

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Московский политехнический университет»**

**Кафедра «Информационные технологии»
Образовательная программа «Веб-технологии»**

Отчет по дисциплине

«Проектная деятельность»

Тема: «Сервис для заказа пропусков на территорию Московского
Политеха»

Выполнили:

Студенты

Гайдучик Александр Валерьевич, гр. 181-321

Иванов Григорий Александрович, гр. 181-321

Гучустян Валерия Саркисовна, гр. 191-322

Курникова Виктория Александровна, гр.191-322

Ерошин Алексей Анатольевич, гр.191-323

Глава команды

Гайдучик Александр Валерьевич, гр. 181-321

Принял:

Старший преподаватель

Даньшина М.В.

Москва 2022

Содержание

Введение.....	3
Распределение ролей.....	4
Глава 1. Формирование требований к проекту	
1.1 Анализ существующей проблемы.....	5
1.2 Формулирование цели, задач по проекту, критериев для приемки работы.....	5
1.3 Проектирование приложения, UML диаграммы, план приёма работы.....	6
1.4 Используемые технологии при создании проекта.....	6
Глава 2. Реализация проекта	
2.1 Проектирование разделов, структура сайта.....	8
2.2 Проектирование структуры базы данных на основе задачи.....	8
2.3 Конкретная реализация задачи на серверной стороне.....	9
2.4 Конкретная реализация задачи на клиентской стороне.....	10
2.5 Разработка дизайна.....	10
Глава 3. Сопровождение проекта	
3.1 Инструкция для пользователя.....	17
3.2 Правовые вопросы.....	22
Глава 4. Результаты.....	23

Введение

В наши дни очень важно эффективно распоряжаться временными ресурсами и правильно передавать данные. Для того, чтобы не тратить время на долгий и рутинный процесс получения пропусков мы разработали систему, которая автоматизирует все этапы их получения. При разработке нашей системы по выдаче пропусков мы делали упор на следующие аспекты:

- оперативность работы между дирекцией и бюро пропусков
- ведение учета всех выданных пропусков
- ускорение обработки заявок за счет сокращения времени уходящего на дублирование информации при передаче данных из дирекции в бюро пропусков
- исключение вероятности передачи неправильной информации и ее потери, за счет сохранения каждой заявки на сервисе

Система, которую разрабатывает наша команда, позволит сделать выдачу пропусков эффективнее, также мы расскажем о том, что будет реализовано и что уже создано в прототипе, о чем будет описано далее в документации к проекту.

При работе над проектом проводились занятия по работе с фреймворком Angular, работе с Mongoose, а также по работе с Nodejs и Expressjs, что позволило улучшить навыки работы разработчиков прототипа. Занятия проводились во время командного собрания и в индивидуальных консультациях.

Распределение ролей

Наша команда распределила роли следующим образом:

Гайдучик Александр Валерьевич - team lead, frontend, подготовка документации, notion (кейсы, модели).

Иванов Григорий Александрович - backend, работа с базой данных.

Гучустян Валерия Саркисовна - дизайнер, работа с документацией.

Курникова Виктория Александровна - backend, подготовка отчётных материалов.

Ерошин Алексей Анатольевич - frontend, тестирование.

Глава 1. Формирование требований к проекту

1.1 Анализ существующей проблемы

При выполнении проекта был проведен анализ действующей системы выдачи пропусков, которая является не практичной. Проблема заключается в следующем. Для заказа пропуска посетителю нашего университета необходимо сделать несколько звонков в центр по работе со студентами, которые в свою очередь заказывают сам пропуск в Бюро пропусков также посредством звонка. Данная система является надежной, но весь процесс изготовления пропуска занимает довольно большое количество времени.

Вероятно, в анализе не были учтены все ситуации, но в процессе решения и создания сервиса, команда максимально учла все идеи для реализации, в том числе после встречи с представителем кафедры.

Нашей командой, после изучения существующей проблемы была выдвинуты цели, задачи, а также спроектированы кейсы работы с сервисом, о чём описано далее по документации.

1.2 Формулирование цели, задач по проекту, критериев для приемки работы

В ходе изучения анализа существующего решения, анализа востребованности решения и изучения аналогов мы сформулировали следующие задачи и цели:

1. Разработка прототипа
2. Встреча с фокус-группой (сотрудники Политеха, студенты Политеха)
3. Разработка дизайна в Figma
4. Разработка API
5. Разработка frontend части (прототипа)
6. Разработка backend части (прототипа)
7. Проверка на баги/ошибки, а также доработка прототипа
8. Презентация готовой реализации в рамках «Проектной деятельности»

Таким образом, нашей основной целью является создание информационной системы - сервиса для заказа пропусков на территорию Московского Политеха.

