МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

(ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова)

Кафедра «Программное обеспечение»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной квалификационной работе бакалавра на тему:

«Система управления учебным процессом. Серверная часть»

|  |  |
| --- | --- |
| Дипломник  студент гр. Б08-191-2 | М.Л. Поздеев |
| Руководитель | Р.О. Султанов |
| Нормоконтролер | В.П. Соболева |
| Зав. кафедрой ПО | А.В. Коробейников |

Ижевск 2020

РЕФЕРАТ

Пояснительная записка к выпускной квалификационной работе бакалавра размещена на N листах, содержит в себе N рисунков, N из которых являются блок-схемами алгоритмов.

Ключевые слова: управление учебным процессом, серверная часть, серверное приложение, база данных, клиентское приложение, регистрация, авторизация, аутентификация, сервис авторизации, сервис ресурсов.

Цель работы – разработка системы, призванной повысить эффективность управления учебным процессом для студентов и преподавателей. В частности, требуется разработать серверную часть системы.

В процессе работы было проведено исследование уже существующих систем с похожей целью и выявлены их основные недостатки. Также были изучены инструменты, необходимые для выполнения поставленной задачи.

В результате работы были разработаны база данных системы и серверное приложение, предоставляющее доступ к данным, лежащим на сервере, клиентским приложениям.

Разработанная серверная часть позволяет реализовать в клиентских приложениях необходимые функции. Инструменты, которые использовались в разработке – дают возможность установить базу данных и серверное приложение на всех популярных серверных операционных системах, что позволит сократить затраты на содержание системы и на обучение администраторов.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc43132749)

[1. РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ УЧЕБНЫМ ПРОЦЕССОМ 6](#_Toc43132750)

[1.1. Обоснование целесообразности разработки системы 6](#_Toc43132751)

[1.1.1. Назначение системы 6](#_Toc43132752)

[1.1.2. Обоснование цели системы 6](#_Toc43132753)

[1.1.3. Обоснование состава автоматизируемых задач 6](#_Toc43132754)

[1.2. Аналитический обзор 7](#_Toc43132755)

[1.3. Основные требования к системе 7](#_Toc43132756)

[1.3.1. Основные цели создания системы и критерии эффективности ее функционирования 7](#_Toc43132757)

[1.3.2. Функциональное назначение системы 8](#_Toc43132758)

[1.3.3. Особенности системы, условия эксплуатации, определяющие основные требования к системе 9](#_Toc43132759)

[1.3.4. Требования к функциональной структуре системы 9](#_Toc43132760)

[1.3.5. Типовые проектные решения и (или) пакеты прикладных программ, применяемых в системе 12](#_Toc43132761)

[1.3.6. Требования к техническому обеспечению 13](#_Toc43132762)

[1.3.7. Требования к программному обеспечению 13](#_Toc43132763)

[1.3.8. Перспективность системы, возможности ее развития 13](#_Toc43132764)

[1.4. Основные технические решения проекта системы 13](#_Toc43132765)

[1.4.1. Описание системы программного обеспечения 13](#_Toc43132766)

[2. РАЗРАБОТКА ЗАДАЧ ПО СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ УПРАВЛЕНИЯ УЧЕБНЫМ ПРОЦЕССОМ 15](#_Toc43132767)

[2.1. Разработка сервиса авторизации 15](#_Toc43132768)

[2.1.1. Описание постановки задачи 15](#_Toc43132769)

[2.1.2. Описание алгоритма регистрации пользователей 16](#_Toc43132770)

[2.1.3. Описание алгоритма аутентификации и авторизации пользователей 18](#_Toc43132771)

[2.2. Разработка сервиса ресурсов 22](#_Toc43132772)

[2.2.1. Описание постановки задачи 22](#_Toc43132773)

[2.2.2. Описание алгоритма предоставления данных для приложений преподавателя и студента 25](#_Toc43132774)

[2.3. Описание контрольного примера 30](#_Toc43132775)

[2.3.1. Назначение 30](#_Toc43132776)

[2.3.2. Исходные данные 30](#_Toc43132777)

[2.3.4. Результаты расчета 30](#_Toc43132778)

[2.3.5. Результаты испытания программы 32](#_Toc43132779)

ВВЕДЕНИЕ

В настоящее время почти у каждого человека имеется в наличии мобильное устройство под управлением какой-либо операционной системы. Причина такой популярности мобильных устройств заключается в том, что благодаря им люди могут получать быстрый и удобный доступ к необходимым для них ресурсов в любом месте и в любое время.

В связи с этим разработчики программных продуктов адаптируют свои разработки под мобильные устройства, обеспечивая для своих пользователей удобный доступ к своим ресурсам, а кто-то вообще разрабатывает свои приложения только для мобильных устройств. Не должны оставаться позади и образовательные системы, так как использование в учебном процессе мобильных устройств позволит серьезно увеличить эффективность обучения, а также позволит поднять авторитет учебного заведения среди потенциальных студентов и преподавателей.

Для повышения эффективности обучения, система управления учебным процессом должна предоставлять студентам и преподавателям доступ к своим расписаниям, к своим личным данным, к данным обучаемых студентов или обучающих преподавателей, к новостям и объявлениям, связанным с учебой и к набору изучаемых или преподаваемых предметов с детальной информацией о них.

В некоторых учебных заведениях уже реализованы подобные системы, работающие на мобильных устройствах. Они не реализуют всех вышеперечисленных функций, но их главная проблема в том, что они привязаны к своему учебному заведению, поэтому круг пользователей сильно ограничен, что замедляет развитие таких систем.

Из всего вышесказанного следует, что цель всей работы заключается в разработке системы управления учебным процессом, которая предоставляет мобильный доступ преподавателям и студентам к важным данным в учебном процессе и не привязана к какому-либо одному учебному заведению.

