

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных Технологий
институт
Информационные Системы
кафедра

УТВЕРЖДАЮ
Зав. кафедрой ИС
_____ С. А. Виденин
подпись инициалы, фамилия
« ____ » _____ 2016 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 Информационные системы и технологии

Разработка автоматизированной информационной системы управления
общезитием

Руководитель	_____	<u>Е. В. Головчанская</u>
	подпись, дата	
Выпускник	_____	<u>А.М.Трухан</u>
	подпись, дата	
Нормоконтролер	_____	<u>Ю. В. Шмагрис</u>
	подпись, дата	

Красноярск 2016

РЕФЕРАТ

Выпускная квалификационная работа по теме Разработка автоматизированной информационной системы управления общежитием содержит 79 страниц текстового документа, 59 рисунков, 37 использованных источников.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ, DESKTOP-ПРИЛОЖЕНИЕ, БАЗА ДАННЫХ, ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС.

Целью разработки автоматизированной информационной системы управления общежитием является:

- Повышение качества работы управляющего общежитием;
- Сокращение временных затрат на работу со всеми аспектами общежития;
- Создание единой базы данных общежития;
- Автоматизация процесса оповещения о задолженностях за проживание;
- Автоматизация процесса слежения за запланированными мероприятиями;
- Автоматизация процесса отправки сотрудников в определённую комнату для устранения неполадок.

В ходе создания информационной системы были поставлены следующие задачи:

- Провести анализ предметной области;
- Составить техническое задание;
- Осуществить выбор средств и инструментов для реализации информационной системы;
- Реализовать информационную систему с помощью программных средств.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
Глава 1 Общие сведения.....	7
1.1 Структура информационной системы	7
1.2 Средства разработки	11
1.3 Базы данных.....	15
1.4 Способ доступа к базе данных.....	17
1.5 Вывод по главе 1	17
Глава 2 Проектирование ИС	19
2.1 Формирование требований	19
2.1.1 Требования к функциональности	19
2.1.2 Требования к интерфейсу.....	20
2.1.3 Технические требования.....	20
2.2 Проектирование структуры приложения и разработка пользовательского интерфейса.....	21
2.3 Проектирование функциональной части приложения.....	33
2.4 Проектирование структуры взаимодействия базы данных с приложением	40
2.5 Проектирование базы данных	41
2.6 Вывод по главе 2	46
Глава 3 Описание и сравнение ИС	47
3.1 Сценарий работы пользователя с приложением.....	47
3.2 Информационная безопасность ИС	65
3.3 Сравнение ИС	67
3.4 Оценка качества ИС.....	70
3.5 Внедрение	73
3.6 Вывод по главе 3	73
ЗАКЛЮЧЕНИЕ	74
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	75

ВВЕДЕНИЕ

В современном мире, в любой организации особое значение имеет человек, который управляет определённым видом деятельности. Общежития не являются исключением.

Управляющий общежитием — профессия, связанная с работой и обработкой больших объёмов информации. В настоящее время, в большинстве общежитий коменданты, администраторы и тьюторы управляют этой информацией при помощи старого проверенного метода — в бумажном варианте. Но такой способ уже не подходит для современного компьютеризированного общества, так как человек физически не может уследить за всеми аспектами своей работы, а ошибки могут привести к необратимым последствиям. Даже если в общежитие есть компьютерная поддержка и специализированное программное обеспечение, то это программное обеспечение не всегда отвечает требованиям самих работников и не содержит в себе полного спектра функций.

Безошибочно быстрая и удобная работа автоматизированных информационных систем напрямую влияет на эффективность деятельности любой современной организации или предприятия.

Исходя из всего вышесказанного, целью данной бакалаврской работы является разработка автоматизированной информационной системы управления общежитием, с помощью которой будет осуществляться взаимодействие управляющего с различными информационными аспектами его деятельности.

