

Новосибирский Государственный Университет
Факультет Информационных Технологий

Техническое описание проекта по курсу ООАД

NoteHub

**Студенты ФИТ НГУ
Худорожков Ян Андреевич
Трайковская Екатерина Сергеевна
группа 21210**

Версия 1.0.0

Содержание

1. [Введение](#)
 - 1.1 [Цель](#)
 - 1.2 [Область действия](#)
 - 1.3 [Определения и сокращения](#)
 - 1.4 [Ссылки](#)
 - 1.5 [Краткое описание](#)
2. [Предметная область проекта](#)
 - 2.1 [Существующие проблемы](#)
 - 2.2 [Предполагаемое решение](#)
3. [Требования к программному решению](#)
 - 3.1 [Роли](#)
 - 3.2 [Функциональные требования для роли Роль1](#)
 - 3.2.1 [<Use Case Name 1>](#)
 - 3.2.2 [<Use Case Name 2>](#)
 - 3.3 [Функциональные требования для роли Роль2](#)
 - 3.3.1 [<Use Case Name 1>](#)
 - 3.3.2 [<Use Case Name 2>](#)
 - 3.4 [Нефункциональные требования](#)
4. [Обзор архитектуры](#)
 - 4.1.1 [Компонентная модель системы](#)
 - 4.1.1.1 [Компонент 1](#)
 - 4.1.1.2 [Компонент 2](#)
 - 4.1.2 [Компоненты сторонних производителей](#)
 - 4.1.3 [Схема развертывания приложения](#)
5. [Допущения и ограничения](#)
6. [Известные проблемы](#)
 - 6.1 [Невысокая производительность приложения](#)

Техническое описание проекта по курсу ООАД

1. Введение

1.1. Цель

Данный документ представляет собой техническое описание проекта NoteHub и содержит основные требования к разрабатываемой в рамках проекта программной системе и описание архитектуры программного решения.

1.2. Область действия

Документ разработан в рамках проекта NoteHub на основе стандартного шаблона и предназначен для использования студентами ФИТ и преподавателями дисциплины ООАД.

1.3. Определения и сокращения

Таблица 1: Определения и сокращения

Термин	Описание
Telegram бот	Telegram бот - это роботизированный аккаунт в мессенджере Telegram, который запрограммирован на автоматическое совершение действий
Директория	Директория (Каталог) - объект в файловой системе, упрощающий организацию файлов; позволяет сгруппировать файлы и, возможно, другие каталоги (для иерархических файловых систем)

1.4. Ссылки

В тексте содержатся ссылки на следующие документы:

1. <Имя файла документа>, v<версия> - <описание документа>

Ссылки приводятся в виде [N], где N – номер документа в вышеприведенном списке.

1.5. Краткое описание

Содержание данного документа построено таким образом, чтобы дать ответ на следующие вопросы:

- Какие проблемы предметной области должен решать будущий программный продукт
- Посредством какой функциональности системы будут достигнуто решение проблем предметной области
- Какова архитектура программного решения

Описание предметной области и проблем, для решения которых предназначен будущий программный продукт, приведены в разделе 2.

Раздел 3 содержит описание требований к программному решению, раздел 4 – описание архитектуры выбранного решения.

2. Предметная область проекта

В современном мире через людей каждый день проходит огромный поток информации. Очень сложно все запомнить, поэтому многие используют такой инструмент для ведения записей, как заметки. Люди ищут удобные способы для хранения, структурирования и передачи информации. Однако быстрый доступ, мобильность, возможность передачи и удобный поиск предоставить в одном сервисе довольно сложно.

2.1. Существующие проблемы

Многие сталкиваются с необходимостью быстро и удобно записать какую-либо информацию. Также периодически возникает необходимость поделиться своими заметками с другими людьми. Сторонние приложения не могут в удобном формате и на высоком уровне удовлетворить эти потребности. К тому же их использование может занимать относительно много времени из-за необходимости установки и поиска способа для передачи другим контактам.