Целью данной работы является разработка серверной части системы. Серверная часть системы должна иметь возможность установки на всех популярных операционных системах, чтобы любое учебное заведение могло без особых проблем развернуть ее на своей инфраструктуре, а также должна обеспечить безопасность хранимых и передаваемых на мобильные приложения данных.

1. РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ УЧЕБНЫМ ПРОЦЕССОМ

1.1. Обоснование целесообразности разработки системы

1.1.1. Назначение системы

Назначение системы заключается в увеличении эффективности ведения учебного процесса в высших учебных заведениях (ВУЗах). Эффективность учебного процесса повышается за счет предоставления пользователям системы (преподавателям и студентам) быстрого и удобного доступа к важной для учебного процесса информации через мобильные приложения.

Система предназначена для работы в большом количестве ВУЗов.

1.1.2. Обоснование цели системы

Так как в настоящее время мобильные устройства обладают огромной популярностью, открываются новые горизонты для увеличения эффективности учебного процесса в ВУЗах. Благодаря мобильным устройствам люди имеют возможность быстро и удобно получать необходимую для них информацию, но в образовательной сфере, качественных инструментов, реализующих полный потенциал мобильных устройств – просто недостаточно. Из этого и следует цель системы – предоставить участникам учебного процесса в ВУЗах универсальное программное решение, которое поможет облегчить многие аспекты в обучении.

От количества пользователей напрямую зависит качество системы и скорость ее улучшения, поэтому система предназначена для работы в разных учебных заведениях. Большое количество пользователей позволит получить огромное количество отзывов и предложений, что поспособствует развитию системы, а также, увеличенный финансовый доход от системы позволит сделать куда более качественное решение, чем у аналогичных разработок, созданных и поддерживаемых на чистом энтузиазме.

1.1.3. Обоснование состава автоматизируемых задач

Текущая работа заключается в разработке серверной части системы. Задача серверной части состоит в том, чтобы предоставить мобильным приложениям единый способ доступа к данным, независимо от университета, в котором система развернута. Серверная часть должна обеспечить безопасное хранение данных, безопасность пользовательских данных (с помощью безопасного хранения паролей и безопасного способа аутентификации и авторизации) и предоставлять данные в удобном для обработки на мобильных приложениях виде.

Из всего вышесказанного следует список задач для разработки серверной части:

* разработка базы данных (БД);
* разработка сервиса авторизации;
* разработка сервиса ресурсов.

1.2. Аналитический обзор

Подобные системы, предоставляющие информацию для пользователей через мобильные приложения, существуют, например, у таких университетов, как УрФУ, ИТМО и БГУ. Однако их главная проблема заключается в том, что они привязаны только к одному учебному заведению. Отсутствие масштабируемости системы серьезно замедляет ее развитие. В ИжГТУ подобный функционал на данный момент выполняют веб-сайты университета istu.ru и ee.istu.ru, их проблема заключается в отсутствии удобного доступа через мобильные устройства и в разделении полезного функционала на несколько ресурсов. Также вышеописанные ресурсы имеют достаточное количество других проблем, но на данный момент разрабатываемая система их не решает.

Главное преимущество разрабатываемой системы в возможности масштабирования на любое количество учебных заведений. Такой возможности не имеет ни одна известная система для ВУЗов. И такую возможность можно реализовать благодаря наличию собственной серверной части системы. Детальное сравнение существующих решений с разрабатываемой системой представлено в таблице 1 (для сравнения выбраны ресурсы ИжГТУ и мобильное приложение ИТМО, так как они наиболее функциональны).

Таблица 1

Сравнение существующих систем с разрабатываемой

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Ресурс istu.ru | Ресурс ee.istu.ru | Мобильное приложение ИТМО | Разрабатываемая система |
| Масштабируемость на любое количество учебных заведений | Отсутствует | Отсутствует | Отсутствует | Имеется |
| Расписание пар | Имеется (неудобно просматривать) | Отсутствует | Имеется | Имеется |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Ресурс istu.ru | Ресурс ee.istu.ru | Мобильное приложение ИТМО | Разрабатываемая система |
| Список изучаемых предметов | Имеется | Имеется (студент должен сам записаться на каждый предмет) | Имеется | Имеется |
| Отдельные функции для преподавателей | Имеются | Имеются | Отсутствуют | Имеются |
| Мобильное приложение | Отсутствует | Отсутствует | Имеется | Имеется |
| Веб версия | Имеется | Имеется | Отсутствует | На данный момент отсутствует |
| Просмотр списка группы студентов | Отсутствует | Имеется (только записанных на курс) | Отсутствует | Имеется |
| Просмотр информации о преподавателях | Имеется | Имеется | Отсутствует | Имеется |
| Просмотр новостей ВУЗа | Имеется | Отсутствует | Отсутствует | Имеется |

В таблице выше перечислены не все возможности и недочеты системы, но перечисленного достаточно для понимания, в какую сторону будет развиваться система, ее архитектура позволяет с легкостью добавлять новый функционал, а установка системы в нескольких учебных заведениях потенциально даст большие финансовые возможности для развития.

В ИжГТУ уже существует серверная инфраструктура, позволяющая университетским ресурсам получать данные, но так как разрабатываемая система должна иметь возможность установки в разных ВУЗах, она должна иметь свою собственную серверную часть. Для обеспечения безопасности данных каждого ВУЗа, разрабатываемая серверная часть должна будет устанавливаться в каждом ВУЗе, который будет использовать систему. Чтобы серверную часть можно было развернуть без технических проблем в любом ВУЗе, она должна иметь возможность разворачивания на любой популярной операционной системе (ОС) и быть максимально дешевой в обслуживании. Поэтому для разработки серверной части были выбраны кроссплатформенные инструменты: СУБД PostgreSQL, платформа разработки общего назначения с открытым кодом .NET Core и язык программирования C#. Выбор пал на эти инструменты, так как они достаточно популярны (не трудно будет найти разработчиков), дешевы в обслуживании и достаточно просты в эксплуатации.

