

УДК 004.93

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

Рівень вищої освіти другий (магістерський)

ГЮИК. 508000.006 ПЗ

Дослідження та розробка математичного забезпечення інформаційно-аналітичних систем підбору об'єктів
(тема)

Виконав:

Студент 2 курсу, групи СПРМ-19-2

Спеціальність 122 – Комп'ютерні науки
(код і повна назва напрямку)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування
(повна назва освітньої програми)

Надточій О.Ю.

(прізвище, ініціали)

Керівник проф. Нечипоренко А.С.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Гребеннік І. В.
(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 – Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Надточію Олександрю Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження та розробка математичного забезпечення інформаційно-аналітичних систем підбору об'єктів»

затверджена наказом по університету від «23» 03 2021 р. № 389Ст

2. Термін подання студентом роботи до екзаменаційної комісії 24 травня 2021 р.

3. Вихідні дані до роботи Функції системи: перегляд одягу, відображення інформації про одяг, підбір необхідного розміру одягу на основі даних користувача. Перелік використовуваних програмних засобів: програма для створення 3D графіки Blender, засоби для проектування, СУБД SQLite, програма для роботи з реляційними базами даних SQLite Database Browser, середа розробки Visual Studio, міжплатформена середа розробки Unity.

4. Перелік питань, що потрібно опрацювати в роботі Вступ. Аналіз сучасного стану розглянутої проблеми. Розробка вимог до компонентів інформаційно-аналітичної системи підбору одягу. Математичне моделювання як основа побудови тривимірних моделей. Реалізація програмного забезпечення компонентів інформаційно-аналітичної системи підбору одягу. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів, комп'ютерних ілюстрацій) Плакати на аркушах формату А4: контекстна діаграма потоків даних, декомпозиція діаграми потоків даних, логічна діаграма моделі даних, фізична діаграма моделі даних

6. Консультанти розділів роботи (проекту)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	<i>Отримання та аналіз завдання, уточнення плану роботи</i>	<i>23.03.2021</i>	
2	<i>Аналіз існуючих аналогів</i>	<i>29.03.2021</i>	
3	<i>Постановка задачі та вибір методу її вирішення</i>	<i>07.04.2021</i>	
4	<i>Розробка програмного засобу</i>	<i>22.04.2021</i>	
5	<i>Оформлення пояснювальної записки</i>	<i>28.04.2021</i>	
6	<i>Підготовка презентації</i>	<i>20.05.2021</i>	
7	<i>Подання закінченої роботи науковому керівникові</i>	<i>20.05.2021</i>	
8	<i>Подання роботи на рецензування</i>	<i>22.05.2021</i>	
9	<i>Попередній захист</i>	<i>22.05.2021</i>	
10	<i>Подання роботи до комісії</i>	<i>23.05.2021</i>	

Дата видачі завдання 23 березня 2021 р.

Студент

(підпис)

Надточій О.Ю.

Керівник роботи

(підпис)

проф. Нечипоренко А.С.

(посада, прізвище, ініціали)

РЕФЕРАТ

Атестаційна робота: 72 с., 52 рис., 2 табл., 3 додатку, 13 джерел. Графічний матеріал атестаційної роботи містить 4 плаката.

ДОПОВНЕНА РЕАЛЬНІСТЬ, ЛОГІЧНА МОДЕЛЬ, БАЗА ДАНИХ, МОБІЛЬНИЙ ДОДАТОК, ОДЯГ, VUFORIA, ANDROID, РОЗРОБКА, BLENDER, UNITY, МОДЕЛЬ, ВІРТУАЛЬНА СЦЕНА.

Об'єктом дослідження є процес розробки компонентів мобільного додатку для підбору одягу, із використанням технології доповненої реальності.

Предметом дослідження є методи та інструменти для розробки компонентів мобільного додатку для підбору одягу.

Метою атестаційної роботи є дослідження та розробка математичного забезпечення інформаційно-аналітичних систем підбору об'єктів та програмна реалізація компонентів системи.

Продукт, що створюється являє собою додаток на платформу Android, який призначено для компаній, що спеціалізуються на продажі одягу, з метою просування своєї продукції і залучення клієнтів.

Методи дослідження - методи структурного і функціонального моделювання програмних систем з інтерфейсом доступу до баз даних, включаючи платформу доповненої реальності Vuforia, методи структурного моделювання реляційних баз даних і програмування на мові C# (платформа .Net) з використанням бібліотеки sqlite.dll для підключення до бази даних на SQLite із проекту Unity.

В атестаційній роботі наводиться огляд платформи доповненої реальності Vuforia, а також розглядається процес інтеграції Vuforia з кросплатформним движком Unity та залученням бази даних.

В результаті виконання атестаційної роботи було розроблено додаток, який підтримується мобільними пристроями під керуванням ОС Android призначений для компаній, що спеціалізуються на продажі одягу.

ABSTRACT

Master thesis: 72 p., 52 pic., 2 table, 13 source, 3 applications. Graphic material attestation work contains 4 poster.

AUGMENTED REALITY, LOGICAL MODEL, DATABASE, MOBILE APP, CLOTHES, VUFORIA, ANDROID, DEVELOPMENT, BLENDER, UNITY, MODEL, VIRTUAL SCENE.

The object of research is the process of developing components of a mobile application for the selection of clothes, using the technology of augmented reality.

The subject of research is the methods and tools for developing components of a mobile application for the selection of clothes.

The purpose of master thesis is the development of components of a mobile application for the selection of clothes. The product being created is an application for the Android mobile platform, which is intended for companies that specialize in the sale of clothing, in order to promote their products and attract customers.

Research methods - methods of structural and functional modeling of software systems with an interface to access databases, including the Vuforia platform of augmented reality, methods of structural modeling of relational databases and programming in C# (platform .Net) with using the sqlite.dll library for database connection on SQLite from the Unity project.

The master thesis provides an overview of the Vuforia platform, as well as the process of integrating Vuforia with the Unity engine and involving the database.

The results of master thesis has developed a mobile application that is supported by mobile devices running Android OS designed for companies specializing in the sale of clothing.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
1 АНАЛІЗ СУЧАСНОГО СТАНУ РОЗГЛЯНУТОЇ ПРОБЛЕМИ	10
1.1 Опис предметної області дослідження.....	10
1.2 Огляд існуючих аналогів додатку	11
1.3 Постановка задачі	19
2 РОЗРОБКА ВИМОГ ДО КОМПОНЕНТІВ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ПІДБОРУ ОДЯГУ.....	20
2.1 Розробка моделі потоків даних	20
2.2 Опис основних функцій у додатку.....	22
2.3 Реалізація бази даних	23
3 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ЯК ОСНОВА ПОБУДОВИ ТРИВИМІРНИХ МОДЕЛЕЙ	35
3.1 Існуючі підходи та методи створювання 3D моделей.....	35
3.2 Розробка тривимірних моделей	37
4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМПОНЕНТІВ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ПІДБОРУ ОДЯГУ	43
4.1 Реалізація додатку	43
4.2 Доповнена реальність та структура платформи Vuforia	45
4.3 Реалізація інтерфейсу користувача	51
4.4 Реалізація логіки додатку	52
ВИСНОВКИ.....	69
ПЕРЕЛІК ПОСИЛАНЬ	71
Додаток А Графічні матеріали	73
Додаток Б Перевірка атестаційної роботи на плагіат	80
Додаток В Відомість дипломної роботи	83

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних;

ОС – операційна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

СУБД – система управління базами даних;

API – application programming interface, інтерфейс програмування додатків;

AR – augmented reality, доповнена реальність;

GUI – graphical user interface, графічний користувальницький інтерфейс;

IDE – integrated development environment, інтегроване середовище розробки;

SDK – software development kit, комплект засобів розробки;

SQL – structured query language, мова структурованих запитів.

ВСТУП

Існуючий рівень розвитку інформаційних систем і мобільних додатків дозволяє споживачеві купувати одяг в будь-якій точці світу. Зростання динаміки покупок одягу через інтернет багато в чому пояснюється вдосконаленням технологій онлайн-продажів і їх доступністю завдяки розвитку різноманітних додатків до мобільних пристроїв, таким як смартфони, планшети і т.д.

У даній магістерській атестаційній роботі представлено дослідження та розробку математичного забезпечення інформаційно-аналітичних систем підбору об'єктів, а саме одягу. Це додаток для мобільних пристроїв для пошуку необхідного одягу та підбору образу із доступних позицій для платформи Android.

Завдяки використанню камери пристрою та технології доповненої реальності можна побачити, як на людині певного розміру буде виглядати одяг.

Для перегляду одягу користувач повинен ввести параметри тіла: свій зріст, обхват талії та обхват грудей. Після підтвердження введених даних буде відображений манекен відповідного розміру. Програма самостійно визначить, які розміри одягу підходять користувачеві.

При дослідженні та розробці математичного забезпечення інформаційно-аналітичної системи підбору об'єктів необхідно вирішити основні завдання, а саме:

- дослідження існуючих аналогів для підбору об'єктів (одягу);
- розробку тривимірної моделі манекену для відображення одягу;
- розробку логіки для здійснення підбору одягу на моделі манекена для створення унікального образу;
- розробку бази даних (DB частини), для зберігання та отримання інформації;
- розробку алгоритму для визначення розмірів одягу та зміни статури манекену на основі введених параметрів;
- розробку інтерфейсу користувача (UI частини).

Об'єктом дослідження є процес підбору одягу, із використанням технології доповненої реальності.

Предметом дослідження є методи розробки компонентів мобільного додатку для підбору одягу.

Науковою новизною є модифікація методу підбора одягу.

Практичне значення роботи полягає у можливості підбору одягу користувачами додатку за різними категоріями.

1 АНАЛІЗ СУЧАСНОГО СТАНУ РОЗГЛЯНУТОЇ ПРОБЛЕМИ

1.1 Опис предметної області дослідження

Об'єктом дослідження даної роботи є мобільний додаток с технологією доповненої реальності під назвою «ClothesAR» на базі кросплатформеного движка Unity. Він створений для того, щоб користувачі могли здійснювати пошук необхідного одягу [13].

Головною особливістю цього додатку є те, що за допомогою свого пристрою можна не тільки дізнатися інформацію про одяг, а також і роздивитися його з усіх сторін. Це відтворює відчуття перебування, як у магазині.

Ще однією важливою особливістю є те, що програма допомагає підібрати необхідний розмір одягу, якщо користувач вірно вкаже свої розміри тіла. На їх основі буде пропонуватися потрібний розмір, а при відображенні додаткової інформації про обраний одяг програма сама підкаже користувачеві, чи є його розмір у наявності.

Також є реалістична модель манекену, яка підлаштовується під введені дані зросту та обхвату грудей та талії.

Інтерфейс було розроблено спеціально для користувачів смартфонів. Користувачі можуть у будь який час скористатися додатком, якщо вони мають мітку (target точку) на якій буде будуватися зображення моделі манекену.

Цільовою аудиторією є чоловіки, тому що одяг, який розміщений у цьому додатку, орієнтований на чоловічу половину населення. Це люди, які зацікавлені у покупці одягу і являються активними користувачами електронних пристроїв.

На ринку є не багато схожих на цей продукт додатків. Дана програма за своїм функціоналом має суттєві відмінності від інших продуктів, так як вона включає в себе підбір оптимального розміру одягу та приблизне відображення одягу на манекені на основі заданих розмірів тіла.

1.2 Огляд існуючих аналогів додатку

Сьогодні, щоб бути в тренді, необов'язково підписуватися на десятки модних журналів. Досить встановити на смартфон спеціальні програми та отримати доступ до статей з корисними порадами та інтерв'ю з провідними дизайнерами, каталогам стильних луків, різним інтернет-магазинах.

Нижче приведені додатки для підбору одягу, його продажу та додатки, які пов'язані з fashion-індустрією.

Stylebook. Додаток дозволяє управляти своїм гардеробом, створюючи креативні образи з того, що є в наявності. Завантажуйте фотографії свого одягу і аксесуарів або аналогічних моделей, щоб моделювати комплекти на кожен день: для офісу, подорожей або вечірок. Розділ "Натхнення" і "Магазин" підходить для того, щоб планувати нові покупки.



Рисунок 1.1 – Stylebook

The Cut. Додаток журналу NY Mag має повноцінну гаджет-версію. За допомогою The Cut можна переглядати в деталях фото і відео зі світових подій вже через пару годин після показу, а також читати свіжі новини fashion-сфери.

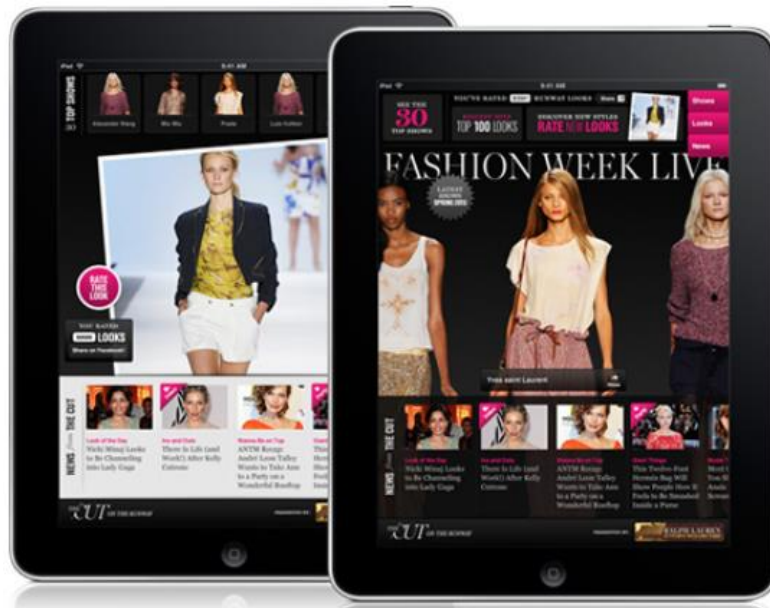


Рисунок 1.2 – The Cut

Net-a-Porter. Ніяких кордонів для шопінгу, впевнені в net-a-porter.com. За допомогою додатку від ведучого онлайн-рітейлера можна замовляти вбрання з будь-якої точки світу і при будь-яких обставинах, а також відправляти вподобані речі в wish-list і надихатися підібраними стилістами образами на основі вибраних речей.

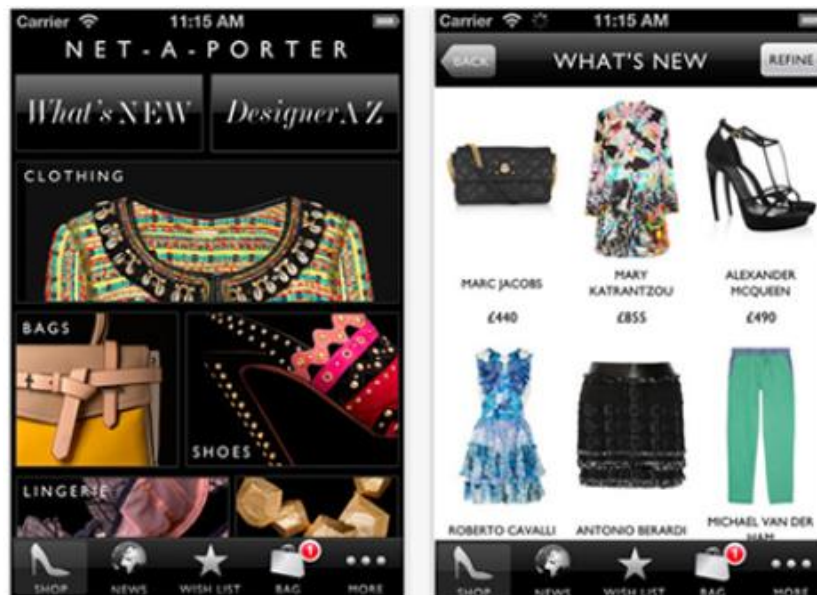


Рисунок 1.3 – Net-a-Porter

Street Snap. За перегортанням додатка Street Snap можна непомітно провести пару годин в залі очікування або в дорозі. У ньому оперативно з'являються street-

fashion фотографії з усього світу, включаючи знаменитостей рівня Алекси Чанг і Кейт Босуорт.

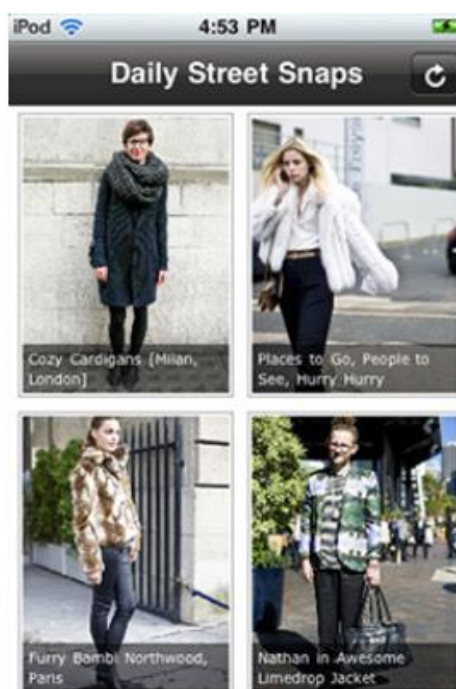


Рисунок 1.4 – Street Snap

What's My Size. Ця програма особливо зручно для онлайн-шопінгу: за допомогою What's My Size можна з'ясувати, який розмір в сітці бренду відповідає вашому. Найпростіше зробити добірку улюблених марок (в додатку доступно більше 500), і мати під рукою таблицю під час замовлення наряду в інтернет-магазині.

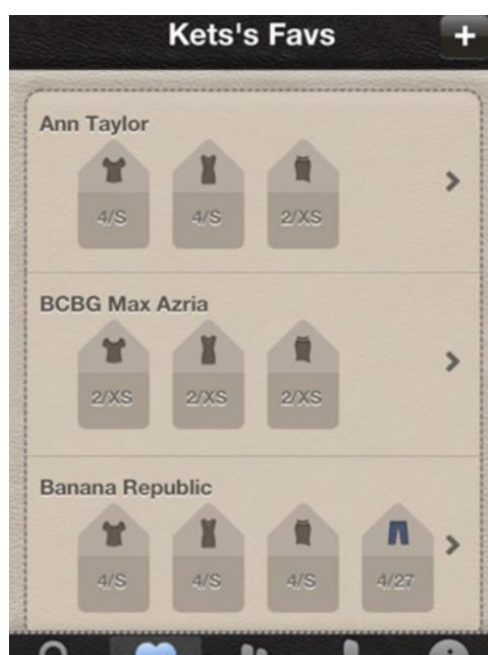


Рисунок 1.5 – What's My Size

Гардероб Getwardrobe. Це сервіс для систематизації гардероба, пошуку поєднань і планування. Він визначить, наскільки часто ви носите речі і сети, допоможе зібрати в дорогу сумку і підкаже, скільки вона буде важити. В останньому оновленні покращилася робота ІІ-асистента: при автоматичному підборі лука додаток знаходить більш вдалі комбінації.