Для достижения поставленной цели были выделены и выполнены следующие задачи:

- Выбор средств и среды разработки;
- Разработка пользовательского интерфейса проекта;
- Реализация проекта с помощью программных средств;

Основной задачей, проектируемой информационной системы является упрощение и автоматизация работы с информационной базой общежития.

Этот продукт представляет собой базу данных под управлением приложения на языке Си#.

Данная информационная система будет использоваться потенциальными управленцами общежитий. Вход в систему будет производиться после регистрации, а в дальнейшем после авторизации пользователя непосредственно в самом приложении.

В качестве направлений автоматизации информационной системы можно выделить такие аспекты:

- Возможность отправки студентам автоматических сообщений о задолженностях за проживание;
- Автоматическое напоминание о запланированных мероприятиях при входе в приложение;
- Автоматическое списывание средств за проживание;
- Удобство работы с системой;
- Отправка сообщений сотрудникам для уведомления о назначенной работе.

Коменданты и тьюторы получают возможность управления всеми аспектами общежития, не отходя от своего рабочего места. Это позволит сильно сократить время не только самого управляющего общежитием, а также сотрудников и студентов.

В первой главе данной работы излагаются основные сведения об информационной системе, описывается общая структура информационной системы, даются определения основным понятиям. Большая часть главы посвящена выбору средств разработки и обзору используемых при разработке технологий. Так же в этой главе рассмотрены понятия баз данных и варианты взаимодействия с ней различных приложений.

Во второй части проекта выявлены требования к системе, которые послужили основой при ее проектировании. В этой части была создана

структура проекта, разработан интерфейс, спроектирована функциональная часть, создана структура базы данных и проведено моделирование.

В третьей главе приведен сценарий работы пользователя с системой, рассмотрены средства обеспечения информационной безопасности системы, проведён сравнительный анализ информационной системы с другими системами такого направления. Ещё был проведен анализ критериев качества системы и описаны основные этапы внедрения системы.

Глава 1 Общие сведения

1.1 Структура информационной системы

Сама по себе информационная система — это прикладная программная система, целью которой является поиск, сбор, обработка, хранение информации и своевременное обеспечение соответствующих людей данными в рамках некой предметной области. [1]

Структура информационных систем состоит из элементов, имеющих взаимосвязь и называемых «подсистемами».

Таким образом, подсистема является частью любой системы, в основе которой лежит общий признак, и которая обеспечивает эффективное функционирование.

Общую структуру подсистем можно проанализировать вне того, в какой предметной области она применяется. Исходя из этого, говорят о структурном признаке классификации, а сами подсистемы обычно называют обеспечивающими. Следовательно, структура любой информационной системы — это ничто иное как совокупность обеспечивающих подсистем. (рисунок 1). Как правильно, в состав данных подсистем входит техническое, информационное, математическое, программное, организационное и правовое обеспечение. [2]



Рисунок 1 - Структура информационной системы как совокупность обеспечивающих подсистем

Целью создания подсистем информационного обеспечения является формирование и выдача некоторого объема информации максимально своевременно. Именно благодаря этому принимаются верные и безотлагательные управленческие решения.

Под термином «информационное обеспечение» понимается совокупность системы классификации и кодирования, а также комплекс схем информационных потоков, унифицированных систем документации и методологий построения баз данных. [2]

Чтобы создать информационную систему во внимание нужно брать такие аспекты как: изучение данных, непосредственно циркулирующие в организации; создание базы данных с целью обслуживания запросов, появляющихся в организации.

Маршруты движения информации, ее объемы, места возникновения и использования отражены в схемах потоков информации.

Анализируя данные схемы, можно сформировать меры, способствующие модернизации всей управленческой системы в целом.

Для того, чтобы осуществить детальный анализ, необходимо построение схем информационных потоков, которые обеспечивают: устранение повторяющейся и не применённой информации, а также классификацию и целесообразное представление информации.