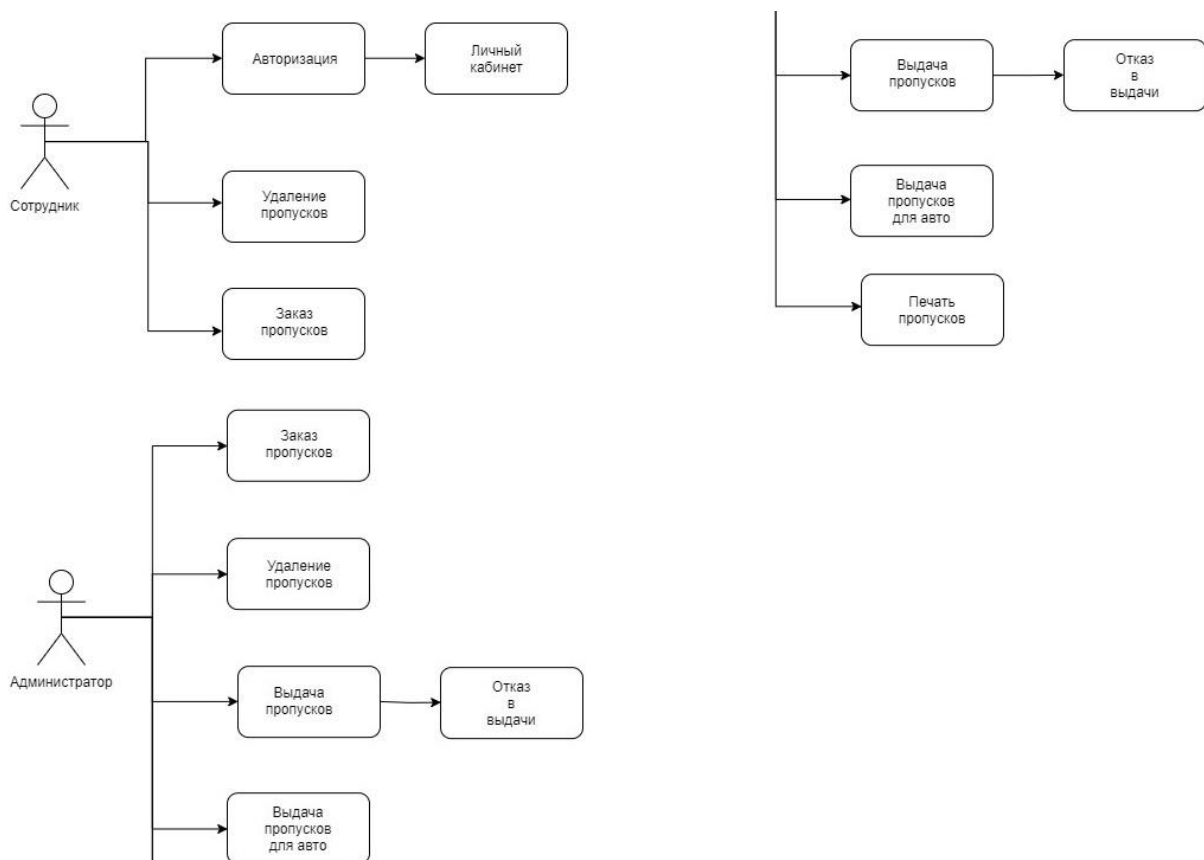
Дополнительно был разработан прототип системы оповещений, который позволит получать уведомления о том, что пропуск был создан, соответственно намечены дополнительные задачи и цели:

1. Разработка прототипа (описание идеи на основе разработанного прототипа сервиса продажи автозапчастей)
2. Дополнительные вопросы, касаемые оповещения продавцов во время встречи с фокус-группой
3. Разработка API
4. Разработка backend части прототипа
5. Разработка frontend части для тестирования прототипа

1.3 Проектирование приложения, UML диаграммы, план приема работы

Проектирование приложения выдвинуто в несколько этапов. Для удобства ведения был использован сервис Trello, где каждый участник видел свои задачи, а также сервис Notion, в котором были описано ТЗ по каждому участнику.

На следующей UML-диаграмме расписаны методы взаимодействия пользователя с системой, покупателя с продавцом:



1.4 Используемые технологии при создании проекта

При создании проекта использовались свободно распространяемые инструменты, фреймворки и язык программирования.

В качестве языка программирования для backend выступает JavaScript (информация о языке - <https://ru.wikipedia.org/wiki/JavaScript>), в качестве платформы для работы сервиса – Node.js (информация - <https://nodejs.org/>), также некоторые дополнительные компоненты, о которых будет описано далее в документации.

В качестве языка программирования для frontend использовался JavaScript, фреймворк Angular (информация - <https://angular.io/docs>), а также Node.JS

Так же, при работе с проектом использовался «git-flow» - распределенная система контроля версий (version control system – VCS).

Контроль версий означает что мы храним все версии редактируемых документов и можем вернуться к любой сохраненной версии в любой момент времени, в случае появления каких-либо проблем, либо ошибок.

Команда использовала технологию «CI/CD», что означает концепцию непрерывной интеграции и доставки (CI/CD) — основа наших тестирований.

