «Дата: [значення]» - дата перегляду, текст;
- назва об'єкта - текст;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- матеріал, який відображається при розпізнаванні - відео або зображення;
- опис об'єкта;
- «Перейти» - кнопка.

При натисканні на кнопку «Назад» додаток повинен відобразити сторінку «Профіль».

Якщо вибраний один з прапорців, користувач повинен отримувати push-повідомлення при відповідних змінах.

3 Модуль «Адміністратор. Мої об'єкти»

Мета: модуль призначений для перегляду і редагування об'єктів для розпізнавання. Модуль доступний тільки користувачам з роллю «Адміністратор» в рекрутингу.

Модуль складається з наступних розділів:

- «Мої об'єкти» - сторінка;
- «Інструкція» - сторінка;
- «Додавання об'єкта» - сторінка;
- «Редагування об'єкта» - сторінка.

3.1 Сторінка «Мої об'єкти»

Сторінка призначена переглянути список об'єктів для розпізнавання.

Сторінка повинна містити:

- «Допомога» - кнопка;
- Заголовок;
- «Додати новий об'єкт» - кнопка;
- блок об'єкта:
 - превью об'єкта - зображення;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- назва об'єкта - текст;
- головне меню (див. ТЗ «Меню додатка»);
- бічне меню (див. ТЗ «Меню додатка»), за замовчуванням приховано.

При натисканні на кнопку «Додати об'єкт» додаток повинен відобразити сторінку «Додавання об'єкта».

При натисканні на блок об'єкта додаток повинен відобразити сторінку «Редагування об'єкта».

3.2 Сторінка «Інструкція»

Сторінка призначена для відображення користувачеві інструкції по додаванню об'єкта. Сторінка повинна бути відображена користувачеві при першому переході в розділ «Мої об'єкти».

Сторінка повинна містити:

- «Назад» - кнопка;
- «Як додати об'єкт» - заголовок;
- текстовий блок;
- зображення;
- блок перемикачів;
- «Почати» - кнопка, відображена тільки на останній вкладці.

Схематично зовнішній вигляд сторінки з вкладками відображений на малюнку нижче.

- текстовий блок

Текстовий блок повинен містити таке значення:

- на першій вкладці: «Наведіть камеру на потрібний об'єкт, де присутня ваш бренд або логотип. Це може бути чашка, ручка, блокнот і т.п. »;
- на другій вкладці: «Врахуйте всі деталі: хороше освітлення, ракурс, розташування об'єкта.»;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

➤ на третій вкладці: «За допомогою пальців виберіть область, яка буде розпізнаватися додатком і натисніть "Далі "»;

➤ на четвертій вкладці: «Введіть назву об'єкта, прикріпіть до нього матеріал і посилання на цільову сторінку. Потім натисніть "Зберегти". ».

При натисканні на кнопку «Почати» додаток повинен відкрити сторінку «Додавання об'єкта».

3.3 Сторінка «Додавання об'єкта»

Сторінка призначена для додавання і налаштування об'єкта для розпізнавання.

Сторінка повинна містити:

- «Допомога» - кнопка;
- область вибору об'єкта;
- «Далі» - кнопка.

Схематично зовнішній вигляд сторінки відображений на малюнку нижче.

При натисканні на кнопку «Допомога» додаток доложно відкрити сторінки «Інструкція».

При натисканні на кнопку «Далі»:

- обрана область повинна бути збережена;
- додаток повинен відкрити сторінку «Налаштування об'єкту».

3.3.1 Сторінка «Налаштування об'єкту»

Сторінка призначена для вказівки налаштувань об'єкту при його створенні або редагуванні.

Сторінка повинна містити:

- «Назад» - кнопка;
- «Налаштування об'єкту» - заголовок;
- блок назви об'єкта:

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- іконка;
- «Назва об'єкта» - однорядкове текстове поле;
- блок матеріалу:
 - область з матеріалом;
 - блок вибору матеріалу:
 - іконка;
 - «Виберіть матеріал» – однорядкове текстове поле;
- блок опису:
 - іконка;
 - «Введіть опис» - многострочное текстове поле;
- «Далі» - кнопка.

При отриманні фокусу полем «Виберіть матеріал» повинна бути відкрита сторінка вибору матеріалу (див. П. 1.3.2).

При натисканні на кнопку «Далі»:

- повинна бути виконана перевірка, чи заповнені всі поля;
- при успішному проходженні перевірки програма має бути переправлено на сторінку «Налаштування кнопки» (див. п. 1.3.3).

3.3.2 Сторінка вибору матеріалу

Сторінка призначена для вибору матеріалу для відтворення при розпізнаванні об'єкта.

Сторінка повинна містити:

- «Назад» - кнопка;
- «Виберіть матеріал» - заголовок;
- «Використовувати кнопку» - прапорець;
- «Посилання на Youtube» - блок:

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- іконка;
- однорядкове текстове поле;
- «Додати з пристрою» - блок:
 - іконка;
 - однорядкове текстове поле;
- «Додати» - кнопка.

При отриманні фокусу полем «Виберіть файл» повинна бути у відкритих стандартна сторінка вибору файлів.

При натисканні на кнопку «Додати»:

- повинна бути виконана перевірка, заповнене чи хоча б одне поле;
- при успішному проходженні перевірки матеріал повинен бути прикріплений до об'єкта.

Якщо дані були введені в обидва поля, то пріоритетне до відтворення відео за посиланням з Youtube. Якщо відео за посиланням відтворити не вдається, має бути відтворено прикріплене з файлового сховища відео.

3.3.3 Сторінка «Налаштування кнопки»

Сторінка призначена для вказівки, чи буде використовуватися кнопка в додатку, і редагування її налаштувань.

Сторінка повинна містити:

- «Назад» - кнопка;
- «Налаштування кнопки» - заголовок;
- «Використовувати кнопку» - прапорець;
- блок настройки кнопки - за замовчуванням прихований:
 - перев'ю кнопки згідно налаштувань;
 - «Текст кнопки» - блок:
 - іконка;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

– **однорядкове текстове поле;**

- «Колір кнопки» - кнопка;
- «Колір тексту» - кнопка;
- «Реферальна посилання» - блок:

– **однорядкове текстове поле;**

– **«Використовувати реферальну посилання користувача» – прапорець;**

➤ «Зберегти» - кнопка.

Якщо обраний прапорець «Використовувати кнопку», повинен бути відображений блок настройки кнопки.

При натисканні на кнопку «Колір кнопки» або «Колір тексту» повинен бути відображений віджет з вибором кольорів. Вибрані кольори повинні бути застосовані до кнопки.

При виборі прапорця «Використовувати реферальну посилання користувача»:

- однорядкове текстове поле повинно бути неактивно;
- до кнопки повинна бути прикріплена реферальна посилання, по якій перейшов користувач, що розпізнає об'єкт.

При натисканні на кнопку «Зберегти» повинна бути виконана перевірка на наповненість усіх даних. При успішному проходженні перевірки:

- об'єкт повинен бути збережений;
- зазначені настройки повинні бути збережені для об'єкта
- об'єкт повинен бути доступний в списку об'єктів компанії.

3.4 Сторінка «Редагування об'єкта»

Сторінка призначена для редагування властивостей вже створеного об'єкта.

Сторінка має аналогічний вигляд і функціональність зі сторінкою «Додавання об'єкта». Всі поля повинні бути заповнені попередньо записаними значеннями.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

4 Модуль «Меню додатка»

Мета: меню програми необхідно для зручної навігації користувача.

Модуль складається з наступних розділів:

- "Головне меню":
 - для неавторизованого користувача;
 - для авторизованого користувача
- "Бокове меню":
 - для неавторизованого користувача;
 - для авторизованого користувача;
 - для адміністратора.

4.1 Меню «Головне меню»

Меню призначене для відображення основних функцій програми на всіх сторінках і зручної навігації по ним.

4.1.1 «Головне меню» - неавторизований користувач

Меню призначене для відображення основних функцій програми на всіх сторінках і зручної навігації по ним для неавторизованого користувача.

Меню повинно містити:

- «Меню» - кнопка;
- «Розпізнати» - кнопка;
- «Допомога» - кнопка.

При натисканні на кнопку «Меню» має бути відображена бічне меню програми.

При натисканні на кнопку «Розпізнати» повинна бути відкрита сторінка «Розпізнати».

При натисканні на кнопку «Допомога» повинна бути відображена сторінка «Допомога».

4.1.2 «Головне меню» - авторизований користувач

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Меню призначене для відображення основних функцій програми на всіх сторінках і зручної навігації по ним для авторизованого користувача.

Меню повинно містити:

- «Мої запрошені» - кнопка;
- «Розпізнати» - кнопка;
- «Запросити» - кнопка.

При натисканні на кнопку «Мої запрошені» повинна бути відкрита сторінка «Мої запрошені».

При натисканні на кнопку «Розпізнати» повинна бути відкрита сторінка «Розпізнати».

При натисканні на кнопку «Запросити» повинна бути відображена сторінка «Запросити».

4.2 Меню «Бічне меню»

Меню призначене для відображення всіх функцій програми і зручної навігації по ним. Меню викликається свайпом з лівого краю до центру.

4.2.1 «Бічне меню» - неавторизований користувач

Меню призначене для відображення всіх функцій програми і зручної навігації по ним для неавторизованого користувача.

Меню повинно містити:

- логотип програми;
- «Розпізнати» - блок:
- «Увійти» - блок:

Схематично зовнішній вигляд меню відображений на малюнку нижче.

Меню повинно містити:

- «Мої запрошені» - кнопка;
- «Розпізнати» - кнопка;
- «Запросити» - кнопка.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

При натисканні на кнопку «Розпізнати» повинна бути відкрита сторінка «Розпізнати».

При натисканні на кнопку «Увійти» повинна бути відкрита сторінка «Авторизація».

4.2.2 «Бічне меню» - авторизований користувач

Меню призначене для відображення всіх функцій програми і зручної навігації по ним для авторизованого користувача.

Меню повинно містити:

- логотип програми;
- «Розпізнати» - блок:
- «Мої запрошені» - блок:
- «Запросити» - блок:
- «Новини» - блок:
- «Профіль» - блок:
- «Історія» - блок:
- «Вийти» - блок:

При натисканні на кнопку «Розпізнати» повинна бути відкрита сторінка «Розпізнати».

При натисканні на кнопку «Мої запрошені» повинна бути відкрита сторінка «Мої запрошені».

При натисканні на кнопку «Запросити» повинна бути відкрита сторінка «Запросити».

При натисканні на кнопку «Новини» повинна бути відкрита сторінка «Новини».

При натисканні на кнопку «Профіль» повинна бути відкрита сторінка «Профіль».

При натисканні на кнопку «Історія» повинна бути відкрита сторінка «Історія».

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

При натисканні на кнопку «Вийти»:

- користувач повинен бути розлогінен;
- повинна бути відкрита сторінка «Авторизація».

4.2.3 «Бічне меню» - адміністратор

Меню призначене для відображення всіх функцій програми і зручної навігації по ним для адміністратора.

