Учреждение образования

«Белорусский государственный технологический университет»

Кафедра программной инженерии

Лабораторная работа №1

«Формирование требований к информационной системе»

Выполнила:

студентка 4 курса 4 группы ФИТ

Коржова В. С.

Преподаватель: Якунович А.В.

Минск 2023

**Оглавление**

[Введение 3](#_Toc146046005)

[1 Основания для разработки 4](#_Toc146046006)

[2 Назначение разработки 5](#_Toc146046007)

[3 Требования к программному средству 6](#_Toc146046008)

[3.1 Требования к функциональным характеристикам 6](#_Toc146046009)

[3.1.1 Требования к клиентской части 6](#_Toc146046010)

[3.1.2 Требования к серверной части 6](#_Toc146046011)

[3.2 Требования к надёжности 7](#_Toc146046012)

[3.3 Условия эксплуатации 7](#_Toc146046013)

[3.4 Требования к составу и параметрам технических и программных средств 7](#_Toc146046014)

[3.5 Требования к информационной и программной совместимости 8](#_Toc146046015)

[4 Требования к программной документации 9](#_Toc146046016)

[5 Стадии и этапы разработки 10](#_Toc146046017)

# Введение

В данном документе представлен полный перечень требований к созданию приложения по отслеживанию расписания городского общественного транспорта "Rover". В современном мире общественный транспорт играет важную роль в повседневной жизни горожан, облегчая им передвижение и связывая различные части города. Несмотря на существующие приложения для отслеживания расписания и маршрутов, всегда есть потребность в улучшении и инновациях в этой области.

Основной целью приложения "Rover" является предоставление удобного и надежного инструмента для отслеживания общественного транспорта с уникальными функциональными возможностями, спроектированными для удовлетворения потребностей разнообразной аудитории. Наше приложение будет ориентировано на путешественников всех возрастов и предоставит им возможность легко находить оптимальные маршруты, получать актуальную информацию о расписании и эффективно планировать свои поездки.

В современных городах существует множество видов общественного транспорта, разнообразных маршрутов и сложных сетей передвижения. Наше приложение стремится стать универсальным инструментом, который поможет не только обычным пассажирам, но и профессионалам в сфере планирования маршрутов и городской мобильности.

В последующем разделе данного документа мы более подробно рассмотрим требования к разработке нашего приложения "Rover", его функциональные характеристики и ожидаемые результаты.

# Основания для разработки

Веб-приложение «Rover» разрабатывается в качестве Дипломного проекта для получения квалификации «Инженер-программист» по специальности «Программное обеспечение информационных технологий» на Факультете Информационных Технологий Белорусского Государственного Технологического Университета. На основании стандарта организации подготовки, представления и защиты дипломных работ СТП БГТУ 001-2019, утвержден и введён в действие приказом № 108 от 20.03.2019 Министерством образования Республики Беларусь.

# 2 Назначение разработки

Функциональным назначением нашего приложения является создание платформы, обеспечивающей удобство и эффективность в использовании общественного транспорта. Основные функции включают:

Расписание и маршруты: Приложение предоставляет пользователю доступ к актуальным расписаниям и маршрутам общественного транспорта. Пользователи могут легко находить информацию о времени прибытия автобусов, трамваев, метро и других видов транспорта.

Поиск ближайших остановок: Пользователи могут быстро найти ближайшие к ним остановки и узнать информацию о расстоянии до них. Это позволяет оптимизировать маршрут и сэкономить время.

Интерактивные карты: Пользователи могут просматривать интерактивные карты с отмеченными маршрутами и остановками, что помогает им лучше понимать географию общественного транспорта в своем городе.

Эксплуатационным назначением нашего приложения является обеспечение пользователям удобного и информативного инструмента для планирования поездок, мониторинга состояния общественного транспорта и улучшения их общего опыта в городском перемещении.

# Требования к программному средству

## 3.1 Требования к функциональным характеристикам

Функционал, доступный пользователям, отличается в зависимости от роли пользователя. В приложении доступны следующие роли:

– гость;

* авторизованный пользователь;
* администратор.