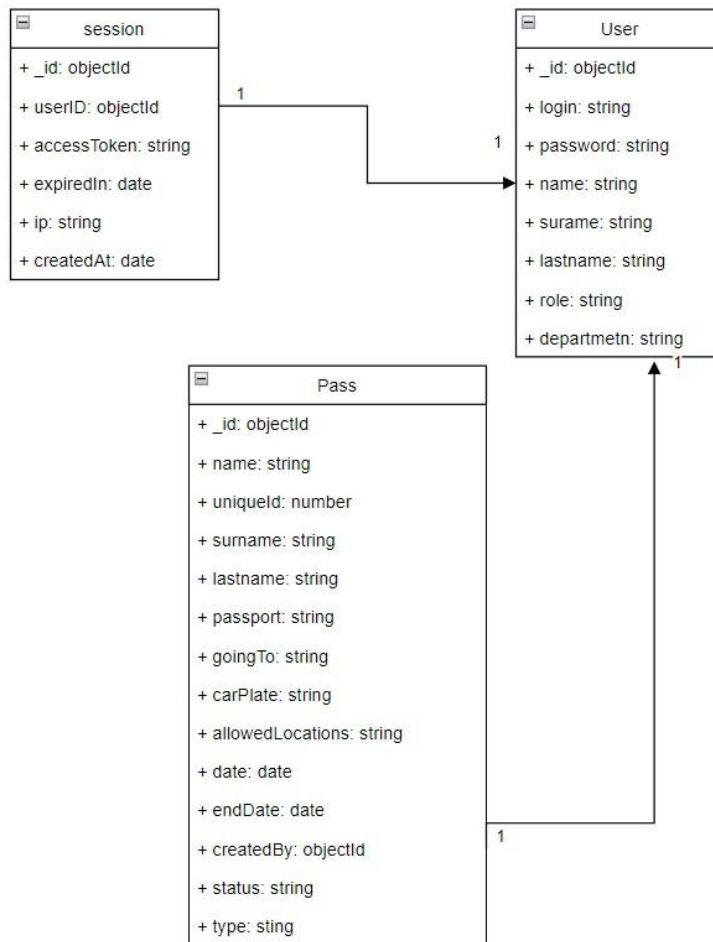
Информация о CI/CD — это концепция, которая реализуется как конвейер, облегчая слияние только что закомиченного кода (то есть, только что написанного кода программы и отправленного в Github) в основную кодовую базу. Концепция позволяет запускать различные типы тестов на каждом этапе и завершать его запуском с разворачиванием закомиченного кода в фактический продукт, который видит наша команда, во время тестирования.

В свою очередь, для удобства работы с проектом, мы использовали таск менеджер, о котором было написано ранее – Notion. Это - инструмент для управления проектами, который значительно упрощает рабочий процесс.

Благодаря использованию, мы экономим время на переписках и оффлайн-обсуждениях. Кроме этого, такие платформы позволяют узнать, насколько эффективно и слаженно работает команда.

Были проведены встречи со студентами разных групп для обсуждения работы системы – принцип получения уведомлений и новостей, а также дополнения к интерфейсу, такие как личный кабинет и т.д.

Проводилось дополнительно проектирование БД:



Глава 2. Реализация проекта

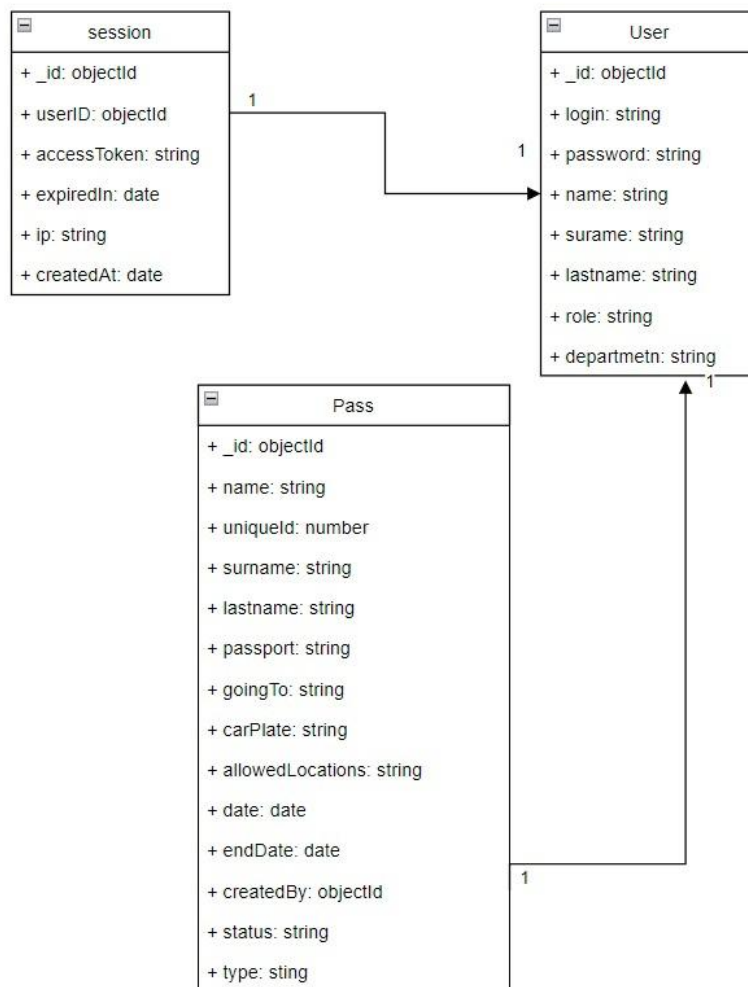
2.1 Проектирование разделов, структура сайта

В результате изучения задачи были выведены следующие разделы сервиса (для удобства введем «Администратор», «Пользователь»):

1. «Администратор» - управление системой, базой данных и основным функционалом информационной системы, а также просмотр статистики
2. «Пользователь» - разделен на два подтипа:
 - а. Бюро - реализован интерфейс резервации деталей
 - б. Работник - реализован интерфейс работы с деталями (добавление, удаление, управление резервом и иные функции)

2.2 Проектирование структуры базы данных на основе задачи.