Несомненно, то, что именно на основах проектирования строится методология построения баз данных.

Существуют два последовательных этапа, в которых представлены основные идеи методологии. Первый этап заключается в анализе функциональных отделов организации. Цель данного этапа состоит в том, чтобы познать особенности и структуру деятельности, установить информационные объекты и надлежащий комплекс параметров и характеристик, отображающий свойства и назначение.

Следующий этап основывается на создании информационно-логической модели данных для сферы деятельности, проверенной на

начальном этапе. Значимость этого этапа состоит в том, что в модели должны быть зафиксированы и упорядочены все связи между всеми частями и их параметрами.

Основой для создания базы данных и есть данная модель. Требуется наличие многих пунктов, чтобы создать информационное обеспечение. Например, представление всей системы, ее функций, цели и задач; продумывание функционирования информации от зарождения до ее реализации; модернизация комплекса делопроизводства; существование и применение системы классификации и кадрирования; умение оперировать методикой создания моделей, отражающих взаимосвязь информации в целом; существование технического обеспечения, а именно формирование массивов информации для различного вида носителей.

Под термином «техническое обеспечение» понимается совокупность технических средств, при помощи которых осуществляется бесперебойная работа информационных систем, а также соответствующая документация. [2]

Комплекс технических средств, как правило, включает в себя компьютеры; устройства сбора, накопления, обработки, передачи и вывода информации; устройства передачи данных; линии связи; различную оргтехнику и устройства съёма информации, а также эксплуатационную документацию.

Документация позволяет оформлять начальный набор технических средств с их реализацией, процесс обработки данных и технологическое оснащение.

Множество математических моделей, алгоритмов, методов и программ, служащих для выполнения задач информационной системы и бесперебойной работы технических средств, содержит программное и математическое обеспечение. [2]

В состав математического обеспечения входят методы математического программирования, статистики и теории массового

обслуживания; средства моделирования процессов управления; типовые задачи управления.

Специальные общесистемные программные продукты и техническая документация являются неотъемлемой частью программного обеспечения.

Общесистемное программное обеспечение включает в себя комплексы, ориентированные на пользователей, а также цель которых — решение задач обработки данных. Данное программное обеспечение дает возможность повышать функциональность компьютера и управления процессов обработки данных.

Специальное программное обеспечение — это ничто иное как разработанные при создании информационных систем программы, реализующие модели варьирующейся степени тождественности, которые отражают работу реального объекта. [2]

Техническая документация позволяет разрабатывать программные средства и состоит из описания задач и задания на алгоритмизацию, основных примеров и экономико-математической модели.

Организационное обеспечение — это, своего рода, комплекс методов и средств, цель которых — корректировка сотрудничества между работниками, в том числе с помощью технических средств. [2]

Все эти процессы совершаются в ходе разработки и реализации информационных систем.

Функциями организационного обеспечения являются: исследование системы управления; выявление задач, подлежащих автоматизации; подготовку задач к решению на компьютере; разработка решений по управлению структурой организации, для увеличения эффективности.

Правовое обеспечение состоит из комплекса правовых норм, которые определяют юридический статус и реализацию информационных систем, регламентирующих установленный порядок получения, обработки и использования данных с целью упрочнения законности.

Правовое обеспечение включает в себя законы и приказы, указы и инструкции и другое. А также состоит из локальной и общей части. Целью общей части является контроль функционирования информационной системы, а локальной — контроль функционирования определённой системы. Правовое обеспечение создания информационной системы включает в себя нормативные акты, связанные с договорными связями разработчика или заказчика и правовым регулированием отклонений от заключенного договора.

Таким образом, правовое обеспечение этапов функционирования состоит из статуса информационной системы, прав, обязанностей персонала, правового положения видов процесса управления и порядка создания и использования данных.