Для разработки API для взаимодействия мобильного приложения с сервером – была выбрана технология платформы .NET Core – ASP.NET Core Web API. Данная технология предоставляет удобные методы для обработки HTTP-запросов и технологию для взаимодействия с СУБД из программного кода – Entity Framework Core. Выбранные технологии позволят без особого труда в дальнейшем разрабатывать новые клиентские приложения, например, для браузеров.

1.3. Основные требования к системе

1.3.1. Основные цели создания системы и критерии эффективности ее функционирования

Основные цели создания системы:

* предоставление важной в учебном процессе информации для преподавателей и студентов через мобильное приложение;
* возможность установки системы на инфраструктуре любого ВУЗа.

В связи с вышеперечисленными целями требуется выполнять следующие требования:

* серверная часть должна иметь возможность установки на все популярные ОС;
* приложение должно иметь раздельный функционал для студента и преподавателя;
* доступ к информации должен быть осуществлен через разрабатываемое API;
* должна иметься возможность переноса в базу данных системы необходимых данных университета, с соблюдением требований информационной безопасности;
* архитектура системы должна быть построена таким образом, функционал системы можно было без труда увеличивать.

1.3.2. Функциональное назначение системы

На данный момент система предоставит для пользователей данные, описанные ниже.

Приложение для студентов:

* расписание занятий группы студента;
* список изучаемых предметов;
* информация об изучаемом предмете;
* нумерованный список студентов группы;
* профиль студента группы;
* новости ВУЗа;
* личный профиль с возможностью редактирования контактной информации.

Диаграмма вариантов использования приложения для студентов представлена на рисунке 1.1.



Рис. 1.1. Диаграмма вариантов использования (студент)

Приложение преподавателя:

* расписание занятий преподавателя;
* список обучаемых групп студентов;
* профиль обучаемого студента;
* список предметов, преподаваемых в группах;
* информация о преподаваемом предмете;
* новости ВУЗа;
* личный профиль с возможностью редактирования контактной информации.

Диаграммы вариантов использования приложения для преподавателей представлена на рисунке 1.2.



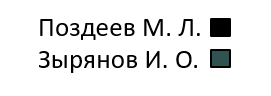
Рис. 1.2. Диаграмма вариантов использования (преподаватель)

1.3.3. Особенности системы, условия эксплуатации, определяющие основные требования к системе

Серверная часть системы будет устанавливаться на инфраструктуре учебного заведения. Способы установки разные в зависимости от операционной системы, поэтому данный вопрос необходимо будет обсуждать дополнительно с каждым ВУЗом-клиентом. В зависимости от потенциального количества пользователей системы в одном ВУЗе – будут рассчитываться требуемые вычислительные мощности для сервера.

1.3.4. Требования к функциональной структуре системы

Структурная схема системы представлена на рисунке 1.3. Разделение частей между разработчиками выделено соответствующим цветом.



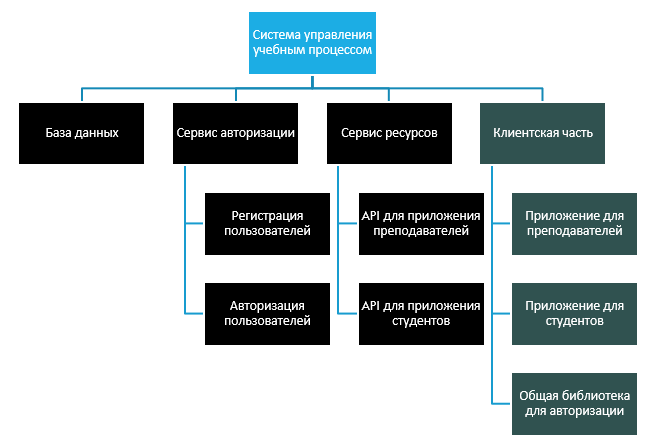


Рис. 1.3. Структурная схема

Часть системы, разрабатываемая в данной работе, состоит из 3 частей:

* база данных. Хранилище всех данных системы, развернутое на сервере ВУЗа;
* сервис авторизации. Сервис, отвечающий за предоставление доступа для пользователей;
* сервис ресурсов. Сервис, отвечающий за обеспечение взаимодействия мобильного приложения и базы данных.

Сервис авторизации состоит из следующих модулей:

* модуль регистрации пользователей. Модуль получает данные регистрируемых пользователей и с соблюдением правил информационной безопасности сохраняет их в БД;
* модуль авторизации. Модуль идентифицирует пользователей и предоставляет им доступ к ресурсам системы.

Сервис ресурсов состоит из следующих модулей:

* модуль API для приложения преподавателей. Модуль предоставляет данные, предназначенные для преподавательского приложения;
* модуль API для приложения студента. Модуль предоставляет данные, предназначенные для студенческого приложения.

1.3.5. Типовые проектные решения и (или) пакеты прикладных программ, применяемых в системе

Для разработки сервиса авторизации и сервиса ресурсов используется платформа разработки общего назначения с открытым кодом .NET Core и язык программирования C#, как удобное кроссплатформенное решение. Для базы данных используется СУБД PostgreSQL.

Применяемые средства разработки: среда разработки .NET Core – Visual Studio 2019, среда разработки СУБД PostgreSQL – dbForge for PostgreSQL Express.

1.3.6. Требования к техническому обеспечению

Минимальные требования к техническому обеспечению сервера:

* процессор: Intel Pentium 4, 2 ГГц;
* размер оперативной памяти: 1 Гб;
* объем жесткого диска: 1 Гб.

1.3.7. Требования к информационному обеспечению

Информационное обеспечение системы составляет СУБД PostgreSQL.

1.3.8. Требования к программному обеспечению

На сервере должно быть установлено следующее программное обеспечение:

* .NET Core версии 3.1;
* веб-сервер Nginx версии 1.17.10;
* PostgreSQL версии 12.

