**Техническое задание.**

**Тема РГР**: Управление проектами

**Исполнитель**: Кислицын Александр Иванович

**Группа**: ФИ101

**Номер зачетной книжки**: 212090

**Направление**: 02.03.02, «Фундаментальная информатика и информационные технологии»

**Профиль**: «Инженерия программного обеспечения»

Оглавление

[1 Термины и определения 3](#_Toc185176194)

[1. 1 Общие термины 3](#_Toc185176195)

[1. 2 Технические термины 3](#_Toc185176196)

[2 Назначение документа 3](#_Toc185176197)

[3 Цели создания Системы 3](#_Toc185176198)

[4 Основные функциональные возможности Системы 3](#_Toc185176199)

[5 Функциональные требования 4](#_Toc185176200)

[5. 1 Регистрация пользователя 5](#_Toc185176201)

[5. 2 Авторизация пользователя 5](#_Toc185176202)

[5. 3 Создание проекта 5](#_Toc185176203)

[5. 4 Просмотр проектов 5](#_Toc185176204)

[5. 5 Удаление проекта 5](#_Toc185176205)

[5. 6 Хранение данных 5](#_Toc185176206)

[5. 7 Уведомления 5](#_Toc185176207)

[6 Описание Вариантов Использования 6](#_Toc185176208)

[6. 1 Войти в Систему 6](#_Toc185176209)

[6. 2 Зарегистрировать нового пользователя 7](#_Toc185176210)

[6. 3 Создать проект 7](#_Toc185176211)

[6. 4 Просмотр проекта 8](#_Toc185176212)

[6. 5 Удалить проект 8](#_Toc185176213)

[7 Требования к экранным формам 9](#_Toc185176214)

[7. 1 Экран авторизации (LoginPage) 9](#_Toc185176215)

[7. 2 Экран регистрации (RegisterPage) 10](#_Toc185176216)

[7. 3 Экран списка проектов (ProjectListPage) 11](#_Toc185176217)

[7. 4 Экран создания проекта (CreateProjectPage) 11](#_Toc185176218)

[7. 5 Экран деталей проекта (ProjectDetailPage) 12](#_Toc185176219)

[7. 6 Требования к общему интерфейсу 13](#_Toc185176220)

[8 Нефункциональные требования 14](#_Toc185176221)

[8. 1 Производительность 14](#_Toc185176222)

[8. 2 Масштабируемость 14](#_Toc185176223)

[8. 3 Безопасность 14](#_Toc185176224)

[8. 4 Удобство использования 15](#_Toc185176225)

[8. 5 Совместимость 15](#_Toc185176226)

1. Термины и определения
   1. Общие термины

**FAQ** – Frequently Asked Questions. Часто задаваемые вопросы.

**ВИ** – Вариант Использования или Use Case.

* 1. Технические термины

**ОС** – операционная система.

**ИС** – информационная система.

1. Назначение документа

Данный документ описывает требования и функциональные возможности системы управления проектами для пользователей, которые могут регистрироваться, входить в систему, создавать проекты, просматривать проекты и удалять их. Система ориентирована на обеспечение удобства управления проектами с минимальными усилиями, хранения данных пользователей и проектов в локальном файле и взаимодействия через графический интерфейс.

1. Цели создания Системы

Цель создания системы заключается в разработке приложения для управления проектами, которое позволяет пользователям:

Регистироваться в системе и авторизовываться для доступа к личным данным и проектам.

Создавать новые проекты, назначать ответственных за их выполнение и задавать описание.

Просматривать проекты с детальной информацией о них.

Удалять проекты, которые больше не нужны.

Хранить данные пользователей и проектов в формате JSON на устройстве для легкости управления и обработки информации.

1. Основные функциональные возможности Системы

Система должна обеспечивать следующие ключевые функциональные возможности:

1. Регистрация пользователей: Возможность создания нового пользователя с вводом имени пользователя и пароля.
2. Авторизация пользователей: Проверка введённых данных (имя пользователя и пароль) и предоставление доступа к функционалу приложения.
3. Создание проектов: Возможность создания новых проектов с указанием названия, описания, ответственного лица и даты создания.
4. Просмотр проектов: Возможность просмотра списка всех проектов, созданных текущим пользователем, с возможностью перехода на страницу деталей проекта.
5. Удаление проектов: Возможность удаления проектов, если они больше не нужны.
6. Хранение данных в JSON файлах: Система сохраняет и загружает данные пользователей и проектов из локальных файлов (JSON), что позволяет сохранять данные между сессиями.
7. Уведомления о действиях: Информация о результате действия (например, успешная регистрация, неудачная попытка входа, успешное создание или удаление проекта) отображается через всплывающие уведомления.
8. Функциональные требования
   1. Регистрация пользователя

Пользователь должен иметь возможность создать новый аккаунт, указав имя пользователя и пароль.

