

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра Прикладная математика и информатика
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки / специальности)

Мобильные и сетевые технологии
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка ПО для учета посещаемости занятий студентами и преподавателями
на основе событий системы контроля и управления доступом»

Обучающийся	<u>П.А. Безроднов</u> (Инициалы Фамилия)	<u>_____</u> (личная подпись)
Руководитель	<u>канд. пед. наук, доцент, Т.А. Агошкова</u> (ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)	
Консультант	<u>А.В. Москалюк</u> (ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)	

Тольятти 2023

Аннотация

Тема: «Разработка ПО для учета посещаемости занятий студентами и преподавателями на основе событий системы контроля и управления доступом».

Ключевые слова: система контроля и управления доступом (СКУД), система учета посещаемости, язык программирования, фреймворк, интегрированная среда разработки, веб-приложение, пользовательский интерфейс.

Объект исследования – системы контроля и управления доступом, различные системы учета посещаемости занятий в университете.

Материал исследования – электронные ресурсы, содержащие информацию о компонентах СКУД, аналогах систем учета посещаемости, архитектуре разрабатываемого программного обеспечения.

Выпускная квалификационная работа состоит из 50 листов, включает в себя 1 таблицу, 24 рисунка, 20 литературных источников.

Abstract

The purpose of the final qualifying work is to develop software for recording attendance of classes by students and teachers based on the events of the access control and management system.

The problem of attendance control is one of the important problems of the organization of the educational process. But it can be optimized using information technology.

In the course of the work, the following tasks were solved:

- studying the concept of the attendance control system (ACS) and its components;
- disassembling existing ACS analogues;
- developing UML diagrams of the dialog and the user's work with the attendance tracking system;
- developing the software.

The result of the final qualification work was the development of software for the attendance control system for students and teachers based on the access control and management system in the form of a web application. This application has a friendly interface and functionality. For software development, the multi-paradigm programming language JavaScript was chosen, which is used to create browser applications. The created software is designed to work on a personal computer running the Windows 10 operating system. The interface of the software shell is a standard GUI interface in the style of a multi-document application, when the main form is the launch form, and the child forms are opened within it.

Содержание

Введение.....	5
1 Аналитический обзор СКУД и систем учета посещаемости	7
1.1 Понятие и разновидности СКУД.....	7
1.2 Компоненты СКУД	9
1.3 Системы учета посещаемости занятий студентами и преподавателями	20
2 Разработка программного обеспечения для учета посещаемости занятий студентами и преподавателями	24
2.1 Моделирование ПО для учета посещаемости занятий студентами и преподавателями на основе событий системы контроля и управления доступом.....	24
2.2 Выбор средств реализации программного обеспечения.....	29
3 Реализация и тестирование программного обеспечения	40
3.1 Контрольный пример разработанного ПО	40
3.2 Тестирование программного обеспечения	44
Заключение	47
Список используемой литературы	49

Введение

Проблема контроля за посещаемостью занятий является одной из важных проблем организации учебного процесса. Ее можно оптимизировать, используя информационные технологии. На данный момент в преобладающем количестве университетов учет посещаемости ведется на бумажных носителях. Однако уже появляются программные продукты, реализующие автоматизированный учет и анализ журнала посещаемости, но пока они ориентированы только на крупные учебные заведения.

Целью выпускной квалификационной работы является разработка программного обеспечения для учета посещаемости занятий студентами и преподавателями на основе событий системы контроля и управления доступом.

Для решения этого вопроса необходимо решить следующие задачи:

- изучить понятие СКУД и компоненты СКУД;
- разобрать существующие аналоги систем учета посещаемости;
- разработать UML-диаграммы диалога и работы пользователя с системой учета посещаемости;
- разработать программное обеспечение.

Данная выпускная квалификационная работа состоит из введения, основной части и заключения.

Во введении представлены цели и задачи выпускной квалификационной работы.

Основная часть состоит из трёх разделов. В первом разделе рассказывается, что такое система контроля и управления доступом, обозреваются компоненты СКУД. Также в этой главе описываются существующие системы учета посещаемости занятий в университетах.

Во втором разделе описывается работа разрабатываемой ПО в виде UML-диаграмм, обозреваются различные технологии разработки программ и

на основе их плюсов и минусов выбираются те, что будут использоваться в разработке программного обеспечения.

В третьем разделе показывается реализация разработанного программного обеспечения.

В заключении указываются выводы, полученные из исследования СКУД и систем учета посещаемости занятий и разработки программного обеспечения.

1 Аналитический обзор СКУД и систем учета посещаемости

1.1 Понятие и разновидности СКУД

Система контроля и управления доступом (СКУД) – это тип физической безопасности, предназначенный для ограничения или разрешения доступа к определенной области или зданию. Часто СКУД устанавливаются для защиты предприятий и имущества от вандализма, краж и вторжений, и особенно полезны на объектах, требующих более высокий уровень безопасности и защиту. В отличие от физических барьеров, СКУД контролирует, кто, как и когда может получить доступ [20].

На рисунке 1 представлен один из видов системы контроля и управления доступом.



Рисунок 1 – Распознавательный турникет, работающий с использованием системы контроля и управления доступом

Комплекс оборудования для управления доступом в сочетании с ПО может быть использован для выполнения следующих задач:

- защита информации, составляющей государственную или коммерческую тайну;
- защита имущества и материальных ценностей;
- контроль количества людей, одновременно занимающих помещение;
- контроль входа в охраняемые зоны лиц, целью которых является нарушение нормальной работы организации, путем саботажа или промышленного шпионажа;
- организация пропускного режима, запрет доступа посторонних лиц в охраняемые зоны;
- учет рабочего времени персонала и фиксация опозданий, ранних уходов, прогулов и сверхурочной работы;
- учет транспортных средств, которые въезжают на территорию объекта и выезжают с нее [12].

На данный момент существует четыре категории СКУД:

- СКУД I класса выделяется автономным режимом и небольшим набором функций. Данная СКУД обеспечивает проход пользователей, данные которых есть в базе данных;
- СКУД II класса характеризуется работой в автономном и сетевом режимах, настройкой прохода пользователей по идентификатору/заданному времени;
- СКУД III и IV классов являются сетевыми системами, характеризующимися сложными идентификаторами, многоуровневым взаимодействием и объемной комплектацией;

Системы контроля и управления доступом обладают как рядом преимуществ, так и рядом недостатков.

К основным плюсам системы контроля и управления доступом относятся:

- безопасность объекта: СКУД может быть оснащена системой видеонаблюдения и может осуществлять круглосуточный мониторинг. Все действия будут зафиксированы системой видеонаблюдения;

- автоматизация доступа: система турникетов и точек доступа, проход через которые обеспечивается с использованием индивидуальных карт. Система позволяет ограничить доступ в определенные помещения сотрудникам, и наоборот. Попасть на объект или покинуть его возможно только с помощью карты;

- возможность интеграции: СКУД можно интегрировать с программами, которые используют HR-специалисты и бухгалтерия. Это позволит автоматизировать отчетность и учет кадров. Приложив карту на пропускном пункте, сотрудник автоматически отмечается в таблице учета рабочего времени [11].

В свою очередь, к недостаткам и проблемам систем контроля и управления доступом можно отнести:

- уязвимость заключается в том, что невозможно обеспечить полную защиту данных.

- стоимость: на крупных предприятиях, таких как заводы, цена может составлять 1 миллион рублей, В офисах эта система контроля может быть очень дорогой;

- отсутствие обеспечения контроля над сотрудниками и гарантии выполнения ими своих обязанностей;

- неспособность создать благоприятные условия для взаимодействия между членами компании.

Рассмотрим компоненты СКУД.

1.2 Компоненты СКУД

Для того, чтобы система контроля и управления доступом функционировала, необходимы следующие компоненты:

- специальные датчики, определяющие условия и события;
- механизмы исполнения;
- электронное устройство, которое задает алгоритм, т.е. контроллер или ПК (персональный компьютер).

Рассмотрим каждый из них.

Одной из главных задач СКУД является идентификация пользователя для применения к нему ограничений. Одним из комплектующих СКУД являются датчики – считыватели. Они делятся на два типа: считыватели электронных кодов и сканеры биометрических данных.

Считыватели кодов необходимы для получения от сотрудника кода (идентификатора), который передается с помощью аппаратных идентификаторов. Самые распространенные считыватели кодов – это оптические, магнитные, бесконтактные, контактные, считыватели виртуальных данных.

Оптические считыватели работают с информацией, хранящейся на жестких носителях в виде определенных символов, например штрих-код на ID-карте. У таких считывателей СКУД высокая степень защиты. Оптические считыватели могут сразу же распознать подделку, поскольку оригинальную комбинацию штрих-кодов трудно воспроизвести без специального оборудования. Однако такие считыватели очень дороги, поскольку аппараты, способные распознавать текстовые коды, стоят очень дорого.