1.3.9. Перспективность системы, возможности ее развития

В перспективе система может дополняться различным функционалом, таким как чат с преподавателем, генератор расписания, веб-версия приложения, iOS-версия приложения. Также, с учетом некоторых особенностей, систему можно будет масштабировать на несколько ВУЗов.

1.4. Основные технические решения проекта системы

1.4.1. Описание организации информационной базы

Таблица staff содержит информацию о сотрудниках ВУЗа:

* id – внутренний идентификатор;
* name – имя;
* surname – фамилия;
* patronymic – отчество.

Таблица staff\_requisites содержит детальную информацию о сотрудниках ВУЗа:

* staff\_id – внешний идентификатор сотрудника;
* e\_mail – почтовый адрес;
* mobile\_phone – номер мобильного телефона;
* home\_phone – номер домашнего телефона.

Таблица institutes содержит информацию об институтах и факультетах ВУЗа:

* id – внутренний идентификатор;
* name – наименование;
* short\_name – краткое наименование;
* director\_id – внешний идентификатор сотрудника руководителя (директора или ректора);
* director\_deputy\_id – внешний идентификатор сотрудника заместителя руководителя.

Таблица departments содержит информацию о кафедрах:

* id – внутренний идентификатор;
* institute\_id – внешний идентификатор института или факультета;
* name – наименование;
* short\_name – краткое наименование;
* director\_id – внешний идентификатор сотрудника руководителя.

Таблица staff\_departments содержит связи между сотрудниками ВУЗа и кафедрами:

* department\_id – внешний идентификатор кафедры;
* staff\_id – внешний идентификатор сотрудника.

Таблица subject\_information содержит информацию о предметах:

* id – внутренний идентификатор;
* name – наименование;
* short\_name – краткое наименование;
* description – описание.

Таблица subjects содержит связи между сотрудниками (преподавателями) и информацией о предметах:

* id – внутренний идентификатор;
* lecturer\_id – внешний идентификатор сотрудника преподавателя;
* information\_id – внешний идентификатор информации о предмете.

1.4.1. Описание системы программного обеспечения

Для реализации и нормального функционирования проекта необходимо наличие:

* смартфона под управлением операционной системы Android 4.4 или выше;
* компьютер под управлением операционной системы Windows или Linux.

Для разработки данной системы были использованы:

* Microsoft Visual Studio 2019;
* Android Studio;
* dbForge for PostgreSQL.

2. РАЗРАБОТКА ЗАДАЧ ПО СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ УПРАВЛЕНИЯ УЧЕБНЫМ ПРОЦЕССОМ

2.1. Разработка сервиса авторизации

2.1.1. Описание постановки задачи

2.1.1.1. Характеристика задачи

Задача – разработать сервис авторизации.

Сервис авторизации разрабатывается с целью обеспечения безопасности данных, хранимых в системе. Для реализации возможности масштабирования системы в будущем на большое количество университетов – было решено разработать собственный сервис авторизации, который будет предоставлять доступ мобильным приложениям пользователей к API системы.

Разработка сервиса авторизации связана с разработкой сервиса ресурсов системы. С помощью проверки доступа к методам API сервиса ресурсов – будет проверяться работоспособность сервера авторизации.

Разработка задачи будет завершена при успешном прохождении тестирования сервера.

2.1.1.2. Входная информация

Перечень входных данных при регистрации пользователей:

* номер зачетной книжки (в случае, если пользователь – студент);
* идентификатор преподавателя в системе (в случае, если пользователь – преподаватель);
* уникальный логин пользователя;
* пароль.

Перечень входных данных при аутентификации и авторизации пользователей:

* логин пользователя, созданный при регистрации;
* пароль пользователя, созданный при регистрации.

Перечень входных данных при валидации токена доступа:

* токен доступа.

Перечень входных данных при обновлении токена доступа:

* токен восстановления доступа;
* старый токен доступа.

Вся вышеперечисленная информация передается на сервер с помощью http запросов.

2.1.1.3. Выходная информация

Перечень выходных данных при регистрации пользователей:

* сообщение об успешной или о неудачной регистрации.

Перечень выходных данных при аутентификации и авторизации пользователей:

* токен доступа;
* токен восстановления доступа;
* идентификатор пользователя в системе.

Перечень выходных данных при валидации токена доступа:

* сообщение об успешной или неудачной валидации.

Перечень выходных данных при обновлении токена доступа:

* новый токен доступа при успешном обновлении токена;
* новый токен восстановления доступа при успешном обновлении токена;
* идентификатор пользователя в системе при успешном обновлении токена;
* ошибка обновления токена при не правильных входных данных.

Вся вышеперечисленная информация выдается сервером в ответ на http запросы.

2.1.2. Описание алгоритма регистрации пользователей

2.1.2.1. Назначение и характеристика

Регистрация необходима для создания пользователя в системе и его сопоставления с сущностью студента или преподавателя в базе данных.

Алгоритм должен сохранять данные пользователей в базе данных, при этом проверив существование этого пользователя как студента или преподавателя в системе. Пароли пользователей должны записываться в базу в виде хеша таким образом, чтобы одинаковые пароли имели разное представление в базе данных. Полученный хеш должен быть устойчивым к атакам перебора паролей.

2.1.2.2. Используемая информация

Для алгоритма регистрации используются:

1. номер зачетной книжки студента или идентификатор преподавателя в системе;
2. уникальный логин пользователя;
3. пароль пользователя.

2.1.2.3. Результаты решения

В результате выполнения алгоритма формируется следующая информация:

* оповещение клиентского приложения об успешной или неудачной регистрации;
* сохранение регистрационных данных пользователя (логин, пароль в виде хеша, идентификатор пользователя) в базе данных.