После успешной регистрации, система сохраняет данные в файл и выводит уведомление о том, что регистрация прошла успешно.

* 1. Авторизация пользователя

Пользователь должен иметь возможность войти в систему, введя своё имя пользователя и пароль.

Если введенные данные соответствуют зарегистрированным пользователям, система должна предоставить доступ и отобразить главную страницу.

* 1. Создание проекта

После авторизации, пользователь может создать новый проект, указав название, описание, ответственное лицо и дату создания.

Проект сохраняется в базе данных (JSON файле), и пользователь перенаправляется на страницу со списком своих проектов.

* 1. Просмотр проектов

Пользователь может просматривать список проектов, которые он создал, с краткой информацией о каждом проекте (название, описание и ответственное лицо).

Пользователь может выбрать проект для просмотра детальной информации о нём.

* 1. Удаление проекта

Пользователь имеет возможность удалить проект, который он создал. После этого проект удаляется из базы данных и выводится уведомление о результате действия.

* 1. Хранение данных

Все данные о пользователях и проектах должны сохраняться в локальных JSON файлах.

Данные должны загружаться из файлов при старте системы и сохраняться при изменении.

* 1. Уведомления

Система должна информировать пользователя о результате его действия, например, о регистрации, ошибке при входе, успешном создании или удалении проекта.

2. Описание Вариантов Использования
   1. Войти в Систему

**Основной поток действий**:

1. Неавторизированный Пользователь в главном меню выбирает пункт "Войти в Систему".
2. Система отображает форму авторизации, где пользователь должен ввести имя пользователя и пароль.
3. Пользователь вводит имя пользователя и пароль в соответствующие поля формы.
4. Система проверяет, что такой пользователь существует в базе данных с введёнными именем пользователя и паролем.
5. Если пользователь найден и данные правильные, переход к шагу 7.
6. Если пользователь не найден или данные неверные, система отображает сообщение об ошибке, предлагая пользователю повторить ввод.
7. Система регистрирует вход пользователя, устанавливает глобальную переменную для текущего пользователя и отображает главную страницу с возможностью управлять проектами.
8. Пользователь считается вошедшим в систему и получает доступ к функционалу.

**Альтернативный поток**:

Если введённый пароль неверен или имя пользователя не найдено:

1. Система уведомляет пользователя о неправильных данных и предлагает повторить попытку.
   1. Зарегистрировать нового пользователя

**Основной поток действий**:

1. Неавторизированный Пользователь в главном меню выбирает пункт "Зарегистрироваться".
2. Система отображает форму для ввода имени пользователя и пароля.
3. Пользователь вводит имя пользователя и пароль в соответствующие поля.
4. Система проверяет, что имя пользователя ещё не используется. Если имя уже существует, система уведомляет пользователя об ошибке и предлагает выбрать другое имя.
5. Система сохраняет нового пользователя в базу данных и выводит сообщение об успешной регистрации.
6. Пользователь может перейти к форме авторизации для входа в систему.

**Альтернативный поток**:

Если имя пользователя уже существует в базе данных:

1. Система уведомляет пользователя, что имя пользователя занято, и предлагает выбрать другое имя.
   1. Создать проект

**Основной поток действий**:

1. Авторизованный Пользователь на главной странице выбирает кнопку для создания нового проекта.
2. Система отображает форму для ввода данных о проекте (название, описание, ответственный человек).
3. Пользователь заполняет все поля формы и нажимает кнопку "Сохранить".
4. Система проверяет, что все поля заполнены корректно.
5. Если поля заполнены, проект сохраняется в базе данных (JSON файле), и пользователь возвращается на страницу со списком проектов.
6. Если поля не заполнены, система выводит сообщение об ошибке и предлагает пользователю заполнить форму.

**Альтернативный поток**:

Если поля формы не заполнены полностью:

1. Система уведомляет пользователя о необходимости заполнить все поля.
   1. Просмотр проекта

**Основной поток действий**:

1. Авторизованный Пользователь на странице со списком проектов выбирает проект для просмотра.
2. Система отображает детальную информацию о проекте (название, описание, ответственный, дата создания).
3. Пользователь может вернуться на страницу списка проектов или удалить проект.

**Альтернативный поток**:

Если проект удалён или не существует:

1. Система уведомляет пользователя, что проект не найден.
   1. Удалить проект

**Основной поток действий**:

1. Авторизованный Пользователь на странице подробностей проекта нажимает кнопку "Удалить проект".
2. Система запрашивает подтверждение действия.
3. Пользователь подтверждает удаление.
4. Система удаляет проект из базы данных (JSON файла) и возвращает пользователя на страницу со списком проектов.
5. Система выводит уведомление о том, что проект успешно удалён.

