

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук
Кафедра программирования и информационных технологий

Техническое задание
на разработку мобильного приложения
«Сервис по заказу цветов и подарков Giftly»

Исполнители

_____ Д.В. Стеблев
_____ А.В. Одинцов
_____ Д.А. Пальчикова
_____ М.С. Нетесова
_____ Д.А. Кондратьев
_____ П.Н. Мироненко

Заказчик

_____ В.С. Тарасов

Воронеж 2025

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	5
1 Общие сведения	7
1.1 Полное наименование системы и её условное обозначение	7
1.2 Разработчики и заказчик	7
1.3 Перечень документов, на основании которых создаётся приложение	7
1.4 Плановые сроки начала и окончания работ по созданию АС	7
1.5 Порядок оформления и предъявления заказчику результатов работы	8
2 Цели и назначение создания приложения.....	9
2.1 Цели создания приложения.....	9
2.2 Назначение создания приложения.....	9
3 Характеристика объектов автоматизации.....	10
4 Требования к приложению	11
4.1 Функциональные требования	11
4.2 Нефункциональные требования	12
4.2.1 Требования к программному обеспечению приложения	12
4.2.2 Требования к оформлению и верстке страниц.....	13
4.2.3 Требования к системе администрирования приложения	13
4.2.4 Требования к безопасности.....	13
4.2.5 Требования к масштабируемости	14
4.2.6 Требования к документированию.....	14
4.2.7 Требования к языковым версиям.....	14
4.2.8 Технические риски	14
5 Состав и содержание работ по созданию приложения	16
6 Описание разрабатываемого приложения	17
6.1 Группы пользователей	17
6.2 Навигация по приложению	17
6.2.1 Нижнее меню (Tab Bar).....	17
6.2.2 Поисковая строка и фильтрация	17

6.3	Описание страниц покупателя.....	18
6.3.1	Загрузочный экран.....	18
6.3.2	Главный экран.....	18
6.3.3	Карточка продавца.....	19
6.3.4	Экран общения с ИИ	20
6.3.5	Экран корзины	21
6.3.6	Экран избранного	22
6.3.7	Экран профиля.....	23
6.4	Описание экранов продавца.....	24
6.4.1	Экран профиля.....	24
6.4.2	Экран заказов	25
7	Сценарии.....	27
7.1	Регистрация.....	27
7.2	Авторизация.....	27
7.3	Поиск, сортировка, фильтрация товаров.....	28
7.4	Добавление товара в корзину (для покупателей)	29
7.5	Добавление товара в избранное (для покупателей).....	29
7.6	Оформление заказа (для покупателей).....	29
7.7	Оплата заказа (для покупателей)	29
7.8	Общение с ИИ (для покупателей)	30
7.9	Редактирование личного кабинета (для покупателей)	30
7.10	Обращение в поддержку (для покупателей и продавцов).....	30
7.11	Просмотр статистики продаж (для продавцов)	30
7.12	Просмотр списка заказов (для продавцов).....	31
7.13	Просмотр карточки заказа от клиента (для продавцов)	31
7.14	Управление личным кабинетом (для продавцов)	31
7.15	Удаление/создание/редактирование товара (для продавцов).....	31
ПРИЛОЖЕНИЕ А		33
ПРИЛОЖЕНИЕ Б.....		34
ПРИЛОЖЕНИЕ В		35

ПРИЛОЖЕНИЕ Г	36
ПРИЛОЖЕНИЕ Д	37
ПРИЛОЖЕНИЕ Е	38
ПРИЛОЖЕНИЕ Ж	39

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем техническом задании применяют следующие термины с соответствующими определениями:

API (Application Programming Interface) – это набор правил и инструкций, с помощью которых различные приложения и сервисы могут взаимодействовать друг с другом

Back-end – серверная часть приложения

БД – база данных Веб-безопасность

SQL-инъекция – это уязвимость веб-безопасности, используемая злоумышленниками для вмешательства в запросы приложений к базе данных

Figma – инструмент для создания дизайна мобильных и веб-приложений (UI/UX)

Front-end – клиентская (видимая) часть приложения

GitHub – платформа, на которой разработчики могут хранить свои проекты, управлять и делиться ими

ИИ – искусственный интеллект

Jira – программное обеспечение для управления проектами и задачами

Клиент (клиентская часть) – это часть компьютерного оборудования или программного обеспечения, которая получает доступ к сервису, предоставляемому сервером

Клиент–серверное приложение – это распределённая архитектура, в которой задачи выполняются на двух типах устройств: клиенте и сервере

Miro – онлайн-платформа для командной работы. Аналог маркерной доски

Нефункциональные требования – это требования к качествам, свойствам и характеристикам системы или продукта в целом, которые не связаны с конкретными функциями системы

ORM (Object Relational Mapping) – это технология, которая позволяет работать с базами данных так, как если бы это были объекты из языков программирования

ПО – программное обеспечение

Реляционная база данных – это тип базы данных, в которой данные организованы в виде связанных таблиц, также называемых отношениями

REST API – архитектурный стиль взаимодействия между клиентом и сервером

Сервер (серверная часть) – компьютер, обрабатывающий запросы от других компьютеров

Фреймворк – заготовка, готовая модель в программировании для быстрой разработки, на основе которой можно дописать собственный код

Функциональные требования – это описание, какие возможности должна предоставлять программа и как она обрабатывает ввод пользователя

1 Общие сведения

1.1 Полное наименование системы и её условное обозначение

Полное наименование приложения: «Сервис по продаже цветов и подарков Giftly».

Условное обозначение приложения: «Giftly».

1.2 Разработчики и заказчик

Заказчик: Старший Преподаватель Тарасов Вячеслав Сергеевич, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

Разработчик: 4 команда группы 1.

Состав команды разработчика:

- Стеблев Данила Владимирович;
- Одинцов Артём Владимирович;
- Пальчикова Дарья Александровна;
- Нетесова Мария Сергеевна;
- Кондратьев Дмитрий Александрович;
- Мироненко Пётр Николаевич.

1.3 Перечень документов, на основании которых создаётся приложение

Приложение создаётся на основе:

- Федеральный закон "О персональных данных" от 27.07.2006 N 152–ФЗ;
- ГОСТ 34.602-2020;
- ГОСТ 7.32;
- Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149–ФЗ.

1.4 Плановые сроки начала и окончания работ по созданию АС

Дата начала работ: февраль 2025.

Дата окончания работ: июнь 2025.

1.5 Порядок оформления и предъявления заказчику результатов работы

Предварительные отчёты по работе будет проводиться во время рубежных аттестаций:

1 аттестация (конец марта 2025) - создан репозиторий проекта на GitHub, распределены задачи проекта в таскменеджере Jira, создан проект Miro с общей логикой системы, предоставлено готовое техническое задание, создан проект в Figma с предварительным дизайном приложения и пользовательскими сценариями;

2 аттестация (конец апреля 2025) - написана основополагающая часть кода приложения, реализована БД и ее взаимодействие с сервером, проведена отладка и доработка кода, проведено тестирование по работе системы;

3 аттестация (конец мая 2025) - разработан курсовой проект, выполнены завершающие работы по доработке приложения, предоставлена готовая система.

2 Цели и назначение создания приложения

2.1 Цели создания приложения

Целями создания приложения являются:

- реализация процесса купли-продажи цветов и подарков через мобильное приложение;
- получение прибыли за счёт комиссий за использования сервиса;
- создание платформы для связывания продавцов и покупателей.

2.2 Назначение создания приложения

Система должна решать следующие задачи:

1. Хранение и управление личными данными:

- редактирование личных данных в профиле;
- хранение и выборка личных данных из БД.

2. Реализация процесса купли-продажи товаров:

- выбор, оформление заказа и оплата понравившихся товаров покупателем;
- продажа своих товаров продавцами и получение прибыли с продаж.

3. Возможность подбора цветов и подарков по запросу пользователя ИИ-консультантом:

- общение с ИИ-консультантом через встроенный чат;
- предложение товаров (цветов, подарков) от ИИ по запросу пользователя.

3 Характеристика объектов автоматизации

Объектом автоматизации в мобильном приложении по заказу цветов и подарков являются следующие процессы:

— Оформление и обработка заказов – автоматизация приема заказов через мобильное приложение;

— Оплата и финансовые операции – возможность проведения платежей за товар банковской картой.

4 Требования к приложению

4.1 Функциональные требования

Разрабатываемое приложение должно соответствовать следующим функциональным требованиям:

Система должна позволять Неавторизованному пользователю:

- зарегистрироваться в приложении;
- авторизоваться в приложении, если аккаунт уже создан;
- просмотреть каталог товаров;
- осуществить поиск товаров;
- просмотреть карточку товара, информацию о нём.

