# ЗАДАНИЕ 1. МЕНЕДЖМЕНТ ПРОЕКТА В СТИЛЕ KANBAN

## Упражнение 1.1. Изучить возможности управления проектами

1. Выбрать систему управления проектами из перечня: *Github Project, или Trello* или *Targetprocess*.

Выбрали trello

1. Познакомиться с учебными материалами для выбранной системы управленияпроектами.

Ознакомились

## Упражнение 1.2. Создать проект и распределить задачи

1. Создать проект типа Kanban в выбранной системе управления проектами согласно:

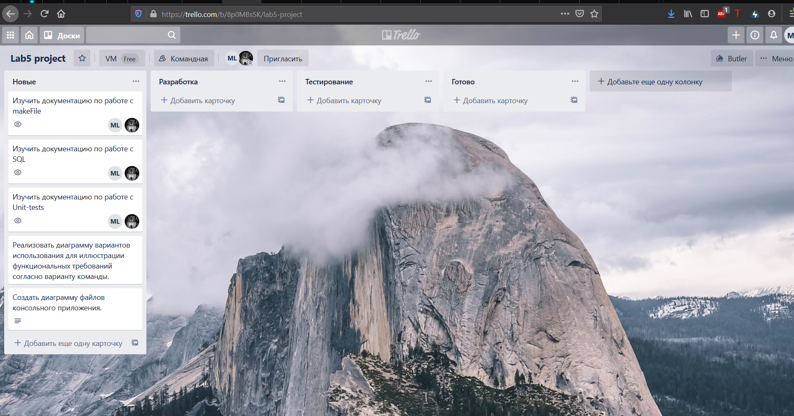
◦ Порядок действий при создании проекта в Github Projects

◦ Порядок действий при создании проекта в Trello ◦ Порядок действий при создании проекта в Targetprocess

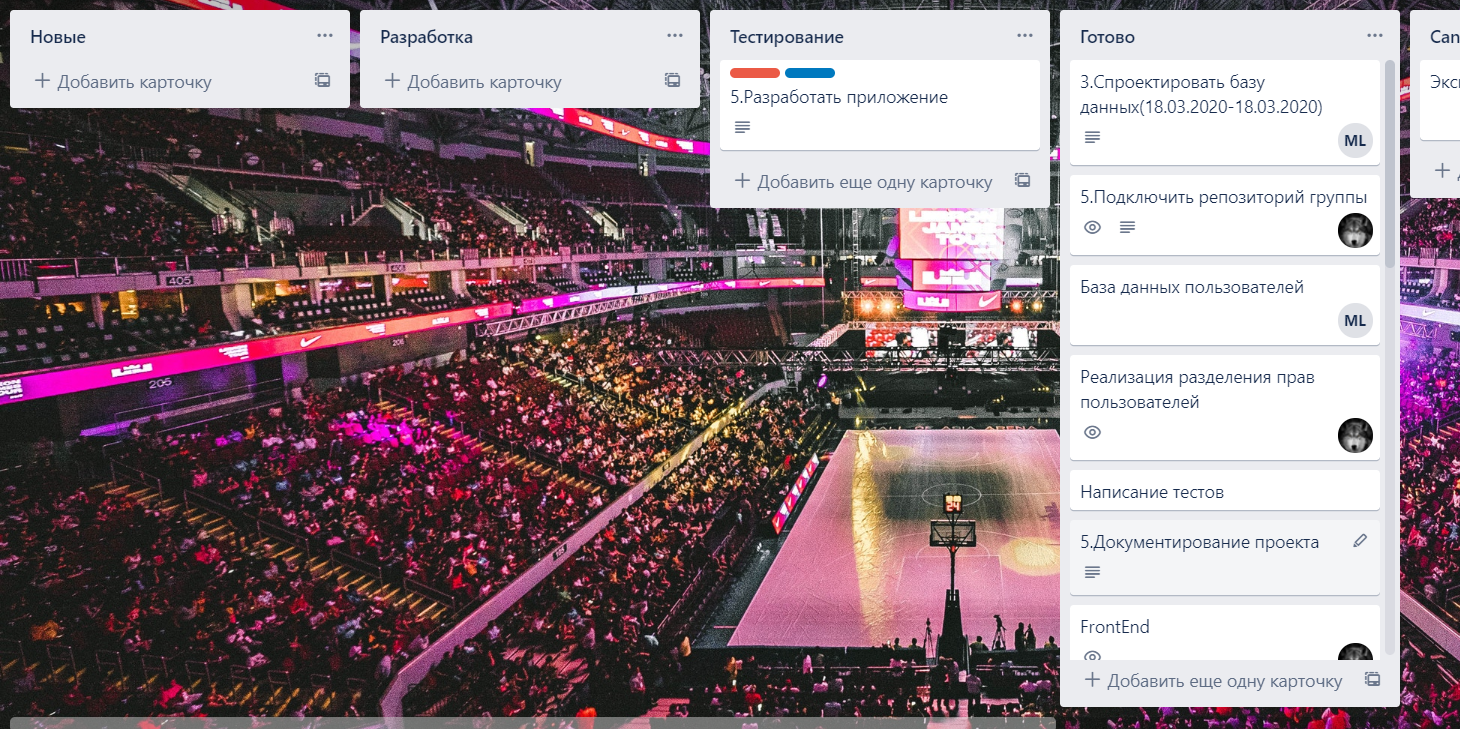
1. Добавить в проект участников.
2. Добавить списки и задачи.
3. Распределить задачи.
4. Использовать проект для управления командой.