Рисунок 1.6 – Гардероб Getwardrobe

ShopLook Outfit Maker. Це зручний менеджер гардероба з категоріями, сортуванням за кольорами і можливістю збереження своїх добірок у вигляді колажу.

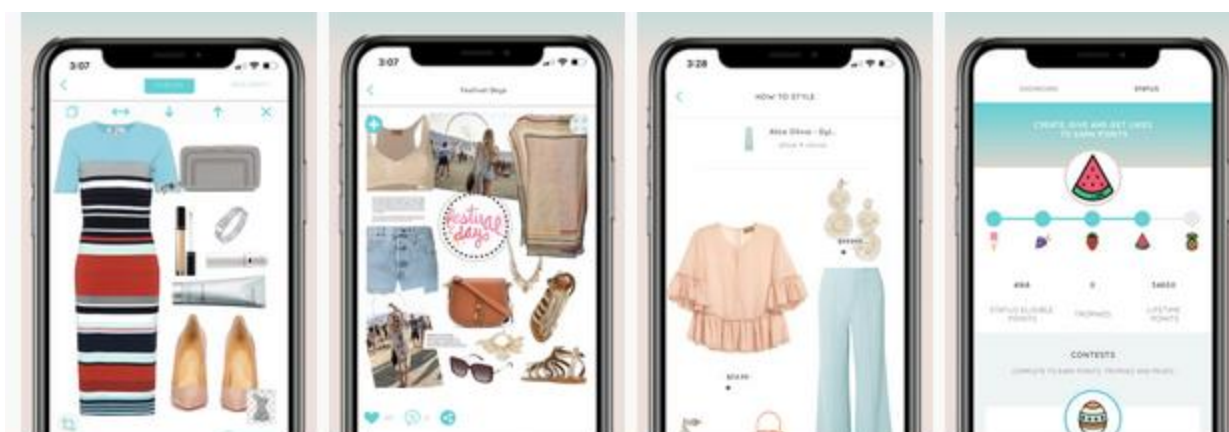


Рисунок 1.7 – ShopLook Outfit Maker

Outfit Mind розумний підбір одягу. Outfit Mind допоможе впорядкувати ваш гардероб і знайти в ньому вдалі комбінації. Щоб користуватися програмою необхідно

завантажити в нього фотографії свого одягу і розподілити за категоріями. Після цього ви зможете скористатися автоматичним режимом «Одягни мене» або самостійно підібрати цибулю. Красиві комбінації можна зберегти в галереї. Додаток підкаже, які аксесуари допоможуть освіжити образ.

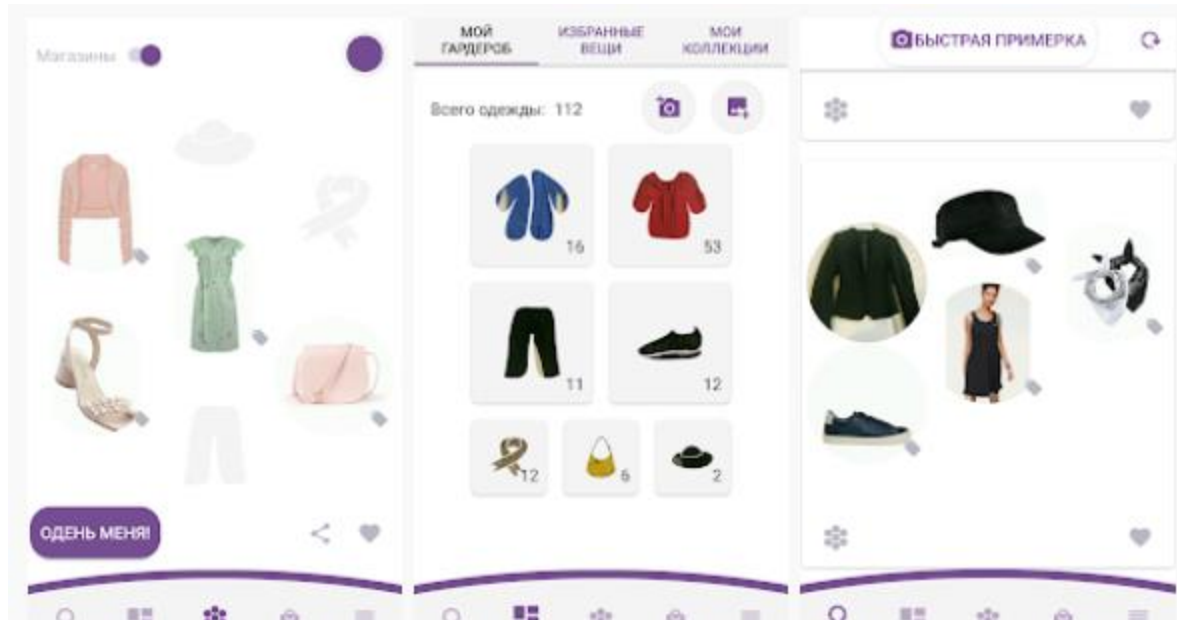


Рисунок 1.8 – Outfit Mind розумний підбір одягу

Your Closet Smart Fashion. Ця програма являє собою персонального кишенькового стиліста, який допоможе підібрати ідеальний образ для будь-якого випадку. Все, що потрібно зробити користувачеві - це сфотографувати предмети особистого гардеробу і завантажити знімки в додаток. На їх основі програма складе оптимальні варіанти для різних приводів.

При цьому зовсім не обов'язково сліпо слідувати порадам мобільного стиліста - можна самостійно складати цікаві комплекти з власного одягу і аксесуарів.

Ще одна корисна функція Your Closet - можливість подивитися, наскільки виграно виглядатиме певна річ з магазину. При виборі одягу під час шопінгу ця опція дозволить не тільки зробити правильний вибір, а й істотно заощадити час.

Інші корисні функції сервісу:

- швидка і зручна організація гардеробу;
- підбір кольору;
- планування образів з календарем;

- пошук речі або її альтернативи в інтернеті.

Завдяки функції синхронізації можна не турбуватися, що створені в Your Closet образи будуть втрачені.

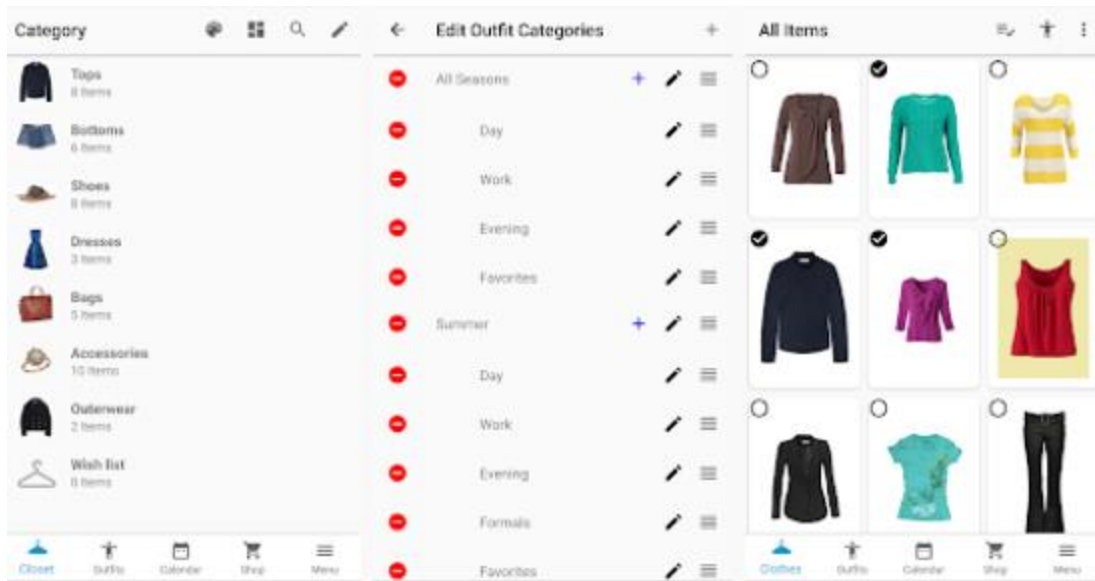


Рисунок 1.9 – Your Closet Smart Fashion

Lamoda інтернет магазин одягу. Додаток від великого інтернет-магазину, що продає одяг, взуття та аксесуари на території Росії, України, Казахстану і Білорусії. Користувачам доступний великий вибір модних речей від популярних брендів, сортування позицій по різних фільтрів, зручний інструментарій для оформлення та оплати замовлень, а також регулярні знижки та спеціальні пропозиції.

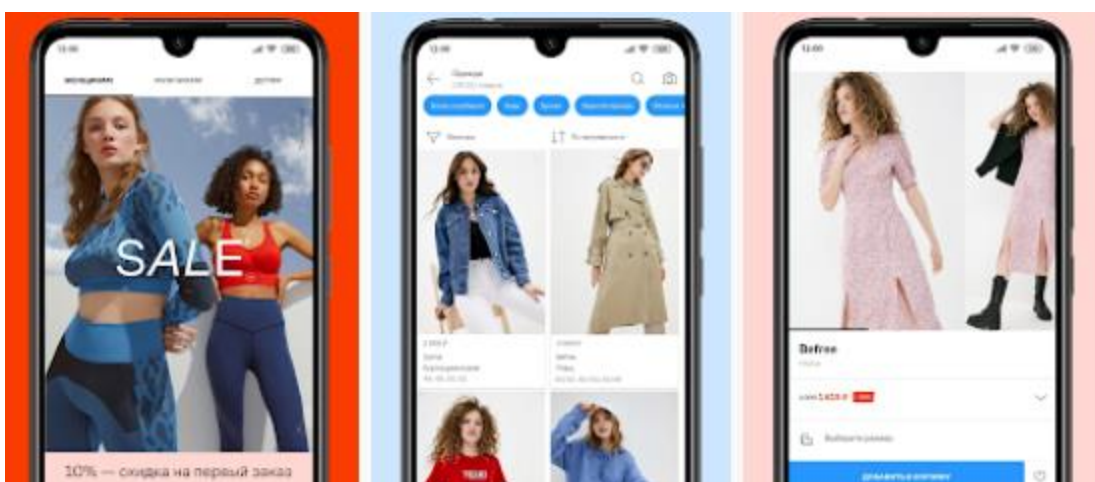


Рисунок 1.10 – Lamoda

The Iconic. Цей сервіс підходить для тих, хто любить відстежувати модні новинки в магазинах, але не хоче витратити на це занадто багато часу. Додаток The Iconic моніторить надходження з нових колекцій найбільших брендів і попереджає власницю смартфона про те, коли одяг надійде в продаж. Більше не доведеться постійно стежити за соцмережами брендів, щоб не пропустити вихід образів з останніх показів Тижнів моди. Додаток адаптовано під іноземного користувача.

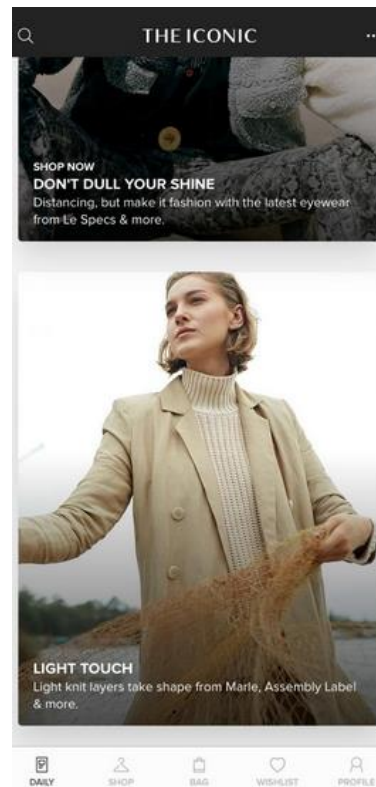


Рисунок 1.11 – The Iconic

Combyne - your perfect outfit. Додаток для моди Combyne з легкістю замінить послуги стилістів по підборі одягу, адже ви зможете не тільки істотно заощадити час при самостійному створенні різних стильних нарядів, але і зберігши їх, коментувати колажі модних «луків», а також оцінювати, ділитися в соціальних мережах з найбільш яскравими образами. Співпраця з 30-ю партнерськими інтернет-магазинами дозволить придбати вподобані предмети одягу та аксесуари від відомих світових брендів. У вашій особистій віртуальній вбиральні розмістіть свої речі в майже 70 категоріях і починайте комбінувати. На сьогоднішній день програма Combyne доступна тільки для створення жіночих образів. У додатку можна зберігати

зображення речей, які хочеться купити і тих предметів одягу, які вже є у вашій шафі, це дозволить продумати можливі комбінації і уникнути непотрібних витрат. Користувачам пропонуються вигідні акції, знижки, різні статті на теми краси і моди, поради щодо підбору одягу від стилістів.

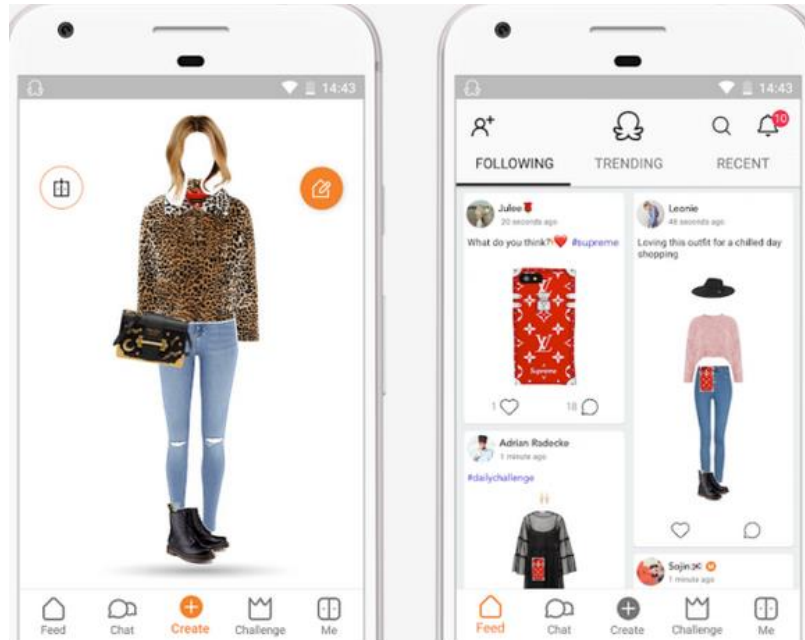


Рисунок 1.12 – Combyne - your perfect outfit

Мобільний додаток «Що вдягнути?» відмінний сервіс для грамотного підбору одягу відповідно до власного кольоротипу. Завантажте свою фотографію без макіяжу і цифровий стиліст зробить аналіз вашої зовнішності, ґрунтуючись на кольорі волосся, шкіри, очей і губ. І таким чином, ви зможете правильно підібрати найбільш підходящу вам колірну гамму, визначити свій базовий цветотип, отримати рекомендації щодо вибору одягу і аксесуарів. Сумніваєтеся або не знаєте, що сьогодні одягнути? Завантажте фотографії двох предметів одягу та натисніть «Порівняти» - програма автоматично визначить кращий для вас варіант. Для визначення, який відтінок речі підійде до вашої особи, використовуйте колірне колесо, кольору всередині якого підбираються відповідно до вашою фотографією. Вирушили на шопінг? Віртуальний помічник завжди поруч. Приміряйте сподобалися комплекти, робіть селфі, завантажуйте знімки, помічник порівняє всі комбінації і вибере для вас краще. Скористайтеся продуманими онлайн-рішеннями і ви завжди будете в тренді і «на стилі».

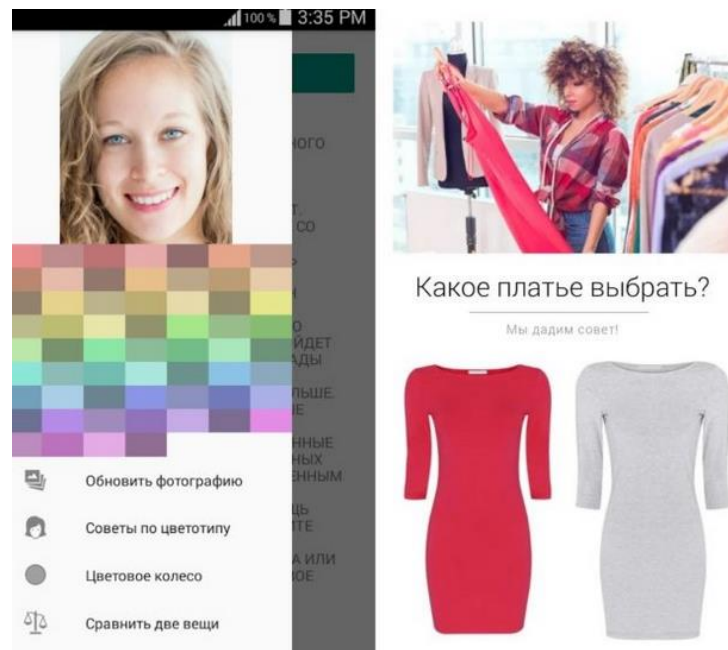


Рисунок 1.13 – Що вдягнути?

1.3 Постановка задачі

Виходячи з аналізу предметної області, завданням даної атестаційної роботи магістра є дослідження та розробка математичного забезпечення інформаційно-аналітичної системи підбору об'єктів, а саме компонентів мобільного додатку для підбору одягу. Особливістю даної системи є те, що вона призначена для цільової категорії користувачів додатків вибору підбору одягу.

Вимоги до компонентів системи повинні містити у собі сукупність технологічних та функціональних моделей, які у свою чергу реалізують такі функції:

- можливість користувачем мобільного додатку обирати потрібний йому одяг враховуючи розмір, колір та його уподобання;
- можливість приміряти обраний одяг за допомогою тривимірної моделі манекену;
- можливість інтеграції розробленого мобільного додатку у існуючі системи, які спеціалізуються на продажі одягу або fashion індустрії.

2 РОЗРОБКА ВИМОГ ДО КОМПОНЕНТІВ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ПІДБОРУ ОДЯГУ

2.1 Розробка моделі потоків даних

Діаграми DFD (Data Flow Diagramming) можна використовувати для опису документообігу та обробки інформації, оскільки вони докладно описують потоки даних, дозволяють відстежити, як відбувається обмін інформацією у системі між бізнес-процесами та між самою системою з зовнішнім середовищем. Сфера застосування DFD-діаграм стосується області моделювання інформаційних потоків організації. У цій нотації моделюється не послідовність робіт, а саме потоки інформації (даних) між роботами і об'єктами.