Система должна позволять Авторизованному пользователю:

- просмотреть каталог товаров;
- осуществить поиск товаров;
- просмотреть карточку товара, информацию о нём;
- просмотреть информацию о продавце;
- добавить товар в избранное;
- добавить товар в корзину;
- осуществить оплату товара;
- управлять личным кабинетом (изменить личные данные, сменить пароль);
- заполнить личный кабинет (дата рождения, ФИО для сбора информации и индивидуальных предложений);

Система должна позволять Продавцу:

- управлять каталогом товаров (добавление, редактирование, удаление своих товаров);
- выставить новый товар на личную страницу;
- создать карточку нового товара с добавлением фото, описания, цены;

- просмотреть личную подробную статистику продаж, отзывов, рейтинга;

- выйти из профиля.

Система должна позволять Администратору:

- блокировать продавца;

- блокировать пользователя.

4.2 Нефункциональные требования

4.2.1 Требования к программному обеспечению приложения

Приложение должно иметь архитектуру, соответствующую шаблону клиент – серверного приложения и иметь разделение на back-end (сервер) и front-end (клиент), связанных между собой REST API (см. приложение А).

Для реализации серверной части мобильного приложения будут использоваться следующие средства:

- фреймворк Django;

Django является мощным и универсальным фреймворком для создания API на Python, который ускоряет создание сложных приложений. Фреймворк сразу предлагает встроенную админ-панель, систему аутентификации, ORM для работы с базами данных и механизмы обработки форм.

- СУБД PostgreSQL;

Данная СУБД – мощная реляционная база данных, которая идеально сочетается с Django благодаря своей надежности, гибкости и богатому функционалу.

- инструмент Swagger ;

Инструмент необходим для написания документации к API.

- Docker;

Инструмент позволит упаковать части приложения в отдельные «контейнеры» и повысит удобство при развертывании.

Для реализации клиентской части мобильного приложения будут использоваться следующие средства:

- язык программирования Dart;

— фреймворк Flutter.

Flutter — это фреймворк, который позволяет писать один код для iOS, Android, веба и даже десктопных приложений, экономя время и ресурсы. Он обеспечивает высокую производительность благодаря компиляции в нативный код, а также предлагает богатую библиотеку готовых виджетов для создания красивого и отзывчивого интерфейса.

Dart, в свою очередь, идеально подходит для Flutter благодаря своей скорости, простоте и предсказуемости. Его синтаксис интуитивно понятен (похож на Java и JavaScript).

4.2.2 Требования к оформлению и верстке страниц

Оформление и верстка страниц приложения должны удовлетворять следующим требованиям:

— приложение должно быть оформлено в едином цветовом стиле, используя цвета #9191E9, #B3B3B3, #000000, #FFFF. Допускается использование других цветов в соответствии с изменениями дизайна в Figma;

— приложение должно корректно отображаться на экранах размерами: 4.7" – 5.5" дюймов (классические смартфоны: iPhone 8, Samsung Galaxy A10);

— интерфейс должен быть адаптивным (автоматически подстраиваться под размер экрана);

— составляющие элементы приложения (надписи, кнопки, значки, иконки) должны хорошо читаться без масштабирования.

4.2.3 Требования к системе администрирования приложения

В целях администрирования приложения, должна выделяться специальная роль – администратор.

4.2.4 Требования к безопасности

Мобильное приложение должно соответствовать следующим требованиям к безопасности:

— защита от SQL-инъекций и других уязвимостей на стороне API за счёт использования фреймворка Django и встроенного в него ORM;

— пароли следует шифровать и записывать в БД в зашифрованном виде.

4.2.5 Требования к масштабируемости

Мобильное приложение должно соответствовать следующим требованиям к масштабируемости:

— система должна поддерживать увеличение числа пользователей и объема данных, используя «горизонтальный» метод масштабирования, то есть с помощью добавления новых серверов и равномерным распределением нагрузки между ними при необходимости.

4.2.6 Требования к документированию

Вся документация должна оформляться на платформе GitHub в созданном репозитории.

По результатам работы необходимо предоставить (выложить) следующие документы:

- предпроектное исследование;
- техническое задание;
- ссылка на макеты Figma;
- ссылка на таск-трекер Jira;
- ссылка на Miro;
- курсовая работа;
- все презентации сервиса;
- видеозаписи/файлы презентаций сервиса.

4.2.7 Требования к языковым версиям

На данном этапе разработки предполагается только русскоязычная версия приложения. Поддержка иностранных языков не предусмотрена.

4.2.8 Технические риски

Выделяются следующие технические риски при разработке приложения:

- разрастание объема - неконтрольное увеличение объема работы;

Решение: правильная постановка исходных задач, равномерное распределение обязанностей между командой, согласование объёма работ с заказчиком в самом начале и придерживание этой позиции.

— низкая производительность;

Решение: использовать ПО для управления проектами, грамотно распределять задачи между командой.

— большие затраты;

Решение: разумно оценить каждый элемент проекта и придерживаться исходному бюджету.

— отсутствие ясности – недопонимание среди заинтересованных сторон;

Решение: проверять все ли участники проекта в курсе происходящего, доводить задачи до каждого из участников.

— недостаточность ресурсов;

Решение: создание плана распределения ресурсов.

— недостаток времени;

Решение: отвести больше времени на выполнение задач и предусмотреть запас времени на случай непредвиденных обстоятельств, не откладывать в «долгий ящик».

5 Состав и содержание работ по созданию приложения

Состав и содержание работ по созданию мобильного приложения включают в себя следующие этапы:

- Сбор необходимой информации, постановка целей, задач системы, которые в будущем должны быть реализованы 16.02.25 – 01.03.25;
- Анализ предметной области, анализ конкурентов и построение структуры требований, ведущих к решению поставленных задач и целей 01.03.25 – 16.03.25;
- Построение модели программы, описание спецификаций данных, определение связей между сущностями, разработка модели БД 16.03.25 – 01.04.25;
- Разработка рабочего проекта, состоящего из написания кода, отладки и корректировки кода программы 16.04.25 – 16.05.25;
- Проведение тестирования программного обеспечения 16.05.25 – 01.06.25.

6 Описание разрабатываемого приложения

6.1 Группы пользователей

Пользователи приложения должны делиться на следующие группы:

Основные группы:

- неавторизированный пользователь;
- авторизированный пользователь.

Дополнительные роли (уже авторизованы):

- продавец;
- администратор.

6.2 Навигация по приложению

6.2.1 Нижнее меню (Tab Bar)

На всех экранах приложения снизу должно быть расположено нижнее меню, необходимое для перехода на другие страницы. Набор кнопок зависит от роли.

Для покупателя нижнее меню должно включать в себя следующие кнопки:

- главная;
- ИИ – помощник;
- избранное;
- корзина;
- профиль.

Для продавца нижнее меню должно включать в себя следующие кнопки:

- главная;
- статистика продавца;
- список заказов;
- профиль.

6.2.2 Поисковая строка и фильтрация

Также на экранах приложения сверху должна быть расположена поисковая строка для поиска необходимых товаров. Рядом с поисковой строкой должны быть расположены кнопки фильтрации и сортировки.

6.3 Описание страниц покупателя

6.3.1 Загрузочный экран

Данный экран должен показываться при входе в приложение, при загрузке всех его основных элементов вне зависимости от роли пользователя. В центре экрана должно находиться название сервиса «Giftly» с его слоганом «Ваши подарки, наши заботы». Внешний вид загрузочного экрана можно увидеть на рисунке 1.