Меню повинно містити:

- логотип програми;
- «Розпізнати» - блок:
- «Мої об'єкти» - блок:
- «Мої запрошені» - блок:
- «Запросити» - блок:
- «Новини» - блок:
- «Профіль» - блок:
- «Історія» - блок:
- «Вийти» - блок:

При натисканні на кнопку «Розпізнати» повинна бути відкрита сторінка «Розпізнати».

При натисканні на кнопку «Мої об'єкти» повинна бути відкрита сторінка «Мої об'єкти».

При натисканні на кнопку «Мої запрошені» повинна бути відкрита сторінка «Мої запрошені».

При натисканні на кнопку «Запросити» повинна бути відкрита сторінка «Запросити».

При натисканні на кнопку «Новини» повинна бути відкрита сторінка «Новини».

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

При натисканні на кнопку «Профіль» повинна бути відкрита сторінка «Профіль».

При натисканні на кнопку «Історія» повинна бути відкрита сторінка «Історія».

При натисканні на кнопку «Вийти»:

- користувач повинен бути разлогінен;
- повинна бути відкрита сторінка «Авторизація».

5 Модуль «Управління додатком. Суперадмін »

Мета: модуль необхідний для управління додатком, користувачами і об'єктами всіх компаній користувачеві з роллю «Суперадмін».

Модуль складається з наступних розділів:

- «Управління додатком» - сторінка:
 - «Компанії» - вкладка;
 - «Користувачі» - вкладка;
 - «Компанії» - вкладка.

5.1 Сторінка «Управління додатком»

Сторінка призначена для відображення списку всіх користувачів і об'єктів програми. Сторінка доступна в адмін-панелі користувачам роллю «Адміністратор».

Сторінка повинна містити:

- Заголовок;
- навігаційна ланцюжок;
- інформаційний блок:
 - «Завантажено: [значення]» - інформація про те, скільки разів скачали додаток, текст;
 - «Просканувати об'єктів: [значення]» - інформація про те, скільки сумарно раз просканували об'єкти;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- «Об'єкти» - вкладка, відкрита за замовчуванням;
- «Користувачі» - вкладка;
- «Компанії» - вкладка;

5.1.1 Вкладка «Об'єкти»

Вкладка відображає інформацію про створені об'єктах.

Вкладка повинна містити:

- блок настройки посторінкового переходу;
- таблиця з полями:
 - «Прев'ю» - зображення;
 - «Назва» - текст;
 - «Компанія» - текст;
 - «Дата додавання» - текст;
 - «Контент» - текст;
 - «Користувач» - гіперпосилання;
 - «Переглянуто» - текст;
 - «Дії»:
 - «Заблокувати» – кнопка, відображається, якщо об'єкт не заблокований;
 - «Розблокувати» – кнопка, відображається, якщо об'єкт заблокований;
 - «Видалити» – кнопка;
- блок посторінкового переходу.

Поле «Контент» в таблиці повинно містити наступну інформацію:

- «[Тип матеріалу]: [посилання на матеріал]» - інформація про матеріал, прикріпленому до об'єкта. Тип матеріалу може бути зображення або відео. При

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

натисканні на посилання браузер користувача повинен в новій вкладці відтворити прикріплений

➤ «Посилання: [значення]» - посилання, прикріплена до об'єкта. Якщо до об'єкту не прикріплена посилання, поле не повинно бути відображено.

Поле «Користувач» містить логін користувача, який додав об'єкт. При натисканні на гіперпосилання браузер користувача повинен в новій вкладці відкрити сторінку перегляду профілю користувача.

При натисканні на кнопку «Заблокувати» має бути відображена модальне вікно підтвердження дії. При позитивно відповіді:

➤ кнопка повинна бути замінена на кнопку «Розблокувати»;

➤ об'єкт повинен бути недоступний для відтворення через додаток;

➤ об'єкт повинен бути недоступний в додатку в списку об'єктів компанії.

При натисканні на кнопку «Розблокувати» має бути відображена модальне вікно підтвердження дії. При позитивно відповіді:

➤ кнопка повинна бути замінена на кнопку «Заблокувати»;

➤ об'єкт повинен бути доступний для відтворення через додаток;

➤ об'єкт повинен бути доступний в додатку в списку об'єктів компанії.

При натисканні на кнопку «Видалити» має бути відображена модальне вікно підтвердження дії. При підтвердженні дії об'єкт повинен бути вилучений з програми і списку об'єктів компанії.

5.1.2 Вкладка «Користувачі»

Вкладка призначена для відображення інформації по зареєстрованим користувачам, які завантажили додаток.

Вкладка повинна містити:

➤ «Налаштування фільтра» - блок:

- «Логін» - однорядкове текстове поле;
- «Прізвище» - однорядкове текстове поле;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- «Ім'я» - однорядкове текстове поле;
 - «По батькові» - однорядкове текстове поле;
 - «Email» - однорядкове текстове поле;
 - «Логін спонсора» - однорядкове текстове поле;
 - «Дата скачування програми» - віджет вибору дати;
 - «Роль» - список, що випадає ролей на рекрутингу;
 - «Застосувати» - кнопка;
 - «Скинути» - кнопка;
- блок настройки посторінкового переходу;
- таблиця з полями:
- «Фотографія» - зображення;
 - «Логін» - текст;
 - «ПІБ» - текст;
 - «Спонсор» - гіперпосилання;
 - «Компанія» - текст;
 - «Додаток завантажено» - дата скачування, текст;
 - «Об'єктів переглянуто» - текст;
 - «Дії»:
 - «Перегляд» – кнопка;
 - «Заблокувати» – кнопка, відображається, якщо користувач не заблокований;
 - «Розблокувати» – кнопка, відображається, якщо користувач заблокований;
- блок посторінкового переходу.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

При натисканні на гіперпосилання в поле «Спонсор» або на кнопку «Перегляд» браузер користувача повинен бути перенаправлений на сторінку перегляду профілю відповідного користувача.

При натисканні на кнопку «Заблокувати» має бути відображена модальне вікно підтвердження дії. При позитивній відповіді:

- кнопка повинна бути замінена на кнопку «Розблокувати»;
- авторизація та використання додатки повинні бути недоступні для користувача.

При натисканні на кнопку «Розблокувати» має бути відображена модальне вікно підтвердження дії. При позитивній відповіді:

- кнопка повинна бути замінена на кнопку «Заблокувати»;
- авторизація та використання додатки повинні бути доступні для користувача.

5.1.3 Вкладка «Компанії»

Вкладка призначена для відображення інформації про компанії, які використовують додаток.

Вкладка повинна містити:

- «Налаштування фільтра» - блок:
 - «Назва компанії» - однорядкове текстове поле;
 - «Застосувати» - кнопка;
 - «Скинути» - кнопка;
- блок настройки посторінкового переходу;
- таблиця з полями:
 - «Назва компанії» - однорядкове текстове поле;
 - «Адміністратор» - гіперпосилання на профіль користувача;
 - «Об'єктів компанії» - текст;
 - «Об'єктів переглянуто» - текст;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- «Користувачів додатка» - текст;
- «Дії»:
 - «Заблокувати» – кнопка, відображається, якщо компанія не заблокована;
 - «Розблокувати» – кнопка, відображається, якщо компанія заблокована;
- блок посторінкового переходу.

При натисканні на кнопку «Заблокувати» має бути відображена модальне вікно підтвердження дії. При позитивній відповіді:

- кнопка повинна бути замінена на кнопку «Розблокувати»;
- користувачам компанії повинні бути недоступні функції в додатку для авторизованих користувачів;
- об'єкти компанії повинні бути недоступні для розпізнавання.

При натисканні на кнопку «Розблокувати» має бути відображена модальне вікно підтвердження дії. При позитивній відповіді:

- кнопка повинна бути замінена на кнопку «Заблокувати»;
- користувачам компанії повинні бути доступні функції в додатку для авторизованих користувачів;
- об'єкти компанії повинні бути доступні для розпізнавання.

6 Модуль «Управління додатком. компанія »

Мета: модуль необхідний для управління додатком, користувачами і об'єктами адміністратором компанії.

Модуль складається з наступних розділів:

- «Управління додатком» - сторінка:
 - «Компанії» - вкладка;
 - «Користувачі» - вкладка.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

6.1 Сторінка «Управління додатком»

Сторінка призначена для відображення списку всіх користувачів і об'єктів програми. Сторінка доступна в адмін-панелі користувачам роллю «Адміністратор».

Сторінка повинна містити:

- Заголовок;
- навігаційна ланцюжок;
- інформаційний блок:
 - «Завантажено: [значення]» - інформація про те, скільки разів скачали додаток, текст;
 - «Просканувати об'єктів: [значення]» - інформація про те, скільки сумарно раз просканували об'єкти;
- «Об'єкти» - вкладка, відкрита за замовчуванням;
- «Користувачі» - вкладка.

6.1.1 Вкладка «Об'єкти»

Вкладка відображає інформацію про створені об'єктах.

Вкладка повинна містити:

- блок настройки посторінкового переходу;
- таблиця з полями:
 - «Прев'ю» - зображення;
 - «Назва» - текст;
 - «Дата додавання» - текст;
 - «Контент» - текст;
 - «Користувач» - гіперпосилання;
 - «Переглянуто» - текст;
 - «Дії»:

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

– «Заблокувати» – кнопка, відображається, якщо об'єкт не заблокований;

– «Розблокувати» – кнопка, відображається, якщо об'єкт заблокований;

– «Видалити» – кнопка;

➤ блок посторінкового переходу.

Поле «Контент» в таблиці повинно містити наступну інформацію:

➤ «[Тип матеріалу]: [посилання на матеріал]» - інформація про матеріал, прикріпленому до об'єкта. Тип матеріалу може бути зображення або відео. При натисканні на посилання браузер користувача повинен в новій вкладці відтворити прикріплений

➤ «Посилання: [значення]» - посилання, прикріплена до об'єкта. Якщо до об'єкту не прикріплена посилання, поле не повинно бути відображено.

Поле «Користувач» містить логін користувача, який додав об'єкт. При натисканні на гіперпосилання браузер користувача повинен в новій вкладці відкрити сторінку перегляду профілю користувача.

При натисканні на кнопку «Заблокувати» має бути відображена модальне вікно підтвердження дії. При позитивно відповіді:

➤ кнопка повинна бути замінена на кнопку «Розблокувати»;

➤ об'єкт повинен бути недоступний для відтворення через додаток;

➤ об'єкт повинен бути недоступний в додатку в списку об'єктів компанії.

При натисканні на кнопку «Розблокувати» має бути відображена модальне вікно підтвердження дії. При позитивно відповіді:

➤ кнопка повинна бути замінена на кнопку «Заблокувати»;

➤ об'єкт повинен бути доступний для відтворення через додаток;

➤ об'єкт повинен бути доступний в додатку в списку об'єктів компанії.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

При натисканні на кнопку «Видалити» має бути відображена модальне вікно підтвердження дії. При підтвердженні дії об'єкт повинен бути вилучений з програми і списку об'єктів компанії.

