|  |  |
| --- | --- |
|  | **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 ИУ5 «Системы обработки информации и управления»**

Дисциплина «Разработка интернет-приложений»

Расчетно-пояснительная записка

Тема: «Полёты к Gateway»

Студент: Саргсян А.А.

Группа ИУ5-53Б

Преподаватель: Канев А.И.

2023г.

# **ВВЕДЕНИЕ**

Стремительный рост космической индустрии и повышенный интерес к исследованиям Луны и её окрестностей стали причиной создания международной обитаемой окололунной станции Lunar Gateway, возглавляемой NASA. Появление такой станции создаст необходимость в эффективной и надежной системе, способной обеспечить мониторинг и оптимизацию космических миссий с использованием модулей станции. В условиях увеличения сложности миссий и ресурсных ограничений, такая система становится ключевым инструментом для успешной реализации исследовательских и инженерных задач, а также для подготовки к будущим человеческим миссиям в область Луны.

Целью работы является реализация системы для образования и мониторинга космических миссий с использованием модулей станции, включающей в себя веб-сервис, веб-приложение, мобильное приложение и выделенный сервис расчета результата одобрения финансирования миссии.

Система предназначена для рядовых пользователей – представителей космических агентств и модераторов NASA. В системе предусмотрен ограниченный доступ к миссиям. Для получения доступа пользователю необходимо создать заявку на использование выбранных модулей станции. Система предоставляет автоматизированный способ создания, учета и ведения заявок. Модераторы могут взаимодействовать с базой данных и выполнять модерацию заявок. Также, администраторы имеют возможность редактировать существующие и создавать новые модули.

Нефункциональные требования к разрабатываемой системе:

* 1. Должна поддерживаться кроссплатформенность.
  2. Интерфейс системы и текст ошибок должны быть русифицируемы.

В ходе работы необходимо выполнить следующие задачи:

* 1. Разработать дизайн приложения.
  2. Создать базу данных в PostgreSQL.
  3. Создать веб-сервис на Go 1.21.5 с использованием gin-gonic 1.6.0
  4. Реализовать интерфейс гостя на технологии React.
  5. Развернуть веб-приложение React на Github Pages.
  6. Добавить авторизацию и аутентификацию в веб-сервис.
  7. Реализовать интерфейс пользователя в React.
  8. Реализовать интерфейс модератора в React.
  9. Создать мобильное приложение на React Native.
  10. Создать асинхронный сервис для расчета результата одобрения финансирования космической миссии на Django.
  11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.

# **БИЗНЕС-ПРОЦЕСС**

Существует много различных модулей, которые могут быть включены в миссии пользователями. Бывают ситуации, когда несколько агентств желают осуществить полёт с участием тех или иных модулей. Для того, чтобы другие пользователи были осведомлены, какие модули уже участвуют в миссии, пользователи могут оставлять заявки на использование модулей в определённой миссии. Каждая заявка вначале создается как черновик, в который можно добавлять и из которого можно удалять модули. В одной заявке можно указать сразу несколько модулей.

Когда пользователь определится с модулями, он формирует миссию, и она отправляется на модерирование. После этого её уже нельзя редактировать. Можно также посмотреть историю своих миссий. К выбору модулей для миссий нужно относиться с внимательностью, ведь каждый лишний модуль – это огромные затраты сил, денег и времени. Для этого менеджеры отклоняют и принимают заявки таким образом, чтобы модули в разных миссиях не повторялись. На случай, если пользователь совершил ошибку или передумал начинать миссию, также присутствует возможность отменить заявку.

Модуль представляет собой сложную конструктивно законченную часть орбитальной станции, предназначенную для выполнения определённых задач, которые перечисляются в описании. Также модули бывают определенной массы и размеров. Чтобы было примерно понятно, что из себя представляют модули вживую, они имеют собственное изображение. Если модуль по каким-либо причинам будет выведен из строя или его эксплуатация будет временно или постоянно невозможна, то на этот случай администратор может временно или навсегда скрыть модуль. В случае изменения характеристик модуля в процессе его разработки и эксплуатации, модератор может отредактировать модуль, обновить фотографию или исправить какие-либо ошибки. Она может отредактировать какую-либо карту, обновить её, или исправить какие-то ошибки. Также модератор может создать новый модуль. Функции пользователей с разными ролями описаны на диаграммах прецедентов (рис. 1).

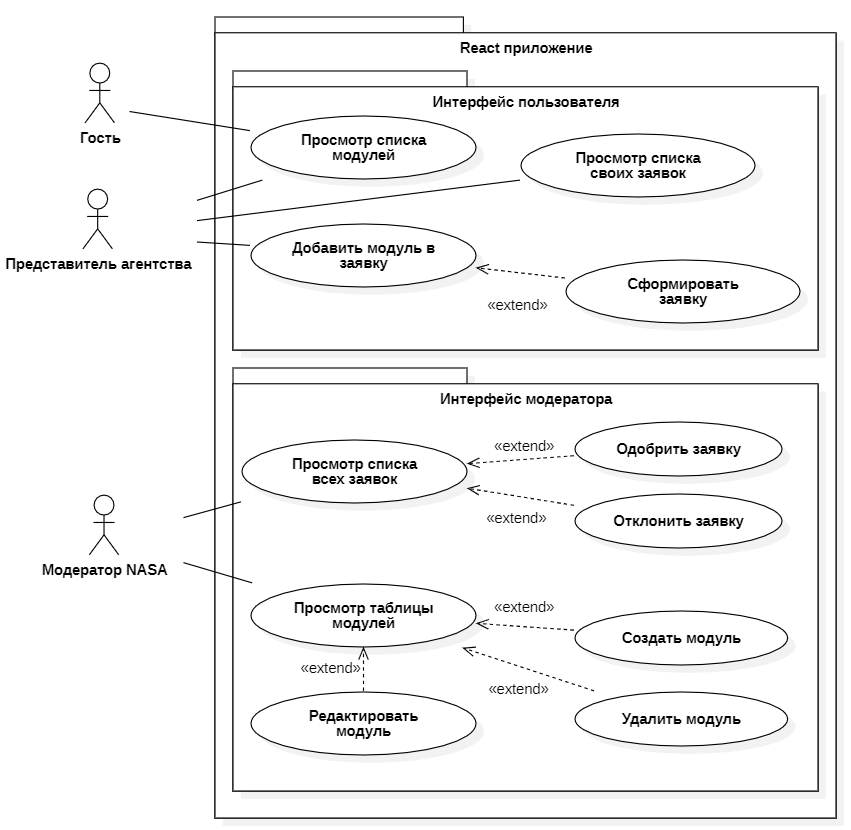


Рисунок 1 - Диаграмма прецедентов

Гостям доступен просмотр модулей. Зарегистрированные гости – рядовые пользователи – представители космических агентств. Они могут добавлять модули в заявку, просматривать список своих заявок и формировать текущую заявку. Заявки обрабатываются модераторами. В результате обработки заявки её либо одобряют, либо отклоняют. Модератору также доступны: просмотр всех модулей, редактирование, создание и удаление модулей, а также просмотр списка всех модулей в табличном виде. Процесс оформления заявки отражен на диаграмме деятельности (рис. 2).

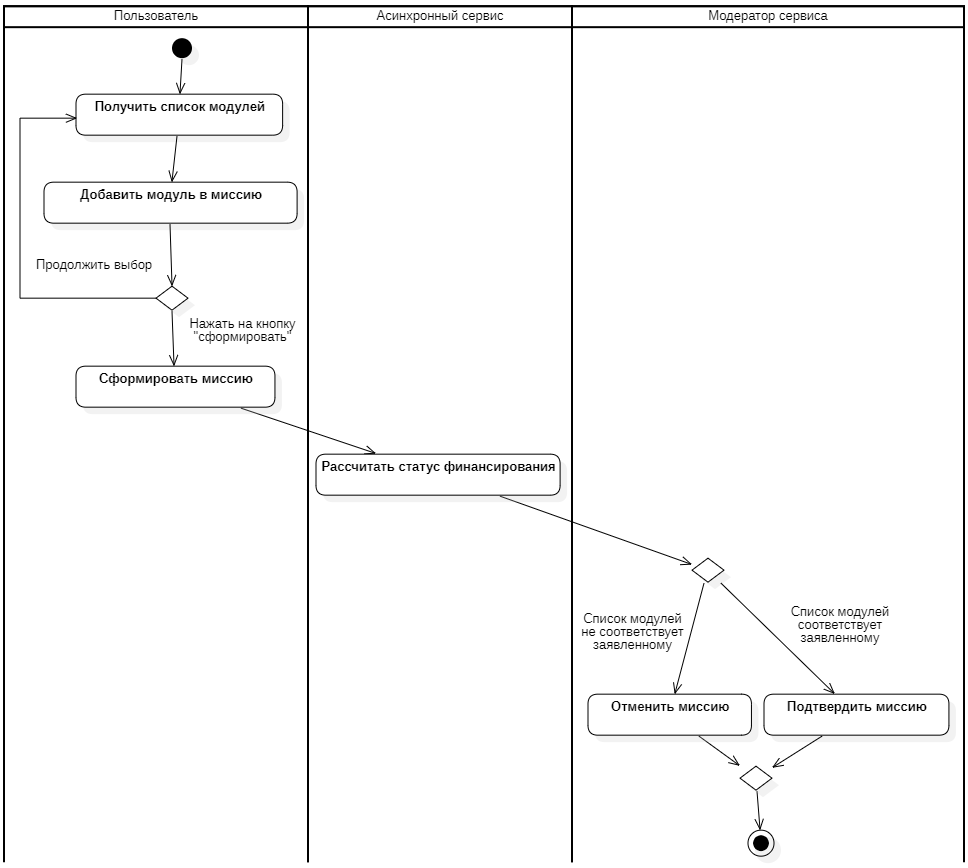


Рисунок 2 - Диаграмма деятельности

Пользователь выбирает модули, затем формирует на основе выбранных модулей заявку. Эту заявку затем обрабатывает асинхронный сервис, а затем и модератор. В случае одобрения для заявки запускается расчёт результата одобрения финансирования миссии в асинхронном сервисе. Возможные состояния заявки отражены на диаграмме состояний (рис. 3).