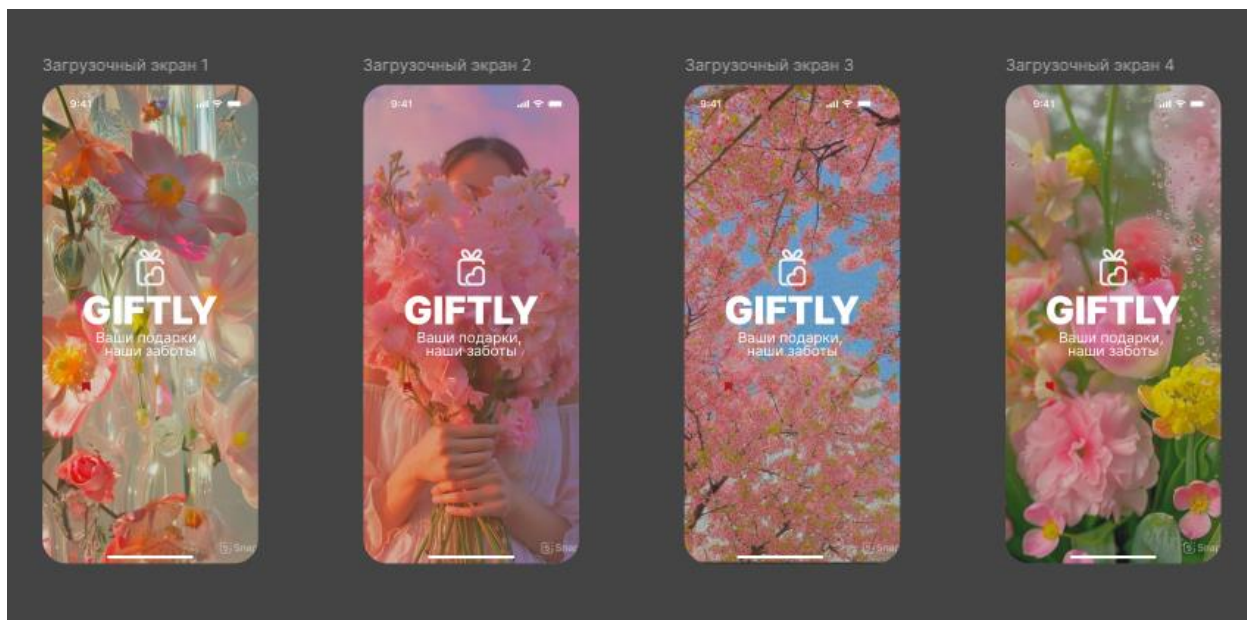


Рисунок 1 — Варианты загрузочного экрана

После загрузки всех компонентов приложения загрузочный экран пропадает и пользователь переходит на главную страницу (каталог товаров).

6.3.2 Главный экран

Экран должен иметь разный вид в зависимости от роли и должна быть доступна всем пользователям. На странице должны находиться:

- сверху – поисковая строка и кнопки сортировки и фильтрации;
- посередине – карточки товаров;
- снизу – нижнее меню, набор кнопок которого зависит от роли.

На рисунке 2 показан внешний вид главной страницы.

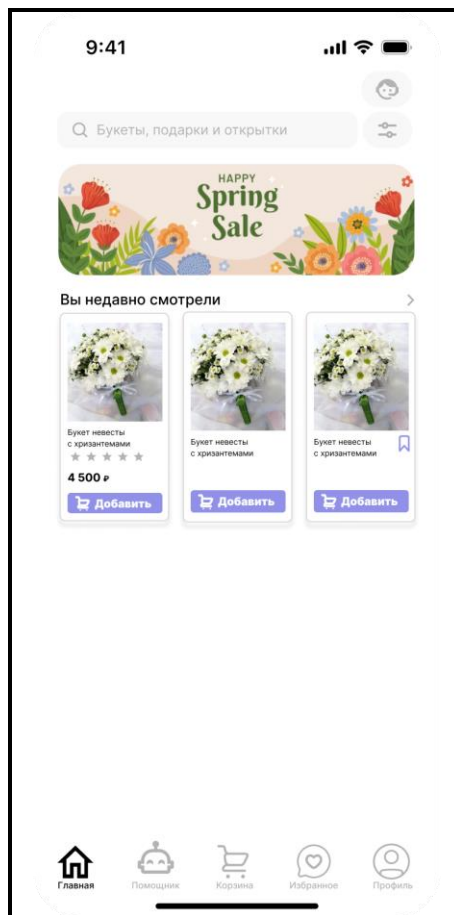


Рисунок 2 — Главный экран

6.3.3 Карточка продавца

На странице должны находиться:

- сверху – поисковая строка и кнопка сортировки и фильтрации;
- под поисковой строкой – логотип магазина, его название и рейтинг;
- по центру – карточки товаров;
- снизу - нижнее меню, набор кнопок которого зависит от роли.

На рисунке 3 изображён внешний вид страницы карточки продавца.

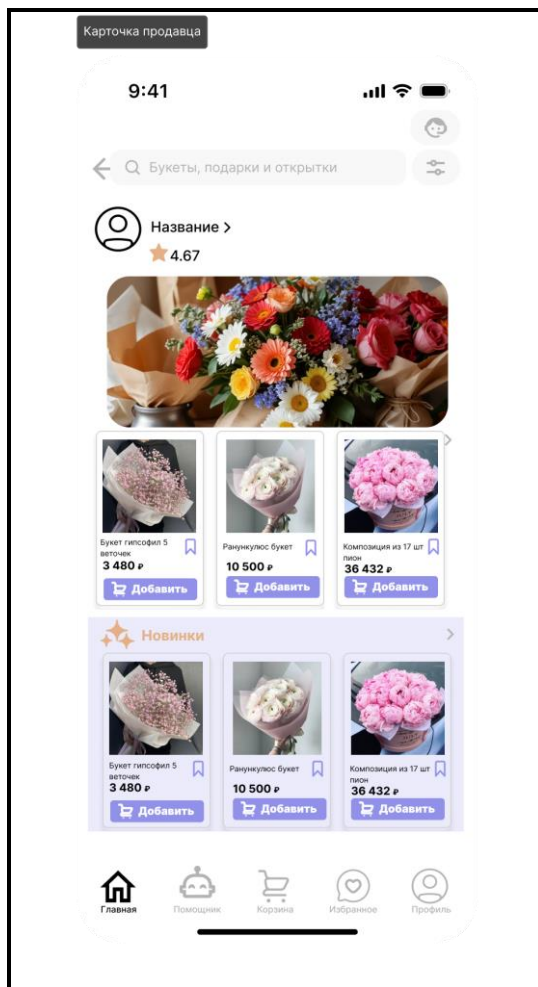


Рисунок 3 — Карточка продавца

6.3.4 Экран общения с ИИ

Данный экран должна быть доступна только авторизованному пользователю.

На странице должен быть представлен чат с ИИ, который предлагает товары покупателю. Снизу должно быть расположено нижнее меню.

На рисунке 4 изображен экран общения с ИИ.

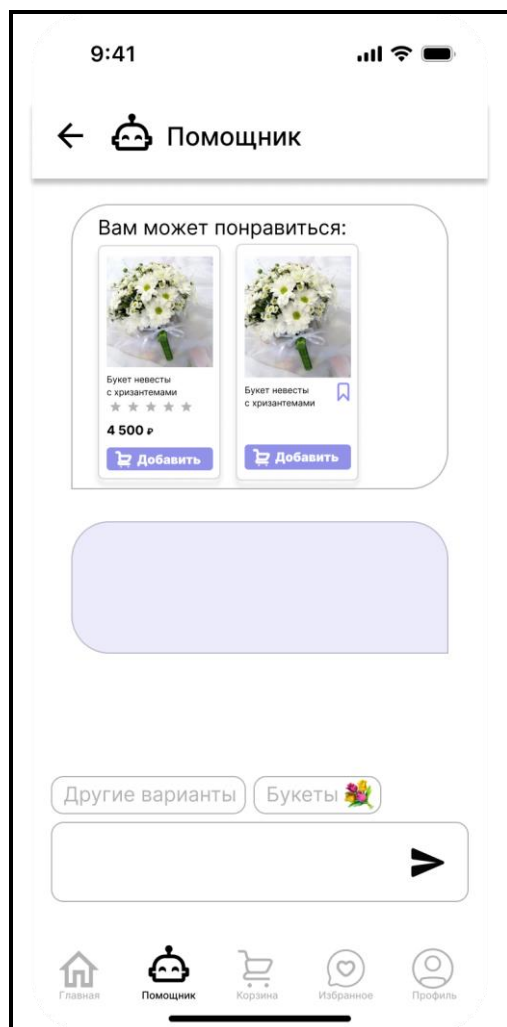


Рисунок 4 — Экран общения с ИИ

6.3.5 Экран корзины

Экран должен быть доступен только авторизованному пользователю.

Экран должен представлять собой список товаров, которые пользователь желает приобрести. Также должна быть указана сумма покупок и кнопка оформления заказа.

Внешний вид экрана корзины изображён на рисунке 5.