1.2 Средства разработки

В наше время информационные технологии активно развиваются и всё больше внедряются в различные сферы деятельности человека. Так же развиваются различные средства разработки приложений. Для разработки desktop-приложений стандартом становятся такие языки, как Java, С, С++ и С#.

Язык С относительно "низкого уровня". Это можно объяснить тем, что это язык работает с такими объектами, что и большинство электронно-вычислительных машин, а именно, с числами, адресами и символами. Они могут соединяться и пересылаться при помощи арифметических и логических операций, производимых реальными электронно-вычислительными машинами. С не имеет операций с комбинированными объектами, например, строки, множества, списки или массивы, которые рассматриваются как целое. В общем, это язык больше подходит для разработки драйверов и других системных вещей. [3]

C++ является компилируемым строго типизированным языком общего назначения. Данный язык поддерживает различные парадигмы программирования: функциональную, процедурную, обобщённую; наибольшее внимание уделено объектно-ориентированному программированию. Естественная область применения для него это — системное программирование. Если нужно создать что-то громоздкое, коммерческое или приложение, нуждающееся в высокой производительности, то тогда C++ подойдёт. [4]

Java представляет из себя объектно-ориентированный язык программирования высокого уровня. Большой плюс приложений Java — это транслирование в специальный байт-код. С помощью этого приложения могут работать на разных виртуальных Java-машинах независимо от архитектуры, которую использует компьютер. Этот язык, в основном, используется для написания апплетов и клиентских приложений серверного программного обеспечения. [5]

Язык программирования C# — это объектно-ориентированный язык. Он прост, типобезопасен и принадлежит семейству языков с C-подобным синтаксисом. Такой синтаксис очень близок к C++ и Java, так как создавался на их основе. C# иногда называют гибридом этих языков, который унаследовал всё самое лучшее. Этот язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов явного и неявного типа, делегаты, атрибуты, события, свойства, обобщённые методы, итераторы, анонимные функции с поддержкой замыканий, исключения и много другое. Благодаря большому количеству нововведений C# даёт возможность быстрой разработки приложений. Он наиболее стабильный и гибкий язык и чаще всего его используют для написания бизнес-приложений. [6]

В качестве среды для разработки проекта был выбран Visual Studio от компании Microsoft (рисунок 2). Этот продукт позволяет разработчику создавать консольные приложения и приложения с графическим интерфейсом, в том числе с поддержкой среды Windows Forms. В стороне не

остаются различные веб разработки, такие как, сайты, веб-приложения и веб-службы. Всё это можно делать как в родном, так и в управляемом кодах, поддерживаемых Windows, Windows Mobile, .NET Framework, Xbox, Windows Phone .NET Compact Framework, Silverlight и другими платформами. [7]

Также Visual Studio поддерживает различные дополнения. Вот самые известные из них:

- ReSharper — он служит для поиска ошибок в коде во время написания кода программы, до компиляции);
- Visual Assist — расширяет работу технологии IntelliSense и подсветку синтаксиса кода, содержит средства проверки орфографии комментариев и рефакторинга, может обнаружить небольшие синтаксические ошибки при наборе кода;
- AnkhSVN — позволяет следить и управлять изменениями в файлах проекта. [8]