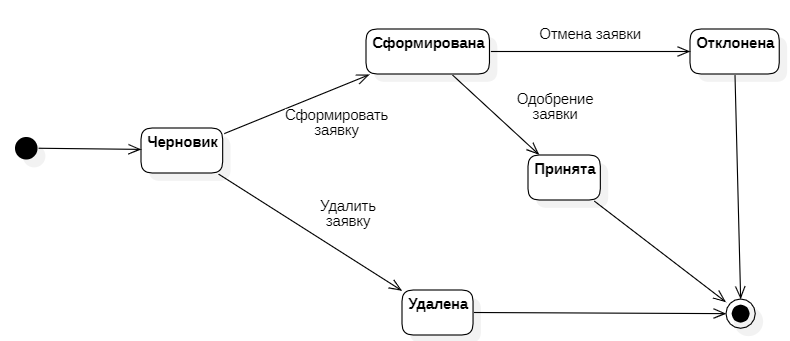


Рисунок 3 - Диаграмма состояний заявки

При выборе первого модуля формируется черновик. Последующие выбранные модули добавляются в этот черновик. Пользователь затем формирует заявку, удаляет её или выходит из приложения. Сформированную заявку обрабатывает модератор. Он может одобрить или отклонить её.

# **АРХИТЕКТУРА**

Архитектура системы отображена на диаграмме развертывания (рис. 4). Бекенд, фронтенд и асинхронный сервис разворачиваются отдельно друг от друга. Часть бекенда разворачивается в кластере докер контейнеров, это позволяет быстро и удобно получить доступ к серверам на любом устройстве, поддерживающем докер. Кластер описан в docker-compose.yml файле.

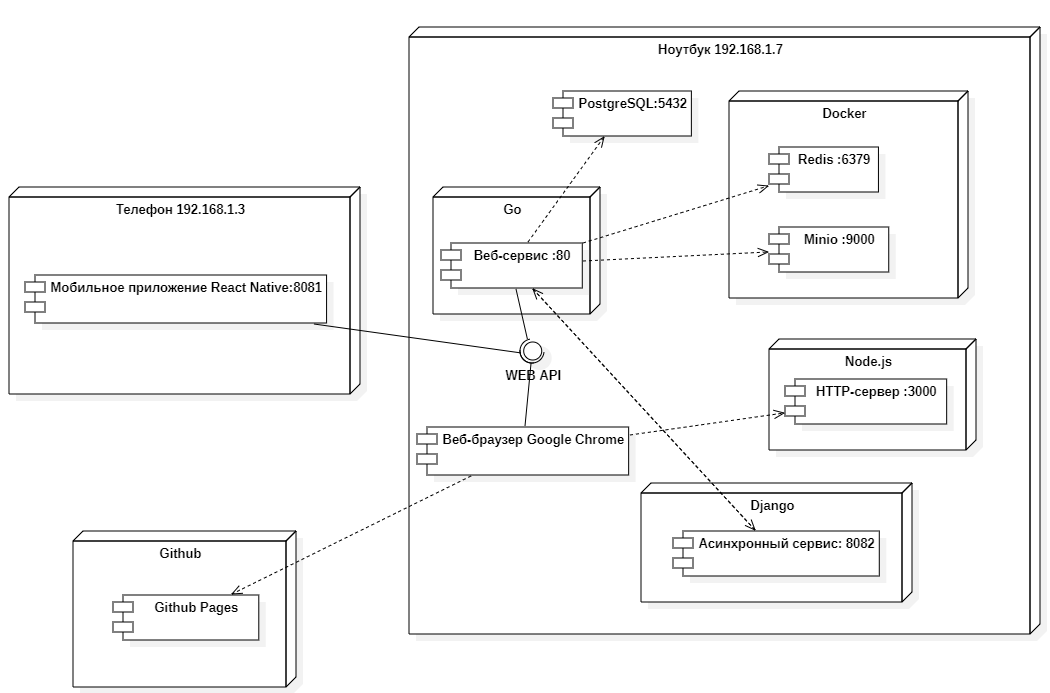


Рисунок 4 - Диаграмма развертывания

Мобильное и браузерное приложения обращаются к веб-сервису на базе языке Golang 1.20 через REST Web-API. Использование Golang обосновано его проверенной эффективностью, признанием в индустрии и современностью языка.

Данные хранятся в СУБД PostgreSQL, их структура отражена на ER диаграмме (рис. 5). СУБД PostgreSQL является одним из стандартов индустрии, поэтому было решено использовать её. Структура данных довольна проста. Модель модулей представляет собой набор полей, необходимых исключительно для бизнес-логики. Для хранения в одной заявке нескольких карт используется промежуточная таблица Полёты, реализующая связь М-М.

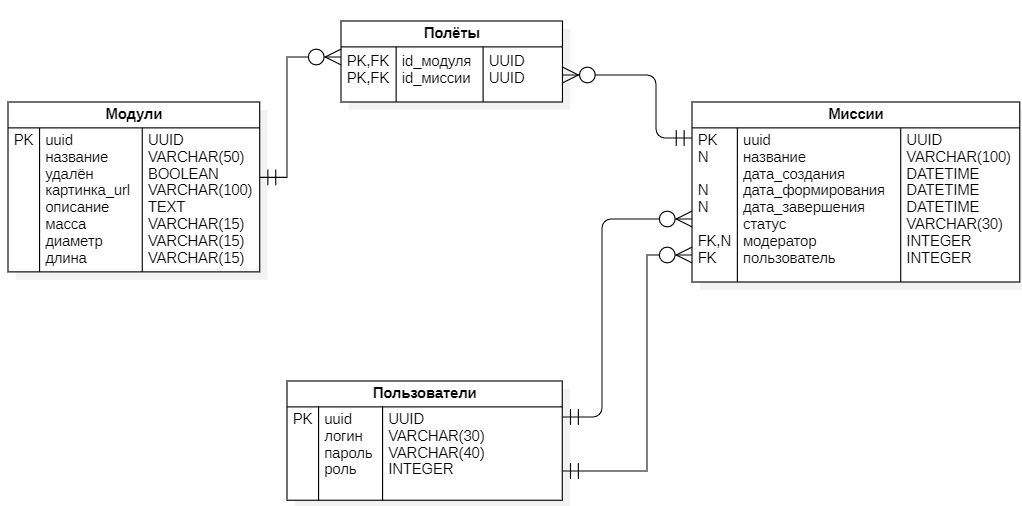


Рисунок 5 - ER диаграмма

Устройство бекенда приложения изображено на диаграмме классов бекенда (рис 6.). Модели имеют связи с таблицами в базе данных. Также некоторые модели имеют связи с внешними сервисами. В частности, модули имеют связь с сервером статических файлов, т.к. в картах хранится ссылка на их изображение, хранимое на сервере статических файлов.

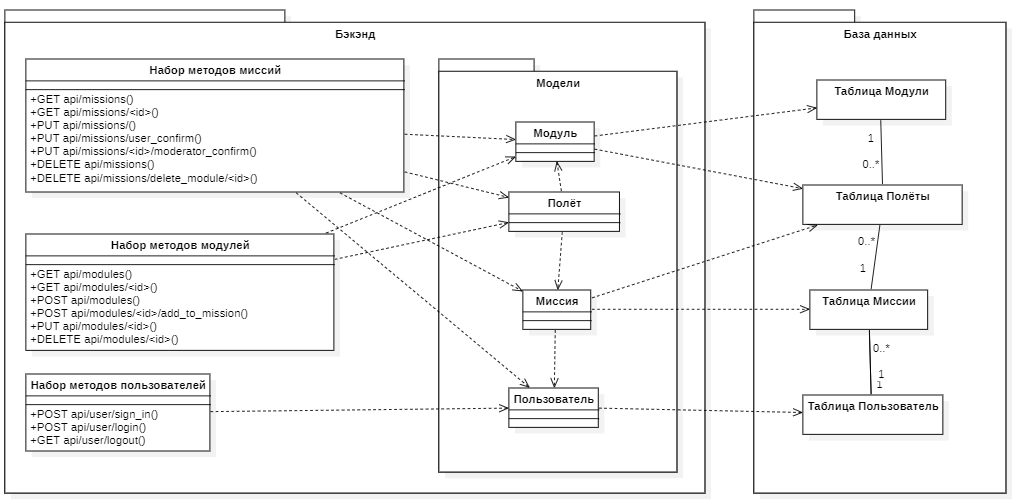


Рисунок 6 *-* Диаграмма классов бекенда

Связь фронтенда и бекенда отражена на диаграмме классов фронтенда (рис. 7). Ключевые страницы имеют связь с API аутентификации, т.к. доступ к ним осуществляется только для авторизированных пользователей с определенными правами (ролями).