2.1.2.4. Алгоритм решения

Схема алгоритма представлена на рисунке 2.1:



Рис. 2.1

Описание алгоритма:

1. сервер авторизации получает регистрационные данные от клиентского приложения;
2. сервер авторизации сопоставляет полученный номер зачетной книжки или идентификатор преподавателя с данными в базе;
3. если такого студента или преподавателя не существует, то сервер посылает на клиентское приложение ответ с сообщением об ошибке регистрации и алгоритм завершается;
4. если такой студент или преподаватель существует в университете, то алгоритм выполняется дальше;
5. сервер авторизации генерирует «соль» для хеширования пароля – случайный текст, который добавится к введенному паролю для обеспечения уникальности хешей при введении пользователями одинаковых паролей;
6. сервер генерирует хеш пароля и добавлением «соли»;
7. сервер сохраняет регистрационные данные в базу данных пользователей системы;
8. сервер посылает на клиент ответ с сообщением об успешной регистрации и алгоритм завершается.

2.1.3. Описание алгоритма аутентификации и авторизации пользователей

2.1.3.1. Назначение и характеристика

Аутентификация необходима для подтверждения идентичности пользователя. Авторизация необходима для предоставления пользователю прав доступа на необходимые ресурсы.

Алгоритм должен сверять данные пользователя (логин и пароль), пришедшие с клиентского приложения с данными в базе данных, и если все указано верно, то алгоритм должен предоставить доступ пользователю клиентского приложения с помощью токена доступа к необходимым для него ресурсам. Время действия токена должно быть ограничено, для обеспечения безопасности. Алгоритм также должен предоставить метод обновления токена доступа с помощью токена восстановления доступа, для того чтобы токен доступа можно было восстановить без введения логина и пароля пользователя. В добавок к токенам, алгоритм должен отправлять на клиентское приложение идентификатор пользователя в системе.

2.1.3.2. Используемая информация

При первой аутентификации используются:

* логин пользователя;
* пароль пользователя.

При наличии токена доступа для аутентификации и авторизации используется этот токен доступа.

При обновлении токена доступа используются:

* токен доступа;
* токен восстановления доступа.

2.1.3.3. Результаты решения

В результате выполнения алгоритма при первой аутентификации и при обновлении токена доступа формируется следующая информация:

* токен доступа;
* токен восстановления доступа;
* идентификатор пользователя в системе.

Токен восстановления доступа сохраняется в базе данных пользователей. Все сформированные данные отправляются на клиентское приложение.

При наличии актуального токена – пользователю предоставляется доступ к необходимым ресурсам.

2.1.3.4. Алгоритм решения

Схема алгоритма при наличии токена доступа не представлена, этот вариант рассмотрен в текстовом описании алгоритма. Схемы алгоритма при первой аутентификации и при обновлении токена доступа представлены на рисунках 2.2 и 2.3, соответственно.

|  |  |
| --- | --- |
|  |  |
| Рис. 2.2 | Рис. 2.3 |

Описание алгоритма при первой аутентификации пользователя:

1. сервер получает данные для аутентификации с клиентского приложения (логин и пароль);
2. сервер сверяет введенные данные с данными из БД, пароль сверяется с помощью вычисление хеша как в алгоритме регистрации пользователей;
3. если данные введены неверно, то на клиентское приложение возвращается ошибка;
4. если данные введены верно, то сервер генерирует токен доступа и токен восстановления доступа;
5. сервер сохраняет токен восстановления доступа в базе данных;
6. сервер отправляет на клиент токен доступа, токен восстановления доступа и идентификатор пользователя в системе.

Описание алгоритма при наличии токена доступа:

1. сервер получает токен доступа;
2. сервер проверяет валидность полученного токена доступа;
3. если токен не валиден, то на клиентское приложение отправляется ошибка авторизации с информацией, показывающей, почему токен не валиден (если время действия токена истекло);
4. если токен валиден, то сервер предоставляет пользователю доступ к запрашиваемому ресурсу.

Описание алгоритма при восстановлении токена доступа:

1. сервер получает токен доступа и токен восстановления доступа;
2. сервер проверяет валидность токена доступа (опуская проверку времени действия токена);
3. если токен доступа не валиден, то сервер отправляет на клиентское приложение ошибку с сообщением о неправильном токене доступа;
4. сервер проверяет валидность токена восстановления доступа;
5. если токен восстановления доступа не валиден, то сервер отправляет на клиентское приложение ошибку с сообщением о неправильном токене восстановления доступа;
6. если полученные токены валидны, то сервер генерирует токен доступа и токен восстановления доступа;
7. сервер сохраняет токен восстановления доступа в базе данных;
8. сервер отправляет на клиент токен доступа, токен восстановления доступа и идентификатор пользователя в системе.

2.2. Разработка сервиса ресурсов

2.2.1. Описание постановки задачи

2.2.1.1. Характеристика задачи

Задача – разработка сервиса ресурсов.

Сервис ресурсов разрабатывается с целью предоставления клиентским приложениям для студентов и преподавателей интерфейса взаимодействия с базой данных системы. Для реализации возможности масштабирования системы в будущем на большое количество университетов – было решено спроектировать собственную базу данных и разработать сервис ресурсов, который будет предоставлять доступ (API) мобильным приложениям пользователей к данным системы.

Разработка сервиса ресурсов связана с разработкой сервиса авторизации, так как через сервис авторизации пользователям будет предоставляться доступ к методам сервера ресурсов.

Разработка задачи будет завершена при успешном прохождении тестирования сервера.

2.2.1.2. Входная информация

Вся входная информация приходит в составе http запросов.

2.2.1.2.1. Входная информация студенческого приложения

Входная информация при запросе расписания:

* идентификатор группы студента пользователя;
* дата, за которую необходимо выдать расписание.

Входная информация при запросе информации о предмете:

* идентификатор предмета.

Входная информация при запросе списка студентов группы:

* идентификатор группы студента пользователя.

Входная информация при запросе списка изучаемых предметов группы:

* идентификатор студента пользователя.