2.2. Предполагаемое решение

Мы предлагаем создать Telegram-бота, который будет решать эти проблемы. Пользователи смогут быстро и удобно создавать заметки в чате с ботом. Он предоставит пользователю возможность структурировать свои записи. Также функционал Telegram'a позволит легко делиться своими заметками с другими контактами. Функция поиска поможет пользователям быстро находить нужную информацию. Важным преимуществом является то, что бот будет доступен на любом устройстве, где установлен Telegram. Пользователи смогут получать доступ к своим записям и редактировать их в любое время и в любом месте. Наш бот станет отличным помощником в повседневной жизни для каждого человека.

3. Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта NoteHub.

3.1. Роли

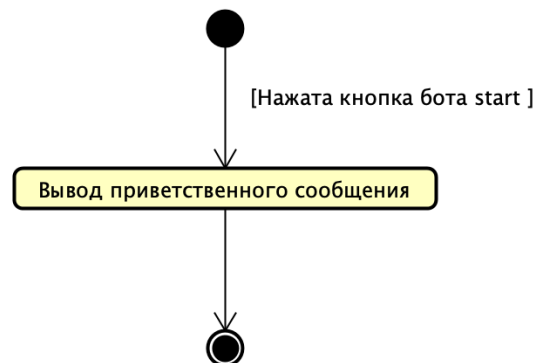
Роль - это что-то (например: другая система) или кто-то (например: человек) вне системы, которые взаимодействуют с ней. В предлагаемой к разработке системе идентифицированы следующие роли:

1. Неавторизованный пользователь - пользователь, который еще не осуществил аутентификацию в боте. Не может пользоваться услугами, предоставляемыми ботом.
2. Авторизованный пользователь - пользователь, имеющий доступ ко всем предоставляемым услугам бота с возможностью просмотра и редактирования своих записок, а также возможностью поделиться запиской

3.2. Функциональные требования для роли Неавторизованный пользователь

3.2.1. Авторизация

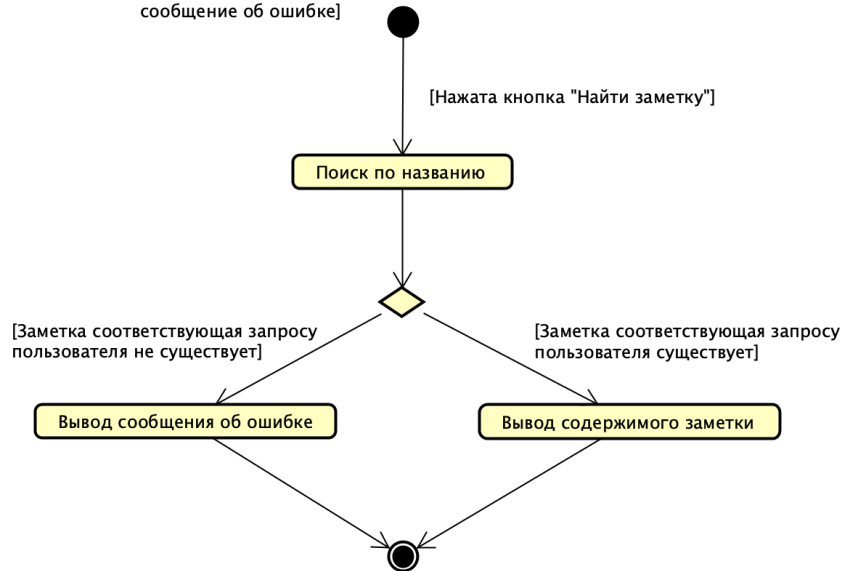
[Пользователь через поиск в Telegram или по ссылке находит бота и начинает с ним взаимодействие путем нажатия кнопки Старт.]