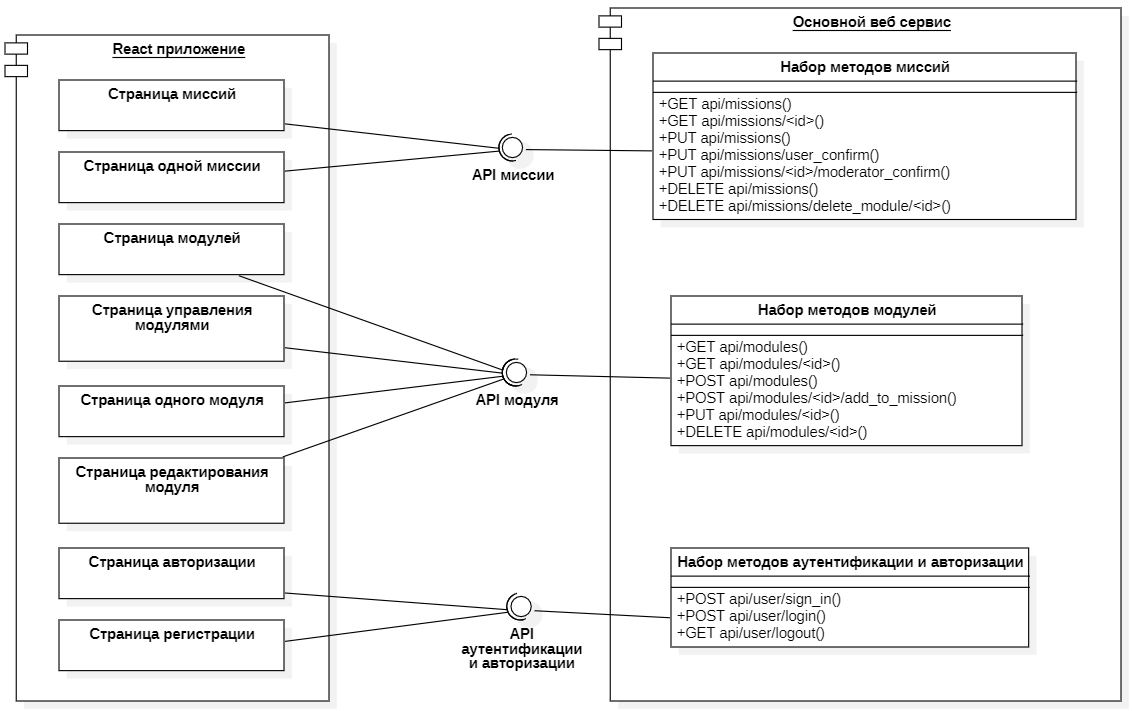


Рисунок 7 *-* Диаграмма классов фронтенда

# **АЛГОРИТМЫ**

Алгоритм работы системы отображен на диаграмме последовательности (рис. 8). В основе системы лежит веб-сервис, реализующий внутри себя всю бизнес-логику. Он предоставляет доступ к методам из следующих доменов: модули, миссии, пользователи и авторизация. Методы следуют правилам REST API.

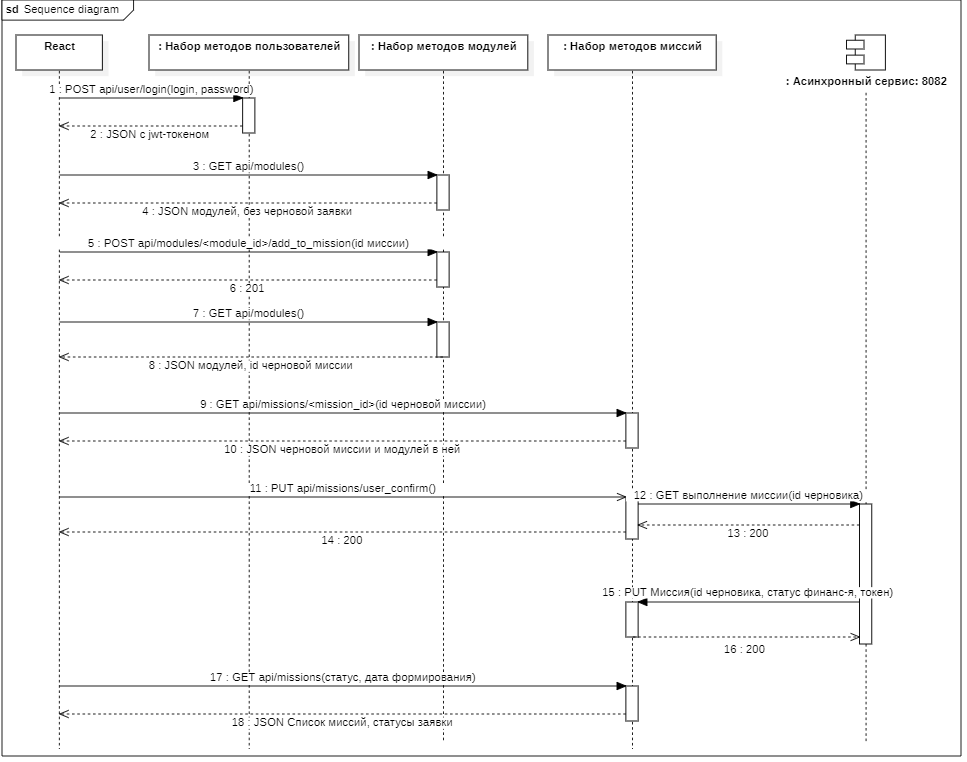


Рисунок 8 - Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация пользователя. Для этого он отправляет через графический интерфейс запрос, передавая в нем логин и пароль. Если аккаунт с такими данными существует, то пользователь получает JWT в ответном запросе. Если же такого аккаунта не существует, или пароль введен неверно, пользователь получит ошибку. В таком случае ему надо либо пройти регистрацию, либо ввести пароль верно. Затем графический интерфейс пользователя запрашивает у веб-сервиса список модулей, которые возвращаются в JSON формате. Пользователь выбирает модуль, который хочет добавить в миссию, и, нажимая на кнопку «Добавить» в графическом интерфейсе, отправляет запрос на добавление модуля в свою черновую миссию. Этот процесс может продолжаться несколько раз.

Когда пользователь определится с выбором, он нажимает на кнопку «Сформировать» в графическом интерфейсе. После этого приложение запрашивает id черновой заявки пользователя и затем отправляет запрос на формирование этой заявки. В этот момент основной веб-сервис выполняет асинхронный запрос к другому сервису на то, чтобы он рассчитал статус одобрения финансирования миссии и вернул его в основной сервис. Когда заявка будет одобрена, пользователь сможет увидеть через некоторое время, одобрено ли финансирование его миссии.

Процесс рассмотрения миссий происходит также через графический интерфейс. Модераторы могут просматривать списки всех миссий и, нажимая на соответствующие кнопки, отправлять запросы на одобрение или отклонение миссий в основной веб-сервис. В эти запросы также можно включить фильтры по имени создателя миссии и диапазону дат, в котором должны были быть созданы миссии. Также, через графический интерфейс модераторы могут управлять непосредственно модулями. Им доступны такие функции, как создание и редактирование модулей, просмотр списка модулей и удаление их. Для каждой из этих функция присутствует свой метод, отправляемый на основной веб-сервис.

# **ОПИСАНИЕ ИНТЕРФЕЙСА**

Главное меню приложения включает пункты, которые доступны в зависимости от роли пользователя (рис. 9, 10).

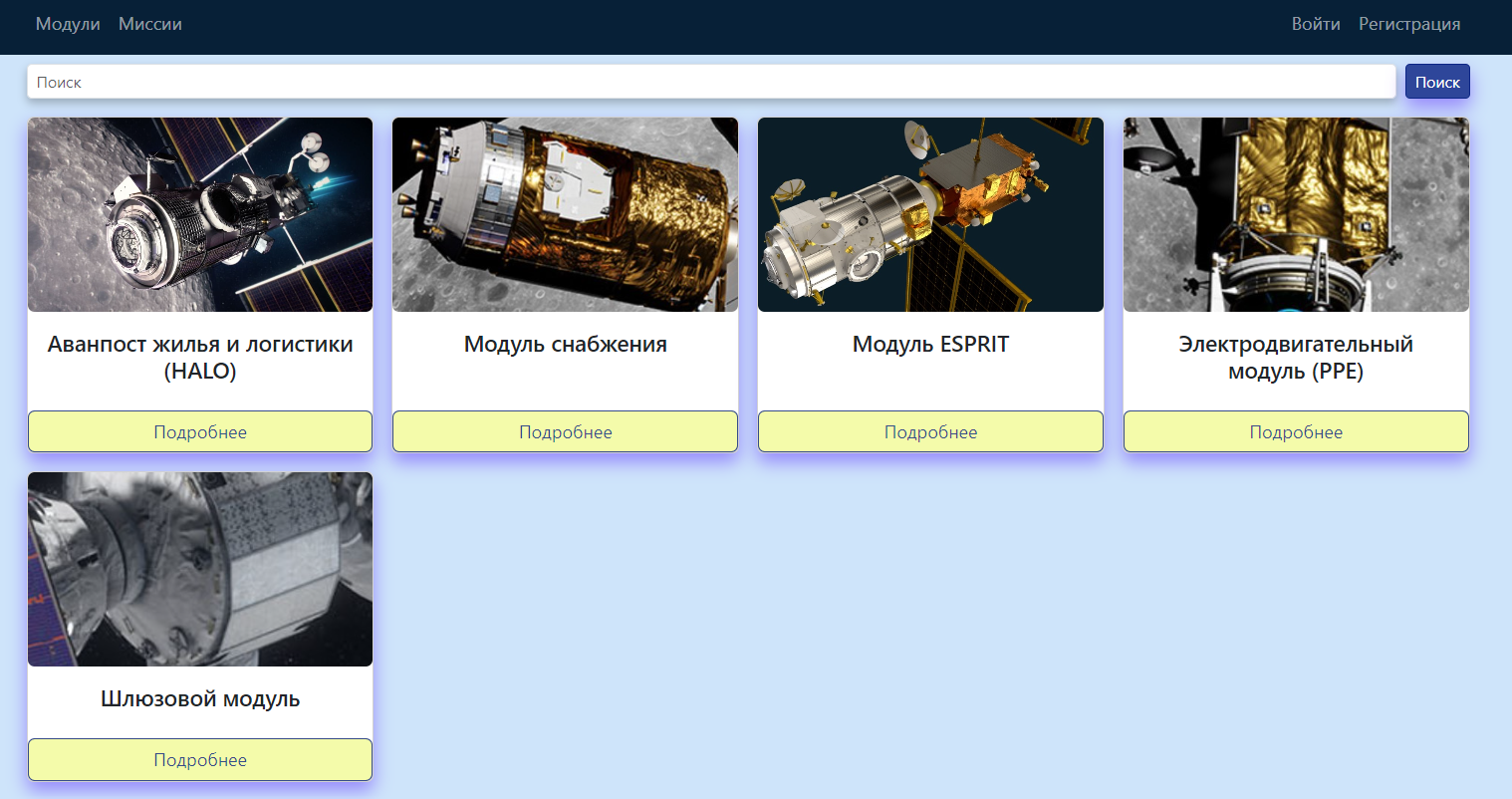
****

Рисунок 9 - Главное меню (неавторизированный пользователь)