Входная информация при запросе информации о преподавателе:

* идентификатор преподавателя.

Входная информация при запросе личного профиля:

* идентификатор студента пользователя.

Входная информация при редактировании личного профиля:

* идентификатор пользователя студента;
* редактируемая информация (почтовый адрес или номер телефона).

Входная информация при запросе новостей и объявлений:

* идентификатор группы студента пользователя.

Входная информация при опубликовании объявления:

* идентификатор пользователя студента.

2.2.1.2.2. Входная информация преподавательского приложения

Входная информация при запросе расписания:

* идентификатор преподавателя пользователя;
* дата, за которую необходимо выдать расписание.

Входная информация при запросе информации о предмете:

* идентификатор предмета.

Входная информация при запросе списка обучаемых групп:

* идентификатор преподавателя пользователя.

Входная информация при запросе списка студентов одной из обучаемых групп:

* идентификатор группы студентов.

Входная информация при запросе списка преподаваемых предметов в группе:

* идентификатор преподавателя пользователя;
* идентификатор группы студентов.

Входная информация при запросе информации о студенте:

* идентификатор студента.

Входная информация при запросе личного профиля:

* идентификатор преподавателя пользователя.

Входная информация при редактировании личного профиля:

* идентификатор пользователя преподавателя;
* редактируемая информация (почтовый адрес или номер телефона).

Входная информация при запросе новостей и объявлений:

* идентификатор преподавателя пользователя.

Входная информация при опубликовании объявления:

* идентификатор пользователя преподавателя.

2.2.1.3. Выходная информация

Вся выходная информация отправляется в качестве ответа на http запрос. Для получения все данных – необходим токе доступа, который формируется сервисом авторизации.

2.2.1.3.1. Выходная информация студенческого приложения

Выходная информация при запросе расписания:

* расписание на запрошенный день в виде списка предметов с датами.

Выходная информация при запросе информации о предмете:

* данные по предмету (описание, преподаватель).

Выходная информация при запросе списка студентов группы:

* список студентов группы.

Выходная информация при запросе списка изучаемых предметов группы:

* список изучаемых предметов группы.

Выходная информация при запросе информации о преподавателе:

* информация о преподавателе.

Выходная информация при запросе личного профиля:

* информация из личного профиля студента пользователя.

Выходная информация при редактировании личного профиля:

* сообщение об успешном редактировании.

Выходная информация при запросе новостей и объявлений:

* список новостей и объявлений.

Выходная информация при опубликовании объявления:

* сообщение об успешной публикации.

2.2.1.3.2. Выходная информация преподавательского приложения

Выходная информация при запросе расписания:

* расписание в виде списка предметов с датами.

Выходная информация при запросе информации о предмете:

* данные по предмету (описание, преподаватель).

Выходная информация при запросе списка обучаемых групп:

* список обучаемых групп.

Выходная информация при запросе списка студентов одной из обучаемых групп:

* список студентов группы.

Выходная информация при запросе списка преподаваемых предметов в группе:

* список преподаваемых предметов в группе.

Выходная информация при запросе данных о студенте:

* данные о студенте.

Выходная информация при запросе личного профиля:

* информация из личного профиля преподавателя.

Выходная информация при редактировании личного профиля:

* сообщение об успешном редактировании.

Выходная информация при запросе новостей и объявлений:

* список новостей и объявлений.

Выходная информация при опубликовании объявления:

* сообщение об успешной публикации.

2.2.2. Описание алгоритма предоставления данных для приложений преподавателя и студента

2.2.2.1. Назначение и характеристика

Алгоритм предоставления данных для приложений преподавателя и студента необходим для сбора и формирования в удобный вид данных из БД, с последующей отправкой их на клиентское приложение.

Алгоритм должен вытаскивать данные для приложения преподавателя из БД, при этом формируя их в тот вид, в котором они необходимы в клиентском приложении, чтобы как можно больше уменьшить количество бизнес-логики на клиенте, что поспособствует увеличению производительности. Алгоритм должен предоставлять доступ к своим функциям только при наличии валидного токена доступа, который можно получить, используя сервис авторизации.

2.2.2.2. Используемая информация

Для авторизации пользователя используется токен доступа.

Информация, которая используется только для приложения преподавателя или только для приложения студента – подписана соответствующим образом. Информация, используемая в обоих случаях – не подписана.

При выполнении запроса на выдачу расписания используется следующая информация:

* идентификатор преподавателя-пользователя (приложение преподавателя);
* идентификатор группы студента-пользователя (приложение студента)
* данные таблицы расписания занятий из БД;
* данные таблицы студенческих групп из БД (приложение преподавателя);
* данные таблицы преподавателей из БД (приложение студента);
* данные таблицы предметов из БД.

При выполнении запроса на выдачу данных об определенном предмете используется следующая информация:

* идентификатор этого предмета;
* данные таблицы предметов из БД.

При выполнении запроса на выдачу списка обучаемых преподавателем студенческих групп используется следующая информация (приложение преподавателя):

* идентификатор преподавателя, выполняющего запрос;
* данные таблицы, связывающей студенческие группы и преподавателей из БД.

При выполнении запроса на выдачу списка студентов одной из групп используется следующая информация:

* идентификатор требуемой группы;
* данные таблицы студенческих групп из БД.

При выполнении запроса на выдачу списка преподаваемых предметов в определенной группе используется следующая информация (приложение преподавателя):

* идентификатор группы;
* идентификатор преподавателя, выполняющего запрос;
* данные таблицы предметов из БД;
* данные таблицы связывающей предметы и группы из БД.

При выполнении запроса на выдачу данных о студенте используется следующая информация:

* идентификатор требуемого студента;
* данные таблицы реквизитов студентов из БД.

При выполнении запроса на выдачу данных о преподавателе используется следующая информация:

* идентификатор требуемого преподавателя;
* данные таблицы реквизитов сотрудников университета из БД.