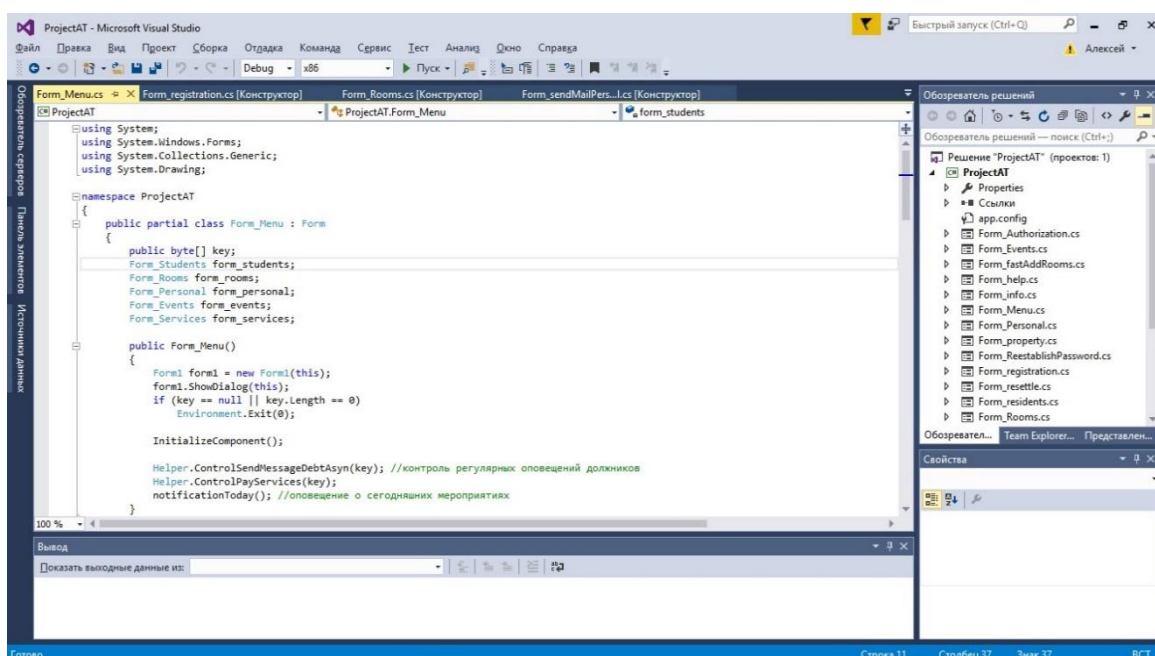


Рисунок 2 - Интерфейс Visual Studio

Еще одним программным средством разработки стал фреймворк — .NET Framework.

Фреймворк (в информационных системах) — структура программной системы; программное обеспечение, упрощающее процесс разработки и объединения разных компонентов программного проекта. Фреймворк содержит большое количество разных по назначению библиотек, в отличие от обычных библиотек, объединяющих подпрограммы, которые близки по функциональности. [9]

.NET Framework — это компьютерная платформа, упрощающая разработку приложений в очень распределенной среде. Она содержит некоторые основные компоненты: общая языковая среда выполнения, узлы среды выполнения и библиотека классов. Основа .NET Framework — это общая языковая среда выполнения. Эта среда управляет выполнением программ и обеспечивает некоторыми базовыми службами, такими как управление памятью, управление потоками и удаленное выполнение. У .NET Framework очень обширная библиотека классов. Она включает объектно-ориентированный сбор классов многоразового использования, применяемые при создании приложений. Также, .NET Framework содержит некоторое количество узлов среды выполнения, включая Windows Forms, ADO.NET и ASP.NET, работающие напрямую со средой выполнения для предоставления масштабируемой среды. [10]

Windows Forms — интерфейс программирования приложений, отвечающий за графический интерфейс пользователя. Оно представляет собой событийно-ориентированное приложение, поддерживаемое Microsoft.NET Framework. В отличие от пакетных программ, большая часть времени тратится на ожидание от пользователя каких-либо действий, как, например, ввод текста в текстовое поле или клика мышкой по кнопке. [11]

1.3 Базы данных

Сейчас трудно создать информационную систему, не имеющую базы данных. Совершенствование систем автоматизированной обработки информации всегда было тесно связано с концепциями развития баз данных.

База данных является совокупностью связанных данных, которые организованы по некоторым правилам, предусматривающим общие принципы хранения, управления и описания, независимо от прикладных программ. [12]

Реализация структуры базы данных происходит из соображений адекватности описываемого объекта (концептуальный и логический уровень), удобства использования, внедрения, учёта и использования данных (физический уровень).

