1. Изучение предметной области

Перед началом разработки системы необходимо ее спроектировать. Для этого были реализованы следующие диаграммы:

- диаграмма прецедентов;

- диаграмма последовательности;

- диаграмма развертывания;

- «ER»-диаграмма;

- диаграмме User Story Mapping.

Все диаграммы разрабатывались в онлайн редакторе диаграмм. Диаграмма прецендентов представлена в соответствии с рисунком 1.

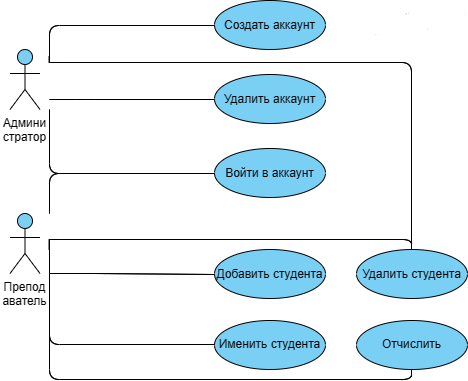


Рисунок 1 – Диаграмма прецендентов

На диаграмме прецендентов изображены возможные действия пользователей в системе. В системе существует два вида пользователей – администратор и персонал. Администратор отвечает за аккаунты сотрудников, имея возможность изменять и удалять их, а также добавлять новые.

Сотрудник имеет возможность авторизоваться. После авторизации он может работать с данными студентов, добавлять новых, изменять существующих, а также удалять и отчислять.

Диаграмма последовательности представлена в соответствии с рисунком 2.

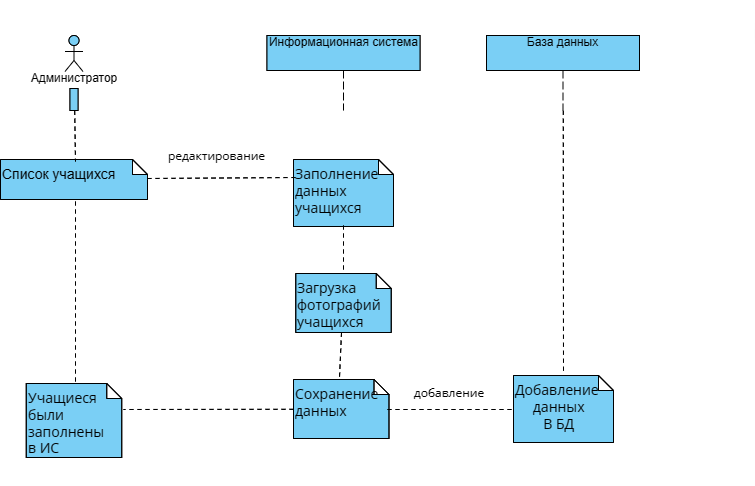


Рисунок 2 – Диаграмма последовательности

На данной диаграмме изображена последовательность действий сотрудника для добавления учащегося в систему. В начале он заполняет данные об учащемся, после чего нажимает на кнопку, сохраняя данные в базе данных.

Диаграмма развертывания представлена в соответствии с рисунком 3.

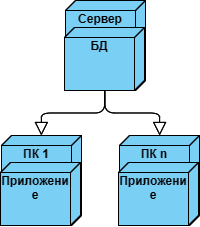


Рисунок 3 – Диаграмма развертывания

На данной диаграмме изображено развернутое подключение компьютеров с установленной на них системой к удаленному серверу с базой данных.

ER-диаграмма представлена в соответствии с рисунком 4.