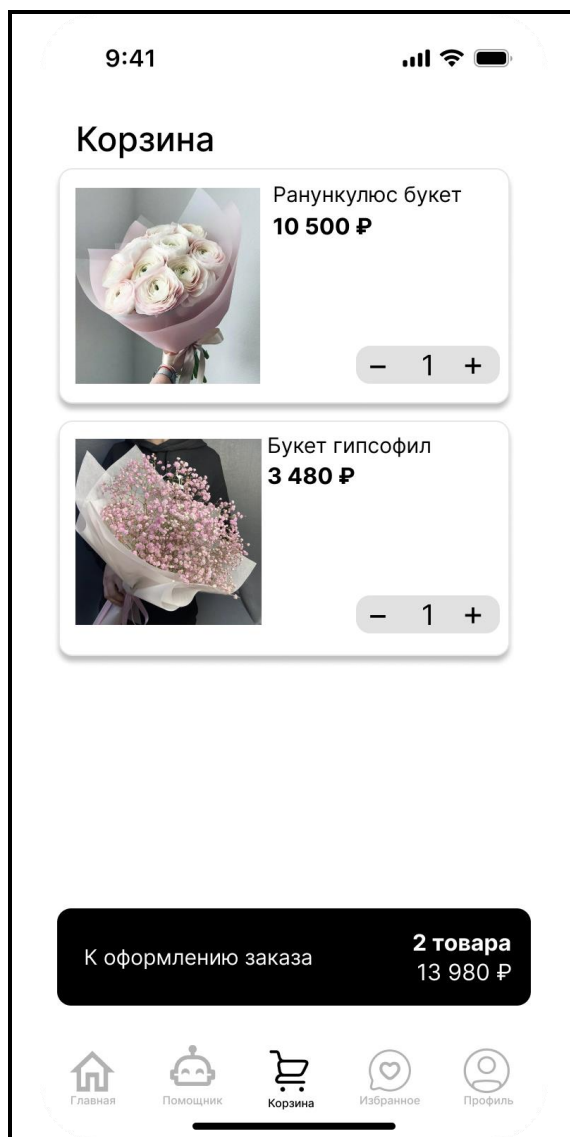


Рисунок 5 — Экран корзины

6.3.6 Экран избранного

Экран должен быть доступен только авторизованному пользователю.

Экран должен представлять собой список понравившихся пользователю товаров.

На рисунке 6 изображен экран избранного.

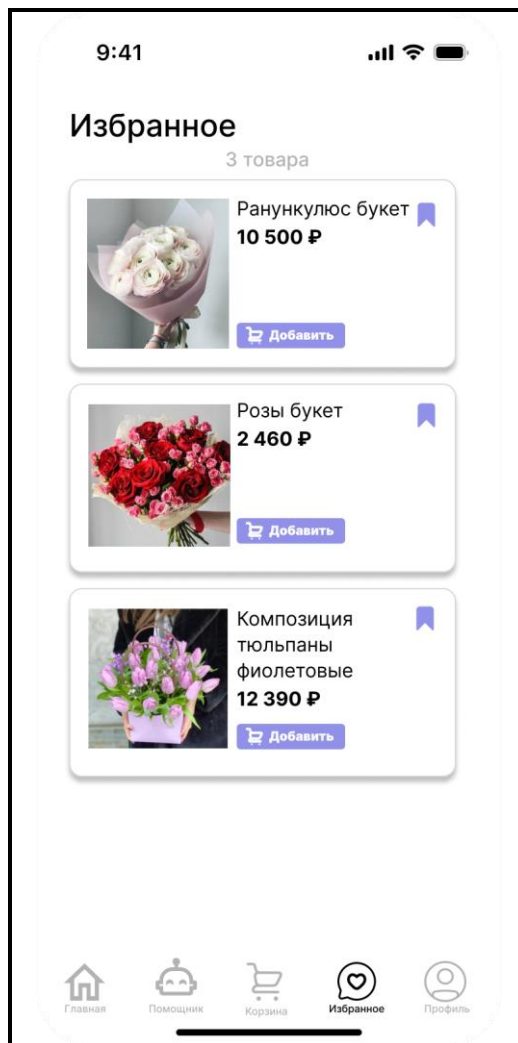


Рисунок 6 — Экран избранного

6.3.7 Экран профиля

Экран должен быть доступен только авторизованному пользователю.

На экране должны быть расположены:

- аватарка клиента;
- контактная информация клиента;
- история заказов клиента;
- личные бонусы, скидки, акции;
- настройки;
- кнопка обращения к поддержке.

Экран профиля клиента показана на рисунке 7.

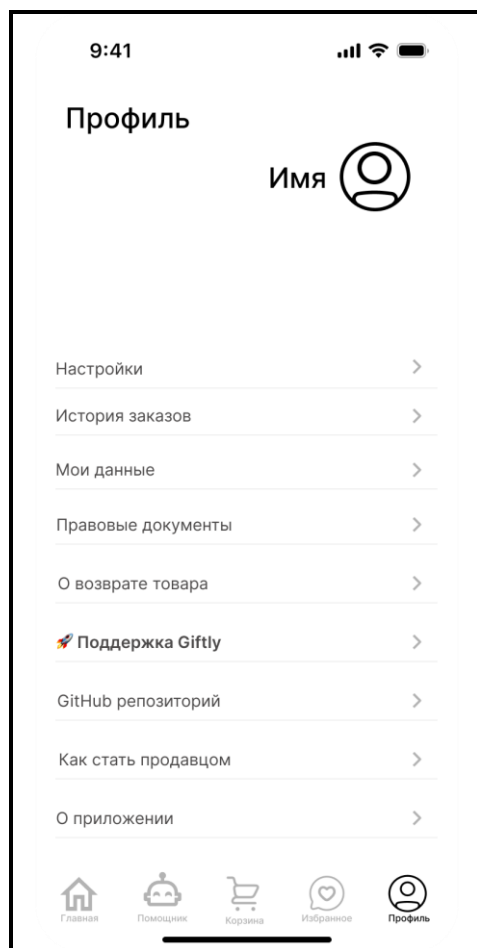


Рисунок 7 — Экран профиля клиента

6.4 Описание экранов продавца

6.4.1 Экран профиля

Экран должен быть доступен только авторизованному пользователю.

На экране должны быть расположены:

- аватарка продавца;
- контактная информация продавца;
- товары продавца;
- настройки;
- кнопка обращения к поддержке.

На рисунке 8 показан экран профиля продавца.

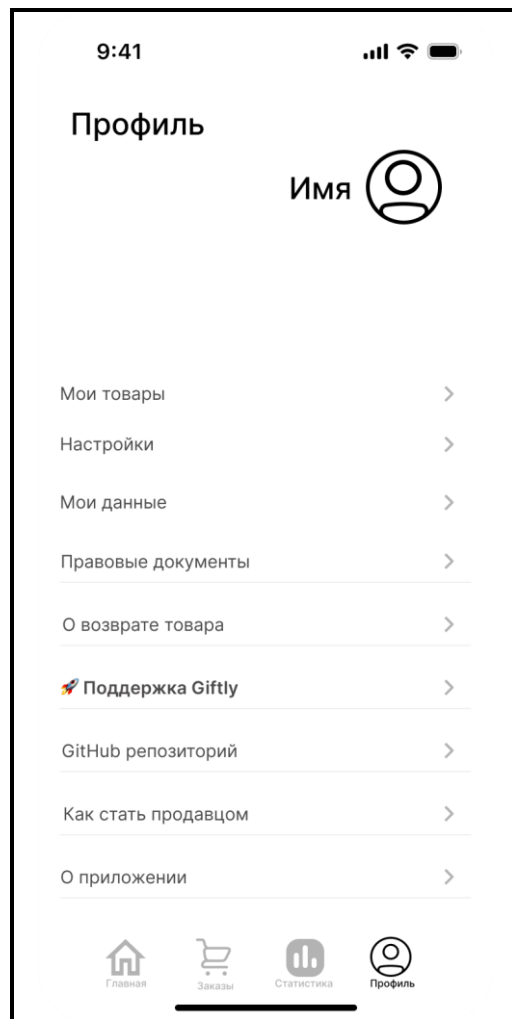


Рисунок 8 — Экран профиля продавца

6.4.2 Экран заказов

Экран должен быть доступен только авторизованному пользователю.

На экране должны находиться:

- сверху страницы – заголовок «Заказы» и подзаголовок «Невыполненные заказы»;
- карточка заказанного товара и данные о заказчике;
- снизу – нижнее меню продавца с набором кнопок.

Внешний вид экрана заказов показан на рисунке 9.

