СОДЕРЖАНИЕ

Введение 4

1 Технический проект 8

1.1 Анализ предметной области 8

1.2 Постановка задачи 9

1.3 Проектирование базы данных 10

1.4 Проектирование функциональности программы 13

1.5 Проектирование структуры программы 14

1.6 Выбор средств реализации 17

1.7 Требования к программному обеспечению и техническим средствам 21

Приложение А – Use Case диаграмма 24

ВВЕДЕНИЕ

Для выполнения своих задач многим учебным заведениям необходимо осуществлять контроль за учащимися и аппаратным обеспечением, находящимся в имуществе предприятия. Поэтому возникла необходимость автоматизировать некоторые процессы с использованием современных подходов и средств реализации информационных систем.

Любая информационная система включает в себя одну или несколько баз данных (БД), что значительно упрощает процессы хранения и обработки информации.

Одной из основных задач отделения является накопление и обработка сведений об аппаратном обеспечении и приложениях запущенных в данный момент на персональном компьютере (ПК). Так же немаловажной частью программы будет возможность управления ПК через локальную сеть. Именно данные процессы будет автоматизироваться в рамках данного дипломного проекта.

Темой дипломного проекта является разработка и создание электронных карточек обучающихся отделения информационных технологий МГГТК ФГБОУ ВО «АГУ».

Объектом исследования является деятельность отделения информационных технологий по контролю действия за обучающимися.

Предметом исследования являются ПК, установленные на отделении, а также локальная сеть, которая позволит осуществлять контроль за всеми ПК удаленно.

Целью данного дипломного проекта является разработка расширяемой информационной системы, которая позволит производить контроль и собирать информацию об ПК, установленных в кабинете организации.

Написание данного дипломного проекта является целесообразным, потому что информационная система:

* позволит производить контроль за ПК, подключенными в локальной сети предприятия;
* облегчит работу учителей за счет возможности просмотра рабочего стола учащегося;
* позволит собирать актуальную информацию об аппаратном обеспечении, находящимся внутри ПК;
* снизит объем бумажной документации.

Разработка нового приложения позволит учесть индивидуальные предпочтения пользователя в интерфейсе и предоставляемых функциях, не перегружая информационную систему лишним, реализованным «впрок», инструментарием. Это снизит системные требования к развертыванию данной информационной системы.

Для достижения поставленной цели будут рассмотрены и решены следующие задачи:

* произведен анализ предметной области и выделены необходимые данные;
* разработаны концептуальная и логическая модели БД;
* разработаны структура и функциональность приложения для управления созданной БД;
* произведён выбор средств реализации;
* описаны минимальные требования к программно-техническим средствам;
* произведена физическая реализация БД;
* описаны входные и выходные данные;
* разработан программный код приложения;
* разработана пользовательская система помощи;
* разработана инсталляция.

Практическая значимость проекта заключается в возможности использования приложения в образовательных учреждения с целью автоматизации процесса контроля за учащимися в процессе работы за ПК.

Основными пользователями данной программы будут учителя, работающие в кабинете, где находится большое количество ПК.

Данные для приложения будут взяты на основе иностранных аналогов и документации отделения.

Теоретическими основами и методами решения поставленных задач будут методологии проектирования функциональности приложения, структуры БД, особенности реализации программного кода с помощью конкретного языка программирования, принципы и алгоритмы применения инструментальных средств на различных этапах разработки программного продукта.

В техническом проекте будет произведен анализ предметной области и представлены этапы проектирования архитектуры, интерфейса и функциональности реализуемой информационной системы, включая разработку логической модели БД. Также в нем будет произведен выбор инструментов для физической реализации построенных моделей данных и информационной системы.

Наиболее удобной для реализации поставленной задачи является свободная реляционная система управления базами данных. В соответствии с этим будут подобраны система управления БД (СУБД) и инструментальные средства реализации интерфейса приложения, которое будет работать с данной БД.

Для проектирования интерфейса и функциональности приложения, а также архитектуры информационной системы будут использоваться UML-диаграммы.

В рабочем проекте представлены достигнутые практические результаты, в частности, описаны физическая модель данных и разработанные классы, реализующие функциональность приложения. Также в нем описаны процессы установки информационной системы и особенности ее эксплуатации.