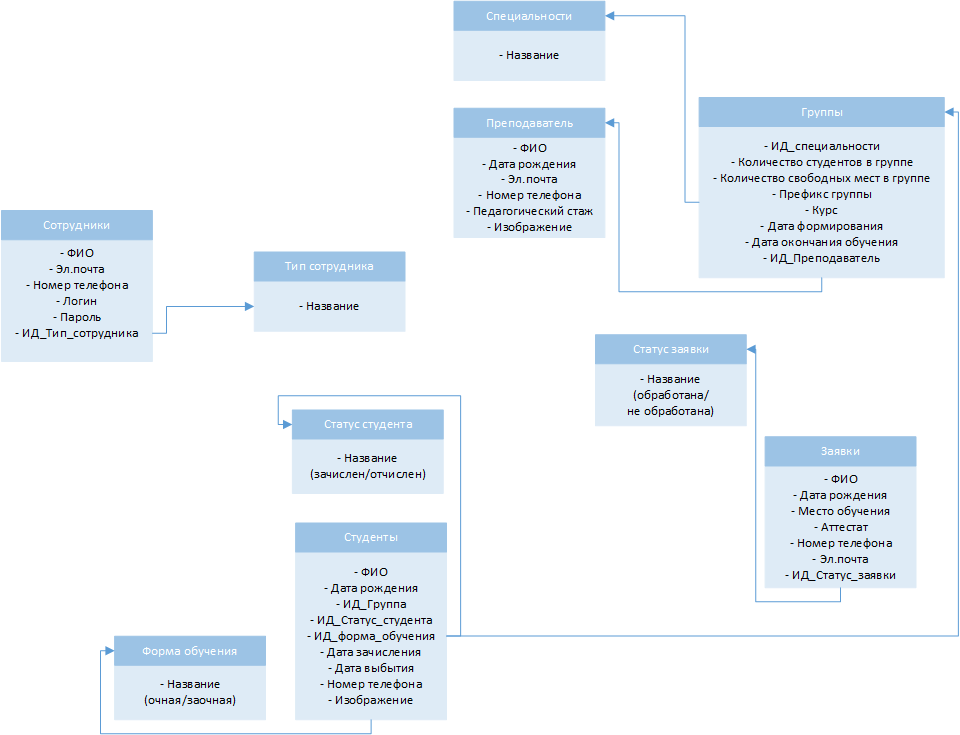


Рисунок 4 – ER-диаграмма

На ER-диаграмме представлены сущности базы данных, интегрированной в систему. На диаграмме представлены следующие сущности:

- «Группы»;

- «Форма\_обучения»;

- «Студенты»;

- «Статус\_студента»;

- «Статус\_заявки»;

- «Заявки»;

- «Специальность»;

- «Преподаватель»;

- «Тип\_сотрудника»;

- «Сотрудники».

Диаграмма «User Story Mapping» представлена в соответствии с рисунком 5.

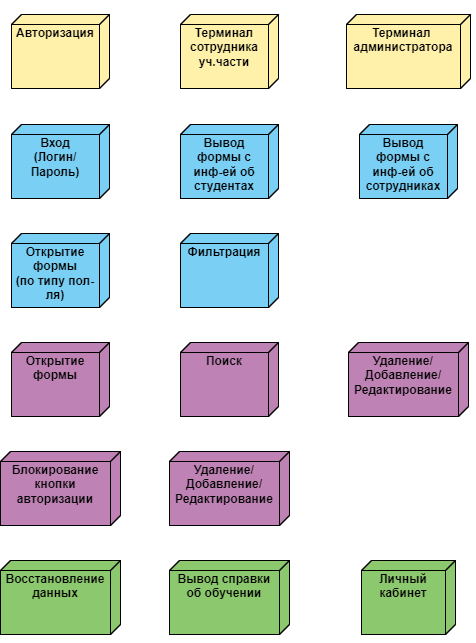


Рисунок 5 – Диаграмма «User Story Mapping»

На данной диаграмме представлен функционал приложения по версиям разработки.

Сбор данных для дипломного проектирования

Предметной областью является колледж. Колледж - среднее специальное учебное заведение, являющееся в системе образования более высокой ступенью после средней школы и более низкой - перед высшим учебным заведением.

