|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ**

**Техническое задание**

**по дисциплине «Разработка** **интернет приложений»**

**Тема предметной области «Полёты к Gateway»**

**Выполнил:**

Студент группы ИУ5-53Б

Саргсян А.А.

(дата, подпись)

**Проверил:**

Преподаватель

Канев А.И.

(дата, подпись)

Москва, 2023

**СОДЕРЖАНИЕ**

[1. Цель 3](#_Toc151468118)

[2. Назначение 3](#_Toc151468119)

[3. Задачи 3](#_Toc151468120)

[4. Методы веб-сервиса 3](#_Toc151468121)

[5. Описание функцоинальных требований 8](#_Toc151468122)

[6. Требования к аппаратному обеспечению 12](#_Toc151468123)

[7. Требования к программному обеспечению 13](#_Toc151468124)

# Цель

Разработка системы для ведения учета полетов космических модулей лунной станции Lunar Gateway в рамках определенных миссий на станцию и в области Луны. Система включает в себя веб-сервис, веб-приложение, мобильное приложение и выделенный сервис расчета результата одобрения финансирования космических миссий.

# Назначение

Система предназначена для представителей космических агентств и для администраторов NASA – агентства, возглавляющего программу по созданию и управлению Lunar Gateway. Система предоставляет пользовательский интерфейс для просмотра и управления услугами и заявками. Для получения доступа пользователю необходимо создать заявку на получение доступа к выбранной космической миссии. Администраторы могут взаимодействовать с базой данных и выполнять модерацию заявок. Также, администраторы имеют возможность редактировать существующие и создавать новые миссии.

# Задачи

* 1. Разработка дизайна системы;
  2. Создание структуры базы данных в PostgreSQL;
  3. Реализация REST-API на Go 1.21.5;
  4. Создание интерфейса гостя на React-Vite;
  5. Добавление аутентификации и Swagger в веб-сервис;
  6. Добавление интерфейса пользователя;
  7. Добавление нативного приложения React Native, Redux Toolkit;
  8. Разработка асинхронного сервиса для расчета результата одобрения финансирования миссий на Django;
  9. Разработка интерфейса администратора;
  10. Разработка и документирование ТЗ, РПЗ и диаграммы.
  11. Развёртка веб-приложения на GitHub Pages для предварительной демонстрации работы.

