МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационные системы и технологии

Специальность 1-98 01 03 “Программное обеспечение информационной

безопасности мобильных систем”

Специализация Инженер-программист

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«Мобильное приложение “Wishlist”»

Выполнил студент Ковалевич Алина Сергеевна

(Ф.И.О.)

Руководитель проекта асс. Уласевич Н. И. (учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой к.т.н., ст. преп. Блинова Е. А. (учен. степень, звание, должность, Ф.И.О., подпись)

Консультант: асс. Уласевич Н. И. (учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2024

**Содержание**

[Содержание 2](#_Toc167101852)

[Введение 4](#_Toc167101853)

[1. Анализ прототипов и формирование требований к проектируемому мобильному приложению. 5](#_Toc167101854)

[1.1 Анализ прототипов 5](#_Toc167101855)

[1.2 Функциональные требования 10](#_Toc167101856)

[1.3 Вывод по разделу 10](#_Toc167101857)

[2. Проектирование архитектуры проекта 11](#_Toc167101858)

[2.1. Проектирование схемы БД 11](#_Toc167101859)

[2.2. Построение диаграммы использования 12](#_Toc167101860)

[2.3. Выбор цветового решения 12](#_Toc167101861)

[3. Программная реализация приложений 13](#_Toc167101863)

[3.1Технические средства для разработки 13](#_Toc167101864)

[3.2 Разработка мобильного приложения 13](#_Toc167101865)

[3.3 Вывод по разделу 14](#_Toc167101866)

[4. Тестирование приложения 15](#_Toc167101867)

[Заключение 19](#_Toc167101872)

[Список использованных источников 20](#_Toc167101873)

**Введение**

Все люди любят праздники. В течении года их предостаточно: начиная днём рождения и заканчивая корпоративами на работе. И вне зависимости от того, любит человек больше получать подарки или же дарить, часто возникает вопрос «Что подарить?» или «Что я хочу себе в подарок?». Хотя на первый взгляд кажется, что задача проста, при возникновении такой ситуации в реальной жизни человек попадает в ступор. Существует множество аспектов, которые влияют на выбор подарка: отношения между дарящими (коллеги, друзья, семья), повод для подарка (какой праздник) и, конечно же, цена.

Целью данного проекта является создание интуитивно понятного и функционального мобильного приложения «Wishlist», которое станет незаменимым помощником в этом деле, ведь благодаря ему всего лишь обменявшись ссылками на свои аккаунты люди смогут подобрать идеальный подарок друг другу.

В данном документе будут рассмотрены основные этапы проектирования и разработки приложения, начиная с анализа аналогов и заканчивая тестированием. В качестве основы для разработки выбран язык dart (flutter). Процесс создания программы включает в себя применение концепций ООП, интеграцию с базой данных Firebase.

1. **Анализ прототипов и формирование требований к проектируемому мобильному приложению.**

Формулировка задачи и анализ существующих решений — ключевые этапы в процессе выполнения любой работы. Эти этапы способствуют: ясности в установке целей, минимизации повторения уже существующих инициатив и ошибок, а также повышению эффективности и качества конечного продукта.

* 1. **Анализ прототипов**

**Wishlists** [1] представляет собой мобильное приложение с эстетичным дизайном, где доминируют пастельные оттенки, которые не напрягают глаза. В нижней части страницы расположено меню с основными разделами (сами списки желаний, друзья, кнопка для добавления желаний и списков, уведомления и настройки). Дизайн сайта направлен на удобство пользователя, благодаря крупным элементам, обтекаемым формам и плавным переходам.

При входе в приложение необходимо пройти регистрацию с помощью привязки электронной почты. Данное действие является обязательным в этом приложении. На рисунке 1.1 показана главная страница с активной кнопкой добавления.

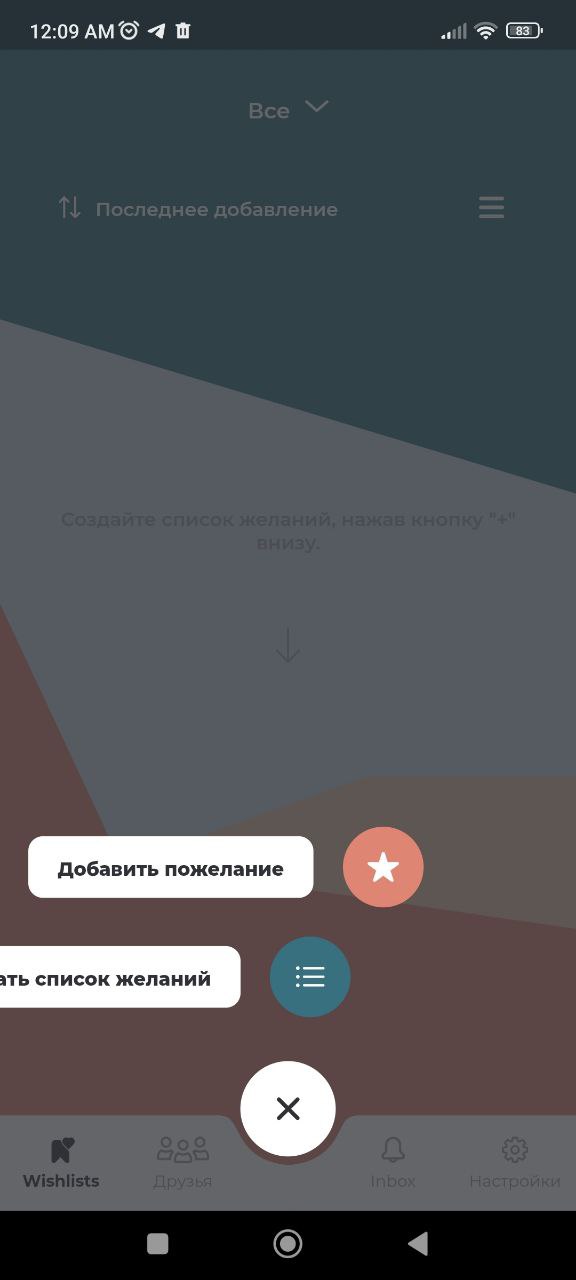


Рисунок 1.1 — Главная страница приложения «Wishlists»

Как видно на картинке выше, есть недоработка в плане отображения кнопки, часть которой выходит за рамки видимости. На рисунке 1.2 показаны варианты добавления желаний. В данном приложении их целых три, что явно дает преимущество этому приложению.