Выбрали trello.

Состояние trello на момент начала разработки:



Состояние trello на момент окончания разработки:



# ЗАДАНИЕ 2. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

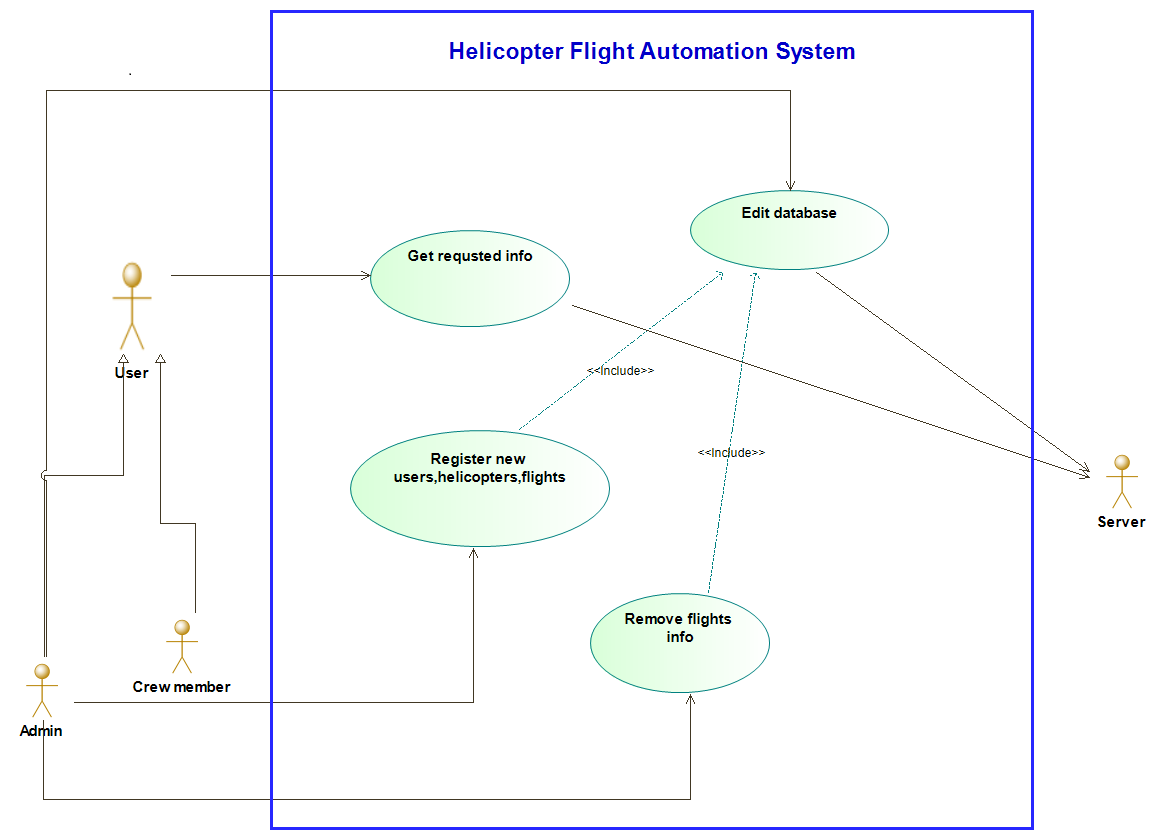
1. Познакомиться с документом [UML and C.pdf](https://drive.google.com/file/d/0B7CWhSn9h7v0VUJWRWtsS3FnWFk/view).
2. Реализовать диаграмму вариантов использования для иллюстрации функциональных требований согласно варианту команды.
3. Используя диаграмму классов, создать диаграмму файлов консольного приложения, иллюстрирующую файлы, типы, атрибуты, функции и отношения.