Плюсы считывателей ID-карт:

- безопасность: считыватели идентификационных карт могут проверять личность и разрешать доступ к определенным ресурсам, тем самым обеспечивая безопасность информации, содержащейся на идентификационной карте;
- эффективность: использование считывателей ID-карт упрощает процесс идентификации и авторизации и сокращает время, необходимое для получения доступа к определенным ресурсам;

- удобство: считыватели ID-карт, как правило, компактны и могут быть легко переносимы, что делает их удобными в использовании.

Минусы считывателей ID-карт:

- цена: считыватели ID-карт могут быть достаточно дорогими, особенно при использовании в большом количестве;

- технические проблемы: считыватели ID-карт могут быть уязвимыми к техническим проблемам, таким как поломки или проблемы с подключением к сети;

- неполадки: если считыватель ID-карт не может правильно подтвердить идентификационные данные, это может привести к ошибкам в авторизации доступа или даже отказу в доступе.

На рисунке 2 представлен оптический считыватель СКУД.



Рисунок 2 – Считыватель ID-карт

Магнитные считыватели были наиболее распространенными устройствами для распознавания кодов, поскольку магнитные карты быстро изготавливались и на них можно было легко записать закодированную информацию. Однако, из-за массового производства магнитных карточек, устройств считывания, этот метод считывания перестал соответствовать стандартам безопасности. Из-за этого магнитные считыватели были заменены на более современные и хорошо защищенные считыватели.

Плюсы считывателей магнитных карт:

- простота использования;
- низкая стоимость;
- высокая скорость чтения данных;

- на карте можно хранить большие объемы информации.

Минусы считывателей магнитных карт:

- низкая надежность, так как магнитные карты могут быть повреждены или стерты;
- ограниченный срок эксплуатации, так как магнитная информация может сгореть со временем;
- ограниченный уровень защиты от мошенничества и подделки;
- возможность несанкционированного доступа к данным.

На рисунке 3 показан магнитный считыватель.



Рисунок 3 – Считыватель магнитных карт

Контактные считыватели идентифицируют информацию, записанную в память транспондера (RFID-метки). Эти контактные устройства имеют большой объем информации, необходимый для записи информации, что делает микрочипы очень надежными. Однако у таких считывателей изнашиваются контакты, что ограничивает срок службы идентификационной карты.

Плюсы RFID-считывателей карт:

- быстрота: RFID-считыватели могут считывать информацию с карт быстрее, чем обычные считыватели на основе штрих-кода;
- удобство: карты, которые используют технологию RFID, часто меньше и легче для ношения в сравнении с традиционными картами с магнитной полосой;

- безопасность: RFID-технологии могут помочь улучшить безопасность в различных областях, таких как платежи и доступ в здания.

Минусы RFID-считывателей карт:

- цена: RFID-карты и считыватели могут быть дороже, чем традиционные карты и считыватели;

- распространенность: некоторые места, такие как небольшие магазины, могут не использовать RFID-технологии, что может ограничивать возможности использования карт;

- проблемы конфиденциальности: RFID-карты могут быть сканированы без вашего ведома, что может представлять угрозу для конфиденциальности пользователя.

На рисунке 4 показан контактный считыватель СКУД.



Рисунок 4 – RFID-считыватель карт

Бесконтактные считыватели работают практически так же, как и контактные. Разница заключается в том, что для передачи информации, записанной на носителе, устройство не нужно физически подключать к контакту. Бесконтактные считыватели обладают рядом преимуществ: неограниченный срок службы, простота конструкции и наличие аппаратного обеспечения для распознавания сигналов транспондера. Однако, доступность устройств использующих RFID-метки, для широкой публики, привела к быстрому созданию дубликатов.

Плюсы бесконтактных считывателей карт:

- удобство использования: не нужно вставлять карту в устройство, достаточно просто поднести ее к считывателю;
- безопасность: при использовании бесконтактного считывания карты информация передается по защищенному каналу связи, что снижает риск кражи конфиденциальных данных;
- быстрота операций: передача информации происходит мгновенно, что позволяет быстрее обрабатывать транзакции;
- универсальность: бесконтактные считыватели карт могут работать с разными видами карт, например, с банковскими картами, проездными билетами и т.д.

Минусы бесконтактных считывателей карт:

- риск несанкционированного доступа: при отсутствии должных мер безопасности злоумышленники могут перехватить данные карты в бесконтактном режиме;
- ограниченный диапазон чтения: часто требуется подносить карту к считывателю на расстоянии не более 5-10 см;
- необходимость заряда: некоторые бесконтактные считыватели работают от аккумуляторов или батарей, требующих периодической зарядки;
- стоимость: бесконтактные считыватели карт могут стоить дороже, чем традиционные устройства чтения карт.

На рисунке 5 показан бесконтактный считыватель.



Рисунок 5 – Бесконтактный считыватель карт

Считыватели виртуальных данных получают данные от имеющихся у пользователей устройств, например от установленных на них приложений. Недостатком является то, что он подключается к терминалу пользователя и может быть удаленно перехвачен во время передачи данных.

Сканеры биометрических данных позволяют обеспечить надежное распознавание пользователя по следующим биометрическим данным:

- папиллярный узор на пальце руки;
- радужка глаза;
- геометрические параметры формы лица;
- рисунок кровеносных сосудов на запястье или на наружной стороне кисти.

Биометрические сканеры включают программный пакет, который преобразует мультимедийную информацию в цифровой код. Программный пакет увеличивает стоимость оборудования, но повышает надежность и безопасность распознавания, что оправдывает его использование.

Использование алгоритмов биометрического распознавания в системах контроля и управления доступом требует надежной защиты персональных данных. База оборудования хранит только цифровые коды и не может быть использована для восстановления параметров личности [9].

Также, к набору датчиков системы, помимо считывателей и идентификаторов относятся:

- датчики открывания дверей;
- инфракрасные датчики присутствия и движения;
- видеокамеры и фотокамеры внутреннего наблюдения;
- датчики состояния внешней среды, определяющие температуру и влажность, показания которые могут скорректировать коды для повышения достоверности идентификации [13].

В качестве датчика, идентифицирующего условия и события будет использоваться бесконтактный считыватель карт, так как в отличие от остальных датчиков, в бесконтактный считыватель не нужно вставлять

карту, достаточно просто приложить. Исполнительный механизм СКУД – это оградительная конструкция, блокирующая доступ на объект посторонним лицам и транспортным средствам. Они могут быть «представлены различными устройствами, такими как шлагбаумы, распознавательные турникеты, ворота, раздвижные круглые или вращающиеся двери» [2].

Выделим самые распространенные виды исполнительных механизмов.

Электромагнитные и электромеханические замки. При использовании замков необходимо учитывать механическую нагрузку удержания в зависимости от характеристики дверей. Электромагнитные замки потребляют меньше энергии, чем электромеханические, но их работа полностью зависит от наличия источника питания.

Плюсы электромагнитного замка:

- высокая степень безопасности: электромагнитный замок является одним из наиболее надежных замков, поскольку он физически блокирует дверь или окно;
- простота установки: установка электромагнитного замка в большинстве случаев не требует сверления дополнительных отверстий и легко монтируется на большинстве типов дверей;
- быстрый доступ: вход с использованием электромагнитного замка обеспечивает быстрый доступ без необходимости использования ключа;
- управление доступом: с помощью системы контроля доступа авторизованным пользователям можно предоставить доступ к зонам, где нужны повышенные требования безопасности.

Минусы электромагнитного замка:

- возможность обойти замок: если дверь расположена далеко от контрольной точки, то возможно ее простое обход блокировки;
- требует электропитания: электромагнитный замок требует постоянного электропитания, и если питание прекращается, возможно, что замок может открыться.

На рисунке 6 представлен электромагнитный замок.



Рисунок 6 – Электромагнитный замок

Шлюзовые кабины тамбурного типа и ротанты состоят из пары вращающихся дверей, которые предоставляют доступ после предъявления пользователем идентификатора. Для большей безопасности устройство оснащают системой взвешивания и металлодетекторами. Это оборудование предотвращает проход нескольких человек с одним и тем же удостоверением личности и пронос оружия.

Плюсы ротант:

- удобство использования: ротанты легко устанавливаются и использование их не требует большого количества времени и усилий;
- быстрый доступ: вход с использованием ротанта обеспечивает быстрый доступ без необходимости использования ключа;
- управление доступом: с помощью системы контроля доступа авторизованным пользователям можно предоставить доступ к зонам, где нужны повышенные требования безопасности.

Минусы ротант:

- высокая стоимость: ротанты, как правило, дороже других видов замков и могут превышать стоимость других видов защиты;
- требует персонала: ротанты, как правило, требуют присутствия персонала для управления их работой, что может увеличить расходы на обслуживание объекта;
- низкая скорость прохода: ротанты не пропускают более одного человека за раз, что может затруднить быстрый проход.

Ротанта показана на рисунке 7.



Рисунок 7 – Ротанта

Поясные и полноростовые турникеты используются в офисных центрах и на проходных, где дополнительно находится охранник. При предъявлении идентификатора турникет поворачивается и дает доступ человеку на объект.