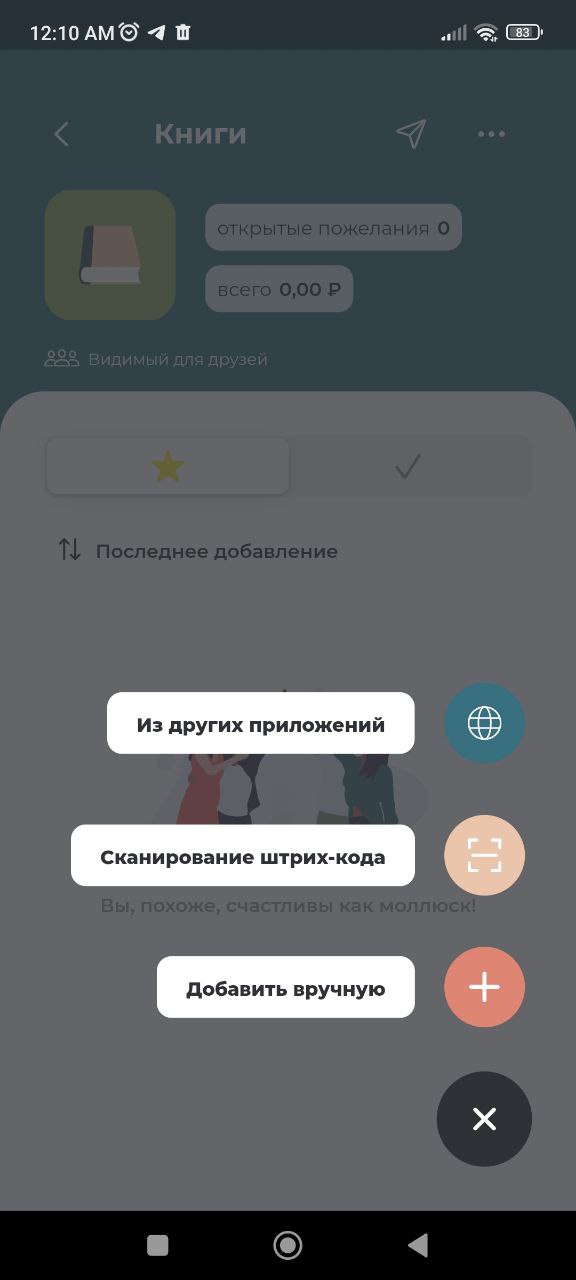


Рисунок 1.2 — Варианты добавления желаний в приложении «Wishlists»

Добавление желания реализовано неудобно (рисунок 1.3), особенно это касается цены. К тому же отсутствует выбор валюты. Разделы с примерами и рекомендациями подарков отсутствуют в данном приложении.

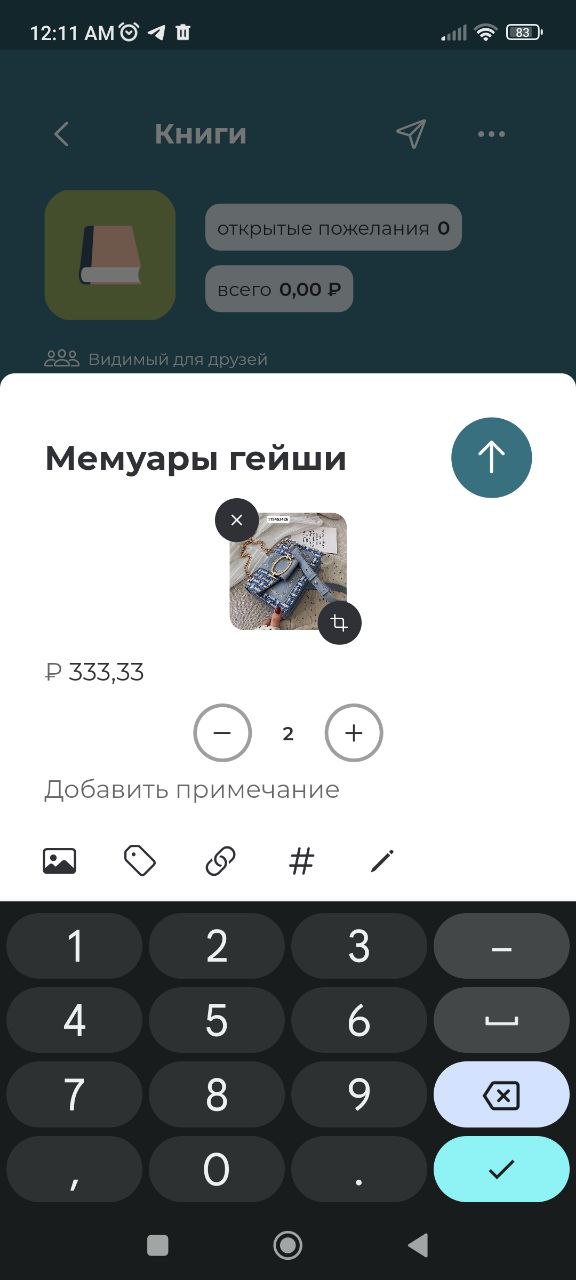


Рисунок 1.3 — Добавление желания в приложении «Wishlists»

Главным минусов приложения является то, что можно создать только четыре бесплатных списка. Создание большего количества списков желаний доступно в премиум версии приложения (рисунок 1.4).