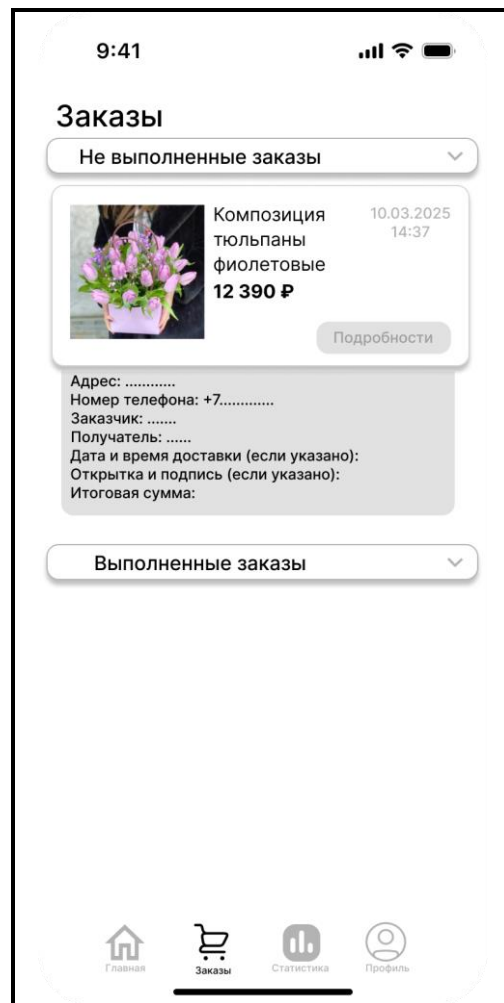


Рисунок 9 — экран заказов

7 Сценарии

Приложение должно предоставлять некоторый ряд функций по различным запросам пользователей. Функциональность приложения отражена в функциональных схемах (см. ПРИЛОЖЕНИЕ Б - Ж)

7.1 Регистрация

Пользователь, который впервые зашёл в сервис, должен иметь возможность зарегистрировать аккаунт. Для этого ему нужно будет принять пользовательское соглашение и поставить галочку в чекбоксе. Также необходимо ввести логин и пароль, которые будут использоваться в дальнейшем.

Экран регистрации изображен на рисунке 10.

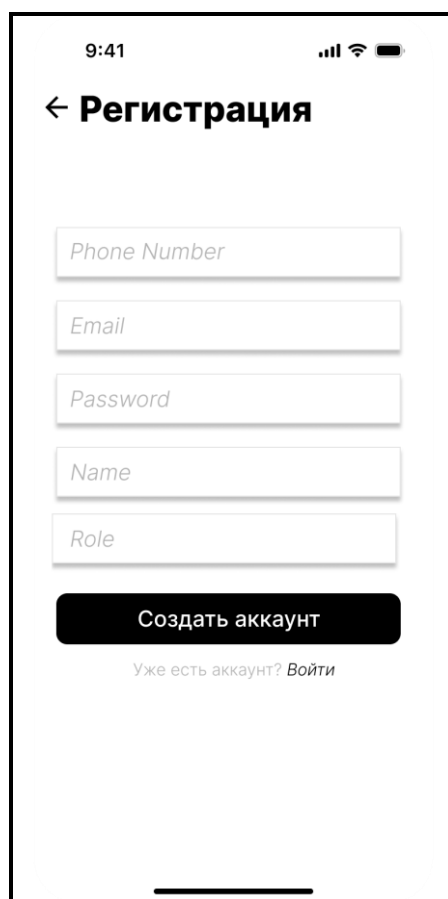


Рисунок 10 — Экран регистрации

7.2 Авторизация

Пользователь, создавший аккаунт, должен иметь возможность авторизоваться. Пользователь попадает на экран авторизации. На нем представлены поля для ввода логина и пароля, кнопка «Забыли пароль» и кнопка «Зарегистрироваться». Пользователь решает какой сценарий ему использовать. Если пользователь корректно вводит данные, то он попадает на главный экран приложения. Если пользователь забыл свой пароль, то ему предлагается его восстановить. Если пользователь не зарегистрирован, ему предлагается кнопка создать аккаунт с переходом на экран регистрации. Внешний вид экрана авторизации показан на рисунке 11.

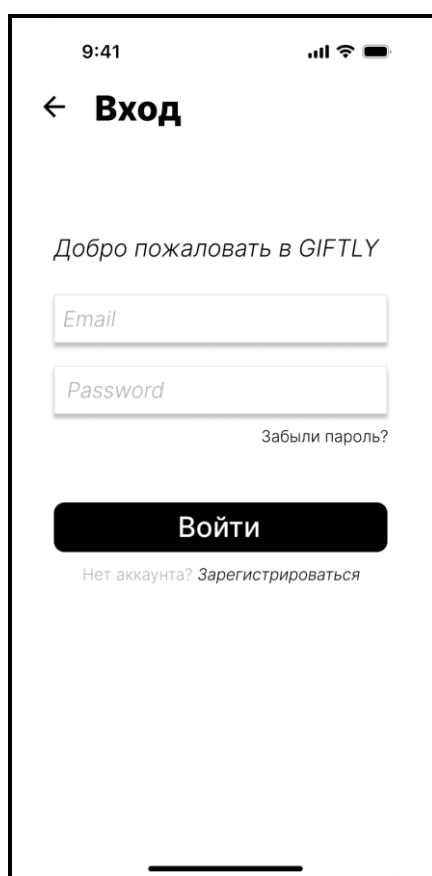


Рисунок 11 — Экран авторизации

7.3 Поиск, сортировка, фильтрация товаров

Когда пользователь попадает на главный экран (каталог) он должен видеть перед собой карточки товаров. Если пользователь не может найти нужный ему товар, он должен иметь возможность воспользоваться поисковой строкой. Для этого ему необходимо ввести запрос, пример: «свадебный букет», и дождаться ответа от сервиса. После чего ему выдаст

полученные результаты, если таковые были найдены. В случае, если ничего не было найдено, пользователь должен получить ответ – ничего не найдено. Также пользователь должен иметь возможность отсортировать уже полученные товары или отфильтровать по различным категориям.

7.4 Добавление товара в корзину (для покупателей)

В случае, если пользователь решился приобрести определённый товар, он должен иметь возможность добавить его в корзину для дальнейшей оплаты и доставки. Для этого ему необходимо на карточке выбранного товара найти кнопку «Добавить в корзину» и нажать на неё. После этого, зайдя в раздел «Корзина» (сделать это можно в нижнем меню, нажав на иконку с корзиной), пользователь должен увидеть добавленный товар и должен иметь возможность оформить заказ.

7.5 Добавление товара в избранное (для покупателей)

В случае, если пользователю понравился определённый товар, но он пока его не может/не хочет приобрести, должна быть возможность его добавить в раздел «Избранное». Для этого на карточке товара должна быть иконка в виде сердечка и возможность нажать на неё. После этого, зайдя в раздел «Избранное» (сделать это можно в нижнем меню, нажав на иконку с сердечком), пользователь должен увидеть список всех понравившихся ему товаров и должен иметь возможность добавить их в корзину или удалить из избранного.

7.6 Оформление заказа (для покупателей)

Когда товар добавлен в корзину, и пользователь готов произвести оформление заказа, ему необходимо нажать на кнопку «Оформить заказ» в корзине. При этом он увидит сумму заказа и остальную информацию. После нажатия на кнопку «Оформить заказ», пользователь должен попасть на страницу оформления заказа, где должен иметь возможность выбрать способ получения: доставка или самовывоз и оплатить заказ.

7.7 Оплата заказа (для покупателей)

При оформлении заказа покупателю предложат оплатить его картой. Для этого необходимо будет ввести данные карты и получить ответ. В случае успешной оплаты, пользователь получит окно «Успешно», иначе «Ошибка, повторите снова».

7.8 Общение с ИИ (для покупателей)

В сервис внедрен рекомендательный бот – помощник, который должен будет предлагать вам товары на основе ваших прошлых заказов. Для общения с ним необходимо перейти в раздел «ИИ – помощник», нажав на иконку робота в нижнем меню. После этого должны появиться рекомендации товаров от ИИ.

7.9 Редактирование личного кабинета (для покупателей)

Также авторизированные пользователи должны иметь возможность редактировать фото, данные в личном кабинете для сбора информации и предложений товаров. Для этого необходимо нажать на главной странице на значок человека в нижнем меню, после чего пользователь должен попасть на страницу личного кабинета. Экран должен предоставлять следующее:

- фотография клиента;
- контактная информация (ФИО, дата рождения и т.п.);
- история заказов;
- личные бонусы, размеры скидок, акции;
- привязанные способы оплаты;
- кнопка обращения в поддержку.

7.10 Обращение в поддержку (для покупателей и продавцов)

Любой пользователь в праве обратиться в поддержку для решения какого-либо вопроса, исправления бага и т.п. Для этого необходимо зайти в личный кабинет (как это сделать см. в пункте 8.9) и нажать на кнопку «Поддержка». После чего должно открыться поле ввода обращения, поле ввода вашего email для обратной связи.

7.11 Просмотр статистики продаж (для продавцов)

В случае, если вход произведён в роли продавца, то пользователь должен иметь возможность посмотреть статистику своего магазина, продажи, выручку. Для этого на главной странице (каталог товаров) необходимо в нижнем меню перейти в раздел «Статистика». Далее пользователя должно перекинуть на страницу статистики, где он должен иметь возможность подробно изучить её.

7.12 Просмотр списка заказов (для продавцов)

Когда авторизованный пользователь оформляет заказ, он должен «упасть» в список заказов у продавца. Для того, чтобы попасть в данный раздел необходимо на главном экране перейти в раздел «Список заказов» в нижнем меню, после чего пользователь в лице продавца должен попасть на страницу списка заказов, где должен иметь возможность с ним ознакомиться.