При выполнении запроса на редактирование данных из личного профиля используется следующая информация:

* идентификатор преподавателя-пользователя (приложение преподавателя);
* идентификатор студента-пользователя (приложение студента);
* редактируемые данные;
* данные таблицы реквизитов сотрудников университета из БД (приложение преподавателя);
* данные из таблицы реквизитов студентов (приложение студента);

При выполнении запроса на выдачу новостей и объявлений используется следующая информация:

* идентификатор преподавателя-пользователя (приложение преподавателя);
* идентификатор студента-пользователя (приложение студента);
* данные таблицы новостей из БД;
* данные таблицы пользовательских объявлений из БД.

При выполнении запроса на публикацию объявления используется следующая информация:

* идентификатор преподавателя-пользователя (приложение преподавателя);
* идентификатор студента-пользователя (приложение студента);
* идентификаторы групп, для которых публикуется объявление (приложение преподавателя);
* текст объявления;
* данные таблицы пользовательских объявлений из БД.

При выполнении запроса на выдачу списка изучаемых студентом предметов используется следующая информация (приложение студента):

* идентификатор студента-пользователя.

2.2.2.3. Результаты решения

Информация, которая формируется только для приложения преподавателя или только для приложения студента – подписана соответствующим образом. Информация, формируемая в обоих случаях – не подписана.

В результате выполнения запроса на выдачу расписания формируется список предметов. Объект предмета для расписания включает в себя:

* идентификатор предмета;
* дата и время проведения пары;
* наименование студенческой группы, в которой будет проводиться пара (приложение преподавателя);
* ФИО преподавателя (приложение студента).

В результате выполнения запроса на выдачу данных об определенном предмете формируется объект предмета, в составе которого:

* идентификатор предмета;
* описание предмета;
* идентификатор преподавателя (приложение студента);
* ФИО преподавателя (приложение студента).

В результате выполнения запроса на выдачу списка обучаемых преподавателем студенческих групп формируется список групп (приложение преподавателя). Объект группы включает в себя:

* идентификатор группы;
* наименование группы.

В результате выполнения запроса на выдачу списка студентов одной из групп, формируется список студентов. Объект студента включает в себя:

* идентификатор студента;
* ФИО студента.

В результате выполнения запроса на выдачу данных о студенте формируется объект студента, в составе которого:

* идентификатор студента;
* ФИО студента;
* почтовый адрес студента;
* номер телефона студента;
* фото студента.

В результате выполнения запроса на выдачу данных о преподавателе формируется объект преподавателя, в составе которого:

* идентификатор преподавателя;
* ФИО преподавателя;
* номер телефона преподавателя;
* текущий статус преподавателя;
* должность преподавателя.

В результате запроса на редактирование данных из личного профиля пользователя формируется сообщение об успешном редактировании.

В результате выполнения запроса на публикацию объявления формируется сообщение об успешной публикации.

2.2.2.4. Алгоритм решения

Схема алгоритма представлена на рисунке 2.4.



Рис. 2.4

Описание алгоритма:

1. при поступлении запроса происходит определение метода, который его выполнит (маршрутизация запроса);
2. на данный момент ко всем методам сервиса ресурсов отсутствует анонимный доступ, поэтому перед выполнением запроса происходит валидация токена доступа пользователя с помощью сервиса авторизации;
3. если токен доступа пользователя не валиден, то сервис ресурсов отказывает пользователю в доступе;
4. если токен валиден, то сервис ресурсов выполняет необходимый запрос.

2.3. Описание контрольного примера

2.3.1. Назначение

Контрольным примером проверяются:

* регистрация пользователя в системе;
* занесение нового пользователя в базу;
* хеширование пароля.
* выдача токена доступа и токена восстановления доступа пользователю;
* выдача токенов при первой аутентификации;
* выдача токенов при восстановлении доступа;
* сохранение токена восстановления в базе.
* доступ к методам сервиса ресурсов;
* доступ при валидном токене доступа;
* доступ при не валидном токене доступа;
* работа метода по предоставлению данных о студенте.

2.3.2. Исходные данные

Перечень исходных данных:

* запись в базе о группе Б08-191-2 c идентификатором 1;
* запись в базе о студенте Иванове Иване Ивановиче с номером зачетной книжки 123 и идентификатором группы 1, номером мобильного телефона 12345, номером домашнего телефона 12345;
* логин, который будет использоваться для регистрации этого студента в системе IvanovII;
* пароль, который будет использоваться для регистрации этого студента в системе qwerty.

2.3.4. Результаты расчета

Результаты проверки регистрации пользователя представлены на рисунках 5 и 6.

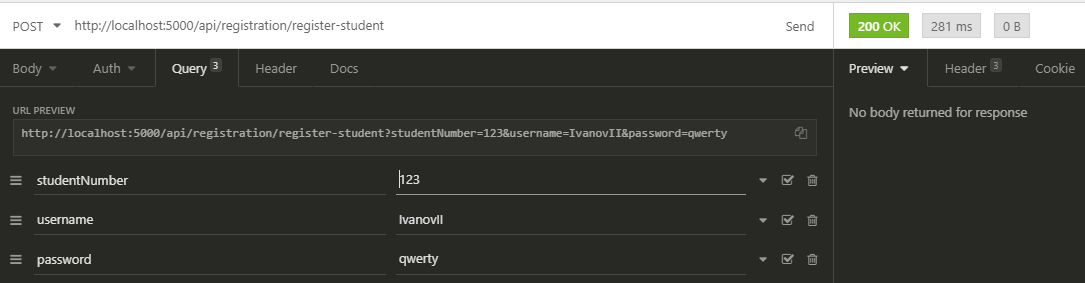


Рис. 5. Пример работы регистрации



Рис. 6. Пример занесения зарегистрированного пользователя в базу

Результаты проверки выдачи токена доступа пользователю представлены на рисунках 7, 8 и 9.