3.3. Функциональные требования для роли Авторизованный пользователь

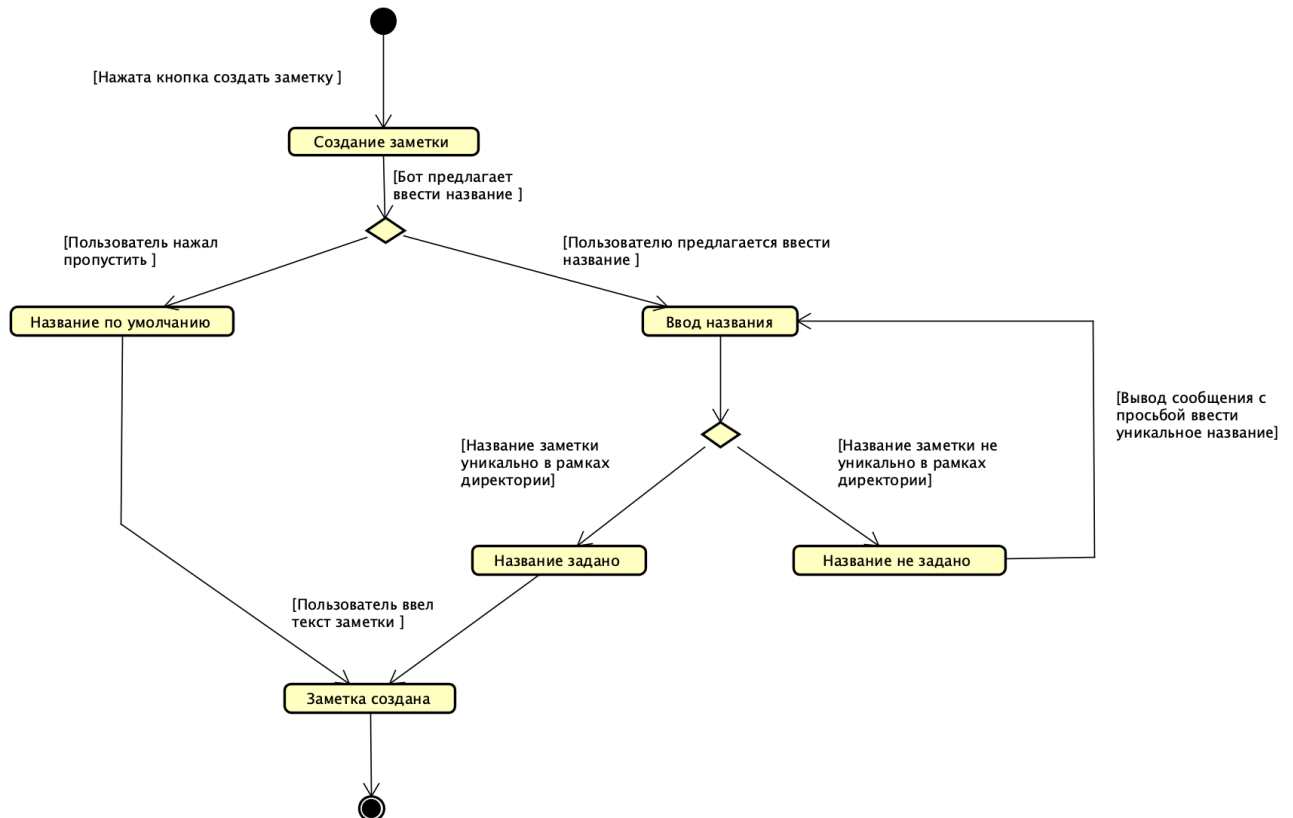
3.3.1. Совершить поиск по названию записки

[Для поиска заметки по названию пользователю необходимо нажать соответствующую кнопку в корневой директории и ввести название заметки. После чего ему будут предоставлены результаты поиска. Если заметки с данным названием не существует, то выводится сообщение об ошибке]