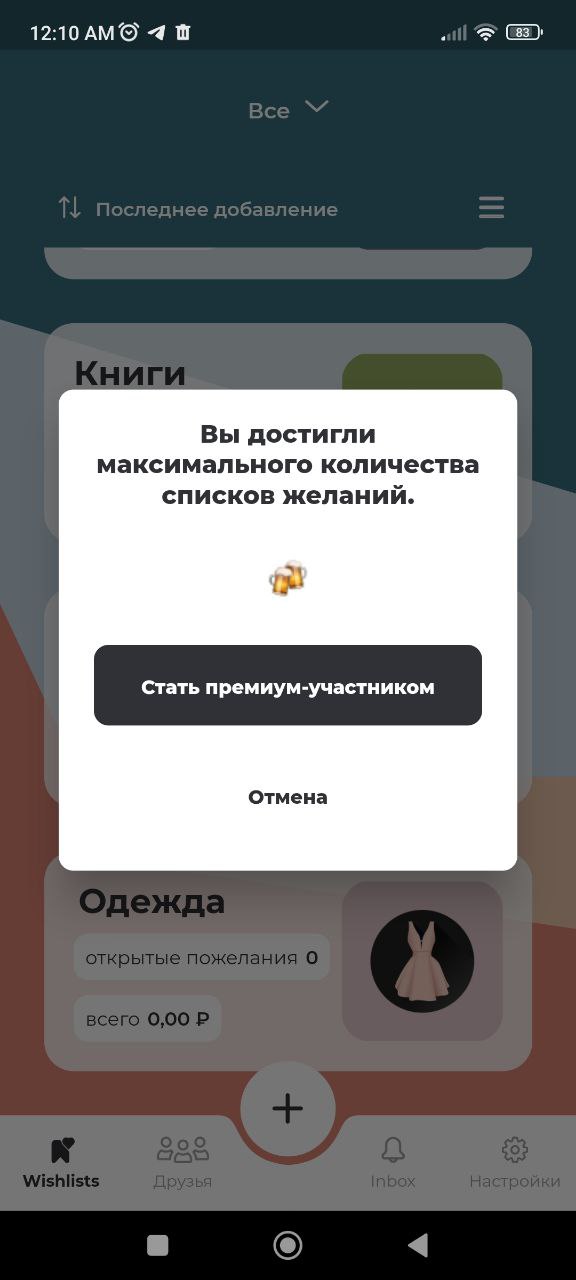


Рисунок 1.4 — Предложение о покупке премиум версии приложения

Таким образом, приложение «Wishlists» имеет приятный и комфортный для пользования дизайн, но плохо проработанный и ограниченный функционал.

**Подаркус** [2] предлагает своим пользователям интерактивный дизайн, создающий приятное первое впечатление. Благодаря необычному началу работы с приложением, пользователей легко заинтересовать. А именно речь идет о выборе праздников и категорий интересующих подарков. Всё это можно увидеть на рисунке 1.5.

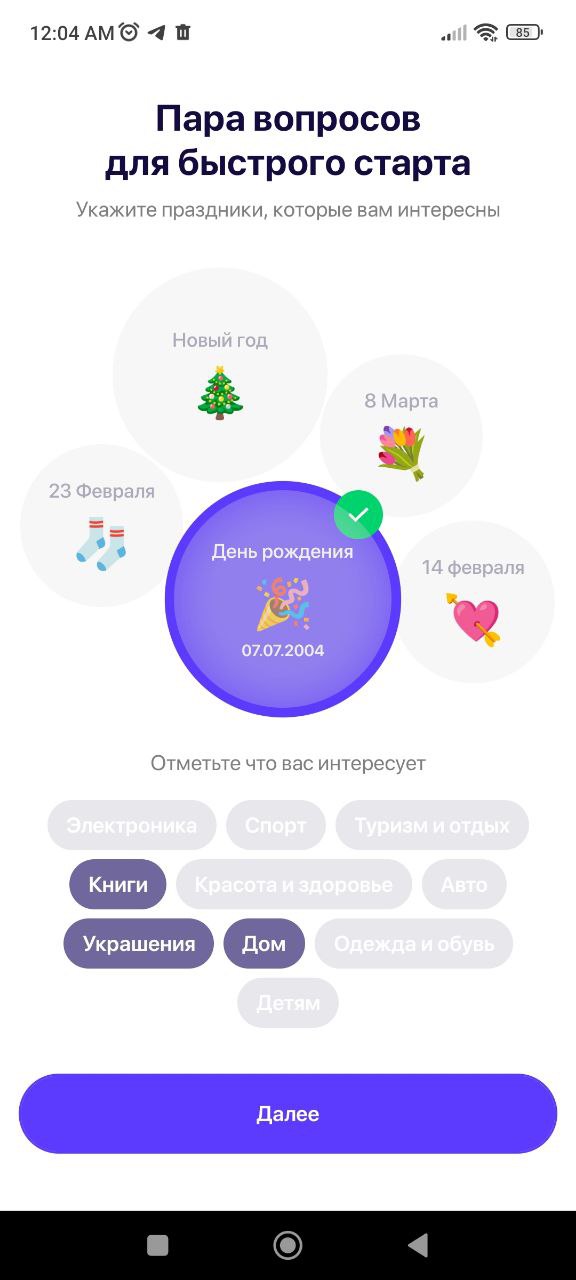


Рисунок 1.5 — Начало работы в приложении «Подаркус»

Зарегистрироваться в приложении можно с помощью уже существующих аккаунтов в WK, Одноклассниках, Яндексе и Google(рисунок 1.6). Такое большое количество вариантов авторизации несомненно является преимуществом для приложения.