В ходе подготовки проекта нами было решено использовать базу данных с, минимум, двумя таблицами: «user» - для хранения данных о конкретных пользователях и «pass» - таблица для хранения информации о пропусках.



2.3 Конкретная реализация задачи на серверной стороне

В вопросе реализации серверной стороны, далее backend, нами были использованы следующие технологии:

1. Node.JS – использование этой программной платформы позволяет сделать проект масштабируемым, позволяет обрабатывать сразу много соединений за раз. Для каждого соединения вызывается функция обратного вызова, однако, когда соединений нет Node.js ожидает в состоянии покоя. Следует обратить внимание, что этот подход контрастирует с более распространенной на сегодняшний день моделью параллелизма, в которой используются параллельные OS потоки. Такой подход является относительно неэффективным и очень сложным в использовании. Кроме того, пользователи платформы могут не беспокоиться о блокировках процессов или же зависаний, поскольку их не существует. Почти ни одна из функций в Node.js не работает напрямую с I/O, поэтому поток никогда не блокируется. В следствии этого на Node.js легко разрабатывать масштабируемые системы, чем наша команда и воспользовалась.

2. Mongoose - представляет специальную ODM-библиотеку (Object Data Modelling) для работы с MongoDB (будет описано далее), которая позволяет сопоставлять объекты

классов и документы коллекций из базы данных. Грубо говоря, Mongoose работает подобно инструментам ORM. ORM – это технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

3. MongoDB – документо ориентированная система управления базами данных с открытым исходным кодом, не требующая описания схемы таблиц. Классифицирована как NoSQL, использует JSON-подобные документы и схему базы данных.

Таким образом, связка всех используемых технологий позволяет создать максимально удобный сервис, как для пользователя, так и для разработчика.

2.4 Конкретная реализация задачи на клиентской стороне

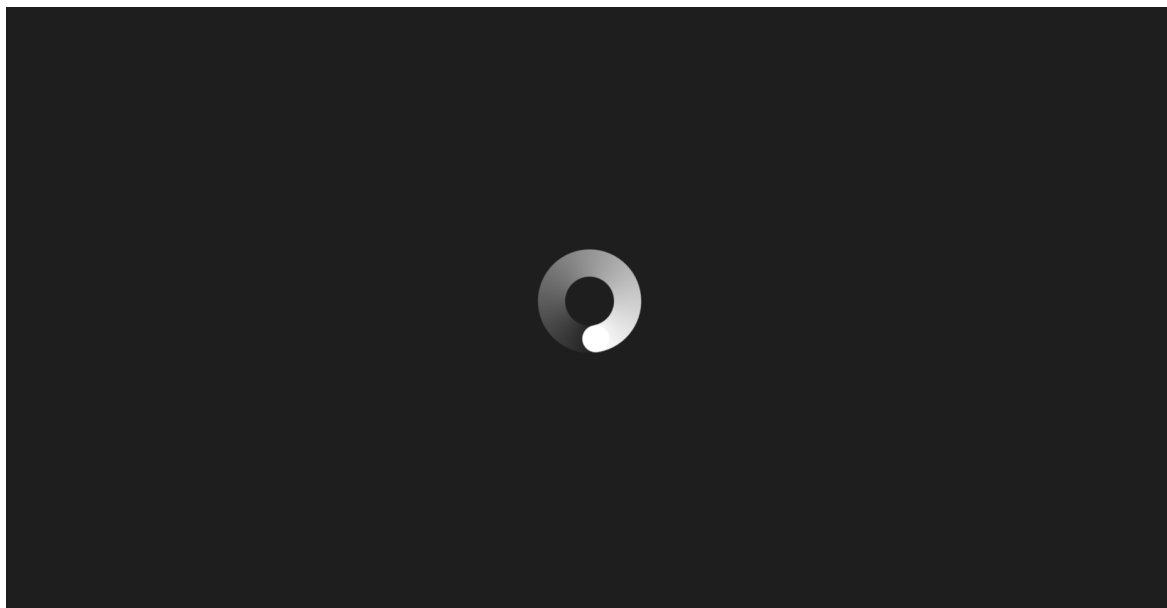
Angular - Открытая и свободная платформа для разработки веб-приложений, написанная на языке TypeScript, разрабатываемая командой из компании Google, а также сообществом разработчиков из различных компаний. Angular - полностью переписанный фреймворк от той же команды, которая написала AngularJS

NodeJS - Программная платформа, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через API, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода

Библиотеки npm - инструмент, управляющий библиотеками программирования Javascript для Node.js, данный инструмент на самом деле необходим для мира открытого исходного кода. Используется для скачивания пакетов из облачного сервера npm, либо для загрузки пакетов на эти сервера.