Плюсы поясных турникетов:

- высокая степень безопасности: поясные турникеты предоставляют высокий уровень безопасности и могут использоваться для защиты доступа в любой охраняемый объект;
- быстрый доступ: проход через поясной турникет обеспечивает быстрый доступ без необходимости использования ключа;
- малые размеры: поясные турникеты имеют компактный дизайн и не занимают много места;
- удобство использования: поясные турникеты легко устанавливаются и использование их не требует большого количества времени и усилий;
- управление доступом: с помощью системы контроля доступа авторизованным пользователям можно предоставить доступ к зонам, где нужны повышенные требования безопасности;
- распространенность: поясные турникеты широко распространены в университетах страны.

Минусы поясных турникетов:

- требует персонала: поясные турникеты, как правило, требуют присутствия персонала для управления их работой, что может увеличить расходы на обслуживание объекта;
- необходимость специальных знаний: установка и настройка поясных турникетов может потребовать от пользователя специальных знаний;
- ограниченное применение: поясные турникеты могут использоваться только в тех случаях, когда необходимо контролировать доступ только одного или нескольких человек;
- низкая скорость прохода: поясные турникеты не пропускают более одного человека за раз, что может затруднить быстрый проход.

На рисунке 8 представлен поясной турникет.



Рисунок 8 – Поясной турникет

Автоматика для ворот и шлагбаумы контролируют въезд и выезд транспортных средств с объекта. Имеются раздвижные, подъемные и рулонные механизмы. Данные СКУД оснащаются устройствами дистанционной идентификации атрибутов, механизмами аварийной остановки транспортных средств и техническими эндоскопами.

В качестве исполнительного механизма СКУД будет использоваться поясной турникет, так как в отличие от ротант и электромагнитных замков:

- поясные турникеты используются в качестве исполнительных механизмов СКУД в большинстве университетах страны;

- поясные турникеты обычно поставляются вместе с бесконтактными считывателями;

1.3 Системы учета посещаемости занятий студентами и преподавателями

Одной из важных проблем любого общеобразовательного учреждения является посещаемость – система присутствия на занятиях в целях усвоения образовательной программы. Посещаемость относится к числу тех категорий, которые, с одной стороны, служат объектом управления образовательным процессом, с другой – во многом определяют образованность и воспитанность молодежи [10].

Система учета посещаемости относится к компьютеризированному программному обеспечению и специальным устройствам, которые собирают данные о посещаемости, такие как время прибытия, ежемесячное совокупное количество часов работы и отсутствие студентов и преподавателей.

Чтобы контролировать посещение занятий студентов и преподавателей были созданы программные средства:

- автоматизированный комплекс учета посещаемости и успеваемости студентов «АРАТО»;
- модуль автоматизированного комплекса «Альма Матер», под названием «Учет успеваемости и посещаемости»;
- автоматизированная система на базе «1С: Предприятие 7.7», под названием «1С: Деканат».

Рассмотрим каждый из них более подробно.

Автоматизированный комплекс «АРАТО» включает в себя базу данных всех зачисленных студентов, электронный учет посещаемости и успеваемости, средства документирования и подсчета статистики. Доступ в программу предоставляется только преподавателям, которые зарегистрированы в системе при вводе пароля. Студенты заносятся в базу

данных и прикрепляются к определенным подгруппам в иерархическом справочнике, где факультеты, группы и подгруппы представляют собой различные уровни, что позволяет делать выборки, как по одной группе, специальности, курсу, факультету или всего учебного заведения [6].

Данная программа является комплексом, который облегчает работу сотрудникам университета. К основным характеристикам относятся:

- поддержка больших объемов данных;
- обеспечение работы как локально, так и удаленно;
- низкий порог вхождения из-за удобного интерфейса.

Данные модуля АК «Альма Матер» позволяют деканату отслеживать количество пропусков занятий студентами за семестр и принимать меры по их предупреждению.

Модуль включает в себя:

- управления учебными карточками студентов;
- учет успеваемости студента;
- формирование рейтингов по успеваемости;
- подготовка данных для приложения к диплому;
- ведение электронного реестра посещаемости;
- создание сводного реестра посещаемости.

Автоматизированная система «1С: Деканат», разработанная на базе «1С: Предприятие», предназначена для управления статусом обучения студентов и решения различных учетных задач, таких как «ведение базы данных студентов, учет экзаменов и зачетов, подготовка справок и выписок результатов, расчет платы за обучение» [7].

Основное удобство этой автоматизированной системы учета для образовательных учреждений заключается в сетевой версии «1С: Бухгалтерия 7.7», которая присутствует на всех компьютерах бухгалтерии и администрации.

Такие системы обладают следующими преимуществами:

- системы могут работать как локально, так и с помощью компьютеров при помощи Интернета;

- система разделения прав гарантирует, что каждый пользователь имеет доступ только к своим собственным данным.

Существуют и другие подобные системы, но они схожи по характеристикам с системами, описанными выше.

Однако такие системы обладают следующими минусами:

- цена программного продукта довольно высока, но не все программы имеют высокую стоимость;

- интерфейсы большинства систем перегружены и имеют множество других функций, которые не нужны в процессе учета посещаемости и успеваемости, что затрудняет взаимодействие пользователей с ними;

- многие программы для учета посещаемости имеют узконаправленную направленность, что затрудняет их эксплуатацию и внедрение. Эти программы приобретаются в едином скопированном файле, поэтому изменения в них невозможны, так как исходные материалы отсутствуют, а закон об авторских правах отсутствует.

Исходя из обзора плюсов и минусов существующих систем учета посещаемости, было принято решение о создании собственного программного обеспечения для автоматизации процесса учета посещаемости, но, в отличие от систем, описанных выше, в данном ПО не будет тех функций, которые не потребуются в процессе учета посещаемости; также доступ в данную систему будут иметь старосты групп.

Вывод по разделу 1

Было проанализировано понятие СКУД, был сделан анализ его компонентов. Также был сделан анализ существующих систем учета посещаемости занятий студентами, были выявлены преимущества и недостатки данным систем. В результате проведенного анализа было принято

решение о разработке собственного программного обеспечения для автоматизации процесса учета посещаемости.

2 Разработка программного обеспечения для учета посещаемости занятий студентами и преподавателями

2.1 Моделирование ПО для учета посещаемости занятий студентами и преподавателями на основе событий системы контроля и управления доступом

Для определения ожидаемой функциональности разрабатываемого программного обеспечения, были созданы следующие диаграммы: диаграмма вариантов использования, диаграмма активности, диаграмма развертывания, диаграмма классов.

Диаграмма вариантов использования (use case diagram) – диаграмма, изображающая отношения между акторами (действующими лицами) и вариантами использования (прецедентами).

На рисунке 9 показана диаграмма вариантов использования.

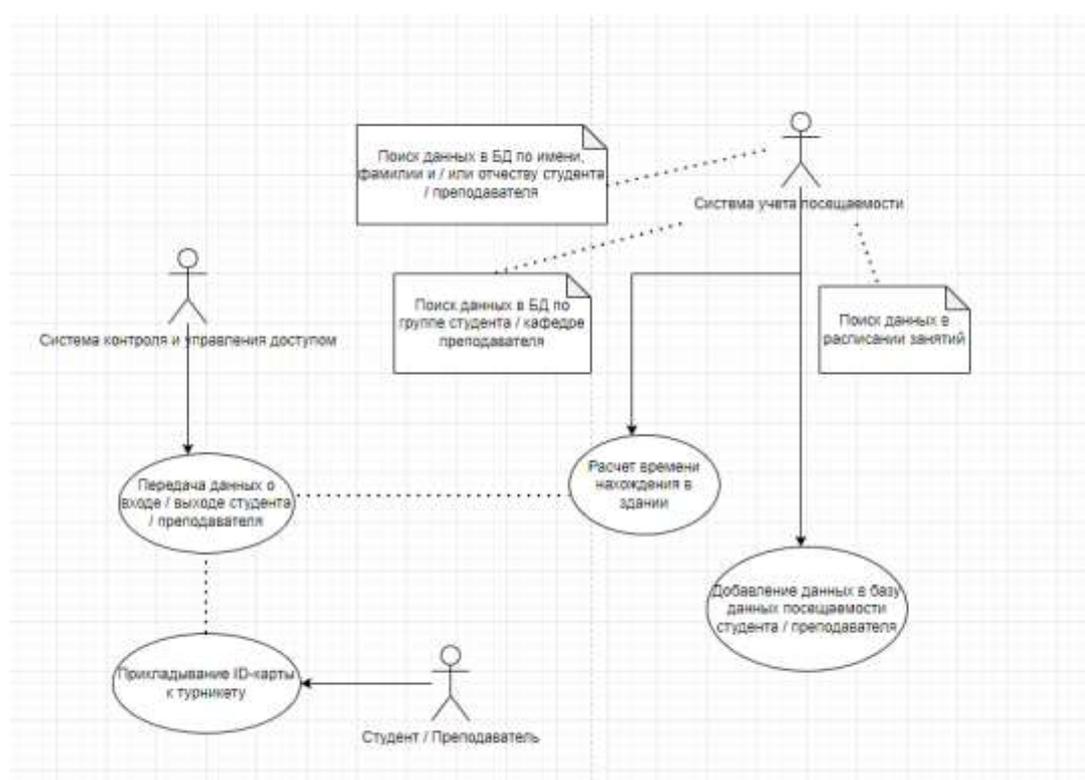


Рисунок 9 – Диаграмма вариантов использования

Цель этой диаграммы: представить проектируемую информационную систему в виде так называемых вариантов использования, в котором взаимодействуют внешние сущности (акторы). Актор или действующее лицо – это любой объект, сущность или система, которая взаимодействует со смоделированной бизнес-системой [3].

