

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

Факультет компьютерных наук
Кафедра программирования и информационных технологий

Курсовая работа
Разработка мобильного приложения
для трекинга прогресса в изучении новых навыков «ProgressTracker»

09.03.02 Информационные системы и технологии
Информационные системы и сетевые технологии

Зав. кафедрой _____ Борисов Д.Н., к.т.н., доцент
Обучающийся _____ Дуреева Е.Ю., 3 курс, д/о
Обучающийся _____ Лагонская А.А., 3 курс, д/о
Руководитель _____ Тарасов В.С., ст. преподаватель

Воронеж 2024

Содержание

Введение	4
1 Постановка задачи.....	5
1.1 Цели создания приложения	5
1.2 Задачи приложения	5
1.3 Требования к приложению	5
1.3.1 Требования к приложению в целом	5
1.3.2 Требования к функциям (задачам), выполняемым приложением	6
1.3.3 Требования к программному обеспечению	7
1.3.4 Требования к оформлению и верстке страниц	7
1.3.5 Требования к защите информации	8
1.4 Задачи, решаемые в процессе разработки.....	8
2 Анализ предметной области	10
2.1 Термины и обозначения	10
2.2 Обзор аналогов	11
2.2.1 Цели и достижения.....	12
2.2.2 TheFor: Habit Tracker	14
3 Графическое описание системы	17
3.1 Диаграммы прецедентов	17
3.1.1 Диаграмма прецедентов для неавторизованного пользователя.....	17
3.1.2 Диаграмма прецедентов для авторизованного пользователя	18
3.1.3 Диаграмма прецедентов для администратора	19
3.2 Диаграмма активности	20
3.3 Диаграмма последовательности	21
3.4 Диаграмма состояния	27
3.5 Диаграмма классов	28
3.6 Диаграмма развёртывания	29
4 Реализация	30
4.1 Средства реализации	30
4.2 Реализация базы данных	31
4.3 Реализация клиентской части	31

4.4 Реализация интерфейса	32
4.5 Реализация серверной части	32
5 Тестирование	33
Заключение	34

Введение

В современном мире все больше людей стремятся к саморазвитию и достижению поставленных целей. Однако, часто на этом пути возникают преграды в виде неорганизованности, недостатка мотивации и нечетко поставленных задач. В такой ситуации эффективный инструмент для трекинга и управления целями становится незаменимым.

Наше приложение для трекинга выполнения целей призвано стать надежным спутником на пути к достижению успеха. Оно предоставляет пользователю удобные инструменты для постановки целей, их систематического отслеживания и мотивации. Мы уверены, что эффективное планирование и контроль прогресса являются ключевыми составляющими успешного достижения целей.

Актуальность данного проекта обусловлена растущим спросом на инструменты саморазвития и управления личным временем в современном обществе.

Целью курсового проекта является разработка инновационного мобильного приложения, которое поможет пользователям управлять своими целями эффективно и результативно.

1 Постановка задачи

1.1 Цели создания приложения

Целями создания приложения являются:

- Реализация приложения, которое позволит пользователям отслеживать прогресс изучения новых навыков и достигать своих целей;
- Предоставление возможности делиться своими путями достижения целей с другими пользователями, что позволит расширить пользовательскую базу.

1.2 Задачи приложения

Приложение позволяет решать следующие задачи:

- Создавать, редактировать, удалять и предоставлять доступ к своей цели;
- Добавлять, редактировать, удалять и отмечать выполненные этапы цели;
- Просматривать цели, которыми поделились другие пользователи, и добавлять их к себе;
- Просматривать количество текущих, просроченных и завершённых целей;
- Осуществлять редактирование данных своего аккаунта после авторизации или регистрации в системе.

1.3 Требования к приложению

1.3.1 Требования к приложению в целом

Пользователь использует клиентское приложение для взаимодействия с серверной частью. Серверная часть приложения возвращает необходимые данные пользователю через REST API.

Серверная часть приложения включает в себя:

- Сервис регистрации и авторизации;
- API сервис;
- Сервис для работы с базой данных.

Приложение должно включать в себя базу данных, где будут храниться данные о пользователях, их этапы к достижению целей, статистика и прогресс.

1.3.2 Требования к функциям (задачам), выполняемым приложением

Приложение должно соответствовать следующим функциональным требованиям.

Неавторизованный пользователь должен обладать возможностью:

- Авторизоваться/зарегистрироваться в приложении;
- Восстановить пароль.

Авторизованный пользователь должен обладать возможностью:

- Редактировать профиль, изменив фото профиля, имя, дату рождения, email;
- Просматривать списки текущих, просроченных, завершённых целей;
- Создавать цель, заполнив название, описание, дату завершения и добавив фото;
- Создавать этап цели, заполнив название, описание;
- Отмечать выполненные этапы цели;
- Редактировать цель, изменив фото, название, описание, дату окончания;

- Редактировать этап цели, изменив название, описание;
- Удалять цель и этапы цели;
- Изменять доступ цели с «индивидуального» на «общий»;
- Просматривать «общие» цели и добавлять их к себе;
- Выйти из системы.

Администратор должен обладать возможностью:

- Удалять пользователей;
- Удалять «общие» цели пользователей.

1.3.3 Требования к программному обеспечению

Требования к программному обеспечению клиентской части:

Приложение должно корректно работать на устройствах, работающих на операционной системе Android 7.0 и новее.

Для реализации серверной части приложения будут использоваться следующие средства:

- Язык программирования Python;
- Фреймворк Django;
- СУБД PostgreSQL.

Для реализации клиентской части приложения будут использоваться следующие средства:

- Язык программирования Dart;
- Фреймворк Flutter.

1.3.4 Требования к оформлению и верстке страниц

Оформление и верстка страниц должны удовлетворять следующим требованиям:

- Приложение должно быть оформлено в едином стиле;

- Должно быть разработано название приложения и логотип, которые будут применяться в оформлении страниц;
- Приложение должно иметь единую цветовую палитру и ограниченный набор шрифтов;
- Необходимо корректное отображение страниц на дисплеях различного размера.

1.3.5 Требования к защите информации

Для предотвращения SQL-инъекции будет использован Django ORM, который с помощью механизма подготовленных запросов автоматически экранирует данные, предотвращая внедрение вредоносного SQL-кода. Параметры запросов обрабатываются таким образом, что они рассматриваются как данные, а не как потенциальный SQL-код.

1.4 Задачи, решаемые в процессе разработки

Были поставлены следующие задачи:

- Анализ предметной области;
- Обзор аналогов;
- Создание репозитория GitHub;
- Распределение задач в task-менеджере Trello;
- Создание проекта Miro с UML диаграммами;
- Разработка требований к приложению в общем, его функциям, программному обеспечению, оформлению и вёрстке страниц, к защите информации;
- Разработка дизайна приложения в Figma;
- Написание технического задания в соответствии с ГОСТ 34.602 – 2020;

- Реализация интерфейса приложения;
- Реализация серверной части приложения;
- Развертывание приложения, его отладка и доработка;
- Написание курсового проекта.

2 Анализ предметной области

2.1 Термины и обозначения

В настоящей работе используются следующие термины и сокращения с соответствующими определениями:

- ProgressTracker: Условное обозначение приложения;
- REST API: Интерфейс приложения, который позволяет взаимодействовать с сервером и получать необходимые данные;
- СУБД PostgreSQL: Система управления базами данных PostgreSQL, используемая для хранения информации о пользователях, их целях, прогрессе и статистике;
- Фреймворк Django: Фреймворк, написанный на языке программирования Python, используемый для разработки серверной части веб-приложений;
- Фреймворк Flutter: Фреймворк для создания кроссплатформенных мобильных приложений, разработанный компанией Google, использующий язык программирования Dart;
- Task-менеджер Trello: Веб-приложение для управления задачами и проектами, использующее карточки и доски для организации работы;
- Git: Система контроля версий, используемая для отслеживания изменений в коде приложения и совместной работы над проектом;
- Miro: Инструмент для создания диаграмм, схем и других визуальных элементов, используемый для проектирования и планирования работы;
- Figma: Инструмент для дизайна интерфейсов и прототипирования приложений;
- Репозиторий на GitHub: Хранилище для кода проекта, используемое для совместной работы над проектом и отслеживания изменений;

— UML диаграммы: Визуальные средства для моделирования структуры и поведения программных систем, используемые для проектирования приложения;

— Техническое задание: Документ, который описывает требования и спецификации к создаваемому продукту, в данном случае, к приложению;

— Курсовой проект: Проект, выполняемый в рамках обучения студентами на определенном курсе или предмете, в данном случае, это приложение;

— Языковые версии: Различные версии приложения, предназначенные для использования на разных языках. В данном случае указывается, что приложение должно быть доступно на русском языке;

— Неавторизованные пользователи: Пользователи, которые ещё не прошли процедуру аутентификации в системе;

— Авторизованные пользователи: Пользователи, которые прошли процедуру аутентификации и имеют доступ к определенным функциям и ресурсам приложения;

— «Индивидуальные» цели: Цели, доступ к которым есть только у пользователя, который их создал;

— «Общие» цели: Цели, которые доступны всем пользователям, потому что пользователи, которые их создали, предоставили к ним доступ.