2.5 Разработка дизайна

Страница загрузки сервиса:

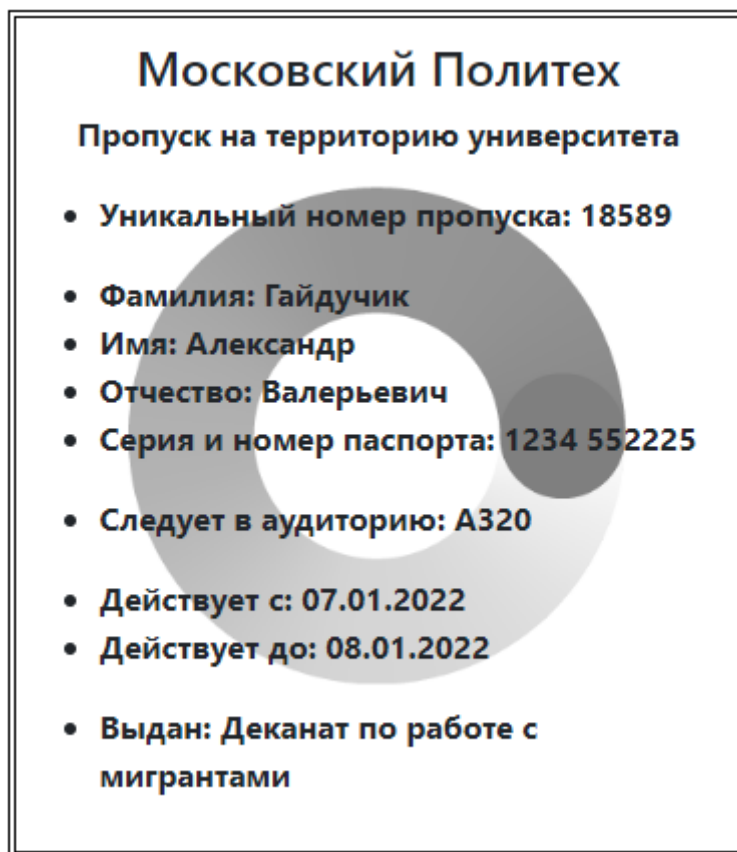


Печатные пропуска

Автомобильный пропуск:

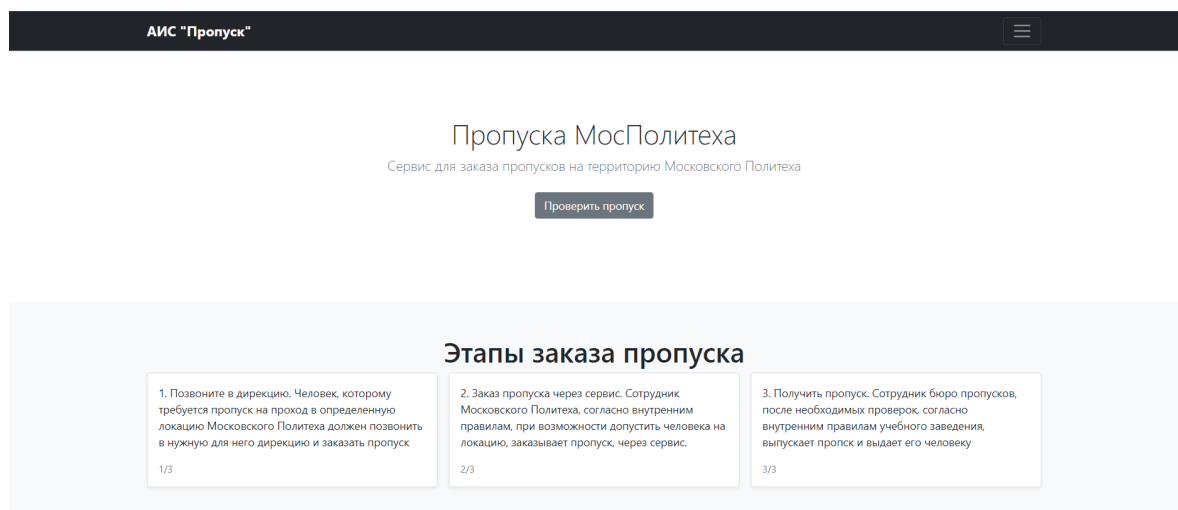


Пропуск посетителя:



Незарегистрированный пользователь

Главная страница сервиса:



Страница просмотра пропуска:

АИС "Пропуск"

Проверка пропуска

Номер пропуска

Фамилия

Проверить

© 2022 Федеральное государственное автономное образовательное учреждение высшего образования «Московский политехнический университет»

Отображение пропуска (автомобильного):

АИС "Пропуск"

Просмотр пропуска

Номер автомобиля: M573 EX799

Разрешенные локации:

Сотрудник: Иванов Григорий Александрович

Действует с: 07.01.2022

Действует до: 06.01.2022

Выдан: Деканат по работе с мигрантами

Статус пропуска: Действителен

Фамилия

Иванов

Проверить

© 2022 Федеральное государственное автономное образовательное учреждение высшего образования «Московский политехнический университет»

Страница авторизации:



Вход

E-Mail

Пароль

Войти

© 2022 Федеральное государственное автономное образовательное учреждение высшего образования «Московский политехнический университет»

Страница 404:

404

Кажется, вы потерялись.