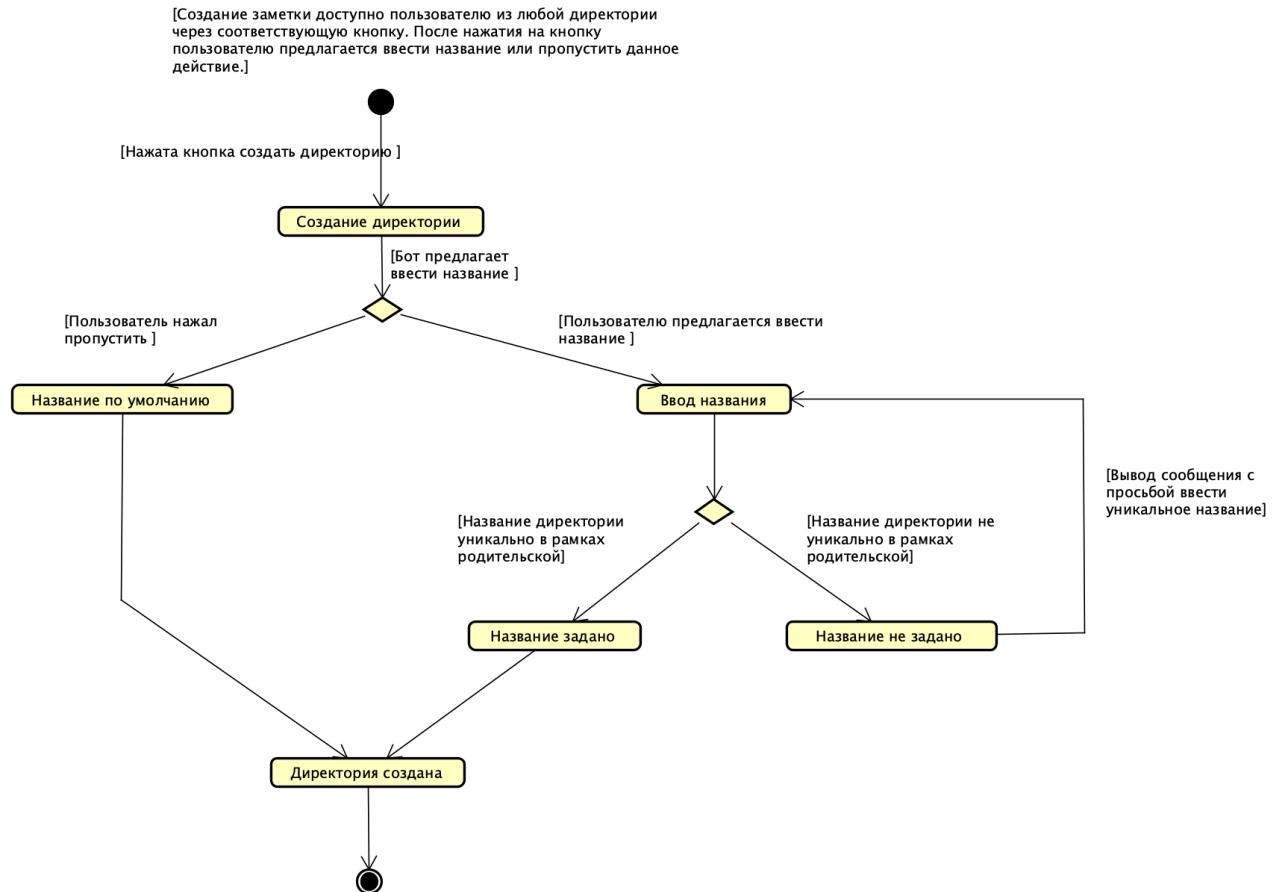


3.3.2. Создать заметку

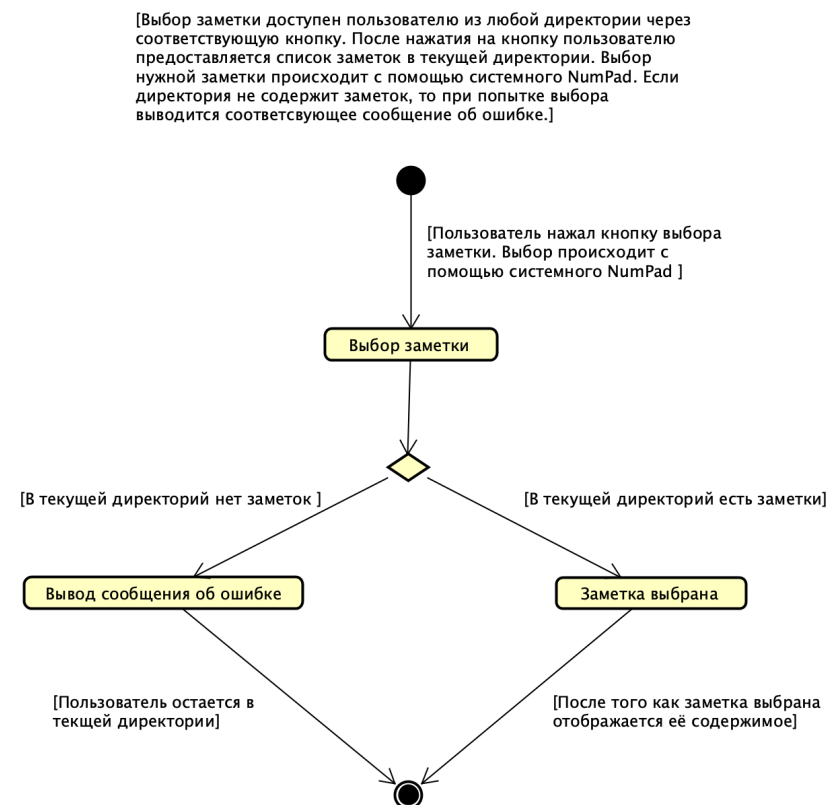
[Создание заметки доступно пользователю из любой директории через соответствующую кнопку. После нажатия на кнопку пользователю предлагается ввести название или пропустить данное действие. После этого пользователь вводит содержание заметки.]