# Методы веб-сервиса

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | | **Метод** | **URL** | **Описание** | **Входные данные** | **Выходные данные** |
| * 1. **Услуга – модуль станции** | | | | | | |
|  | GET | | api/modules/ | Получить список услуг и черновую заявку с фильтрацией по названию  Будут возвращены все модули со статусом is\_deleted = false  Доступно всем пользователям | {  Name: string,  } | {  "draft\_mission":{  "uuid": UUID,  "module\_count": INTEGER  },  "modules": [ {  "uuid": UUID,  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR,  "is\_deleted": BOOLEAN  },  ...  ]  } |
|  | GET | | api/modules/{module\_id}/ | Получить информацию о услуге  Доступно всем пользователям | Отсутствует | {  "uuid": UUID,  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR,  "is\_deleted": BOOLEAN  } |
|  | POST | | api/modules/ | Создать новую услугу  Доступно только авторизированным модераторам | {  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR,  "is\_deleted": BOOLEAN  } | {  "uuid": UUID,  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR,  "is\_deleted": BOOLEAN  } |
|  | POST | | api/modules/{module\_id}/add\_to\_mission/ | Создать заявку по услуге  Доступно только авторизированным пользователям | Отсутствует | Отсутствует |
|  | PUT | | api/modules/{module\_id}/ | Обновить услугу  Доступно только авторизированным модераторам | {  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR,  "is\_deleted": BOOLEAN  } | {  "uuid": UUID,  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR,  "is\_deleted": BOOLEAN  } |
|  | DELETE | | api/modules /{module\_id}/ | Удалить услугу – модуль станции  Доступно только авторизированным модераторам | Отсутствует | Отсутствует |
| * 1. **Заявка – космическая миссия** | | | | | | |
|  | GET | | api/missions?status=&date\_approve\_start=&date\_approve\_end=/ | Получить список заявок с фильтрацией по дате формирования и статусу  Доступно только авторизированным пользователям | Отсутствует | {  "missions":  [  "uuid":UUID,  "name": VARCHAR,  "status": VARCHAR,  "date\_created": TIMESTAMP,  "date\_approve": TIMESTAMP,  "date\_end":  TIMESTAMP,  "date\_start\_mission":  DATE,  "description": TEXT,  "moderator":  VARCHAR,  "customer": VARCHAR  ]  } |
|  | GET | | api/missions/<int:pk>/ | Получить информацию о заявке  Доступно только авторизированным пользователям | Отсутствует | {  "missions":  {  "uuid":UUID,  "name": VARCHAR,  "status": VARCHAR,  "date\_created": TIMESTAMP,  "date\_approve": TIMESTAMP,  "date\_end":  TIMESTAMP,  "date\_start\_mission":  DATE,  "description": TEXT,  "moderator":  VARCHAR,  "customer": VARCHAR  }  "modules":  [  {…}  ]  } |
|  | PUT | | api/mars\_station/<int:pk>/update/ | Обновить заявку – марсианскую станцию  Доступно только авторизированным пользователям | {  "id": INTEGER,  "type\_status": VARCHAR,  "date\_create": VARCHAR,  "date\_form": VARCHAR,  "date\_close": VARCHAR,  "status\_task": INTEGER,  "status\_mission": INTEGER,  "id\_employee": INTEGER,  "id\_moderator": INTEGER,  "id\_transport": INTEGER  } | {  "id": INTEGER,  "type\_status": VARCHAR,  "date\_create": VARCHAR,  "date\_form": VARCHAR,  "date\_close": VARCHAR,  "status\_task": INTEGER,  "status\_mission": INTEGER,  "id\_employee": INTEGER,  "id\_moderator": INTEGER,  "id\_transport": INTEGER  } |
|  | PUT | | api/mars\_station/<int:pk>/update\_by\_user/ | Обновить заявку – марсианскую станцию пользователем  Код статуса:  1 - «Черновик»,  2 - «В работе»,  5 - «Удален»  Доступно только авторизированным пользователям | {  "status\_task": INTEGER  } | Отсутствует |
|  | PUT | | api/mars\_station/<int:pk>/update\_by\_admin/ | Обновить заявку – марсианскую станцию модератором  Код статуса:  3 - «Завершена»,  4 - «Отменена».  Доступно только авторизированным модератором.  Доступно только со статусом:  2 – «В работе» | {  "status\_task": INTEGER,  "status\_mission": INTEGER  } | Отсутствует |
|  | DELETE | | api/mars\_station/<int:pk>/delete/ | Удалить заявку – марсианскую стацнию  Доступно только авторизированным модератором | Отсутствует | Отсутствует |
| * 1. **Местоположение – вспомогательная таблица (М-М)** | | | | | | |
|  | PUT | | api/location/<int:pk>/update/ | Обновить вспомогательные таблицы  Доступно только авторизированным пользователям | {  "id\_geographical\_object": INTEGER,  "id\_geographical\_object": INTEGER  } | {  "id": INTEGER,  "id\_geographical\_object": INTEGER,  "id\_geographical\_object": INTEGER  } |
|  | DELETE | | api/location/<int:pk>/delete/ | Удалить вспомогательные таблицы  Доступно только авторизированным пользователям | Отсутствует | Отсутствует |
| * 1. **Пользователь** | | | | | | |
|  | POST | | api/sign\_up | Создать новый аккаунт | {  "login": VARCHAR,  "password": VARCHAR, "is\_superuser": BOOL,  "is\_staff": BOOL  } | {  "id": INTEGER,  "login": VARCHAR,  "password": VARCHAR, "is\_superuser": BOOL,  "is\_staff": BOOL  } |
|  | POST | | api/login | Войти в аккаунт | {  "login": VARCHAR,  "password": VARCHAR,  } | Отсутствует |
|  | POST | | api/logout | Выход из аккаунта  Доступно только авторизированным пользователям | Отсутствует | Отсутствует |
| * 1. **Основной сервис** | | | | | | |
|  | PUT | | api/update\_async\_result/ | Обновляет результаты отложенного действия, выполненного асинхронным сервисом | {  "task\_id": INTEGER,  "message": VARCHAR  } | {  "success": INTEGER  } |
| * 1. **Выделенный сервис** | | | | | | |
|  | POST | | api/async\_task/ | Выполняет отложенное действие в системе. Действие запускается асинхронно с задержкой 5-10 секунд. Результат действия случайный (например, успех/неуспех).  Результат обновляет одно поле в заявке. | {  "request\_id": INTEGER  } | Отсутствует |