[Вернуться на главную страницу](#)

© 2022 Федеральное государственное автономное образовательное учреждение высшего образования «Московский политехнический университет»

Сотрудник

Страница для сотрудников МосПолитеха, где можно ознакомиться с заказанными, выданными и закрытыми пропусками и сроком их действия:

Управление и заказ пропусков

Создать пропуск ▾

Заказанные пропуска

Выданные пропуска

Закрытые пропуска

ID	Фамилия Имя Отчество	Действует с	Действует до	Статус	Действия
1	Иванов Григорий Александрович	07.12.2021	07.12.2021	Заказан	⚙

© 2022 Федеральное государственное автономное образовательное учреждение высшего образования «Московский политехнический университет»

Для каждого пропуска есть возможность управления:

Управление пропуском



Выберите действие

Открыть пропуск

Распечатать пропуск

Отклонить пропуск

Удалить пропуск

Также на странице есть возможность создать пропуск на человека и на автомобиль:

Создать пропуск ▾

Пропуск на человека

Пропуск для автомобиля

Создание пропуска

×

Фамилия

Имя

Отчество

Серия и номер паспорта

Если не вводится, оставить пустым.

Следует в аудиторию

Дата начала действия

Дата окончания действия

Примечание

☐ Check me out

Submit

Создание пропуска

×

Фамилия

Имя

Отчество

Серия и номер паспорта

Если не вводится, оставить пустым.

Номер автомобиля

Разрешенные локации

☐ БС ☐ ПК ☐ ПР ☐ АВ

Дата начала действия

Дата окончания действия

Примечание

☐ Check me out

Submit

Зарегистрированный пользователь

Страница входа:



Вход

Е-Mail

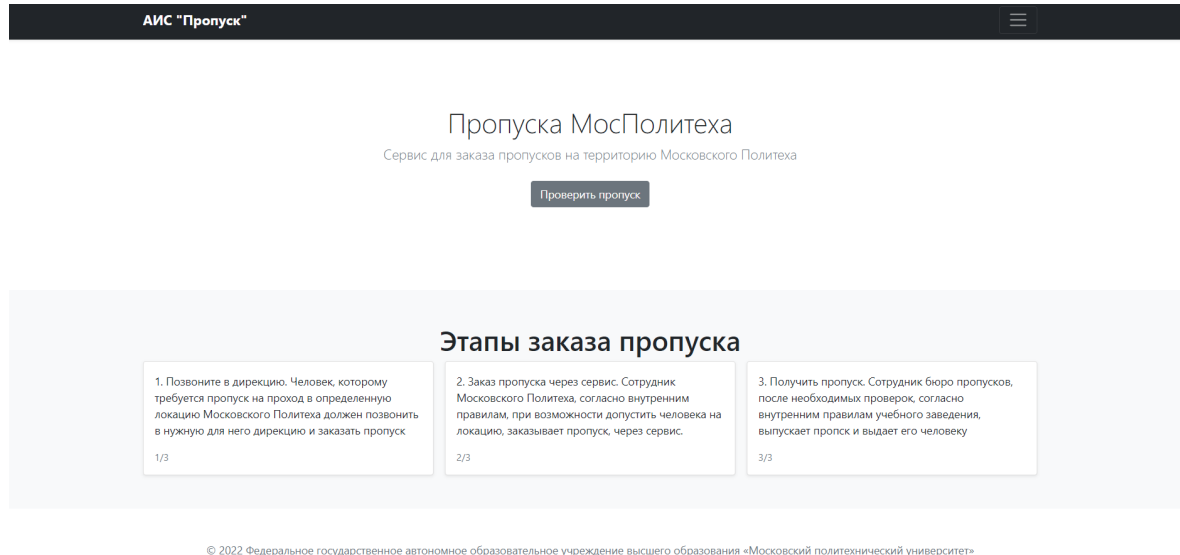
Пароль

Войти

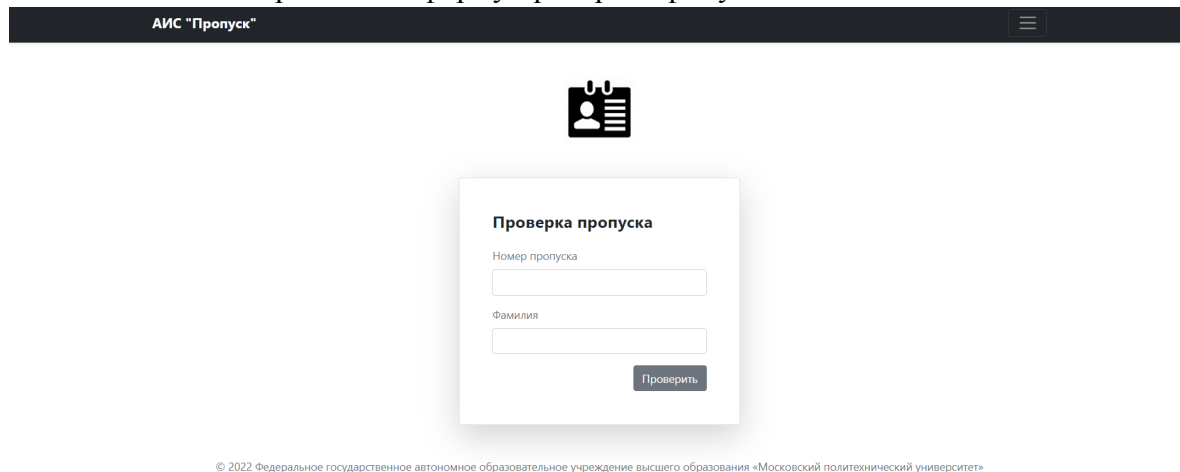
Глава 3. Сопровождение проекта

3.1 Инструкция для пользователя

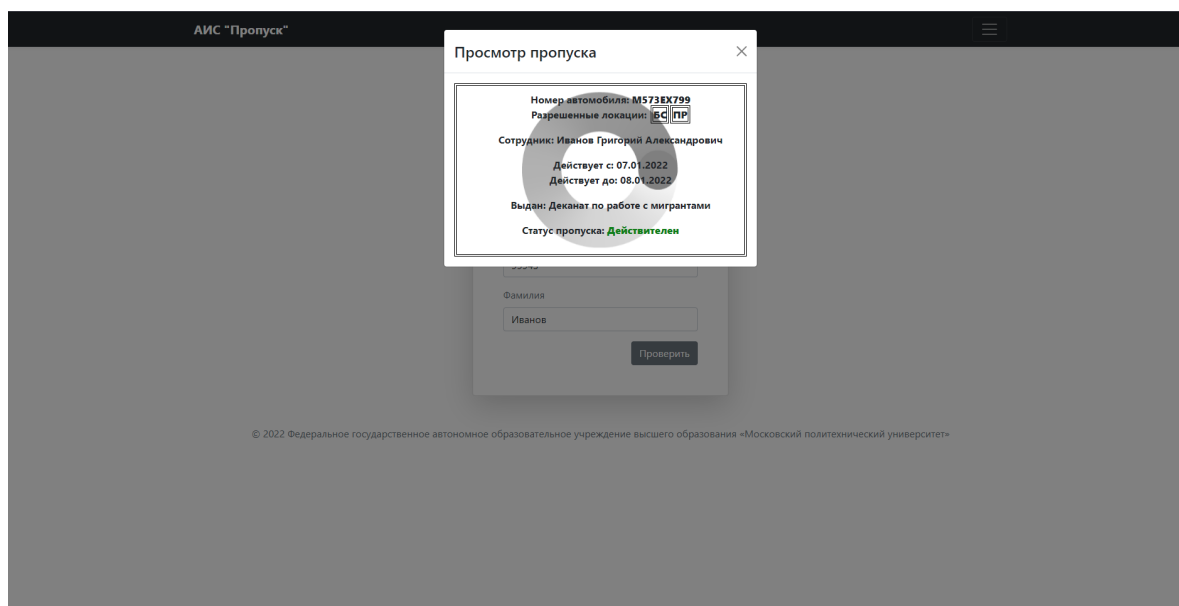
На главной странице есть краткое описание сервиса и кнопка проверить пропуск:



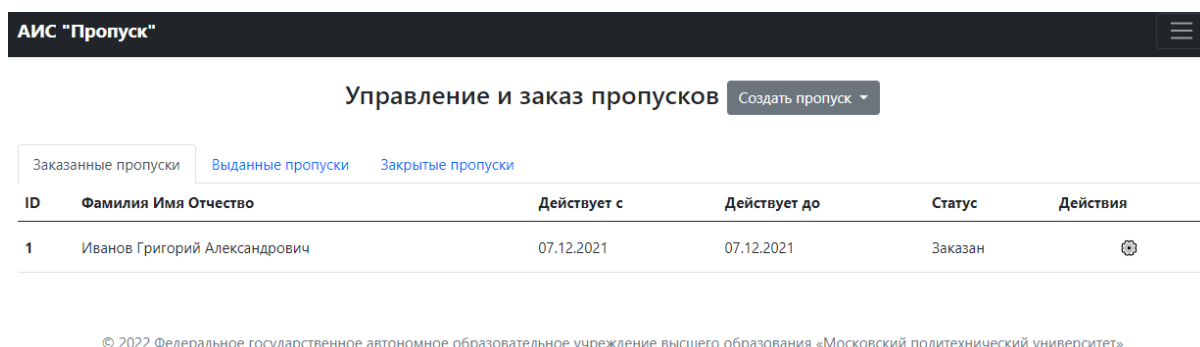
Нажав на неё мы переходим в форму проверки пропуска:



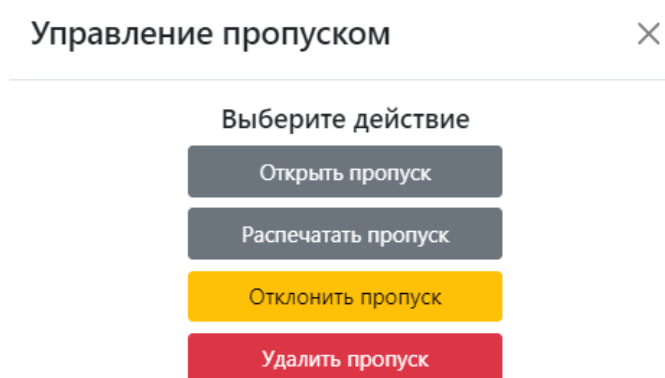
При вводе правильных данных, пользователь получает возможность посмотреть пропуск и его действительность