2.2 Обзор аналогов

На рынке есть множество приложений для автоматизации отслеживания прогресса своих целей. Далее мы рассмотрим некоторые из популярных решений, их достоинства и недостатки, чтобы разработать такой функционал

приложения, который сможет удовлетворить потребности пользователей уже существующих решений.

2.2.1 Цели и достижения

«Цели и достижения» – мобильное приложение для отслеживания прогресса выполнения целей и отслеживания их результата. Данное решение предоставляет возможность ставить свои цели на различные периоды выполнения и отмечать их выполнение. Кроме этого, есть возможность вести дневник цели с добавлением текстовых заметок и изображений.

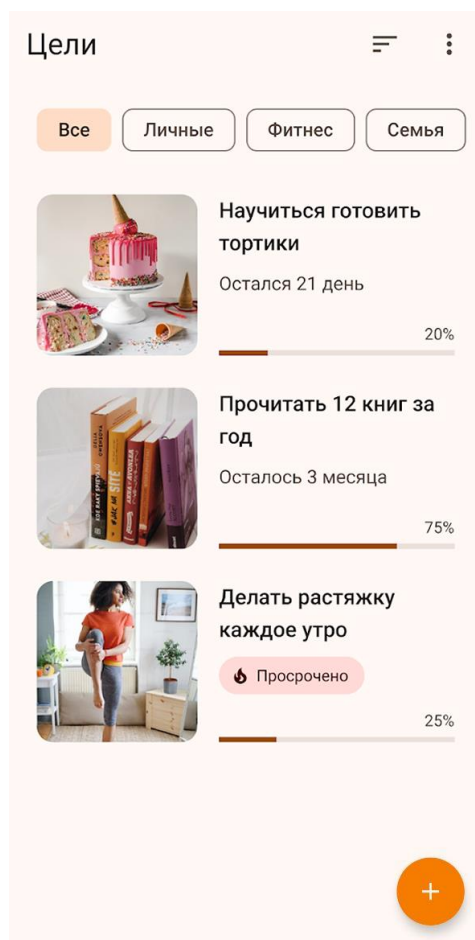


Рисунок 1 - Главная страница целей

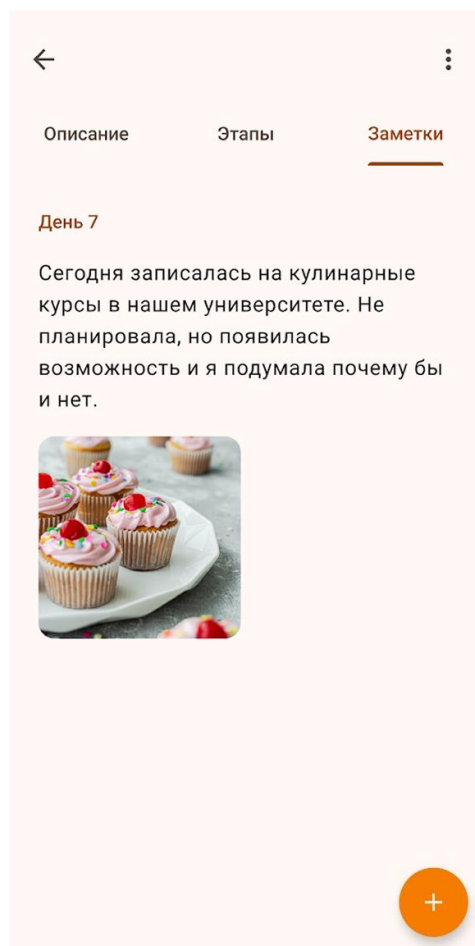


Рисунок 2 - Страница дневника цели

Достоинства:

- Простой интерфейс;
- Возможность устанавливать конкретные цели, разбивать их на подцели и отслеживать прогресс в их достижении без регистрации;
- Присутствует краткий обзор на функционал приложения при первом запуске;
- Возможность добавления изображений к целям, описания мотивации и награды;
- Возможность смены визуальной темы приложения;
- Возможность ведения дневника цели с добавлением заметок к прогрессу.

Недостатки:

- Отсутствие возможности сделать резервную копию целей при помощи авторизации;
- При завершении цели нет индикатора о её выполнении, что может привести к ощущению незаконченности процесса;
- Отсутствие функции сортировки целей по критериям, кроме тематических категорий;
- Приложение предоставляет площадку исключительно для самостоятельного добавления целей без возможности публикации или использования подборок других пользователей;
- Ограниченный функционал бесплатной версии.

2.2.2 TheFor: Habit Tracker

«TheFor: Habit Tracker» также является мобильным приложением для трекинга выполнения целей, призванное помогать сформировать полезные привычки. Пользователи могут создавать список своих привычек, устанавливать их частоту и отслеживать выполнение каждой привычки в течение определенного периода времени. Имеет более расширенный функционал в сравнении с приложением «Цели и достижения».

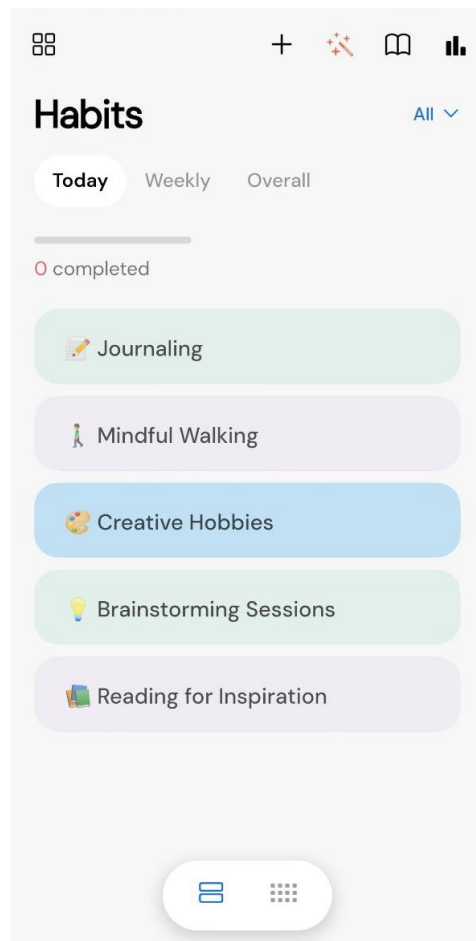


Рисунок 3 - Главная страница привычек

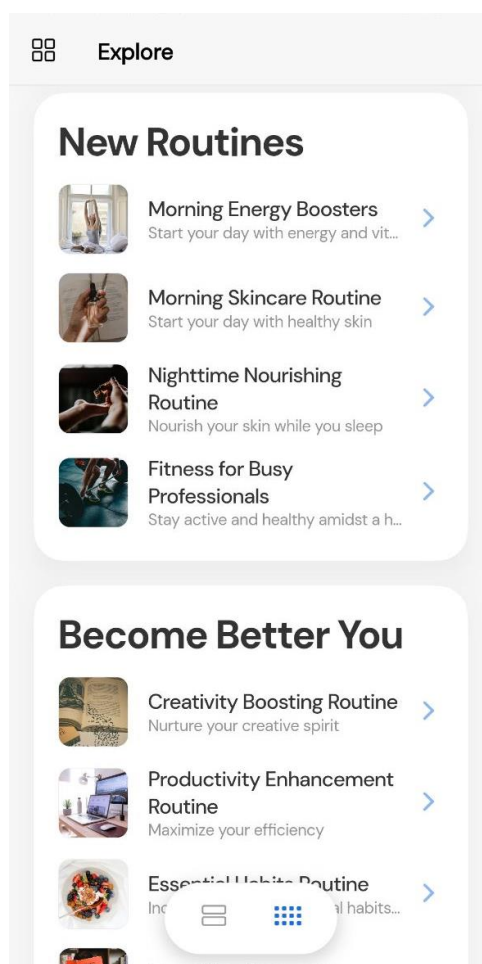


Рисунок 4 - Страница готовых подборок

Достоинства:

