Студент:

ФИО, группа

Руководитель:

Ученая степень, ФИО

Тема:

**Реализация и тестирование системы**

1. **Реализация компонентов системы (с примерами исходного кода)**

В реализации компонентов системы имеется сервис Data Management, который предоставляет функциональность для работы с информацией о дронах и камерах. Этот сервис состоит из следующих файлов:

uav/mod.rs - Основной файл, определяющий структуру Uav и методы для создания экземпляров дронов с заданными параметрами.

Пример структуры Uav:

#[derive(Debug, Deserialize, Serialize)]

pub struct Uav {

id: u64, // uav id

pub name: String, // uav name

pub max\_payload\_mass: u64, // maximum payload in grams

pub flight\_duration: u64, // average flight duration in seconds

pub takeoff\_speed: f64, // average takeoff speed in meters per second

pub flight\_speed: f64, // average flight speed in meters per second

pub min\_altitude: f64, // minimum safe flight altitude in meters

pub max\_altitude: f64, // maximum safe flight altitude in meters

}

**Рис. 1.** Листингфункции для определения количества кластеров

uav/uav\_handle.rs - Файл содержит функции для работы с дронами через Tauri, такие как создание, обновление и удаление дронов, а также получение списка всех дронов.

Пример функции для создания нового дрона:

#[tauri::command]

pub fn new\_uav(uav: Uav) -> String {

let conn = Connection::open("mydatabase.db").expect("Cant open base");

println!("Received new UAV: {:?}", uav);

match uav\_sql::insert(&uav, &conn) {

Ok(\_) => "Ok".to\_string(),

Err(e) => e.to\_string(),

}

}

**Рис. 1.** Листингфункции для определения количества кластеров

1. uav/uav\_sql.rs - Файл содержит функции для работы с базой данных, включая создание таблицы для хранения информации о дронах, а также вставка, обновление, удаление и получение списка всех дронов.

Пример функции для создания таблицы в базе данных:

pub fn create\_table(conn: &Connection) -> Result<usize> {

conn.execute(

"CREATE TABLE IF NOT EXISTS uav (

uav\_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,

uav\_name TEXT NOT NULL,

uav\_max\_payload\_mass INTEGER NOT NULL CHECK (uav\_max\_payload\_mass >= 0),

uav\_flight\_duration INTEGER NOT NULL CHECK (uav\_flight\_duration >= 0),

uav\_takeoff\_speed REAL NOT NULL CHECK (uav\_takeoff\_speed >= 0),

uav\_flight\_speed REAL NOT NULL CHECK (uav\_flight\_speed >= 0),

uav\_min\_altitude REAL DEFAULT 0 NOT NULL CHECK (uav\_min\_altitude >= 0),

uav\_max\_altitude REAL DEFAULT 0 NOT NULL CHECK (uav\_max\_altitude >= 0)

)",

(), // empty list of parameters.

)

}

**Рис. 1.** Листингфункции для определения количества кластеров

camera/mod.rs - Основной модуль, который содержит структуру и методы для работы с камерой. Структура Camera содержит различные свойства камеры, включая ее идентификатор, имя, массу, угол обзора по оси X и разрешение. Здесь также определены метод для создания новых камер.

Пример структуры Camera:

pub struct Camera {

id: u64, // id

pub name: String, // name

pub mass: u64, // mass in grams

pub fov\_x: f64, // x-axis viewing angle in degrees

pub resolution\_x: u16, // camera resolution x

pub resolution\_y: u16, // camera resolution y

}

**Рис. 1.** Листингфункции для определения количества кластеров

camera/camera\_handle.rs - Этот модуль предоставляет функции для обработки запросов к базе данных. Здесь определены функции для создания новой камеры, обновления существующей камеры, удаления камеры и получения всех камер из базы данных. Каждая из этих функций открывает соединение с базой данных и вызывает соответствующую функцию из модуля camera\_sql.

Пример функции для обновления данных о камере.