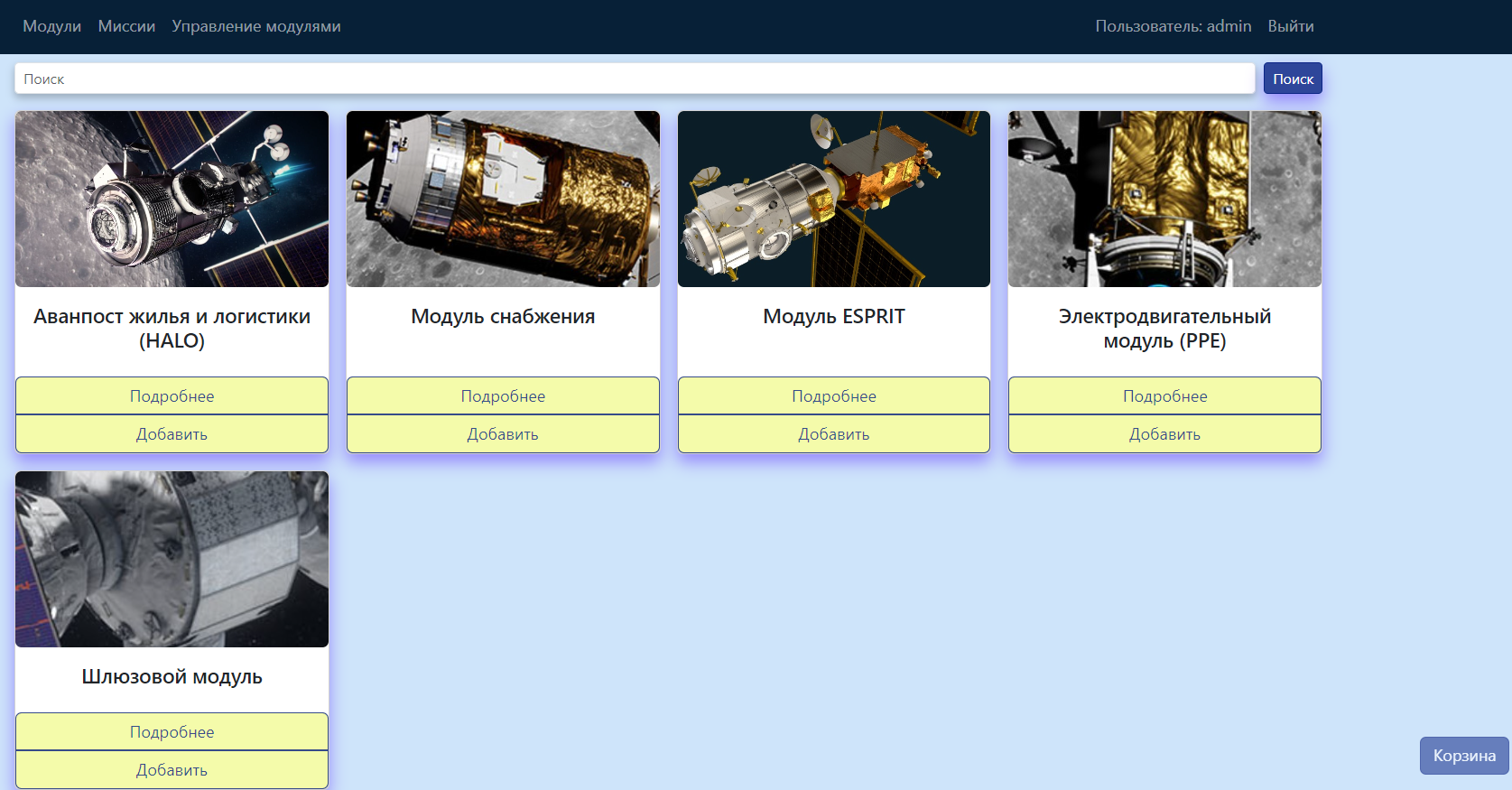


Рисунок 10 - Главное меню (модератор)

Первоначальная страница для всех пользователей и гостей. В зависимости от типа пользователя её содержимое меняется. Для гостей, например, там отображаются только кнопки «Войти» и «Регистрация», а для пользователя – «Модули», «Миссии», «Корзина» и «Выход».

На странице с формой авторизации (рис. 11) отображается форма, через которую гость входит в свой аккаунт. При успешном вводе логина и пароля гость получает JWT, который сохраняется в local storage и используется при отправлении запросов.

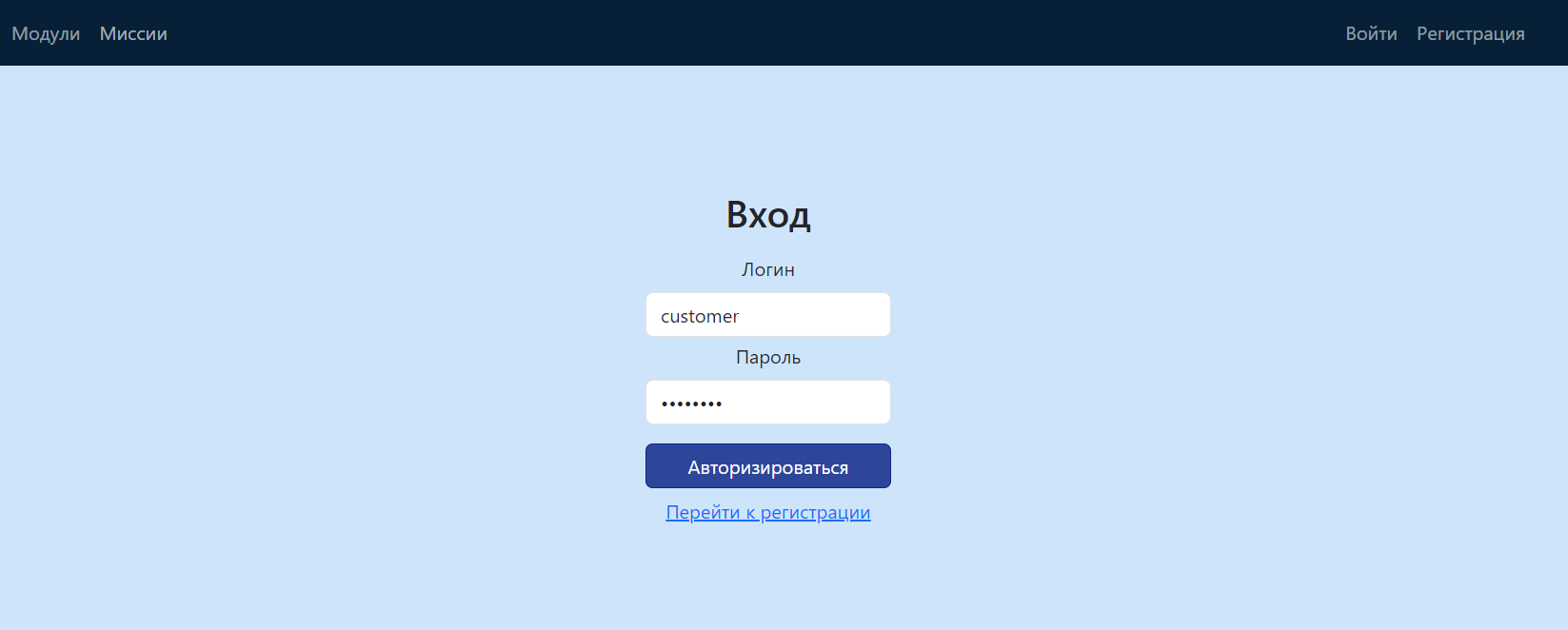
****

Рисунок 11 - Страница авторизации

На странице с формой регистрации (рис. 12) гости могут завести аккаунт. Для этого нужно указать логин, и пароль. Если введенный логин уже занят система попросит пользователя сменить его.