6.1.2 Вкладка «Користувачі»

Вкладка призначена для відображення інформації по зареєстрованим користувачам, які завантажили додаток.

Вкладка повинна містити:

- «Налаштування фільтра» - блок:
 - «Логін» - однорядкове текстове поле;
 - «Прізвище» - однорядкове текстове поле;
 - «Ім'я» - однорядкове текстове поле;
 - «По батькові» - однорядкове текстове поле;
 - «Email» - однорядкове текстове поле;
 - «Логін спонсора» - однорядкове текстове поле;
 - «Дата скачування програми» - віджет вибору дати;
 - «Роль» - список, що випадає ролей на рекрутингу;
 - «Застосувати» - кнопка;
 - «Скинути» - кнопка;
- блок настройки посторінкового переходу;
- таблиця з полями:
 - «Фотографія» - зображення;
 - «Логін» - текст;
 - «ПІБ» - текст;
 - «Спонсор» - гіперпосилання;
 - «Додаток завантажено» - дата скачування, текст;
 - «Об'єктів переглянуто» - текст;

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

- «Дії»:
 - «Перегляд» – кнопка;
 - «Заблокувати» – кнопка, відображається, якщо користувач не заблокований;
 - «Розблокувати» – кнопка, відображається, якщо користувач заблокований;
- блок посторінкового переходу.

При натисканні на гіперпосилання в поле «Спонсор» або на кнопку «Перегляд» браузер користувача повинен бути перенаправлений на сторінку перегляду профілю відповідного користувача.

При натисканні на кнопку «Заблокувати» має бути відображена модальне вікно підтвердження дії. При позитивній відповіді авторизація та використання додатки повинні бути недоступні для користувача.

При натисканні на кнопку «Розблокувати» має бути відображена модальне вікно підтвердження дії. При позитивній відповіді авторизація та використання додатки повинні бути доступні для користувача.

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Додаток Б. Дизайн застосування