7.13 Просмотр карточки заказа от клиента (для продавцов)

После того, как пользователь в лице продавца попал на страницу списка заказов, он должен иметь возможность узнать информацию о любом заказе, кликнув на него. После этого должна появиться карточка заказа, где будет указана вся необходимая информация для его выполнения. Также снизу карточки должны быть расположены кнопки «Выполнить заказ» и «Отмена заказа».

7.14 Управление личным кабинетом (для продавцов)

Продавец, также как и покупатель должен иметь свой личный кабинет. В нём должны быть расположены:

- контактная информация продавца (название, адрес и т.д.);
- аватарка продавца;
- список товаров продавца;
- обложка продавца.

Пользователь в лице продавца также должен иметь возможность добавлять, редактировать, удалять товары из списка своих товаров.

7.15 Удаление/создание/редактирование товара (для продавцов)

Для управления списком своих товаров в личном кабинете продавца должен быть список всех товаров. Продавец должен иметь возможность создавать/редактировать/удалять товары по мере необходимости. Для этого необходимо наличие соответствующей кнопки и формы с необходимыми данными для заполнения (для создания или редактирования товара).

ПРИЛОЖЕНИЕ А

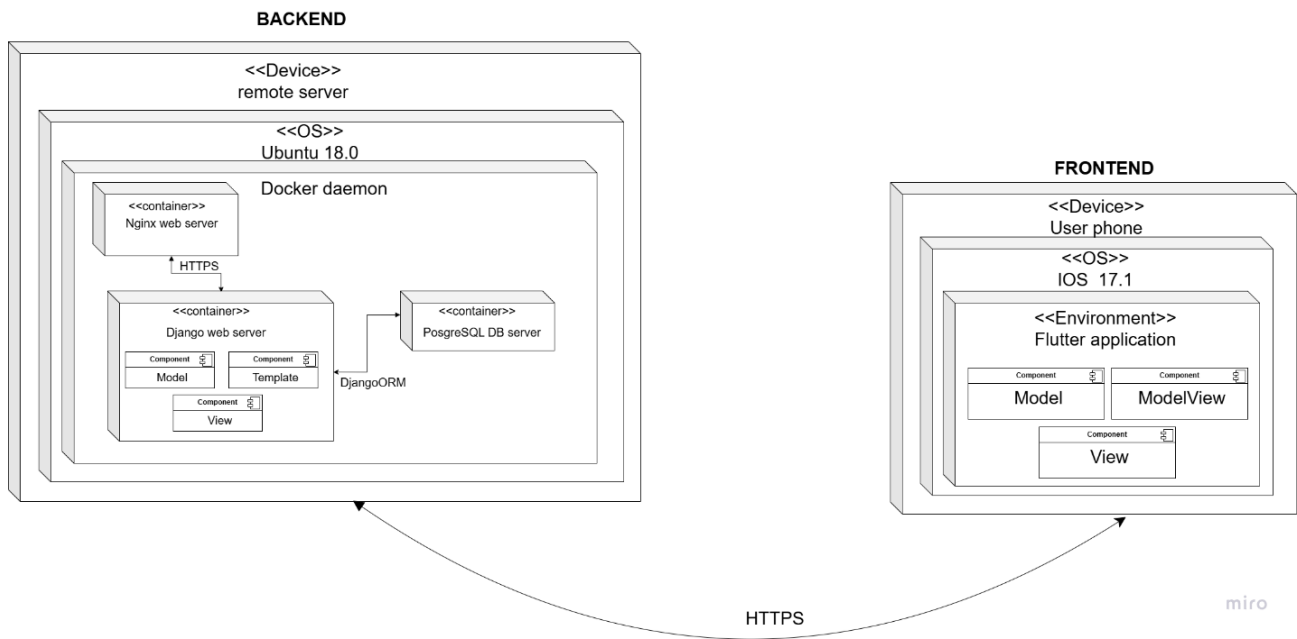


Рисунок 12 — Диаграмма развертывания приложения

ПРИЛОЖЕНИЕ Б

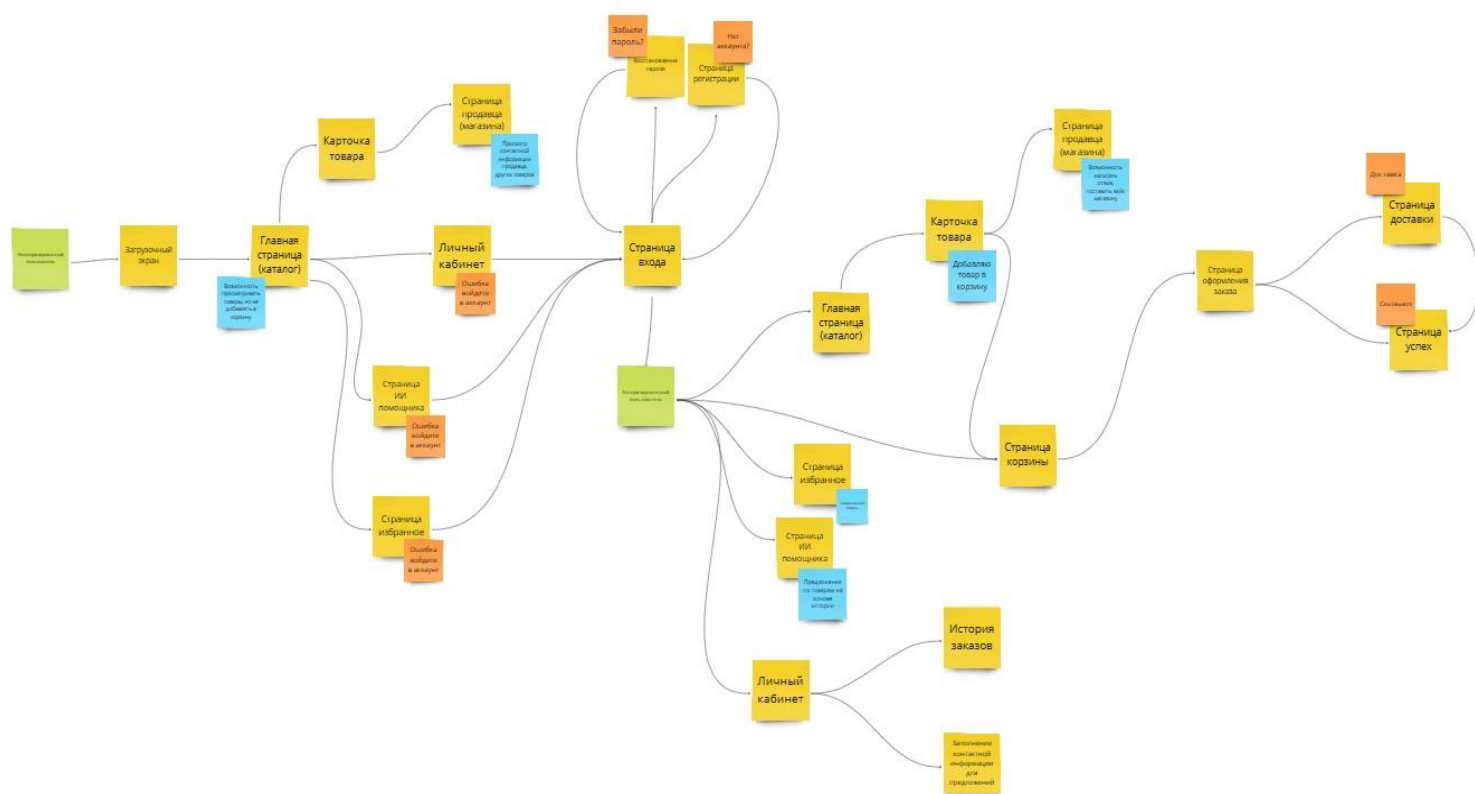


Рисунок 13 — Пользовательские сценарии покупатель

ПРИЛОЖЕНИЕ В

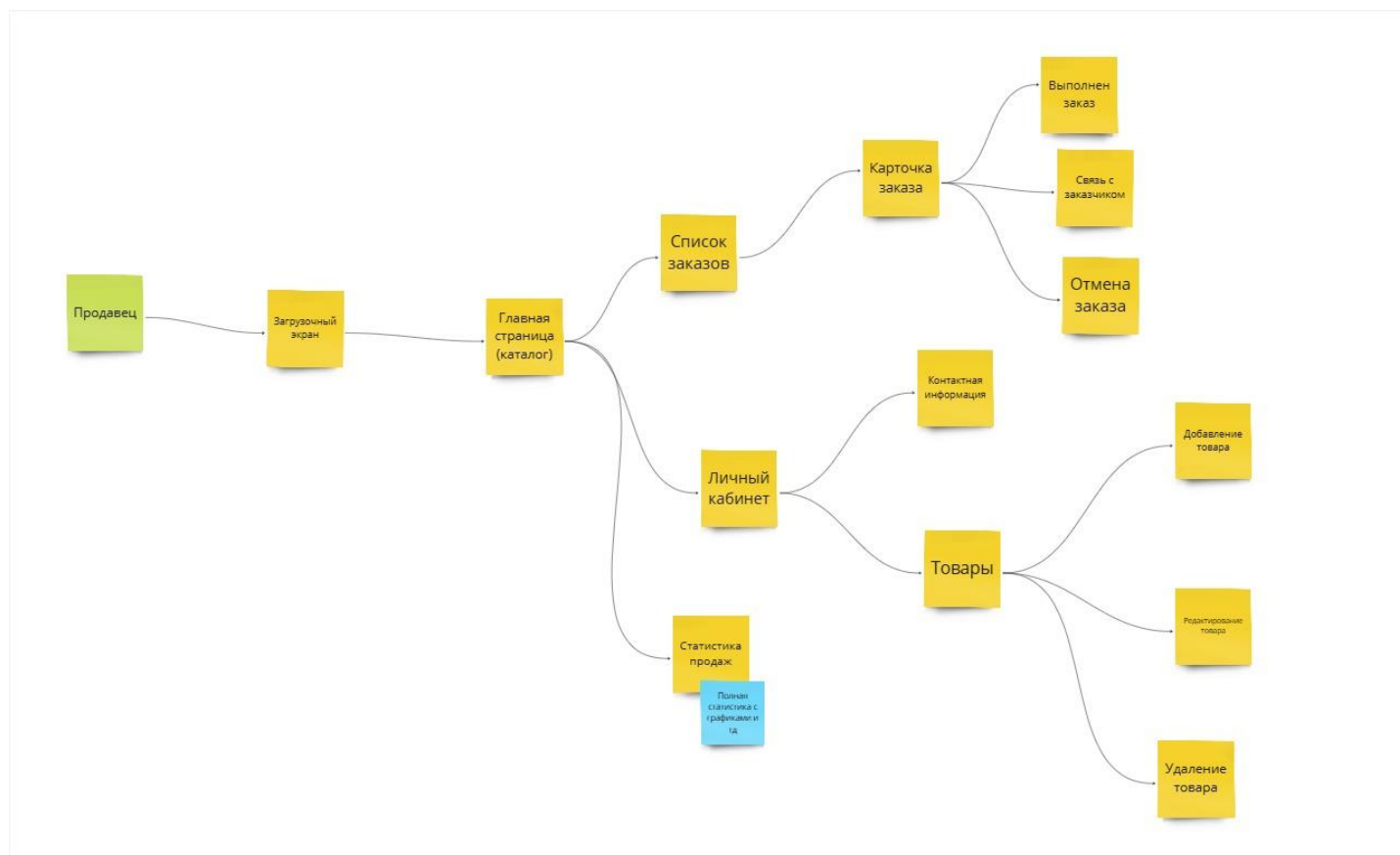


Рисунок 14 — Пользовательские сценарии продавец

ПРИЛОЖЕНИЕ Г

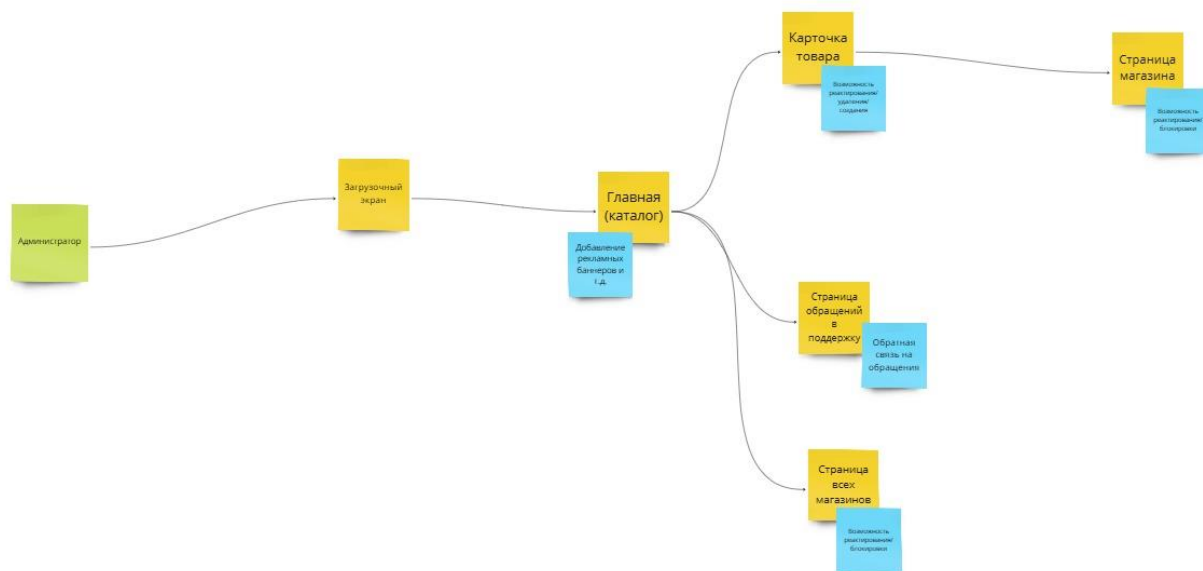


Рисунок 15 — Пользовательские сценарии администратор

ПРИЛОЖЕНИЕ Д

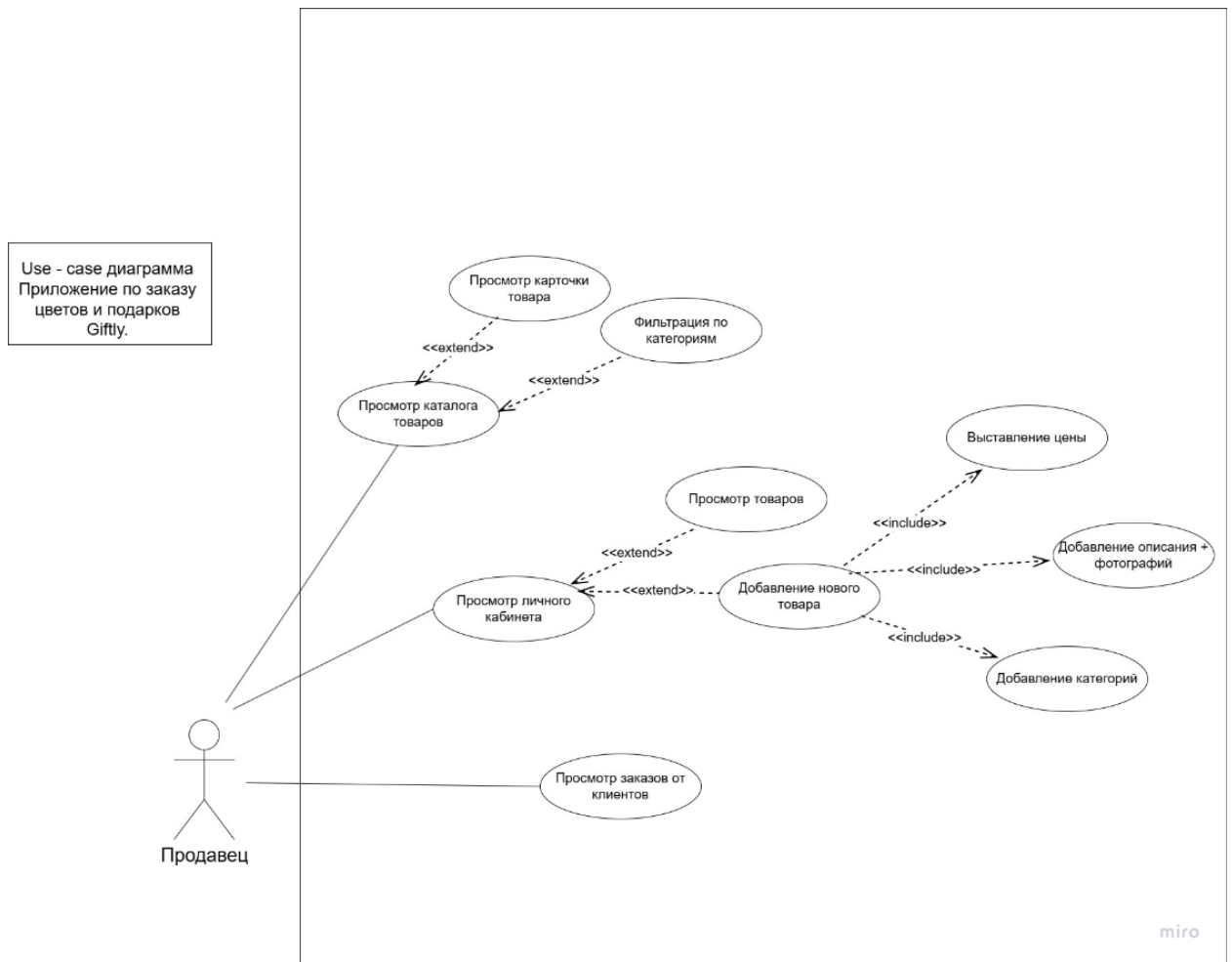


Рисунок 16 — Диаграмма использования Продавец

ПРИЛОЖЕНИЕ Е

Use - case диаграмма
Приложение по заказу
цветов и подарков
Giftly.