На рисунке 9 присутствуют 3 актора: студент / преподаватель, система учета посещаемости, система контроля и управления доступом.

Студент / преподаватель выполняет 1 действие: прикладывает карту доступа. Прикладывание карты запускает систему контроля и управления доступом, которая передает данные о входе системе учета посещаемости. В свою очередь, система учета посещаемости ищет в БД наличие студента / преподавателя, который зашел в здание. После того, как студент / преподаватель вышел из здания, СКУД передает система учета посещаемости данные о выходе. Система учета посещаемости ищет расписание вышедшего из здания в БД, рассчитывает количество часов, которое студент / преподаватель провел в здании. После этого система учета формирует отчет, в котором содержится время входа, время выхода и количество часов, проведенных в здании. После этого отчет добавляется в БД.

Диаграмма активности – это поведенческая диаграмма, то есть она отображает поведение системы. Диаграмма активности отображает поток управления от начальной точки до конечной точки, показывая различные пути принятия решений, которые существуют во время выполнения действия. Есть возможность изобразить как последовательную, так и параллельную обработку действий. Они используются в бизнес-моделировании и моделировании процессов, где их основное применение заключается в отображении динамических аспектов системы. Диаграмма активности очень похожа на блок схему [19].

На рисунке 10 показана диаграмма активности.

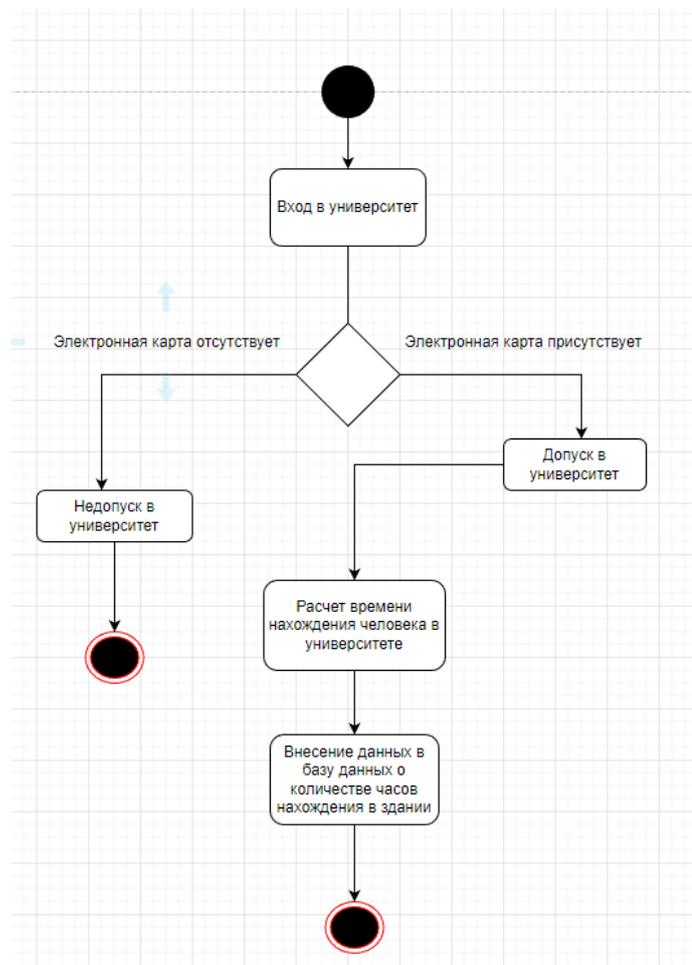


Рисунок 10 – Диаграмма активности

После входа студента / преподавателя в университет, система определяет, пускать его в здание или нет. Если у студента / преподавателя нет карты доступа, то система не пускает его в здание, программа завершает работу. Если же карта имеет система допускает человека в здание. После его выхода, система рассчитывает время его нахождения в здание и заносит результаты в базу данных.

Диаграмма развертывания (deployment diagram) – это диаграмма, отображающая узлы выполнения программных компонентов реального времени, а также процессов и объектов.

Диаграммы развертывания содержат графические изображения процессов, устройств, процессов и связей между ними. В отличие от диаграмм логического представления, диаграммы развертывания являются

уникальными для системы в целом, поскольку они должны отображать все функции ее реализации. Эта диаграмма, по сути, завершает процесс ООАП для конкретной программной системы, разработка которой обычно является заключительным этапом спецификации модели. Диаграмма развертывания создается совместно системными аналитиками, системными и сетевыми инженерами [4].

На рисунке 11 показана диаграмма развертывания.

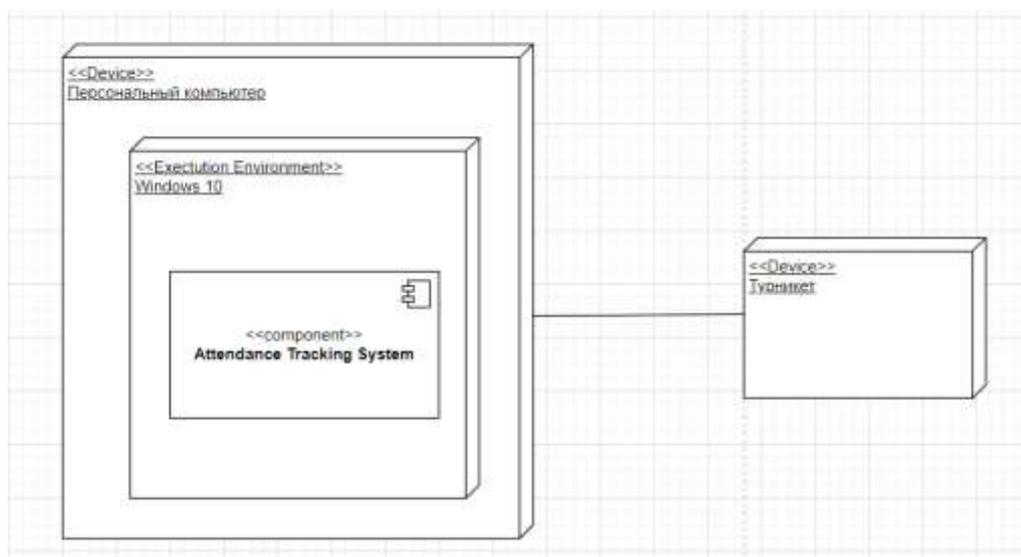


Рисунок 11 – Диаграмма развертывания

На диаграмме развертывания показаны три узла: персональный компьютер, турникет, Windows 10. Узлы – физический элемент, который выполняет один или несколько компонентов. Узел Windows 10 содержит артефакт Attendance Tracking System. Артефакты – это элементы, которые вызваны процессом разработки. Узлы «Персональный компьютер» и «Турникет» связаны коммуникационной ассоциацией.

Диаграмма классов – это диаграмма языка моделирования UML, показывающая иерархическую структуры классов системы и их методов и взаимосвязей между ними. На рисунке 12 показана диаграмма классов.