Рисунок 3.1 – сторінка допомоги

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

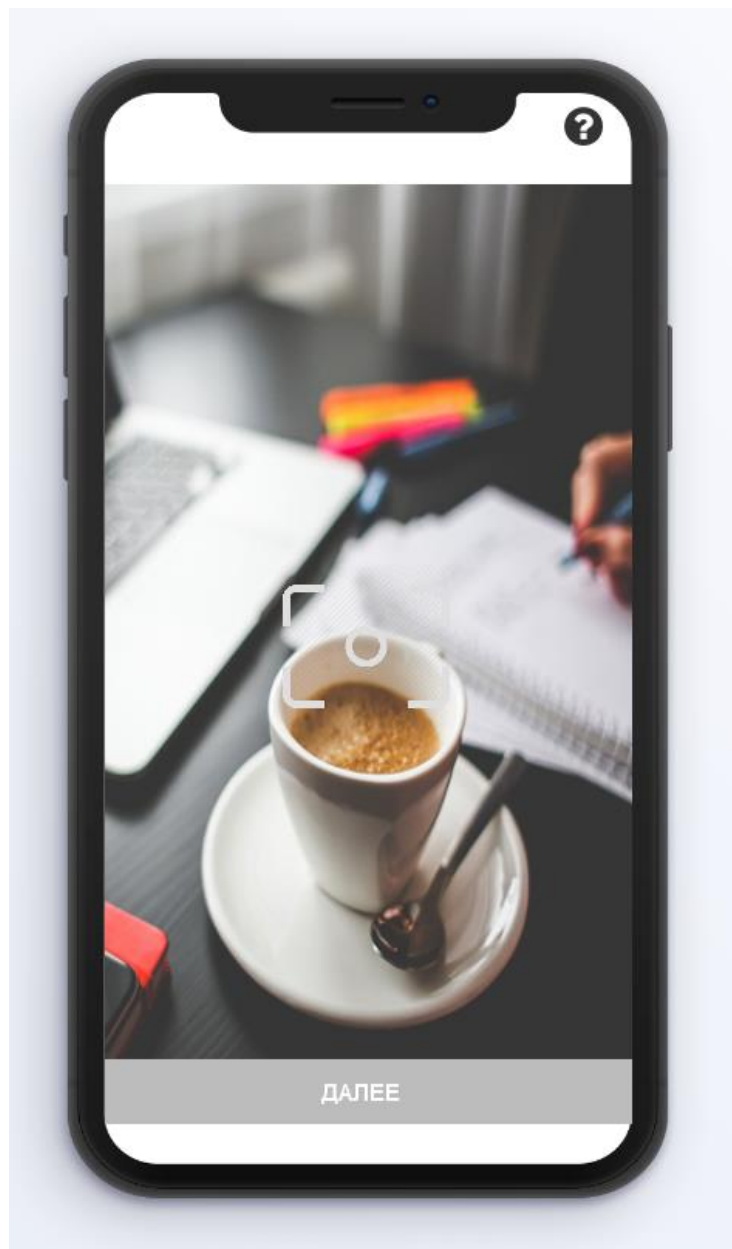


Рисунок 3.2 – сторінка сканера без об'єкта сканування

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

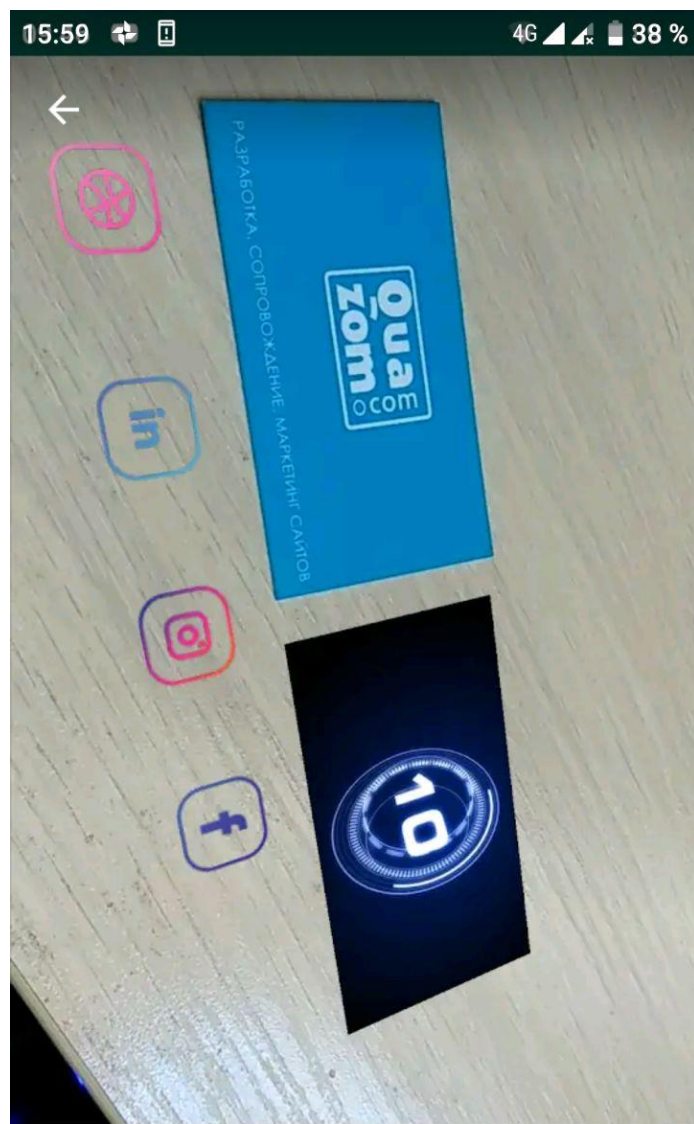


Рисунок 3.3 – сторінка сканера з розпізнаним об'єктом

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		



Рисунок 3.4 – сторінка усіх доданих даним адміністратором об’єктів

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

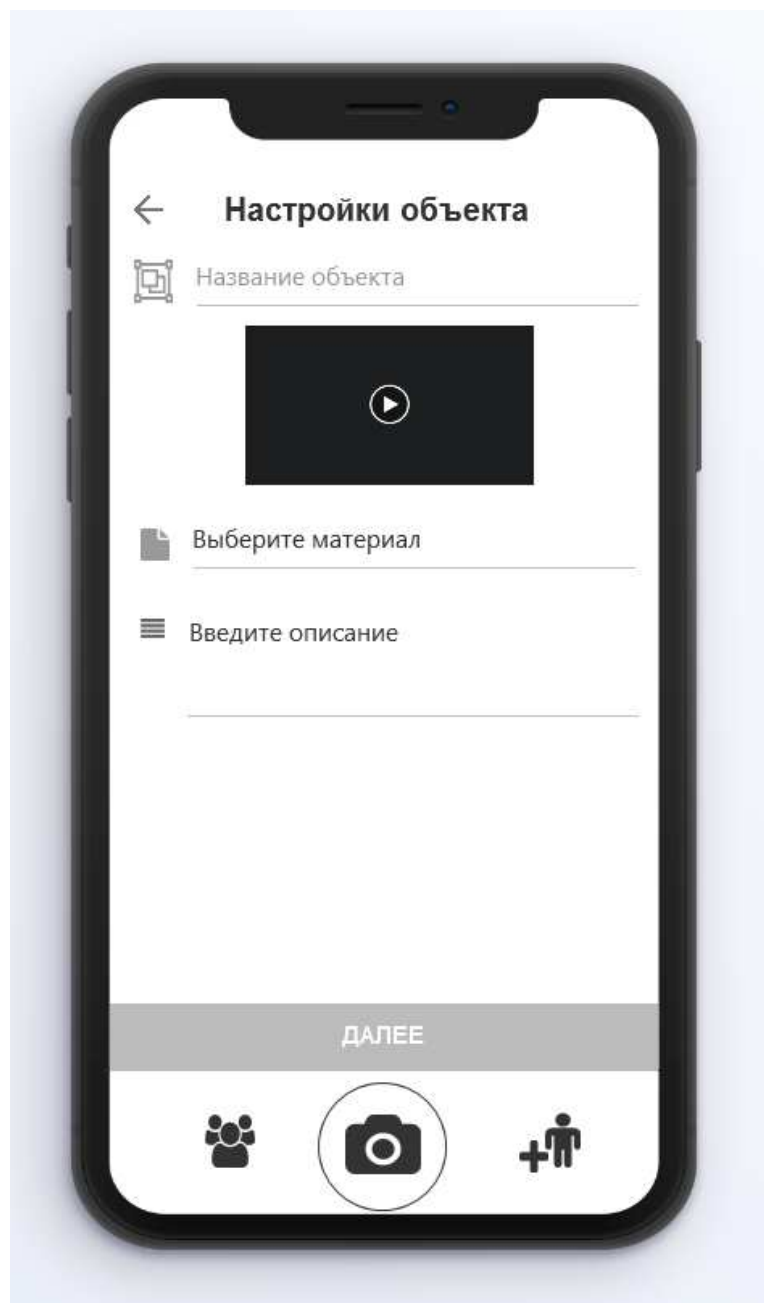


Рисунок 3.5 – Перша сторінка додавання об'єкта

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

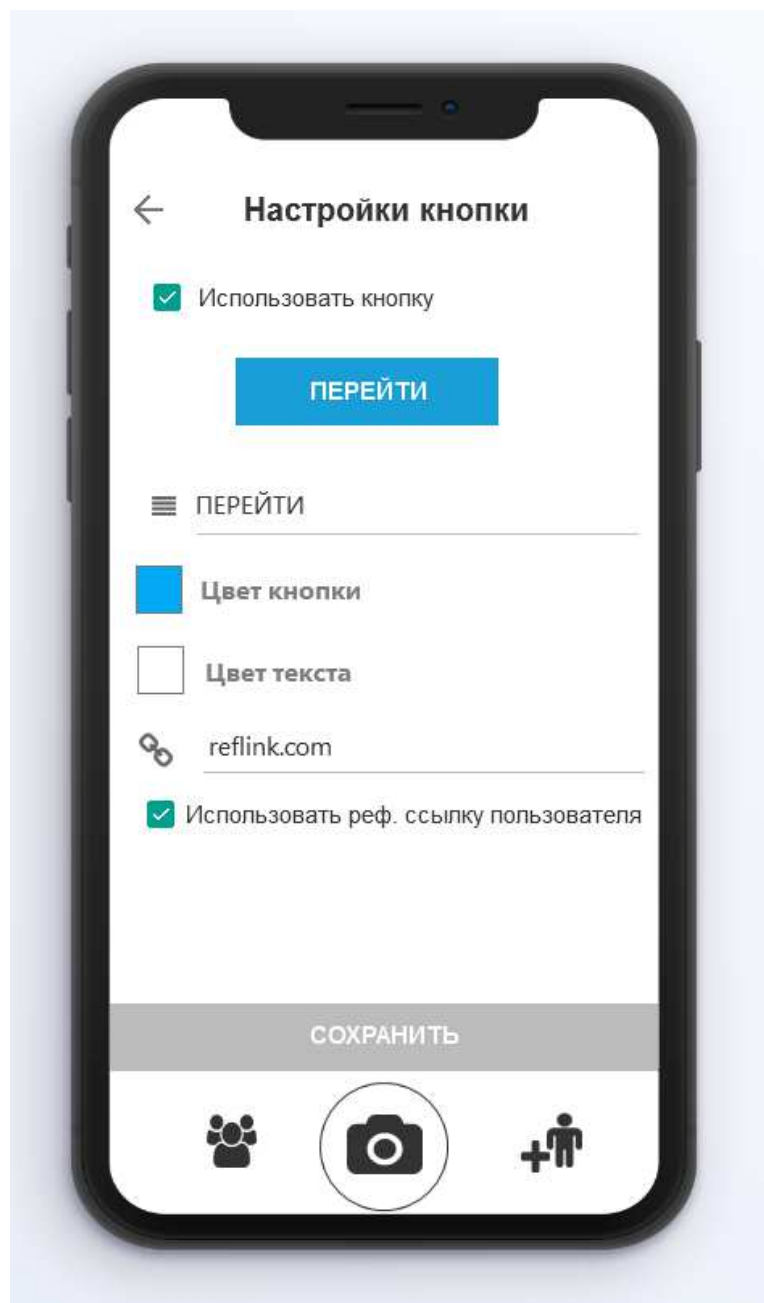


Рисунок 3.6 – Друга сторінка додавання об'єкта

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

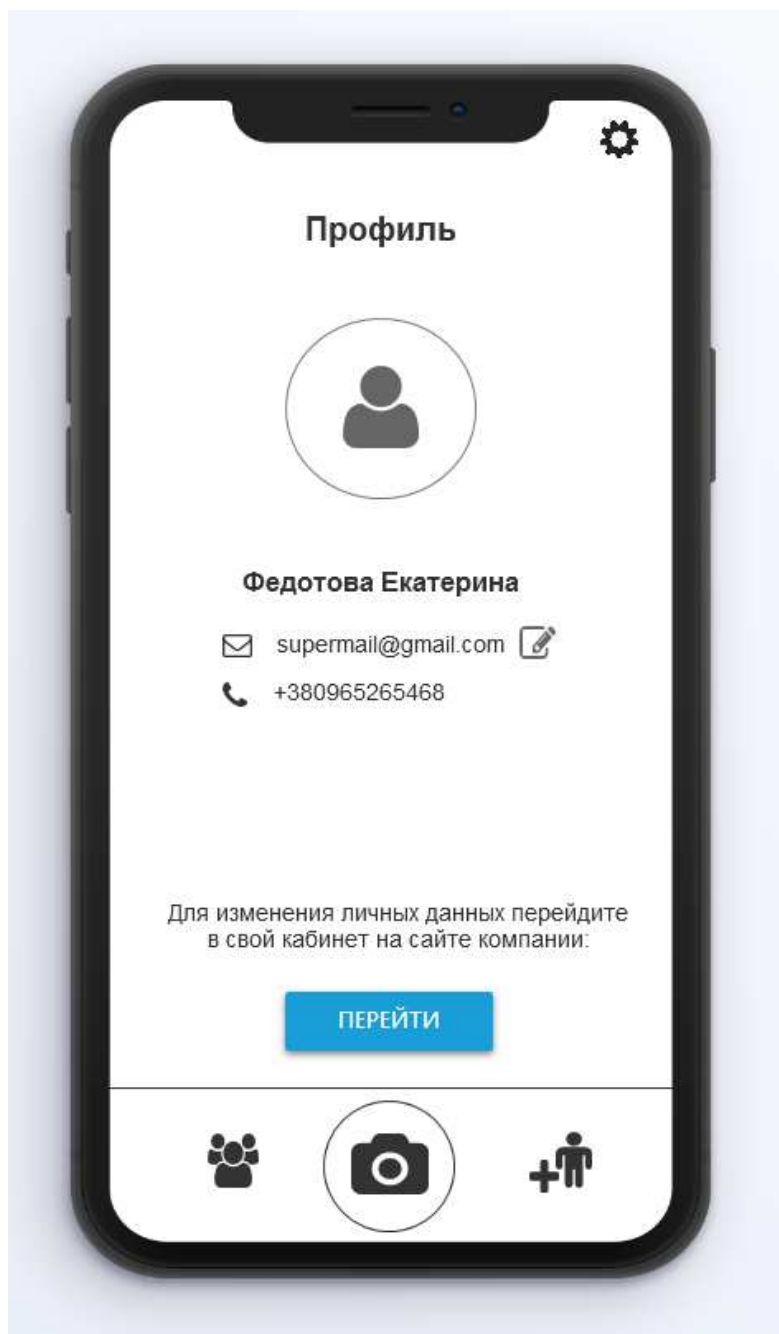


Рисунок 3.7 – Сторінка профілю користувача

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

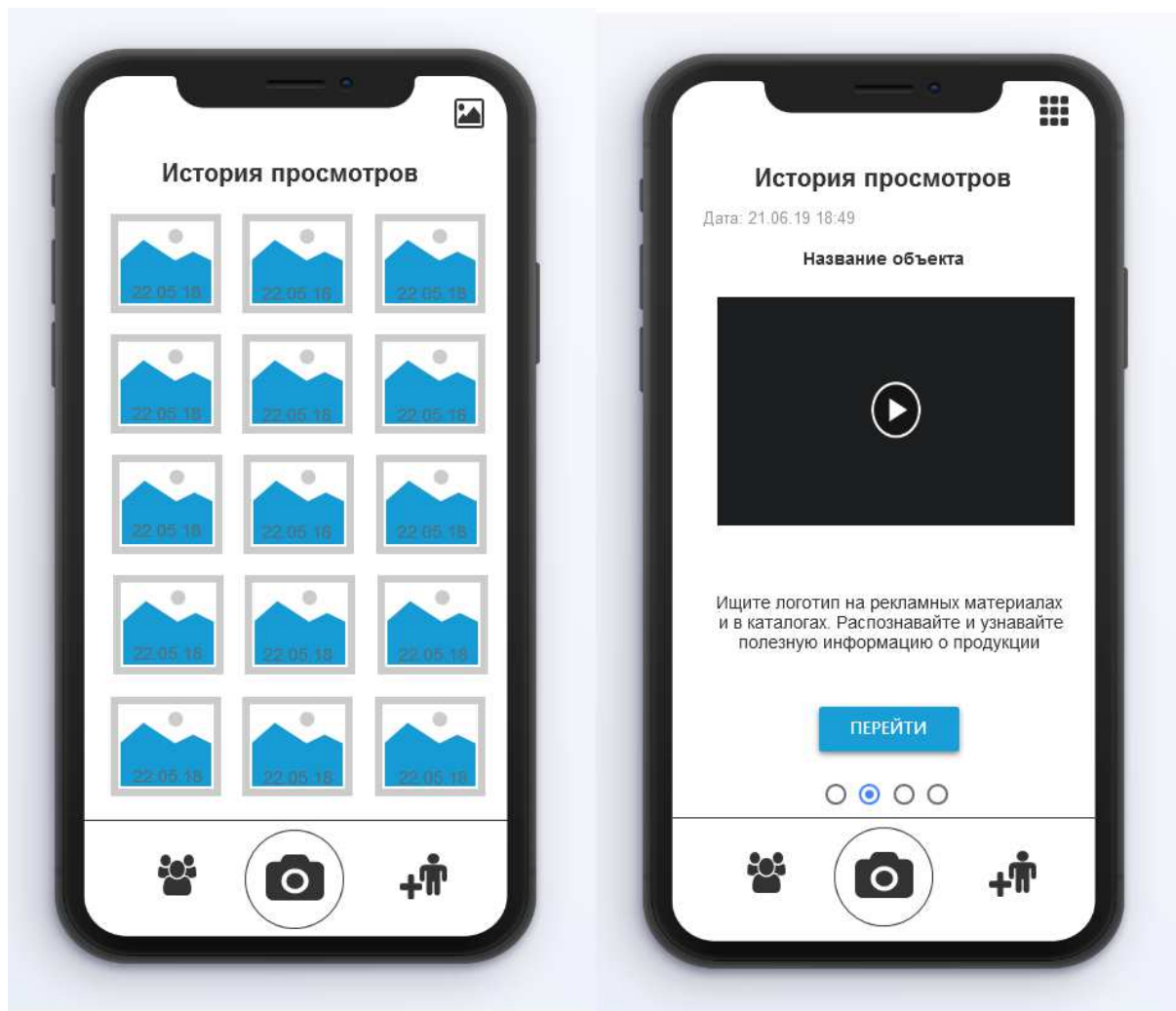


Рисунок 3.8, 3.9 – Сторінка історії переглядів об'єктів

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

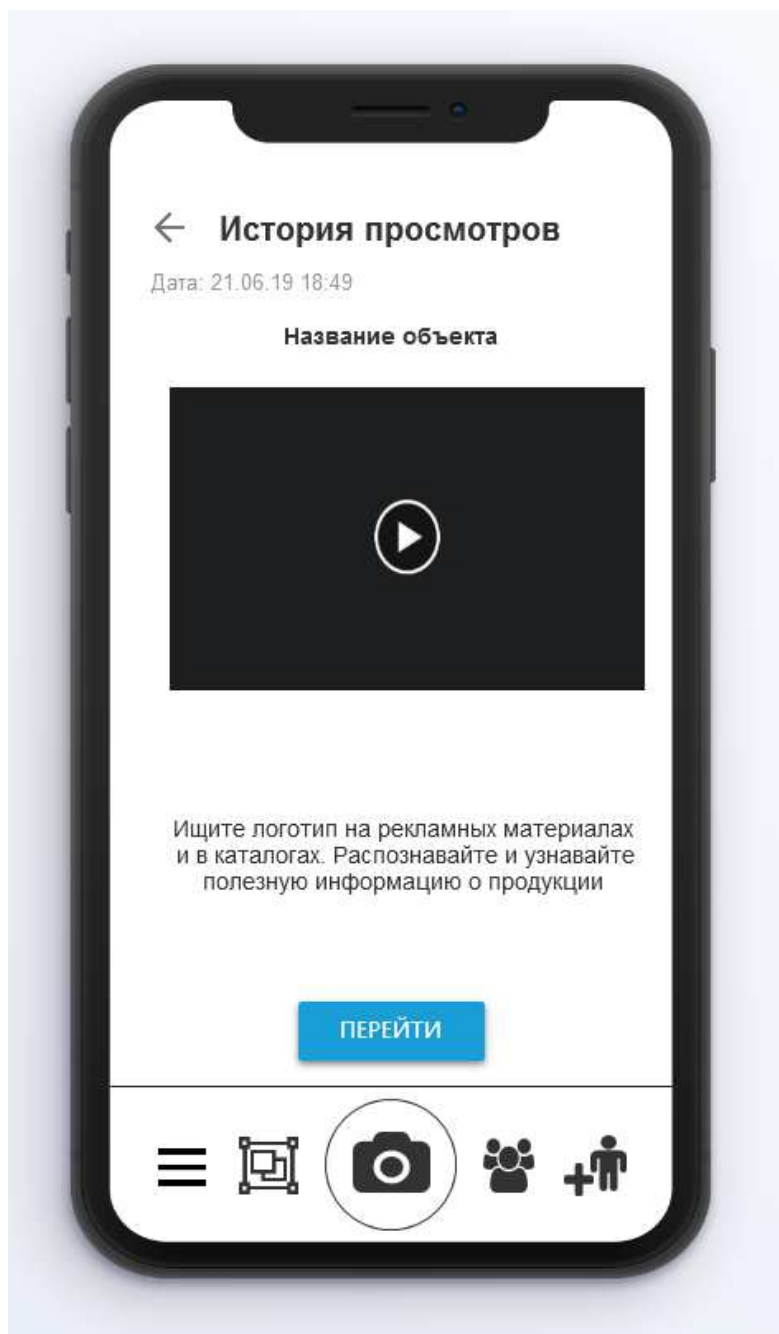


Рисунок 3.10 – Сторінка історії переглядів об’єктів

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

Додаток В. Коди програмних модулів

```
import android.opengl.GLES20;
import android.opengl.Matrix;
import android.util.Log;

import
com.ukrtechinfo.vuforiaTests.Managers.CodingManager.Encryptor;
import com.ukrtechinfo.vuforiaTests.SampleApplication.*;
import
com.ukrtechinfo.vuforiaTests.SampleApplication.SampleApplicationSession;
import com.ukrtechinfo.vuforiaTests.SampleApplication.Utils.*;
import
com.ukrtechinfo.vuforiaTests.app.CloudRecognition.Utils.Utils;
import com.vuforia.DeviceTrackableResult;
import com.vuforia.*;

import java.util.Vector;

/**
 * The renderer class for the CloudReco sample.
 *
 * In the renderFrame() function you can render augmentations to
display over the Target
 */
public class CloudRecoRenderer extends SampleRendererBase implements
SampleAppRendererControl
{
    private static final float OBJECT_SCALE_FLOAT = 0.003f;

    private int shaderProgramID;
    private int vertexHandle;
    private int textureCoordHandle;
    private int.mvpMatrixHandle;
    private int texSampler2DHandle;

    // Object to be rendered
    private Teapot mTeapot;

    private final CloudReco mActivity;

    private boolean mIsTargetCurrentlyTracked = false;

    CloudRecoRenderer(SampleApplicationSession session, CloudReco
activity)
    {
        vuforiaAppSession = session;
        mActivity = activity;
    }
}
```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        // SampleAppRenderer used to encapsulate the use of
        RenderingPrimitives setting
        // the device mode AR/VR and stereo mode
        mSampleAppRenderer = new SampleAppRenderer(this, mActivity,
            vuforiaAppSession.getVideoMode(), 0.010f, 5f);
    }

    public void setActive(boolean active)
    {
        mSampleAppRenderer.setActive(active);
    }

    @Override
    public void initRendering()
    {
        // Define clear color
        GLES20.glClearColor(0.0f, 0.0f, 0.0f,