#[tauri::command]

pub fn update\_camera(camera: Camera) -> String {

let conn = Connection::open("mydatabase.db").expect("Cant open base");

println!("Received updated camera: {:?}", camera);

match camera\_sql::update(&camera, &conn) {

Ok(\_) => "Ok".to\_string(),

Err(e) => e.to\_string(),

}

}

**Рис. 1.** Листингфункции для определения количества кластеров

camera/camera\_sql.rs - Этот модуль содержит функции для взаимодействия с базой данных SQLite. Здесь определены функции для создания таблицы в базе данных, вставки новой камеры в базу данных, обновления существующей камеры, удаления камеры из базы данных и получения всех камер из базы данных. Все эти функции работают непосредственно с базой данных, выполняя соответствующие SQL-запросы.

Пример функции обновления данных о камере представлен на рисунке

Сервис Algorithms, описанный файлом algorithms.rs, представляет собой ключевую составляющую системы и выполняет алгоритмические функции. Он включает в себя следующие функции:

1. discretize\_area: Эта функция принимает вектор кортежей, представляющих собой координаты x и y полигона, и размеры фотографии (ширину и высоту). Функция начинает с инициализации минимальных и максимальных значений x и y. После этого функция применяет алгоритм поиска точек внутри полигона, основанный на лучевом методе. После того как все точки были проверены, функция возвращает вектор кортежей, представляющих собой координаты центров прямоугольников, которые пересекаются с полигоном. Реализация представлена в листинге во вложении
2. nearest\_neighbor: Эта функция принимает вектор точек и начальную точку, а затем использует алгоритм ближайшего соседа для определения оптимального пути через все точки. Функция инициализирует набор оставшихся точек и результатирующий вектор с начальной точкой. Функция затем входит в цикл, продолжающийся, пока все точки не будут обработаны. В каждой итерации цикла функция находит ближайшую точку к текущей точке, удаляет ее из оставшихся точек и добавляет ее в результатирующий вектор.
3. euclidean\_distance: Это вспомогательная функция, которая принимает две точки и возвращает евклидово расстояние между ними. Расстояние вычисляется как квадратный корень из суммы квадратов разностей координат x и y.
4. brute\_force: Эта функция использует алгоритм полного перебора для поиска оптимального пути через набор точек. Функция начинает с создания отдельного потока для каждой точки, каждый из которых запускает функцию brute\_force\_helper. Эта вспомогательная функция рекурсивно исследует все возможные пути через оставшиеся точки, обновляя лучший путь при обнаружении более короткого пути.
5. calculate\_distance: Эта функция вычисляет общую длину пути, проходящего через заданный набор точек
6. **Реализация интерфейса системы (пользовательский и/или программный интерфейс)**

Пользовательский интерфейс системы состоит из трех основных меню: центрального меню, левого меню и правого меню. Центральное меню отображает карту и маршрут дрона, а также содержит набор кнопок для управления. Давайте рассмотрим его подробнее:

1. Центральное меню:

* Карта и маршрут: В центре экрана отображается карта, на которой отображается маршрут дрона. Маршрут представлен в виде линии, соединяющей точки съёмки, по которым дрон должен пролететь. Это визуальное представление планируемого маршрута.
* Кнопки переключения меню: В верхней части интерфейса расположены кнопки, которые позволяют переключать отображение левого и правого меню. Пользователь может скрыть или показать эти меню по своему усмотрению для освобождения места на экране.
* Панель инструментов: На панели инструментов расположены кнопки для выполнения различных действий:
* Отмена (Undo): Кнопка, позволяющая отменить последнее действие при построении полигона.
* Навигация (Navigation): Кнопка, позволяющая пользователю вернуться к режиму навигации после использования других инструментов. Это удобно, если пользователь не хочет случайно переместить точку полигона области съемки.
* Рисование (Draw): Кнопка, активирующая режим построения полигона. В этом режиме пользователь может указать область, которую дрон должен снять.
* Установка стартовой точки (Set Starting Point): Кнопка, позволяющая пользователю установить стартовую точку для маршрута дрона. После выбора этой опции пользователь может щелкнуть на карте, чтобы установить стартовую точку.
* Расчет (Calculate): Кнопка, которая выполняет валидацию введенных данных и отправляет их на обработку на сервере. Результаты расчета маршрута отображаются на карте.

Центральное меню предоставляет основные инструменты и функции для работы с картой, рисования маршрута и выполнения других операций, связанных с управлением маршрутом дрона.

Левое меню включает два блока: блок UAV (Беспилотный Летательный Аппарат) и блок Camera (Камера).