Для разработки использовать десктопные приложения:

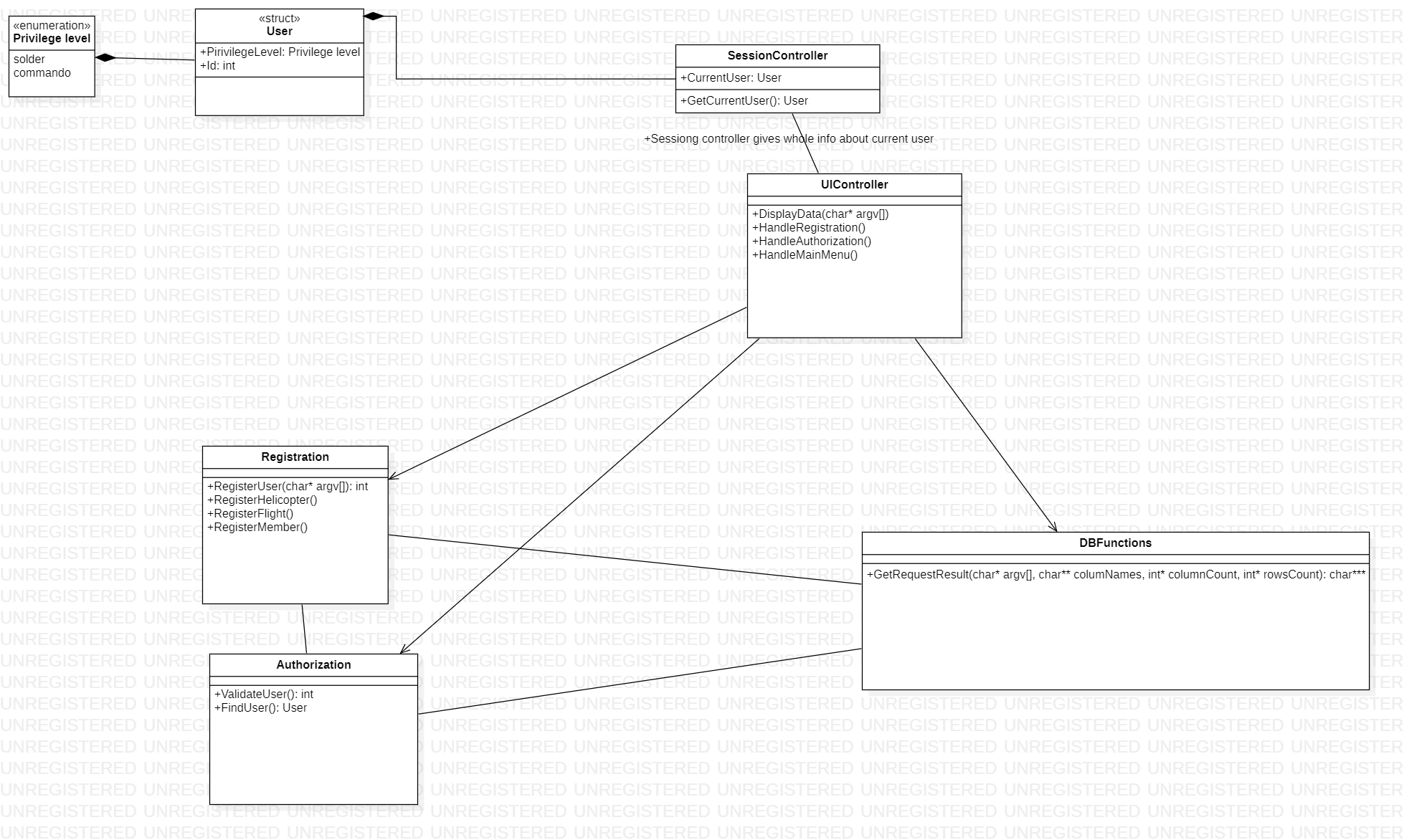
* VisualParadigm Community Edition (<http://visual-paradigm.com/>)
* StarUML 2 (<http://staruml.io/>) или онлайн-сервисы:
* Creately (<http://creately.com/>)
* GenMyModel (<http://www.genmymodel.com/>)