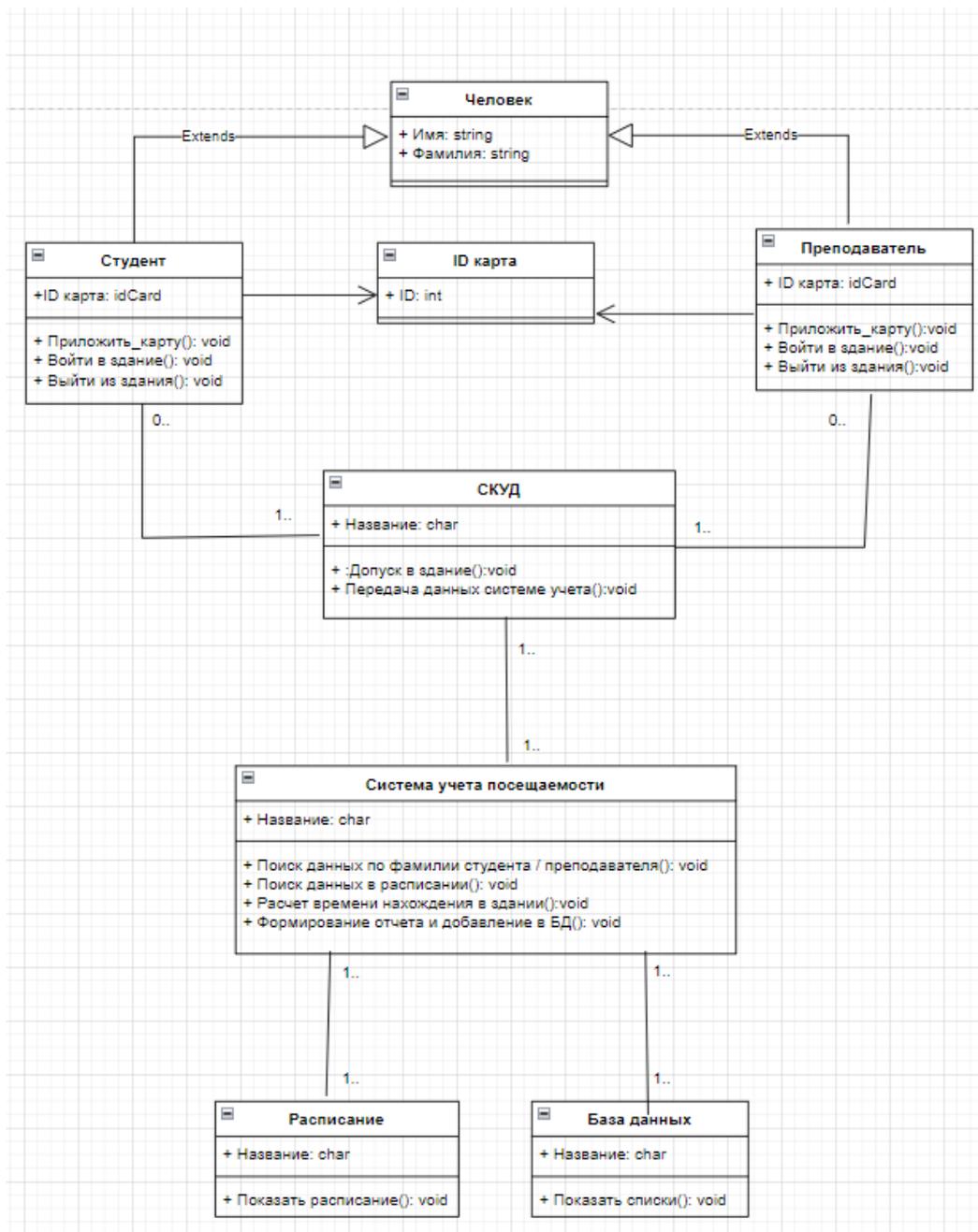


Рисунок 12 – Диаграмма классов

Связи между классами «Студент», «Преподаватель» и классом «Человек» называются наследованием. Данная связь описывает отношение между родительским классом («Человек») и дочерним классом («Студент», «Преподаватель»).

Связи между классами «Студент», «Преподаватель» и классом «ID карта» называются ассоциацией. Классы Студент», «Преподаватель» имеют

поле `card`, у которого тип `idCard`. Из класса объектов «Студент», «Преподаватель» мы можем узнать о связанном с ним объектом типа `idCard`.

Остальные связи являются множественными ассоциациями. Они представляют собой диапазон целых чисел, указывающее количество связанных объектов. Например, количество “0..” означает, что таких объектов может быть много, количество “1..” означает, что обязательно должен быть, как минимум, один объект.

Необходимо выбрать технологии для разработки программного обеспечения.

2.2 Выбор средств реализации программного обеспечения

Рассмотрим несколько языков программирования для начинающих разработчиков, рассмотрим их плюсы и минусы.

C++ - это компилируемый, статически типизированный язык программирования общего назначения.

Он поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщенное программирование. Язык имеет обширную стандартную библиотеку, включающую такие функции, как контейнеры и алгоритмы, ввод-вывод, регулярные выражения и поддержка многопоточности. C++ сочетает в себе черты как высокоуровневых, так и низкоуровневых языков. По сравнению со своим предшественником, языком C, он уделяет наибольшее внимание объектно-ориентированному и обобщенному программированию [15].

К плюсам языка C++ можно отнести:

- наличие перезагрузки функций и операторов;
- наличие наследования классов и пользовательского контроля над управлением памятью;
- высокая вычислительная производительность;

- возможность расширения языка;
- наличие шаблонов, которые дают возможность построения обобщенных алгоритмов и контейнеров.

К минусам языка C++ относится:

- сложность и избыточность, из-за которых C++ трудно изучать;
- в языке практически полностью сохранен набор конструкций C;
- шаблоны в своем исходном виде приводят порождению кода очень большого объема;
- метапрограммирование на основе шаблонов C++ сложно и при этом ограничено в возможностях.

Python – это высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности труда разработчиков, улучшение читаемости и качества кода, а также на обеспечение переносимости написанных на нем программ.

Python – это мультипарадигменный язык программирования, поддерживающий императивное, процедурное, структурное, объектно-ориентированное программирование, метапрограммирование и функциональное программирование [17].

К плюсам языка Python можно отнести:

- подходит для начинающих разработчиков;
- простой минималистичный синтаксис: код легко писать, читать и поддерживать;
- наличие обширной стандартной библиотеки и других дополнительных библиотек;
- большой выбор фреймворков;
- поддержка объектно-ориентированное программирование и другие парадигмы;
- кроссплатформенность и поддержка практически всех современных систем.

К минусам языка Python относятся:

- низкая скорость;
- плохо подходит для разработки мобильных приложений;
- из-за динамической типизации повышается вероятность ошибок при запуске и требуется больше тестов;
- не подходит для низкоуровневой работы с памятью.

JavaScript – это мультипарадигменный язык программирования. Он поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript (стандарт ECMA-262).

JavaScript обычно используется в качестве встроенного языка для программного доступа к объектам приложений. Это наиболее широко используемый язык сценариев в браузерах для добавления интерактивности веб-страницам[16].

К плюсам языка JavaScript можно отнести:

- данный язык необходим для веб-разработке, поскольку он является основной «технологией» модели «клиент-сервер» или программирования «для интернета»;
- JavaScript полностью интегрирован с версткой страниц программированием на стороне сервера;
- JavaScript предлагает рациональность применения и простоту, так как элементарные задачи могут быть решены за несколько минут. Язык программирования имеет очень понятный синтаксис, который не требует углубленного изучения;
- скорость работы. Язык программирования JavaScript позволяет сделать конечный программный продукт более быстрым и простым в использовании;
- производительность;
- простота использования пользовательских интерфейсов;

- наличие уникальной и мощной экосистемы (инфраструктуры). В качестве примера можно привести ряд полезных фреймворков, которые подойдут для любого приложения.

К минусам языка JavaScript относятся:

- невозможность просмотра или загрузки документов из-за соображения безопасности;
- удаленный доступ невозможен, так как полностью сетевое программное обеспечение не использует соответствующий язык;
- типизация не является строгой;
- вольная интерпретация типов, т.е. несоответствия в коде игнорируются;
- доступность для конкурентов. Связано это с высокой читаемостью исходного кода языка программирования JavaScript.

В рамках данной выпускной квалификационной работы будет использоваться язык программирования JavaScript, так как в отличие от C++ и Python:

- данный язык программирования является основой для создания веб-приложений;
- JavaScript гораздо проще изучается.

JavaScript-фреймворки – это библиотеки программирования JavaScript с заранее написанным кодом для использования в стандартных функциях и задачах программирования. На их основе можно создавать веб-сайты и веб-приложения [8].

JavaScript-фреймворки являются обязательной частью нынешней веб-разработки, предоставляя разработчикам проверенные и протестированные инструменты для создания масштабируемых и интерактивных веб-приложений.

Рассмотрим несколько фреймворков языка программирования JavaScript.

AngularJS – это структурированная основа для динамических веб-приложений. Он позволяет программистам использовать HTML в качестве языка шаблонов и расширяет синтаксис HTML для краткого описания компонентов веб-приложения. Реализация в AngularJS связей данных и зависимостей означает, что большая часть кода, который приходится писать программистам, в значительной степени исключается. А поскольку все это происходит в браузере, он является «идеальным партнером для любой технологии на стороне сервера» [14].

Преимущества AngularJS:

- Angular используется в сочетании с Typescript. Он обладает исключительной поддержкой для этого;
- подробная документация, позволяет разработчикам получать необходимую информацию без помощи коллег;
- Angular-language-service – обеспечивает интеллектуальные возможности и автозаполнение шаблона HTML-компонента;
- новые возможности, такие как generation Angular, использующие библиотеки npm из CLI, generation, и разработка компонентов с использованием Angular;
- односторонняя привязка данных, которая обеспечивает исключительное поведение приложения и минимизирует риск;
- MVVM (Model-View-ViewModel), которая позволяет разработчикам работать отдельно над одним и тем же разделом приложения, используя один и тот же набор данных;
- структура и архитектура, специально созданные для большой масштабируемости проекта;
- внедрение зависимостей от компонентов, связанных с модулями и модульностью в целом.