- Возможность разбиения целей на различные промежутки времени;
- Разнообразие готовых подборок привычек;
- Возможность индивидуального выбора целей из подборок;
- Возможность простого отслеживания выполнения\пропуска цели;
- Возможность персонализации цветовой схемы приложения;
- Возможность визуализации прогресса в виде графиков и статистики выполнения привычек;
- Наличие напоминания о выполнении привычек и мотивационные стимулы для поддержания мотивации;
- Возможность создания резервной копии через файл.

Недостатки:

- Фокусировка на рутинности целей, отсутствие разбиения крупных целей на этапы;
- Непонятный, сложный для восприятия интерфейс;
- Отсутствие поддержки русского языка;
- Навязчивая реклама платной подписки;
- Ограниченность функций бесплатной версии.

3 Графическое описание системы

3.1 Диаграммы прецедентов

Диаграмма прецедентов демонстрирует сценарии использования, возникающие при работе с приложением. Ниже приведены диаграммы для различных групп пользователей.

3.1.1 Диаграмма прецедентов для неавторизованного пользователя

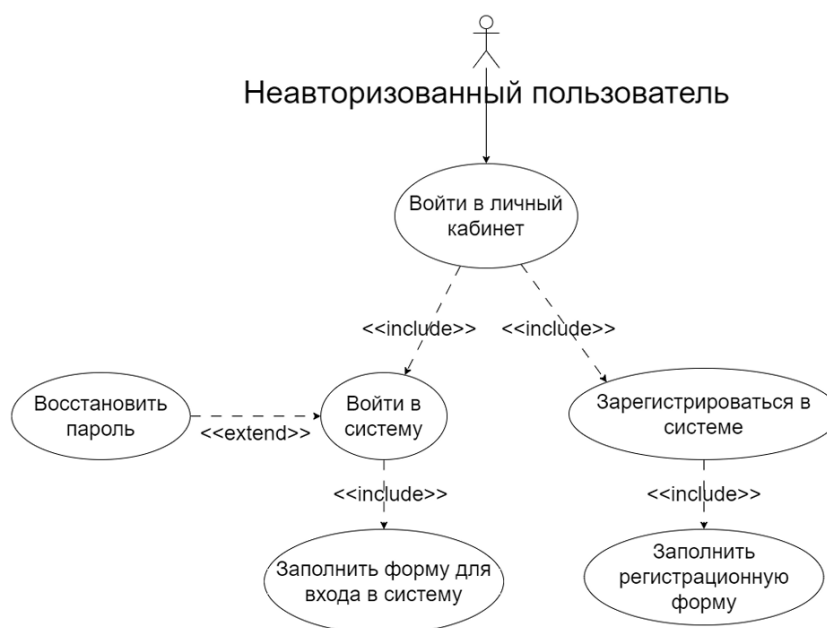


Рисунок 5 - Диаграмма прецедентов (неавторизованный пользователь)

Данная диаграмма описывает варианты сценариев, которые может выполнить неавторизованный пользователь:

- Вход в систему;
- Регистрация в системе.

3.1.2 Диаграмма прецедентов для авторизованного пользователя

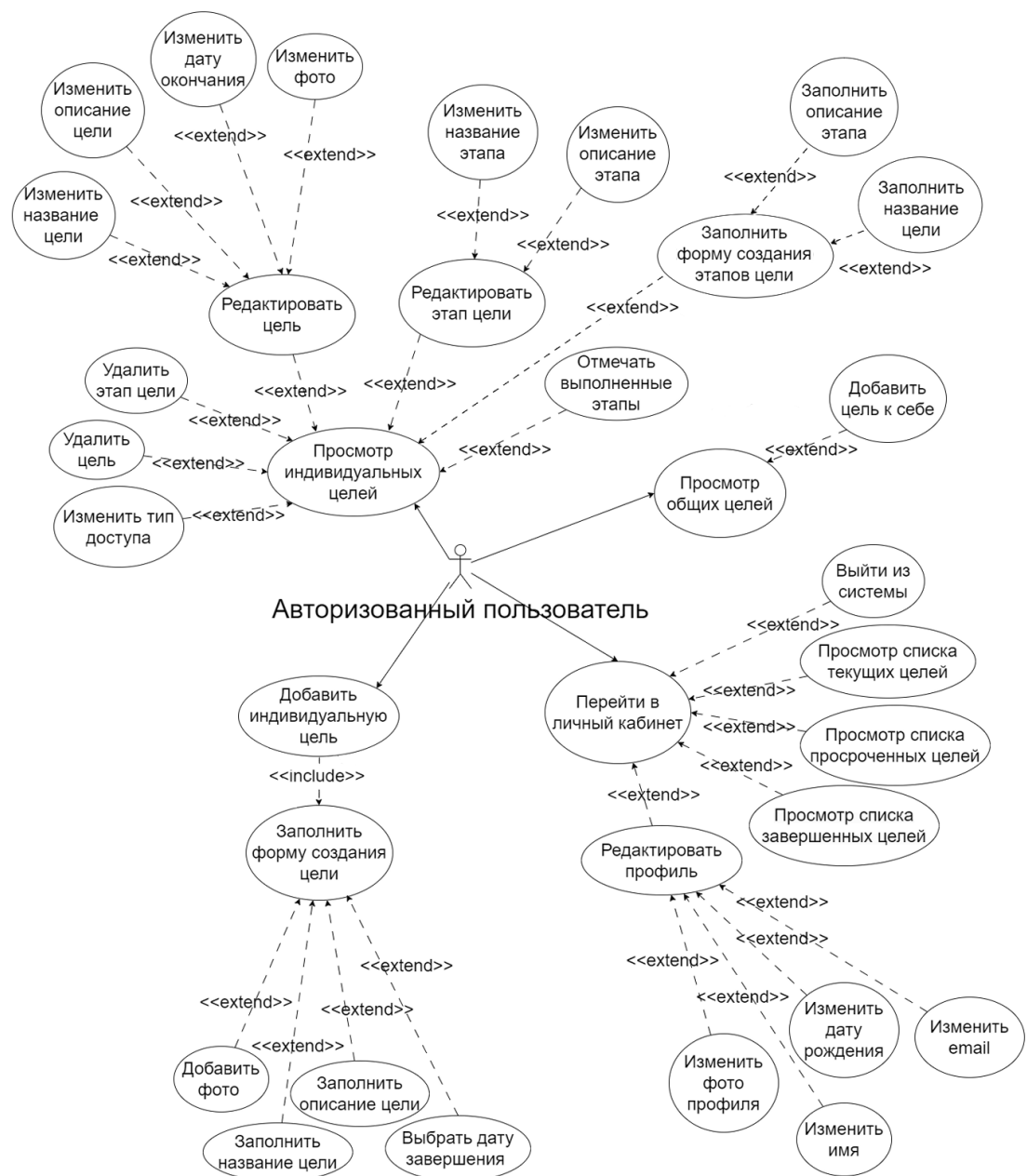


Рисунок 6 - Диаграмма прецедентов (авторизованный пользователь)

На данной диаграмме представлены все действия авторизованного пользователя, их возможные расширения и включения:

- Добавление индивидуальной цели;
- Просмотр индивидуальных целей;
- Просмотр общих целей;
- Переход в личный кабинет.

При добавлении индивидуальной цели пользователь заполняет форму для её создания с возможностью указать название, описание, дату завершения цели и добавить фото.