Использовали StarUML

Use-case:



UML:



# ЗАДАНИЕ 3. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ БАЗЫ ДАННЫХ И НАПОЛНЕНИЕ ДАННЫМИ

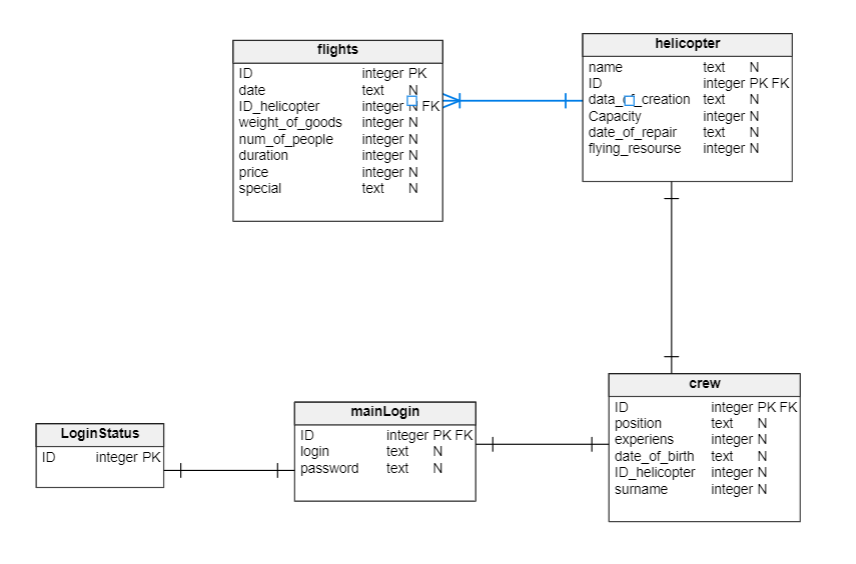
Для решения необходимо использовать [http://www.vertabelo.com](http://www.vertabelo.com/) для проектирования архитектуры базы данных.

## Упражнение 3.1. Изучить функционал Vertabelo

Ознакомьтесь с видео <https://youtu.be/hU-A08K08-Y>.

## Упражнение 3.2. Спроектировать базу данных

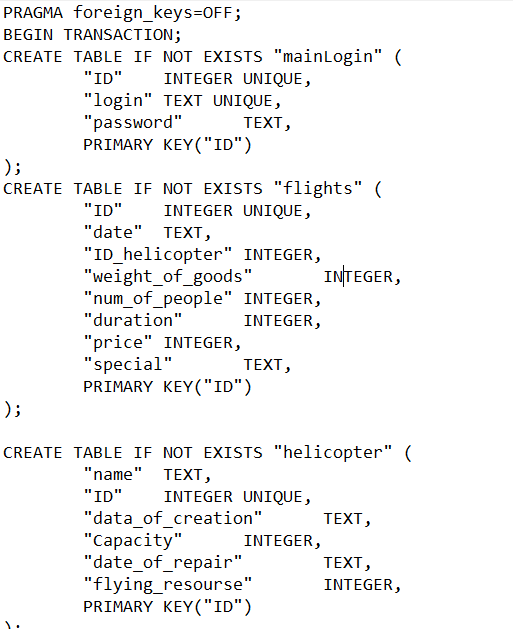
Спроектируйте базу данных для хранения данных о пользователях и других сущностях согласно варианту задания.