Учёба в колледже похожа на учёбу в высших учебных заведениях. Обучают студентов по семестрам, есть лекции, семинары, сдаются сессии. Среднее профессиональное образование в колледже получается за три года, а программа углубленной подготовки на четвертом году. Поступить в колледж можно после девяти или одиннадцати классов или по диплому о начальном или среднем профессиональном образовании. Колледжи предлагают большое разнообразие специализаций - технических, творческих или узкоспециализированных. По окончанию выдаётся диплом о среднем профессиональном образовании, присваивается квалификация по изучаемой специальности.

Функционал системы основан на приеме абитуриентов и учете учащихся в колледже. Для принятия абитуриента на обучения, он должен отправить следующие данные о себе:

- фамилию;

- имя;

- отчество;

- дату рождения;

- специальность;

- место учебы;

- норме телефона;

- аттестат.

После принятия абитуриента, его зачисляют на специальность, которую он указал при поступлении. Для принятия абитуриента, он должен пройти лимит среднего балла в аттестате.

Таким образом колледж занимается обучение учащихся профессиональному образования, а для поступления абитуриенту необходимо указать свои личные данные, и пройти по среднему баллы в аттестате.

Выбор программных средств

Интегрированная среда разработки Visual Studio - это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

SQL Server - это система управления базами данных, в работе с которой используется язык программирования SQL. СУБД SQL Server используются для создания, размещения, хранения и управления реляционными базами данных на специальных серверах или в облаке.

СУБД SQL server используются для создания, размещения, хранения и управления реляционными (табличными) базами данных на специальных серверах или в облаке. Они работают через настольные приложения и web-сайты. К основным преимуществам их функционирования относятся:

- высокоскоростной доступ к данным, обеспечиваемый надежной клиент-серверной архитектурой СУБД;

- простота работы и администрирования, обусловленные понятной структурой языка программирования SQL;

- безопасность хранения информации в БД - благодаря возможности шифрования данных и резервного копирования.

Специфика работы сервера базы данных SQL server заключается в транзакционной обработке данных. Это означает, что по каждому запросу от СУБД обрабатывается и сохраняется небольшое количество информации.

Выбор языковых средств разработки.

C# - объектно-ориентированный язык программирования общего назначения. Разработан в 1998—2001 годах группой инженеров компании Microsoft язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

SQL - декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

Является, прежде всего, информационно-логическим языком, предназначенным для описания, изменения и извлечения данных, хранимых в реляционных базах данных. В общем случае SQL (без ряда современных расширений) считается языком программирования неполным по Тьюрингу, но вместе с тем стандарт языка спецификацией SQL/PSM предусматривает возможность его процедурных расширений.

Этапы разработки.

Анализ требований и составление технического задания. Необходимо определить функциональность программы, требования к интерфейсу, базе данных, а также установить технические ограничения и требования.

Проектирование архитектуры программного продукта. На основе технического задания разрабатывается диаграмма классов и схема базы данных.

Написание кода. На этом этапе происходит написание кода, реализующий требования, определенные на предыдущем этапе. В данном случае, это может быть код, обеспечивающий возможность добавления, удаления и редактирования информации о студентах.

Релиз программного продукта. Готовый продукт устанавливается на сервере или на компьютерах пользователей.

Поддержка и развитие. После релиза разработчики продолжают тестирование и устранение возможных ошибок, а также добавление новой функциональности и улучшение программы в соответствии с требованиями пользователей.

Интерфейс проекта.

Интерфейс проекта выполнен в приятных и спокойных тонах, что создает комфортное и уютное ощущение при работе с ним. Без лишнего, пользователь легко ориентируется в интерфейсе и быстро находит нужные функции и элементы. Визуально проект выполнен в сдержанных цветах, что создает впечатление профессиональности и надежности.

Каждый элемент интерфейса выполнен в соответствии с лучшими практиками дизайна, что сделало его легким в использовании и приятным для глаз. В результате, интерфейс проекта создает положительное впечатление и делает процесс работы с ним максимально эффективным и комфортным.