У основі даної методології полягає побудова моделі інформаційної системи (ІС), яка проектується або вже існує. Відповідно до методології модель системи визначається як ієрархія діаграм потоків даних, що описують асинхронний процес перетворення інформації від її вводу до системи до видачі користувачеві. Діаграми верхніх рівнів ієрархії (контекстні діаграми) визначають основні процеси інформаційної системи з зовнішніми входами і виходами. Вони деталізуються за допомогою діаграм нижнього рівня. Така декомпозиція продовжується, створюючи багаторівневу ієрархію діаграм, поки не буде досягнутий рівень декомпозиції, на якому процеси стають елементарними і декомпозувати далі їх неможливо.

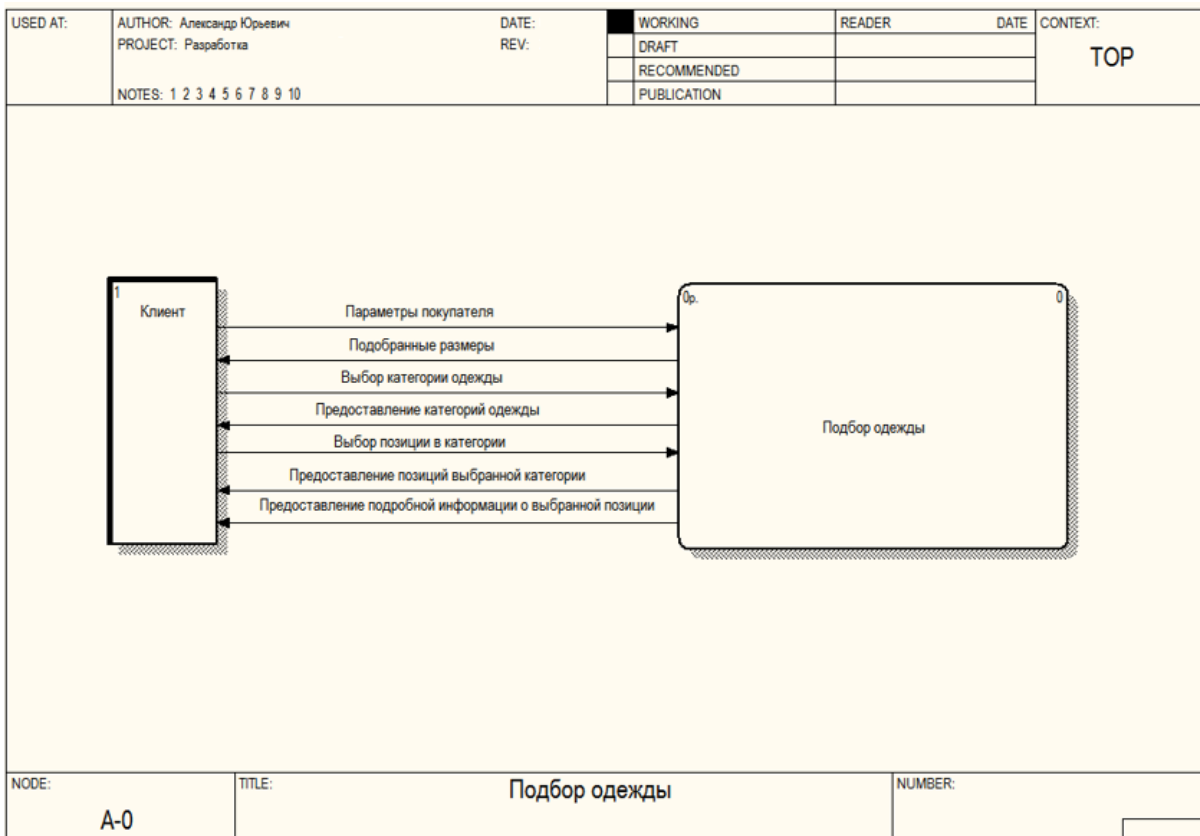


Рисунок 2.1 – Контекстна діаграма потоків даних

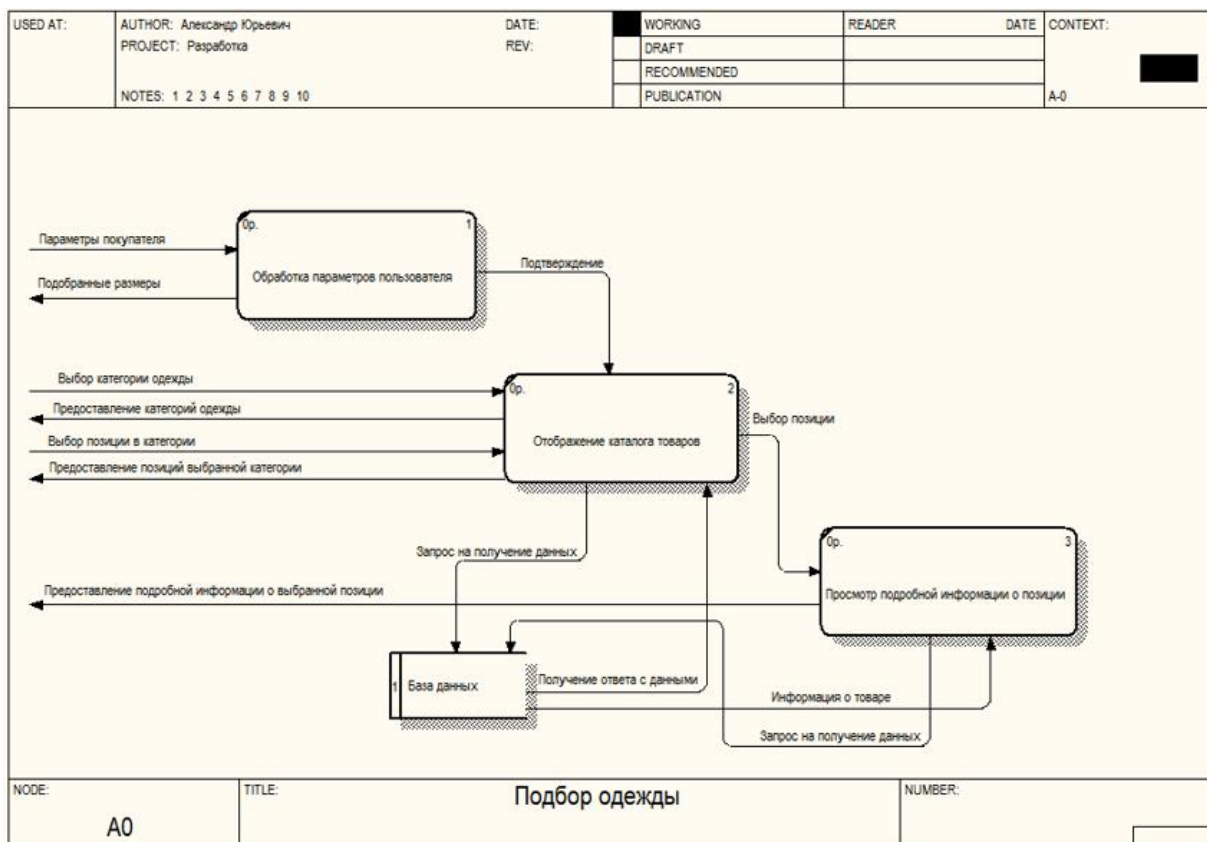


Рисунок 2.2 – Діаграма потоків даних задачі «Підбір одягу»

2.2 Опис основних функцій у додатку

Для того щоб визначити варіанти використання (так звані прецеденти) системи, а також її користувачів (ініціаторів прецедентів) будується діаграма варіантів використання, яка є узагальненою моделлю функціонування ПЗ в навколишньому середовищі. На рис. 2.3 відображена діаграма варіантів використання (Use Case diagram), на якій описані прецеденти і актори, а також відносини між ними.



Рисунок 2.3 – Use Case (діаграма прецедентів)

У системі буде один тип користувачів – клієнт.

У таблиці 2.1 перелічені варіанти використання для цього типу користувача.

Таблиця 2.1 – Короткий опис варіантів використання

Варіант використання	Опис варіанту використання
Підбір образу	Функція для автоматичного підбора одягу за сезонами року

Вибір категорії одягу	Користувач має можливість перемикає категорії за допомогою відповідних кнопок для зміни сорочки на футболку або светр чи джинсів на шорти
Вибір одягу	У обраній категорії можна обирати доступний одяг
Перегляд одягу	Обраний одяг одразу можна переглянути (відображається на манекені)
Отримання інформації про одяг	Кожна позиція одягу має загальну інформацію, таку як: назва, бренд. Також є додаткова інформація: ціна, наявність.
Встановлення та зміна значень параметрів для визначення розмірів одягу	Щоб програма мала змогу самостійно підібрати необхідні розміри одягу треба знати та вказати значення зросту, обхвату талії та грудей. У будь-який момент часу можна змінити вказані початкові параметри.
Перегляд підібраних розмірів одягу	Для того, щоб побачити, які розміри були обрані додатком, треба звернутися до меню або обрати перегляд повної інформації про обраний одяг
Обертання манекену	Ця функція необхідна для того, щоб можна було роздивитися одяг зі всіх сторін

2.3 Реалізація бази даних

Далі необхідно спроектувати базу даних проекту, так як це являється основою для будь-якої системи.

Саме база даних дозволяє експлуатувати інформаційні системи, виконувати їх обслуговування, модифікувати і обробляти інформацію [3].

На першому кроці була створена логічна та фізична діаграми моделі даних у ПЗ ERwin Data Modeler.

ER означає entity relationship, тобто сутність-зв'язок.

ERwin має два рівня уявлення моделі - логічний і фізичний.

Логічний рівень - це абстрактний погляд на дані, на ньому дані представляються так, як виглядають в реальному світі, і можуть називатися так, як вони називаються в реальному світі, наприклад, "Постійний клієнт", "Відділ" або "Прізвище співробітника". Об'єкти моделі, що представляються на логічному рівні, називаються сутностями і атрибутами.

Логічна модель даних являється універсальною і ніяк не пов'язана з конкретною реалізацією СУБД.

Фізична модель даних, залежить від конкретної СУБД, фактично будучи відображенням системного каталогу. У фізичній моделі міститься інформація про всі об'єкти БД. Оскільки стандартів на об'єкти БД не існує, фізична модель залежить від конкретної реалізації СУБД. Отже, однією і тією ж логічної моделі можуть відповідати кілька різних фізичних моделей. Якщо в логічній моделі не має значення, який саме тип даних має атрибут, то у фізичній моделі важливо описати всю інформацію про конкретних фізичних об'єктах - таблицях, колонках і т. п [5].

На рис. 2.4 та 2.5 відображено логічну та фізичну діаграму відповідно.

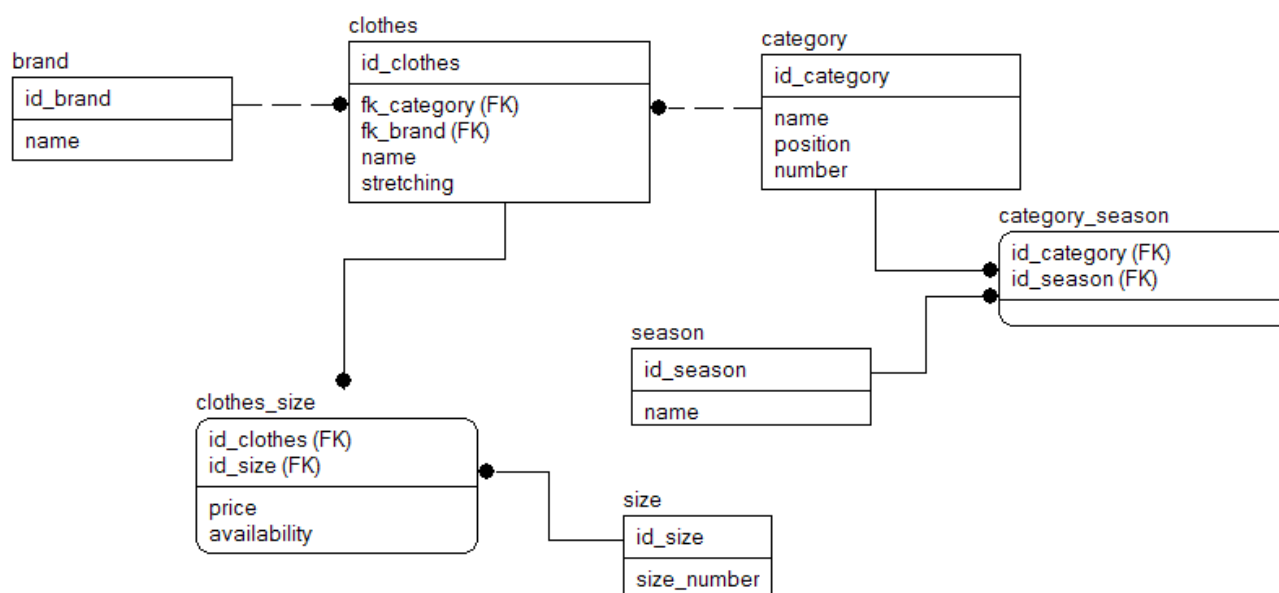


Рисунок 2.4 – Логічна діаграма моделі даних

Сутності предметної області, які увійшли до логічної моделі представлені в табл.2.2.

Таблиця 2.2 – Сутності логічної моделі даних системи

№	Найменування сутності	Найменування атрибута	Тип даних (домен)	Призначення
1.	brand	id_brand	Лічильник, довге ціле	Первинний ключ
		name	Текстовий рядок	Назви брендів
2.	category	id_category	Лічильник, довге ціле	Первинний ключ
		name	Текстовий рядок	Назви категорій одягу
		position	Текстовий рядок	Розташування одягу на манекені
		number	Довге ціле	Номера категорій
3.	size	id_size	Лічильник, довге ціле	Первинний ключ
		size_number	Текстовий рядок	Розміри одягу
4.	clothes	fk_brand	Довге ціле	Зовнішній ключ
		fk_category	Довге ціле	Зовнішній ключ
		name	Текстовий рядок	Назви одягу
		stretching	Довге ціле	Розтягування одягу
5.	clothes_size	id_clothes	Довге ціле	Зовнішній ключ
		id_size	Довге ціле	Зовнішній ключ
		price	Довге ціле	Ціна
		availability	Довге ціле	Доступність
6.	season	id_season	Лічильник, довге ціле	Первинний ключ
		name	Текстовий рядок	Назви сезонів
7.	category_season	id_category	Довге ціле	Зовнішній ключ
		id_season	Довге ціле	Зовнішній ключ

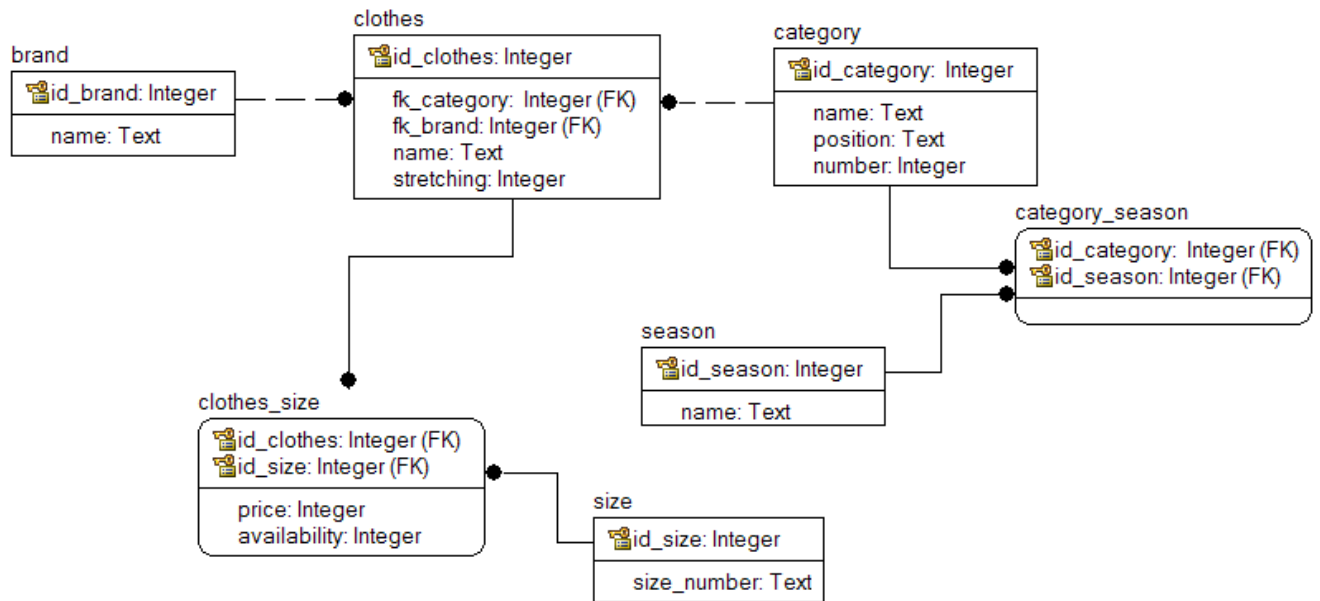


Рисунок 2.5 – Фізична діаграма моделі даних

Оскільки додаток орієнтований під мобільні платформи фізична модель даних була пов'язана з реалізацією на СУБД SQLite.

SQLite - вільно розповсюджувана вбудована система управління базами даних. Є найпопулярнішою і широко використовуваною кросплатформеною СУБД, вбудованої практично в усі мобільні пристрої і більшість комп'ютерів, поставляється всередині безлічі різних додатків.

SQLite є компактною бібліотекою, написану на мові програмування C і реалізовує надійний двигун реляційних баз даних SQL. При всіх включених функціях розмір бібліотеки може бути менше 600 КБ. Важливою відмінністю від інших рішень є її автономність: для роботи SQLite не потрібна наявність сторонніх бібліотек або служб.

СУБД легко інтегрується з різним програмним забезпеченням, підтримує достатній набір команд SQL і ACID транзакції. На відміну від клієнт-серверних баз даних, SQLite призначена для локального зберігання даних окремих додатків і пристроїв.

SQLite - однофайлова СУБД. Вся база, включаючи визначення, таблиці, індекси і дані, зберігається в одному файлі на комп'ютері або пристрої, де виконується програма. Формат файлу бази даних є кросплатформним, тобто його

можна копіювати між різними архітектурами, 32-розрядними і 64-розрядними системами. З SQLite, пов'язані наступні файлові формати: SQLITE, SQLITE3, DB-SHM, DB-WAL, DB, DB3, MDDATA.

СУБД безкоштовна для використання в будь-яких цілях, комерційних або приватних. Вихідний код SQLite знаходиться у вільному доступі і може компілюватися розробниками для різних додатків і цільових платформ.

База даних SQLite не вимагає адміністрування та підходить для роботи на мобільних пристроях, телевізорах, ігрових приставках, камерах, безпілотних літальних апаратах, автомобільних мультимедійних системах і т.д. На сьогоднішній день це найпопулярніша СУБД, яка використовується на кожному пристрої Android і iOS, кожному комп'ютері Mac і Windows 10, а також безліччю програм: Firefox, Chrome, Skype, XnView, Safari, Dropbox, AIMP, Viber і іншими.