## Упражнение 3.3. Сгенерировать sql-скрипт для sqlite

Сгенерируйте файл SQL-скрипта для БД SQLite. Импортируйте файл в Valentina Studio или SQLBrowser и заполните таблицы БД данными.

Экспортировали



# ЗАДАНИЕ 4. НЕПРЕРЫВНАЯ СБОРКА ПРОЕКТА И ТЕСТИРОВАНИЕ

1. Изучите основные материалы по использованию travis-ci для сборки и тестирования консольного проекта.
2. Продемонстрируйте результаты сборки и тестов

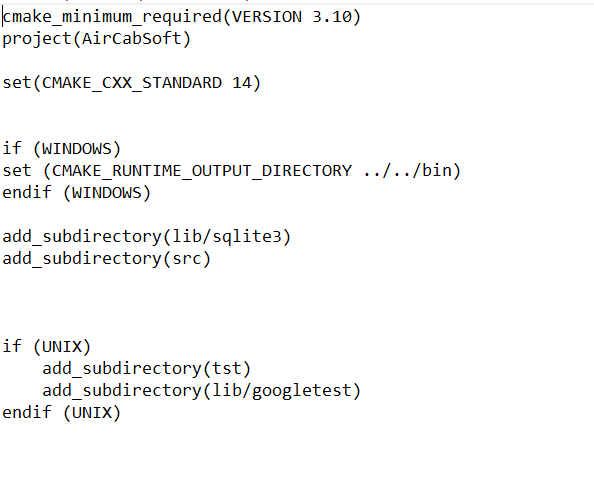
**Сборка:**

Travis .yaml:



Сборка происходит в папке Build. Для удобства сборки в облаке был использован cmake .

Корневой CMakeLists:



В облаке, для удобства, бинарники создаются не в папке bin , а в специальной папке внутри Build. При сборке под windows ( версия которая идет в production) бинарники хранятся согласно стандарту в папке bin.

В случае сборки для релиза , файлы сборки с самим бинарником будут залиты на сервер (AWS S3 bucket),

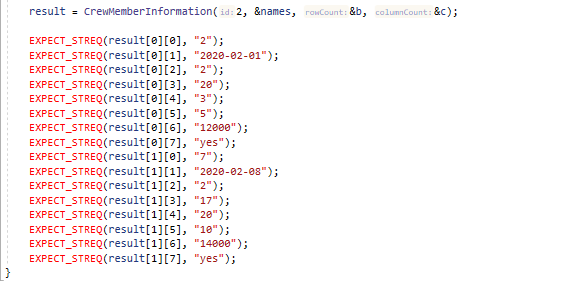
**Тестирование:**

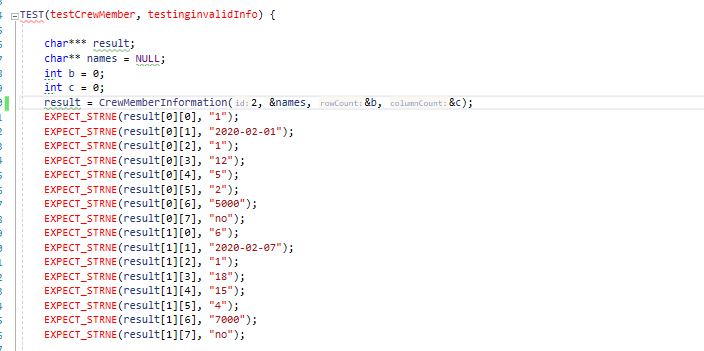
Для тестирования использовалась библиотека google Tests.

По неизвестным причинам, данная библиотека не билдилась под windows, поэтому все тесты проходили исключительно в облаке.

