|  |  |
| --- | --- |
|  | **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»

Техническое задание

Тема: «Автоматический подбор карт для игроков»

Студент: Оразов А.В.

Группа ИУ5-51Б

Преподаватель: Канев А.И.

2023г.

1. **Цель работы**

Реализовать систему для подборка игроками карт (игровых локаций), включающую в себя веб-сервис, веб-приложение, десктопное приложение и выделенный сервис расчета уровня игрока.

1. **Назначение**

Система предназначена для игроков, модераторов и администраторов игры. В игре предусмотрен ограниченный доступ к картам. Для получения доступа пользователю необходимо создать заявку на получение доступа к выбранной карте. Система предоставляет автоматизированный способ создания, учета и ведения заявок. Также она позволяет модераторам и администраторам принимать или отклонять заявки. Администраторы имеют возможность редактировать существующие и создавать новые карты.

1. **Задачи**:
   1. Разработать дизайн приложения.
   2. Создать базу данных в PostgreSQL.
   3. Создать веб-сервис на технологии dotnet 7.0.
   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 игрока в React.
   8. Реализовать интерфейс модератора и администратора в React.
   9. Создать десктопное приложение на Tauri.
   10. Создать выделенный сервис для расчета уровня игроков на Rust и axum.
   11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.
2. **Методы веб-сервиса:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Метод** | **Описание** | **Url** | **Входные данные** | **Выходные данные** |
| * 1. Методы авторизации и аутентификации | | | | | |
| 4.1.1. | POST | Регистрация | /api/authorization | login: string  password: string |  |
| 4.1.2. | POST | Вход в аккаунт | /api/authentication/login | login: string  password: string | auth cookies |
| 4.1.3. | GET | Выход из аккаунта Доступно только авторизированным пользователям | /api/authentication/logout | auth cookies |  |
| * 1. Методы карт | | | | | |
| 4.2.1. | GET | Возвращает файл, сохраненный в файловом хранилище сервиса. urlSuffix – путь до файла. | api/static\_data/{urlSuffix} | urlSuffix: string | Файл |
| 4.2.2. | GET | Возвращает список карт, удовлетворяющих переданным критериям | api/map/ | {  filterName: string,  maxPlayersCount: int  } | {  id: int,  name: string  width: int,  height: int,  maxPlayersCount: int,  coverImageUrl: string?,  mapStatus: MapStatus  }[] |
| 4.2.3. | GET | Возвращает список карт, удовлетворяющих переданным критериям и черновую заявку пользователя | api/map/ | auth cookies  {  filterName: string,  maxPlayersCount: int  } | {  id: int,  name: string  width: int,  height: int,  maxPlayersCount: int,  coverImageUrl: string?,  mapStatus: MapStatus  }[],  searchMapRequest |
| 4.2.4. | GET | Возвращает карту | api/map/{id}/ | id: int | {  id: int,  name: string  width: int,  height: int,  maxPlayersCount: int,  coverImageUrl: string?,  mapStatus: MapStatus  } |
| 4.2.5. | POST | Создает карту  Доступно только администраторам | api/map | auth cookies  {  name: string,  width: int,  height: int,  maxPlayersCount: int,  coverImageUrl: string?,,  coverImageFile: file,  } |  |
| 4.2.6. | PUT | Обновляет информацию о карте  Доступно только администраторам | api/map/{id} | auth cookies  {  id: int,  name: string?,  width: int?,  height: int?,  maxPlayersCount: int?,  coverImageUrl: string?,  coverImageFile: file?  } |  |
| 4.2.7. | DELETE | Помечает карту как удаленную  Доступно только администраторам | api/map/{id} | id: int |  |
| 4.2.8. | PUT | Делает карту доступной  Доступно только администраторам | api/map/make\_available/{id} | id: int |  |
| * 1. Методы заявок | | | | | |