При сценарии просмотра индивидуальных целей приложение позволяет редактировать цели и их этапы, изменять тип доступа цели, создавать этапы цели и отмечать их выполненными, удалять цели и их этапы.

При просмотре общих целей пользователь может добавить конкретную цель к себе.

При переходе в личный кабинет у пользователя есть возможность редактирования профиля, просмотра списка текущих, просроченных и завершённых целей, а так же возможность выхода из системы.

3.1.3 Диаграмма прецедентов для администратора

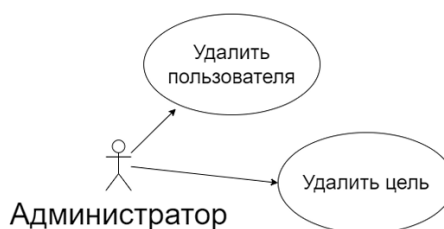


Рисунок 7 - Диаграмма прецедентов (администратор)

На данной диаграмме показано, какие сценарии может выполнить администратор:

- Удаление пользователя;

— Удаление цели.

3.2 Диаграмма активности

Диаграмма активности используется для моделирования процессов и последовательности действий, выполняемых при взаимодействии пользователя и системы.

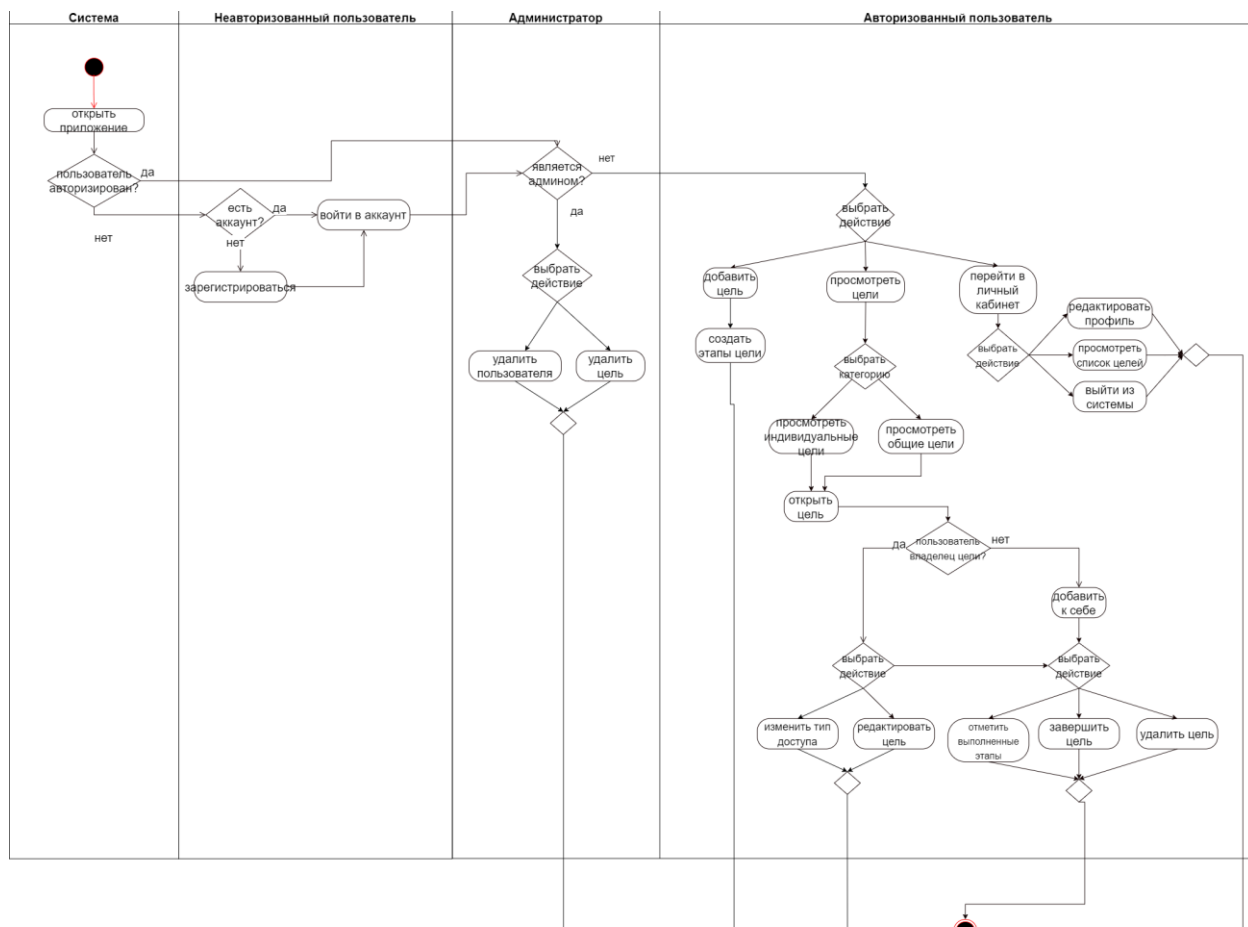


Рисунок 8 - Диаграмма активности

На данной диаграмме можно видеть, что пользователь может открыть приложение, и если он не авторизован, то ему будет необходимо войти в систему или зарегистрироваться. После этих действий последовательность, которые будут происходить.

3.3 Диаграмма последовательности

Диаграмма последовательности позволяет описать последовательность происходящих событий при взаимодействии пользователя с системой.

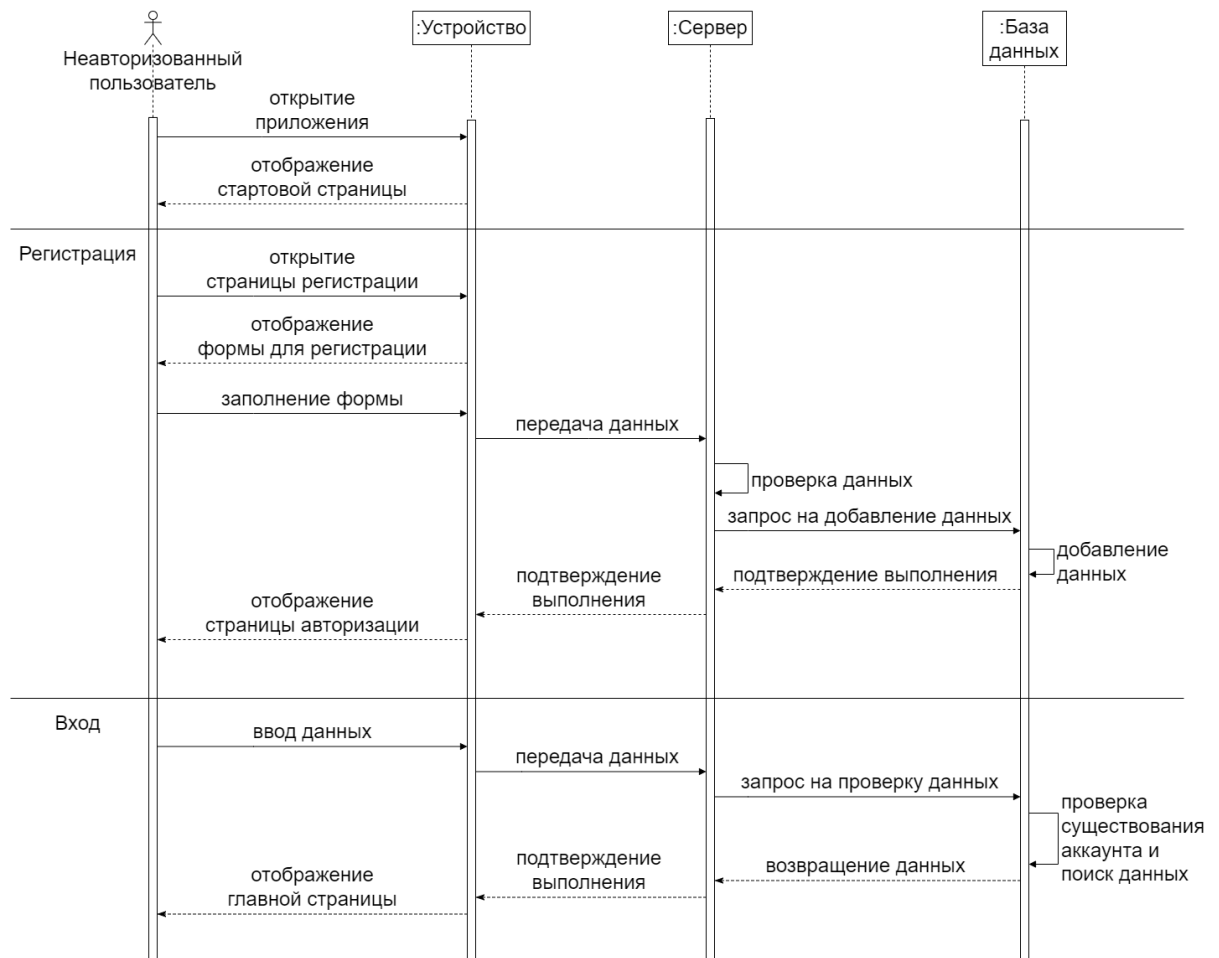


Рисунок 9 - Диаграмма последовательности (неавторизованный пользователь)