Пример тестов (получение информации о члене экипажа):

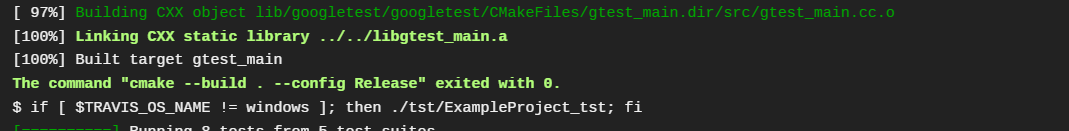




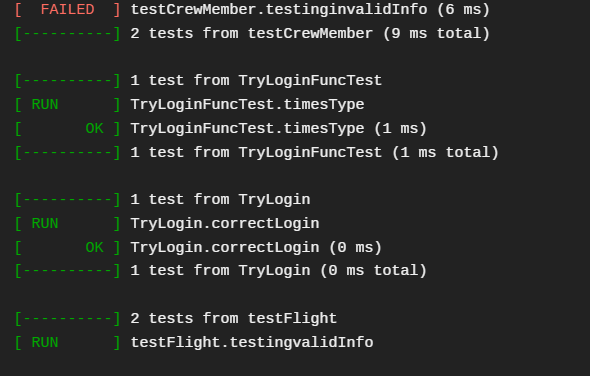


Примеры сборки и тестирования в облаке:

Успешная сборка:



Тестирование:



# ЗАДАНИЕ 5. РАЗРАБОТКА ПРИЛОЖЕНИЯ

## Упражнение 5.1. Подключить репозиторий группы

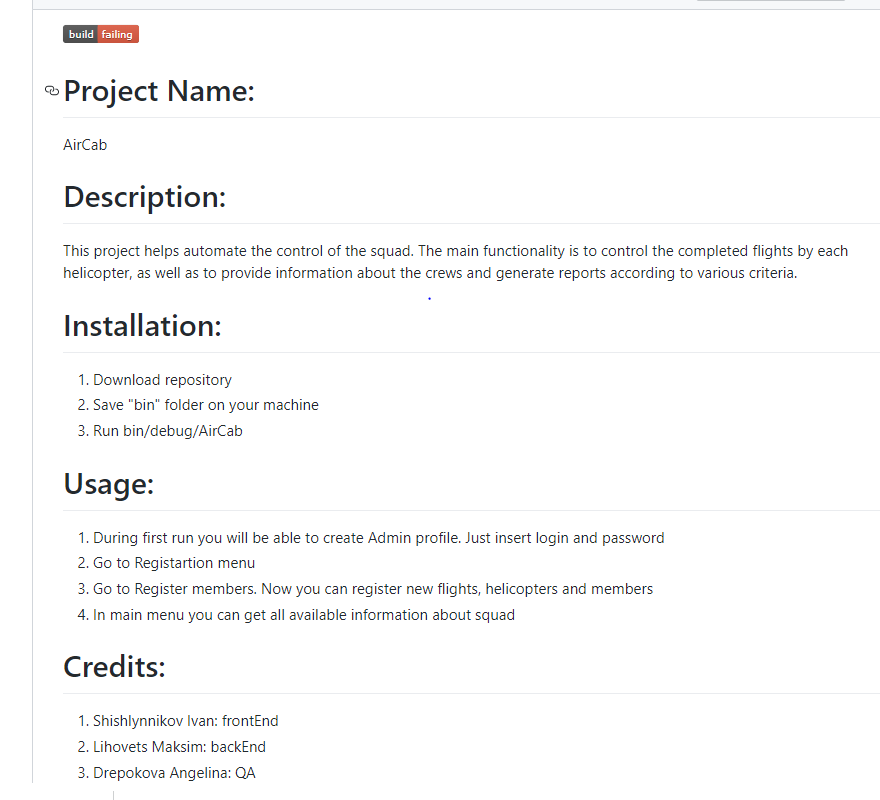
Подключить репозиторий на github согласно ссылке для Вашей группы.

В процессе выполнения разработки проиллюстрировать работу с несколькими ветками в репозитории проекта, добавление и приём коммитов, слияние и перемещение веток и т. д.