            Vuforia.requiresAlpha() ? 0.0f
                : 1.0f);

        for (Texture t : mTextures)
        {
            GLES20.glGenTextures(1, t.mTextureID, 0);
            GLES20.glBindTexture(GLES20.GL_TEXTURE_2D,
                t.mTextureID[0]);
            GLES20.glTexParameterf(GLES20.GL_TEXTURE_2D,
                GLES20.GL_TEXTURE_MIN_FILTER, GLES20.GL_LINEAR);
            GLES20.glTexParameterf(GLES20.GL_TEXTURE_2D,
                GLES20.GL_TEXTURE_MAG_FILTER, GLES20.GL_LINEAR);
            GLES20.glTexImage2D(GLES20.GL_TEXTURE_2D, 0,
                GLES20.GL_RGBA,
                t.mWidth, t.mHeight, 0, GLES20.GL_RGBA,
                GLES20.GL_UNSIGNED_BYTE, t.mData);
        }

        shaderProgramID = SampleUtils.createProgramFromShaderSrc(
            CubeShaders.CUBE_MESH_VERTEX_SHADER,
            CubeShaders.CUBE_MESH_FRAGMENT_SHADER);

        vertexHandle = GLES20.glGetAttribLocation(shaderProgramID,
            "vertexPosition");
        textureCoordHandle =
            GLES20.glGetAttribLocation(shaderProgramID,
                "vertexTexCoord");
        mvpMatrixHandle =
            GLES20.glGetUniformLocation(shaderProgramID,
                "modelViewProjectionMatrix");
        texSampler2DHandle =
            GLES20.glGetUniformLocation(shaderProgramID,
                "texSampler2D");
        mTeapot = new Teapot();
    }

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

    }

    public void updateRenderingPrimitives()
    {
        mSampleAppRenderer.updateRenderingPrimitives();
    }

    // The render function.
    // This function is called from the SampleAppRenderer by using
    the RenderingPrimitives views.
    // The state is owned by SampleAppRenderer which is controlling
    its lifecycle.
    // NOTE: State should not be cached outside this method.
    public void renderFrame(State state, float[] projectionMatrix)
    {
        // Renders video background replacing
        Renderer.DrawVideoBackground()
        mSampleAppRenderer.renderVideoBackground(state);

        // Set the device pose matrix as identity
        Matrix44F devicePoseMatrix = SampleMath.Matrix44FIdentity();
        Matrix44F modelMatrix;

        GLES20.glEnable(GLES20.GL_DEPTH_TEST);
        GLES20.glEnable(GLES20.GL_CULL_FACE);

        // Start the target finder if we can't find an Image Target
        result.
        // If the Device pose exists, we can assume we will receive
        two
        // Trackable Results if the ImageTargetResult is available:
        // ImageTargetResult and DeviceTrackableResult
        int numExpectedResults = state.getDeviceTrackableResult() ==
        null ? 0 : 1;
        if (state.getTrackableResults().size() <=
        numExpectedResults)
        {
            mActivity.startFinderIfStopped();
        }

        // Read device pose from the state and create a
        corresponding view matrix (inverse of the device pose)
        if (state.getDeviceTrackableResult() != null)
        {
            int statusInfo =
            state.getDeviceTrackableResult().getStatusInfo();
            int trackerStatus =
            state.getDeviceTrackableResult().getStatus();