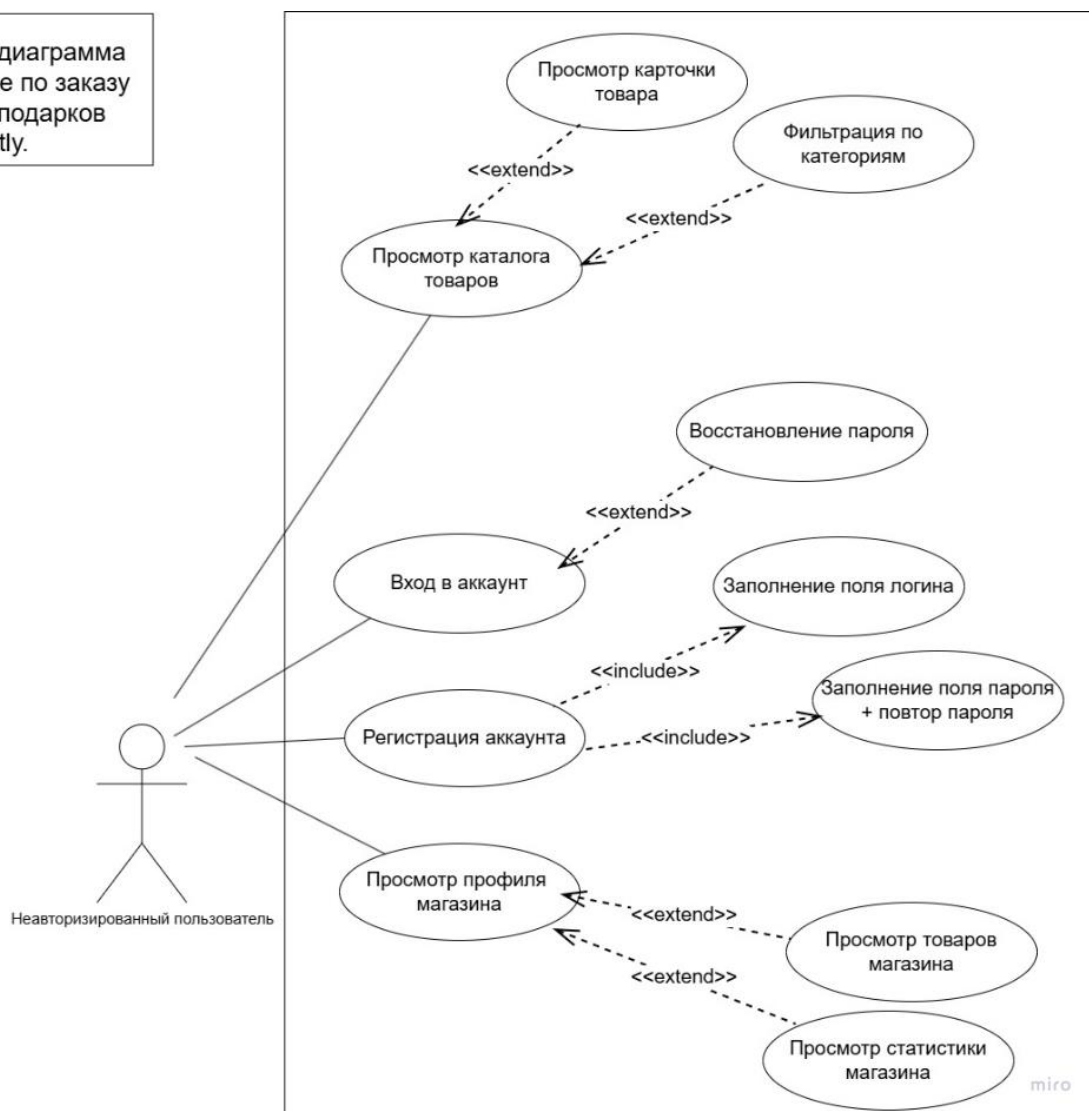


Рисунок 17 — Диаграмма использования Неавторизированный

ПРИЛОЖЕНИЕ Ж

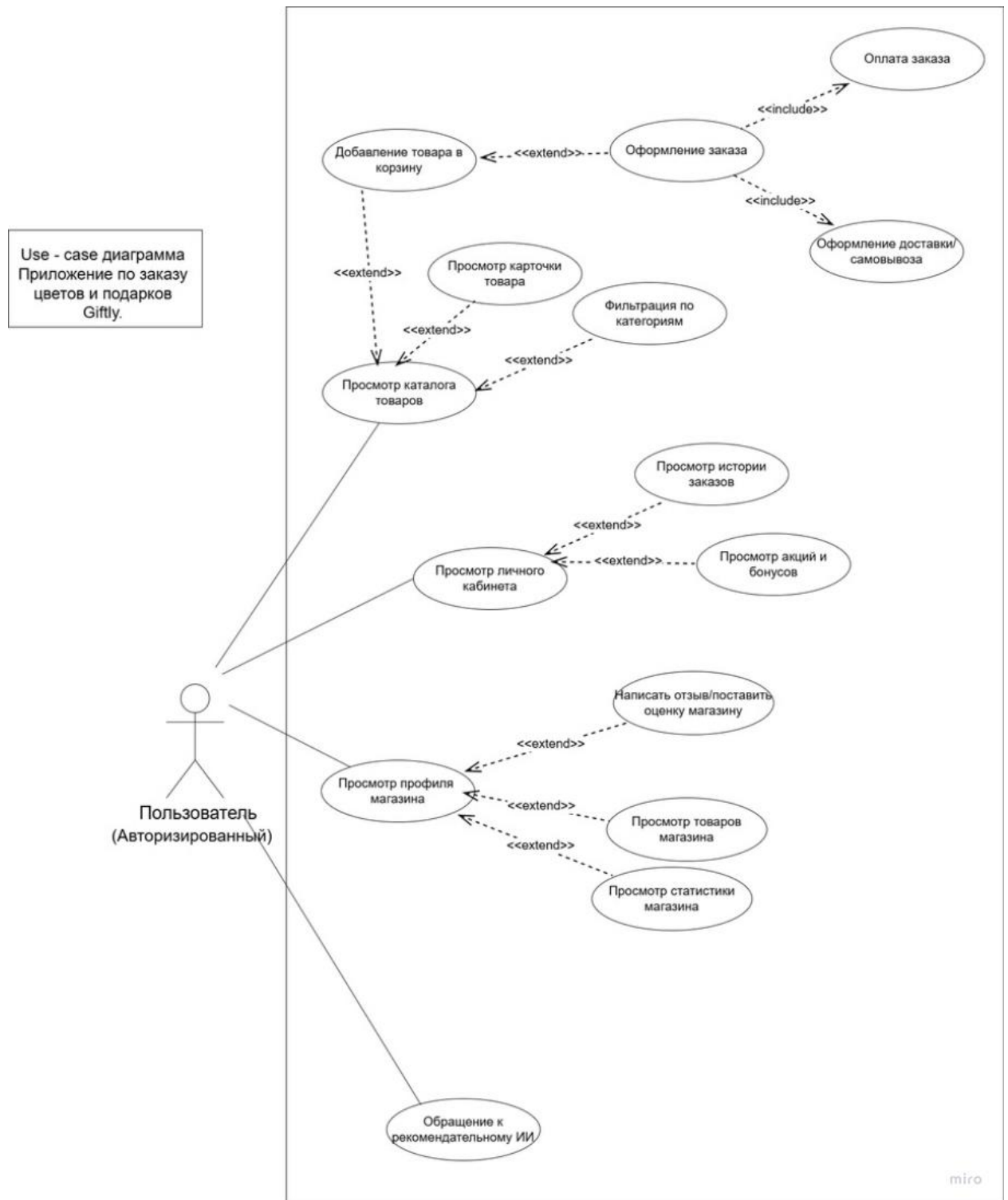


Рисунок 18 — Диаграмма использования Пользователь