Интерфейс необходимо сделать под функционал системы, описанный в техническом задании. Система состоит из 6 окон.

Функционал.

Окно под название «MainWidnow», в котором осуществляется вход в систему по почте и паролю, а также восстановление пароля через сообщение на почту. Окно авторизации представлено в соответствии с рисунком 6.

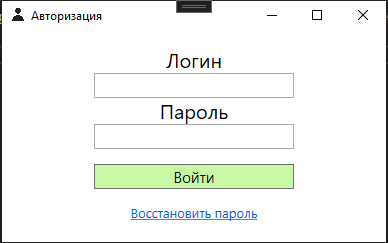


Рисунок 6 – Окно авторизации

Восстановление пароля происходит в отдельном окне, «AccRecovery». Окно восстановления представлен в соответствии с рисунком 7.

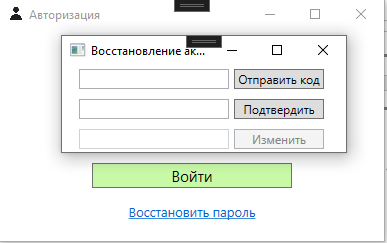


Рисунок 7 – Окно восстановления пароля

Окно под названием «winmain\_sotrdnik», в котором в «ListView» отображаются данные о студентах из базы данных, а также имеется строка поиска, кнопка добавления нового студента и кнопка обновления данных в «ListView». Окно отображения студентов представлен в соответствии с рисунком 8.

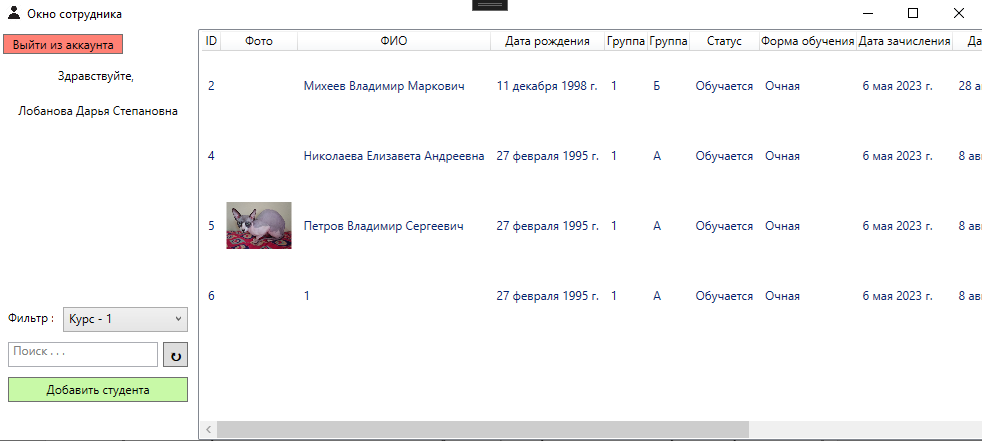


Рисунок 8 – Окно сотрудника

При нажатии на кнопку добавить в окне «winmain\_sotrdnik», открывается следующее окно, под название «winRealaddStud». В нем осуществляется добавление нового пользователя. Окно добавления представлено в соответствии с рисунком 9.

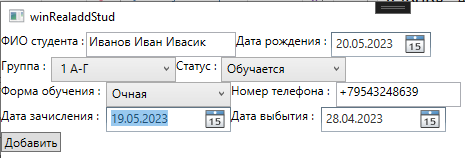


Рисунок 9 – Окно добавления

Пример добавления представлен в соответствии с рисунком 10.



Рисунок 10 – Пример добавления

При двойном нажатии по студенты, открывается окно редактирования. Данное окно идентично окну добавления, за исключением того, что все данные заполняются при его открытии данными того студента, который был выбран. Пример окна редактирования представлен в соответствии с рисунком 11.