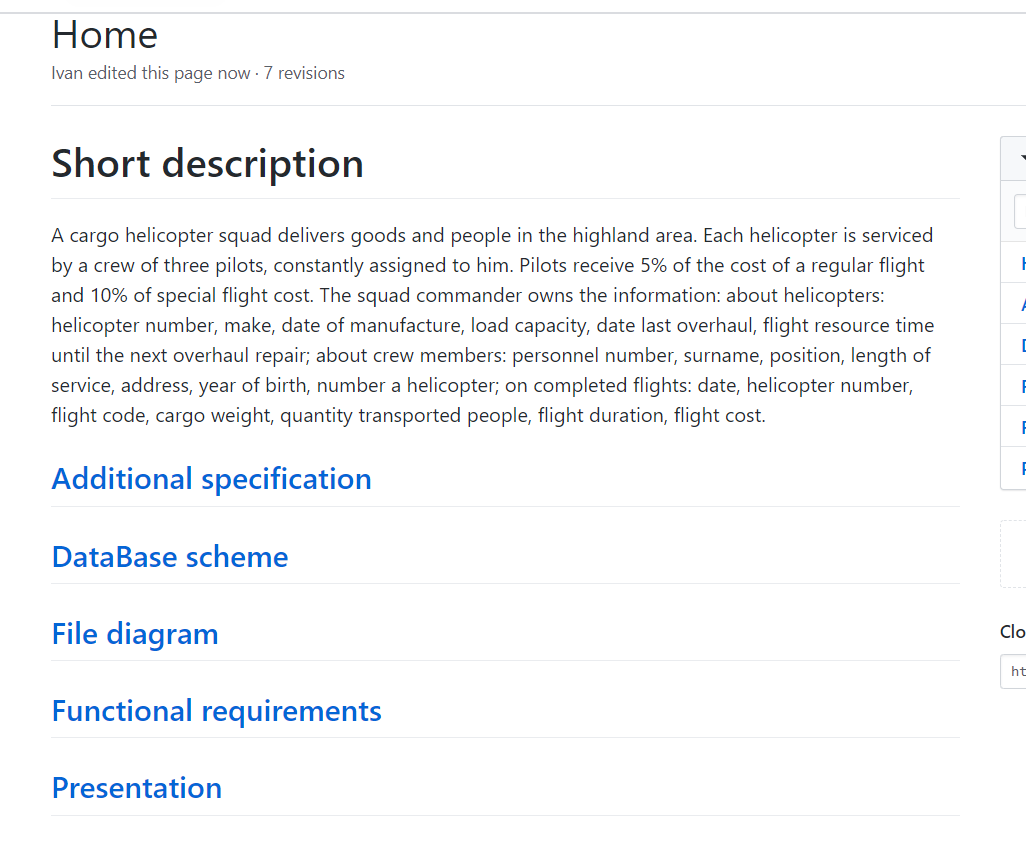
## Упражнение 5.2. Документирование проекта

Изучить документацию <https://guides.github.com/features/wikis/> и документировать проект в Readme и wiki репозитория согласно следующим требованиям:

Readme:



Wiki:



Более подробно с Readme и Wiki можно ознакомиться в самом репозитории

## Упражнение 5.3. Разработать приложение

Разработать консольное приложение на языке Си, которое позволяет пройти аутентификацию и авторизацию и выполнять операции согласно варианту задания.

Проект должен состоять из нескольких .c файлов. Структура проекта должна соответствовать модели КИС, содержать папки bin, build, includes, src.

Документацию проекта, включая постановку задачи и другие материалы представить в вики проекта.

Проиллюстрировать различные виды запросов к базе данных.

### Вариант 3. «Воздушный извозчик».

Отряд грузовых вертолетов осуществляет доставку грузов и людей в высокогорном районе. Каждый вертолет обслуживается экипажем из трех пилотов, постоянно закрепленных за ним. Летчики получают по 5 % от стоимости обычного рейса и 10 % от стоимости спецрейса.

**Командир авиаотряда владеет информацией:**

*о вертолетах:* номер вертолета, марка, дата изготовления, грузоподъемность, дата последнего капитального ремонта, летный ресурс времени до следующего капитального ремонта;

*о членах экипажа:* табельный номер, фамилия, должность, стаж, адрес, год рождения, номер вертолета;

*о выполненных рейсах:* дата, номер вертолета, код рейса, масса груза, количество перевезенных людей, длительность полета, стоимость рейса.