3.3.3. Создать директорию

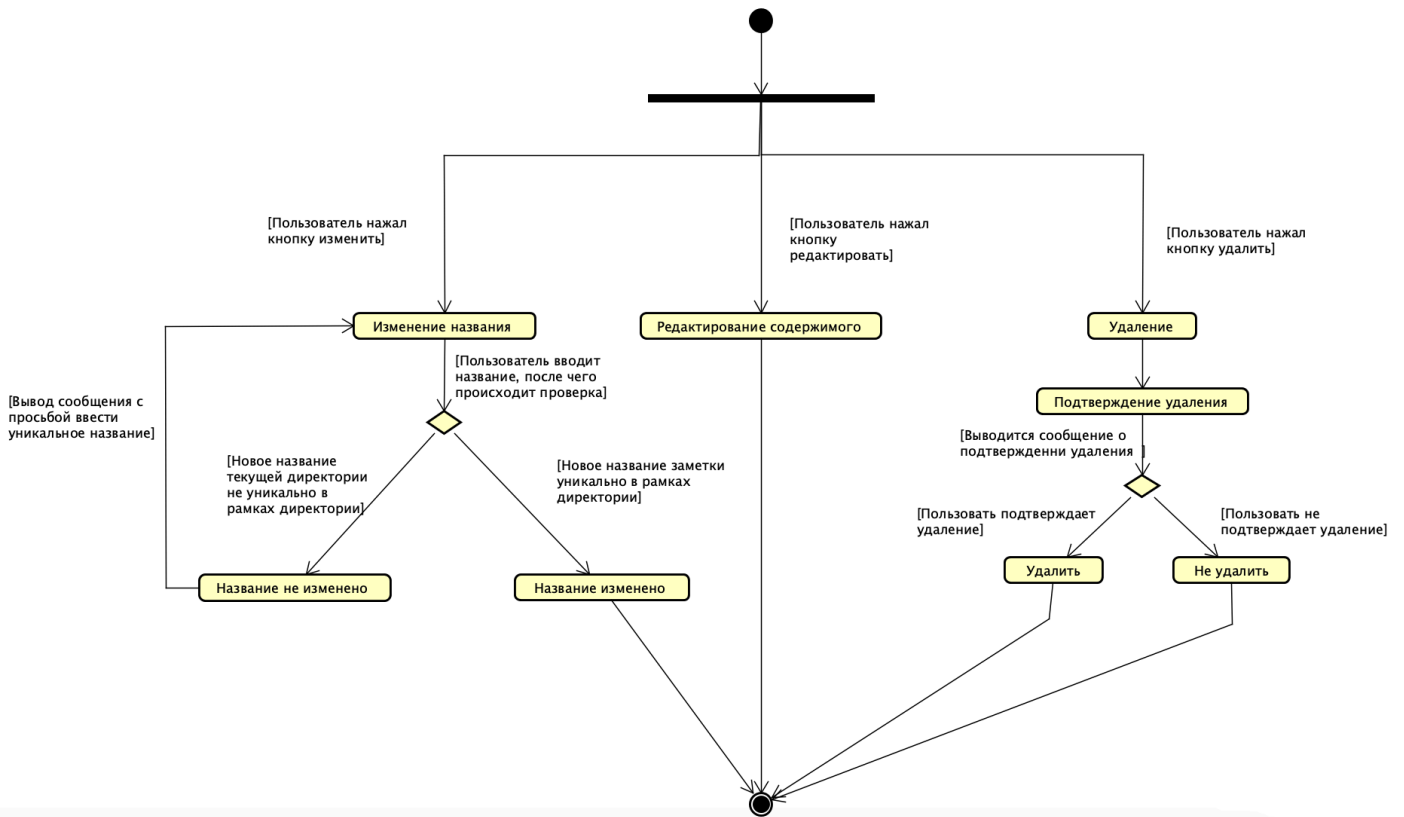


3.3.4. Выбрать заметку



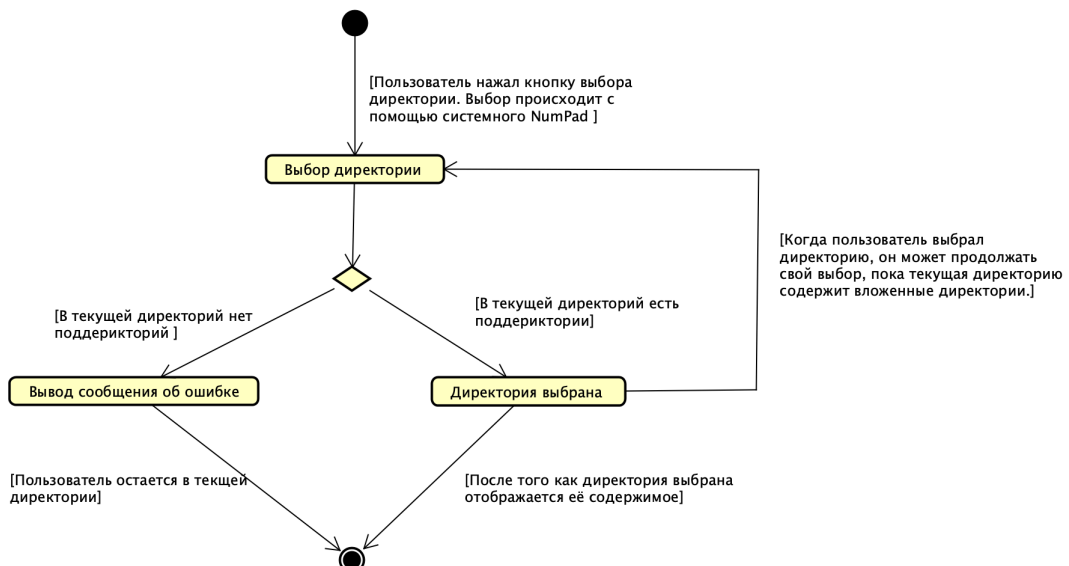
3.3.5. Редактировать заметку

[После выбора заметки, пользователю доступно её редактирование. Для заметок доступно изменение названия, редактирование содержимого и удаление. При изменении названия в директории, где находится заметка, проверяется наличие заметки с таким же названием. Если названия совпадают, то пользователю будет предложено ввести другое название. При редактировании содержимого Telegram гарантирует, что содержанием не будет пустым. При удалении у пользователя запрашивается подтверждение данного действия.]