## 3.1.1 Требования к клиентской части

Клиентская часть должна поддерживать онлайн режим работы. Обеспечивать работу в любом размере окна веб-браузера с адаптацией пользовательского интерфейса. Интерфейс клиентского приложения должен быть информативным и незагруженным. Также обеспечивать выполнение перечисленных ниже функций:

* регистрация пользователя (гость);
* авторизация пользователя (гость);
* ввод и изменение личных данных (авторизованный пользователь);
* поиск остановок по названию (авторизованный пользователь);
* изменение пароля (авторизованный пользователь);
* восстановление пароля (авторизованный пользователь);
* просмотр расписания (авторизованный пользователь);
* покупка проездного билета (авторизованный пользователь);
* просмотр информации о всех пользователях (администратор);
* добавление, удаление типов проездных билетов (администратор);
* добавление, удаление расписания транспорта (администратор);
* добавление, удаление остановок (администратор).

Клиентская часть должна быть реализована на React Js.

## 3.1.2 Требования к серверной части

Сервер должен быть создан на базе платформы Node.js с использованием языка программирования JavaScript.

Серверная часть должна обеспечивать следующую функциональность:

* проверка прав пользователя;
* аутентификация и авторизация;
* доступ к базе данных MongoDB;
* защита персональных данных;
* обработка возникающих ошибок;
* предоставление API для работы с клиентской частью приложения.

Перечисленные требования обеспечивают необходимую работоспособность и безопасность серверной части приложения. Это позволяет обеспечить надежное и эффективное функционирование системы.

## 3.2 Требования к надёжности

Программное средство должно:

* обеспечивать проверку корректности данных авторизации пользователя;
* хранение пароля пользователя в базе данных в зашифрованном виде;
* обеспечивать защиту личных данных пользователя от внешнего вмешательства;
* ограничить возможность создания нескольких учетных записей одному и тому же пользователю.

## 3.3 Условия эксплуатации

Эксплуатация приложения рассчитана людей без каких-либо возрастных диапазонов, ограничений по половому признаку нет. Приложение должно иметь интуитивно понятный интерфейс, при этом он должен быть неперегруженным и в меру информативным. Требуется рассчитывать, что эксплуатация приложения возможна в условиях плохого соединения с интернетом, либо людьми с нарушением зрения.

## 3.4 Требования к составу и параметрам технических и программных средств

Минимальные рекомендованные аппаратные требования к пользовательской машине:

* процессор архитектуры x86 или arm;
* минимальное количество ядер процессора – 4;
* минимальный объем оперативной памяти – 8 ГБ;
* минимальный доступный объем дискового пространства – 20 ГБ.

Рекомендованные аппаратные требования для серверной машины:

* процессор архитектуры x86 с поддержкой виртуализации;
* количество ядер от 8 до 16;
* минимальный объем оперативной памяти – 16 ГБ;
* минимальный доступный объем дискового пространства – 512 ГБ.

Программные средства требующиеся для работы приложения:

* клиент – операционная система Windows, Linux, MacOS (ПК);
* клиент – операционная система Android, iOS (Мобильный телефон);
* сервер – операционная система Windows, Linux.

## 3.5 Требования к информационной и программной совместимости

Приложение клиент и сервер общаются с использованием сети интернет по протоколу HTTP. Приложение сервер общается с СУБД локально (если они расположены на одной машине) или по сети интернет по протоколу RDA.

# 4 Требования к программной документации

Программная документация должна быть представлена в виде отдельной конечной точки приложения для клиентской и серверной части. Так же возможно наличие комментариев внутри кода программы.

Документация для серверной части должна быть представлена в виде описания конечных точек и должна содержать:

* URL конечной точки;
* название используемого HTTP метода;
* пример данных, передаваемых на сервер по указанной конечной точке;
* пример данных, возвращаемых с сервера.

Для достижения этих целей может быть использована спецификация описания API под названием Open API.

# Стадии и этапы разработки

Разработка должна быть проведена в 3 этапа:

* техническое задание;
* технический проект;
* внедрение.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

* разработка программы;
* разработка программной документации;
* испытание программы.

На этапе разработки программы должна быть выполнена работа по программированию и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.