**Техническое описание проекта по курсу ООАД**

**Planning&Doing**

**Студентов ФИТ НГУ**

**Плугатарева Кирилла Андреевича**

**Курбатова Максима Андреевича**

**группа 20212**

**Версия 1.2.3**

Содержание

[1.](#_heading=h.30j0zll) Введение 3

[1.1](#_heading=h.1fob9te) Цель 3

[1.2](#_heading=h.3znysh7) Область действия 3

[1.3](#_heading=h.2et92p0) Определения и сокращения 3

[1.4](#_heading=h.tyjcwt) Ссылки 3

[1.5](#_heading=h.3dy6vkm) Краткое описание 3

[2.](#_heading=h.1t3h5sf) Предметная область проекта 4

[2.1](#_heading=h.4d34og8) Существующие проблемы 4

[2.2](#_heading=h.2s8eyo1) Предполагаемое решение 4

[3.](#_heading=h.17dp8vu) Требования к программному решению 4

[3.1](#_heading=h.3rdcrjn) Роли 4

[3.2](#_heading=h.26in1rg) Функциональные требования для роли Роль1 4

[3.2.1](#_heading=h.lnxbz9) <Use Case Name 1> 4

[3.2.2](#_heading=h.35nkun2) <Use Case Name 2> 4

[3.3](#_heading=h.1ksv4uv) Функциональные требования для роли Роль2 4

[3.3.1](#_heading=h.44sinio) <Use Case Name 1> 4

[3.3.2](#_heading=h.2jxsxqh) <Use Case Name 2> 4

[3.4](#_heading=h.z337ya) Нефункциональные требования 4

[4.](#_heading=h.3j2qqm3) Обзор архитектуры 5

[4.1.1](#_heading=h.4i7ojhp) Компонентная модель системы 5

[4.1.1.1](#_heading=h.2xcytpi) Компонент 1 5

[4.1.1.2](#_heading=h.1ci93xb) Компонент 2 5

[4.1.2](#_heading=h.3whwml4) Компоненты сторонних производителей 5

[4.1.3](#_heading=h.2bn6wsx) Схема развертывания приложения 5

[5.](#_heading=h.qsh70q) Допущения и ограничения 5

[6.](#_heading=h.1pxezwc) Известные проблемы 6

[6.1](#_heading=h.49x2ik5) Невысокая производительность приложения 6

Техническое описание проекта по курсу ООАД

# Введение

## Цель

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

## Область действия

Документ разработан в рамках проекта *Planning&Doing* на основе стандартного шаблона и предназначен для использования студентами ФИТ и преподавателями дисциплины ООАД.

## Определения и сокращения

**Таблица 1: Определения и сокращения**

| **Термин** | **Описание** |
| --- | --- |
| CST | Control System Tasks - система контроля заданиями |
| Note | Note - заметка |
| TRASH | Корзина со всеми задачами пользователя, с которыми еще непонятно что делать |
| EXCT | Execution tasks. Задачи, которые на исполнении у пользователя в текущий момент |
| DONE | Задачи уже выполненные пользователем |

## 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.

## 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.

# Предметная область проекта

*Рабочее пространство для создания заметок.*

*Все мы хотим структурировать информацию и получать к ней доступ по первому требованию. Например, мы хотим узнать, что нам нужно сделать завтра или в ближайшее время. Чтобы не хранить все в голове или не вести записную книжку, страницы в которой могут неожиданно закончиться, мы придумали наш проект - Planning&Doing, который решает эти проблемы. С помощью этого приложения ты сможешь легко создавать, фильтровать и управлять своими заданиями, а главное - никогда про них не забывать.*

## Существующие проблемы

## *Множество неструктурированных данных;*

## *Сложность держать все задачи в голове;*

## *Отсутствии инструментов на сегодняшний день, которые могли бы быть универсальными или доступны любому пользователю компьютера;*

## Предполагаемое решение

*Разработать ПО с понятным графическим интерфейсом, бесплатным общим доступом, которое будет решать проблемы хранения, структурирования и фильтрации данных пользователя.*

# Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта *Planning&Doing*

* 1. **Кликабельный прототип**

Ссылка на кликабельный прототип в Figma с доступом на комментирование:

* [Главной страницы до регистрации](https://www.figma.com/file/k686GfxJWBRokJWn8utMfn/RegiterPage?node-id=0%3A1)
* [Главная страница после регистрации](https://www.figma.com/file/aC8TrhK2nRCbre1Ht0D6ld/MainPage?node-id=0%3A1)