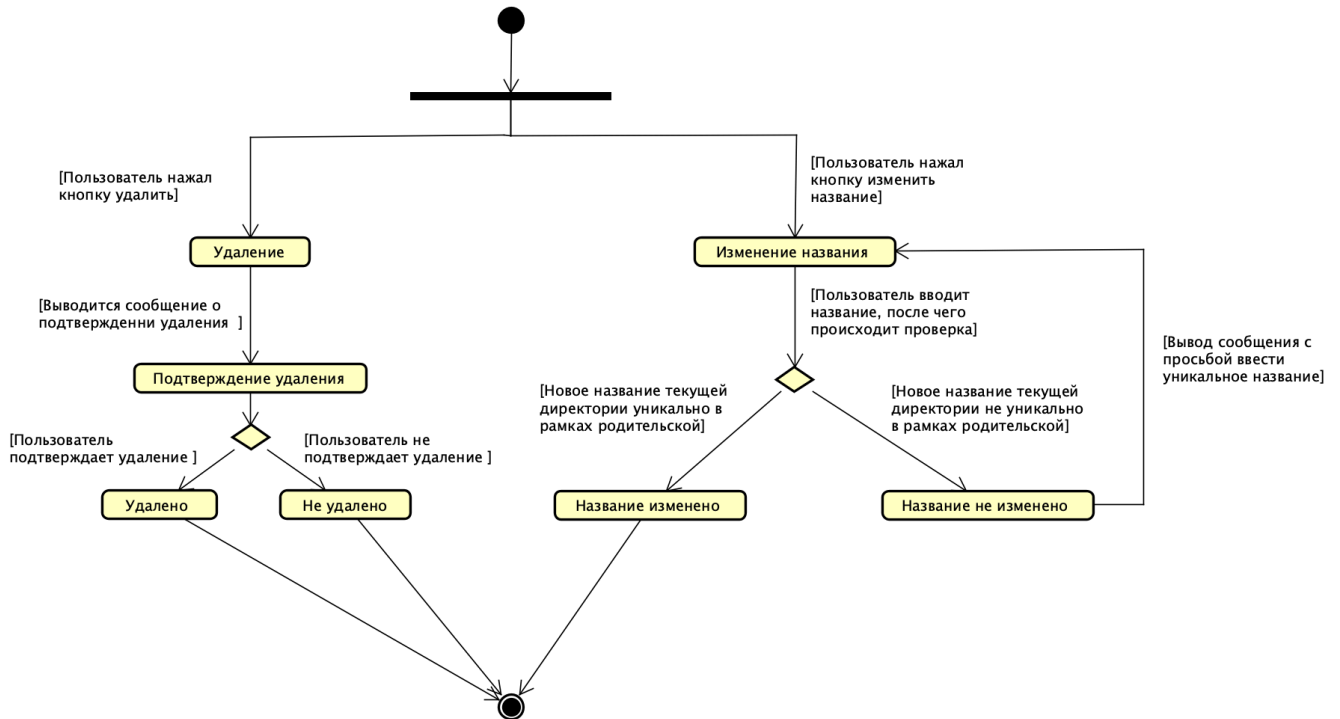
3.3.6. Выбрать директорию

[Выбор директории доступен пользователю из любой директории через соответствующую кнопку. После нажатия на кнопку пользователю предоставляется список директорий в текущей директории. Выбор нужной директории происходит с помощью системного NumPad. Если директория не содержит вложенных директорий, то при попытке выбора выводится соответствующее сообщение об ошибке.]



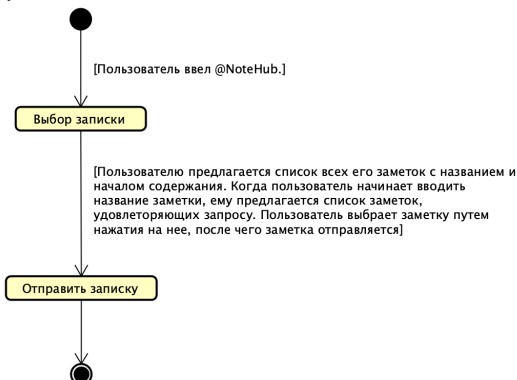
3.3.7. Редактировать директорию

[Редактирование директории доступно пользователю из любой, кроме корневой, директории через соответствующие кнопки. Для директорий доступно изменение названия и удаление. При изменении названия в родительской директории проверяется наличие директории с таким же названием. Если названия совпадают, то пользователю будет предложено ввести другое название. При удалении у пользователя запрашивается подтверждение данного действия.]