Недостатки Angular:

- низкая производительность, учитывая различные метрики. С другой стороны, эту проблему можно легко решить с помощью

«ChangeDetectionStrategy», который помогает вручную контролировать процесс рендеринга компонентов;

- различные структуры, такие как Injectables, Components, Pipes, Modules и т.д. затрудняют изучение по сравнению с React и Vue.js, у которых есть только «Component».

React – это библиотека JavaScript с открытым исходным кодом для создания внешних пользовательских интерфейсов. В отличие от других библиотек JavaScript, которые представляют собой полноценную платформу для разработки приложений, «React фокусируется исключительно на создании представлений приложения с инкапсулированными единицами (компонентами), которые хранят состояние и генерируют элементы пользовательского интерфейса» [18].

Преимущества React:

- простой дизайн, использование JSX (HTML-подобного синтаксиса) для шаблонов и очень подробная документация делают язык легким для изучения, поскольку программисты тратят больше времени на написание современного языка JS и меньше беспокоятся о коде, специфичном для фреймворка;

- привязка данных является односторонней, что означает меньше нежелательных побочных эффектов;

- поддержка Progressive Web App (PWA) с генератором приложений ‘create-react-app’;

- наличие Redux – самой популярной платформы для управления состоянием приложений в React, ее легко учить и использовать;

- отличный рендеринг на стороне сервера делает его мощной платформой для контент-ориентированных приложений;

- React реализует концепции функционального программирования (FP), что упрощает тестирование и повторное использование кода;

- приложения можно создавать с помощью TypeScript или Facebook’s Flow с поддержкой JSX;

- навыки, полученные в React, могут быть применены к разработке на React Native.

Недостатки React:

- Данный язык программирования неоднозначен и разработчикам необходимо выбрать лучший метод разработки. Это можно решить с помощью хорошо скоординированным управлением проектами;

- Смесь паттернов и логики (JSX) может сбить с толку некоторых разработчиков, впервые столкнувшихся React;

- React отходит от компонентов на основе классов, что может стать препятствием для разработчиков, привыкших к объектно-ориентированному программированию (ООП);

Vue.js – это прогрессивный фреймворк для создания пользовательских интерфейсов. В отличие от других фреймворков, Vue предназначен для постепенного внедрения. Его ядро в основном решает задачи на уровне представлений (view), что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, «Vue также хорошо подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications) при использовании в сочетании с современными инструментами и дополнительными библиотеками» [1].

Преимущества Vue.js:

- наличие расширенного HTML. Это значит, что Vue.js имеет большое множество характеристик, схожих с Angular. Благодаря использованию различных компонентов, данная схожимость помогает оптимизировать HTML-блоки;

- Vue.js имеет очень подробную документацию, которая ускоряет процесс обучения разработчиков и может значительно сократить время разработки приложения при наличии лишь базовых знаний HTML и JavaScript;

- интеграция: Vue.js можно использовать как для создания одностраничных приложений, так и для более сложных веб-интерфейсов приложений;

- адаптивность: дизайн и архитектура схожи с Angular и React, что позволяет быстро перейти с других фреймворков на Vue.js;

- Vue.js обеспечивает гораздо более высокую производительность, по сравнению с другими платформами, сохраняя при этом скорость и гибкость;

- масштабируемость: Vue.js позволяет разрабатывать достаточно большие многократно используемые шаблоны, которые могут быть сделаны примерно за тоже время, что и более простые.

Недостатки Vue.js:

- из-за отсутствия опыта работы с решениями, Vue.js может быть проблематичным при интеграции в огромные проекты;

- недостаток ресурсов.

В рамках данной выпускной квалификационной работы будет использоваться фреймворк React.js, так как в отличие от Angular и Vue:

- данный фреймворк изучается легче;

- React обладает поддержкой PWA, что позволяет запускать приложения на всех устройствах из одной базы кода.

«IDE (Integrated development environment – интегрированная среда разработки) - набор программных инструментов, используемых программистами для разработки программного обеспечения» [5].

Среда разработки выглядит следующим образом:

- средства автоматизации сборки;

- текстовый компилятор;

- трансляторы (компиляторы и/или интерпретаторы).

Рассмотрим несколько IDE.

WebStorm – это интегрированная среда разработки, специализирующаяся на JavaScript и TypeScript. Большое количество веб-

разработчиков используют WebStorm для внутренней и внешней разработки. Она поддерживает фреймворки JavaScript, такие как Node.js, Angular.js и React. Среди ее возможностей - интеграция с GitHub, интеллектуальное автозаполнение и рефакторинг кода для всех файлов проекта.

Плюсы:

- бесплатно для студентов;
- надежные обновления и поддержка пользователей;
- доступно в Windows, Linux и macOS;
- настраивается с помощью плагинов.

Минусы:

- потребляет память устройства;
- загрузка иногда может занять много времени.

IntelliJ IDEA Ultimate – это Java-ориентированная IDE, которая также популярна для разработки на JavaScript. На момент выхода IntelliJ была одной из первых сред Java IDE, в которой были интегрированы такие функции, как навигация по коду и рефакторинг кода. На сегодняшний день в IntelliJ интегрирована поддержка таких JavaScript-фреймворков, как Node.js.

Плюсы:

- интуитивно понятный пользовательский интерфейс;
- наличие специальных возможностей, позволяющих включать программу чтения с экрана;
- точное автозаполнение на основе модели DOM.

Минусы:

- медленное индексирование;
- дорогая IDE; бесплатная версия не поддерживает JavaScript.

NetBeans – это бесплатная среда IDE, хорошо известная программистам на Java. Хотя NetBeans чаще всего используется для работы с Java, но поддерживает множество программирования, включая JavaScript, HTML5 и CSS3. Редактор JavaScript в NetBeans поддерживает подсветку и автозаполнение, а также имеет встроенную поддержку jQuery. NetBeans

поддерживает такие JavaScript-фреймворки, как Angular, Express, Vue, React и Mocha.

Плюсы:

- наличие отладчика, встроенной поддержки Git и сворачивания кода;
- легкая настройка пользовательского интерфейса;
- доступность на Windows, Linux и macOS

Минусы:

- сложность интеграции с другими инструментами;
- очень мало дополнительных функций;
- требует больших затрат ресурсов;
- может работать медленно даже для небольших front-end проектов.

Таким образом, будет использоваться интегрированная среда разработки WebStorm, так как, в отличие от NetBeans и IntelliJ:

- данная IDE бесплатная для студентов;
- На WebStorm выходят больше обновлений;
- WebStorm более популярен у начинающих разработчиков.

Таким образом, для разработки программного обеспечения были выбраны следующие инструменты:

- Язык программирования JavaScript;
- Фреймворк React;
- Среда разработки WebStorm.

После выбора технологий перейдем к реализации разработанного приложения.

Вывод по разделу 2

Были спроектированы UML-диаграммы работы программного обеспечения: диаграмма вариантов использования, диаграмма активности и диаграмма развертывания. Также был сделан анализ языков

программирования, фреймворков и IDE. На основе сравнения их плюсов и минусов были выбраны те, что будут использоваться при разработке программного обеспечения.

3 Реализация и тестирование программного обеспечения

3.1 Контрольный пример разработанного ПО

Разработанное программное обеспечение представляет собой веб-приложение с возможностью фиксации студентов и преподавателей, входящих в здание и посещающих занятия. На рисунке 13 показано главное меню веб-приложения.

Главное меню

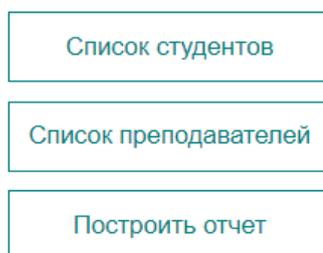


Рисунок 13 - Главное меню веб-приложения

Сначала нужно добавить студентов и их учебные группы, а также преподавателей и их принадлежность к определенному направлению. При переходе на вкладку «Список студентов» по умолчанию отображается пустой список. На рисунке 14 показано окно с пустым списком

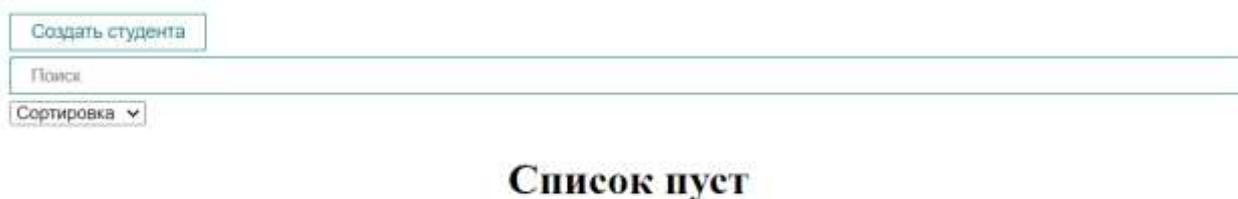


Рисунок 14 - Окно с пустым списком

При нажатии на кнопку «Создать студента» всплывает модальное окно, в котором можно ввести данные студента и добавить их в список. Данное окно показано на рисунке 15.