**Член экипажа может узнать информацию:** только свои данные или данные о своём вертолете - смотрите ниже по пункту, помеченному \* (звездочкой).

**Необходимо выполнить:**

1. Создать таблицы БД с учетом ограничений целостности данных.
2. Используя оператор Select, выдать следующую информацию:

* по каждому вертолету – общее количество часов, которые они налетали после капитального ремонта, и ресурс летного времени (\*);
* по каждому вертолету – перечень выполненных рейсов с указанием общей массы перевезенных грузов и количества человек за указанный период(\*);
* по всем вертолетам, выполнявшим спецрейсы, – общее количество рейсов, общая масса перевезенных грузов, общая сумма заработанных денег;

по всем вертолетам, выполнявшим обычные рейсы, – общее количество рейсов,

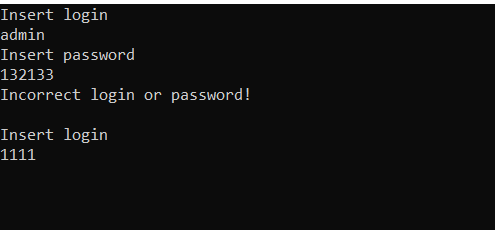
общая масса перевезенных грузов, общая сумма заработанных денег;

* по вертолету, выполнившему максимальное количество рейсов, – все сведения об его экипаже и количестве заработанных денег;
* по экипажу, заработавшему максимальное количество денег, – все сведения о выполненных им рейсах.
* по экипажу(или члену экипажа), – все сведения о выполненных им рейсах (\*).
  1. Обеспечить с помощью операторов Insert, Update, Delete обновление информации в указанных таблицах.
  2. Создать функцию, который при внесении информации в таблицу рейсов, проверяет, небудет ли превышен ресурс летного времени для вертолета, и если это так, то запрещает вносить информацию в таблицу.
  3. Создать функцию, которая за указанный период определяет количество денег,начисленных экипажам авиаотряда за перевозки. В качестве параметра передать начальную дату периода и конечную дату периода. Результаты занести в специальную таблицу.

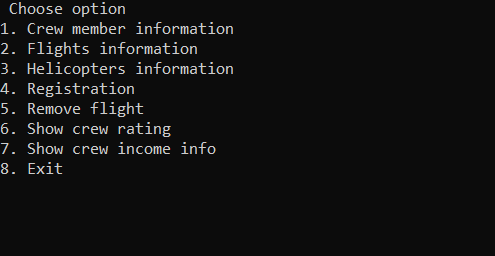
6.Создать функцию, которая за указанный период определяет количество денег, начисленных указанному летчику(\*).

7.Создать функцию, которая за указанный период определяет количество денег, начисленных указанному летчику за указанный рейс(сы) или спецрейс(сы) (\*).

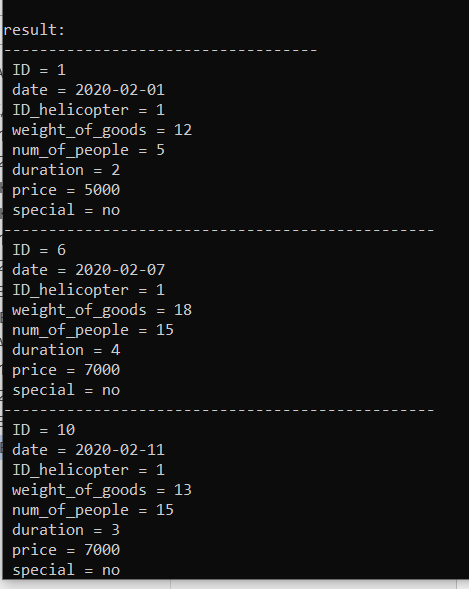
Авторизация:



Главное меню:



Пример вывода информации о члене экипажа:



Более подробно с программой можно ознакомиться в самом репозитории

(<https://github.com/VMTeamBSU/Lab5TPNo-money>)