## Роли

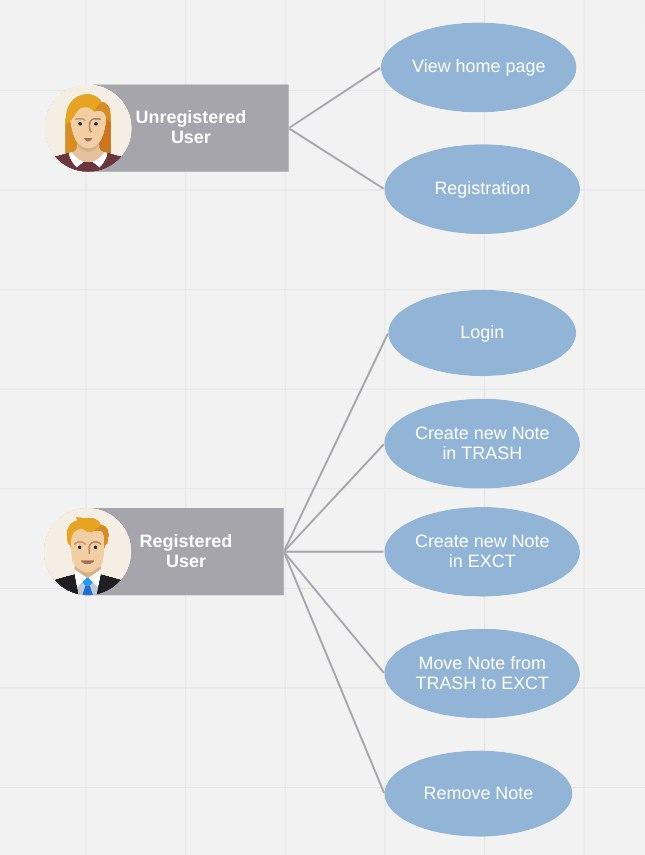
Роль - это что-то (например: другая система) или кто-то (например: человек) вне системы, которые взаимодействуют с ней. В предлагаемой к разработке системе идентифицированы следующие роли:

### <Unregistered user>

Пользователь, который еще не зарегистрировался. Он может просмотреть начальную страницу приложения, а может зарегистрироваться, чтобы получить доступ ко всем функциям сервиса.

### <Registered user>

Пользователь, который уже зарегистрировался, имеет собственный уникальный логин и пароль. Имеет доступ ко всем функциям сервиса.



| Номер: | **UC.1** | | |
| --- | --- | --- | --- |
| Название: | **Регистрация** | | |
| Прототип: | Ссылка в Figma (п. 1-2 основного потока). Ссылка в Figma (исключение). | | |
| Автор: | Плугатарев | Последнее обновление: | 15.10.22 |
| Дата создания: | 15.10.22 | Дата обновления: |  |
| Действующие лица: | Пользователь, система. | | |
| User-story: | Пользователь может зарегистрироваться в сервисе, чтобы получить доступ к функционалу сервиса. | | |
| Предусловия: | Пользователь загрузил страницу сервиса. | | |
| Постусловия: | Пользователь имеет учетную запись. | | |
| Триггер: | Пользователь инициировал вызов операции зарегистрироваться, кликнув левой кнопкой мыши на кнопку “Register”. | | |
| Основной поток: | 1. Система предлагает заполнить следующие поля с отображением:    1. Username (Имя пользователя);    2. Password; (Required: one uppercase letter and one number)    3. Conﬁrm password. 2. Пользователь заполняет поля и кликает на кнопку “Register”. 3. Система проверяет поля на корректность:    1. Корректность Username;    2. Соблюдение Password валидации;    3. Совпадение полей Password и Conﬁrm password. 4. Система отправляет данные о пользователе на сервер. 5. Сервер сохраняет данные о пользователе в базе данных. | | |
| Альтернативные потоки: | - | | |
| Исключения: | 3.а. Поля заполнены некорректно.  3.а.1. Пользователь заполнил не все поля.  3.а.2. Система уведомляет пользователя о том, что заполнены не все поля. | | |

|  | 3.а.3. Пользователь заполняет поля.  3.а.4. Выполнение п. 2-5 основного потока.  3.а.1.1. Пользователь ввел некорректный “Username”.  3.a.1.2. Система уведомляет пользователя о том, что “Username” заполнен некорректно (Invalid format).  3.а.1.3. Пользователь заполняет “Username”. 3.а.1.4. Выполнение п. 2-5 основного потока.  3.а.2.2.1. Пользователь ввел “Username” который уже зарегистрирован.  3.а.2.2.2. Система уведомляет пользователя о том, что данный Username уже зарегистрирован в системе.  3.а.2.2.3. Пользователь заполняет “Username”. 3.а.2.2.4. Выполнение п. 2-5 основного потока.  3.а.3.1. Пользователь ввел невалидный “Password”.  3.a.3.2. Система уведомляет пользователя о том, что “Password” заполнен некорректно и напоминаем валидацию (Required: one uppercase letter and one number).  3.а.3.3. Пользователь заполняет “Password”. 3.а.3.4. Выполнение п. 2-5 основного потока.  3.а.4.1. Поля “Password“ и “Conﬁrm password” не совпадают.  3.a.4.2. Система уведомляет пользователя о том, что поля “Password” и “Conﬁrm password” должны совпадать (Passwords don’t match).  3.а.4.3. Пользователь заполняет “Password” и “Conﬁrm password”.  3.а.4.4. Выполнение п. 2-5 основного потока. |
| --- | --- |
| Дополнительные требования: |  |