1. **Функциональные требования:**
   1. Главное меню. На этой странице находится текст с приветствием, если пользователь не авторизирован, и меню со списком доступных пунктов.
      1. Доступна всем пользователям.
      2. Действия
         1. Войти – переход на страницу 5.2. Только для гостей.
         2. Зарегистрироваться – переход на страницу 5.3. Только для гостей.
         3. Открыть список карт – переход на страницу 5.4. Только для авторизированных пользователей.
         4. Открыть список выбранных карт – переход на страницу 5.7. Только для авторизированных пользователей.
         5. Список заявок – переход на страницу 5.6. Только для авторизированных пользователей.
         6. Создать карту – переход на страницу 5.8. Только для администраторов.
         7. Таблица карт – переход на страницу 5.9. Только для администраторов.
   2. Страница с формой авторизации. На этой странице находится форма авторизации.
      1. Доступна гостям
      2. Действия
         1. Войти – производит запрос (метод 4.1.2.).
         2. Зарегистрироваться – перенаправляет на страницу 5.3.
   3. Страница с формой регистрации. На этой странице находится форма авторизации.
      1. Доступна гостям
      2. Действия
         1. Зарегистрироваться – производит запрос (метод 4.1.1.), в котором передаются введенные данные формы.
         2. Войти - перенаправляет пользователя на страницу 5.2.
   4. Страница со списком карт. На этой странице располагается список всех карт и панель фильтрации карт.
      1. Доступна всем пользователям.
      2. Выводится информация о картах в виде карточек (метод 4.2.2.).
         1. Изображение карты.
         2. Название карты.
         3. Ширина карты.
         4. Высота карты.
         5. Максимальное количество игроков.
      3. Действия
         1. Отфильтровать карты по указанному названию и максимальному количеству игроков (метод 4.2.2.).
         2. Добавить карту в заявку (метод 4.3.4.). Только для авторизированных пользователей.
         3. Удалить карту из заявки (метод 4.3.5.). Только для авторизированных пользователей.
         4. Подробнее – перенаправляет на страницу 5.5.
   5. Страница с подробным описанием карты
      1. Доступна всем пользователям.
      2. Выводится информация о карте в построчном формате.
         1. Изображение карты.
         2. Название карты.
         3. Ширина карты.
         4. Высота карты.
         5. Максимальное количество игроков.
         6. Описание карты.
      3. Действия
         1. Добавить карту в заявку (метод 4.3.4.). Если карта доступна. Только для авторизированных пользователей.
         2. Удалить карту из заявки (метод 4.3.5.). Только для авторизированных пользователей.
         3. Редактировать – перенаправляет на страницу 5.8. Только для администраторов.
         4. Назад – перенаправляет на предыдущую страницу.
   6. Страница со списком заявок
      1. Доступна только авторизированным пользователям.
      2. Выводится информация о заявках в табличном формате (методы 4.3.1. и 4.3.8.).
         1. Статус заявки.
         2. Дата создания.
         3. Дата формирования.
         4. Дата завершения.
         5. Кем обработана. Только для модераторов и администраторов.
         6. Список выбранных карт (в выпадающем списке).
      3. Действия