            mActivity.checkForRelocalization(statusInfo);
        }
    }

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		


```

        if (trackerStatus != TrackableResult.STATUS.NO_POSE)
        {
            modelMatrix =
Tool.convertPose2GLMatrix(state.getDeviceTrackableResult().getPose()
);

            // We transpose here because Matrix44FInverse
returns a transposed matrix
            devicePoseMatrix =
SampleMath.Matrix44FTranspose(SampleMath.Matrix44FInverse(modelMatri
x));
        }
    }

    TrackableResultList trackableResultList =
state.getTrackableResults();

    // Determine if target is currently being tracked
setIsTargetCurrentlyTracked(trackableResultList);

    // Iterate through trackable results and render any
augmentations
    for (TrackableResult result : trackableResultList)
    {
        Trackable trackable = result.getTrackable();

        modelMatrix =
Tool.convertPose2GLMatrix(result.getPose());

        Log.e("Render", trackable.getName());

mActivity.setText(Encrptor.decrypt(((ImageTarget)trackable).getMeta
Data()));
        mActivity.stopFinderIfStarted();

        if (result.isOfType(ImageTargetResult.getClassType()))
        {
            // Renders the augmentation
//            renderModel(projectionMatrix,
devicePoseMatrix.getData(), modelMatrix.getData());
//
//            SampleUtils.checkGLError("CloudReco renderFrame");
        }
    }

//    GLES20.glDisable(GLES20.GL_DEPTH_TEST);

    Renderer.getInstance().end();
}

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

private void renderModel(float[] projectionMatrix, float[]
viewMatrix, float[] modelMatrix)
{
    int textureIndex = 0;
    float[] modelViewProjection = new float[16];

    // Apply local transformation to our model
    Matrix.translateM(modelMatrix, 0, 0.0f, 0.0f,
OBJECT_SCALE_FLOAT);
    Matrix.scaleM(modelMatrix, 0, OBJECT_SCALE_FLOAT,
OBJECT_SCALE_FLOAT, OBJECT_SCALE_FLOAT);

    // Combine device pose (view matrix) with model matrix
    Matrix.multiplyMM(modelMatrix, 0, viewMatrix, 0,
modelMatrix, 0);

    // Do the final combination with the projection matrix
    Matrix.multiplyMM(modelViewProjection, 0, projectionMatrix,
0, modelMatrix, 0);

    // activate the shader program and bind the
vertex/normal/texture coords
    GLES20.glUseProgram(shaderProgramID);
    GLES20.glVertexAttribPointer(vertexHandle, 3,
GLES20.GL_FLOAT, false,
0, mTeapot.getVertices());
    GLES20.glVertexAttribPointer(textureCoordHandle, 2,
GLES20.GL_FLOAT,
false, 0, mTeapot.getTexCoords());

    GLES20.glEnableVertexAttribArray(vertexHandle);
    GLES20.glEnableVertexAttribArray(textureCoordHandle);

    // activate texture 0, bind it, and pass to shader
    GLES20.glActiveTexture(GLES20.GL_TEXTURE0);
    GLES20.glBindTexture(GLES20.GL_TEXTURE_2D,
mTextures.get(textureIndex).mTextureID[0]);
    GLES20.glUniform1i(texSampler2DHandle, 0);

    // pass the model view matrix to the shader
    GLES20.glUniformMatrix4fv(mvpMatrixHandle, 1, false,
modelViewProjection, 0);

    // finally draw the teapot
    GLES20.glDrawElements(GLES20.GL_TRIANGLES,
mTeapot.getNumObjectIndex(),
GLES20.GL_UNSIGNED_SHORT, mTeapot.getIndices());

    // disable the enabled arrays
    GLES20.glDisableVertexAttribArray(vertexHandle);
    GLES20.glDisableVertexAttribArray(textureCoordHandle);
}

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

    public void setTextures(Vector<Texture> textures)
    {
        mTextures = textures;
    }

    private void setIsTargetCurrentlyTracked(TrackableResultList
trackableResultList)
    {
        for(TrackableResult result : trackableResultList)
        {

            Log.e("result", result.toString());

            // Check the tracking status for result types
            // other than DeviceTrackableResult. ie:
ImageTargetResult
            if
(!result.isOfType(DeviceTrackableResult.getClassType()))
            {
                int currentStatus = result.getStatus();
                int currentStatusInfo = result.getStatusInfo();

                // The target is currently being tracked if the
status is TRACKED|NORMAL
                if (currentStatus == TrackableResult.STATUS.TRACKED
                    || currentStatusInfo ==
TrackableResult.STATUS_INFO.NORMAL)
                {
                    mIsTargetCurrentlyTracked = true;
                    return;
                }
            }

            mIsTargetCurrentlyTracked = false;
        }

        boolean isTargetCurrentlyTracked()
        {
            return mIsTargetCurrentlyTracked;
        }
    }

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.pm.ActivityInfo;
import android.content.res.Configuration;
import android.graphics.Color;
import android.os.Bundle;

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

import android.os.Handler;
import android.util.Log;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup.LayoutParams;
import android.view.animation.Animation;
import android.view.animation.LinearInterpolator;
import android.view.animation.TranslateAnimation;
import android.widget.Button;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;

import
com.ukrtechinfo.vuforiaTests.app.ui.SampleAppMenu.SampleAppMenu;
import
com.ukrtechinfo.vuforiaTests.app.ui.SampleAppMenu.SampleAppMenuGroup
;
import
com.ukrtechinfo.vuforiaTests.app.ui.SampleAppMenu.SampleAppMenuInter
face;
import com.ukrtechinfo.vuforiaTests.R;
import com.ukrtechinfo.vuforiaTests.SampleApplication.*;
import
com.ukrtechinfo.vuforiaTests.SampleApplication.utils.LoadingDialogHa
ndler;
import
com.ukrtechinfo.vuforiaTests.SampleApplication.utils.SampleAppTimer;
import
com.ukrtechinfo.vuforiaTests.SampleApplication.utils.SampleApplicati
onGLView;
import com.ukrtechinfo.vuforiaTests.SampleApplication.utils.Texture;
import
com.ukrtechinfo.vuforiaTests.SampleApplication.utils.VuforiaUtils;
import com.vuforia.*;
import com.ukrtechinfo.vuforiaTests.app.ui.SampleAppMessag;

import java.util.ArrayList;
import java.util.Vector;

/**
 * The main activity for the CloudReco sample.
 * Cloud Recognition allows users to detect a dataset using an
internet connection by
 * creating and storing the dataset on the cloud (via the Target
Manager) as opposed
 * to storing the dataset within your project.
 *
 * This class does high-level handling of the Vuforia lifecycle and
any UI updates

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

*
* For CloudReco-specific rendering, check out
CloudRecoRenderer.java
* For the low-level Vuforia lifecycle code, check out
SampleApplicationSession.java
*/
public class CloudReco extends SampleActivityBase implements
SampleApplicationControl,
    SampleAppMenuInterface {

    private static final String LOGTAG = "CloudReco";

    private SampleApplicationSession vuforiaAppSession;

    // Cloud Recognition specific error codes
    // These codes match the ones defined for the TargetFinder in
Vuforia.jar
    private static final int UPDATE_ERROR_AUTHORIZATION_FAILED = -1;
    private static final int UPDATE_ERROR_PROJECT_SUSPENDED = -2;
    private static final int UPDATE_ERROR_NO_NETWORK_CONNECTION = -
3;
    private static final int UPDATE_ERROR_SERVICE_NOT_AVAILABLE = -
4;
    private static final int UPDATE_ERROR_BAD_FRAME_QUALITY = -5;
    private static final int UPDATE_ERROR_UPDATE_SDK = -6;
    private static final int UPDATE_ERROR_TIMESTAMP_OUT_OF_RANGE = -
7;
    private static final int UPDATE_ERROR_REQUEST_TIMEOUT = -8;

    private SampleApplicationGLView mGLView;

    private CloudRecoRenderer mRenderer;

    private SampleAppMenu mSampleAppMenu;
    ArrayList<View> mSettingsAdditionalViews = new ArrayList<>();

    private SampleAppMessag mSampleAppMessage;
    private SampleAppTimer mRelocalizationTimer;
    private SampleAppTimer mStatusDelayTimer;

    private int mCurrentStatusInfo;

    private boolean mDeviceTracker = false;
    private boolean mFinderStarted = false;
    private boolean mResetTargetFinderTrackables = false;

    // The textures we will use for rendering:
    private Vector<Texture> mTextures;

    // The keys necessary in order to use Cloud Recognition
    // Generate your own keys via the Target Manager on the Vuforia
developer website

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

// View overlays to be displayed in the Augmented View
private RelativeLayout mUILayout;
private Button mBtnLayout;

// Error message handling:
private int mlastErrorCode = 0;
private int mInitErrorCode = 0;
private boolean mFinishActivityOnError;

// Alert Dialog used to display SDK errors
private AlertDialog mErrorDialog;

private GestureDetector mGestureDetector;

private final LoadingDialogHandler loadingDialogHandler = new
LoadingDialogHandler(
    this);

// Scan line and animation
private View scanLine;
private TranslateAnimation scanAnimation;

private double mLastErrorTime;

private boolean mIsDroidDevice = false;

// The TargetFinder is used to dynamically search
// for targets using an internet connection
private TargetFinder mTargetFinder;

private TextView textView;

@Override
protected void onCreate(Bundle savedInstanceState)
{
    Log.d(LOGTAG, "onCreate");
    super.onCreate(savedInstanceState);
    setResult(RESULT_OK);

    vuforiaAppSession = new SampleApplicationSession(this);

    startLoadingAnimation();

    vuforiaAppSession
        .initAR(this,
ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

    // Creates the GestureDetector listener for processing
double tap
    mGestureDetector = new GestureDetector(this, new
GestureListener());

    mTextures = new Vector<>();

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        loadTextures();

        mIsDroidDevice =
        android.os.Build.MODEL.toLowerCase().startsWith("droid");

        // Relocalization timer and message
        mSampleAppMessage = new SampleAppMessag(this, mUILayout,
        mUILayout.findViewById(R.id.topbar_layout), false);
        mRelocalizationTimer = new SampleAppTimer(10000, 1000)
        {
            @Override
            public void onFinish()
            {
                if (vuforiaAppSession != null)
                {
                    vuforiaAppSession.resetDeviceTracker();
                }

                super.onFinish();
            }
        };

        mStatusDelayTimer = new SampleAppTimer(1000, 1000)
        {
            @Override
            public void onFinish()
            {
                if (mRenderer.isTargetCurrentlyTracked())
                {
                    super.onFinish();
                    return;
                }

                if (!mRelocalizationTimer.isRunning())
                {
                    mRelocalizationTimer.startTimer();
                }

                runOnUiThread(new Runnable()
                {
                    @Override
                    public void run()
                    {
                        mSampleAppMessage.show(getString(R.string.instruct_relocalize));
                    }
                });

                super.onFinish();
            }
        };
    }
}

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

@Override
public void onPointerCaptureChanged(boolean hasCapture) {

}

public void setText(String name) {
    CharSequence text = textView.getText();
    Log.e("name", text.toString());
//    textView.setVisibility(View.VISIBLE);

    if(!name.equals(text.toString())){
        textView.setText(name);
    }
}

private class GestureListener extends
    GestureDetector.SimpleOnGestureListener
{
    // Used to set autofocus one second after a manual focus is
    triggered
    private final Handler autofocusHandler = new Handler();

    @Override
    public boolean onDown(MotionEvent e)
    {
        return true;
    }

    // Process Single Tap event to trigger autofocus
    @Override
    public boolean onSingleTapUp(MotionEvent e)
    {
        boolean result =
CameraDevice.getInstance().setFocusMode(
            CameraDevice.FOCUS_MODE.FOCUS_MODE_TRIGGERAUTO);
        if (!result)
            Log.e("SingleTapUp", "Unable to trigger focus");

        // Generates a Handler to trigger continuous auto-focus
        // after 1 second
        autofocusHandler.postDelayed(new Runnable()
        {
            public void run()
            {
                final boolean autofocusResult =
CameraDevice.getInstance().setFocusMode(
CameraDevice.FOCUS_MODE.FOCUS_MODE_CONTINUOUSAUTO);

                if (!autofocusResult)
                    Log.e("SingleTapUp", "Unable to re-enable
continuous auto-focus");
            }
        }, 1000);
    }
}

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		


```

        }
        }, 1000L);

        return true;
    }
}

// Load specific textures from the APK, which we will later use
for rendering.
private void loadTextures()
{
mTextures.add(Texture.loadTextureFromApk("TextureTeapotRed.png",
        getAssets()));
}

@Override
protected void onResume()
{
    Log.d(LOGTAG, "onResume");
    super.onResume();

    showProgressIndicator(true);

    // This is needed for some Droid devices to force portrait
    if (mIsDroidDevice)
    {

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    }

    vuforiaAppSession.onResume();
}

// Called whenever the device orientation or screen resolution
changes
@Override
public void onConfigurationChanged(Configuration config)
{
    Log.d(LOGTAG, "onConfigurationChanged");
    super.onConfigurationChanged(config);

    vuforiaAppSession.onConfigurationChanged();
}

@Override
protected void onPause()

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

{
    Log.d(LOGTAG, "onPause");
    super.onPause();

    vuforiaAppSession.onPause();

    if (mGableView != null)
    {
        mGableView.setVisibility(View.INVISIBLE);
        mGableView.onPause();
    }
}

@Override
protected void onDestroy()
{
    Log.d(LOGTAG, "onDestroy");
    super.onDestroy();

    deinitCloudReco();

    try
    {
        vuforiaAppSession.stopAR();
    } catch (SampleApplicationException e) {
        Log.e(LOGTAG, e.getString());
    }

    System.gc();
}

private void deinitCloudReco()
{
    if (mTargetFinder == null)
    {
        Log.e(LOGTAG,
            "Could not deinit cloud reco because it was not
initialized");
        return;
    }

    mTargetFinder.deinit();
}

private void startLoadingAnimation()
{
    // Inflates the Overlay Layout to be displayed above the
    Camera View
    mUILayout = (RelativeLayout) View.inflate(this,
        R.layout.camera_overlay_with_scanline,