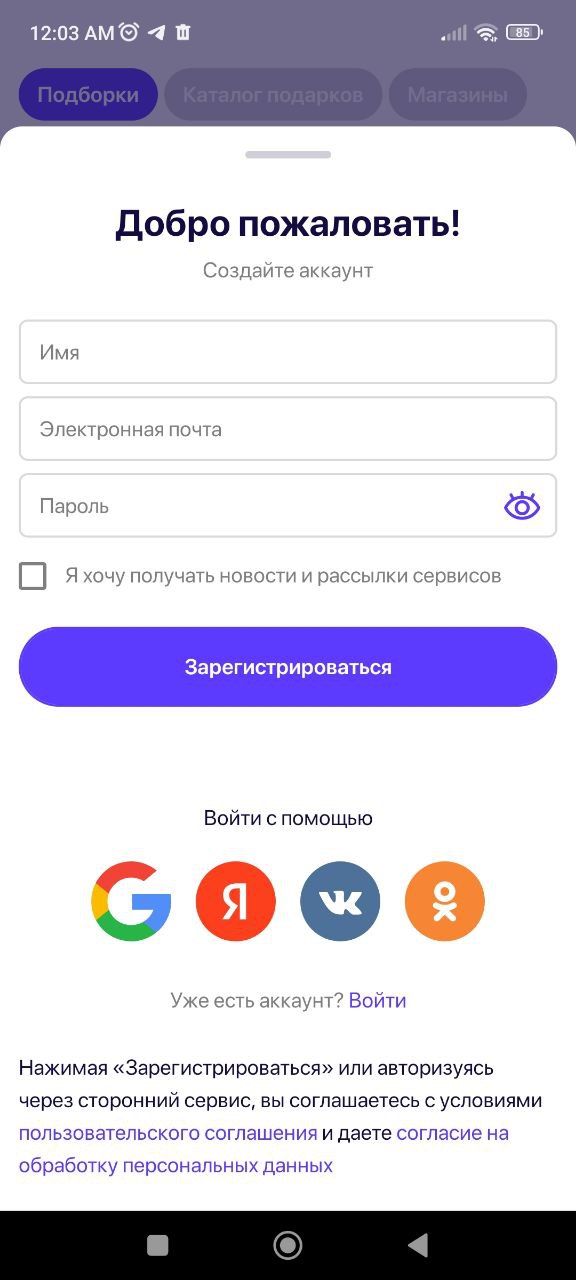


Рисунок 1.6 — Варианты авторизации в приложении «Подаркус»

Недостаток, который был выявлен в ходе пользования данным приложением, это отсутствие главной функции приложения – нет возможности добавить собственное желание (рисунок 1.7). Можно добавить в список желаний только что-то из предложенного в каталоге. Но хотелось бы отметить, что в приложении существую неплохие подборки похожих идей для подарков с достаточно детализированным описанием.

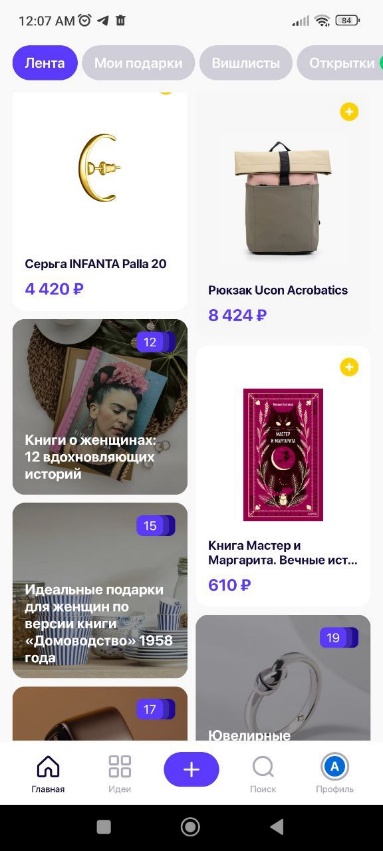


Рисунок 1.7 — Подборки и примеры желаний в приложении «Подаркус»

Следовательно, это приложение подойдёт для поиска подарка или идеи для кого-то, но собственные желания оформить не получится. Возможно, приложение разработано для определенной локации (Россия, Москва).

**Ohmywishes** [3] – мобильное приложение с комфортным и минималистическим дизайном, которое позволяет как делать подборки из предложенных идей подарков (раздел «Идеи»), так и создавать собственные желания (рисунок 1.8).