Опис бази даних:

Сутність «category» зберігає доступні категорії одягу.

Містить поля:

- id_category (первинний ключ) – ідентифікатор категорії;
- name – назва категорії;
- position – розташування одягу на частині тіла (верх / низ);
- number – особистий номер категорії.

Сутність «brand» зберігає назви брендів одягу.

Містить поля:

- id_brand (первинний ключ) – ідентифікатор бренду;
- name – назва бренду.

Сутність «size» - таблиця, призначена для зберігання розмірів одягу.

Містить поля:

- id_size (первинний ключ) – ідентифікатор розміру одягу;
- size_number – розмір.

Сутність «clothes» являє собою таблицю, яка містить в собі інформацію про користувачів, які зареєстровані на ресурсі.

Містить поля:

- id_clothes (первинний ключ) – ідентифікатор одягу;
- fk_category (зовнішній ключ) – ідентифікатор категорії;
- fk_brand (зовнішній ключ) – ідентифікатор бренду;
- name – назва одягу;
- stretching – можливість одягу тягнутися.

Сутність «clothes_size» - таблиця, призначена для зберігання інформації про розмір одягу, його ціну та наявність.

Містить поля:

- id_clothes (зовнішній ключ) – ідентифікатор одягу;
- id_size (зовнішній ключ) – ідентифікатор розміру;
- price – ціна одягу;
- availability – наявність одягу.

Сутність «season» зберігає назви сезонів.

Містить поля:

- id_season (первинний ключ) – ідентифікатор сезону;
- name – назва сезону.

Сутність «category_season» зберігає інформацію про відношення категорій одягу до сезонів року.

Містить поля:

- id_category (зовнішній ключ) – ідентифікатор категорії;
- id_season (зовнішній ключ) – ідентифікатор сезону.

Для створення у базі даних таблиць використовується скрипт, який представлений у лістингу 2.1.

Лістинг 2.1 – Запити для створення таблиць у БД

```
CREATE TABLE `category` (
    `id_category` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    `name` TEXT NOT NULL,
    `position` TEXT NOT NULL,
    `number` INTEGER NOT NULL
```

```

);
CREATE TABLE `brand` (
    `id_brand` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    `name` TEXT NOT NULL
);
CREATE TABLE `clothes` (
    `id_clothes` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    `fk_category` INTEGER NOT NULL,
    `fk_brand` INTEGER NOT NULL,
    `name` TEXT NOT NULL,
    `stretching` INTEGER NOT NULL,
    FOREIGN KEY(`fk_brand`) REFERENCES `brand`(`id_brand`),
    FOREIGN KEY(`fk_category`) REFERENCES `category`(`id_category`)
);
CREATE TABLE `size` (
    `id_size` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    `size_number` TEXT NOT NULL
);
CREATE TABLE `clothes_size` (
    `id_clothes` INTEGER NOT NULL,
    `id_size` INTEGER NOT NULL,
    `price` INTEGER NOT NULL,
    `availability` INTEGER NOT NULL,
    PRIMARY KEY(`id_clothes`,`id_size`),
    FOREIGN KEY(`id_clothes`) REFERENCES `clothes`(`id_clothes`),
    FOREIGN KEY(`id_size`) REFERENCES `size`(`id_size`)
);
CREATE TABLE `season` (
    `id_season` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    `name` TEXT NOT NULL
);
CREATE TABLE `category_season` (
    `id_category` INTEGER NOT NULL,
    `id_season` INTEGER NOT NULL,
    PRIMARY KEY(`id_category`,`id_season`),

```

```

FOREIGN KEY(`id_category`) REFERENCES `category`(`id_category`),
FOREIGN KEY(`id_season`) REFERENCES `season`(`id_season`)
);

```

Таким чином, база даних була реалізована у необхідній СУБД, далі - заповнена даними за допомогою SQL запитів (які представлені у лістингу 2.2) у програмі для роботи з реляційними базами даних SQLite Database Browser (рис. 2.6).

Лістинг 2.2 – Запити для заповнення таблиць у БД

```

INSERT INTO clothingshop.category (`name`, `position`, `number`)
VALUES ('Pullover', `up`, `1`), ('T-shirt', `up`, `2`), ('Shirt', `up`, `3`), ('Jeans', `down`, `1`),
('Shorts', `down`, `2`);
INSERT INTO clothingshop.brand (`name`)
VALUES ('Piazza Italia'), ('Levi's'), ('Garcia Jeans'), ('Colin's'), ('MR520'), ('MONTANA'), ('G-
Star Raw'), ('PEAK'), ('Weekday'), ('Under Armour'), ('No Excess'), ('Time of Style'), ('Koton'), ('H&M'),
('Jack & Jones'), ('United Colors of Benetton'), ('Fruit of the loom');
INSERT INTO clothingshop.clothes (`fk_category`, `fk_brand`, `name`, `stretching`)
VALUES
(1, 9, '02007960', 1),
(1, 12, '498F005', 1),
(1, 12, '498F001', 1),
(1, 12, '498F013', 1),
(1, 4, 'CLTTRMPLV026203018-0201', 1),
(2, 17, 'Valuweight 061036041', 1),
(2, 16, '3CNBJ1E46-907', 1),
(2, 1, '99253-13', 1),
(2, 13, '8YAM14243OK5', 1),
(2, 15, '5713026396112', 1),
(2, 5, 'MR 125 1238 0517', 1),
(2, 14, '0499402', 1),
(3, 1, '94997-261', 0),
(3, 1, '95008-58064', 1),
(3, 2, 'Barstow Western Gray 658160163', 0),
(3, 3, 'L51228-1567', 0),

```

(3, 4, 'CL1033215GRA', 0),
 (3, 5, 'MR 123 1330 0817', 0),
 (4, 2, '501 Original Fit Stonewash 00501-0114', 0),
 (4, 6, '1015802 Stone Bleached', 0),
 (4, 7, '3301 Tapered 51003.7209', 1),
 (4, 2, '504 Regular Straight Avatar Worn', 0),
 (5, 1, '95483-56 Verde', 0),
 (5, 1, '95483-62', 0),
 (5, 8, 'F352165-KHA', 1),
 (5, 8, 'F352951-BLU', 0),
 (5, 9, '02004472', 1),
 (5, 10, 'Ua Launch Sw 7 Print Short 1300057-016', 0),
 (5, 11, 'NX 858110384 019', 0);

-- таблицу clothes заполняем третьей, т.к. она ссылается на другие таблицы (по внешним ключам), которые должны быть заполнены перед ней (category, brand)

-- т.е. по внешним ключам обеспечивается целостность данных

```
INSERT INTO clothingshop.size (`size_number`)
```

```
VALUES
```

-- размеры для одежды на верхней части тела

('XS'), ('S'), ('M'), ('L'), ('XL'), ('XXL'), ('3XL'),

-- размеры для брюк

-- ('W-L') W - waist (обхват талии), L - long (длина штанины) *в дюймах

('28-32'), ('28-34'), ('28-36'),

('29-32'), ('29-34'), ('29-36'),

('30-32'), ('30-34'), ('30-36'),

('31-32'), ('31-34'), ('31-36'),

('32-32'), ('32-34'), ('32-36'),

('33-32'), ('33-34'), ('33-36'),

('34-32'), ('34-34'), ('34-36'),

('36-32'), ('36-34'), ('36-36'),

('38-32'), ('38-34'), ('38-36'),

('40-32'), ('40-34'), ('40-36'),

-- размеры для шорт (не зависят от длины)

-- W *в дюймах

('28'), ('29'), ('30'), ('31'), ('32'), ('33'), ('34'), /*('35'),*/ ('36'), ('38'), ('40');

/*~ ~ ~ ~ ~

S - 44 / M - 46 / L - 48 / XL - 50 / XXL - 52 / 3XL - 54 (международный и украинский размеры)

78, 31 / 82, 32 / 86, 34 / 90, 35 / 94, 36 / 98, 38 (размеры талии - см., дюймы)

~ ~ ~ ~ ~*/

INSERT INTO clothingshop.clothes_size (`id_clothes`, `id_size`, `price`, `availability`)

VALUES

-- above the waist

(1, 2, 1299, 1), (1, 3, 1299, 1), (1, 4, 1299, 1),

(2, 6, 608, 1), (2, 7, 608, 1),

(3, 3, 654, 1),

(4, 2, 585, 1), (4, 3, 585, 1), (4, 4, 585, 1), (4, 5, 585, 1),

(5, 2, 999, 0),

(6, 2, 99, 1), (6, 3, 99, 1), (6, 4, 99, 1), (6, 5, 99, 1), (6, 6, 99, 1),

(7, 3, 549, 1),

(8, 1, 469, 1), (8, 2, 469, 1), (8, 3, 469, 1), (8, 4, 469, 1), (8, 5, 469, 1),

(9, 3, 449, 1), (9, 4, 449, 1), (9, 5, 449, 1),

(10, 2, 380, 1),

(11, 2, 459, 0), (11, 3, 459, 0), (11, 4, 509, 0), (11, 5, 249, 0), (11, 6, 459, 0),

(12, 1, 687, 0), (12, 2, 687, 0), (12, 3, 687, 1), (12, 4, 687, 0),

(13, 1, 499, 1), (13, 2, 499, 1), (13, 3, 499, 1), (13, 4, 499, 1), (13, 5, 499, 1), (13, 6, 499, 1),

(14, 2, 779, 1), (14, 3, 779, 1), (14, 4, 779, 1), (14, 5, 779, 1),

(15, 2, 3089, 1), (15, 3, 3089, 0), (15, 5, 3089, 0), (15, 7, 3089, 0),

(16, 2, 1799, 0), (16, 3, 1799, 1), (16, 5, 1799, 0), (16, 6, 1799, 0), (16, 7, 1799, 0),

(17, 2, 499, 0), (17, 3, 277, 0), (17, 4, 277, 0), (17, 5, 277, 0),

(18, 2, 829, 1), (18, 3, 829, 0), (18, 4, 829, 0), (18, 5, 1019, 0), (18, 6, 829, 0),

-- below the waist

(19, 8, 3599, 1), (19, 11, 3599, 1), (19, 14, 3599, 1), (19, 15, 3599, 1), (19, 17, 3599, 1), (19, 18, 3599, 1), (19, 20, 3599, 1), (19, 21, 3599, 1), (19, 22, 3599, 1), (19, 23, 3599, 1), (19, 24, 3599, 1), (19, 25, 3599, 1), (19, 26, 3599, 1), (19, 27, 3599, 1), (19, 28, 3599, 1), (19, 30, 3599, 1), (19, 31, 3599, 1), (19, 32, 3599, 1), (19, 33, 3599, 1), (19, 34, 3599, 1), (19, 35, 3599, 1), (19, 36, 3599, 1), (19, 29, 3599, 0),

(20, 8, 2480, 1), (20, 21, 2480, 0), (20, 24, 1250, 0), (20, 27, 2480, 0),

(21, 18, 3490, 1), (21, 21, 3490, 1), (21, 24, 3490, 1), (21, 27, 3490, 1), (21, 30, 3490, 1), (21, 14, 3490, 1), (21, 17, 3490, 1), (21, 20, 3490, 1), (21, 26, 6980, 0), (21, 15, 3490, 0), (21, 33, 3490, 0),

(22, 15, 4179, 1), (22, 18, 4179, 1), (22, 21, 4179, 1), (22, 30, 4179, 0), (22, 29, 4179, 0), (22, 11, 4179, 0), (22, 17, 4179, 0),

```
(23, 42, 469, 1), (23, 45, 469, 1), (23, 44, 469, 1), (23, 41, 469, 1), (23, 46, 469, 0), (23, 47, 469, 0),
(24, 41, 469, 1), (24, 45, 469, 1), (24, 44, 469, 1), (24, 47, 469, 0), (24, 42, 469, 0), (24, 46, 469, 0),
(25, 42, 1199, 1), (25, 45, 1199, 1), (25, 41, 1199, 0), (25, 44, 1199, 0), (25, 46, 1199, 0),
(26, 42, 1099, 1), (26, 41, 1099, 0), (26, 44, 1099, 0), (26, 45, 1099, 0), (26, 46, 1099, 0),
(27, 44, 1049, 1),
(28, 41, 1396, 1), (28, 42, 1390, 0), (28, 44, 1390, 0), (28, 45, 1390, 0),
(29, 47, 899, 1), (29, 41, 1099, 0), (29, 43, 899, 0), (29, 46, 899, 0), (29, 40, 899, 0), (29, 42, 899, 0),
(29, 44, 899, 0), (29, 45, 899, 0);
```

```
INSERT INTO clothingshop.season (`name`)
```

```
VALUES ('весна-осень'), ('лето');
```

```
INSERT INTO clothingshop.category_season (`id_category`, `id_season`)
```

```
VALUES
```

```
(1, 1),
```

```
(2, 2),
```

```
(3, 1), (3, 2),
```

```
(4, 1), (4, 2),
```

```
(5, 2);
```

```
-- заполнять таблицу только запросом
```

	id_clothes	id_size	price	availability
	Фильтр	Фильтр	Фильтр	Фильтр
1	1	2	1299	1
2	1	3	1299	1
3	1	4	1299	1
4	2	6	608	1
5	2	7	608	1
6	3	3	654	1
7	4	2	585	1
8	4	3	585	1
9	4	4	585	1

Рисунок 2.6 – Заполнена таблица “clothes_size”

Браузер баз даних для SQLite є візуальним інструментом з відкритим вихідним кодом для створення, розробки і редагування файлів баз даних, сумісних з SQLite.

Інструмент для користувачів і розробників, яким потрібно створювати бази даних, шукати і редагувати дані в них. Він використовує звичний табличний інтерфейс, тому не потрібно вивчати складні SQL-команди.

Вся інформація про одяг, що виводиться у додатку, дістається із бази даних за допомогою відповідних SQL запитів.

Приклад запиту наведений у лістингу 2.3.

Лістинг 2.3 – Приклад SQL запиту

```
// вывод названия бренда и одежды
SELECT b.name || ' ' || c.name FROM clothes c INNER JOIN brand b ON c.fk_brand = b.id_brand
WHERE c.name = '02007960'
```

3 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ЯК ОСНОВА ПОБУДОВИ ТРИВИМІРНИХ МОДЕЛЕЙ

3.1 Існуючі підходи та методи створювання 3D моделей

Полігональне моделювання дає можливість проводити різні маніпуляції з сіткою 3d об'єкта на рівні підоб'єктів: вершин, ребер, граней. Сам полігон складається з граней, але в системах, які підтримують багатосторонні межі, полігони і межі будуть рівнозначні.

Це найперший і основний вид моделювання, так як за допомогою його можна створити об'єкт будь-якої складності шляхом з'єднання груп полігонів.

Полігональне моделювання поділяється на три типи: низькополігональне, середньополігональне і високополігональне.

Низькополігональне моделювання (Low-Poly) призначене для створення об'єктів з невеликим числом полігонів, зазвичай, для економії ресурсів, коли не потрібна висока деталізація, а так само для створення низькополігональних ілюстрацій, які набирають велику популярність останнім часом.

Середньополігональне моделювання (Mid-Poly) орієнтоване, зазвичай, тільки на необхідний результат при рендеринге, тобто при моделюванні потрібної геометрії, наприклад, із застосуванням булевих операцій; над полігональної сіткою ніякі роботи по її оптимізації не виробляють, або вони мінімальні.

Високополігональне моделювання (High-Poly) являє собою створення об'єкта з великим числом полігонів, зазвичай, точної його копії.

Стандартна схема High-Poly моделювання відбувається з поступовим нарощуванням рівня деталізації 3d об'єкта:

- перший рівень є базовим, і являє собою загальну форму об'єкта;
- на другому рівні відбувається уточнення базової форми, зазвичай, шляхом додавання фасок;
- третій рівень завершальний, тобто на ньому виробляється чітка деталізація об'єкта, зазвичай, шляхом застосування плагінів згладжування.

Моделювання сплайна є створення 3D об'єктів за допомогою кривих ліній (сплайнів). Сплайнами можуть виступати лінії різної форми: кола, прямокутники, дуги і т.д. Об'єкти при цьому виходять плавної форми, в зв'язку з чим, даний метод отримав широке застосування в створенні органічних моделей, таких як рослини, люди, тварини і т.д.

Перевага даного методу в гнучкості зміни форми сплайна.

Даний вид моделювання часто порівнюють з полігональним, як векторну графіку з растровою. Перевага векторної графіки в тому, що при збільшенні об'єкта, його якість не змінюється, на відміну від растрового, де стають видні пікселі. Так само і при збільшенні об'єкта, створеного сплайнами, його якість залишиться незмінним, а при полігональному моделюванні будуть вже видно полігони.

NURBS розшифровується як «Non-Uniform Rational B-Spline», і являє собою технологію створення 3d об'єктів за допомогою спеціальних кривих, які називаються B-сплайнами. Деякі фахівці виділяють даний вид моделювання в окремий, а деякі - в підвид моделювання сплайна.

Принцип моделювання полягає у наступному: за допомогою B-сплайнів, розташованих по вертикалі і горизонталі, будується потрібна форма об'єкта, а потім все це з'єднується за допомогою полігонів.

Існують два різновиди цього моделювання:

-При допомогі P-кривих (Point), форму яких можна змінювати за допомогою вершин, які розташовані безпосередньо на самій лінії;

-При допомогі CV-кривих (Control Vertex), форму яких можна змінювати за допомогою вершин, які розташовані за межами лінії.

NURBS моделювання застосовується майже у всіх популярних програмах 3d моделювання спільно з іншими видами.

Як же прикладів програм, які використовують моделювання кривими як основний метод, можна привести «Rhinoceros», «Autodesk Alias», «MOI 3D», «SolidThinking».

3d-скульптинг він же «цифрова скульптура» являє собою імітацію процесу «ліплення» 3d моделі, тобто деформування її полігональної сітки спеціальними

інструментами - кистями. Можна провести аналогію з ліпленням фігур руками з пластиліну або глини. Тільки в програмах 3d моделювання пальці замінені на інструмент «кисть», а «пластиліном» є полігональна сітка.

Програмами-представниками даного виду моделювання є «ZBrush», «Sculptris», «Autodesk Mudbox» і ін.

Таким чином, можна зробити наступний висновок. При створенні моделей не складної форми краще використовувати полігональне моделювання. Для отримання гладкої форми нескладних об'єктів - сплайнова або NURBS моделювання, або полігональне з використанням інструментів згладжування [12].

При створенні складних біологічних організмів зручніше використовувати 3d-скульптинг. Коли ж необхідно створити точну модель з необхідними зазорами і урахуванням фізичних властивостей матеріалу, то тут найбільш підходять методи промислового моделювання.

