**Техническое задание.**

**Тема РГР**: Заказы клининговой компании

**Исполнитель**: Панченко Степан Алексеевич

**Группа**: ФИ101

**Номер зачетной книжки**: 212096

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

**Профиль**: «Инженерия программного обеспечения»

Оглавление

[1 Термины и определения 4](#_Toc185189681)

[1. 1 Общие термины 4](#_Toc185189682)

[1. 2 Технические термины 4](#_Toc185189683)

[2 Назначение документа 4](#_Toc185189684)

[3 Цели создания Системы 4](#_Toc185189685)

[4 Основные функциональные возможности Системы 4](#_Toc185189686)

[5 Функциональные требования 6](#_Toc185189687)

[5. 1 Регистрация пользователя 6](#_Toc185189688)

[5. 2 Авторизация пользователя 6](#_Toc185189689)

[5. 3 Добавление заказа 6](#_Toc185189690)

[5. 4 Просмотр заказов 6](#_Toc185189691)

[5. 5 Редактирование заказов 6](#_Toc185189692)

[5. 6 Хранение данных 6](#_Toc185189693)

[5. 7 Уведомления 6](#_Toc185189694)

[6 Описание Вариантов Использования 7](#_Toc185189695)

[6. 1 Войти в Систему 7](#_Toc185189696)

[6. 2 Зарегистрировать нового пользователя 7](#_Toc185189697)

[6. 3 Добавить заказ 8](#_Toc185189698)

[6. 4 Просмотр заказа 8](#_Toc185189699)

[6. 5 Редактировать заказ 8](#_Toc185189700)

[7 Требования к экранным формам 10](#_Toc185189701)

[7. 1 Экран авторизации (login\_screen) 10](#_Toc185189702)

[7. 2 Экран регистрации (registration\_screen) 10](#_Toc185189703)

[7. 3 Экран списка заказов (order\_list\_screen) 11](#_Toc185189704)

[7. 4 Экран добавления заказа (add\_order\_screen) 12](#_Toc185189705)

[7. 5 Экран деталей заказа (order\_detail\_screen) 13](#_Toc185189706)

[7. 6 Требования к общему интерфейсу 13](#_Toc185189707)

[8 Нефункциональные требования 14](#_Toc185189708)

[8. 1 Производительность 14](#_Toc185189709)

[8. 2 Масштабируемость 14](#_Toc185189710)

[8. 3 Безопасность 14](#_Toc185189711)

[8. 4 Удобство использования 15](#_Toc185189712)

[8. 5 Совместимость 15](#_Toc185189713)

[9 UML диаграмма 16](#_Toc185189714)

[10 Диаграмма компонентов 17](#_Toc185189715)

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 заказов клининговой компании, которые могут регистрироваться, просматриваться и редактироваться. Система ориентирована на обеспечение удобства управления заказами с минимальными усилиями, хранения данных пользователей и сотрудников в локальном файле и взаимодействия через графический интерфейс.

1. Цели создания Системы

Цель создания системы заключается в разработке приложения для управления заказами клининговой компании, которое позволяет пользователям:

Регистрироваться в системе и авторизовываться для доступа к личным данным и заказам.

Добавлять новые заказы, назначать ответственных за их выполнение и задавать описание, стоимость и статус выполнения.

Просматривать заказы с детальной информацией о них.

Редактировать заказы в зависимости от потребностей.

Хранить данные пользователей и заказов 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. Добавление заказов: Возможность добавления новых заказов с указанием названия, описания, ответственного лица, стоимости и статуса.
4. Просмотр заказов: Возможность просмотра списка всех заказов, с возможностью перехода на страницу деталей заказа.
5. Редактирование заказов: Возможность редактирования заказов.
6. Хранение данных: Система сохраняет и загружает данные пользователей и заказов из локальных файлов.
7. Уведомления о действиях: Информация о результате действия (например, неудачная попытка входа) 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. Добавление заказа

После авторизации, пользователь может добавить новый заказ, указав название, описание, ответственное лицо, стоимость и статус.

Заказ сохраняется в локальных данных, и пользователь перенаправляется на страницу со списком заказов.

* 1. Просмотр заказов

Пользователь может просматривать список заказов, которые находятся в системе, с краткой информацией о каждом заказе.

Пользователь может выбрать заказ для просмотра детальной информации о нём.

* 1. Редактирование заказов

Пользователь имеет возможность редактирования заказов. После этого данные заказа изменяются.

* 1. Хранение данных

Все данные о пользователях и заказах должны сохраняться в локальных файлах.

* 1. Уведомления

Система должна информировать пользователя о результате его действия, например, ошибке при входе.