1 ТЕХНИЧЕСКИЙ ПРОЕКТ

1.1 Анализ предметной области

Отделение является основным учебно-научным структурным подразделением средних учебных заведений, которое проводит учебную, методическую и научно-исследовательскую деятельность по одной или нескольким родственным специальностям.

На отделение возлагаются следующие задачи и обязанности:

* реализация федеральных государственных образовательных стандартов (ФГОС СПО), целевая ориентация учебного процесса на формирование общих и профессиональных компетенций, установленных ФГОС СПО;
* обеспечение качественного образования, повышение ответственности за результаты образовательной деятельности;
* внедрение в образовательный процесс современных подходов к обучению и воспитанию обучающихся;
* усиление работы по сохранению контингента и адаптации обучающихся.

Реализация данных задач предполагает планирование и управление процессом проведения лекций, лабораторных, практических, семинарских и других видов учебных занятий, руководства учебной и производственной практикой, исследовательскими работами, курсовыми и дипломными проектами (работами), проведения экзаменов и зачётов на промежуточной аттестации, а также итоговой аттестации.

Работа учащихся за учебными местами требует жесткого контроля. Не менее важным является фиксация, хранение и обработка данных об аппаратном и программном обеспечении.

Сопровождение данной деятельности требует оперирование большими объемами информации. Поэтому существует практическая потребность в автоматизации данных функций для облегчения процесса накопления, хранения, контроля и обработки информации.

Разрабатываемое приложение будет автоматизировать и упрощать следующие процессы:

* хранение данных об аппаратном обеспечении каждого ПК в локальной сети;
* возможность редактирования информации об сотруднике отделения;
* добавление новых сотрудников в систему;
* обеспечение контроля во время учебной деятельности;

Разрабатываемое приложение будет оперировать данными об аппаратном и программном обеспечении всех ПК в локальной сети, а также данными о сотрудниках отделения.

1.2 Постановка задачи

Необходимо разработать клиент-серверную информационную систему для контроля, поддержки и мониторинга за учащимися отделения информационных технологий МГГТК ФГБОУ ВО «АГУ», а также создать документационное сопровождение приложения, включающее в себе технический и рабочие проекты.

Необходимые данные должны храниться в удалённой БД.

Данная программа должна будет содержать все нужные сведения о преподавателях и ПК в локальной сети, иметь удобный и понятный пользователю графический интерфейс, обеспечивать необходимый уровень защиты данных от несанкционированного использования.

Программный продукт должен включать в себя две части:

* информационно-справочная часть, которая будет отображать необходимую информацию;
* операционная часть, в которой преподаватели и сотрудники отделения смогут изменять необходимую информацию и контролировать учебный процесс.

Приложение должно обеспечивать выполнение следующих функций:

* авторизация;
* отображение меню;
* просмотр, добавление и редактирование информации о сотрудниках,
* хранить и передавать информацию об аппаратном и программном обеспечении;
* удаленная поддержка и работа с ПК, работающим в локальной сети;
* взаимодействие с данными, хранящимися в удалённой БД.

Данные будут предоставлены в виде графической и текстовой информации, загруженной из целевой БД и файлов.

Входными данными в приложении будут данное сотрудников и сведения об аппаратном и программном обеспечении ПК.

Выходными данными в приложении будут информация об действиях на ПК в локальной сети.

Приложение будет состоять из окон и страниц, на которых будет располагаться вся необходимая пользователю информация.

Способ решения поставленных задач зависит от выбора средств разработки.

Тестирование программы будет проводиться путем выбора пунктов меню и проверки корректности получаемого отклика программы.

1.3 Проектирование базы данных

Первым шагом на пути реализации проекта будет создание БД.

В широком смысле БД – это хранилище элементов данных, называемых «записями», имеющее определенную физическую и логическую структуру, а также программный интерфейс, позволяющий пользователю взаимодействовать с сохраняемой в ней информацией. Чтобы универсальным способом извлекать из нее группы записей, обрабатывать их, изменять и удалять, требуются специальные программы, которые называются СУБД [2].

Наиболее удобной для реализации поставленной задачи является cвободная реляционная система управления базами данных, которая использует декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

В теории баз данных эти таблицы называют отношениями (relations) –

поэтому и базы данных называются реляционными. Отношение – это

математический термин. При определении свойств таких отношений

используется теория множеств. В терминах данной теории строки таблицы

будут называться кортежами (tuples), а колонки – атрибутами. Отношение

имеет заголовок, который состоит из атрибутов, и тело, состоящее из

кортежей. Количество атрибутов называется степенью отношения, а

количество кортежей – кардинальным числом.