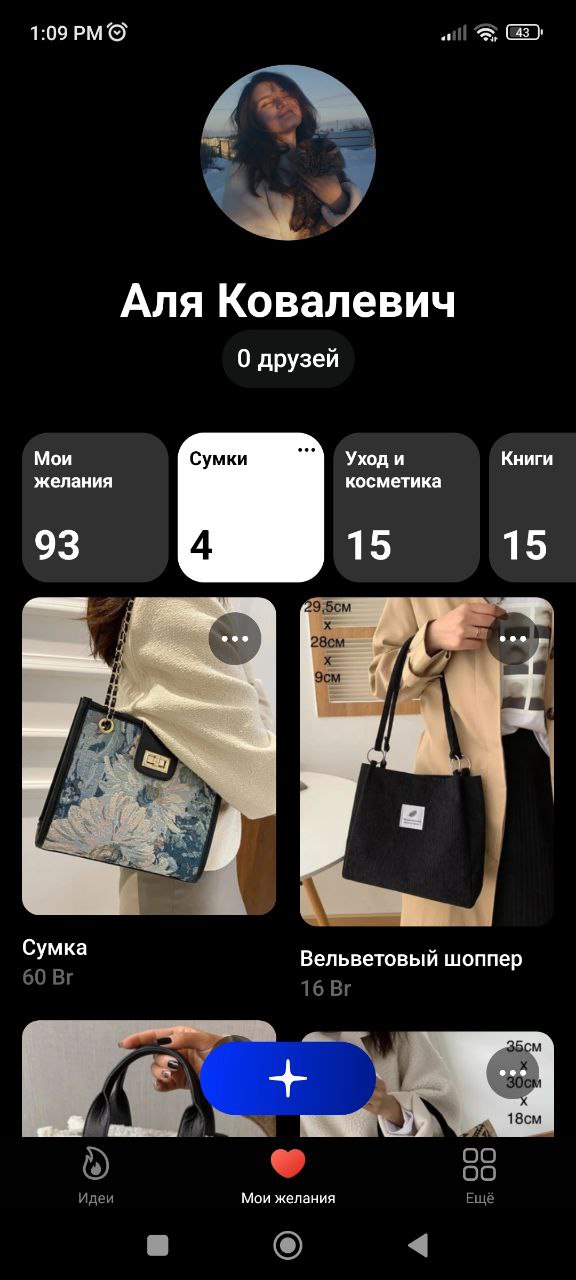


Рисунок 1.8 — Главная страница в приложении «Ohmywishes»

Элементы интерфейса крупные, шрифт достаточно большой и лаконичный. Изображения желаний в зависимости от загруженного размера и ориентации интересно размещаются на странице в 2 колонки. Количество списков не ограничено, более того, есть дополнительные списки, например, список «Исполнено» (пример на рисунке 1.9). В других приложениях исполненные желания просто удалялись.

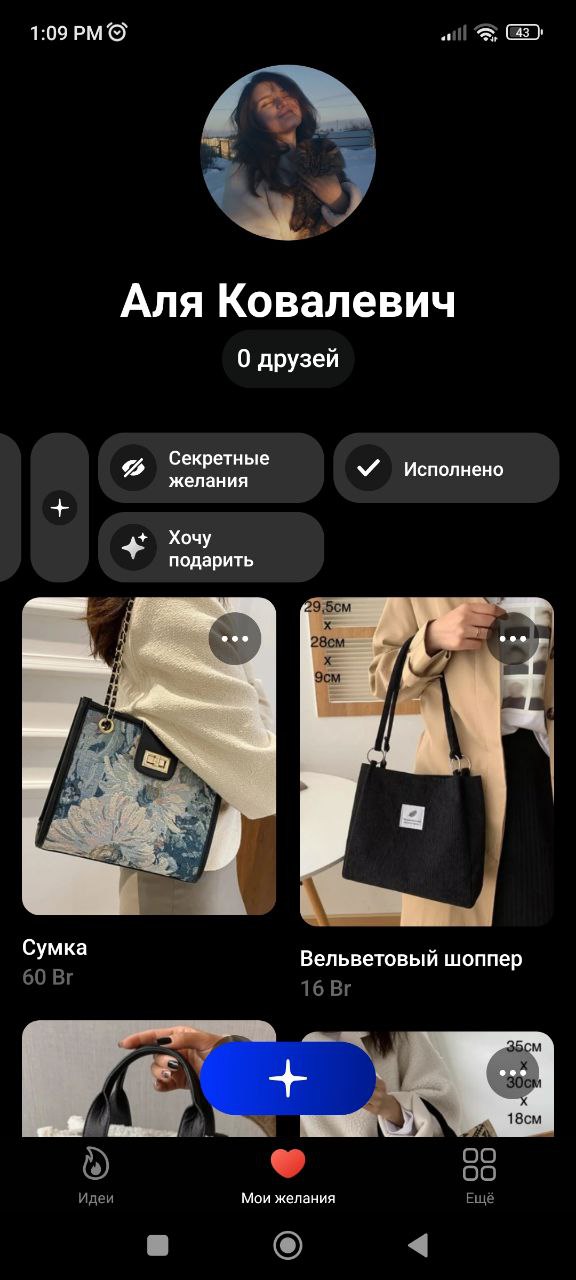


Рисунок 1.9 — Дефолтные списки в приложении «Ohmywishes»

В этом приложении просторная форма для оформления желаний (рисунок 1.10). Здесь вы можете указать название, цену (с выбором валюты), ссылку, описание и указать, к какому списку вы относите это желание.