При створенні складних моделей вищеописані методи моделювання часто використовуються спільно, так як це прискорює процес моделювання. Так, наприклад, при створенні персонажа для ігор використовується 3d-скульптинг, за допомогою якого вимальовуються необхідні дрібні деталі, а потім на її основі створюється Low-Poly модель полігональним моделюванням.

У багатьох пакетах 3d моделювання є інструменти майже для всіх видів моделювання, наприклад, в «Blender», «Modo». Але, «На смак і колір товаришів немає», тому зустрічаються і такі фахівці, хто моделює в одній програмі, скульптинг робить в іншій, а ретопологію в третій і т.д.

3.2 Розробка тривимірних моделей

Наступний етап - створення тривимірних моделей, а саме моделі манекену та моделей одягу, з подальшим текстуруванням. Для цього буде застосована програма для роботи з 3D графікою Blender.

Blender - це безкоштовне програмне забезпечення для створення і редагування тривимірної графіки. З огляду на платформ, відкритого вихідного коду, доступності

і функціональності пакет отримав заслужену популярність не тільки серед новачків, а й серед просунутих 3D-моделерів. У міру розвитку програми її вибирають в якості робочого інструменту для все більш серйозних проектів, що не дивно. По суті, це додаток практично не поступається за кількістю можливостей і функціоналу більш просунутим пакетам 3D графіки. І при цьому все безкоштовно.

Blender позиціонується як додаток для створення і редагування тривимірної графіки, візуалізації, анімації, створення комп'ютерних ігор і навіть скульптінга. Цілком серйозна програма, для якої потрібні серйозні ресурси апаратного забезпечення, скажете ви. Але перший сюрприз пакет підносить на етапі скачування настановних файлів - їх вага становить близько 70 Мбайт. Немислимо для програмного забезпечення такого рівня! Проте, розробники примудрилися забезпечити програму усіма необхідними функціями, які повноцінно функціонують і демонструють відмінну продуктивність. Додатковою перевагою є стабільне і стрімкий розвиток пакета завдяки професійній команді розробників.

На сьогоднішній день це повноцінний 3D редактор, в якому користувача зустрічає повністю програмований інтерфейс і унікальна внутрішня файлова система. Оболонка програми на перший погляд може здатися незручною і незрозумілою, але після налаштування гарячих клавіш працювати в Blender стає просто і зручно. В якості мови програмування додаток використовує Python, володіючи яким ви можете створювати власні інструменти, редагувати інтерфейс і сам принцип роботи програми. Приємним бонусом є доступність пакета на різних операційних системах обох розрядностей: освоїти програму зможуть власники комп'ютерів з ОС Windows, GNU / Linux і Mac OSX.

3D моделювання. Представлено практично всіма існуючими способами створення і роботи з об'ємними моделями. Доступно проектування об'єктів на основі примітивів, полігонів, NURBS-кривих, кривих Безьє, метасфер, булевих операцій, Subdivision Surface і базових інструментів для скульптінга. Як і в 3Ds Max, програма пропонує велику кількість різних модифікаторів, які застосовуються до моделі.

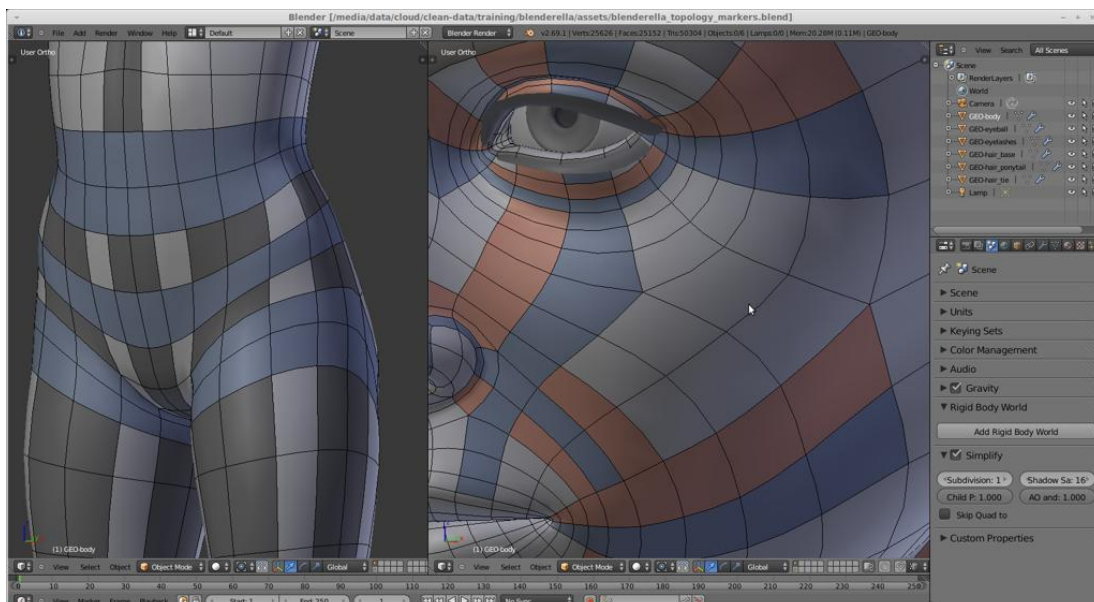


Рисунок 3.1 – Огляд можливостей

Анімація. Дійсно добре поставлена в пакеті. У розпорядженні користувача такі інструменти, як рігінг (скелетна анімація), інверсна кінематика, сіткова деформація, обмежувачі, ключові кадри, редагування вагових коефіцієнтів вершин і т.д. Відмінно реалізована динаміка твердих і м'яких тіл, а також анімація частинок.

Текстурування і набори шейдерів. Програма дозволяє накладати кілька текстур на один об'єкт, і оснащена рядом інструментів для текстурування, включаючи UV-маппінг і часткове настроювання текстур. Ряд шейдерів, що налаштовуються додає гнучкості в роботі з матеріалами.

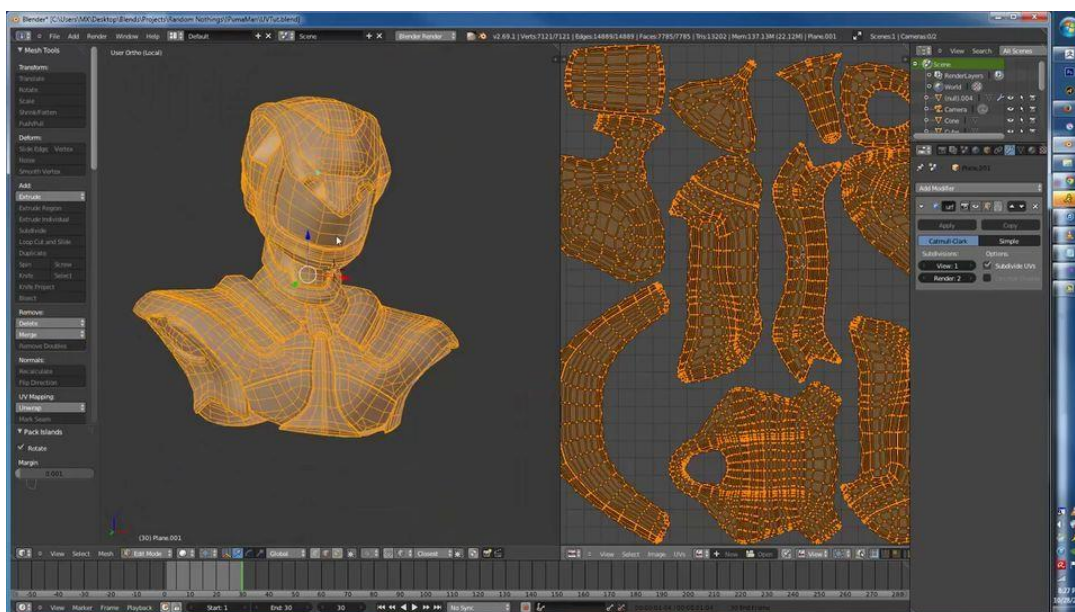


Рисунок 3.2 – Огляд можливостей

Можливість малювання. Так, ця програма для 3D моделювання надає можливість створювати начерки різними типами кистей прямо у вікні програми. Поточне призначення такої функції - допомога у створенні 2D анімації, для чого ця функція також оснащена можливістю гнучкого налаштування, зокрема, роботи з шарами.

Візуалізація. Пакет оснащений декількома вбудованими інструментами візуалізації, а також підтримує інтеграцію з різними зовнішніми рендерами.

Базовий відеоредактор. Функція, про яку не здогадуються навіть багато просунуті користувачі програми. У Blender є вбудований відеоредактор, що не настільки потужний, як спеціалізоване ПО для цих цілей, але вельми непоганий.

Ігровий движок. Надзвичайно цікава функція програми - вбудований ігровий движок для створення інтерактивних 3D додатків. А програмний інтерфейс програми Python API дозволяє самостійно вносити будь-які корективи в створювану гру.



Рисунок 3.3 – Створений манекен з одягом



Рисунок 3.4 – Створені моделі з текстурами

Для побудови моделей була використана полігональна сітка (Meshes).

За допомогою сіток все будується з трьох основних структур: вершин, країв та граней.

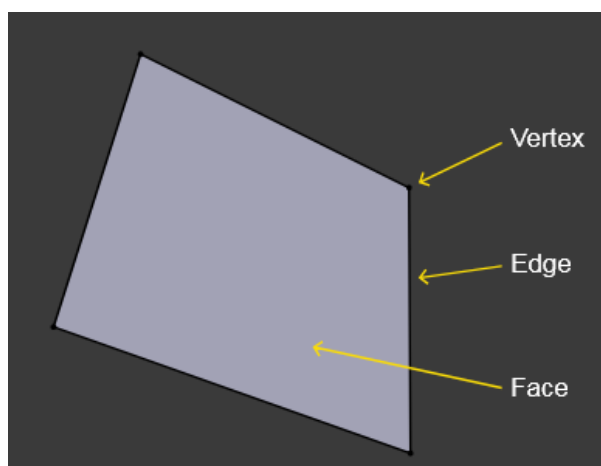


Рисунок 3.5 – Приклад структури полігональної сітки

Вершини (Vertices). Елементарною складовою полігональної сітки є вершина з однією єдиною координатою положення в тривимірному просторі. Вершини зберігаються в масиві з координатами.

Ребра (Edges). Ребро завжди з'єднує дві вершини прямою лінією. Ребра виглядають як «дріт», коли ви дивитесь на полісітку в каркасному (wireframe) режимі. Вони зазвичай не видно при візуалізації. Ребра беруть участь в побудові граней.

Грані (Faces). Грані утворюють фактичну поверхню об'єкта. Вони залишаються видимими при візуалізації полігональної сітки. Якщо область не містить межі, то вона буде прозорою або бути відсутнім на зображенні при візуалізації.

Грань визначається як площа між трьома (трикутниками), чотирма (чотирикутниками) або більше (n-кутниками) вершинами, з ребром на кожній стороні.

Трикутники завжди утворюють площину і тому легко обчислюються. З іншого боку, чотирикутники «більш пластичні» що переважно для анімації і при техніці subdivision modeling.

Також при створенні моделей була застосована динамічна топологія.

Динамічна топологія (вона ж dyntopo) - це метод динамічного тесселяційного ліплення, який додає та видаляє деталі на льоту, тоді як звичайне ліплення впливає лише на форму сітки.

Це дає змогу виліплювати складні форми з простої сіті, а не тільки просто додавати деталі на змодельовану базову сіть.

Усі текстури для одягу створювались на основі реально існуючого одягу.



Рисунок 3.6 – Створення текстур одягу на основі їх фото

Для усіх категорій одягу, а саме: футболки, сорочки, светри, джинси та шорти, було створено декілька текстур (відповідно доданим даним про позиції одягу із БД).

4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМПОНЕНТІВ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ПІДБОРУ ОДЯГУ

4.1 Реалізація додатку

Тепер можемо перейти до створення проекту у середовищі для створення ігрових додатків Unity (рис. 4.1).



Рисунок 4.1 – Перенесена модель манекену

Серед ігрових движків Unity займає далеко не останнє місце. Його використовують і великі розробники, і (набагато частіше) невеликі незалежні студії. У цій статті ми розповімо про особливості, сильні та слабкі сторони движка, а також типах проектів, в яких його використання буде найбільш доцільним.

Unity - більше, ніж движок, це середовище для розробки комп'ютерних ігор, в якій об'єднані різні програмні засоби, що використовуються при створенні ПЗ - текстовий редактор, компілятор, відладчик і так далі. При цьому, завдяки зручності використання, Unity робить створення ігор максимально простим і комфортним, а мультиплатформеність движка дозволяє ігоробів охопити якомога більшу кількість ігрових платформ і операційних систем.

Чим гарний Unity? В першу чергу, як ми вже згадували, движок Unity3D дає можливість розробляти гри, не вимагаючи для цього якихось особливих знань. Тут використовується компонентно-орієнтований підхід, в рамках якого розробник створює об'єкти (наприклад, головного героя) і до них додає різні компоненти (наприклад, візуальне відображення персонажа і способи управління ним). Завдяки зручному Drag & Drop інтерфейсу і функціональним графічному редактору движок дозволяє малювати карти і розставляти об'єкти в реальному часі і відразу ж тестувати вийшов результат.

Друга перевага движка - наявність величезної бібліотеки Ассет і плагінів, за допомогою яких можна значно прискорити процес розробки гри. Їх можна імпортувати і експортувати, додавати в гру цілі заготовки - рівні, ворогів, патерни поведінки П і так далі. Ніякої метушні з програмуванням. Багато Ассет надаються безкоштовно, інші пропонуються за невелику суму, і при бажанні можна створювати власний контент, публікувати його в Unity Asset Store і отримувати від цього прибуток.

Третя сильна сторона Unity 3D - підтримка величезної кількості платформ, технологій, API. Створені на движку гри можна легко перенести між ОС Windows, Linux, OS X, Android, iOS, на консолі сімейств PlayStation, Xbox, Nintendo, на VR- і AR-пристрої. Unity підтримує DirectX і OpenGL, працює з усіма сучасними ефектами рендеринга, включаючи новітню технологію трасування променів в реальному часі.

Фізика твердих тіл, ragdoll і тканин, система Level of Detail, колізії між об'єктами, складні анімації - все це можна реалізувати силами движка. Стереотипна думка про те, що движок придатний тільки для невеликих інді-ігор і нездатний видавати красиву картинку, давно вже не актуально: досить подивитися технодемо ADAM, The Blacksmith і Book of the Dead від творців середовища Unity, щоб переконатися в її видатних здібностях.

Нарешті, Unity доступний безкоштовно, що відкриває перед незалежними розробниками двері в ігрову індустрію. Звичайно, існують обмеження: безкоштовна версія движка демонструє лого Unity перед запуском гри, а проект, створений з її допомогою, не повинен приносити розробнику більше \$ 100 тисяч в рік. Втім, тарифи

на підписку не спустошать гаманці навіть починає команди: Про-версія коштує \$ 125 на місяць, що не так вже й багато в порівнянні з іншими двигунами, причому базова версія містить рівно той же функціонал, що і професійна.

При всіх своїх перевагах, движок має і свої недоліки. Так, якщо команда захоче розробити що-небудь складніше простого клікера або платформера, то їй доведеться шукати хорошого програміста на C#, який напише скрипти і компоненти, запровадить їх у гру і змусить працювати.

З цього впливає інша проблема движка Unity - повільність. Створення масштабних, складних сцен з великою кількістю компонентів може негативно вплинути на продуктивність гри, в результаті чого розробникам доведеться витратити додатковий час і ресурси на оптимізацію, а можливо - і видалення деяких елементів з проекту.

Крім того, додатки, створені на Unity, досить «великовагових»: навіть найпростіша піксельна гра може займати кілька сотень мегабайт на ПК. Так, для жорсткого диска комп'ютерів це невеликий обсяг, але, якщо проект розробляється і для мобільних платформ, слід задуматися про оптимізацію його розміру.

4.2 Доповнена реальність та структура платформи Vuforia

Доповнена реальність - це технологія додавання, впровадження в реальне життя, в тривимірне поле сприйняття людини віртуальної інформації, яка сприймається як елементи реального життя. При створенні доповненої реальності в режимі реального часу поміщаються об'єкти за допомогою спеціального програмного забезпечення і гаджетів, таких як: окуляри доповненої реальності («розумні окуляри»), планшети, смартфони з функцією AR і інші гаджети.

Маркетинг і рекламний бізнес одними з перших взяли на озброєння можливості AR. Доповнена реальність в рекламі використовується в усіх сферах. Функціями доповненої реальності забезпечуються вітрини, примірочні, інтерактивні кіоски, промо-стенди. Рекламні кампанії, проведені з використанням технологій доповненої реальності, викликають широкий резонанс і, що найголовніше,

ефективно впливають на цільову аудиторію. Покупець може розглянути предмет з усіх боків, «повертати» його на 360 градусів.

Біологія, архітектура і будівництво, археологія - в цих сферах діяльності активно впроваджується і використовується технологія доповненої реальності. Завдяки AR візуалізація в архітектурі і будівництві переходить на якісно новий рівень, дозволяючи на етапі проектування побачити кінцевий результат, завдяки чому знижуються ризики в виробничому процесі.

Виставкова, експозиційна, музейна діяльність, мистецтво і дизайн, відеоінсталяції та інтерактивні макети із застосуванням доповненої реальності - це далеко не вичерпний список сфер нашого життя, де можна використовувати технології доповненої реальності.

Vuforia - це програмна платформа для розробки додатків доповненої реальності (AR, AugmentedReality). За допомогою Vuforia розробник може легко додавати функціонал комп'ютерної візуалізації віртуальних 3D-об'єктів в будь-який додаток, реалізуючи можливості розпізнавання і більш точного розуміння зображень і об'єктів, що спостерігаються в дійсності або виконуючи 3D-реконструкцію навколишнього і спостережуваного простору в реальному світі.

У Vuforia ви можете створювати AR-додатки для широкого кола індустріальних завдань, будь то інтерактивні інструкції робочого місця, маркетингові матеріали, сервісні керівництва - для усього розмаїття можливих AR-додатків в Vuforia є закінчений набір функціональності.

Платформа Vuforia надає повноцінну інтелектуальне середовище додатків доповненої реальності, яку можна використовувати як зі своєю власною системою розробки сцен і сценаріїв (VuforiaStudio), так і спільно з найбільш популярними системами розробки сцен і сценаріїв 3D (VuforiaSDK). Vuforia як платформа розробки і в варіанті VuforiaSDK, і в варіанті VuforiaStudio володіє наступними характеристиками.

Сучасна і повноцінна платформа доповненої реальності, лідер в області AR.

Розробка сцен і сценаріїв на базі моделей ведеться практично без програмування.

Об'єднання людей, систем і машин реального світу і віртуального уявлення виробу.

Використання популярного обладнання для візуалізації нижчого цінового діапазону смартфонів, планшетів, очок VR.