         1. Открыть заявку – перенаправляет на страницу 5.7.
         2. Установить заявке статус «отклонена» или «завершена» (метод 4.3.9.). Только для модераторов и администраторов.
         3. Отфильтровать список заявок по диапазону дат и/или имени автора заявки (метод 4.3.8.). Только для модераторов и администраторов.
   7. Страница с подробным описанием заявки. Пользователь может изменить карты в заявке. Модератор может просматривать заявки всех пользователей.
      1. Доступна только авторизированным пользователям.
      2. Выводится информация о заявке в карточной форме. Список карт, выбранных в заявке, в виде карточек (метод 4.3.2.).
         1. Изображение карты.
         2. Название карты.
         3. Ширина карты.
         4. Высота карты.
         5. Максимальное количество игроков на карте.
      3. Действия
         1. Удалить карту из заявки (метод 4.3.5.). Если заявка находится в статусе черновика. Только для владельца заявки.
         2. Вернуть карту в заявку (метод 4.3.4.). Если заявка находится в статусе черновика. Только для владельца заявки.
         3. Сформировать заявку (метод 4.3.6.). Если заявка находится в статусе черновика. Только для владельца заявки.
         4. Удалить заявку (метод 4.3.7.). Если заявка находится в статусе черновика. Только для владельца заявки.
         5. Открыть карту – перенаправляет на страницу 5.5.
   8. Страница редактирования/создания карты. Администратор может изменить существующую или создать новую карту.
      1. Доступна только администраторам.
      2. Выводится информация о карте в построчном формате.
         1. Изображение карты.
         2. Название карты.
         3. Ширина карты.
         4. Высота карты.
         5. Максимальное количество игроков на карте.
         6. Описание
         7. Доступность.
      3. Изменение полей карты. Можно изменять все вышеперечисленные поля карты (методы 4.2.5. и 4.2.6.).
   9. Страница с таблицей карт. Предоставляет администратору удобный способ отображения всех карт.
      1. Доступна только администраторам.
      2. Выводит информация о картах в табличном виде (метод 4.2.2.).
         1. Изображение карты.
         2. Название карты.
         3. Ширина карты.
         4. Высота карты.
         5. Максимальное количество игроков на карте.
         6. Описание (в выпадающем списке).
         7. Статус.
      3. Действия
         1. Удалить карту (метод 4.2.7.)
         2. Редактировать карту – перенаправляет на страницу 5.5.
         3. Добавить карту – перенаправляет на страницу 5.8.
2. **Требования к программному обеспечению:**
   1. Серверная часть
      1. ОС: Linux/Windows
      2. Docker
      3. Докер образы
      4. Go/gin-gonic 1.6.0
      5. redis 7.2.3-alpine
      6. nginx 1.19.2-alpine
      7. quay.io minio RELEASE.2022-10-15T19-57-03Z
      8. postgres 15
      9. django 5.0
      10. django rest framework 3.14
   2. Клиентская часть
      1. ОС: Windows/MacOS/Linux
      2. Веб-браузер: Safari 11.1+/Chrome 40+/Opera 27+/Firefox 44+
3. **Требования к аппаратному обеспечению:**
   1. Серверная часть
      1. Процессор минимум 2-ядерный с частотой от 2 ГГц.
      2. Оперативная память от 4 Гб.
      3. Место на жестком диске от 2 Гб.
   2. Клиентская часть
      1. Процессор с частотой от 1ГГц.
      2. Оперативная память от 512 Мб.