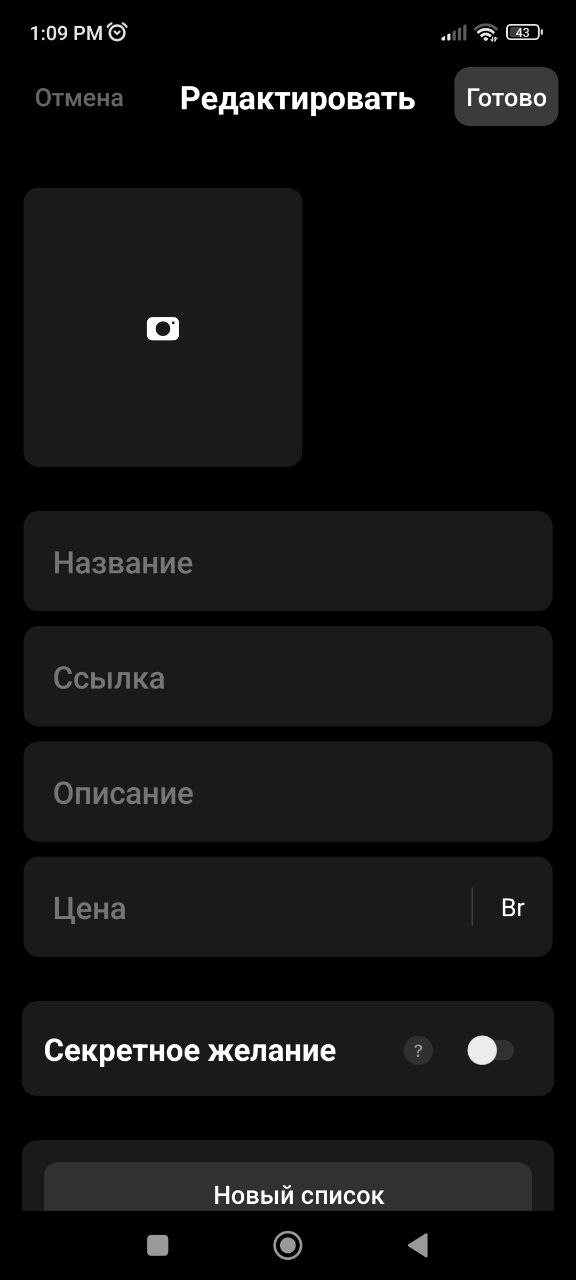


Рисунок 1.10 — Создание желания в приложении «Ohmywishes»

Таким образом, приложение «Ohmywishes» имеет значительное преимущество перед предыдущими аналогами как в решении дизайна, так и в плане функционала и содержания.

**1.2 Функциональные требования**

Исследование рассмотренных выше популярных аналогов дает возможность оценить сильные и слабые стороны существующих решений. Это, в свою очередь, позволяет определить ключевые требования к программному продукту, который разрабатывается в рамках курсовой работы.

Мобильное приложение должно обеспечивать возможность выполнения перечисленных ниже функций:

* Регистрация и авторизация пользователей;
* создание желаний;
* редактирование желаний;
* удаление желаний;
* отображение исполненных желаний в отдельном списке;
* возможность отменить исполненное желание.

**1.3 Вывод по разделу**

Анализируя аналоги данных приложений, были выделены ключевые направления для курсового проекта: создание приложения, которое предоставляет возможность создания собственных списков желаний. Должна быть предусмотрена возможность создания списков и желаний, их редактирование. Интерфейс и дизайн приложения должны быть комфортными, понятными и максимально приятными для пользователей.

**2. Проектирование архитектуры проекта**

Архитектурное проектирование — это ключевой этап в создании любого программного продукта. В этом разделе приведены инструменты, выбранные для реализации проекта, представлена структура базы данных, перечислены интерфейсы, которые будут присутствовать в приложении.

**2.1. Проектирование схемы БД**

Для оптимизированного сохранения и управления информацией приложения создана концептуальная модель базы данных.

На основе коллекций в Firebase была разработана логическая схема (Рисунок 2.1).

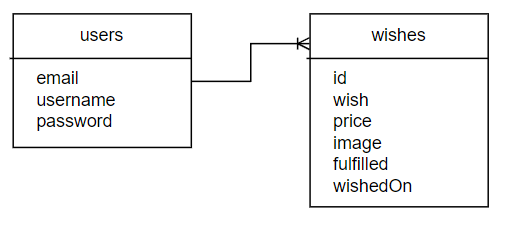


Рисунок 2.1 — Логическая схема БД

Данное количество таблиц является достаточным для реализации задуманного функционала. Здесь есть вся необходимая информация для входа: почта, имя и пароль; Также есть поля, которые содержат информацию о самом желании: название, цена, картинка и чек отображаются на карточке с желанием.

Ниже представлены типы данных полей каждой коллекции:

Коллекция «users»:

* email – string;
* name – string;
* password – string.

Коллекция «wishes»:

* id – string;
* wish – string;
* image – string;
* price – integer;
* fulfilled – boolean;
* wishedOn – timestamp.

Благодаря полю с чекбоксом можно отображать желания в разных списках. Подробнее об это в следующих разделах.

**2.2. Построение диаграммы использования**

Помимо этого, хотелось бы рассмотреть возможности пользователя в приложении. Наглядно это показано на рисунке 2.2.

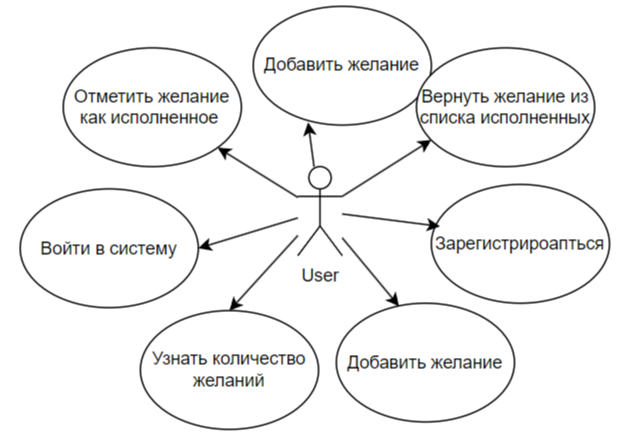


Рисунок 2.2 — Диаграмма использования

Как видно на диаграмме, приложение имеет достаточно широкий спектр возможностей. Но есть и место для творчества в будущем: создать подборки по желаниям, добавить кошелек, добавить возможность обмениваться желаниями с друзьями и пополнять баланс своих друзей.

**2.3. Выбор цветового решения**

Так как приложение связано с такими чувствами как мечтательность, радость, предвкушение, то разумным будет выбрать мягкие светлые оттенки.

Но чтобы не напустить на приложение лишнего романтизма было принято решение в пользу палитры, продемонстрированной на рисунке 2.3.