Реляционная модель данных – данные представлены посредством строк в таблицах.

В данном дипломном проекте предметной областью является хранение, контроль и поддержка ПК в локальной сети. БД будет предназначена для хранения данных о сотрудниках отделения, а также информацию о действиях, происходящих на каждом ПК.

Первый этап процесса проектирования БД заключается в создании концептуальной модели данных.

Компонентами концептуальной модели являются сущности и взаимосвязи. Она служит средством общения между различными пользователями, и поэтому разрабатывается без учета особенностей физического представления данных. При проектировании концептуальной модели все усилия разработчика должны быть направлены в основном на структуризацию данных и выявление взаимосвязей между ними без рассмотрения особенностей реализации и вопросов эффективности обработки. Проектирование концептуальной модели базируется на основе анализа решаемых в предметной области задач по обработке данных [2].

На основе анализа предметной области и пункта 1.2 были выделены основные сущности:

* пользователи (идентификатор, логин, пароль, имя, фамилия, отчество, идентификатор роли, идентификатор кабинета);
* кабинеты (идентификатор, название кабинета);
* роли (идентификатор, наименование);
* специальности сотрудников (идентификатор, название специальности);
* названия действий (идентификатор, название)
* статусы действий (идентификатор, название статуса)
* действия в кабинетах (идентификатор, время действия, идентификатор кабинета, идентификатор статуса действия, идентификатор названия действия)

Затем была разработана концептуальная модель БД (рисунок 1) и логическая модель, которая отражает связи между сущностями (рисунок 2).