Хмарні сервіси по зберіганню і надання розроблених сцен і сценаріїв - реалізація принципу «розробляв де хочеш - дивись де можеш».

Розгортання - так, як зручно замовнику.

Розвиток і зростання додатки в майбутньому.

Ядром платформи є бібліотека QCAR, написана на C ++ і ховає за свої інтерфейси практично все від визначення мішені (target) і закінчуючи рендерингом картинки.

Крім бібліотеки, платформа включається в себе: iOS і Android Vuforia SDK для розробників; Target Manager - систему для створення і управління мішенями; а також набір web сервісів (Vuforia Web Services) в які можна винести зберігання мішеней і ще певний функціонал. Платформа розвивається дуже динамічно і нові частини додаються досить часто.

Мішені (targets). Щоб щось додати в реальний світ, потрібно в цьому світі "зачепитися" за який-небудь об'єкт. Саме для цієї мети і служать мішені. Іншими словами, мішень - це якийсь реальний об'єкт, знаючи який наш додаток може розставити віртуальні об'єкти в потрібних місцях і відповідних пропорціях.

Vuforia надає багатий вибір мішеней:



Рисунок 4.2 – Мішені

Image targets - базовий вид мішеней, що представляє собою звичайну картинку, наприклад, обкладинку журналу, фотографію або афішу нового фільму. Картинка виконує роль свого роду двовимірного штрих-коду, тільки без чорно-білих регіонів.

По ній ми можемо визначити, яка саме картинка потрапила в об'єктив камери, а також її розташування в просторі і масштаб. Варто сказати, що не будь-яка картинка підійде для створення мішені. Добрими мішенями є ті, в яких багато контрастних деталей. Саме на цих деталях і будується опорна матриця для подальшого розпізнання мішеней.

Simple 3D targets (Cube and Cuboid) - це мішені у вигляді прямокутних паралелепіпедів (включаючи куб). Наприклад, такий мішенню може служити упаковка з-під сухих сніданків, сірникову коробку або тільки що куплена настільна гра. Як і будь-яка коробка, така мішень складається з шести площин, і щоб створити її, нам знадобиться шість картинок для кожної з них. Якщо ви коли-небудь склеювали кубик з паперу - то швидко впораєтесь із завданням.

Cylinder targets - цей вид мішеней, незважаючи на назву, являє собою усічений конус з можливістю задавати діаметри підстав. Звичайно якщо вибрати однакові діаметри, то вийдуть якраз циліндр, але все ж це окремий випадок. Для того, щоб створити таку мету нам знадобиться не тільки вказати діаметри підстав і висоту, але також додати три картинки - по одній для кожного з двох підстав, і ще одну для бічної поверхні.

Frame markers - ця мета у вигляді спеціально підготовленої рамки, яка вже куди більше скидається на штрих-код. У таку рамку можна помістити будь-яку картинку. Даний вид мішеней відмінно підходить в разі, якщо картинка не була достатньо деталізована і створити з неї розумну image target не вийшло.

Text (word targets). У бібліотеку Vuforia вбудовано ще і розпізнання тексту, тому будь-яке слово або їх поєднання може бути мішенню. На даний момент підтримується тільки латиниця, але і це вже дуже цікаво. Тільки уявіть наскільки цікавим може стати вивчення іноземних слів.

В залежність від кількості мішеней, необхідних для вашого застосування, ви можете або зберігати їх в так званій Device Database, завжди мати до них доступ і розпізнавати їх безпосередньо на самому пристрої, або перекласти частину цього тягаря на Cloud Databases - сервіс з набору Vuforia Web Services, призначений для зберігання мішеней і визначення їх на підставі надісланих з пристрою даних. Обидва

підходи мають свої плюси і мінуси, але визначившись з цілями вашого застосування вибрати буде не так складно.

Крім цього, мішені можуть бути створені безпосередньо з самого додатка використовуючи картинку отриману з камери.

Варто також відзначити, що QCAR не вимагає попадання всій мішені в кадр. Для того, щоб її розпізнати і використовувати досить буде лише частини. Це дуже важливо і дозволяє розглядати всю магію вашого застосування під різними кутами.

Тепер, отримавши можливість "зацепитися", ми можемо починати додавати ту саму магію, яку можна буде побачити на екрані смартфона в нашому реальному світі.

Для цього давайте розберемося яку магію ми можемо собі дозволити. Для цього я постарався зібрати основні приклади використання augmented reality і все з них можуть бути використані з Vuforia.

Статичні і динамічні 3D моделі. Ми можемо додати будь-який 3D об'єкт починаючи від улюбленого всіма чайника і закінчуючи складною анімацією. Vuforia активно підтримує можливість інтеграції з Unity, що ще сильніше розв'язує нам руки. Обмежені ми тільки фантазією і потужністю смартфонів / планшетів.

Віртуальний кнопки взаємодіяти з якими доведеться цілком собі предметами реальними. Наприклад, це може бути кнопка на вашій мішені, натискання на яку запускає віртуальну ракету земля-повітря з вашого робочого столу розглянути яку можна тільки через екран смартфона.

Відео та картинки, які дозволяють замінити нашу мету або зробити її трішечки більш живою.

Зміна фону, крім усього іншого, ви можете змінювати і саму картинку отриману з камери. Іншими словами, ми можемо не тільки доповнювати картинку, але і оперувати з нею самою.

Маючи такий значний набір засобів можна зробити практично все, що завгодно.

Таким чином до проекту був доданий плагін Vuforia, який дозволяє працювати програмі у режимі AR використовуючи камеру мобільного пристрою. Демонстрація роботи відображена на рис. 4.3.



Рисунок 4.3 – Перша версія додатку

Зображення мітки (target) типу Image Target, яке буде використане у додатку, для подальшого виведення на ньому 3d моделі манекену (рис. 4.4).



Рисунок 4.4 – Мітка, яка використовується

4.3 Реалізація інтерфейсу користувача

Наступним кроком являється створення інтерфейсної частини програми. Спочатку був створений концепт (макет) UI (рис. 4.5) на основі якого в подальшому був створений інтерфейс додатку (рис. 4.6).

Інтерфейс було розроблено для користувачів смартфонів.



Рисунок 4.5 – Концепт інтерфейсу основного екрану додатку

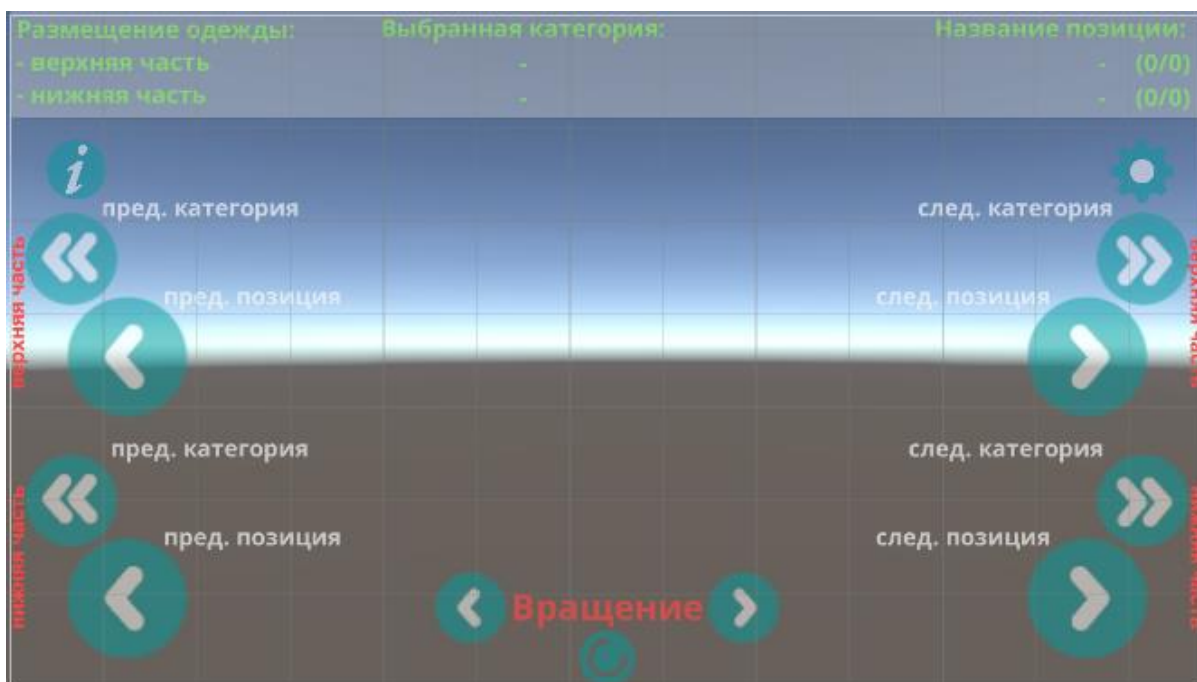


Рисунок 4.6 – Кінцеве представлення елементів на основному екрані

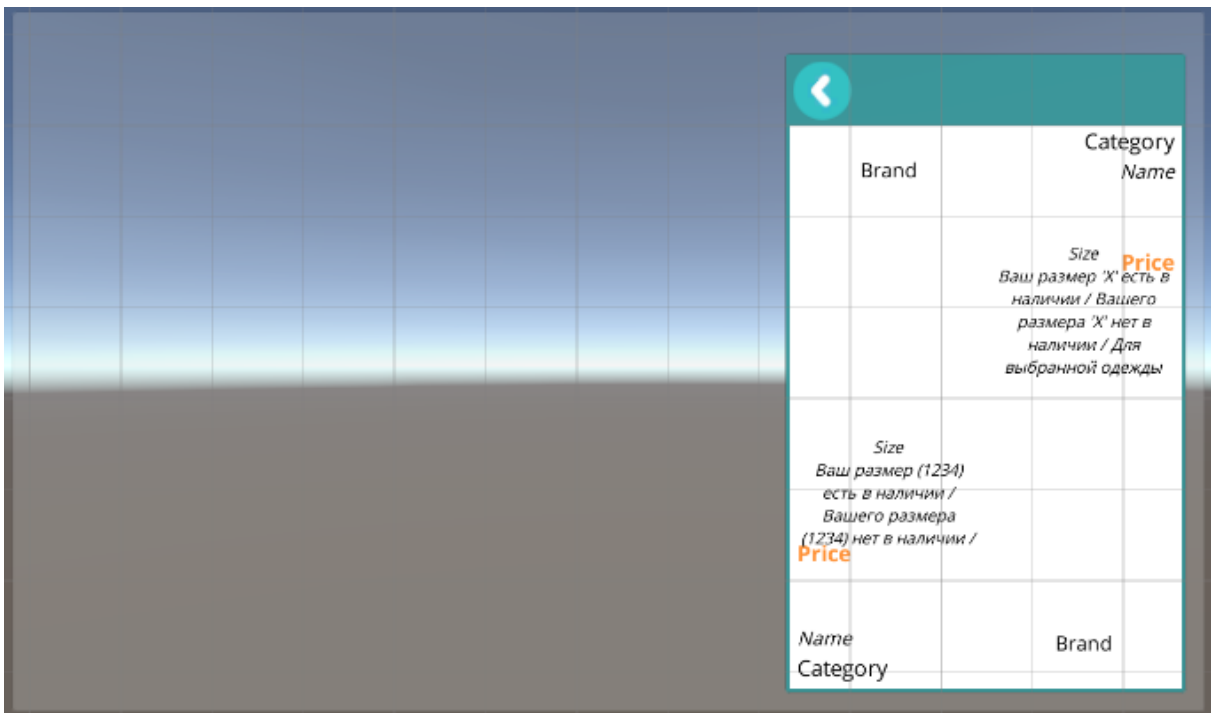


Рисунок 4.7 – Представлення блоку додаткової інформації

4.4 Реалізація логіки додатку

У ході виконання була розроблена база даних проекту, а зараз з'явилася необхідність у реалізації способів роботи з цією БД та її змістом всередині створюваного додатку.

Зробимо тестове під'єднання бази даних до додатку.

Лістинг 4.1 – Код під'єднання (файл DBcontroller.cs)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
// ссылки на добавленные компоненты (папка Plugins)
using Mono.Data.Sqlite;
using System.Data;
using System.IO;
using System;
using UnityEngine.UI;

public class DBcontroller : MonoBehaviour {

    // объявление необходимых переменных для работы с БД
    public SqliteConnection con_db;
    public SqliteCommand cmd_db; // команда с SQL запросом (для отправки в БД)
    public SqliteDataReader rdr; // для чтения данных из БД

    // путь для подключения
```

```

private string path;

Text text;

// Use this for initialization
void Start () {
    text = GameObject.Find("testText").GetComponent<Text>();
}

// Update is called once per frame
void Update () {

}

// метод для подключения
public void Connection()
{
    // блок try-catch для отлова ошибок
    try
    {
        // если текущая платформа не android - путь к БД один (БД в папке Assets), если
        android - путь другой (БД для android в папке StreamingAssets)
        if (Application.platform != RuntimePlatform.Android)
        {
            // подключение к БД для Windows
            path = Application.dataPath + "/ClothingShop.bytes";
        }
        else
        {
            path = Application.persistentDataPath + "/ClothingShop.bytes";
            if (!File.Exists(path))
            {
                // если по указанному пути нет файла БД - будет создан класс
                WWW load = new WWW("jar:file://" + Application.dataPath + "!/assets/" +
                "ClothingShop.bytes"); // копирование файла с БД
                while (!load.isDone) { }
                File.WriteAllBytes(path, load.bytes); // в путь path копируется БД (из уже
                упакованного арк файла)
            }
            // установка соединения
            con_db = new SqlConnection("URI = file:" + path);
            con_db.Open();
            if (con_db.State == ConnectionState.Open)
            {
                // если соединение открыто
                text.text = path.ToString() + " - is connected";
            }
            // создание команды
            cmd_db = new SqlCommand("SELECT * FROM brand WHERE id_brand = 1", con_db); //
            указание запроса, указание созданного соединения
            rdr = cmd_db.ExecuteReader(); // чтение ответа на запрос
            while (rdr.Read()) // действия во время чтения
            {
                text.text = rdr[1].ToString(); // [n] - вывод последнего значения из n столбца
                (от 0)
            }
        }
        catch (Exception ex)
        {
            text.text = ex.ToString(); // отображение ошибки (при её наличии)
        }
    }
}

```

Наприкінці залишилось додати функціоналу для усіх кнопок та інших елементів програми (перемикання категорій, перехід в категорії з одного одягу до іншого, відображення детальної інформації, обертання моделі манекену та ін.), але найважливішою функціональною частиною є логіка здійснення вибору необхідних розмірів одягу та підлаштовування моделі манекену під задані параметри розмірів користувача.

Лістинг 4.2 – Код логіки для обрання розмірів одягу на основі даних, які були введені користувачем (частина файлу BodyParameters.cs)

```
public void DeterminationClothingSize()
{
    /* Для одежды на верхней части тела */
    /* Определение международного размера по обхвату груди */
    if (chestGirth <= 89 && chestGirth >= 86)
    {
        numberSizeChestGirth = 1;
        // XS
    }
    else if (chestGirth <= 93 && chestGirth >= 90)
    {
        numberSizeChestGirth = 2;
        // S
    }
    else if (chestGirth <= 97 && chestGirth >= 94)
    {
        numberSizeChestGirth = 3;
        // M
    }
    else if (chestGirth <= 101 && chestGirth >= 98)
    {
        numberSizeChestGirth = 4;
        // L
    }
    else if (chestGirth <= 105 && chestGirth >= 102)
    {
        numberSizeChestGirth = 5;
        // XL
    }
    else if (chestGirth <= 109 && chestGirth >= 106)
    {
        numberSizeChestGirth = 6;
        // XXL
    }
    else if (chestGirth <= 113 && chestGirth >= 110)
    {
        numberSizeChestGirth = 7;
        // 3XL
    }
    /* В случае, если нет подходящих размеров - берём по ближайшим */
    else if (chestGirth < 86)
    {
        numberSizeChestGirth = 1;
    }
}
```

```

        chestGirth = 85; // установка значения из диапазона (но разница в единицу - чтобы
было ясно, что пределы были нарушены)
        InputFieldTwo.GetComponent<InputField>().text = Convert.ToString(chestGirth); //
отображение этого значения в поле
        // XS
    }
    else if (chestGirth > 113)
    {
        numberSizeChestGirth = 7;
        chestGirth = 114;
        InputFieldTwo.GetComponent<InputField>().text = Convert.ToString(chestGirth);
        // 3XL
    }

    /* Определение международного размера по обхвату талии */
    if (waistGirth <= 77 && waistGirth >= 73)
    {
        numberSizeWaistGirthUP = 1;
        // XS
    }
    else if (waistGirth <= 81 && waistGirth >= 78)
    {
        numberSizeWaistGirthUP = 2;
        // S
    }
    else if (waistGirth <= 85 && waistGirth >= 82)
    {
        numberSizeWaistGirthUP = 3;
        // M
    }
    else if (waistGirth <= 89 && waistGirth >= 86)
    {
        numberSizeWaistGirthUP = 4;
        // L
    }
    else if (waistGirth <= 94 && waistGirth >= 90)
    {
        numberSizeWaistGirthUP = 5;
        // XL
    }
    else if (waistGirth <= 100 && waistGirth >= 95)
    {
        numberSizeWaistGirthUP = 6;
        // XXL
    }
    else if (waistGirth <= 109 && waistGirth >= 101)
    {
        numberSizeWaistGirthUP = 7;
        // 3XL
    }
    /* В случае, если нет подходящих размеров - берём по ближайшим */
    else if (waistGirth < 73)
    {
        numberSizeWaistGirthUP = 1;
        waistGirth = 72;
        InputFieldThree.GetComponent<InputField>().text = Convert.ToString(waistGirth);
        // XS
    }
    else if (waistGirth > 109)
    {
        numberSizeWaistGirthUP = 7;
        waistGirth = 110;
        InputFieldThree.GetComponent<InputField>().text = Convert.ToString(waistGirth);
        // 3XL
    }
}

```

```

        if ((numberSizeWaistGirthUP != numberSizeChestGirth) && (numberSizeChestGirth -
numberSizeWaistGirthUP == 1 || numberSizeWaistGirthUP - numberSizeChestGirth == 1))
        {
            chooseSizeUP = (numberSizeChestGirth > numberSizeWaistGirthUP) ? numberSizeChestGirth
: numberSizeWaistGirthUP;
            Debug.Log(0);
        }
        else if (numberSizeChestGirth - numberSizeWaistGirthUP == 2 || numberSizeWaistGirthUP -
numberSizeChestGirth == 2)
        {
            if (numberSizeChestGirth - numberSizeWaistGirthUP == 2)
            {
                chooseSizeUP = --numberSizeChestGirth;
                Debug.Log(1);
            }
            else
            {
                chooseSizeUP = --numberSizeWaistGirthUP;
                Debug.Log(2);
            }
        }
        else
        {
            chooseSizeUP = numberSizeChestGirth; // если размеры совпали / при большой разнице -
обхват груди в приоритете
            Debug.Log(3);
        }