| Номер: | **UC.2** | | |
| --- | --- | --- | --- |
| Название: | **Добавление новой заметки в TRASH** | | |
| Прототип: | Ссылка в Figma (отдельная страница). Ссылка в Figma (всплывающее окно). | | |
| Автор: | Плугатарев | Последнее обновление: | 15.10.22 |
| Дата создания: | 15.10.22 | Дата обновления: |  |
| Действующие лица: | Пользователь, система. | | |
| User-story: | Пользователь может добавлять новые в TRASH, чтобы все свои мысли и возможные задачи выгрузить из своей головы в систему. | | |
| Предусловия: | Пользователь авторизовался в системе. | | |
| Постусловия: | Пользователь создал новую заметку в таблице TRASH. | | |
| Триггер: | Пользователь инициировал вызов операции добавить заметку в TRASH, кликнув левой кнопкой мыши на кнопку “Add new note” снизу таблицы TRASH. | | |
| Основной поток: | 1. Система предлагает заполнить следующие поля с отображением:    1. Note name (Название заметки);    2. Description (Описание работы); 2. Пользователь заполняет поля и кликает на кнопку “Add”. 3. Система проверяет поля на корректность:    1. Корректность Name note 4. Система отправляет данные о пользователе на сервер. 5. Сервер сохраняет данные о пользователе в базе данных. | | |
| Альтернативные потоки: |  | | |
| Исключения: | 3.а. Поле с именем заметки не заполнено | | |

|  | 3.a.1.1. Система уведомляет пользователя о том, что “Note name” не заполнен.  3.а.1.2. Пользователь заполняет “Note name”. 3.а.1.3. Выполнение п. 2-5 основного потока. |
| --- | --- |
| Дополнительные требования: | 1. Описание нотификаций. |

| Номер: | **UC.3** | | |
| --- | --- | --- | --- |
| Название: | **Удаление заметки** | | |
| Прототип: | Ссылка в Figma (отдельная страница). Ссылка в Figma (всплывающее окно). | | |
| Автор: | Курбатов | Последнее обновление: | 22.10.22 |
| Дата создания: | 22.10.22 | Дата обновления: |  |
| Действующие лица: | Пользователь, система. | | |
| User-story: | Пользователь может удалять свои заметки навсегда. | | |
| Предусловия: | Пользователь авторизовался в системе и существует хотя бы одна заметка в таблицах. | | |
| Постусловия: | Пользователь удалил заметку из таблицы. | | |
| Триггер: | Пользователь инициировал вызов операции удалить заметку из таблицы, кликнув левой кнопкой мыши на кнопку “Move to trash” снизу заметки. | | |
| Основной поток: | 1. Система предлагает пользователю 1.1 to trash   1.2 delete permanently  1.3 cancel   1. Пользователь кликает “to trash” или “delete permanently” 2. Система отправляет данные о пользователе на сервер. 3. Система сохраняет данные о пользователе в базе данных. | | |
|  | 1. Пользователь кликает “cancel” 2. Система откатывается на страничку с заметками, но ничего не посылает серверу | | |
| Исключения: | - | | |

|  |  |
| --- | --- |
| Дополнительные требования: | - |

# Обзор архитектуры

Этот раздел описывает архитектуру системы.

### Компонентная модель системы

В приложении использовался архитектурный шаблон MVC (model-view-controller).

В приложении используются следующие пакеты: controllers, entity, mappers, models, repositories, security, services.

Вся REST API логика работы приложения и обработка данных от клиента реализуется в пакете controllers.

Логика авторизации и регистрации пользователей написана в пакете security.

Работа с базой данных: repositories, services.

Преобразование данных между бд и объектами для отображения пользователю описаны в пакетах: mappers, entity, models.

Страницы отображения пользователю: resources/templates/\*

В application.properties описана конфигурация приложения и соединения с базой данных.

#### **models, entity, mappers**

В приложении используется паттерн проектирования Data Transfer Object(DTO).