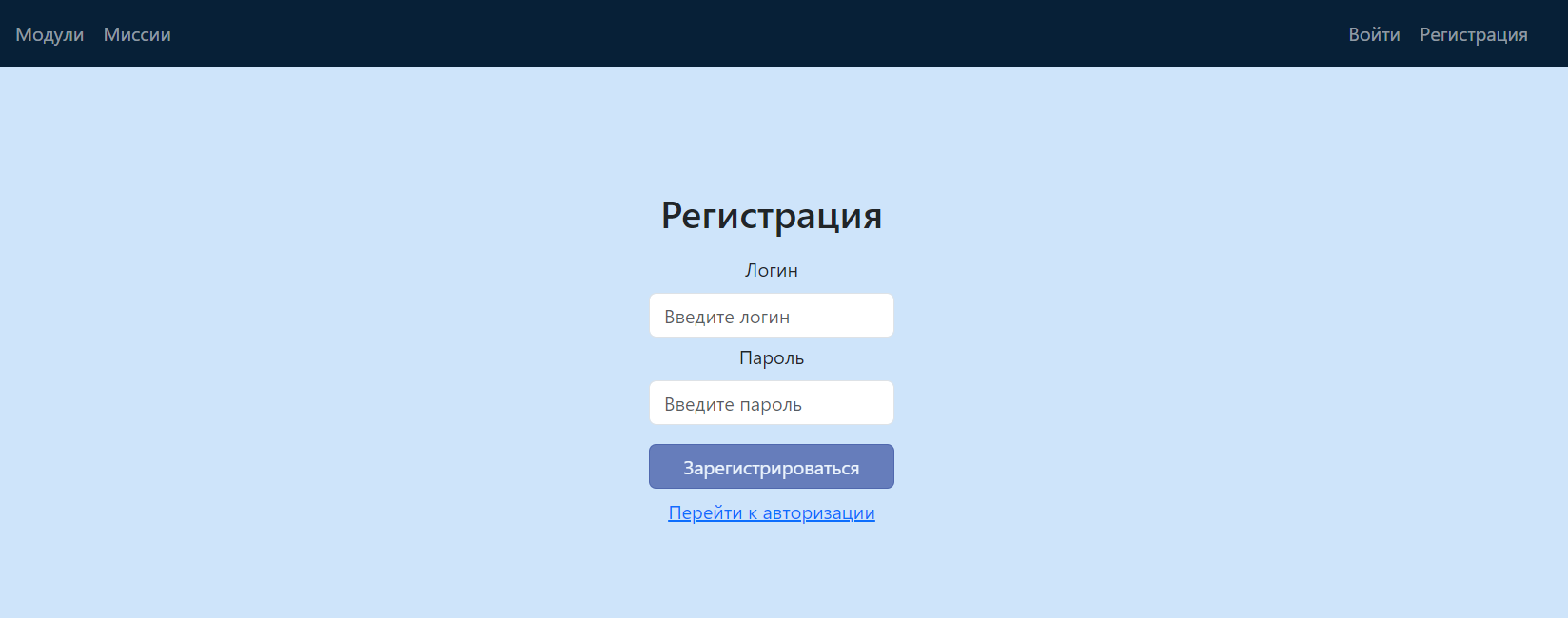
****

Рисунок 12 - Страница регистрации

На странице со списком модулей (рис. 13) отображается список модулей в виде карточек. У каждой карточки есть кнопка «Подробнее», переносящая пользователя на страницу с подробной информацией о модуле, и кнопка «Добавить» для добавления модуля в корзину. Сверху находится фильтр модулей, там можно ввести фрагмент названия модуля, чтобы отображались только отфильтрованные модули.

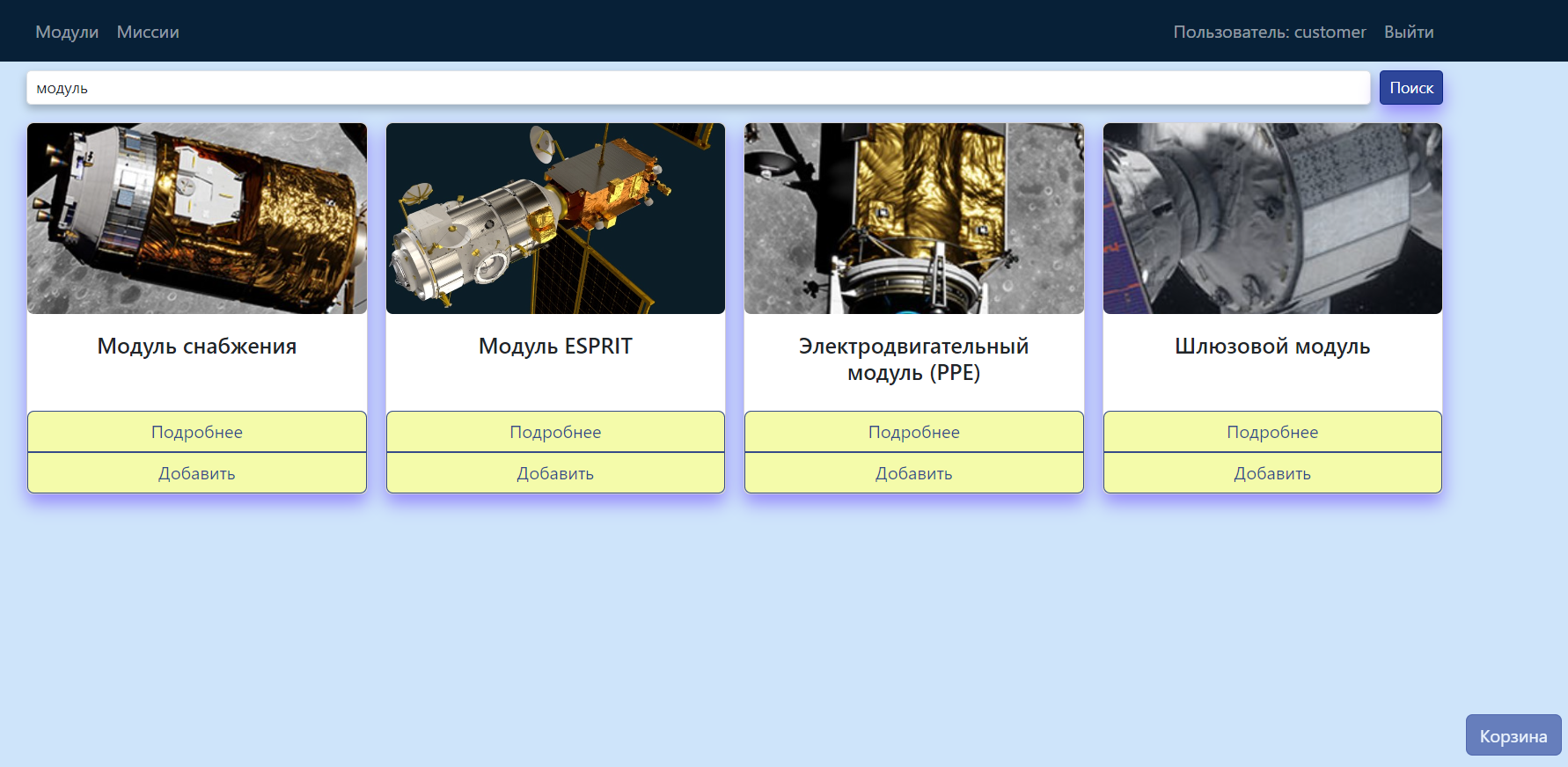
****

Рисунок 13 - Страница с отфильтрованным списком модулей

На странице с подробным описанием модуля (рис. 14) отображается подробная информация о модуле: название, текстовое описание, масса, длина и диаметр.

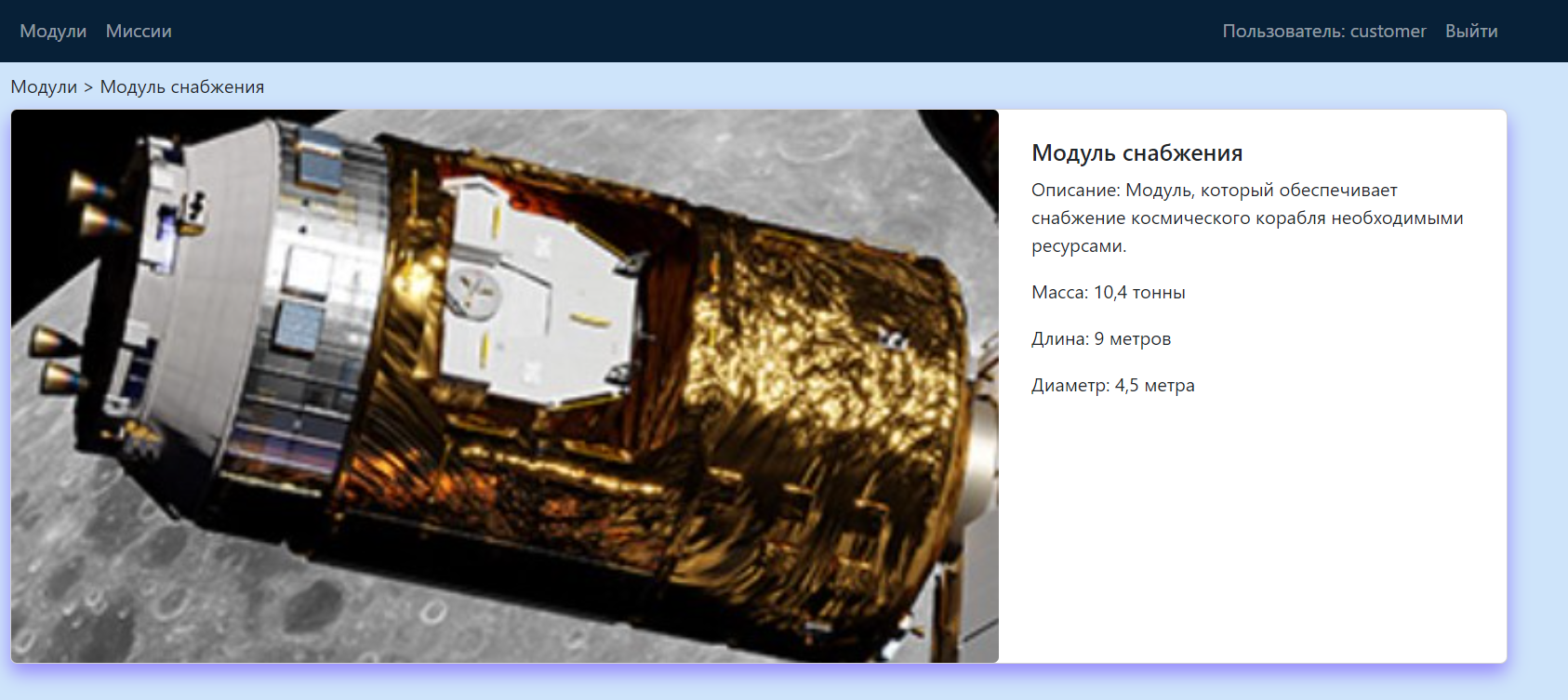


Рисунок 14 - Страница с подробным описанием модуля

На странице со списком миссий (рис. 15, 16) отображается список миссий. В зависимости от типа пользователя этот список будет функционально отличаться. Так, для пользователей отображается список созданных ими заявок: статус, статус финансирования, дата создания, дата формирования, дата завершения, название и кнопка «Подробнее» для открытия заявки. Также, можно отфильтровать миссии по статусу и по дате формирования миссии.