        /* Итоговый выбор размера для одежды на верхней части тела */
        if (chooseSizeUP == 1)
        {
            SizeOne.text = "XS";
            // XS
        }
        else if (chooseSizeUP == 2)
        {
            SizeOne.text = "S";
            // S
        }
        else if (chooseSizeUP == 3)
        {
            SizeOne.text = "M";
            // M
        }
        else if (chooseSizeUP == 4)
        {
            SizeOne.text = "L";
            // L
        }
        else if (chooseSizeUP == 5)
        {
            SizeOne.text = "XL";
            // XL
        }
        else if (chooseSizeUP == 6)
        {
            SizeOne.text = "XXL";
            // XXL
        }
        else if (chooseSizeUP == 7)
        {
            SizeOne.text = "3XL";
            // 3XL
        }
    }

```

```

/* "Запасной" размер */
if (chooseSizeUP != 1)
{
    chooseSizeUpTEMP = Convert.ToByte(chooseSizeUP - 1);
    if (chooseSizeUpTEMP == 1)
    {
        SizeOneTEMP.text = "XS";
    }
    else if (chooseSizeUpTEMP == 2)
    {
        SizeOneTEMP.text = "S";
    }
    else if (chooseSizeUpTEMP == 3)
    {
        SizeOneTEMP.text = "M";
    }
    else if (chooseSizeUpTEMP == 4)
    {
        SizeOneTEMP.text = "L";
    }
    else if (chooseSizeUpTEMP == 5)
    {
        SizeOneTEMP.text = "XL";
    }
    else if (chooseSizeUpTEMP == 6)
    {
        SizeOneTEMP.text = "XXL";
    }
}

/* Для одежды на нижней части тела */
/* Определение размера в дюймах по обхвату талии */
/* Для шорт достаточно одного размера */
if (waistGirth <= 74 && waistGirth >= 73)
{
    SizeTwo.text = "28";
    // 28
}
else if (waistGirth <= 77 && waistGirth >= 75)
{
    SizeTwo.text = "29";
    // 29
}
else if (waistGirth <= 81 && waistGirth >= 78)
{
    SizeTwo.text = "30";
    // 30
}
else if (waistGirth <= 85 && waistGirth >= 82)
{
    SizeTwo.text = "31";
    // 31
}
else if (waistGirth <= 89 && waistGirth >= 86)
{
    SizeTwo.text = "32";
    // 32
}
else if (waistGirth <= 93 && waistGirth >= 90)
{
    SizeTwo.text = "33";
    // 33
}
else if (waistGirth <= 97 && waistGirth >= 94)
{

```

```

        SizeTwo.text = "34";
        // 34
    }
    else if (waistGirth <= 101 && waistGirth >= 98)
    {
        SizeTwo.text = "36";
        // 36
    }
    else if (waistGirth <= 105 && waistGirth >= 102)
    {
        SizeTwo.text = "38";
        // 38
    }
    else if (waistGirth <= 109 && waistGirth >= 106)
    {
        SizeTwo.text = "40";
        // 40
    }
    /* В случае, если нет подходящих размеров (выход из диапазона допустимых значений) -
берём по ближайшим */
    else if (waistGirth < 73)
    {
        SizeTwo.text = "28";
        // 28
    }
    else if (waistGirth > 109)
    {
        SizeTwo.text = "40";
        // 40
    }
    }
    /* "Запасной" размер */
    if (SizeTwo.text != "28") // если выбран не минимально возможный размер
    {
        TEMP = Convert.ToByte(SizeTwo.text);
        if (Convert.ToByte(SizeTwo.text) <= 34) // -1 единица
        {
            SizeTwoTEMP.text = Convert.ToString(--TEMP);
        }
        else // -2 единицы
        {
            SizeTwoTEMP.text = Convert.ToString(TEMP - 2);
        }
        Debug.Log(TEMP + "!!!" + SizeTwoTEMP.text);
    }
}

/* Определение длины штанины (для джинсов) в дюймах по росту */
if (height <= 177 && height >= 165)
{
    SizeThree.text = SizeTwo.text + "-32"; // обхват талии + длина штанины
    // 32
    /* "Запасной" размер */
    /*
        Длину желательно не уменьшать
    */
    SizeThreeTEMP.text = SizeTwoTEMP.text + "-32";
}
else if (height <= 185 && height >= 178)
{
    SizeThree.text = SizeTwo.text + "-34";
    // 34
    SizeThreeTEMP.text = SizeTwoTEMP.text + "-34";
    Debug.Log(SizeThreeTEMP.text + "!!!");
}
else if (height <= 190 && height >= 186)
{

```

```

    SizeThree.text = SizeTwo.text + "-36";
    // 36
    SizeThreeTEMP.text = SizeTwoTEMP.text + "-36";
}
/* В случае, если нет подходящих размеров - берём по ближайшим */
else if (height < 165)
{
    SizeThree.text = SizeTwo.text + "-32";
    height = 164;
    InputFieldOne.GetComponent<InputField>().text = Convert.ToString(height);
    // 32
    SizeThreeTEMP.text = SizeTwoTEMP.text + "-32";
}
else if (height > 190)
{
    SizeThree.text = SizeTwo.text + "-36";
    height = 191;
    InputFieldOne.GetComponent<InputField>().text = Convert.ToString(height);
    // 36
    SizeThreeTEMP.text = SizeTwoTEMP.text + "-36";
}
}
}

```

Реалізація алгоритму обрання необхідних розмірів одягу була здійснена на підставі розмірних сіток, які відображені на рис. 4.8.

Размерные сетки

Джинсы

Размер	28	29	30	31	32	33	34	36	38	40 *
Обхват талии	73-74	75-77	78-81	82-85	86-89	90-93	94-97	98-101	102-105	106-109 **
* Длина L32	165-177	165-177	165-177	165-177	165-177	165-177	165-177	165-177	165-177	165-177 **
* Длина L34	178-185	178-185	178-185	178-185	178-185	178-185	178-185	178-185	178-185	178-185 **
* Длина L36	186-190	186-190	186-190	186-190	186-190	186-190	186-190	186-190	186-190	186-190 **

полный рост

Шорты

Размер	28	29	30	31	32	33	34	36	38	40 *
Обхват талии	73-74	75-77	78-81	82-85	86-89	90-93	94-97	98-101	102-105	106-109 **

Футболки, рубашки, свитера

Размер	XS	S	M	L	XL	XXL	3XL ***
Обхват груди	86-89	90-93	94-97	98-101	102-105	106-109	110-113 **
Обхват талии	73-77	78-81	82-85	86-89	90-94	95-100	101-109 **

*дюймы

**сантиметры

***международный размер

Рисунок 4.8 – Розмірна сітка

Тепер необхідно реалізувати зміну статури манекену на основі отриманих параметрів користувача. Нижче розглянуто існуючу модель SMPL, яка вміє це робити.

Модель SMPL - це параметрична модель, яка представляє собою метод моделювання людського тіла, запропонований Максом Планком. Цей метод може

виконувати довільне моделювання людського тіла і анімацію. Найбільша різниця між цим методом і традиційним LBS полягає в пропонованому їм методі топографії поверхні тіла із зображенням положення тіла, який може імітувати опуклості і депресії м'язів людини під час руху кінцівок. Таким чином, можна уникнути деформації поверхні людського тіла під час тренування, і можна точно описати появу розтягування і скорочення людських м'язів.

У цьому методі β і θ є вхідними параметрами, де β представляє 10 параметрів росту людини, низького зросту, товстого, худого, відносини голови до тіла, а θ - 75 параметрів, які представляють загальну рухому позу людини і відносний кут 24 суглобів.

Параметр β - це параметр ShapeBlendPose, який може управляти зміною форми людського тіла за допомогою 10 додаткових шаблонів: Зокрема: кожен параметр управляє зміною форми людського тіла і може бути описаний за допомогою анімації:

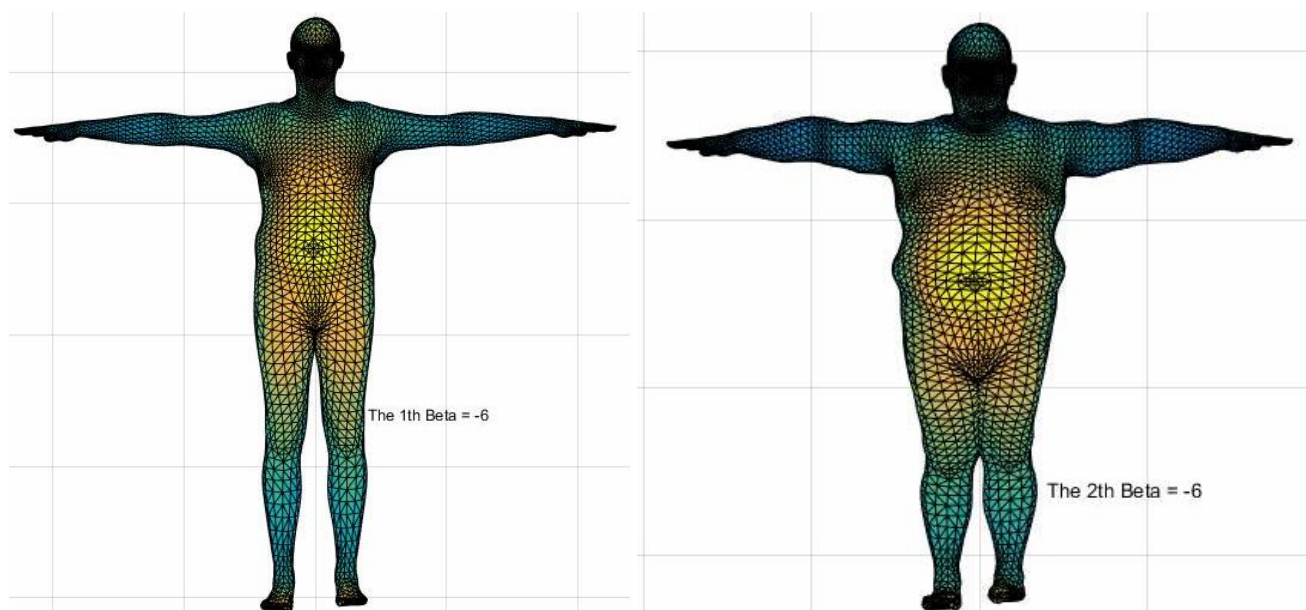


Рисунок 4.9 – Модель SMPL

Параметр PoseBlendShape дає опуклість і рецесію м'язів і положення тіла, але зазвичай алгоритм оптимізації не використовує параметр PoseBlendShape.

Таким чином було прийнято рішення розробити схожий аналог метода для зміни розмірів манекену, але з меншою кількістю вхідних даних, щоб спростити задачу.

Зміна статури манекена при різних вхідних параметрах (алгоритм наведений у лістингу 4.3 нижче).

Настройки

Введите ваш рост: 165-190

Обхват груди: 86-113

Обхват талии: 73-109

Рисунок 4.10 – Допустимі діапазони значень, що використовуються у програмі

Настройки

Введите ваш рост: 165

Обхват груди: 99

Обхват талии: 90

Подобранные размеры:

Верхняя часть тела - XL

Шорты - 33

Брюки - 33-32

Рисунок 4.11 – Мінімальний зріст, середні обхвати (з доступного діапазона значень)



Рисунок 4.12 – Зміна статури манекену

Настройки

Введите ваш рост: 190

Обхват груди: 99

Обхват талии: 90

Подобранные размеры:

Верхняя часть тела - XL

Шорты - 33

Брюки - 33-36

Рисунок 4.13 – Максимальний зріст, середні обхвати



Рисунок 4.14 – Зміна статури манекену

Настройки		Подобранные размеры:
Введите ваш рост:	<input type="text" value="178"/>	Верхняя часть тела - XS
Обхват груди:	<input type="text" value="86"/>	Шорты - 28
Обхват талии:	<input type="text" value="73"/>	Брюки - 28-34

Рисунок 4.15 – Мінімальні обхвати та середній зріст



Рисунок 4.16 – Зміна статури манекену

Настройки		Подобранные размеры:
Введите ваш рост:	178	Верхняя часть тела - 3XL
Обхват груди:	113	Шорты - 40
Обхват талии:	109	Брюки - 40-34

Рисунок 4.17 – Максимальні обхвати та середній зріст



Рисунок 4.18 – Зміна статури манекену

Лістинг 4.3 – Метод для зміни розмірів манекену