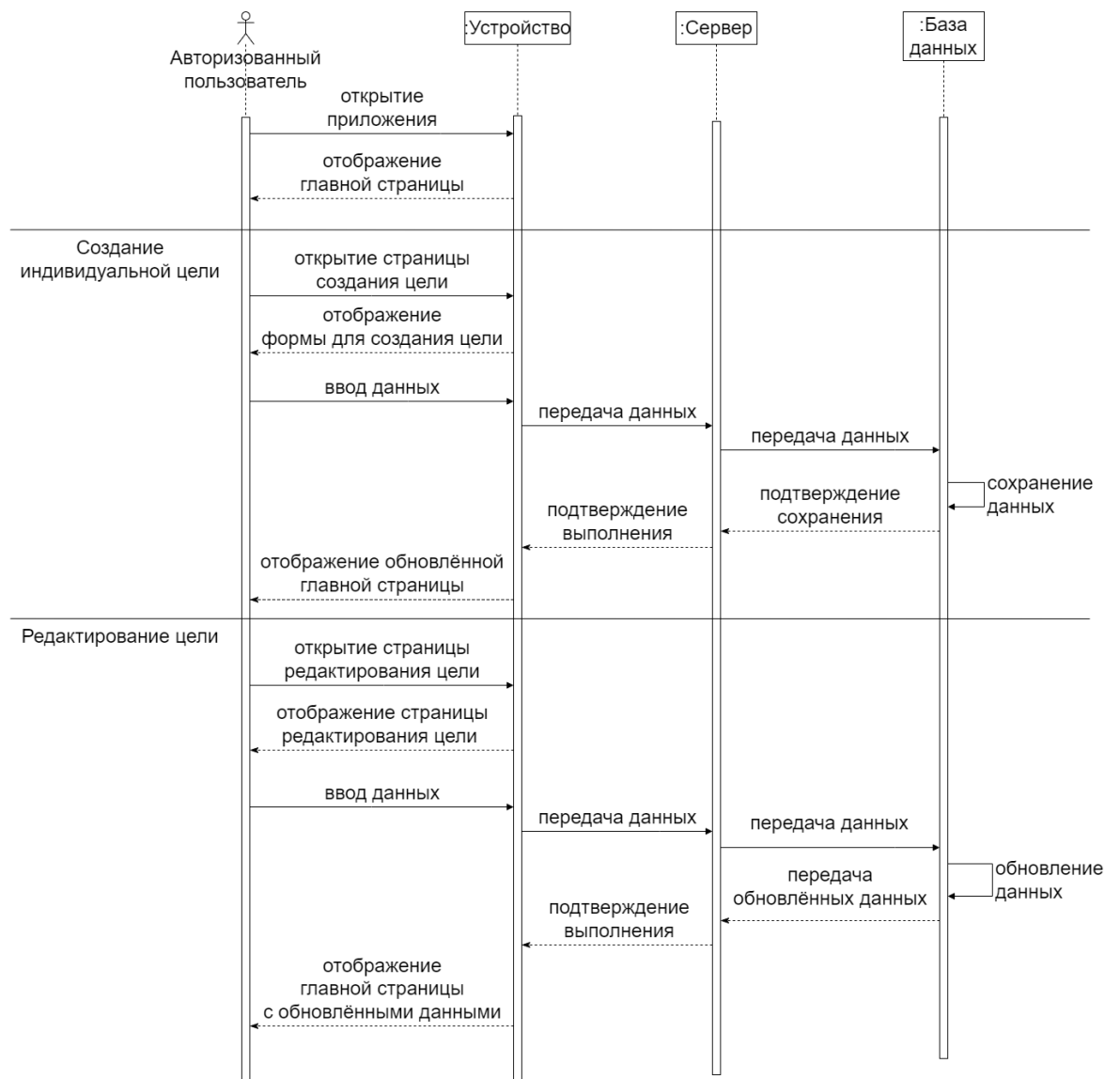


Рисунок 10 - Диаграмма последовательности (авторизованный пользователь)

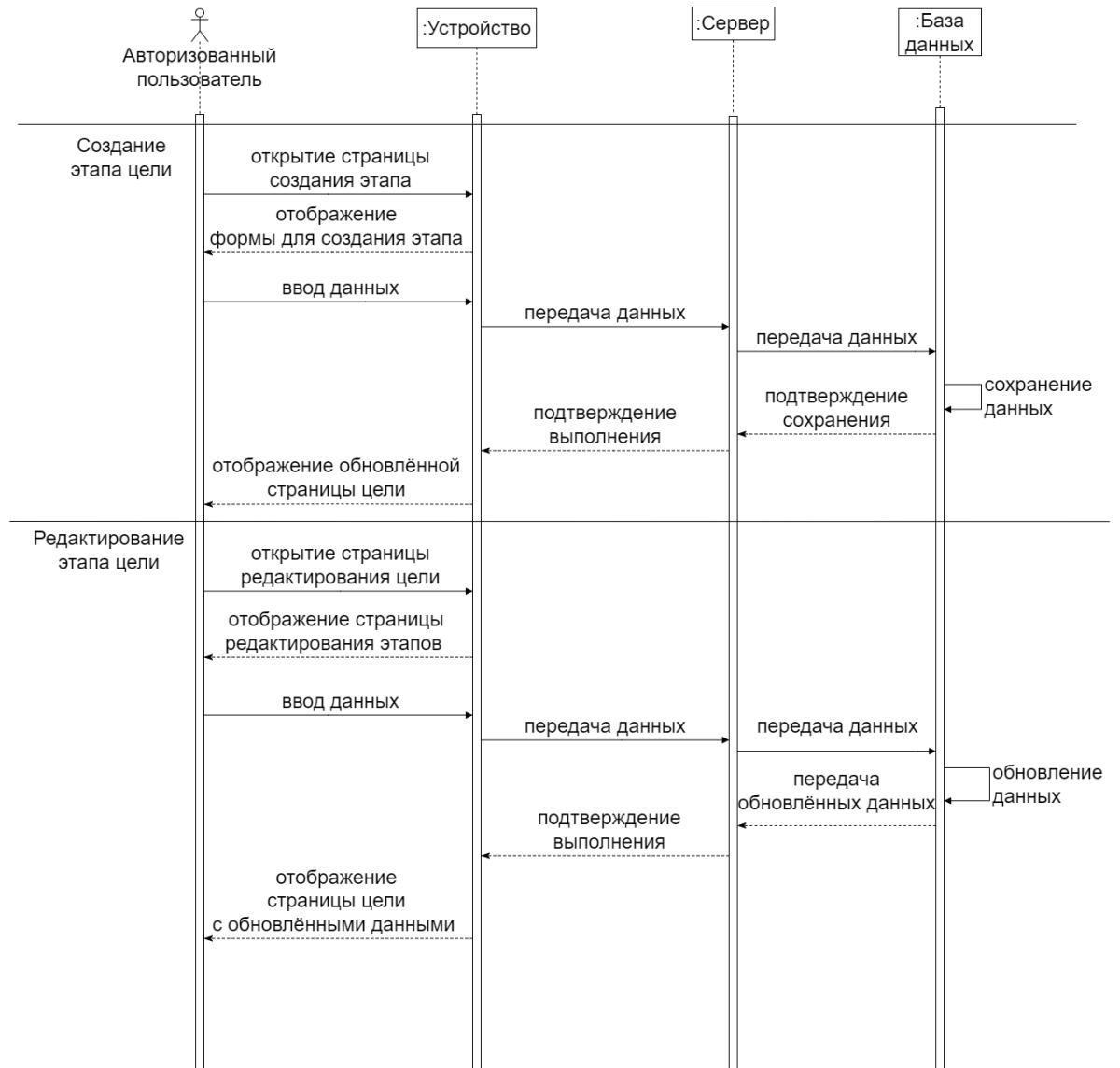


Рисунок 11 - Диаграмма последовательности (авторизованный пользователь)