3.3.8. Поделиться заметкой

[Пользователю необходимо в чате с другим пользователем Telegram'a в текстовом поле ввести @NoteHub. После этого пользователь вводит название заметки и бот предложит список заметок соответствующих этому названию. При выборе одной из заметок происходит отправка сообщения с содержанием выбранной заметки.]



3.4. Нефункциональные требования

3.4.1. Производительность

- Бот должен иметь быстрое время ответа, не превышающее 3 секунд, чтобы обеспечить хороший пользовательский опыт.

- Система должна поддерживать обработку как минимум 20 запросов в секунду.

3.4.2. Надежность

- Бот должен быть доступен более 97% времени в сутки

3.4.3. Безопасность

- Система должна быть защищенной от атак, таких как SQL-инъекции.

3.4.4. Масштабируемость

- Система должна быть модульной

3.4.5. Документация

- Код должен быть снабжен комментариями

4. Обзор архитектуры

Этот раздел описывает архитектуру системы.

4.1.1. Компонентная модель системы

[Здесь приводится Component diagram - диаграмма компонентов системы, со связями между компонентами и интерфейсами между ними, а также описание их взаимодействия. Для каждого компонента дается краткое описание его места и предназначения в системе]

4.1.1.1. Компонент 1

[Здесь приводится более подробное описание предназначения компонента и Package diagram – диаграмма пакетов, из которых состоит данный компонент. Обязательно выделение на диаграмме интерфейсов пакета, служащих для связи с другими пакетами (фасад пакета), а также ключевых классов, используемых другими пакетами в use-case реализациях]

4.1.1.2. Компонент 2

[Здесь приводится более подробное описание предназначения компонента и Package diagram – диаграмма пакетов, из которых состоит данный компонент. Обязательно выделение на диаграмме интерфейсов пакета, служащих для связи с другими пакетами (фасад пакета), а также ключевых классов, используемых другими пакетами в use-case реализациях]

4.1.2. Компоненты сторонних производителей

[Здесь приводится список использованных компонент сторонних производителей, использованных при разработке системы, с указанием их предназначения в системе]

4.1.3. Схема развертывания приложения

[Здесь приводится Deployment diagram - диаграмма развертывания системы, со связями между узлами и указанием способа связи (протокола). На диаграмме обязательно указать, какие компоненты находятся на том или ином узле]

5. Допущения и ограничения

[Краткое описание допущений, которые подразумевает данный проект, и любых ограничений (например, по бюджету, участникам, требуемому оборудованию, срокам и т.п.), накладываемых на его выполнение.]

Пример: При разработке проекта принято допущение, что число транзакций в единицу времени значительно (более чем в 10 раз) снижается в ночное время, что позволяет в период с 01:00 до 6:00 производить автоматическое обновление программного обеспечения системы, требующее полной перезагрузки и остановки сервиса на период до 5 минут.

6. Известные проблемы

Ниже приводятся известные на данный момент проблемы и недоработки выработанного программного решения, а также возможные пути их устранения в последующих итерациях проекта.

6.1. Невысокая производительность приложения

Проблема	Производительность приложения экспоненциально деградирует при общем числе пользователей выше 10000 и числе одновременных сессий выше 100.
Ранг	10 (высокий)
Влияние на проект	Невозможность использования системы при числе пользователей более 10000.
Пути решения	Кластеризация веб-сервера и сервера базы данных, а также применение load balancer в точке маршрутизации запроса к веб-серверу.

Лист регистрации изменений

Дата	Версия	Описание	Автор

[В качестве описания версии можно указывать какие изменения/дополнения были сделаны в этой версии по отношению к предыдущей.]

Лист регистрации проверок

Дата	Версия	Описание	Автор

[Здесь описываются результаты проверки документа. Для каждой проверки указывается число, версия документа, описание результатов проверки и имя человека, который делал проверку.]