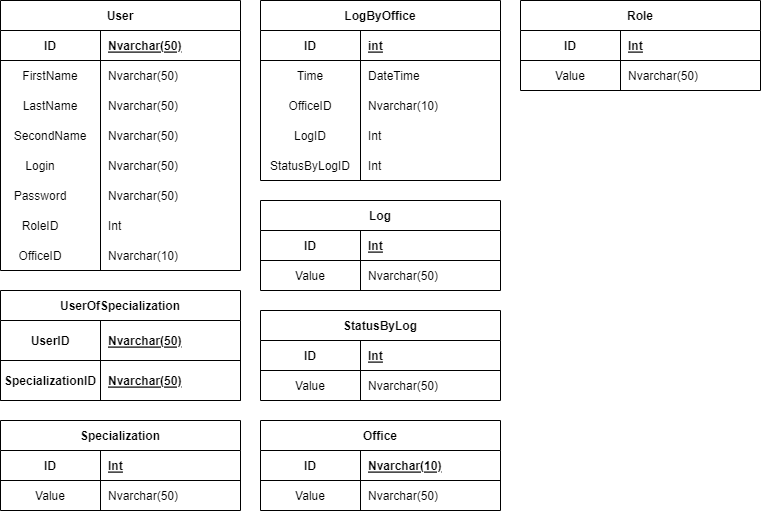


Рисунок 1 – Концептуальная модель БД

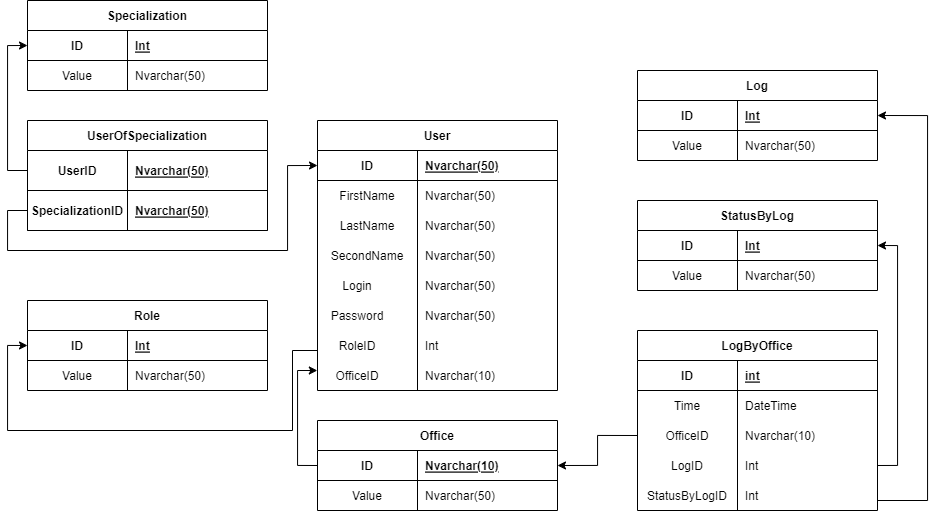


Рисунок 2 – Логическая модель БД

1.4 Проектирование функциональности приложения

Для описания основных функций проектируемой информационной системы воспользуемся методологией UML и построим диаграмму Use Case (приложение А). Она определяет:

* основных пользователей системы, демонстрируя их иерархию;
* основные функции системы и их взаимосвязи;
* группирование функций системы по доступности определенным пользователям;
* зависимость функционирования проектируемой системы от внешних приложений.

На ней видно разграничение уровней доступа к реализованным функциям и основные сценарии использования представленной информационной системы, предусмотренные при её разработке.

Обучающемуся, который является одним из пользователей системы доступны функции просмотра характеристик ПК, закрытие программы, а также настройка приложения.

Секретарю, кроме функции настройки приложения, доступны функции редактирования информации о сотрудниках, а также возможность создания нового сотрудника.

Преподавателю, опять же, доступна функция настройка приложения, а также функции с выбранным ПК: запросить информацию, выключить. Также учитель может сформировать отчеты по всем ПК за день.

При выполнении всех функций система взаимодействует с удаленной БД, расположенной на сервере. Взаимодействие происходит через стандартизированный интерфейс, предоставляемый СУБД.

1.5 Проектирование структуры приложения

При создании информационной системы одним из самых важных аспектов является её архитектура. Она представляет собой концептуальное видение структуры будущих функциональных процессов и технологий на системном уровне и во взаимосвязи.

Так как разработать нужно клиент-серверную информационную систему, то необходимо определиться с её архитектурой. Основными клиент-серверными архитектурами, реализуемыми в настоящее время являются:

* выделенный сервер;
* активный сервер;
* сервер приложений [5].

В архитектуре «выделенный сервер» средства управления БД и БД размещены на машине-сервере, а функциональная обработка данных выполняется на стороне клиента. Сервер только предоставляет данные по запросу клиента. В данной архитектуре предъявляются повышенные требования к аппаратному и программному обеспечению клиента, так как вся бизнес-логика реализуется на стороне клиента, а сервер выполняет только функции доступа к данным и их защиты.

В архитектуре «активный сервер» функции бизнес-логики разделяются между клиентской и серверной частями. Общие или критически значимые функции оформляются в виде хранимых процедур, включаемых в состав БД. Кроме этого, вводится механизм отслеживания событий БД – триггеров, также включаемых в состав БД. При возникновении соответствующего события (обычно изменения данных) СУБД вызывает для выполнения хранимую процедуру, связанную с триггером, что позволяет эффективно контролировать изменение БД. Хранимые процедуры и триггеры могут быть использованы любыми клиентскими приложениями, работающими с БД. Это снижает дублирование программных кодов и исключает необходимость компиляции каждого запроса.

В данной архитектуре снижаются требования к аппаратному и программному обеспечению клиента, но возрастают, применительно к серверу.

В архитектуре «сервер приложений» снижение уровня требований к ресурсам клиента достигается за счет введения промежуточного звена – сервера приложений, на который переноситься значительная часть программных компонентов управления данными и большая часть бизнес-логики. При этом серверы БД обеспечивают исключительно функции СУБД по ведению и обслуживанию БД [4].

Так как клиентская часть информационной системы будет устанавливаться на компьютерах, не обладающих повышенными характеристиками производительности процессора, оперативной и постоянной памяти, то выбор был остановлен на архитектуре «активный сервер».

Сервер БД будет осуществлять целый комплекс действий по управлению данными. Основными его обязанностями будут:

* поддержка ссылочной целостности данных согласно определенным в БД правилам;
* хранение и резервное копирование данных;
* выполнение пользовательских запросов на выбор и модификацию данных и метаданных, получаемых от клиентских приложений, функционирующих на персональных компьютерах локальной сети;
* обеспечение авторизованного доступа к данным на основе проверки прав и привилегий пользователей;

В качестве рабочих мест пользователя будут использоваться персональные компьютеры секретаря и преподавателей, расположенные в локальной сети отделения, что позволяет не отказываться от привычной рабочей среды.