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        null);

mUILayout.setVisibility(View.VISIBLE);
mUILayout.setBackgroundColor(Color.BLACK);

RelativeLayout topbarLayout =
mUILayout.findViewById(R.id.topbar_layout);
topbarLayout.setVisibility(View.VISIBLE);

TextView title = mUILayout.findViewById(R.id.topbar_title);
title.setText(getText(R.string.feature_cloud_reco));

mSettingsAdditionalViews.add(topbarLayout);

textView = mUILayout.findViewById(R.id.textView);
//    textView.setVisibility(View.INVISIBLE);

mBtnLayout = mUILayout.findViewById(R.id.reset_btn);

loadingDialogHandler.mLoadingDialogContainer = mUILayout
    .findViewById(R.id.loading_indicator);
loadingDialogHandler.mLoadingDialogContainer
    .setVisibility(View.VISIBLE);

scanLine = mUILayout.findViewById(R.id.scan_line);
scanLine.setVisibility(View.GONE);
scanAnimation = new TranslateAnimation(
    TranslateAnimation.ABSOLUTE, 0f,
    TranslateAnimation.ABSOLUTE, 0f,
    TranslateAnimation.RELATIVE_TO_PARENT, 0f,
    TranslateAnimation.RELATIVE_TO_PARENT, 1.0f);
scanAnimation.setDuration(4000);
scanAnimation.setRepeatCount(-1);
scanAnimation.setRepeatMode(Animation.REVERSE);
scanAnimation.setInterpolator(new LinearInterpolator());

addContentView(mUILayout, new
LayoutParams(LayoutParams.MATCH_PARENT,
    LayoutParams.MATCH_PARENT));
}

private void initApplicationAR()
{
    // Create OpenGL ES view:
    int depthSize = 16;
    int stencilSize = 0;
    boolean translucent = Vuforia.requiresAlpha();

    // Initialize the GLView with proper flags
    mGLView = new SampleApplicationGLView(this);
    mGLView.init(translucent, depthSize, stencilSize);

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        // Sets up the Renderer of the GLView
        mRenderer = new CloudRecoRenderer(vuforiaAppSession, this);
        mRenderer.setTextures(mTextures);
        mGLView.setRenderer(mRenderer);
        mGLView.setPreserveEGLContextOnPause(true);

        setRendererReference(mRenderer);
    }

    // Returns the error message for each error code
    private String getStatusDescString(int code)
    {
        if (code == UPDATE_ERROR_AUTHORIZATION_FAILED)
            return
                getString(R.string.UPDATE_ERROR_AUTHORIZATION_FAILED_DESC);
        if (code == UPDATE_ERROR_PROJECT_SUSPENDED)
            return
                getString(R.string.UPDATE_ERROR_PROJECT_SUSPENDED_DESC);
        if (code == UPDATE_ERROR_NO_NETWORK_CONNECTION)
            return
                getString(R.string.UPDATE_ERROR_NO_NETWORK_CONNECTION_DESC);
        if (code == UPDATE_ERROR_SERVICE_NOT_AVAILABLE)
            return
                getString(R.string.UPDATE_ERROR_SERVICE_NOT_AVAILABLE_DESC);
        if (code == UPDATE_ERROR_UPDATE_SDK)
            return getString(R.string.UPDATE_ERROR_UPDATE_SDK_DESC);
        if (code == UPDATE_ERROR_TIMESTAMP_OUT_OF_RANGE)
            return
                getString(R.string.UPDATE_ERROR_TIMESTAMP_OUT_OF_RANGE_DESC);
        if (code == UPDATE_ERROR_REQUEST_TIMEOUT)
            return
                getString(R.string.UPDATE_ERROR_REQUEST_TIMEOUT_DESC);
        if (code == UPDATE_ERROR_BAD_FRAME_QUALITY)
            return
                getString(R.string.UPDATE_ERROR_BAD_FRAME_QUALITY_DESC);
        else
        {
            return getString(R.string.UPDATE_ERROR_UNKNOWN_DESC);
        }
    }

    // Returns the error message header for each error code
    private String getStatusTitleString(int code)
    {
        if (code == UPDATE_ERROR_AUTHORIZATION_FAILED)
            return
                getString(R.string.UPDATE_ERROR_AUTHORIZATION_FAILED_TITLE);
        if (code == UPDATE_ERROR_PROJECT_SUSPENDED)
            return
                getString(R.string.UPDATE_ERROR_PROJECT_SUSPENDED_TITLE);
        if (code == UPDATE_ERROR_NO_NETWORK_CONNECTION)

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        return
getString(R.string.UPDATE_ERROR_NO_NETWORK_CONNECTION_TITLE);
        if (code == UPDATE_ERROR_SERVICE_NOT_AVAILABLE)
            return
getString(R.string.UPDATE_ERROR_SERVICE_NOT_AVAILABLE_TITLE);
        if (code == UPDATE_ERROR_UPDATE_SDK)
            return
getString(R.string.UPDATE_ERROR_UPDATE_SDK_TITLE);
        if (code == UPDATE_ERROR_TIMESTAMP_OUT_OF_RANGE)
            return
getString(R.string.UPDATE_ERROR_TIMESTAMP_OUT_OF_RANGE_TITLE);
        if (code == UPDATE_ERROR_REQUEST_TIMEOUT)
            return
getString(R.string.UPDATE_ERROR_REQUEST_TIMEOUT_TITLE);
        if (code == UPDATE_ERROR_BAD_FRAME_QUALITY)
            return
getString(R.string.UPDATE_ERROR_BAD_FRAME_QUALITY_TITLE);
        else
        {
            return getString(R.string.UPDATE_ERROR_UNKNOWN_TITLE);
        }
    }

```

```

    private void showErrorMessage(int errorCode, double errorTime,
boolean finishActivityOnError)
    {
        if (errorTime < (mLastErrorTime + 5.0) || errorCode ==
mlastErrorCode)
            return;

```

```

        mLastErrorTime = errorTime;
        mlastErrorCode = errorCode;
        mFinishActivityOnError = finishActivityOnError;

```

```

        runOnUiThread(new Runnable()
        {
            public void run()
            {
                if (mErrorDialog != null)
                {
                    mErrorDialog.dismiss();
                }

                // Generates an Alert Dialog to show the error
message
                AlertDialog.Builder builder = new
AlertDialog.Builder(
                    CloudReco.this);
                builder
                    .setMessage(
getStatusDescString(CloudReco.this.mlastErrorCode))

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        .setTitle(
getStatusTitleString(CloudReco.this.mlastErrorCode))
        .setCancelable(false)
        .setIcon(0)

        .setPositiveButton(getString(R.string.button_OK),
                                new
DialogInterface.OnClickListener()
                                {
                                    public void
onClick(DialogInterface dialog, int id)
                                    {
                                        if(mFinishActivityOnError)
                                        {
                                            finish();
                                        }
                                        else
                                        {
                                            dialog.dismiss();
                                        }
                                    }
                                }
        ));

        mErrorDialog = builder.create();
        mErrorDialog.show();
    }
    });
}

private void showInitializationErrorMessage(String message)
{
    final String errorMessage = message;
    runOnUiThread(new Runnable()
    {
        public void run()
        {
            if (mErrorDialog != null)
            {
                mErrorDialog.dismiss();
            }

            // Generates an Alert Dialog to show the error
message
            AlertDialog.Builder builder = new
AlertDialog.Builder(
                CloudReco.this);
            builder
                .setMessage(errorMessage)
                .setTitle(getString(R.string.INIT_ERROR))
                .setCancelable(false)
                .setIcon(0)

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

.setPositiveButton(getString(R.string.button_OK),
                    new
DialogInterface.OnClickListener()
                    {
                        public void
onClick(DialogInterface dialog, int id)
                        {
                            finish();
                        }
                    });

        mErrorDialog = builder.create();
        mErrorDialog.show();
    }
});
}

public void startFinderIfStopped()
{
    if(!mFinderStarted)
    {
        if (mTargetFinder == null)
        {
            Log.e(LOGTAG, "Tried to start TargetFinder but was
not initialized");
            return;
        }

        mTargetFinder.startRecognition();
        scanlineStart();
        textView.setText("");
        mFinderStarted = true;
    }
}

public void stopFinderIfStarted()
{
    if(mFinderStarted)
    {
        if (mTargetFinder == null)
        {
            Log.e(LOGTAG, "Tried to stop TargetFinder but was
not initialized");
            return;
        }

        mTargetFinder.stop();
        scanlineStop();
        //        textView.setVisibility(View.VISIBLE);
        mFinderStarted = false;
    }
}

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

    }
}

@Override
public boolean onTouchEvent(MotionEvent event)
{
    // Process the Gestures
    return (mSampleAppMenu != null &&
mSampleAppMenu.processEvent(event))
        || mGestureDetector.onTouchEvent(event);
//    return false;
}

@Override
public boolean doLoadTrackersData()
{
    Log.d(LOGTAG, "initCloudReco");

    // Get the object tracker:
    TrackerManager trackerManager =
TrackerManager.getInstance();
    ObjectTracker objectTracker = (ObjectTracker) trackerManager
        .getTracker(ObjectTracker.getClassType());

    // Start the target finder using keys
    TargetFinder targetFinder = objectTracker.getTargetFinder();

targetFinder.startInit(VuforiaUtils.CLOUDE_CLIENT_ACCESS_KEY,
VuforiaUtils.CLOUDE_CLIENT_SECRET_KEY);

    targetFinder.waitUntilInitFinished();

    int resultCode = targetFinder.getInitState();
    if (resultCode != TargetFinder.INIT_SUCCESS)
    {
        if(resultCode ==
TargetFinder.INIT_ERROR_NO_NETWORK_CONNECTION)
        {
            mInitErrorCode = UPDATE_ERROR_NO_NETWORK_CONNECTION;
        }
        else
        {
            mInitErrorCode = UPDATE_ERROR_SERVICE_NOT_AVAILABLE;
        }