**Альтернативный поток**:

Если проект не может быть удалён по какой-то причине (например, проект не существует):

1. Система уведомляет пользователя об ошибке.
3. Требования к экранным формам
   1. Экран авторизации (LoginPage)

**Цели экрана**:

Предоставить пользователю возможность войти в систему.

Позволить пользователю зарегистрироваться, если он ещё не зарегистрирован.

**Основные элементы**:

**Текстовые поля**:

Имя пользователя (TextField) — для ввода имени пользователя.

Подсказка: "Username"

Пароль (TextField) — для ввода пароля.

Подсказка: "Password"

Поле должно скрывать вводимые символы (обозначение obscureText: true).

**Кнопки**:

Войти в систему (ElevatedButton) — кнопка для отправки данных на проверку и авторизации.

Зарегистрироваться (ElevatedButton) — кнопка для перехода на форму регистрации.

**Сообщения об ошибках**:

В случае неправильных данных (неверный логин или пароль) появляется всплывающее уведомление (SnackBar) с текстом "Invalid login credentials".

**Требования к интерфейсу**:

Формы должны быть компактными, с ясной и понятной разметкой.

Все элементы управления должны быть доступны на экране без необходимости прокрутки, если это возможно.

* 1. Экран регистрации (RegisterPage)

**Цели экрана**:

Предоставить пользователю возможность зарегистрироваться в системе, создав новый аккаунт.

**Основные элементы**:

**Текстовые поля**:

Имя пользователя (TextField) — для ввода имени пользователя.

Подсказка: "Username"

Пароль (TextField) — для ввода пароля.

Подсказка: "Password"

Повторите пароль (TextField) — для подтверждения пароля.

Подсказка: "Confirm Password"

Поле должно скрывать вводимые символы.

**Кнопки**:

Зарегистрироваться (ElevatedButton) — кнопка для отправки данных и регистрации пользователя.

**Сообщения об ошибках**:

В случае, если поля не заполнены или пароли не совпадают, появляется уведомление с ошибкой (SnackBar).

**Требования к интерфейсу**:

Формы должны быть чёткими и минималистичными, с разделением каждого этапа на шаги, чтобы избежать перегрузки пользователю.

* 1. Экран списка проектов (ProjectListPage)

**Цели экрана**:

Показать пользователю список всех проектов, которые он создал.

Позволить пользователю создать новый проект.

**Основные элементы**:

**Кнопки**:

Создать новый проект (ElevatedButton) — кнопка для перехода к экрану создания нового проекта.

Иконка добавления (IconButton) — кнопка для перехода на экран создания проекта (можно использовать иконку Icons.add).

**Список проектов**:

Используется виджет ListView, который отображает проекты, созданные текущим пользователем.

Каждый элемент списка (ListTile) должен содержать:

Название проекта.

Краткое описание.

Ответственного за проект.

**Переход на страницу проекта**:

При нажатии на элемент списка, пользователь перенаправляется на экран подробностей проекта.

**Требования к интерфейсу**:

Список проектов должен быть прокручиваемым, если проектов много.

Каждый проект должен быть представлен в виде карточки или строки с основными данными (название, описание).

* 1. Экран создания проекта (CreateProjectPage)

**Цели экрана**:

Предоставить пользователю возможность создать новый проект.

**Основные элементы**:

**Текстовые поля**:

Название проекта (TextField) — для ввода названия проекта.

Подсказка: "Title"

Описание проекта (TextField) — для ввода описания проекта.

Подсказка: "Description"

Ответственный за проект (TextField) — для указания ответственного лица.

Подсказка: "Responsible Person"

Кнопки:

Сохранить (ElevatedButton) — кнопка для сохранения нового проекта в базе данных.

Сообщения об ошибках:

В случае, если поля не заполнены полностью, появляется уведомление (SnackBar) с текстом "Please fill in all fields".

**Требования к интерфейсу**:

Должна быть ясная визуальная структура формы с обозначениями для каждого поля.

Поля должны быть четко маркированы и располагаться в логичном порядке.

* 1. Экран деталей проекта (ProjectDetailPage)

**Цели экрана**:

Показать детальную информацию о проекте.

Позволить пользователю удалить проект.

**Основные элементы**:

**Текстовые элементы**:

Название проекта (Text) — отображение названия проекта.

Описание проекта (Text) — отображение описания.

Ответственный за проект (Text) — отображение имени ответственного.

Дата создания (Text) — отображение даты создания проекта.

Кнопка для удаления проекта:

Удалить проект (ElevatedButton) — кнопка для удаления проекта из системы.

Кнопка должна быть выделена красным цветом, чтобы подчеркнуть важность действия.