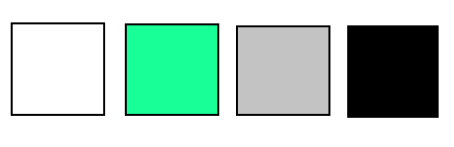


Рисунок 2.3 — Цветовое решение приложения

Зелёный цвет ассоциируется со свежестью и лёгкостью, но в то же время и с деньгами и технологиями. Поэтому логично выбрать этот цвет акцентным. К тому же, мужчины и женщины относятся к зелёному цвету нейтрально.

# **3. Программная реализация приложений**

# 3.1 Технические средства для разработки

Для разработки приложения были использованы следующие средства:

* Visual Studio Code [4]– универсальная среда разработки в том числе и для Android приложений, с хорошим статическим анализатором кода;
* Flutter [5] – это UI фреймворк, разработанный компанией Google, который позволяет создавать кроссплатформенные мобильные приложения.
* Dart [6] – это язык программирования, разработанный компанией Google, который используется для создания мобильных, веб- и настольных приложений.
* Firebase [7] - это платформа Google, помогающая быстро разрабатывать качественные мобильные и веб-приложения, привлекать новых пользователей и повышать доходы. Ее интегрированные инструменты, такие как сервер для мобильных приложений, средства аналитики, инструменты для продвижения приложений и монетизации, можно использовать по отдельности или в любых сочетаниях.

Почти все вышеперечисленные средства для разработки являются современными, популярными и хорошо совместимыми между собой, поэтому выбор и пал на них.

## Разработка мобильного приложения

Мобильное приложение построено с использованием нескольких папок, каждая из которых выполняет определенную функцию и обеспечивает структурированное и логичное разделение кода.

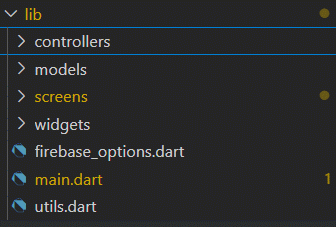


Рисунок 3.1 — Основные папки приложения

Controllers включает классы auth\_controller.dart и home\_controller.dart, которые описывают данные и бизнес-логику приложения. Здесь находятся классы моделей данных, отображающие структуру данных приложения, и логика обработки этих данных.

Models содержит только один класс wish\_model.dart, который Отвечает за структура желаний.

Screens представляет классы auth\_screen.dart и home\_screen.dart, используемые во всем приложении и содержащие основные страницы, на которых отображаются все элементы.

Widgets отвечает за взаимодействие с пользовательским интерфейсом и состоит из login\_widget.dart и register\_widget.dart. Первый класс отвечает за элементы при авторизации, а второй – во время регистрации. Еще здесь находиться класс wish\_item.dart, который отвечает за всплывающее окно при добавлении желания (рисунок 3.2)

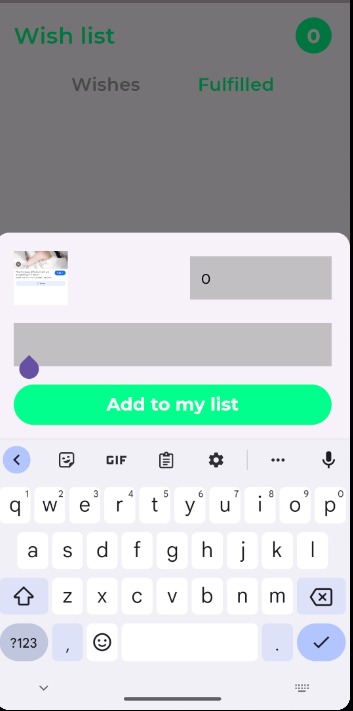


Рисунок 3.2 — Всплывающее окно для добавления желания

Реализация функции, которая отвечает за стилевое отображение текста представлена в листинге 3.1. Эта функция находится в классе uitls.dart

TextStyle textStyle(double size, Color color, FontWeight fw) {

  return GoogleFonts.montserrat(fontSize: size, color: color, fontWeight: fw);

}

Листинг 3.1 – Функция для стилей текста

Есть ещё два важных файла в папке lib: main.dart и firebase\_options.dart. Первый файл отвечает за запуск приложения и отображение стартовых страниц. Второй содержит в себе настройки с сервером Firebase.

## Вывод по разделу

В данной главе была рассмотрена основная структура проекта, стек технологий и среда разработки. На данном этапе была совершена самая трудная и важная работа: настройка и установка необходимых инструментов.

# **4.** Тестирование приложения

Тестирование мобильных приложений важно и необходимо для обеспечения их качества и производительности. Оно помогает уменьшить риск сбоев приложения, гарантирует отсутствие ошибок и улучшает восприятие пользователей, что в свою очередь способствует увеличению количества загрузок и успешному запуску приложения на рынке.

Для начала протестируем страницу авторизации. Если пользователь не зарегистрирован, ввел неверные данные или не ввел какое-либо поле, то он не сможет попасть в свой аккаунт, показанное на рисунке 4.1.

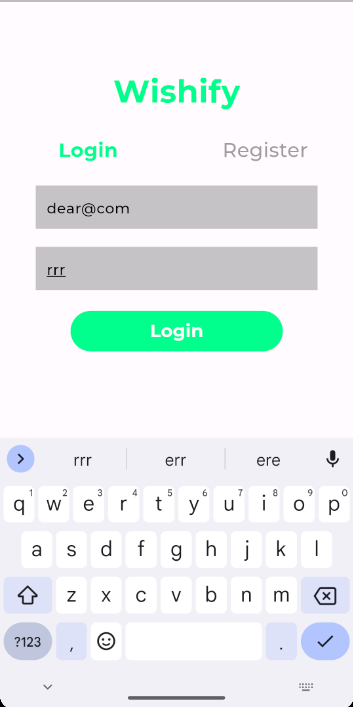


Рисунок 4.1 – Тестирование авторизации

Далее протестируем страницу регистрации. Пользователь может зарегистрироваться с уникальным email и именем, также ему необходимо придумать пароль, который содержит не менее 6 символов. Роль пользователя автоматически считается User. В случае успешной регистрации пользователя переносит на главную страницу. Тестирование продемонстрировано на рисунке 4.2.