(<https://habrastorage.org/webt/ep/7e/_k/ep7e_koby-ugq909n3pvlppiso8.png>)

Взаимодействие между приложением и базой данных осуществляется с помощью объектов из entity(сущности бд).

Взаимодействие между приложением и клиентом, а также внутри приложения осуществляется с помощью объектов из models.

Логика преобразования entity объектов в models и наоборот реализована в mappers.

#### **controllers**

Controller обрабатывает запрос пользователя, создаёт соответствующую модель из пакета models и передаёт её для отображения.

Ссылка на API приложения: <https://github.com/plugatarev/OOAD/tree/master/application> (смотри readme)

* + - 1. **security**

Реализована логика авторизации и регистрации пользователей с шифрованием паролей.

* + - 1. **repositories, services**

Взаимодействие с базой данных для хранения информации пользователей посредством ORM.

Все Repository реализуют CrudRepository - интерфейс, который обеспечивает основные операции с базой

данных CRUD (create-read-update-delete).

service содержат обертки над repository, чтобы инкапсулировать всю бизнес логику из repository в сервисах.

### Компоненты сторонних производителей

Фреймворки и библиотеки:

* Spring-Boot – фреймворк для написания web-приложений
* Hibernate – библиотека для работы с базой данных на базе ORM. Cамая популярная реализация спецификации JPA.
* Thymeleaf – серверный механизм Java-шаблонов для веб и автономных сред.

Программное обеспечение:

* MySQL - база данных
* Docker - средство контейнеризации
* Docker-compose - это средство для определения и запуска приложений Docker с несколькими контейнерами.
* Swagger - средство для описания API приложения.

### Схема развертывания приложения

Для развертывания приложения была описана конфигурация docker-контейнеров, содержащих базу данных и само приложение. Запуск можно осуществить на любом сервере поддерживающем средство контейнеризации Docker с доступом в интернет (достаточно установить Docker, Docker-compose и запустить конфигурацию).

[**Аналитическая модель приложения**](https://github.com/plugatarev/OOAD/blob/master/architecture/AnalyticModel.png)

[**Дизайн модель приложения**](https://github.com/plugatarev/OOAD/blob/master/architecture/DesignModel.png)

# Допущения и ограничения

1. Проект делается за три месяца на исключительном энтузиазме его участников, поэтому бюджет и время очень сильно ограничены. Также проектом занимаются всего два человека.
2. Постоянное поддержание проекта, его развитие, а именно, выкатывание новых фич и обновлений требует много времени, которое пока не оплачивается. В ближайшее время проект не будет развиваться, если нас не купит Яндекс.
3. Создатели проекта не являются дизайнерами, а нанять такого человека возможности нет, поэтому создание уникального, приятного глазу пользовательского интерфейса, который бы был удобен и интуитивно понятен пользователю, на данный момент очень важное ограничение проекта.

# Известные проблемы

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

## Разделение данных пользователей

| **Проблема** | Начали, но не успели реализовать регистрацию и авторизацию в приложении. |
| --- | --- |
| **Ранг** | 10 (высокий) |
| **Влияние на проект** | Невозможно разделять данные для разных пользователей |
| **Пути решения** | Дописать процесс регистрации и авторизации с использованием Spring Security и шифрованием паролей в базе данных |

Лист регистрации изменений

| **Дата** | **Версия** | **Описание** | **Автор** |
| --- | --- | --- | --- |
| 25.09.22 | 1.0.1 | Добавлен Vision | Курбатов, Плугатарев |
| 01.10.22 | 1.1.0 | Добавлены use-cases для регистрации и добавления задачи | Курбатов |
| 07.10.22 | 1.1.1 | Добавлена диаграмма use-cases | Плугатарев |
| 15.10.22 | 1.1.2 | Переписал UC в виде таблиц описания | Плугатарев |
| 22.10.22 | 1.1.2 | Добавил UC 3 | Курбатов |
| 22.10.22 | 1.1.2 | Добавил макеты приложения до регистрации и после | Плугатарев |
| 13.11.22 | 1.1.3 | Правки в UC | Курбатов |
| 26.11.22 | 1.2.0 | Добавил обзор архитектуры. Описал API приложения. | Плугатарев |
| 08.12.22 | 1.2.1 | Добавил аналитическую и дизайн модели | Плугатарев |
| 08.12.22 | 1.2.2 | Добавил допущения и ограничения | Курбатов |
| 23.12.22 | 1.2.3 | Добавил проблему приложения | Плугатарев |

*[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** |
| --- | --- | --- | --- |
| 02.10.22 | 1.1.0 | Разбил use-case добавления заметки на несколько случаев с разными таблицами. Добавил термины | Плугатарев |
| 09.12.22 | 1.2.2 | Небольшое изменение допущений разработки | Плугатарев |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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