**Сообщения об ошибках**:

В случае успешного удаления проекта, появляется уведомление (SnackBar) о том, что проект был удалён.

Если проект не может быть удалён, появляется уведомление с ошибкой.

**Требования к интерфейсу**:

Все элементы должны быть представлением данных о проекте в чёткой и легко воспринимаемой форме.

Кнопка удаления должна быть выделена ярким цветом и располагаться ниже всех данных о проекте, но не быть слишком навязчивой.

* 1. Требования к общему интерфейсу

Все экраны должны иметь адаптивный дизайн, подходящий для различных размеров экранов (мобильных устройств и планшетов).

Интерфейс должен быть простым и интуитивно понятным, с минимальным количеством шагов для выполнения основных действий.

Все кнопки и поля должны быть достаточного размера для удобного взаимодействия.

2. Нефункциональные требования
   1. Производительность

**Загрузка данных**:

Система должна загружать данные (пользователей и проекты) из файлов в течение 2 секунд при старте приложения на устройствах средней мощности.

При добавлении новых проектов или пользователей и сохранении данных в файл, операция сохранения не должна превышать 3 секунд.

**Отображение проектов**:

При отображении списка проектов должно обеспечиваться мгновенное обновление интерфейса, даже если в базе данных несколько сотен проектов.

* 1. Масштабируемость

**Поддержка увеличения объема данных**:

Система должна поддерживать работу с увеличивающимся количеством пользователей и проектов (в пределах десятков тысяч) без значительного ухудшения производительности.

В случае значительного увеличения данных (например, больше 1000 пользователей или проектов) возможно использование баз данных вместо хранения данных в файлах для улучшения масштабируемости.

**Расширяемость архитектуры**:

Система должна быть спроектирована таким образом, чтобы в будущем можно было легко добавить новые функциональные возможности, такие как управление задачами в проекте или интеграция с внешними сервисами.

* 1. Безопасность

**Шифрование данных**:

Пароли пользователей должны храниться в зашифрованном виде (например, с использованием алгоритма хэширования bcrypt или аналогичного).

Все чувствительные данные, такие как пароли, не должны передаваться в открытом виде (например, использовать HTTPS для передачи данных через сеть).

**Защита от несанкционированного доступа**:

Пользователи, не авторизованные в системе, не должны иметь доступ к функционалу, связанному с проектами.

На экране создания проекта доступ к функционалу должен быть ограничен только для зарегистрированных пользователей.

**Управление сессиями**:

Система должна обеспечивать автоматический выход пользователя через 30 минут неактивности.

Все сессионные данные должны храниться в безопасном виде, недоступном для сторонних приложений.

* 1. Удобство использования

**Интерфейс**:

Приложение должно обеспечивать простой и интуитивно понятный интерфейс для пользователей любого уровня.

Навигация по экранам должна быть логичной и доступной через простое взаимодействие с кнопками и полями ввода.

**Доступность**:

Приложение должно поддерживать функциональные возможности для пользователей с ограниченными возможностями, например, иметь возможности для использования экранного читалки и поддержку масштабирования интерфейса.

**Документация**:

В приложении должна быть доступна краткая справка или руководство пользователя, объясняющее основные функции системы.

* 1. Совместимость

**Мобильные платформы**:

Приложение должно работать на мобильных платформах Android и iOS.

Требования к версиям операционных систем:

Android: версия 6.0 (Marshmallow) и выше.

iOS: версия 12.0 и выше.

**Браузерная версия**:

В случае создания браузерной версии системы, приложение должно быть совместимо с последними версиями браузеров, таких как Google Chrome, Mozilla Firefox и Safari.

1. UML диаграмма

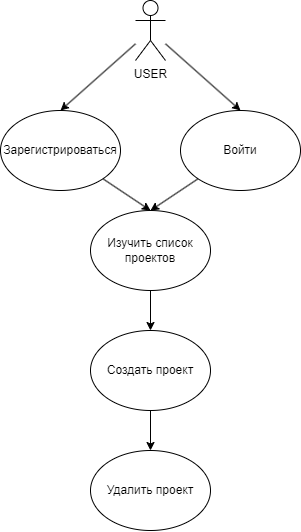


Рисунок 1. UML диаграмма.

1. Диаграмма компонентов

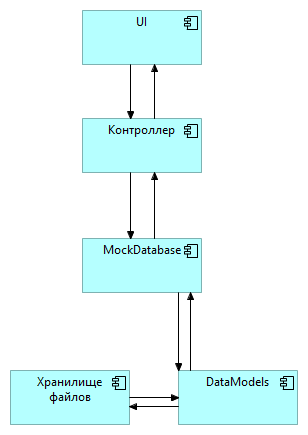


Рисунок 2. Диаграмма компонентов.