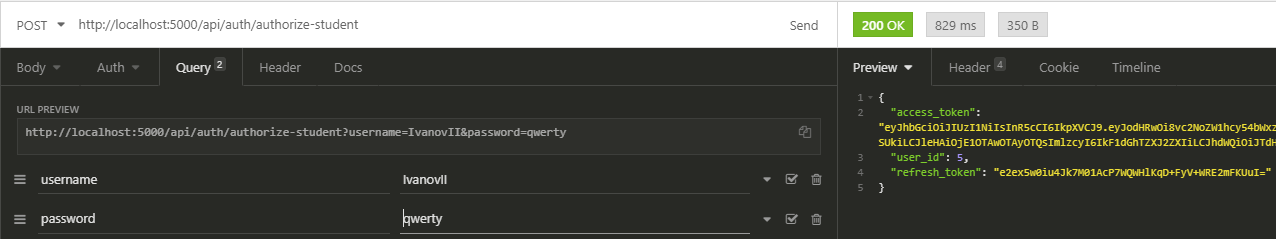


Рис. 7. Пример первой аутентификации

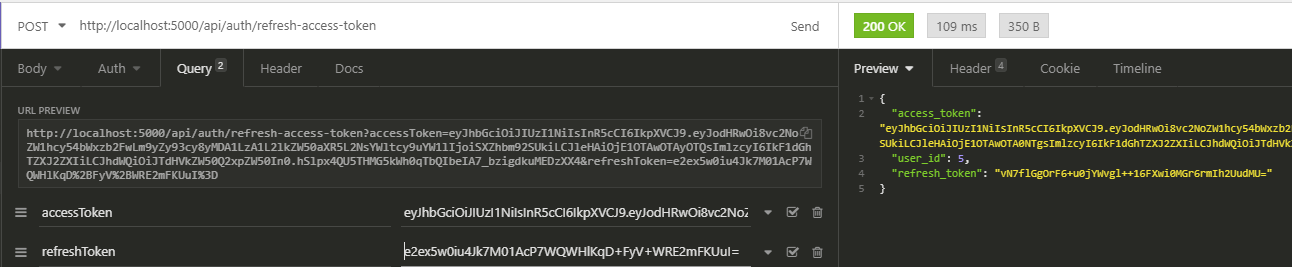


Рис. 8. Пример восстановления токена доступа



Рис. 9. Пример сохранения токена восстановления в базе

Результаты проверки доступа к методам сервиса ресурсов представлены на рисунках 10, 11, 12.

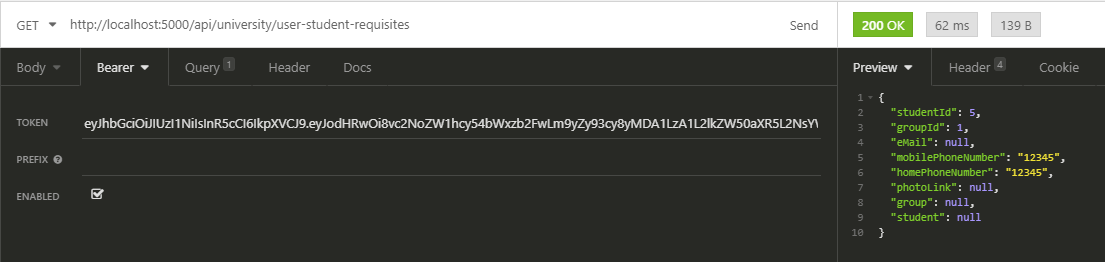


Рис. 10. Пример доступа к методу предоставления данных о студенте

с помощью валидного токена доступа

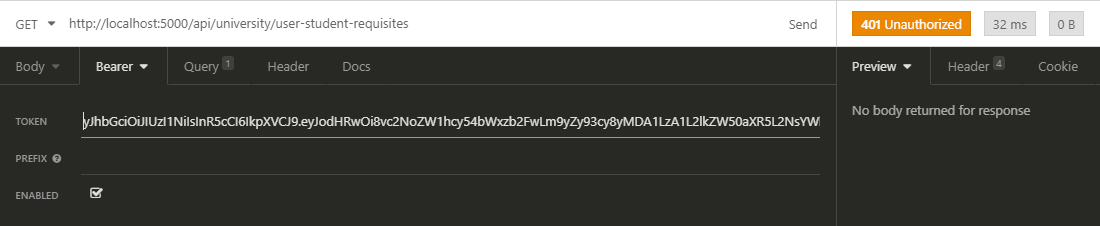


Рис. 11. Пример доступа к методу предоставления данных о студенте

с помощью не валидного токена доступа

2.3.5. Результаты испытания программы

Для проверки работы сервисов использовались: программа для тестирования API – Insomnia, среда разработки СУБД PostgreSQL – dbForge for PostgreSQL, среда разработки платформы .NET Core – Visual Studio 2019.

По результатам проверки удалось выяснить, что разработанная часть системы функционирует безошибочно.

ЗАКЛЮЧЕНИЕ

В результате данной работы была разработана серверная часть системы управления учебным процессом. А именно, были созданы и протестированы: база данных системы, серверное приложение, включающее в себя сервис ресурсов и сервис авторизации.

На данный момент серверная часть имеет следующие возможности:

* хранение необходимых в учебном процессе данных;
* безопасное предоставление мобильному приложению необходимой в учебном процессе информации;
* регистрация пользователей;
* авторизация и аутентификация пользователей;
* защита паролей пользователей.

Использованные в разработке технологии являются широко используемыми среди программистов, поэтому для увеличения функционала и качества системы будет легко найти новых разработчиков. Реализованная архитектура дает возможность разработки разных типов клиентских приложений, что открывает большие возможности для развития и внедрения системы в учебные заведения.

ПРИЛОЖЕНИЕ 1

КОД ПРОГРАММЫ