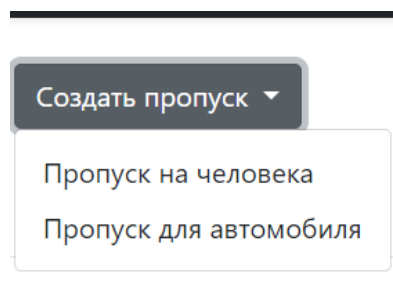
У сотрудников есть возможность ознакомиться с пропусками, перейдя на страницу бюро:



Нажав на кнопку в колонке действий сотрудник может управлять пропусками:



А также в верхней части страницы есть кнопка для создания пропуска:

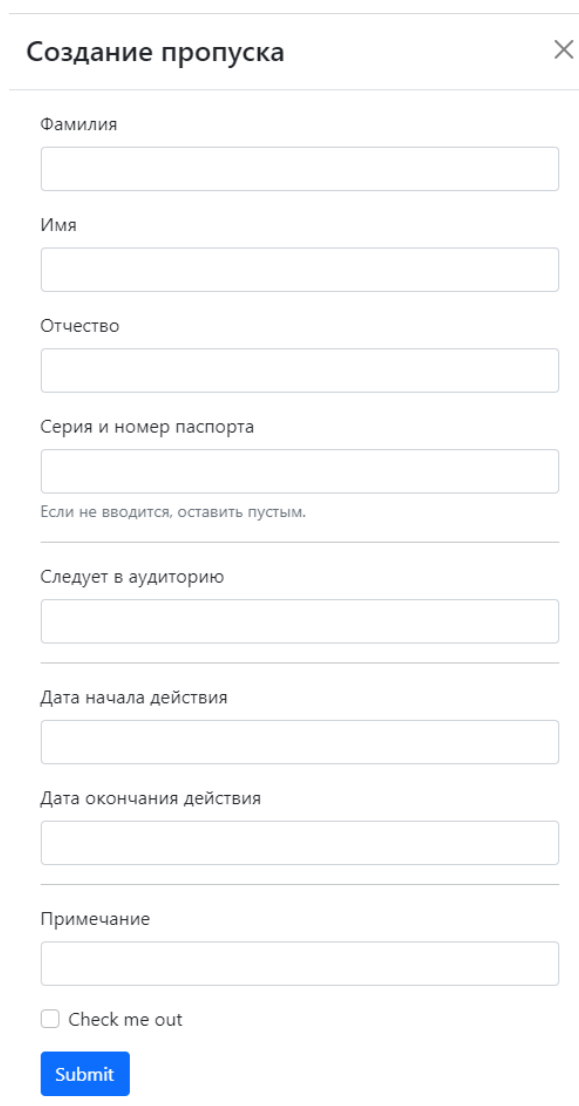


Создать пропуск ▾

- Пропуск на человека
- Пропуск для автомобиля

При выборе вида пропуска появится форма для заполнения данными, необходимыми для выдачи пропуска:

Для человека



Создание пропуска ×

Фамилия

Имя

Отчество

Серия и номер паспорта

Если не вводится, оставить пустым.

Следует в аудиторию

Дата начала действия

Дата окончания действия

Примечание

☐ Check me out

Submit

Для автомобиля:

Создание пропуска

×

Фамилия

Имя

Отчество

Серия и номер паспорта

Если не вводится, оставить пустым.

Номер автомобиля

Разрешенные локации

☐ БС ☐ ПК ☐ ПР ☐ АВ

Дата начала действия

Дата окончания действия

Примечание

☐ Check me out

Submit

3.2 Правовые вопросы

В ходе выбора направления проектной деятельности было решено создавать информационную систему - сервис который реализует продажу автозапчастей между клиентов и продавцом.

Права на проект принадлежат команде: Гайдучик Александр, Иванов Григорий, Курникова Виктория, Гучустян Валерия, Ерошин Алексей.

Также права на приложение (в том числе и на исходный код) принадлежат команде. Далее перечислены лицензии на использование технологий, находящиеся в проекте:

1. Angular – MIT
2. Node.js – Free/Open Source, MIT
3. Express – MIT
4. Mongoose – MIT

5. MongoDB – MIT

MIT - лицензия открытого программного обеспечения, разработанная Массачусетским технологическим институтом. Лицензия MIT является одной из самых ранних свободных лицензий, так как она относительно проста и иллюстрирует некоторые из основных принципов свободного лицензирования.

Она является разрешительной лицензией, то есть позволяет программистам использовать лицензируемый код в закрытом программном обеспечении при условии, что текст лицензии предоставляется вместе с этим программным обеспечением. Таким образом, мы можем свободно использовать компоненты.

Глава 4. Результаты

Репозиторий на github:

https://github.com/Rouvas/pd_2021

Ссылка на проект:

https://drive.google.com/drive/folders/11PHg1IOl1mYs57fuqqv_DJTKpHjiZY0F?usp=sharing

Дизайн (Figma):

<https://www.figma.com/file/0gOpePnGAKgSfXKKtTykqe/pd?node-id=0%3A1>

Лендинг проекта:

<http://pd-2021-2.std-1026.ist.mospolytech.ru/>

Рабочая документация Notion:

<https://cosmic-water-ee7.notion.site/e112c88654714835b9242c8d8e47dfca>

Иные полезные ссылки по проекту:

<https://docs.google.com/document/d/1657xunjRPaTkRnj0sVVXP8sZxn3XprRTafKxpDnxyzk/edit?usp=sharing>