1. Блок UAV:

* Выбор и отображение списка UAV: В верхней части блока расположен выпадающий список (select), который позволяет пользователю выбрать один из доступных UAV. Каждый элемент списка содержит имя UAV. При выборе UAV из списка его параметры отображаются в полях ниже.
* Кнопка "Fetch": Рядом с выпадающим списком находится кнопка "Fetch", которая позволяет пользователю обновить список UAV, загрузив его с базы.
* Кнопка "UAV details": Эта кнопка позволяет пользователю переключать отображение блока с параметрами UAV. При нажатии на кнопку блок с параметрами UAV может быть скрыт или показан на экране.

Внутри блока UAV параметры UAV отображаются в виде полей для ввода которые позволяют пользователю просматривать и редактировать параметры UAV. Некоторые из доступных параметров UAV включают:

* ID: Идентификатор UAV (только для чтения).
* Name: Имя UAV.
* Max Payload Mass: Максимальная масса груза, которую может нести UAV.
* Flight Duration: Длительность полета UAV.
* Takeoff speed: Скорость взлета UAV.
* Flight speed: Скорость полета UAV.
* Min altitude: Минимальная высота полета UAV.
* Max altitude: Максимальная высота полета UAV.

Под блоком с параметрами UAV располагается панель инструментов, которая содержит следующие кнопки:

* "Update": Кнопка, которая позволяет пользователю обновить параметры выбранного UAV.
* "New": Кнопка, которая позволяет пользователю создать новый UAV с указанными в полях параметрами.
* "Delete": Кнопка, которая позволяет пользователю удалить выбранный UAV.
* "Undo": Кнопка, которая позволяет пользователю отменить последнее изменение параметров UAV.

Блок Camera:

Выбор и отображение списка Camera: Аналогично блоку UAV, в верхней части блока расположен выпадающий список (select), который позволяет пользователю выбрать одну из доступных камер. Каждый элемент списка содержит имя камеры. При выборе камеры из списка её параметры отображаются в полях ниже.

* Кнопка "Fetch": Рядом с выпадающим списком находится кнопка "Fetch", которая позволяет пользователю обновить список камер, загрузив его с базы.
* Кнопка "Camera details": Эта кнопка позволяет пользователю переключать отображение блока с параметрами камеры. При нажатии на кнопку блок с параметрами камеры может быть скрыт или показан на экране.

Внутри блока Camera параметры камеры отображаются в виде полей для ввода (input) и меток (label), которые позволяют пользователю просматривать и редактировать параметры камеры. Некоторые из доступных параметров камеры включают:

* ID: Идентификатор камеры (только для чтения).
* Name: Имя камеры.
* Mass (grams): Масса камеры в граммах.
* X-axis FOV (degrees): Угол обзора камеры по оси X в градусах.
* Resolution X: Разрешение камеры по оси X.
* Resolution Y: Разрешение камеры по оси Y.

Под блоком с параметрами камеры располагается панель инструментов, которая содержит следующие кнопки:

* "Update": Кнопка, которая позволяет пользователю обновить параметры выбранной камеры.
* "New": Кнопка, которая позволяет пользователю создать новую камеру с указанными в полях параметрами.
* "Delete": Кнопка, которая позволяет пользователю удалить выбранную камеру.
* "Undo": Кнопка, которая позволяет пользователю отменить последнее изменение параметров камеры.

Правое меню состоит из трех блоков: "Altitude Menu" (меню высоты), "Algorithm Menu" (меню алгоритма) и "Mission Parameters" (параметры миссии). Давайте рассмотрим каждый из них подробнее:

Блок "Altitude Menu":

* Здесь можно выбрать режим ввода высоты. Есть два варианта: "Manual altitude input" (ручной ввод высоты) и "Calculate using sm/px" (расчет высоты на основе sm/px).
* Если выбран режим ручного ввода, можно указать процент перекрытия и ввести высоту вручную.
* Если выбран режим расчета на основе sm/px, нужно указать процент перекрытия и ввести значение sm/px. После ввода всех параметров можно нажать кнопку "Calculate Altitude" (рассчитать высоту) для получения результата.

Блок "Algorithm Menu":

* Здесь можно выбрать алгоритм для расчета маршрута. Есть два варианта: "Nearest Neighbor" (ближайший сосед) и "Brute Force" (полный перебор).