С целью удовлетворения предъявляемых к клиентскому приложению требований, приведенных в разделе 1.2, и представленных на Use Case диаграмме функций, разрабатываемое приложение будет состоять из следующих страниц:

* авторизации – будет запускаться перед отображением главного окна для идентификации пользователя;
* главная – будет содержать навигационные кнопки для перехода на страницы, настроек, редактирования, добавления, страницы с графиком работ за день, страница, с характеристиками выбранного ПК, страница со списком ПК;
* список компьютеров – страница где выведены все зарегистрированные ПК в системе;
* добавления и редактирования – страница для добавления и редактирования сотрудников в системе;
* график – окно для графика действий проведенных за день (учебный план, группы, обучающиеся, приказы, новости).

Необходимые данные будут хранится в БД.

Проектируемая структура информационной системы представлена на рисунке 3.



Рисунок 3 – Схема проектируемой структуры информационной системы

Проектируемая структура клиентского приложения представлена на рисунке 4.



Рисунок 4 – Схема проектируемой структуры клиентского приложения

Тестирование программы планируется проводить путём выбора каждого пункта меню и проверки корректности получаемого отклика программы.

1.6 Выбор средств реализации

Интерфейс и функциональность клиентского приложения могут быть написаны на следующих языках программирования: С++, С#, Python, Java. Все они позволяют работать с БД, файлами, текстом и графикой, с использованием объектно-ориентированного подхода.

Выбор был остановлен на языке программирования C#, с использованием .NET 5, и интегрированной среде разработки Microsoft Visual Studio 2019.

Visual Studio обеспечивает поддержку новейших языковых функций на протяжении всего процесса разработки. Данная среда разработки поддерживает C#, Visual Basic, C++, TypeScript, F# и даже сторонние языки, например, JavaScript [3].

Язык С# актуален потому, что:

* позволяет более рационально создавать популярные на сегодня типы приложений;
* интегрировал в себе преимущества языков Java и С++, при этом в объединённом языке исключены некоторые спорные директивы, макросы, отменены глобальные переменные;
* является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;
* является мощным объектным языком с возможностями наследования и универсализации;
* мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из БД и других хранилищ данных [4].

.NET 5 – следующий шаг в .NET Core. Проект призван улучшить .NET в нескольких ключевых аспектах:

* расширить возможности .NET за счёт лучших наработок из .NET Core, .NET Framework, Xamarin и Mono;
* создать единые исполняющую среду и фреймворк, которые можно использовать везде, с одинаковым поведением в runtime и опытом разработки;
* собрать продукт из единой кодовой базы, над которой разработчики (из Microsoft и сообщества) могут вместе работать и расширять её, что позволит улучшить все возможные сценарии.

Этот новый проект и направление полностью изменят ситуацию с .NET. Благодаря .NET 5 код и файлы проектов будут выглядеть единообразно, вне зависимости от типа создаваемого приложения. Из каждого приложения будет доступ к той же исполняющей среде, тем же API и возможностям языка, включая новые улучшения производительности, которые внедряются в corefx практически ежедневно.

Сохранилось всё, что было улучшено в .NET Core:

* open source и ориентированность на сообщество GitHub;
* кроссплатформенная реализация;
* поддержка использования специфических платформозависимых возможностей, таких как Windows Forms и WPF под Windows, а также нативных привязок (bindings) к каждой нативной платформе из Xamarin;
* высокая производительность;
* side-by-side инсталляция;
* маленький размер файлов проектов (SDK-стиль);
* интерфейс командной строки (CLI) с широкими возможностями;
* интеграция с Visual Studio, Visual Studio for Mac и Visual Studio Code.

Нововведения:

* возможность вызова кода Java из .NET 5 будет доступна на всех платформах;
* вызов кода Objective-C и Swift из .NET 5 будет поддерживаться в нескольких операционных системах;
* CoreFX будет расширен, чтобы поддерживать статическую компиляцию .NET (ahead-of-time – AOT), для уменьшения потребления ресурсов (footprints) и поддержки большего количества операционных систем [7].

