Учреждение образования

«Белорусский государственный технологический университет»

Кафедра программной инженерии

Лабораторная работа №1

**«Формирование требований к информационной системе»**

Выполнил:

Минск 2021

Содержание

[1. Описание целевой аудитории 3](#_Toc83195188)

[1.1 Целевая аудитория 3](#_Toc83195189)

[1.2 Роли пользователей 3](#_Toc83195190)

[2. Функциональные требования 4](#_Toc83195191)

[3. Основные системные требования 6](#_Toc83195192)

[3.1 Архитектура и структура ПО 6](#_Toc83195193)

[3.2 Описание структур данных 7](#_Toc83195194)

[3.3 Описание интерфейсов 7](#_Toc83195195)

[3.4 Используемые технологии 8](#_Toc83195196)

[4. Аппаратные требования 9](#_Toc83195197)

# **Описание целевой аудитории**

## **1.1 Целевая аудитория**

Конечными пользователями программного средства являются студенты факультета учебного учреждения, родители студентов, преподавательский состав, администрация факультета. В администрацию факультета входят декан, заместители декана, секретарь и методист.

## **Роли пользователей**

В приложении выделено 10 ролей: администратор, декан, лектор, преподаватель, куратор группы, методист, секретарь, студент, родитель студента, староста. Каждой роли присвоена своя зона ответственности. Каждый пользователь может иметь одну роль и одну или несколько привилегий. Используется поглощающая модель ролей, то есть каждая вышестоящая роль будет иметь все привилегии нижестоящий.

# **Функциональные требования**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Роли | Администратор | Декан | Лектор | Преподаватель | Секретарь | Методист | Студент | Староста | Родитель студента |
| Функционал |
| Просмотр расписания | + | + | + | + | + | + | + |  | + |
| Добавление, изменение, удаление расписания | + | + |  |  | + |  |  |  |  |
| Просмотр успеваемости | + | + | + | + | + |  | + |  | + |
| Добавление, изменение, удаление успеваемости | + |  | + | + |  |  |  |  |  |
| Просмотр посещения | + | + | + | + | + |  | + |  | + |
| Добавление, изменение, удаление посещения | + |  | + | + |  |  |  | + |  |
| Просмотр объявлений | + | + | + | + | + |  | + |  | + |
| Изменение статуса объявления | + | + | + | + | + |  | + |  | + |
| Добавление, изменение, удаление объявления | + | + | + | + | + |  |  |  |  |
| Просмотр заявлений | + | + |  |  | + |  | + |  |  |
| Добавление, изменение, удаление заявлений | + |  |  |  |  |  | + |  |  |
| Подтверждение/отклонение заявлений | + | + |  |  | + |  |  |  |  |
| Прикрепление комментариев к заявлению | + | + |  |  | + |  | + |  |  |
| Обновление информации о пользователе | + | + |  |  | + |  |  |  |  |
| Добавление, изменение, удаление учебного плана |  |  |  |  |  | + |  |  |  |
| Просмотр учебного плана | + | + | + | + | + | + | + |  | + |
| Добавление комментария к занятию | + |  | + | + |  |  |  |  |  |
| Получение уведомлений | + | + | + | + | + |  | + |  | + |
| Просмотр личного кабинета | + | + | + | + | + |  | + |  |  |
| Изменение интерфейса | + | + | + | + | + | + | + | + | + |

# **Основные системные требования**

## **Архитектура и структура ПО**

Приложение имеет клиент-серверную архитектуру. Сервер представляет собой монолитную архитектуру. Клиент – веб-браузер.

Приложение реализовано согласно паттерну MVC.

## **Описание структур данных**

В приложении используется реляционная база данных – PostgreSQL.

Ниже приведено подробное описание бизнес-модели:

1. Пользователь:

* Имя
* Фамилия
* Отчество
* Номер группы для роли студента
* Контактная информация

1. Группа:

* Номер
* Курс
* Специальность

3. Специальность:

* Название
* Кафедра
* Шифр

4. Кафедра:

* Название
* Факультет
* Заведующий кафедрой

5. Факультет:

* Название
* Декан

6. Дисциплина:

* Название
* Специальность
* Тип оценивания (справочная таблица)

7. Занятие:

* Тип занятия (справочная таблица)
* Дисциплина
* Предователь
* График проведения

8. Оценка:

* Студент
* Оценка
* Описание
* Занятие

9. Пропуск:

* Студент
* Занятие
* Тип пропуска (справочная таблица)

## **Описание интерфейсов**

Клиентское приложение адаптируется под все популярные разрешения.

## **Используемые технологии**

В приложении используются следующие технологии: Spring, Hibernate, Vue.js, JWT, PostgreSQL.

# **Аппаратные требования**

Backend разработан на Java, который является кроссплатформенной средой с открытым исходным кодом для разработки серверных и сетевых приложений.

Frontend будет выстраиваться в браузере, что позволяет не привязываться к определенной платформе.

Программное средство может функционировать на ПЭВМ со следующими минимальными характеристиками:

– процессор с тактовой частотой 2 ГГц и более;

* оперативную память объемом не менее 2 Гб;
* монитор SVGA (800 x 600) или более высоким разрешением.