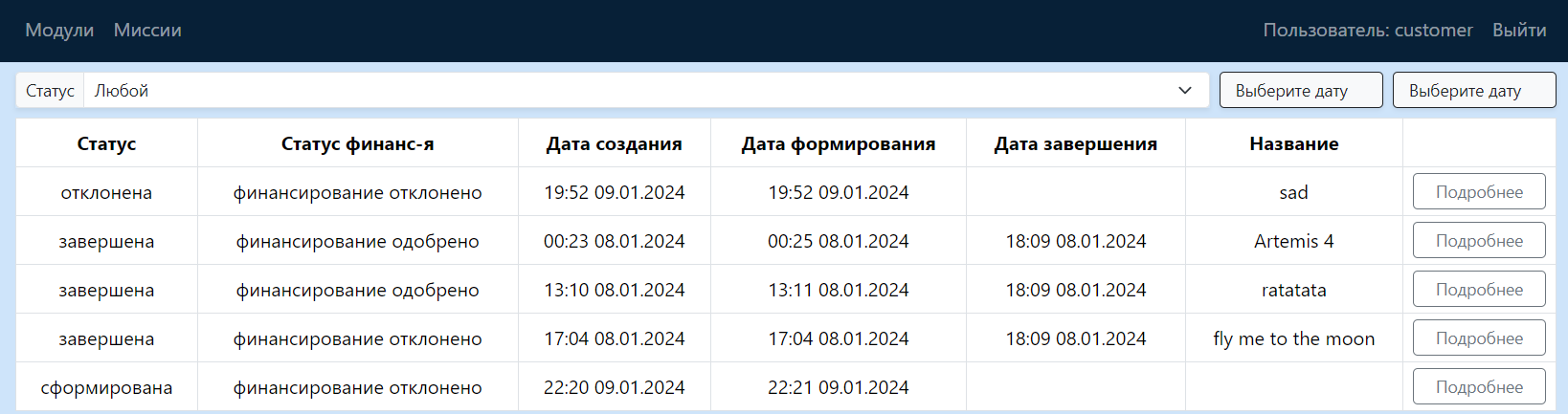


Рисунок 15 - Страница со списком заявок (пользователь)

Для модератора функция этой страницы шире. Для них отображается список всех заявок всех пользователей с более подробной информацией: создатель заявки, дата создания, дата формирования, дата завершения, название, кнопка «Подробнее» для открытия заявки и для заявок со статусом «сформирована» кнопки «Подтвердить» и «Отклонить» для установления статуса миссии.

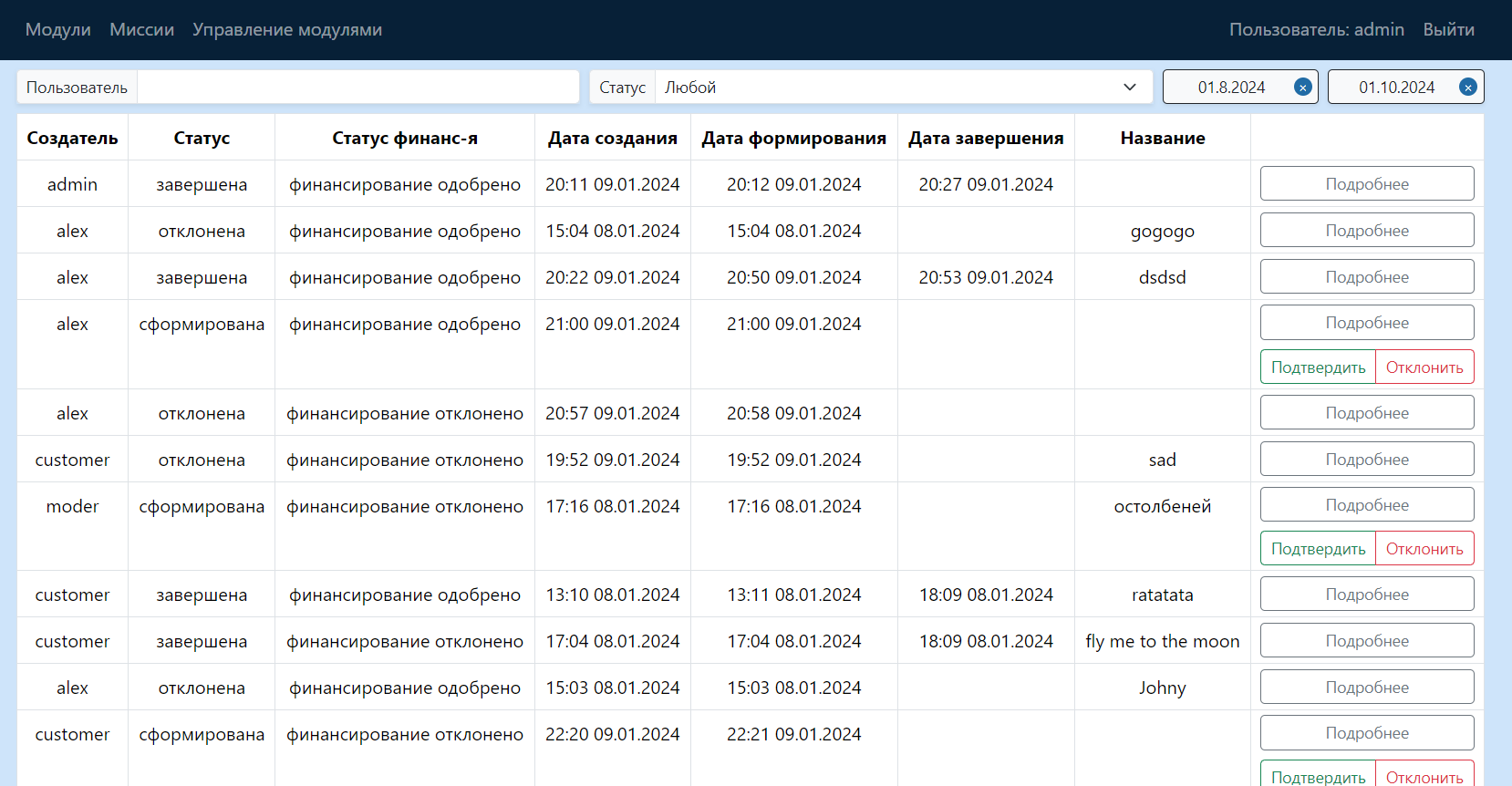


Рисунок 16 - Страница со списком заявок (модератор)

На странице с подробным описанием миссии (рис. 17) отображается подробная информация о миссии, список выбранных модулей в виде карточек, а также все поля миссии. Для сформированных миссий есть кнопки «Подтвердить» и «Отменить» для соответствующих действий с заявкой.

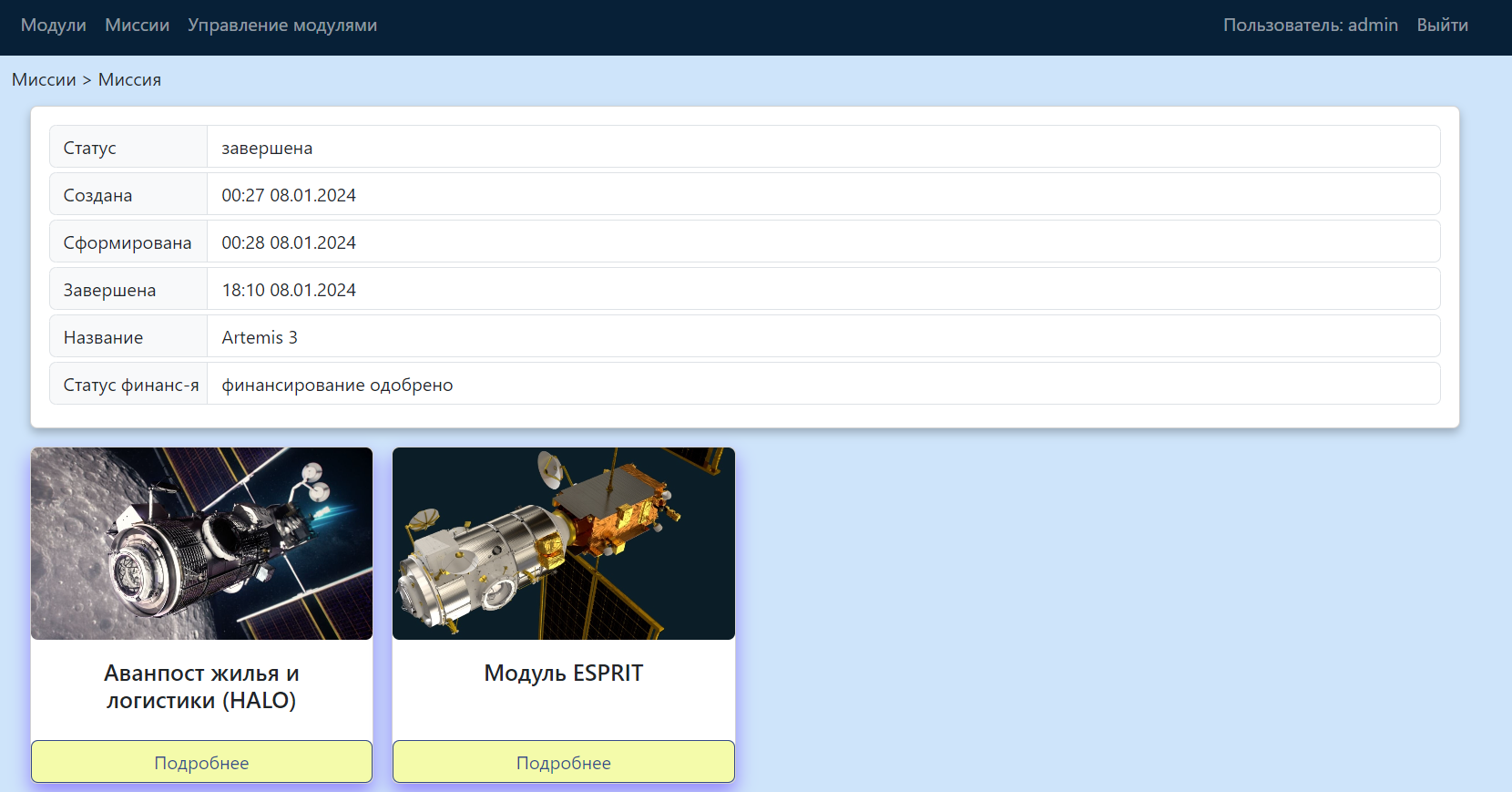


Рисунок 17 - Страница с подробным описанием заявки

На странице с таблицей модулей (рис. 18) модератор может в компактном и удобном формате просмотреть список всех модулей, существующих в системе. Отображаются следующие поля: изображение и название. Также можно отредактировать или удалить существующий модуль нажатием соответствующих кнопок (рис. 19), или перейти на страницу создания нового (рис. 20). Можно отфильтровать модули по названию.