Для проектирования физической модели БД будет использоваться СУБД Microsoft SQL Server и утилита SSMS. Данная СУБД является высокоэффективной и надежной, в ней есть возможности для использования ее в любых критичных бизнес-приложениях.

Основные преимущества Microsoft SQL Server:

* масштабируемость и производительность;
* база данных менее уязвима;
* инструменты бизнес-аналитики с поддержкой самообслуживания;
* Базовые выпуски SQL Server обладают расширенными функциями обеспечения безопасности;
* Стандартные и корпоративные выпуски Oracle обеспечивают лишь базовую безопасность;
* Центр обновления Windows позволяет снизить риски в области безопасности, а также свести к минимуму простой системы во время установки пакетов исправлений.

Для создания инсталляции была выбрана Inno Setup – система создания инсталляторов для Windows-программ с открытым исходным кодом. Она представляет собой среду для настройки проекта, создания сценария и конечного релиза дистрибутива.

Ключевыми особенностями Inno Setup являются:

* поддержка всех современных версий Windows: 10, 8, 7, 2008 R2, Vista. Поддерживает установку 64-битных программ на 64-битных выпусках. Также поддерживает 64-разрядные процессоры;
* поддержка создания одиночных EXE для облегчения установки и распространения программ через сеть Интернет;
* стандартный интерфейс мастера установки в стиле Windows 2000/XP;
* возможность выбора типа установки, например: полный, минимальный, выборочный;
* имеет встроенную поддержку DEFLATE, bzip2 и 7-Zip LZMA/LZMA2 сжатий;
* установщик умеет сравнивать версии файлов, заменять встроенные файлы, устанавливать разделяемые файлы, регистрировать DLL/OCX библиотеки и устанавливать шрифты;
* позволяет создавать ярлыки в меню «Пуск» и на «Рабочем столе»;
* позволяет создавать записи в реестре и .ini-файлы;
* поддерживает создание многоязычных инсталляторов программ;
* поддерживает Unicode и языки с направлением письма справа налево;
* поддерживает установку пароля и шифрование инсталляторов программ [10].

1.7 Требования к программному обеспечению и техническим средствам

Программное обеспечение должно удовлетворять всем реализуемым функциям системы, а также иметь определенный набор средств для организации всех требуемых процессов обработки данных, которые позволяют своевременно выполнять все функции во всех режимах функционирования системы.

Системное программное обеспечение должно реализовывать комплекс задач управления. Для корректной работы разрабатываемой программы необходимо, чтобы клиентский компьютер, с целью удовлетворения минимальным требованиям, был оснащен:

* операционной системой Windows 10 / 11;
* лицензионным программным обеспечением Microsoft Office 2019 +;
* .NET 5 +;

Сервер должен быть оснащён:

* операционной системой Windows Server 2019 / Ubuntu OS / CentOS Server / Oracle Linux Server;
* Microsoft SQL Server 2019;
* SQL Server Management Studio.

При выборе технических средств, применяемых для функционирования разрабатываемой программы, должны учитываться следующие требования:

* выбор технических средств должен обеспечивать рациональное соотношение между затратами на создание системы и достигаемым эффектом;
* технические параметры системы управления не должны налагать ограничения на регламент технологического процесса функционирования системы.

Для реализации вышеуказанных требований со стороны клиентского оборудования необходим следующий состав технических средств:

* центральный процессор фирмы INTEL или AMD с тактовой частотой не менее 1 ГГц;
* стабильное Ethernet-соединение.
* оперативная память не менее 2 ГБ;
* дисплей с разрешением не менее 1024x768 точек;
* 130 МБ свободного места на диске;
* клавиатура;
* манипулятор типа «мышь».

Серверное оборудование должно соответствовать следующим аппаратным требованиям:

* центральный процессор фирмы INTEL или AMD с тактовой частотой не менее 4 ГГц;
* стабильное Ethernet-соединение.
* оперативная память не менее 8 ГБ;
* 150 ГБ свободного места на диске.

Для разработки, отладки и расширения реализуемой информационной системы необходим следующий набор технических средств:

* операционная система Windows 10 / 11;
* .NET 5+, включая компоненты для разработки универсальных приложения (Universal Windows Platform);
* Visual Studio 2019 +;
* Microsoft SQL Server 2019;
* SQL Server Management Studio.

ПРИЛОЖЕНИЕ А

Use Case диаграмма проектируемой информационной системы