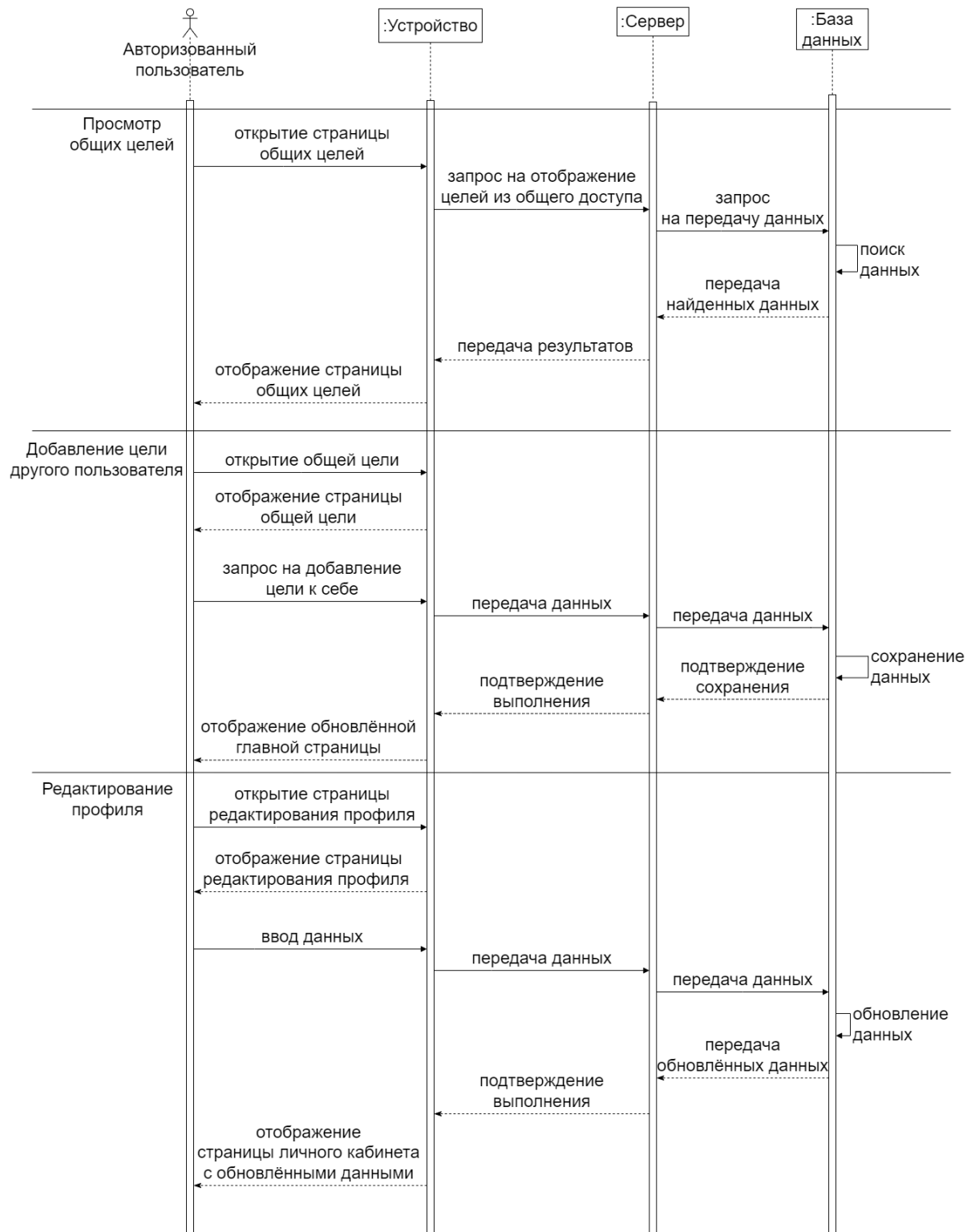


Рисунок 12 - Диаграмма последовательности (авторизованный пользователь)

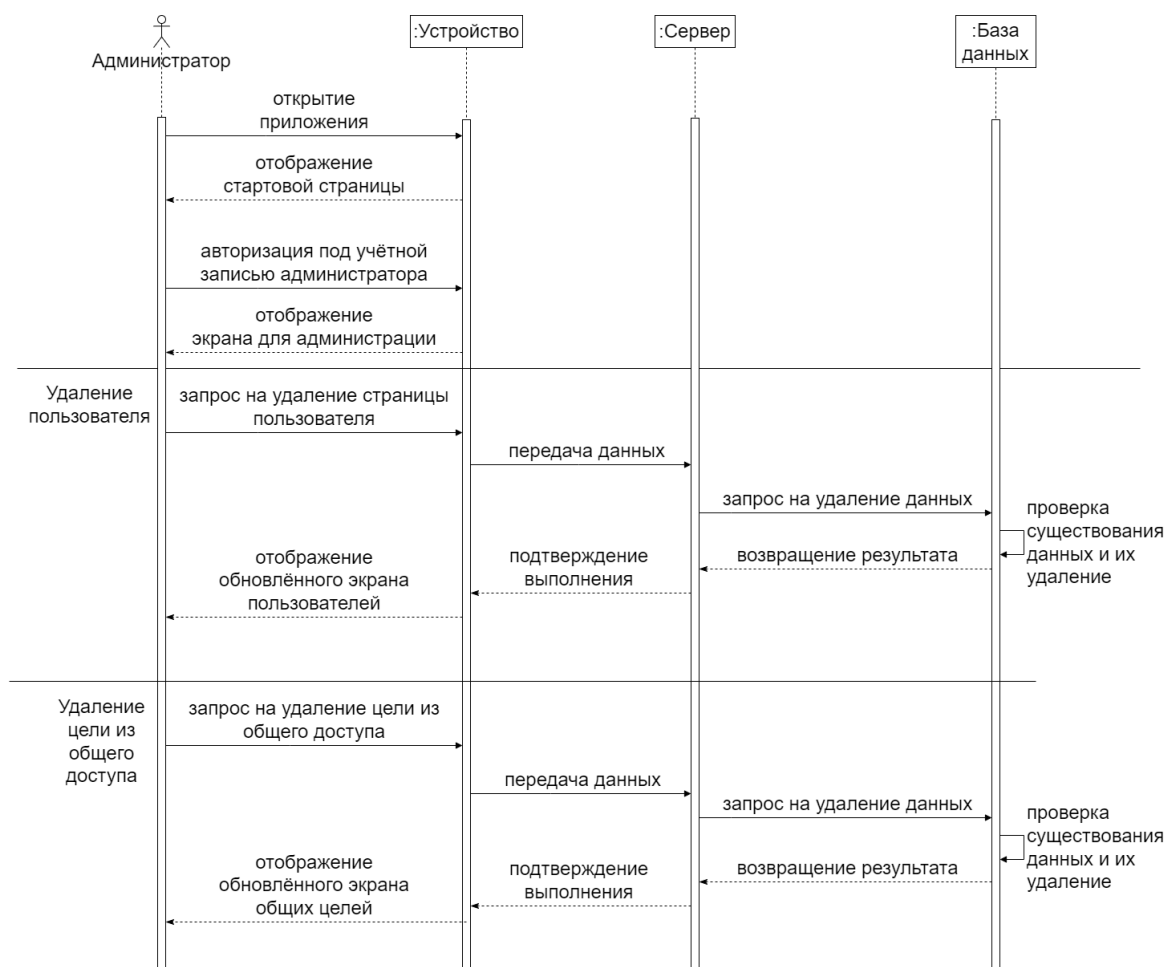


Рисунок 13 - Диаграмма последовательности (администратор)

При запуске приложения открывается главная страница. Неавторизованный пользователь следует по одной из двух последовательностей действий: совершает вход или регистрацию в системе.