The screenshot shows a modal form with four input fields stacked vertically. The first three fields are labeled 'Фамилия студента', 'Имя студента', and 'Группа студента'. Below these fields is a button labeled 'Добавить студента в список'.

Рисунок 15 – Модальное окно

После ввода данных и нажатия на кнопку «Добавить студента» в список добавляются данные о студенте. Окно с заполненным списком показано на рисунке 16.



The screenshot shows a user interface for a student list. At the top left is a button labeled 'Создать студента'. Below it is a search bar labeled 'Поиск'. At the bottom left is a dropdown menu labeled 'Сортировка' with a downward arrow.

Список студентов

1. Безроднов Павел, Моб-1902а
2. Алборов Василий, ПМН6-1902б
3. Волкин Петр, ПН6-1903а

Рисунок 16 – Окно с заполненным списком

Также на данной странице присутствует строка «Поиск» и выпадающий список «Сортировка».

Строка «Поиск» позволяет найти нужного студента по фамилии. Окно с задействованным поиском представлено на рисунке 17.

У поиска отсутствует чувствительность к регистру – особенность программ, которая заключается в том, что даже если два имени отличаются только регистром символов (то есть заглавными буквами), то эти имена считаются разными. В данном программном обеспечении есть возможность находить заглавные символы, вводя только строчные символы.

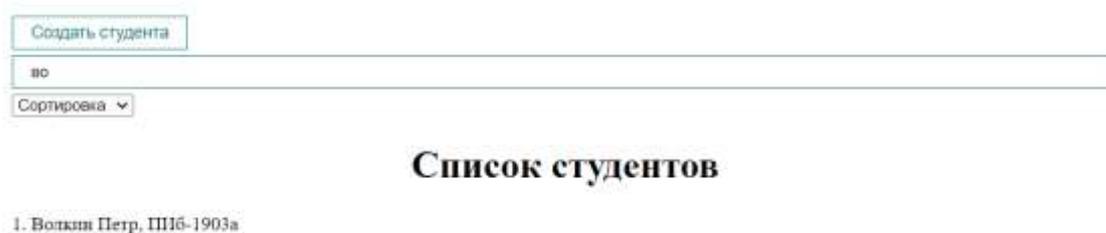


Рисунок 17 – Окно с задействованным поиском

В выпадающем списке «Сортировка» есть 2 пункта: сортировка по фамилии и сортировка по группе. Сортированный список по фамилии показан на рисунке 18.

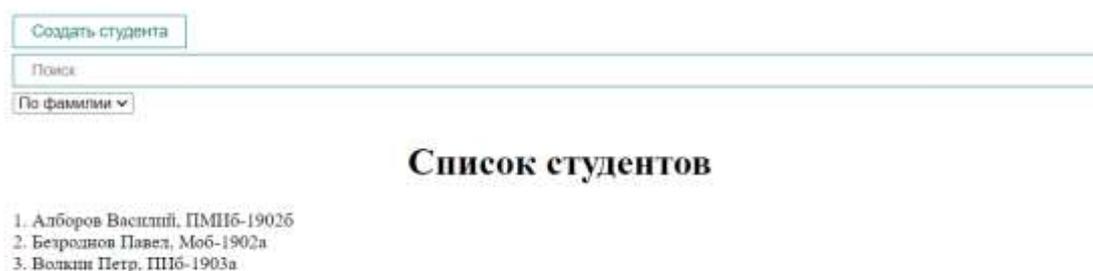


Рисунок 18 – Сортированный список

Аналогично, рассмотрим список преподавателей.

На рисунке 19 показано окно добавления преподавателей.

Фамилия преподавателя
Имя преподавателя
Отчество преподавателя
Наименование института
Добавить преподавателя в список

Рисунок 19 – Модальное окно добавления преподавателей

На рисунке 20 показан заполненный список преподавателей.

Добавить преподавателя
Поиск по фамилии
По фамилии ▾

Список преподавателей

1. Григорьев Ева Львовна, ПМФФИнТ
2. Писнев Александр Егорович, ПМФФИнТ
3. Фролов Александр Иванович, ПМФФИнТ

Рисунок 20 – Список преподавателей

В главном меню, если нажать на кнопку «Построить отчет», на основе списка студентов и системы контроля и управления доступом сформируется отчет, в котором показаны фамилия, имя, отчество студента / преподавателя, группа студента / институт преподавателя, вход и выход из здания, а также количество часов, проведенных в здании. На рисунке 21 показано окно со сформированным отчетом студентов.

ID	Имя и фамилия	Группа	Время входа	Время выхода	Количество часов
1	Безроднов Павел	Моб-1902а	10:31:13	18:31:13	8
2	Алборов Василий	ПМН6-1902б	14:31:13	17:31:13	3
3	Волкин Петр	ПН6-1903а	12:31:13	13:31:13	1

Рисунок 21 – Окно со сформированным отчетом

На рисунке 22 показано окно со сформированным отчетом преподавателей.

ID	ФИО	Институт	Время входа	Время выхода	Количество часов
1	Григорьева Ева Львовна	ИМФИнТ	8:10:1	11:10:1	3
2	Пименов Кирилл Егорович	ИМФИнТ	7:10:1	9:10:1	2
3	Фролов Александр Иванович	ИМФИнТ	10:10:1	15:10:1	5

Рисунок 22 – Окно со сформированным отчетом

Таким образом, был продемонстрирован функционал программного обеспечения. Были показаны функции добавления студентов и преподавателей в списки и функция построения отчета с количеством проведенных в здании часов.

3.2 Тестирование программного обеспечения

Тестироваться будет момент добавления студента в базу данных.

Были выбраны два вида тестирования: компонентное и интеграционное.

Цель компонентного тестирования: проверка реализации функциональных требований.

На рисунке 23 показан пример страницы с модулями.

Страница Students.jsx отвечает за показ блока Students и формы CreateStudent, форма CreateStudent отвечает за отправку данных в блок Student, блок Student отвечает за обработку запросов с формы CreateStudent и воспроизведение их на экран.

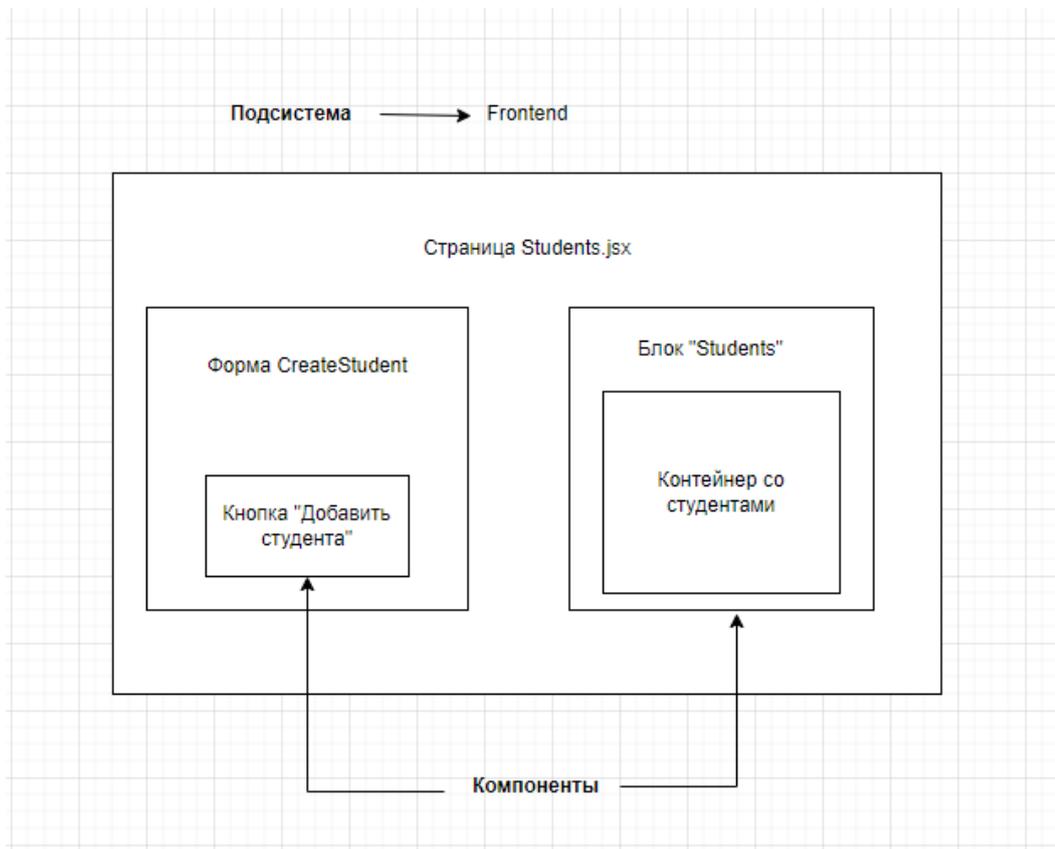


Рисунок 23 – Модули страницы

Рассмотрим компонент Students.jsx. Требования:

- открываться в браузере по указанному URL;
- содержать правильную информацию;
- содержать форму CreateStudent и блок Student и отвечать за их работоспособность.