| 4.3.1. | GET | Возвращает список заявок пользователя  Доступно только авторизированным пользователям | api/search\_maps\_request | auth cookies | searchMapsRequest: {  id: int,  creatorUserId: int,  status: SearchMapsRequestStatus,  creationDate: DateTime,  composeDate: DateTime?,  doneDate: DateTime?,  mapsCount: int,  } |
| 4.3.2. | GET | Возвращает подробную информацию о заявке  Доступно только авторизированным пользователям | api/search\_maps\_request/{id} | auth cookies  id: int | {  id: int,  creatorUser: User,  status: SearchMapsRequestStatus,  creationDate: DateTime,  composeDate: DateTime?,  doneDate: DateTime?,  maps: Map[],  } |
| 4.3.3. | GET | Возвращает текущую черновую заявку пользователя  Доступно только авторизированным пользователям | api/search\_maps\_request/current | auth cookies | {  id: int,  creatorUser: User,  status: SearchMapsRequestStatus,  creationDate: DateTime,  composeDate: DateTime?,  doneDate: DateTime?,  maps: Map[],  }? |
| 4.3.4. | PUT | Добавляет в заявку пользователя карту {map\_id}, возвращает id заявки, в которую была добавлена карта  Доступно только авторизированным пользователям | api/search\_maps\_request/add\_map/{map\_id} | auth cookies  map\_id: int | searchMapsRequestId: int |
| 4.3.5. | DELETE | Удаляет из заявки пользователя карту {map\_id}  Доступно только авторизированным пользователям | api/search\_maps\_request/remove\_map/{map\_id} | auth cookies  map\_id: int |  |
| 4.3.6. | PUT | Устанавливает заявке статус «сформирована»  Только если её текущий статус «черновик»  Доступно только авторизированным пользователям | api/search\_maps\_request/compose/{id} | auth cookies  id: int |  |
| 4.3.7. | DELETE | Устанавливает заявке статус «удалена»  Только если её текущий статус «черновик»  Доступно только авторизированным пользователям | api/search\_maps\_request/delete/{id} | auth cookies  id: int |  |
| 4.3.8. | GET | Возвращает отфильтрованный по датам создания и имени пользователя список заявок  Доступно только модераторам | api/search\_maps\_request/all/?begin\_date &end\_date&username | auth cookies  begin\_date: dtetime?  end\_date: datetime?  username: string? | searchMapsRequest[] |
| 4.3.9. | PUT | Устанавливает заявке статус «отклонена» или «завершена»  Только если её текущий статус «сформирована»  Доступно только модераторам | api/search\_maps\_request/status/{id}?status | id: int  status: “Finished” || “Cancelled” |  |
| * 1. Методы пользователей | | | | | |
| 4.4.1. | PUT | Устанавливает уровень пользователя  Вызывается внешним сервисом level-calculator | api/user/{id}?level | id: int  level: int |  |
| 4.4.2. | GET | Возвращает информацию об аккаунте пользователя. | api/user/me | auth cookies | {  id: int,  name: string,  level: int,  role: UserRole  } |
| * 1. Методы внешнего сервиса level-calculator | | | | | |
| 4.5.1. | POST | Рассчитывает уровень игрока после его участия в картах, указанных в заявке Метод внешнего сервиса level-calculator | api/calculate\_level | user\_id: int  search\_maps\_request\_id: int |  |

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. Действия
         1. Изменение полей карты. Можно изменять все вышеперечисленные поля карты.
         2. Сохранить изменения (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. Открыть карту – перенаправляет на страницу 5.5.
         2. Добавить карту – перенаправляет на страницу 5.
2. **Требования к программному обеспечению:**
   1. Серверная часть
      1. ОС: Linux/Windows
      2. Docker
      3. Докер образы
      4. dotnet/sdk 7.0
      5. dotnet/aspnet 7.0
      6. redis 7.2.3-alpine
      7. nginx 1.19.2-alpine
      8. quay.io minio RELEASE.2022-10-15T19-57-03Z
      9. postgres 14
      10. rust 1.72.0
      11. scratch
   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.