

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

Для аутентификации пользователю необходимо нажать на кнопку «/start» в чате с ботом. После чего будет выведено приветственное сообщение и осуществлен переход в корневую директорию. При этом роль пользователя сменится на роль Авторизованного пользователя в чате с ботом.

3.3. Функциональные требования для роли Авторизованный пользователь

3.3.1. Совершить поиск по названию записки

Для поиска заметки по названию пользователю необходимо нажать кнопку «Найти» и ввести название заметки. После чего ему будут предоставлены результаты поиска.

3.3.2. Создать заметку

Для создания заметки пользователю необходимо нажать кнопку «Создать заметку». После чего пользователю будет предложен выбор: либо ввести название заметки, либо нажать кнопку пропустить. Во втором случае название заметки будет автоматически выбрано ботом. Если же такое название в текущей директории уже есть, то бот выведет сообщение об ошибке и предложит ввести название снова. В случае успеха пользователю необходимо ввести текст записки. После этого заметка будет создана, и пользователю будет выведена текущая директория.

3.3.3. Создать директорию

Для создания директории пользователю необходимо нажать кнопку «Создать директорию». После чего пользователю будет предложен выбор: либо ввести название директории, либо нажать кнопку пропустить. Во втором случае название директории будет автоматически выбрано ботом. Если же такое название в текущей директории уже есть, то бот выведет сообщение об ошибке и предложит ввести название снова. В случае успеха директория будет создана, и пользователю будет выведена текущая директория.

3.3.4. Выбрать заметку

Для выбора заметки пользователю необходимо в директории нажать на кнопку «Выбрать заметку». Бот выведет информацию о заметках в текущей директории. Пользователь с помощью NumPad может выбрать необходимую заметку с соответствующим номером. Если же директория не содержит заметок, то выведется соответствующее сообщение об ошибке

3.3.5. Редактировать заметку

Для редактирования заметки пользователь должен предварительно выбрать заметку. Через кнопки пользователь может:

- **Изменить название заметки**

Пользователь нажал на кнопку «Изменить». В этом случае пользователю будет предложено написать новое название для заметки. В случае, если заметка с таким же названием уже существует в текущей директории, бот отправит сообщение с ошибкой и просьбой повторить ввод названия. При успешном вводе нового названия будет осуществлен переход в предыдущее состояние с сменным названием заметки.

- **Редактировать содержимое**

Пользователь нажал на кнопку «Редактировать». В такой ситуации пользователю будет предложено изменение содержимого. После чего будет осуществлен переход в состояние просмотра заметки.

- **Удалить заметку**

Пользователь нажал на кнопку «Удалить». В таком случае пользователю будет выведено сообщение о подтверждении удаления заметки. В зависимости от нажатия соответствующей кнопки будет: либо произведено удаление заметки, либо же пользователь будет возвращен в в состоянии просмотра заметки. В случае удаления заметки пользователь перейдет в состояния просмотра текущей директории.

3.3.6. Выбрать директорию

Для выбора директории пользователю необходимо нажать на цифру в NumPad панели с соответствующим номером директории в выведенном сообщении с информацией о текущей директории. В этом состоянии пользователь через кнопки может:

- **Выбрать директорию**

В такой ситуации пользователю будут выведены все вложенные в текущую директории

- **Выбрать заметку**

В этой ситуации пользователю будут выведены все записки в текущей директории

3.3.7. Редактировать директорию

Для редактирования директории пользователь должен предварительно выбрать директорию. В этом состоянии пользователь через кнопки может:

- **Удалить директорию**

В случае, если текущая директория является корневой, пользователю выведется сообщение об ошибке. В остальных случаях пользователю будет выведено сообщение о подтверждении удаления директории. В зависимости от нажатия соответствующей кнопки будет: либо произведено удаление директории, либо же пользователь будет возвращен в состояние просмотра директории. В случае удаления пользователь перейдет в состояние просмотра родительской директории.

- **Изменить название директории**

В этом случае пользователю будет предложено написать новое название для директории. В случае, если директория с таким же названием уже существует в текущей директории бот отправит сообщение с ошибкой и просьбой повторить ввод названия. При успешном вводе нового названия будет осуществлен переход в состояние просмотра директории с сменным названием директории.

3.3.8. Поделиться заметкой

Пользователь должен находиться в другом чате. Для того, чтобы поделиться заметкой, ему необходимо в текстовом поле @NoteeHub и через пробел название той записки, которой он хочет поделиться. Пользователю будут предложены варианты записок с таким названием. При выборе одной из них, пользователь автоматически отправит сообщение с содержимым этой записки.

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.]