        Log.e(LOGTAG, "Failed to initialize target finder.");
        return false;
    }

    mTargetFinder = targetFinder;
}

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		


```

        return true;
    }

    @Override
    public boolean doUnloadTrackersData()
    {
        return true;
    }

    @Override
    public void onVuforiaResumed()
    {
        if (mGlowView != null)
        {
            mGlowView.setVisibility(View.VISIBLE);
            mGlowView.onResume();
        }
    }

    // Called once Vuforia has been initialized or
    // an error has caused Vuforia initialization to stop
    @Override
    public void onInitARDone(SampleApplicationException exception)
    {
        if (exception == null)
        {
            initApplicationAR();

            mRenderer.setActive(true);

            // Now add the GL surface view. It is important
            // that the OpenGL ES surface view gets added
            // BEFORE the camera is started and video
            // background is configured.
            addContentView(mGlowView, new
LayoutParams (LayoutParams.MATCH_PARENT,
                LayoutParams.MATCH_PARENT));

            mUILayout.bringToFront();
            mBtnLayout.setVisibility(View.VISIBLE);

            Button resetBtn = (Button)
mUILayout.findViewById(R.id.reset_btn);

            resetBtn.setOnClickListener(new View.OnClickListener()
            {
                @Override
                public void onClick(View view)
                {
                    mResetTargetFinderTrackables = true;
                }
            });
        }
    }

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        }
    });

    // Hides the Loading Dialog

loadingDialogHandler.sendMessage(LoadingDialogHandler.HIDE_LOADING_DIALOG);

    mUILayout.setBackgroundColor(Color.TRANSPARENT);

    mSettingsAdditionalViews.add(mBtnLayout);
    mSampleAppMenu = new SampleAppMenu(this, this, "Cloud
Reco",
        mGableView, mUILayout, mSettingsAdditionalViews);
    setSampleAppMenuSettings();

    vuforiaAppSession.startAR();

}
else // Show error message if an exception was thrown
during the init process
{
    Log.e(LOGTAG, exception.getString());
    if(mInitErrorCode != 0)
    {
        showErrorMessage(mInitErrorCode,10, true);
    }
    else
    {
showInitializationErrorMessage(exception.getString());
    }
}

}

@Override
public void onVuuforiaStarted()
{
    mRenderer.updateRenderingPrimitives();

    // Set camera focus mode

if(!CameraDevice.getInstance().setFocusMode(CameraDevice.FOCUS_MODE.
FOCUS_MODE_CONTINUOUSAUTO))
{
    // If continuous autofocus mode fails, attempt to set to
a different mode

if(!CameraDevice.getInstance().setFocusMode(CameraDevice.FOCUS_MODE.
FOCUS_MODE_TRIGGERAUTO))
{

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

CameraDevice.getInstance().setFocusMode(CameraDevice.FOCUS_MODE.FOCUS_MODE_NORMAL);
    }

    showProgressIndicator(false);
}

private void showProgressIndicator(boolean show)
{
    if (show)
    {

loadingDialogHandler.sendMessage(LoadingDialogHandler.SHOW_LOADING_DIALOG);
    }
    else
    {

loadingDialogHandler.sendMessage(LoadingDialogHandler.HIDE_LOADING_DIALOG);
    }
}

// Called every frame
@Override
public void onVuforiaUpdate(State state)
{

    TargetFinder finder = mTargetFinder;
    if (finder == null)
    {
        Log.e(LOGTAG, "Tried to query TargetFinder but was not initialized");
        return;
    }

    // Check if there are new results available:
    TargetFinderQueryResult queryResult =
finder.updateQueryResults();
    int queryStatus = queryResult.getStatus();

    // Show a message if we encountered an error:
    if (queryStatus < 0)
    {
        boolean closeAppAfterError = (
            queryStatus ==
UPDATE_ERROR_NO_NETWORK_CONNECTION ||
            queryStatus ==
UPDATE_ERROR_SERVICE_NOT_AVAILABLE);

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        showErrorMessagе(queryStatus,
state.getFrame().getTimestamp(), closeAppAfterError);

    }
    else if (queryStatus ==
TargetFinder.UPDATE_RESULTS_AVAILABLE)
    {
        TargetSearchResultList queryResultsList =
queryResult.getResults();

        // Process new search results
        if (!queryResultsList.empty())
        {
            TargetSearchResult result = queryResultsList.at(0);
            CloudRecoSearchResult cloudRecoResult = null;
            if
(result.isOfType(CloudRecoSearchResult.getClassType()))
            {
                cloudRecoResult = (CloudRecoSearchResult)result;

                // Check if this target is suitable for tracking:
                if (cloudRecoResult != null &&
cloudRecoResult.getTrackingRating() > 0)
                {
                    finder.enableTracking(cloudRecoResult);
                }
            }
        }

        if(mResetTargetFinderTrackables)
        {
            finder.clearTrackables();
            mResetTargetFinderTrackables = false;
        }
    }

@Override
public boolean doInitTrackers()
{
    // Indicate if the trackers were initialized correctly
    boolean result = true;

    TrackerManager tManager = TrackerManager.getInstance();
    Tracker tracker;

    tracker =
tManager.initTracker(ObjectTracker.getClassType());
    if (tracker == null)
    {
        Log.e(LOGTAG,

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        "Tracker not initialized. Tracker already
initialized or the camera is already started");
        result = false;
    }
    else
    {
        Log.i(LOGTAG, "Tracker successfully initialized");
    }

    // Initialize the Positional Device Tracker
    PositionalDeviceTracker deviceTracker =
(PositionalDeviceTracker)
tManager.initTracker(PositionalDeviceTracker.getClassType());

    if (deviceTracker != null)
    {
        Log.i(LOGTAG, "Successfully initialized Device
Tracker");
    }
    else
    {
        Log.e(LOGTAG, "Failed to initialize Device Tracker");
    }

    return result;
}

@Override
public boolean doStartTrackers()
{
    // Indicate if the trackers were started correctly
    boolean result = true;

    if (mTargetFinder == null)
    {
        Log.e(LOGTAG, "Tried to start TargetFinder but was not
initialized");
        return false;
    }

    mTargetFinder.startRecognition();
    scanlineStart();
    mFinderStarted = true;

    // Start the Object Tracker
    // The Object Tracker tracks the target recognized by the
target finder
    Tracker objectTracker =
TrackerManager.getInstance().getTracker(
        ObjectTracker.getClassType());

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        if (objectTracker != null)
        {
            result = objectTracker.start();
        }

        // Start device tracker if enabled
        if (isDeviceTrackerActive())
        {
            TrackerManager tManager = TrackerManager.getInstance();
            PositionalDeviceTracker deviceTracker =
(PositionalDeviceTracker) tManager
.getTracker(PositionalDeviceTracker.getClassType());

            if (deviceTracker != null && deviceTracker.start())
            {
                Log.i(LOGTAG, "Successfully started Device
Tracker");
            }
            else
            {
                Log.e(LOGTAG, "Failed to start Device Tracker");
            }
        }

        return result;
    }

    @Override
    public boolean doStopTrackers()
    {
        // Indicate if the trackers were stopped correctly
        boolean result = true;

        TrackerManager trackerManager =
TrackerManager.getInstance();
        ObjectTracker objectTracker = (ObjectTracker) trackerManager
            .getTracker(ObjectTracker.getClassType());

        if(objectTracker != null)
        {
            objectTracker.stop();

            if (mTargetFinder == null)
            {
                Log.e(LOGTAG, "Tried to stop TargetFinder but was
not initialized");
                return false;
            }

            TargetFinder targetFinder = mTargetFinder;
            targetFinder.stop();
        }
    }

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        scanlineStop();
        mFinderStarted = false;

        // Clears the trackables
        targetFinder.clearTrackables();
    }
    else
    {
        result = false;
    }

    // Stop device tracker
    if(isDeviceTrackerActive())
    {
        Tracker deviceTracker =
trackerManager.getTracker(PositionalDeviceTracker.getClassType());

        if (deviceTracker != null)
        {
            deviceTracker.stop();
            Log.i(LOGTAG, "Successfully stopped device
tracker");
        }
        else
        {
            Log.e(LOGTAG, "Could not stop device tracker");
        }
    }

    return result;
}

@Override
protected void onSaveInstanceState(@NonNull Bundle outState) {
}

@Override
public boolean doDeinitTrackers()
{
    TrackerManager tManager = TrackerManager.getInstance();

    // Indicate if the trackers were deinitialized correctly
    boolean result =
tManager.deinitTracker(ObjectTracker.getClassType());

tManager.deinitTracker(PositionalDeviceTracker.getClassType());

    return result;
}

private boolean isDeviceTrackerActive()
{

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        return mDeviceTracker;
    }

    // Menu options
    private final static int CMD_BACK = -1;
    private final static int CMD_DEVICE_TRACKER = 1;

    private void setSampleAppMenuSettings()
    {
        SampleAppMenuGroup group;

        group = mSampleAppMenu.addGroup("", false);
        group.addTextItem(getString(R.string.menu_back), -1);

        group = mSampleAppMenu.addGroup("", true);
        group.addSelectionItem(getString(R.string.menu_device_tracker),
                                CMD_DEVICE_TRACKER, false);

        mSampleAppMenu.attachMenu();
    }

    // In this function you can define the desired behavior for each
    menu option
    // Each case corresponds to a menu option
    @Override
    public boolean menuProcess(int command)
    {
        boolean result = true;

        switch (command)
        {
            case CMD_BACK:
                finish();
                break;

            case CMD_DEVICE_TRACKER:

                TrackerManager trackerManager =
                TrackerManager.getInstance();
                PositionalDeviceTracker deviceTracker =
                (PositionalDeviceTracker)
                trackerManager.getTracker(PositionalDeviceTracker.getClassType());

                if (deviceTracker != null)
                {
                    if (!mDeviceTracker)
                    {
                        if (!deviceTracker.start())
                        {

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		


```

        Log.e(LOGTAG, "Failed to start device
tracker");
        result = false;
    }
    else
    {
        Log.d(LOGTAG, "Successfully started
device tracker");
    }
}
else
{
    deviceTracker.stop();
    clearSampleAppMessage();
}
}
else
{
    Log.e(LOGTAG, "Device tracker is null!");
    result = false;
}

if (result)
{
    mDeviceTracker = !mDeviceTracker;
}
else
{
    clearSampleAppMessage();
}

break;
}

return result;
}

public void checkForRelocalization(final int statusInfo)
{
    if (mCurrentStatusInfo == statusInfo)
    {
        return;
    }

    mCurrentStatusInfo = statusInfo;

    if (mCurrentStatusInfo ==
TrackableResult.STATUS_INFO.RELOCALIZING)
    {
        // If the status is RELOCALIZING, start the timer
        if (!mStatusDelayTimer.isRunning())
        {

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```

        mStatusDelayTimer.startTimer();
    }
}
else
{
    // If the status is not RELOCALIZING, stop the timers
    and hide the message
    if (mStatusDelayTimer.isRunning())
    {
        mStatusDelayTimer.stopTimer();
    }

    if (mRelocalizationTimer.isRunning())
    {
        mRelocalizationTimer.stopTimer();
    }

    runOnUiThread(new Runnable()
    {
        @Override
        public void run()
        {
            mSampleAppMessage.hide();
        }
    });
}

private void clearSampleAppMessage()
{
    runOnUiThread(new Runnable()
    {
        @Override
        public void run()
        {
            if (mSampleAppMessage != null)
            {
                mSampleAppMessage.hide();
            }
        }
    });
}

private void scanlineStart()
{
    this.runOnUiThread(new Runnable()
    {
        @Override
        public void run()
        {
            scanLine.setVisibility(View.VISIBLE);

```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		

```
        scanLine.setAnimation(scanAnimation);
    }
    });
}

private void scanlineStop()
{
    this.runOnUiThread(new Runnable()
    {
        @Override
        public void run()
        {
            scanLine.setVisibility(View.GONE);
            scanLine.clearAnimation();
        }
    });
}
}
```

					ДР.126.6144м.04.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		