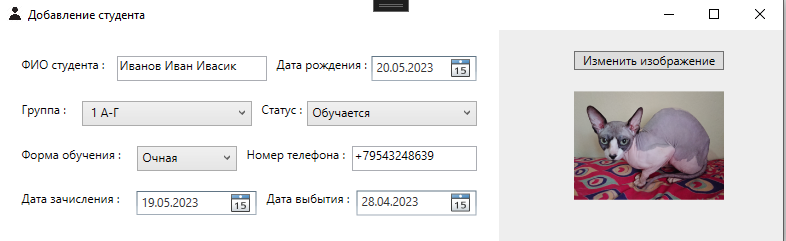


Рисунок 11 – Окно редактирования

Также при авторизации, если у сотрудник является администратором, его перенаправляет на окно администратора, в котором он может просматривать данные о пользователях. Пример входа администратора представлен в соответствии с рисунком 12.

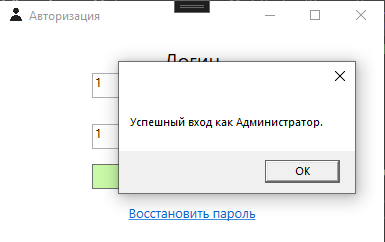


Рисунок 12 – Пример входа администратора

Окно входа представлен в соответствии с рисунком 13.

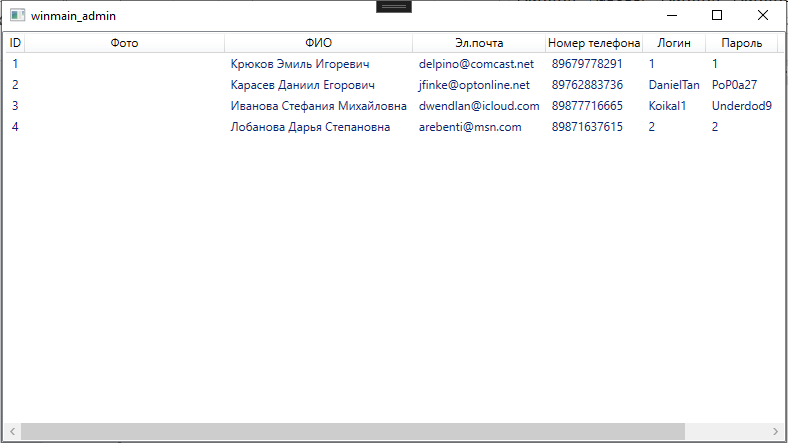


Рисунок 13 – Пример окна администратора

Таким образом был реализован функционал работы со студентами и сотрудниками, а также вход по логину или почте и паролю и его восстановление через сообщение на почту.

Техническое задание

Для реализации проекта необходимо создать следующее:

- физическая модель базы данных. Количество сущностей должно быть на ваше усмотрение, но не менее 10. Соблюдайте общие требования к реализации базы данных. В базе данных должна храниться информация о студентах, группах, сотрудниках, преподавателях и заявках на поступление;

- авторизация. При запуске настольного приложения перед потенциальным пользователем открывается окно авторизации. Для успешной авторизации пользователю необходимо ввести логин/почту и пароль для существующей учетной записи;

- форма со студентами. На форме со студентами выводится информация об студентах в виде таблицы. Необходимо реализовать фильтрацию с помощью выпадающих списков по группе, статусу обучения (обучается или окончил), дате рождения, по форме обучения;

- форма с группами. Форма с группами выглядит аналогичным образом, как и форма со студентами, только данные должны быть представлены в виде списка. Необходимо реализовать форму добавления новой группы;

- форма с педагогами. На данной форме отображается список педагогов с фотографиями в виде списка с фотографией педагога;

- форма с заявками. Заготовить 10 заявок абитуриентов для тестирования функциональной части приложения.

Меры безопасности.

Для безопасности данных в системе имеется вход по логину и паролю. Для тех, кто забыл свой пароль существует возможность его восстановления через сообщение, приходящее на почту.