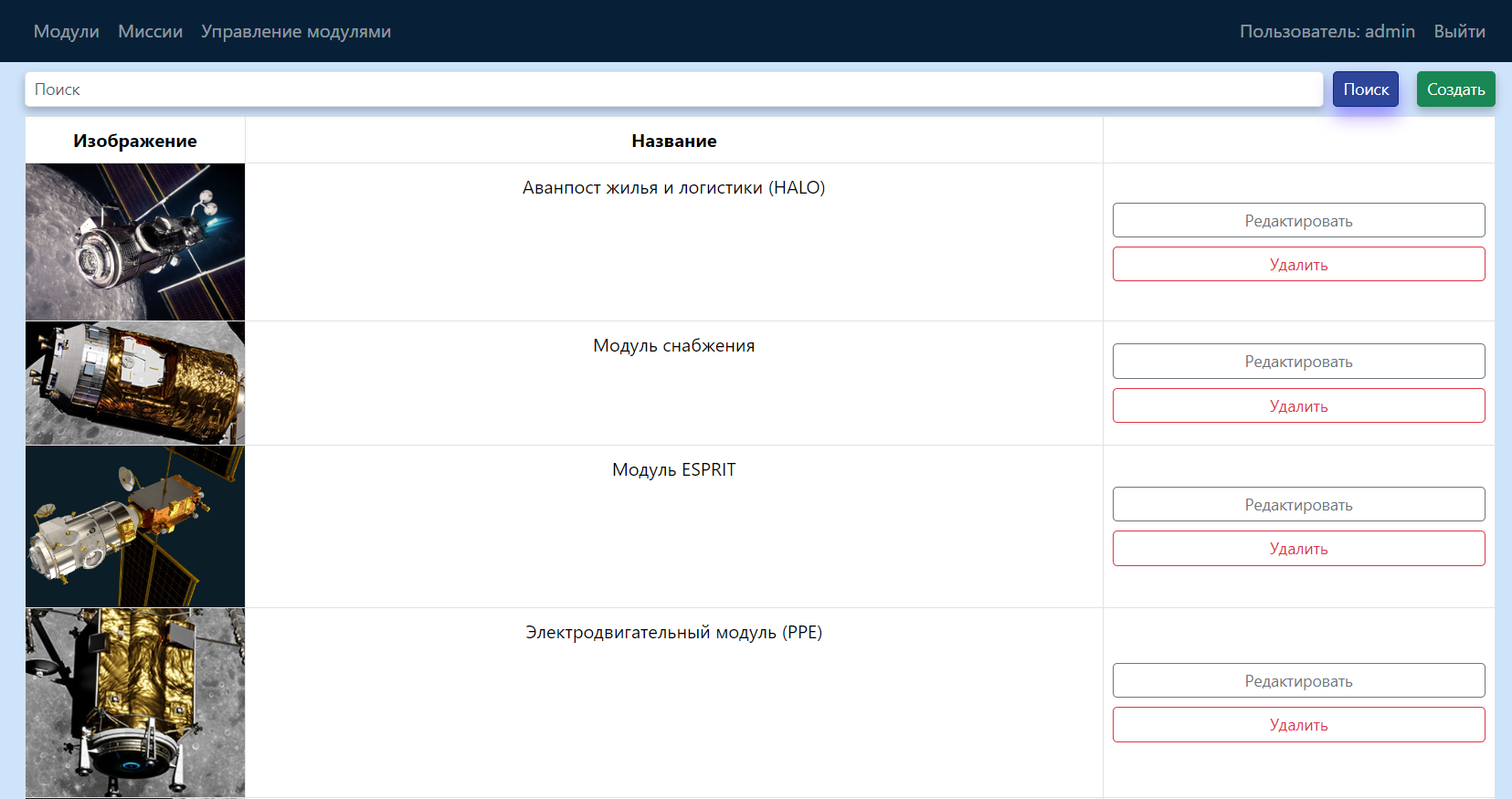


Рисунок 18 - Страница с таблицей модулей



Рисунок 19 - Страница редактирования модуля

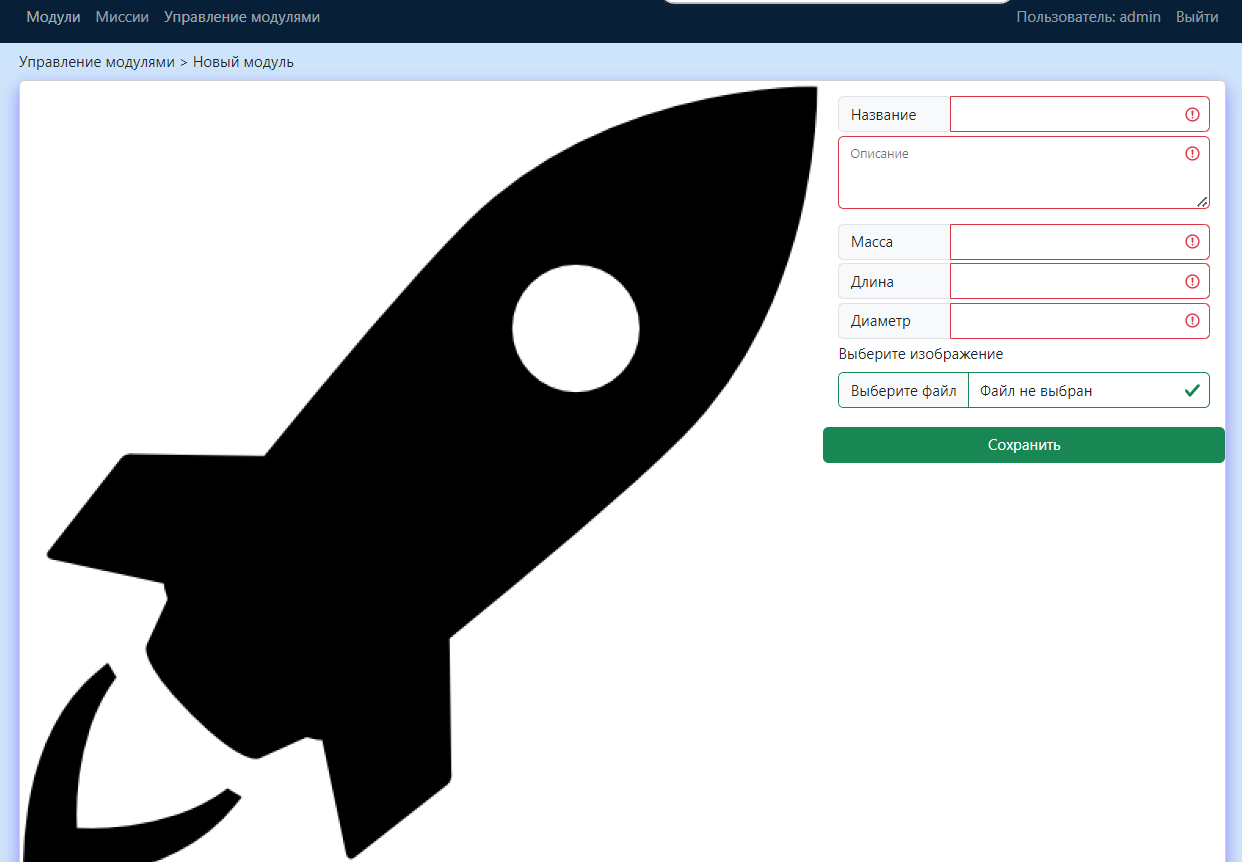


Рисунок 20 - Страница создания модуля

На странице редактирования модуля (рис. 19) модератор может отредактировать существующий модуль. Для редактирования доступны все поля: название, изображение, описание, масса, длина и диаметр.

На странице создания модуля (рис. 20) модератор может создать новый модуль. Для заполнения доступны все поля: название, изображение, описание, масса, длина и диаметр.

# **ЗАКЛЮЧЕНИЕ**

В ходе работы были достигнуты следующие результаты:

1. Был разработан дизайн приложения с помощью набора стилей CSS и HTML тегов.
2. База данных была создана.
3. Был создан веб-сервис на GO с REST API.
4. Разработан интерфейс гостя с использованием технологии React Framework и подключен к веб-сервису.
5. Приложение интерфейса было развернуто на сервисе Github Pages по ссылке https://blessed011.github.io/DevIntApp-frontend.
6. В веб-сервис добавлена авторизация через JWT, а методы задокументированы через Swagger.
7. Реализован интерфейс пользователя. Доступ к нему имеют только авторизированные пользователи.
8. Выделенный сервис был разработан на Django.
9. Реализован интерфейс модератора для подтверждения новых миссий и редактирования/добавления модулей.
10. Было реализовано мобильное приложение на React Native, повторяющее интерфейс веб-приложения на React.
11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор UML диаграмм.
12. Исходный код проекта доступен в GitHub https://github.com/Blessed011?tab=repositories.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* 1. Полное практическое руководство по Docker [Электронный ресурс] // Habr. URL: https://habr.com/ru/articles/310460/ (дата обращения: 17.09.2023).
  2. Руководство по Gin Web Framework [Электронный ресурс] // GoLang. URL: https://github.com/gin-gonic/gin (дата обращения: 20.09.2023).
  3. Руководство по React [Электронный ресурс] // Metanit. URL: https://metanit.com/web/react/ (дата обращения: 09.10.2023).
  4. Quick Start – React [Электронный ресурс] // React. URL: https://react.dev/learn (дата обращения: 10.10.2023).
  5. Руководство по React Bootstrap [Электронный ресурс] // React Bootstrap. URL: https://react-bootstrap.netlify.app/docs/getting-started/introduction (дата обращения: 13.12.2023).
  6. Руководство по PostgreSQL [Электронный ресурс] // Metanit. URL: https://metanit.com/sql/postgresql/ (дата обращения: 01.10.2023).
  7. Руководство по Gorm [Электронный курс] //Gorm URL: https://gorm.io/docs/ (дата обращения: 15.10.2023).