1. Описание Вариантов Использования
   1. Войти в Систему

**Основной поток действий**:

1. Неавторизированный Пользователь, впервые зайдя в систему, видит страницу авторизации.
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 заказами.
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 на странице авторизации может выбрать кнопку "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. Добавить заказ

**Основной поток действий**:

1. Авторизованный Пользователь на главной странице выбирает кнопку для добавления нового заказа.
2. Система отображает форму для ввода данных о заказе (название, описание, ответственный человек, стоимость и статус).
3. Пользователь заполняет все поля формы и нажимает кнопку "Сохранить".
4. Система проверяет, что все поля заполнены корректно.
5. Если поля заполнены, заказ сохраняется в данных), и пользователь возвращается на страницу со списком заказов.
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 заказа

**Основной поток действий**:

1. Авторизованный Пользователь на странице со списком заказов выбирает заказ для просмотра.
2. Система отображает детальную информацию о заказе (название, описание, ответственный, стоимость, статус).
3. Пользователь может вернуться на страницу списка заказов или удалить заказ.

**Альтернативный поток**:

Если заказ удалён или не существует:

1. Система уведомляет пользователя, что заказ не найден.
   1. Редактировать заказ

**Основной поток действий**:

1. Авторизованный Пользователь на странице заказов нажимает на кнопку “Редактировать заказ”.
2. Система выводит диалог редактирования.
3. Пользователь заполняет и подтверждает редактирование.
4. Система изменяет заказ и возвращает пользователя на страницу со списком заказов.
5. Требования к экранным формам
   1. Экран авторизации (login\_screen)

**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 (registration\_screen)

**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 заказов (order\_list\_screen)

**Цели экрана**:

Показать пользователю список всех заказов.

Позволить пользователю добавить новый заказ.

**Основные элементы**:

**Кнопки**:

Добавить новый заказ (FloatinActionButton) — кнопка для перехода к экрану добавления нового заказа.

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

**Список заказов**:

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

Каждый элемент списка (ListTile) должен содержать:

Название заказа.

Краткое описание.

**Переход на страницу заказа**:

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

**Требования к интерфейсу**:

Список заказов должен быть прокручиваемым, если заказов много.

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

* 1. Экран добавления заказа (add\_order\_screen)

**Цели экрана**:

Предоставить пользователю возможность добавить новый заказ.

**Основные элементы**:

**Текстовые поля**:

Название заказа (TextField) — для ввода названия заказа.

Подсказка: "Title"

Описание заказа (TextField) — для ввода описания заказа.

Подсказка: "Description"

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

Подсказка: "Responsible Person"

Адрес заказа (TextField) — для указания адреса.

Подсказка: "address"

Стоимость (TextField) — для указания стоимости.

Подсказка: "price"

Кнопки:

Сохранить (ElevatedButton) — кнопка для сохранения нового заказа 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 заказа (order\_detail\_screen)

**Цели экрана**:

Показать детальную информацию о заказе.

**Основные элементы**:

**Текстовые элементы**:

Название заказа (Text) — отображение названия заказа.

Описание заказа (Text) — отображение описания.

Ответственный за заказ (Text) — отображение имени ответственного.

**Требования к интерфейсу**:

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

Кнопка удаления должна быть выделена ярким цветом и располагаться ниже всех данных о заказе, 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.

1. Нефункциональные требования
   1. Производительность

**Загрузка данных**:

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

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

**Отображение заказов**:

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

* 1. Масштабируемость

**Поддержка увеличения объема данных**:

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

В случае значительного увеличения данных (например, больше 1000 пользователей или заказов) 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, 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 заказами.

На экране добавления заказов 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

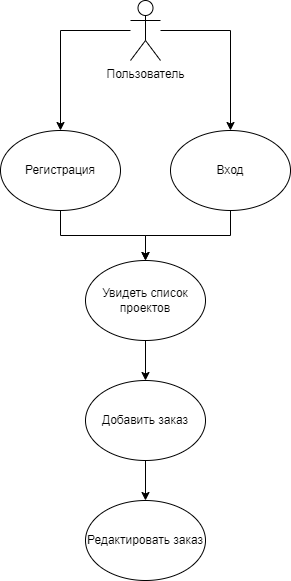


Рисунок 1. UML диаграмма.

1. Диаграмма компонентов

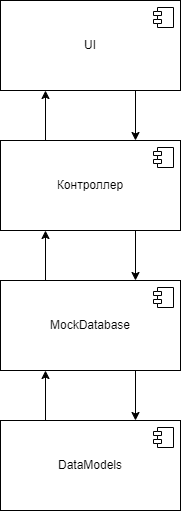


Рисунок 2. Диаграмма компонентов.