```

public void ChangeHumanParameters()
{
    height = height * 0.52325f; // 172 => 90 scale Z (or * 0.52325)
    // если объём талии будет больше объёма груди - это должно отразиться на манекене
    if (waistGirth > chestGirth)
    {
        chestGirth = waistGirth;
    }
    else
    {
        chestGirth = chestGirth + 1; // 89 => 90 scale X
    }
    //humanModelsLow.transform.localScale = new Vector3(humanModelsLow.transform.position.x,
humanModelsLow.transform.position.y, humanModelsLow.transform.position.z);
    modelHuman.transform.localScale = new Vector3(chestGirth, chestGirth, height);
    modelUPver1.transform.localScale = new Vector3(chestGirth, chestGirth, height);
    modelUPver2.transform.localScale = new Vector3(chestGirth, chestGirth, height);
    modelUPver3.transform.localScale = new Vector3(chestGirth, chestGirth, height);
    modelDOWNver1.transform.localScale = new Vector3(chestGirth, chestGirth, height);
    modelDOWNver2.transform.localScale = new Vector3(chestGirth, chestGirth, height);
    humanModelsLow.transform.localPosition = new
Vector3(humanModelsLow.transform.localPosition.x, height / 115 + 0.42f,
humanModelsLow.transform.localPosition.z); // 90 local scale Z - 1.202 local position Y
}
}

```

Після написання логіки програми маємо кінцевий результат, який представлений на рис. 4.19 - 4.24.

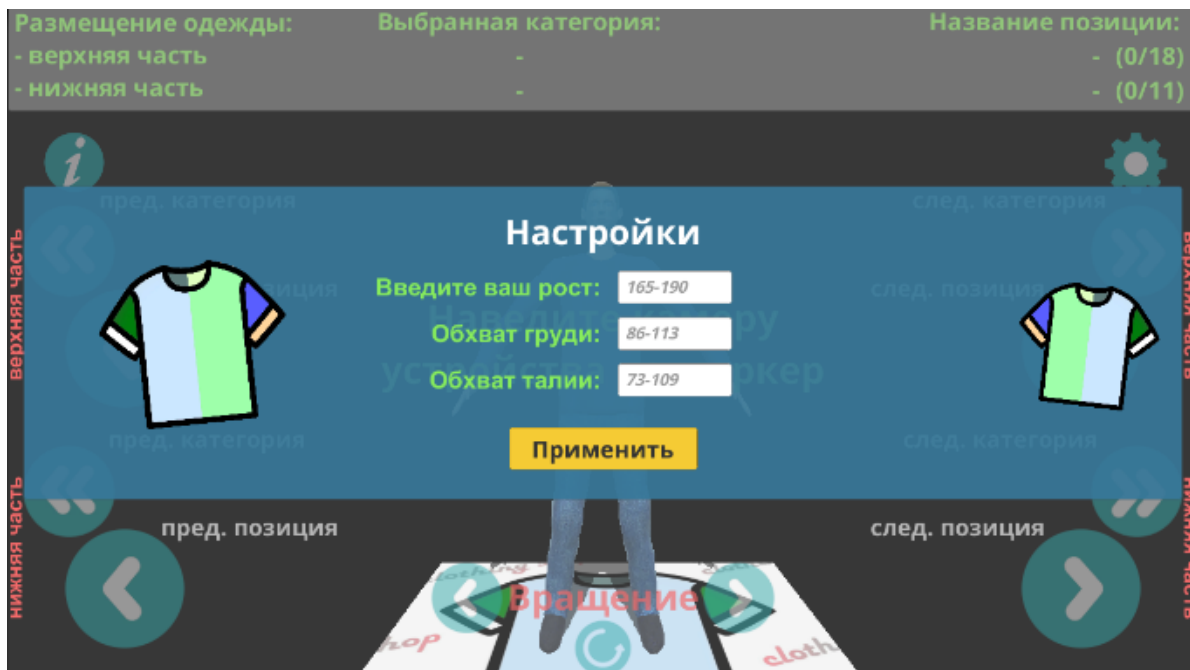


Рисунок 4.19 – Головне меню додатку

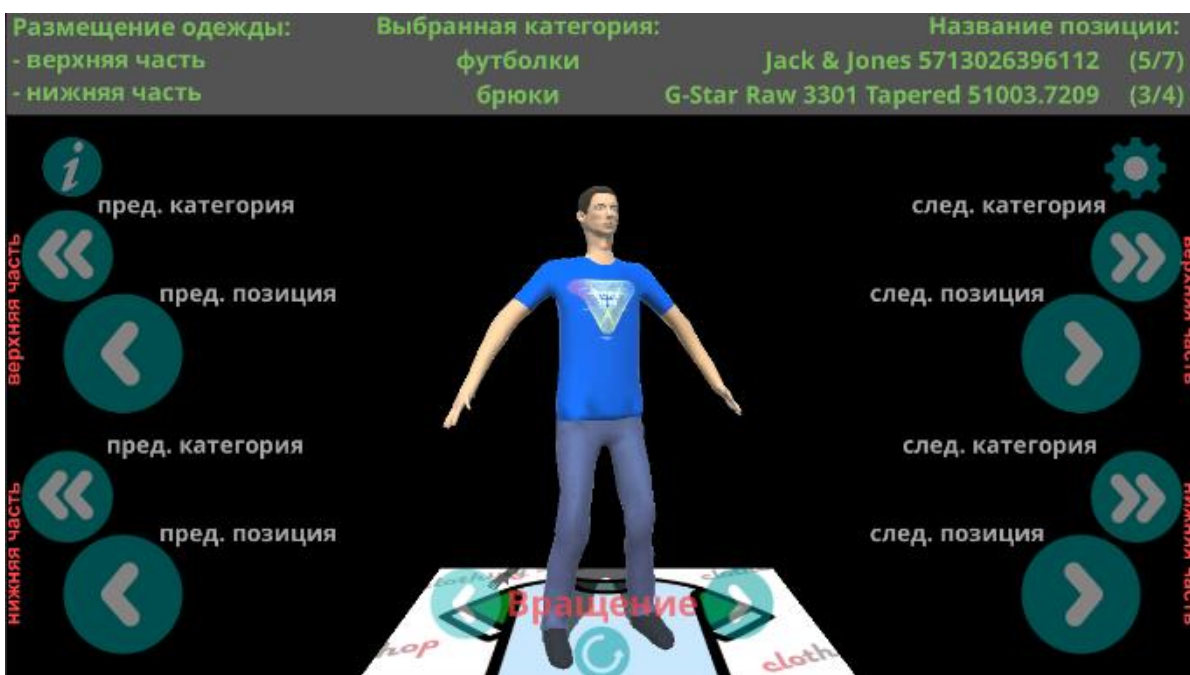


Рисунок 4.20 – Відображення моделі манекену на основі заданих параметрів

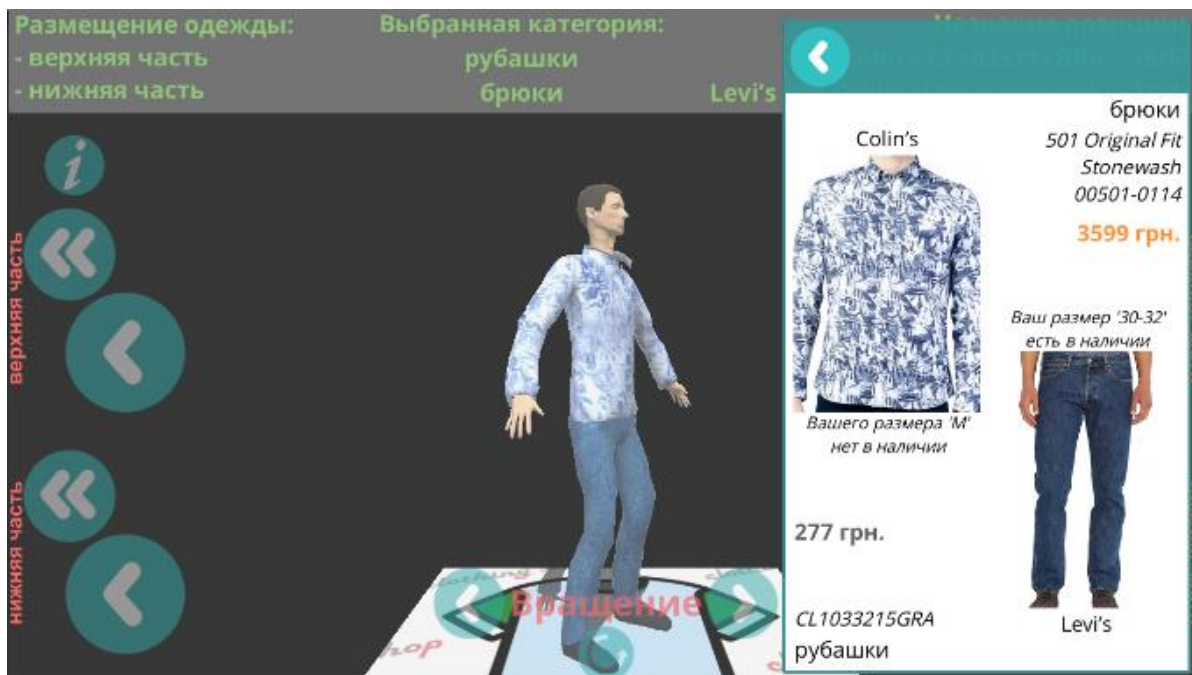


Рисунок 4.21 – Детальна інформація про обраний одяг

Також була додана можливість робити підбір образу за сезонами, які пропонуються у програмі. А саме - весна/осінь та літо. При здійсненні підбора в даній функції одяг автоматично обирається програмою для верхньої та нижньої частини тіла, який буде відповідати обраному сезону. Користувачу залишається тільки знайти образ, який йому до вподоби.

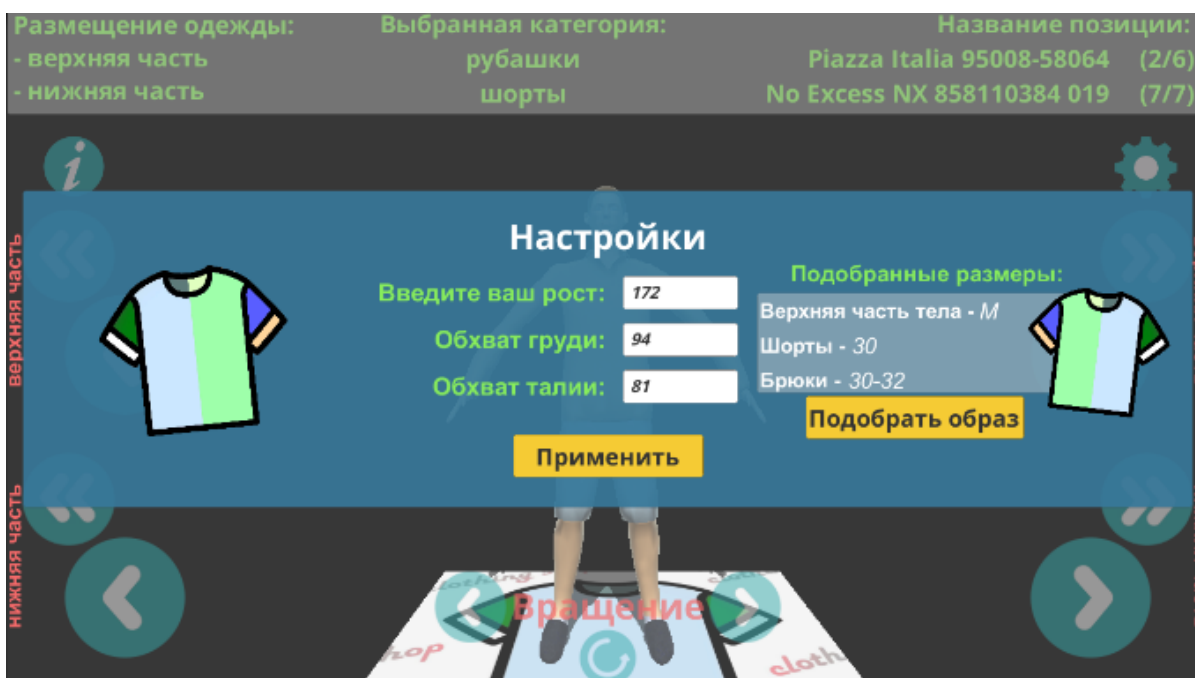


Рисунок 4.22 – Функція підбору образу

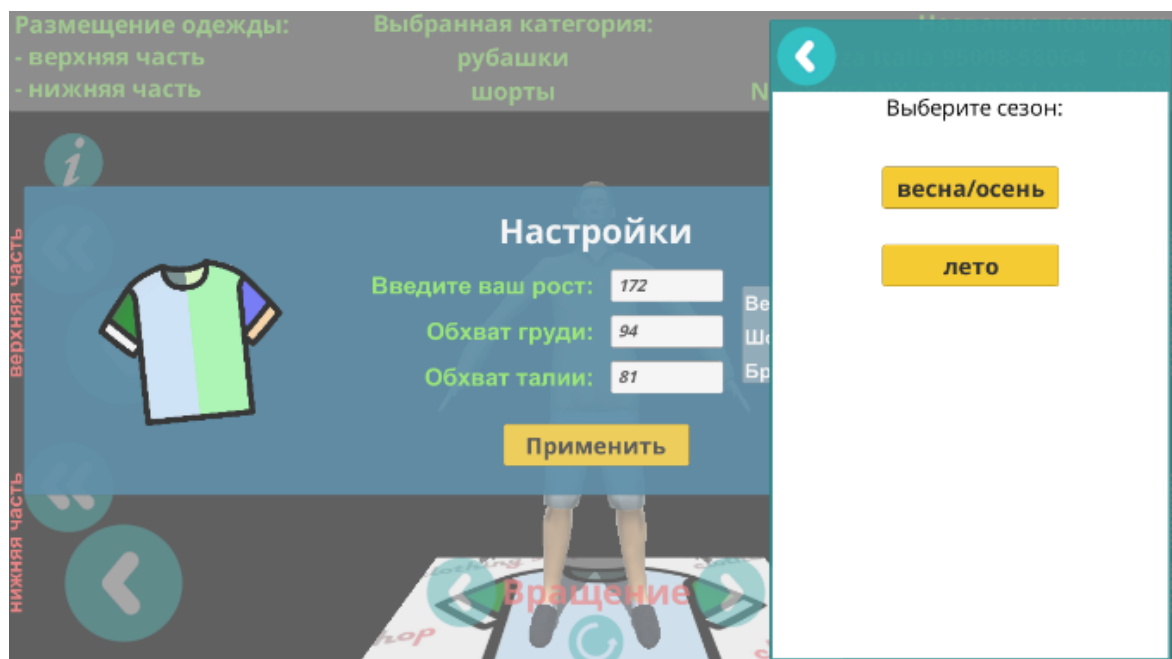


Рисунок 4.23 – Обрання сезону

Обравши сезон літо підбір образу відбувається за категоріями: футболки, сорочки, шорти, брюки.

Сезон весна/осінь включає в себе такі категорії, як: светри, сорочки, брюки.



Рисунок 4.24 – Автоматичний підбір образу

Запуск готового додатку на девайсі проілюстровано нижче (рис. 4.25 – 4.27).

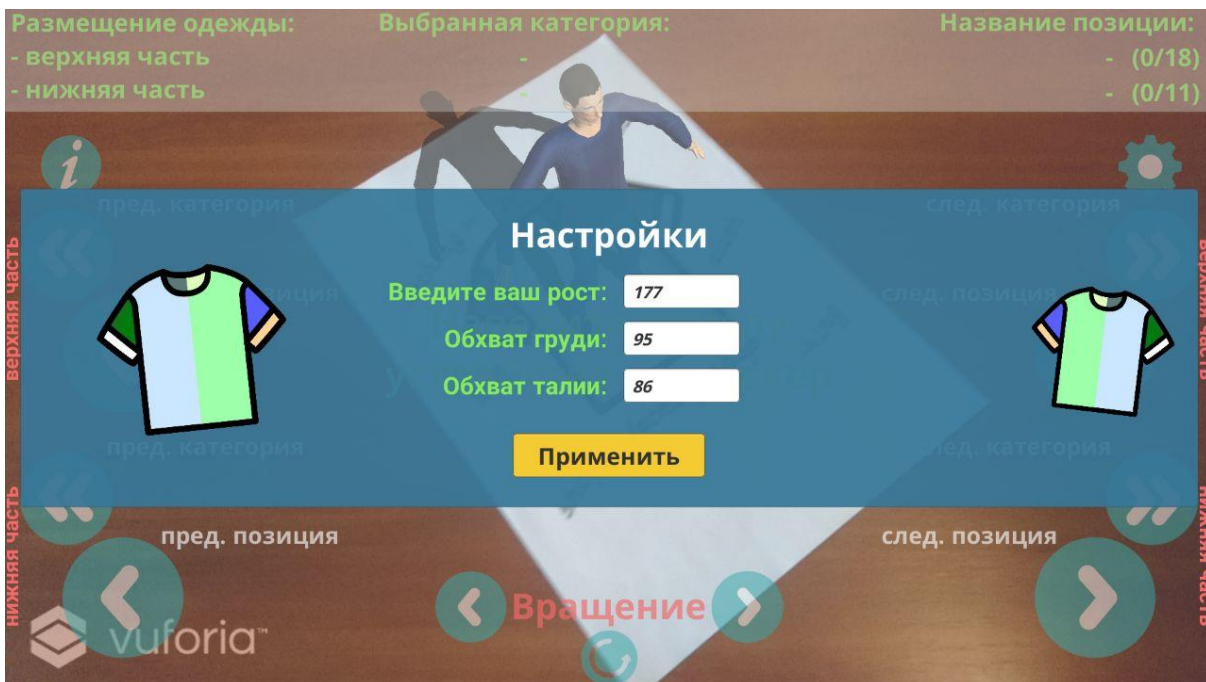


Рисунок 4.25 – Введення значень у поля



Рисунок 4.26 – Розгляд обраного одягу

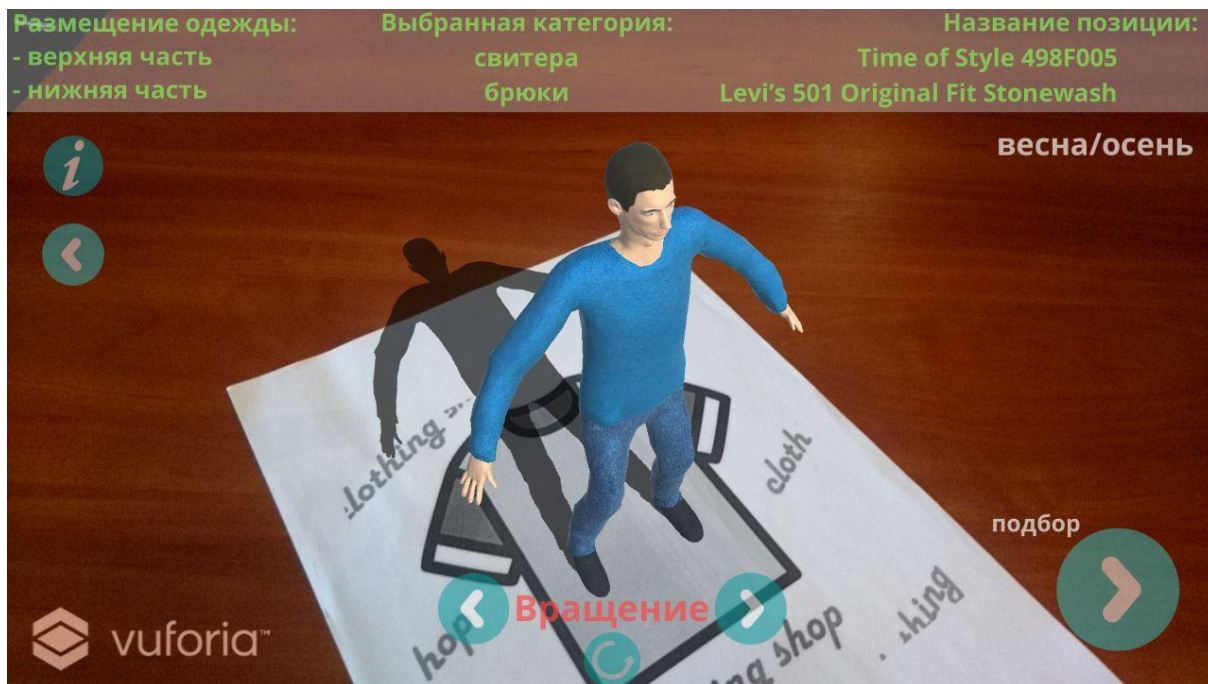


Рисунок 4.27 – Кінцевий результат

Тестування додатку проводилось на мобільному пристрої с архітектурою процесора ARMv8, та роздільною здатністю екрану 1920 на 1080.

ВИСНОВКИ

В магістерській атестаційній роботі було досліджено тему підбору одягу згідно з розмірами тіла користувача та створення унікальних образів.

Даний мобільний додаток використовується для пошуку необхідного одягу для чоловіків із позицій, які пропонуються та отримання про нього інформації під платформу Android.

Завдяки цьому додатку ми можемо уявити як виглядає одяг на людині певного розміру. Дана задача була виконана за допомогою камери пристрою та технології доповненої реальності.

Для того, щоб скористатися програмою необхідно знати параметри свого тіла, які потрібно буде ввести в відповідні поля. Це такі параметри як зріст, обхват талії та обхват грудей. Після цього відобразиться манекен, який буде відповідати цим значенням.

Перевага цього додатку полягає в тому, що використовуючи свій гаджет, ви відтворюєте відчуття перебування, як у магазині. Не виходячи з дому дізнаєтеся інформацію про одяг та роздивляєтесь його з різних боків.

Наступною важливою особливістю є те, що програма підбирає потрібний розмір одягу, при умові, що покупець вірно додасть свої розміри тіла. При відображенні додаткової інформації про обрану позицію одягу програма сама підказує користувачеві, чи є його розмір у наявності.

Програма дозволяє створити та запропонувати покупцеві унікальний образ для таких сезонів, як весна-осінь та літо.

Також є реалістична модель манекену, яка підлаштовується під введені дані зросту, талії та обхвату грудей.

Цільовою аудиторією є чоловіки, тому що одяг, який розташовано у цьому додатку, орієнтовано на чоловічу половину населення. Це люди, які зацікавлені у покупці одягу і є активними користувачами мобільних додатків.

Також були проаналізовані аналоги додатку на ринку. На основі цих досліджень було визначено функціонал для даного додатку.

Підсумовуючи треба зазначити, що на ринку є не багато додатків-аналогів. Розроблені програмні компоненти за своїм функціоналом мають суттєві відмінності від інших продуктів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Велингтон, Т. Розробка додатків Т. Веллинг. – СПб.: ДиаСофтЮП, 2003. – 672 с.
2. Мезенцев К. Автоматизовані інформаційні системи – М.: Academica, 2010. – 176 с.
3. Михелёв В.М. Бази даних і СУБД – Белгород: Издательство БелГУ, 2007. – 200 с.
4. Борри Х. Керівництво розробника баз даних изд. – БХВ.: Петербург 2006. – 1105 с.
5. Erwin. CASE-засоби для розробки інформаційних систем [Текст] / С. Маклаков. – М.: Диалог-МИФИ, 2000. – 256 с.
6. IDEF0: функціональне моделювання процесів [Електронний ресурс] – Режим доступу до ресурсу: http://www.info-system.ru/designing/methodology/sadt/sadt_for_bp.html.
7. The Complete Guide to UML Diagram Types with Examples [Електронний ресурс] – Режим доступу до ресурсу: <http://creately.com/blog/diagrams/uml-diagram-types-examples/#Activity-Diagram>.
8. ДСТУ 3008–95. Документація. Звіти в сфері науки і техніки. Структура і правила оформлення. [Текст] – К.: Держстандарт України, 1995. – 36 с.
9. Загальні методичні вказівки з дипломного проектування в університеті Упоряд.: Ковтун П.С., Дудар З.В., Журавльов В.Я., Шкіль О.С. – Харків: ХНУРЕ, 2003. – 40 с.
10. Галузевий стандарт вищої освіти України за напрямом підготовки 6.050101 «Комп’ютерні науки»: Збірник нормативних документів вищої освіти. – К.: Видавнича група ВНУ, 2011. – 85 с.
11. Закон України «Про доступ до публічної інформації». Відомості Верховної Ради України, 2011, № 32, ст.314 Режим доступу: [www / URL: http://zakon2.rada.gov.ua/laws/show/2939-17](http://zakon2.rada.gov.ua/laws/show/2939-17). – Загол. з екрану.

12. Nechyporenko A., Reshetnik V., Shyian D., Alekseeva D., Radutny R. and Gargin V. Solutions to the 3D Model Problem of Pressure Measurement in the Area of Maxillary Sinus Anastomosis CEUR Workshop Proceedings, 2020, 2753, pp. 275–284.

13. Надточій О.Ю. Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Конференція «Інформаційні інтелектуальні системи» : Міжнар. молодіжний форум Харків, 2021: Том 6 С. 282-283.