При регистрации происходит открытие окна регистрации, затем введённые данные передаются на сервер, после чего отображается ответ сервера.

При авторизации открывается страница входа, после чего данные из формы отправляются на сервер, который осуществляет их проверку. Если проверка успешно пройдена, то отображается страница приложения, иначе происходит повторная попытка ввода.

Далее авторизованный пользователь имеет различные варианты взаимодействия с приложением.

При создании цели происходит отображение формы для добавления новой цели, после ввода данных они отправляются на сервер, проходят проверку и новая цель отображается на странице целей.

При запросе на редактирование цели отображается страница редактирования с полями для ввода данных, после изменения информации, данные обновляются на сервере и отображаются на странице цели.

При создании этапов цели открывается страница для добавления этапов, после ввода данных они проходят проверку и сохраняются на сервере, а затем отображаются на странице просмотра этапов цели.

При запросе на редактирование этапа цели отображается форма для редактирования данных, после изменения они отправляются на сервер,

При просмотре общих целей запрашиваются данные конкретной общедоступной цели на сервере, после их получения, цель отображается на странице.

При добавлении цели другого пользователя на сервер отправляется запрос на сохранение цели в профиле пользователя. После чего цель отображается на главной странице целей.

При редактировании профиля пользователя текущие данные пользователя запрашиваются на сервере, после чего отображаются на странице профиля. Каждый раз новые данные отправляются на сервер, который, после их обновления, присылает соответствующий ответ. Затем отображается страница профиля с обновлёнными данными.

Если авторизованный пользователь является администратором, ему доступны две последовательности: удаление пользователя и удаление цели из общего доступа.

При запросе на удаление пользователя происходит запрос данных о пользователе на сервере, затем они проверяются и удаляются из базы данных, после чего отображается обновлённая страница пользователей.

При удалении цели из общего доступа запрашиваются данные на сервере о конкретной цели и после проверки они удаляются. Затем на странице целей отображаются обновлённые данные.

3.4 Диаграмма состояния

Диаграмма состояния используется для моделирования состояния системы в различных случаях. Эта диаграмма показывает реакцию системы на внешние события и изменение её состояния в соответствии с ними.

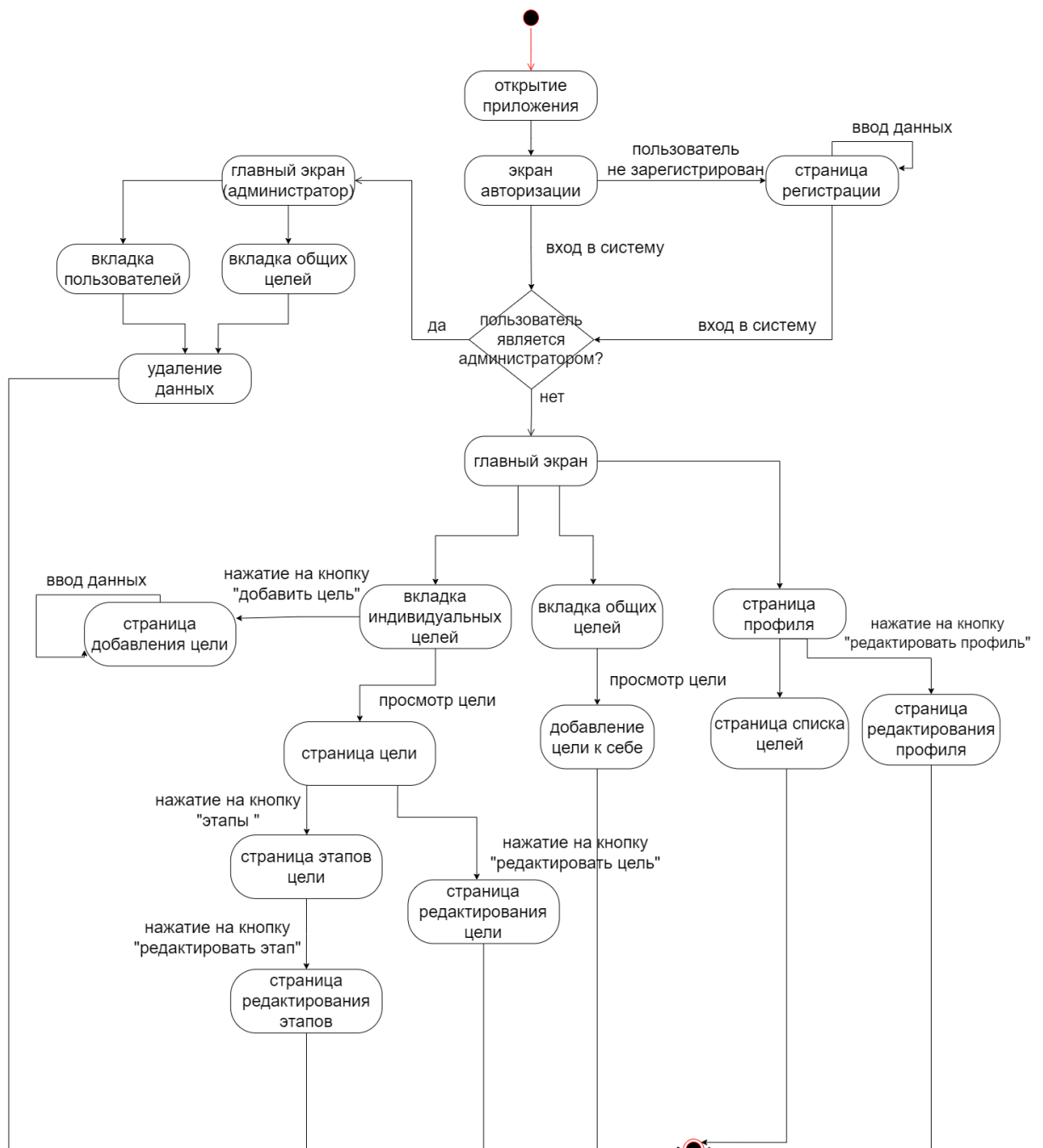


Рисунок 14 - Диаграмма состояния

3.5 Диаграмма классов

Диаграмма классов используется для моделирования структуры классов в системе и их взаимоотношений.