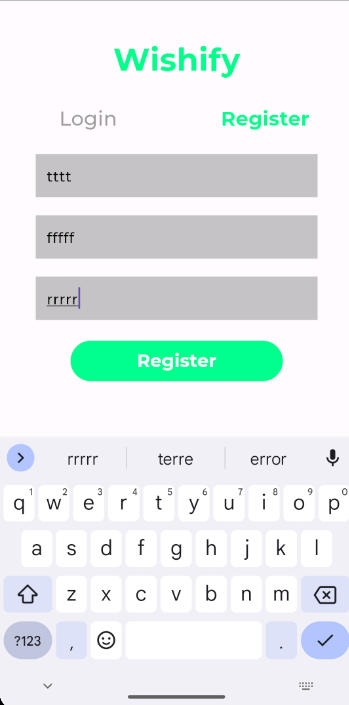
 

Рисунок 4.2 – Тестирование регистрации

После авторизации или регистрации пользователь попадает на главную страницу с активным списком желаний. Его можно легко переключить на список исполненных желаний. Вверху страницы находится счетчик желаний. Тестирование продемонстрировано на рисунке 4.3.

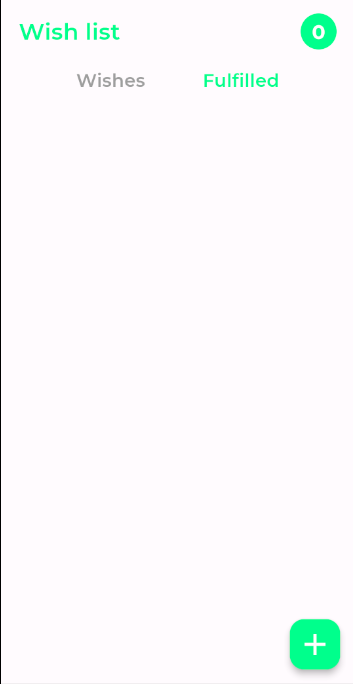
 

Рисунок 4.3 – Тестирование перехода между списками

Теперь перейдём к созданию нового желания.

При добавлении нового желания следует соблюдать типы данных. При нарушении этого правила ничего не добавится.

Если случайно скрыть окно с формой заполнения до того, как добавить желание, то последняя заполненная информация сохраниться. В случае успешного добавления появляется окно с сообщением. Продемонстрировано на рисунке 4.4.

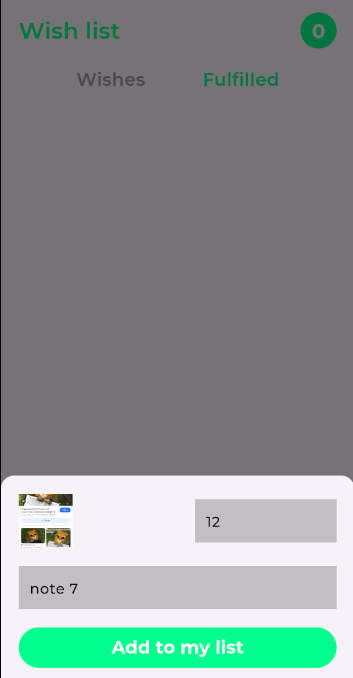
 



Рисунок 4.4 – Тестирование добавления нового желания

# Следующий этап тестирования – изменение статуса желания на исполненное. Меняется счетчик, место отображения карточки и чекбокс. Продемонстрировано на рисунке 4.5

# 

Рисунок 4.5 – Тестирование изменения статуса желания

# Рассмотрим ещё возращение желания в статус не исполненного и удаление желания. Продемонстрировано на рисунке 4.6.

# 

Рисунок 4.5 – Тестирование изменения статуса желания обратно и удаление карточки с желанием

Чтобы удалить желание, нужно дважды кликнуть на карточку.

Таким образом был протестирован функционал приложения. Есть над чем работать: не хватает всплывающих окон с информацией о валидации. Их отсутствие усложняет понимание работы приложения.

.

# Заключение

В результате выполнения данной курсовой работы было разработано мобильное приложение, которое помогает создавать список желаний.

Разработанное программное средство предоставляет пользователю следующие функциональные возможности:

* регистрация пользователей;
* авторизация пользователей;
* добавление желаний в приложение;
* просмотр списков исполненных и не исполненных желаний;
* перемещение желаний между списками;
* просмотр количества желаний в списках;
* удаление желаний.

Проработана концепция приложения и бизнес-логика продукта, в том числе цветовое решение.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме. Но тем не менее есть варианты для развития приложения дальше.

**Список использованных источников**

1 Wishlists [Электронный ресурс]. Режим доступа: [https://www.wishlists-app.com](https://www.wishlists-app.com/). Дата доступа: 23.02.2024.

2 Подаркус [Электронный ресурс]. Режим доступа: https://podarkus.ru/#catalog. Дата доступа: 25.02.2024.

3 Ohmywishes [Электронный ресурс]. Режим доступа: https://ohmywishes.ru. Дата доступа: 17.01.2024.

4 Visual Studio Code [Электронный ресурс]. Режим доступа: https://code.visualstudio.com. Дата доступа: 17.03.2024.

5 Flutter [Электронный ресурс]. Режим доступа: https://flutter.dev/learn Дата доступа: 07.04.2024.

6 Dart [Электронный ресурс]. Режим доступа: https://medium.com/dartlang. Дата доступа: 07.04.2024.

7 Firebase [Электронный ресурс]. Режим доступа: https://console.firebase.google.com/u/0/. Дата доступа: 10.05.2024.