Все описанные выше требования успешно выполнены.

Цель интеграционного тестирования: проверка реализации взаимодействия между компонентами.

На рисунке 24 показано взаимодействие между компонентами страницы.

Стрелка 3 описывает связь между компонентами CreateStudent и Students. В случае нажатия на кнопку данные, введенные в компоненте CreateStudent будут переданы и отображены в компоненте Students.

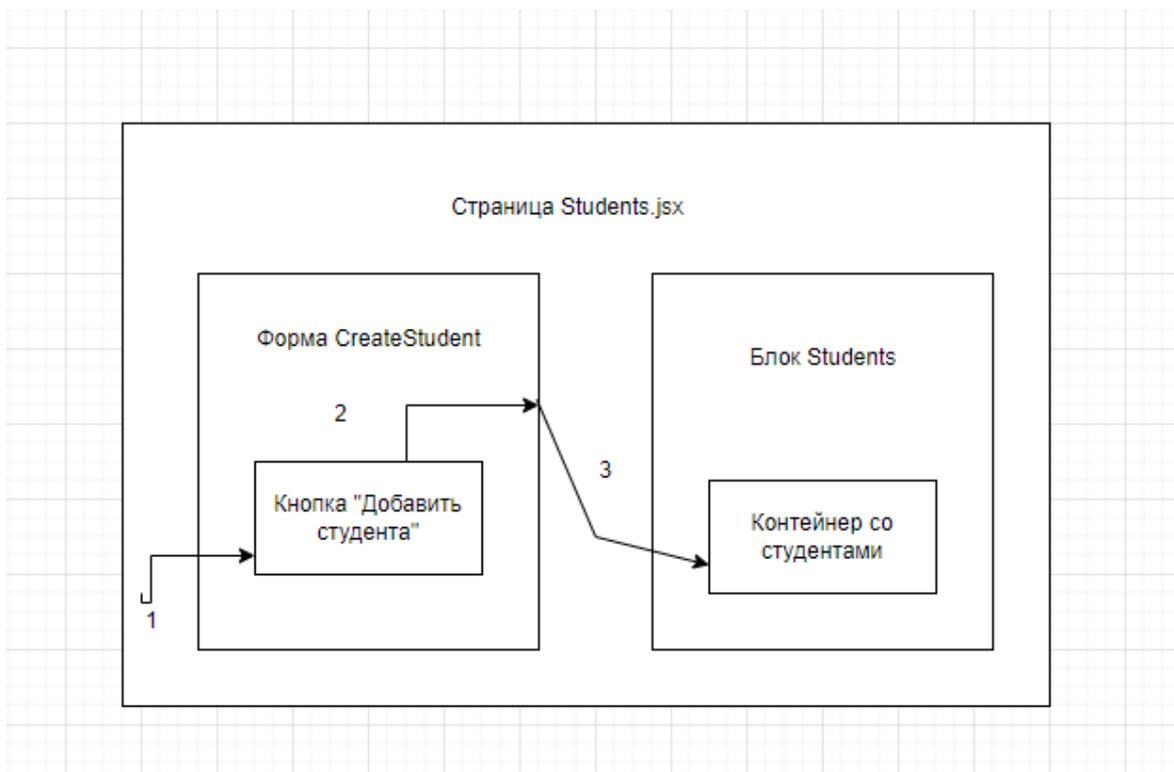


Рисунок 24 – Взаимодействие между компонентами

Таким образом, были проведены два вида тестирования: компонентное и интеграционное. Были созданы соответствующие схемы модулей и компонентов, а также были описаны требования к реализации.

Вывод к разделу 3

В разделе 3 был продемонстрирован функционал разработанного программного обеспечения и проведено тестирование ПО, которое показало, что разработанное программное обеспечение работает исправно.

Заключение

Результатом выпускной квалификационной работы стала разработка программного обеспечения для системы учета посещаемости студентов и преподавателей на основе системы контроля и управления доступом в виде web-приложения. Данное приложение имеет дружелюбный интерфейс и функционал.

В ходе выполнения выпускной квалификационной работы были решены следующие задачи:

- были проанализированы понятие СКУД и его компоненты;
- был сделан анализ на существующие системы учета посещаемости занятий студентами;
- были выбраны технологии разработки программного обеспечения;
- было спроектировано программное обеспечение.

Для разработки программного обеспечения был выбран мультипарадигменный язык программирования JavaScript, который применяется для создания браузерных приложений.

Созданное программное обеспечение предназначено для работы на персональном компьютере под управлением операционной системы Windows 10.

Интерфейс программной оболочки представляет собой стандартный интерфейс GUI в стиле многодокументного приложения, форма запуска является главной формой, а дочерние открывается внутри нее.

В программном обеспечении можно добавлять новых студентов и преподавателей в базу данных, указывая их ФИО, группы (у преподавателей – принадлежность к определенному университету).

Программное обеспечение обладает удобным поиском. Данные можно отбирать по частичному совпадению, что позволяет пользователю ввести часть названия и система выдаст ему результат поиска. Также web-

приложение обладает возможностью сортировки по фамилии и группе (у преподавателей – по институту).

В разработанном программном обеспечении можно спроектировать отчет о входе в здание и выходе из здания студентов и преподавателей и о количестве часов их пребывания в здании.

Список используемой литературы

1. Введение – Vue.js [Электронный ресурс] – Режим доступа: <https://ru.vuejs.org/v2/guide/>.
2. Виды СКУД. Выбор системы контроля доступом [Электронный ресурс] – Режим доступа: <https://domofonov.net/blog/obzory/vidy-skud>.
3. Диаграмма прецедентов (вариантов использования или Use Case) [Электронный ресурс] – Режим доступа: https://github.com/kolei/PiRIS/blob/master/articles/5_1_1_10_uml_use_case.md.
4. Диаграмма развертывания, особенности ее положения [Электронный доступ] – Режим доступа: <https://intuit.ru/studies/courses/32/32/lecture/1024>.
5. Интегрированная среда разработки [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Интегрированная_среда_разработки.
6. Обзор существующих программных средств для компьютерного учета успеваемости [Электронный ресурс] – Режим доступа: https://studwood.net/1685143/informatika/obzor_suschestvuyuschih_programmnyh_sredstv_kompyuternogo_ucheta_uspevaemosti_poseschaemosti.
7. Обзор существующих программных средств для компьютерного учета успеваемости и посещаемости [Электронный ресурс] – Режим доступа: <https://mgutunn.ru/work/1823103/Obzor-sushhestvuyushhix-programmnyh-sredstv>.
8. Популярные фреймворки JavaScript – Режим доступа: <https://vc.ru/dev/147263-populyarnye-freymvorki-javascript>.
9. Системы контроля и управления доступом [Электронный ресурс] – Режим доступа: <http://scremmark.ru/pto/skud>.
10. Учебная посещаемость, как важнейшая педагогическая категория [Электронный ресурс] – Режим доступа: <https://multiurok.ru/files/uchebnaia-posieshchaiemost-kak-vazhnieishaia-piedagoghichieskaia-katieghoriia.html>.

11. Учет рабочего времени по СКУД [Электронный ресурс] – Режим доступа: <https://crocotime.com/ru/uchet-rabochego-vremeni-po-skud-za-i-protiv/>.
12. Что такое СКУД? [Электронный ресурс] – Режим доступа: <https://www.delta.ru/blog/chto-takoe-skud>.
13. Что такое СКУД? Функции и состав системы. Принцип работы [Электронный ресурс] – Режим доступа: <https://smarthomegadget.ru/skud-sistema-kontrolya-i-upravleniya-dostupom/#i-11>.
14. Angular JS: Development Guide: Introduction [Электронный ресурс] – Режим доступа: <https://docs.angularjs.org/guide/introduction>.
15. C++ [Электронный ресурс] – Режим доступа: <https://en.wikipedia.org/wiki/C%2B%2B>.
16. JavaScript [Электронный ресурс] – Режим доступа: <https://en.wikipedia.org/wiki/JavaScript>.
17. Python (programming language) [Электронный ресурс] – Режим доступа: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)).
18. React в Windows | Microsoft Learn [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/ru-ru/windows/dev-environment/javascript/react-overview>.
19. Unified Modeling Language (UML) | Activity Diagrams [Электронный ресурс] – Режим доступа: <https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/>.
20. What is physical access control? [Электронный ресурс] – Режим доступа: <https://www.openpath.com/blog-post/physical-access-control>.