Блок "Mission Parameters":

* Здесь отображаются параметры миссии, такие как длина маршрута, продолжительность миссии и количество фотографий.
* Также есть кнопка "Export to GeoJSON" (экспорт в формате GeoJSON), которая позволяет экспортировать данные миссии в этот формат для дальнейшего использования.

1. **Тестирование системы**

В ходе разработки и последующего программного обеспечения были разработаны и проведены комплексные тесты для проверки функциональности и стабильности работы системы. Проведенные тесты относятся к виду функционального тестирования, а также тестирования пользовательского интерфейса.

Тестирование проводилось в соответствии с предложенной методологией, при которой каждый из тестовых сценариев включал в себя конкретные шаги для воспроизведения действий пользователя, а также определенный ожидаемый результат. В процессе тестирования внимание было уделено как отдельным функциям, так и взаимодействию между различными компонентами системы.

Процесс тестирования осуществлялся с использованием различных входных данных, включая граничные значения и невалидные данные, чтобы оценить устойчивость и надежность системы в различных условиях, а также чтобы выявить и устранить возможные недостатки. Примеры тестов приведены в таблице

| **№** | **Название теста** | **Шаги** | **Ожидаемый результат** | **Тест пройден?** |
| --- | --- | --- | --- | --- |
| 1 | Загрузка дрона | 1. В левом меню выберать идентификатор дрона.  2. Нажать кнопку "Fetch". | В полях деталей дрона отображается информация о выбранном дроне. | да |
| 2 | Редактирование дрона | 1. Выберать дрон.  2. Включить режим редактирования.  3. Изменить параметры дрона.  4. Нажать кнопку "Update". | Информация о дроне в базе данных обновляется. | да |
| 3 | Расчет маршрута | 1. Ввети необходимые параметры, высоту полета, алгоритм, обозначить точку старта, обозначить зону съемки, выбрать дрон, выбрать камеру.  2. Нажать кнопку. "Calculate" | На карте отображается рассчитанный маршрут. | да |
| 4 | Некорректная попытка расчета маршрута | 1. Не ввести необходимые параметры, высоту полета, алгоритм, точку старта, зону съемки, дрон, камеру.  2. Нажать кнопку "Calculate". | Отображается ошибка о том, что не все параметры выбраны | да |
| 5 | Проверка параметров миссии | 1. Просмотрите отображаемые параметры миссии в правом меню. | Параметры миссии (длина маршрута, продолжительность миссии, количество фотографий) отображаются корректно. | да |
| 6 | Визуализация маршрута | 1. После расчета маршрута проверьте визуализацию маршрута на карте. | Маршрут корректно отображается на карте. | да |
| 7 | Экспорт в GeoJSON | 1. После расчета маршрута нажать кнопку "Export to GeoJSON". 2. Выбрать путь сохранения | Файл GeoJSON с информацией о маршруте сохраняется. | да |
| 8 | Загрузка камеры | 1. В левом меню выберать идентификатор камеры.  2. Нажать кнопку "Fetch". | В полях деталей камеры отображается информация о выбранной камере. | да |
| 9 | Редактирование камеры | 1. Выбрать камеру.  2. Включить режим редактирования.  3. Изменить параметры камеры.  4. Нажать кнопку "Update". | Информация о камере в базе данных обновляется. | да |
| 10 | Выбор алгоритма | 1. В правом меню выберите желаемый алгоритм. | Выбранный алгоритм устанавливается как текущий. | да |
| 11 | Ручной ввод высоты | 1. В правом меню выберите режим ручного ввода высоты.  2. Введите желаемую высоту. | Введенная высота устанавливается как текущая. | да |
| 12 | Расчет высоты | 1. В правом меню выберите режим расчета высоты.  2. Введите значение sm/px.  3. Нажмите кнопку "Calculate Altitude". | Высота автоматически рассчитывается и устанавливается как текущая. | да |
| 13 | Изменение параметра перекрытия | 1. В правом меню ввести значение в поле "Overlap (%)". | Значение перекрытия изменяется в соответствии с введенным значением. | да |
| 14 | Проверка работы с отсутствующими данными | 1. Нажать кнопку "Calculate", когда база данных пустая. | Система сообщает о том, что дрон не выбран. | да |