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:

1. **Реализация интерфейса системы (пользовательский и/или программный интерфейс)**
2. **Тестирование системы**