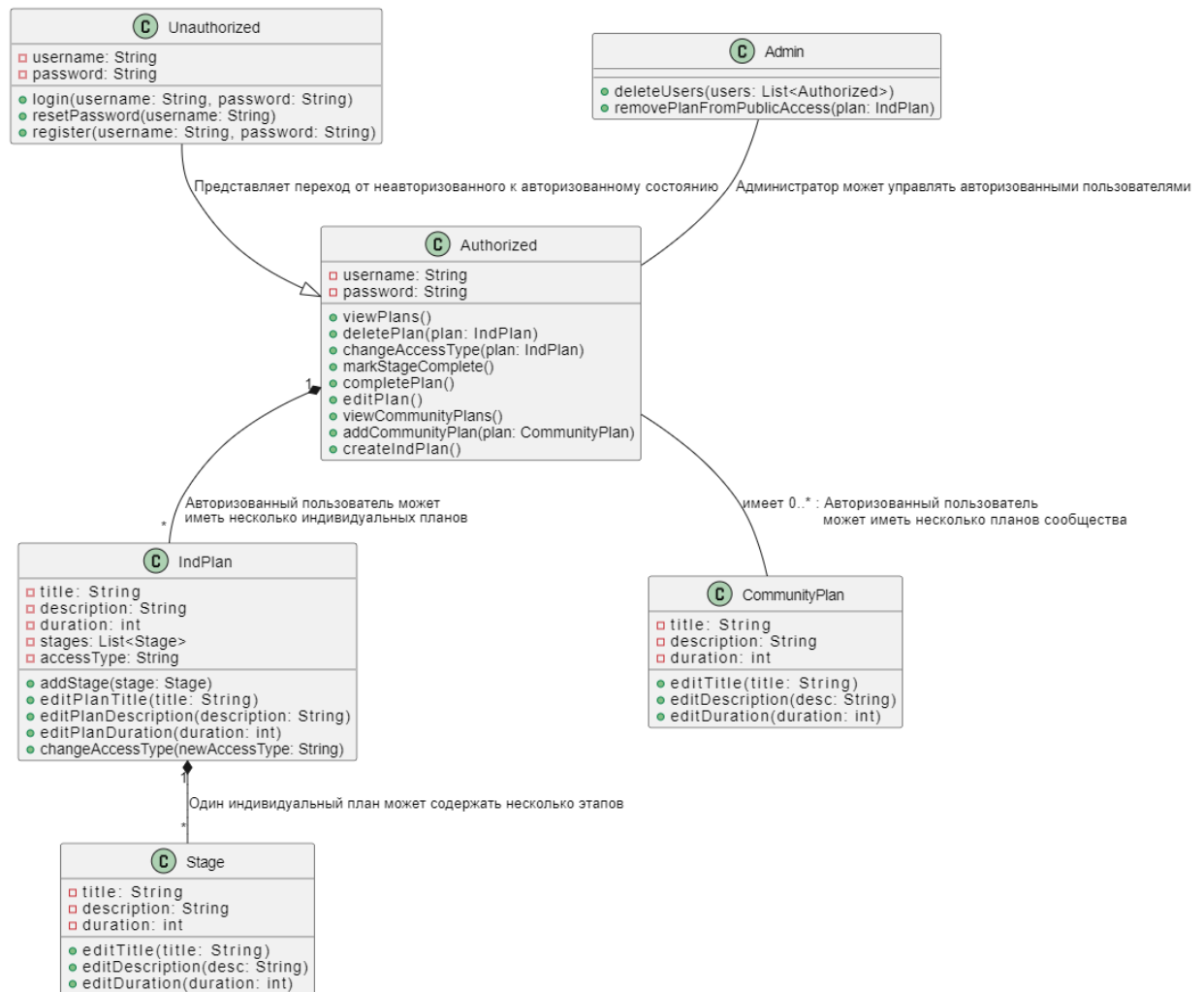


Рисунок 15 - Диаграмма классов

3.6 Диаграмма развёртывания

Диаграмма развёртывания используется для моделирования архитектуры системы, она показывает, как компоненты системы взаимодействуют между собой.

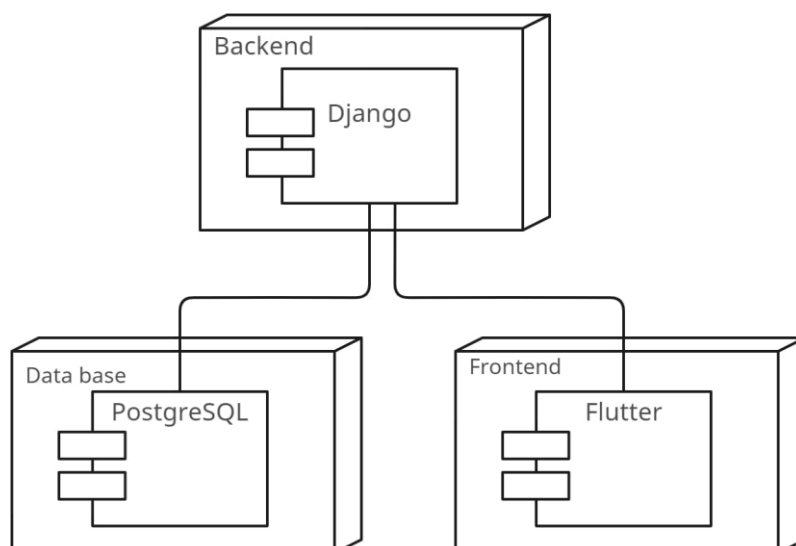


Рисунок 16 - Диаграмма развёртывания

4 Реализация

4.1 Средства реализации

Мобильное приложение использует архитектуру, которая разделяется на две основные части: клиентскую и серверную.

Пользователь использует клиентское приложение для взаимодействия с серверной частью. Серверная часть приложения возвращает необходимые данные пользователю через REST API.

Для реализации серверной части были выбраны следующие технологии:

- язык программирования Python;
- фреймворк Django;
- СУБД PostgreSQL.

Фреймворк Django и язык программирования Python были выбраны для реализации серверной части, поскольку обладают рядом преимуществ:

— Административная панель Django оптимизирует выполнение административных задач, особенно в работе с большим объемом данных;

— Для языка Python разработано большое количество библиотек, которые могут предложить различные готовые решения.

Для реализации клиентской части были выбраны следующие технологии:

— язык программирования Dart;

— фреймворк Flutter.

Фреймворк Flutter и язык программирования Dart были выбраны для реализации клиентской части, потому что:

— Dart предоставляет возможность мгновенного применения изменений в коде без необходимости полного перезапуска приложения;

— Flutter является кроссплатформенным фреймворком, позволяющим использовать программное обеспечение на разных операционных системах.

4.2 Реализация базы данных

Для хранения данных была выбрана PostgreSQL, база данных с открытым исходным кодом, которая широко поддерживается множеством серверов. PostgreSQL способна работать с различными типами данных, такими как числа с различной точностью, тексты с разными кодировками, изображения, аудиофайлы, видео, XML-документы, JSON-объекты и многие другие.

4.3 Реализация клиентской части

Для реализации клиентской части приложения были выбраны язык программирования Dart и фреймворк Flutter.

Клиентская часть приложения отвечает за пользовательский интерфейс и взаимодействие пользователя с данными.

Клиентская часть содержит:

- Файл `main.dart` – основной файл, определяющий навигацию и перемещение между экранами приложения;
- Каталог файлов `Pages` содержит в себе страницы приложения, включая подкаталоги `AuthorizationUser`, `UnauthorizationUser`, `Admin`;
- Папка `Constants` используется для хранения постоянных значений, которые используются в разных частях приложения;
- Каталог `Models` используется для хранения классов, которые описывают структуру данных в приложении;
- В папке `Widgets` хранятся блоки для описания пользовательского интерфейса.

4.4 Реализация интерфейса

Пользовательский интерфейс приложения реализован в соответствии с заявленным дизайном и необходимыми изменениями, внесенными в процессе разработки.

4.5 Реализация серверной части

Проект имеет следующую структуру:

- Файл `Settings.py`, содержащий настройки базы данных, настройки для подключения к ней, валидации полей, настройки JWT-токенов, секретное слово, список разрешённых хостов и т.д.

— Сериализаторы для подготовки данных к отправке и наоборот, для загрузки полученных данных на сервер.

— Файл маршрутизации (urls.py) – определяет пути к конечным точкам API. Также определены контроллеры для конечных точек.

5 Тестирование

В ходе разработки было проведено тестирование приложения и сделаны следующие выводы:

— Корректная работа всех функций: все функции и возможности приложения работают стабильно и соответствуют требованиям, указанным в техническом задании.

— Правильное функционирование интерфейса: все элементы интерфейса работают корректно и соответствуют дизайну.

— Корректное отображение компонентов: все компоненты на страницах приложения отображаются правильно и соответствуют требованиям дизайна.

— Сохранение и восстановление данных: приложение корректно сохраняет и восстанавливает данные пользователя при входе и выходе из аккаунта и смене пароля.

Заключение

В ходе разработки данного курсового проекта было реализовано мобильное приложение, удовлетворяющее поставленным задачам, а именно:

- Позволяет создавать, редактировать, удалять и предоставлять доступ к своей цели;

- Добавляет возможность создавать, редактировать, удалять и отмечать выполненные этапы цели;

- Реализует просмотр целей, которыми поделились другие пользователи, и возможность добавлять их к себе;

- Позволяет просматривать количество текущих, просроченных и завершённых целей;

- Предоставляет возможность осуществлять редактирование данных своего аккаунта после авторизации или регистрации в системе.

Разработанное приложение успешно прошло все тестирования на заявленные сценарии использования, тем самым подтверждая корректность своей работоспособности.