**Приложение. Техническое задание**

|  |  |
| --- | --- |
| 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

# **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 модули.

# **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 авторизации, аутентификации и Swagger в веб-сервис;
  6. Добавление интерфейса создателя заявки;
  7. Добавление нативного приложения React Native, Redux Toolkit;
  8. Разработка асинхронного сервиса для расчета результата одобрения финансирования миссий на Django;
  9. Разработка интерфейса модератора на React;
  10. Разработка и документирование ТЗ, РПЗ и диаграммы.
  11. Развёртка веб-приложения React на GitHub Pages.

# **Методы веб-сервиса**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | | **Метод** | **URL** | **Описание** | **Входные данные** | **Выходные данные** |
| * 1. **Услуга – модуль станции** | | | | | | |
|  | GET | | api/modules/ | Получить список модулей, удовлетворяющих заданным критериям, и черновую заявку пользователя | JWT token  {  Name: string,  } | {  "draft\_mission":{  "uuid": STRING,  "module\_count": INTEGER  },  "modules": [ {  "uuid": STRING,  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR  },  ...  ]  } |
|  | GET | | api/modules/<module\_id> | Получить модуль по уникальному ID | module\_id: INTEGER | {  "uuid": STRING1,  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR  } |
|  | POST | | api/modules/ | Добавление модуля.  Доступно только авторизированным модераторам | JWT token  {  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR  } | {  "uuid": STRING,  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR  } |
|  | POST | | api/modules/<module\_id>/add\_to\_mission/ | Добавляет в черновую миссию пользователя модуль по ID, возвращает число получателей в черновой миссии.  Доступно только авторизированным пользователям | JWT token  module\_id: INTEGER | module\_count: INTEGER |
|  | PUT | | api/modules/<module\_id> | Обновить информацию о модуле.  Доступно только авторизированным модераторам | JWT token  {  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR  } | {  "uuid": STRING,  "name": VARCHAR,  "description": TEXT,  "mass": VARCHAR,  "length": VARCHAR,  "diameter": VARCHAR,  "image\_url": VARCHAR,  } |
|  | DELETE | | api/modules /<module\_id> | Меняет статус модуля на “Удален”.  Доступно только авторизированным модераторам | JWT token | Отсутствует |
| * 1. **Заявка – космическая миссия** | | | | | | |
|  | GET | | api/missions/ | Получить список космических миссий с фильтрацией по дате формирования и статусу  Доступно только авторизированным пользователям | JWT token  {  “status”:  STRING,  “formation\_date\_start”:  DateTime,  “formation\_completion\_date”:  DateTime  } | {  "missions":  [  {  "uuid":STRING,  "name": VARCHAR,  "status": STRING,  "creation\_date": TIMESTAMP,  "formation\_date": TIMESTAMP,  "completion\_date":  TIMESTAMP,  "moderator":  VARCHAR,  "customer": VARCHAR,  “funding\_status”:  VARCHAR  },  …  ]  } |
|  | GET | | api/missions/<mission\_id> | Получить информацию о миссии и её модулях.  Доступно только авторизированным пользователям | JWT token | {  "missions":  {  "uuid":STRING,  "name": VARCHAR,  "status": STRING,  "creation\_date": TIMESTAMP,  "formation\_date": TIMESTAMP,  "completion\_date":  TIMESTAMP,  "moderator":  VARCHAR,  "customer": VARCHAR,  “funding\_status”:VARCHAR  }  "modules":  [  {…},  …  ]  } |
|  | PUT | | api/missions/<mission\_id> | Обновить информацию о миссии.  Доступно только авторизированным пользователям | JWT token  {  “name”:  VARCHAR,  “date\_start\_mission”:  DATE,  “description”:  TEXT  } | {  "uuid": STRING,  "name": VARCHAR,  "status": VARCHAR,  "creation\_date": TIMESTAMP,  "formation\_date": TIMESTAMP,  "completion\_date": TIMESTAMP,  "customer": VARCHAR,  "moderator": VARCHAR,  “funding\_status”: VARCHAR  } |
|  | PUT | | api/missions/<mission\_id>/user\_confirm | Меняет статус миссии с “Черновик” на “Сформирован”.  Доступно только авторизированным пользователям | JWT token | Отсутствует |
|  | PUT | | api/missions/<mission\_id>/moderator\_confirm | Меняет статус миссии с “Сформирован” на “Завершён” или “Отклонён”.  Доступно только авторизированным модераторам. | JWT token  {  “confirm”:  BOOLEAN  } | Отсутствует |
|  | DELETE | | api/missions/<mission\_id> | Меняет статус миссии на “Удалён”.  Доступно только авторизированным модераторам | JWT token | Отсутствует |
|  | DELETE | | api/missions/delete\_module/<module\_id> | Удаляет модуль из миссии.  Доступно только авторизованным пользователям | JWT token | Отсутствует |
| * 1. **Пользователь** | | | | | | |
|  | POST | | api/user/sign\_up | Регистрация | {  "login": string,  "password": string,  } |  |
|  | POST | | api/user/login | Авторизация | {  "login": string,  "password": string,  } | JWT token |
|  | GET | | api/user/logout | Выход из аккаунта | JWT token |  |
| * 1. **Выделенный сервис** | | | | | | |
|  | POST | | api/funding | Рассчитывает новый статус одобрения финансирования | {  "mission\_id": STRING  } | {  “funding\_status”: BOOLEAN,  “token”: VARCHAR  } |

1. **Функциональные требования:**
   1. Главное меню.
      1. Доступна всем пользователям.
      2. Действия
         1. Регистрация – переход на страницу 5.2. Для гостей.
         2. Войти – переход на страницу 5.3. Для гостей.
         3. Список модулей – переход на страницу 5.4. Для всех.
         4. Список выбранных модулей для миссии – переход на страницу 5.7. Для авторизованных пользователей.
         5. Список миссий – переход на страницу 5.6. Для авторизованных пользователей.
         6. Добавить модуль – переход на страницу 5.8. Для администраторов.
         7. Таблица модулей – переход на страницу 5.9. Для администраторов.
   2. Страница регистрации.
      1. Доступна гостям.
      2. Действия
         1. Регистрация – отправляет запрос (метод 4.3.1), в котором передаются введенные данные.
         2. Войти – перенаправление на страницу 5.3.
   3. Страница авторизации.
      1. Доступна гостям.
      2. Действия
         1. Войти – отправляет запрос (метод 4.3.2).
         2. Регистрация – перенаправление на страницу 5.2.
   4. Страница со списком модулей.
      1. Доступна всем пользователям.
      2. Информация о каждом модуле в виде карточки (метод 4.1.1)
         1. Изображение модуля.
         2. Название модуля.
      3. Действия
         1. Фильтрация модулей по названию (метод 4.1.1).
         2. “Выбранные модули” – перенаправление на страницу миссии 5.7.
         3. Добавить модуль в миссию (метод 4.1.6). Для авторизованных пользователей.
         4. Подробнее – перенаправление на страницу 5.5.
   5. Страница с описанием модуля.
      1. Доступна всем пользователям.
      2. Вывод информации о модуле (метод 4.1.2)
         1. Изображение модуля.
         2. Название модуля.
         3. Описание модуля.
         4. Масса модуля.
         5. Длина модуля.
         6. Диаметр модуля.
   6. Страница со списком миссий.
      1. Доступна только авторизованным пользователям.
      2. Вывод информации о миссиях (метод 4.2.1)
         1. Статус миссии.
         2. Дата создания.
         3. Дата формирования.
         4. Дата завершения.
         5. Кем сформирована. Для администраторов.
         6. Список добавленных модулей.
      3. Действия
         1. Открыть миссию – перенаправление на страницу 5.7.
         2. Установить статус «завершен» или «отклонен» (метод 4.2.7). Для администраторов.
         3. Фильтр миссий по статусу и/или диапазону даты формирования (метод 4.2.1). Для авторизованных пользователей.
   7. Страница с описанием миссии.
      1. Доступна только авторизованным пользователям.
      2. Информация о миссии и списке модулей в ней (метод 4.2.2).
         1. Изображение модуля.
         2. Название модуля.
      3. Действия над черновиком
         1. Сформировать миссию (метод 4.2.4).
         2. Удалить миссию (метод 4.2.6).
         3. Удалить модуль из миссии (метод 4.2.7).
         4. Изменить данные в миссии (метод 4.2.3).
   8. Страница изменения модулей. Модератор может изменить или удалить модули.
      1. Доступна только администраторам.
      2. Информация о модуле (метод 4.1.2).
         1. Изображение модуля.
         2. Название модуля.
         3. Описание модуля.
         4. Масса модуля.
         5. Длина модуля.
         6. Диаметр модуля.
      3. Изменение полей модуля (метод 4.1.5).
   9. Страница со списком модулей.
      1. Доступна только администраторам.
      2. Информация о модулях (метод 4.2.2).
         1. Изображение модуля.
         2. Название модуля.
         3. Описание модуля.
         4. Масса модуля.
         5. Длина модуля.
         6. Диаметр модуля.
      3. Действия
         1. Удалить модуль (метод 4.1.6).
         2. Редактировать модуль – перенаправление на страницу 5.8.
         3. Добавить модуль – перенаправление на страницу 5.8.
2. **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.
3. **Требования к программному обеспечению:**
   1. Серверная часть
      1. ОС: Linux/Windows
      2. Docker 18.09.0 и выше
      3. Докер образы
      4. Go 1.20.0 и выше
      5. redis 7.2.3-alpine
      6. nginx 1.19.2-alpine
      7. quay.io minio RELEASE.2022-10-15T19-57-03Z
      8. СУБД: PostgreSQL 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+