При разработке проекта использовалась реляционная база данных.

Представленная база данных включает в себе множество связанных таблиц. Каждая из них содержит какую-то информацию об объектах данного вида. Любая строка содержит информацию о единственном объекте, к примеру, о сотруднике, а столбцы содержат характеристики этих объектов, так называемые атрибуты, такие как: ФИО, номер телефона, дата рождения.

Строки в таблице называют записями. Каждая запись имеет одинаковую структуру. Она состоит из полей — элементов данных, которые содержат атрибуты объекта (рисунок 3). В каждом из полей содержится одна характеристика данного объекта. [13]

Показать должников
Поиск: Информация о проживающих

Фамилия	Имя	Отчество	Институт	Курс	Дата рождения
Шорина	Алена	Михайловна	ИКИТ		11.07.1997 0:
Черникова	Алиса	Викторовна	ИИФиРЭ		02.03.1996 0:
Богдашев	Андрей	Васильевич	ИКИТ		30.05.1995 0:
Давыдов	Андрей	Павлович	ИУБПЭ		01.05.1997 0:
Байкалов	Антон	Викторович	ИКИТ		02.03.1996 0:
Снытко	Антон	Андреевич	ИИФиРЭ		30.05.1995 0:
Дубинина	Анна	Сергеевна	ИУБПЭ		11.07.1994 0:
Суворина	Анна	Михайловна	ИНИГ		17.12.1997 0:

Фамилия	Имя	Отчество	Институт	Курс	Дата рождения
Шорина	Алена	Михайловна	ИКИТ		11.07.1997 0:
Черникова	Алиса	Викторовна	ИИФиРЭ		02.03.1996 0:
Богдашев	Андрей	Васильевич	ИКИТ	3	30.05.1995 0:
Давыдов	Андрей	Павлович	ИУБПЭ	1	01.05.1997 0:
Байкалов	Антон	Викторович	ИКИТ	2	02.03.1996 0:
Снытко	Антон	Андреевич	ИИФиРЭ	3	30.05.1995 0:
Дубинина	Анна	Сергеевна	ИУБПЭ	4	11.07.1994 0:

Запись или строка
Поле или столбец

Рисунок 3 – Название объектов в таблице

В реляционных базах данных таблицы имеют ряд свойств. Основными, но не всеми, являются следующие:

- Таблица не содержит двух идентичных строк;
- Столбцы распределяются в определённом порядке. Этот порядок задаётся при создании таблицы. В ней обязательно должен быть хотя бы один столбец. Строк в ней может и не быть;
- Столбец всегда имеет уникальное имя в пределах таблицы. Все значения одного столбца имеют один тип, к примеру, число, текст или дата;
- На пересечении столбца и строки всегда находится атомарное значение — такое значение, не состоящее из группы значений. Таблицы, которые подходят под это условие, называются нормализованными. [14]

Наиболее известными программными продуктами для формирования и управления реляционными базами данных, которые будут использованы приложением, являются Microsoft SQL Server, Microsoft Access и Oracle Database и др. Также существует способ разработки баз данных с помощью создания зашифрованных файлов, хранящихся на компьютере, где непосредственно и будет содержаться определенная информация.

Преимущество этого способа в том, что установка приложения затрачивает меньшее количество времени за счёт того, что не устанавливается никакого стороннего программного обеспечения. Также при создании самого приложения упрощается реализация связи с данными.

1.4 Способ доступа к базе данных

В большинстве современных программ для доступа к базе данных используется СУБД.

Система управления базами данных, то есть СУБД — это специальное программное обеспечение, при использовании которого пользователи имеют возможность разрабатывать и контролировать базу данных, а также существует возможность осуществления к ней контролируемого доступа. [15]

Но в случае, если структура базы данных проста, можно использовать метод сохранения данных в зашифрованных текстовых файлах и управлять ими посредством языка С#. При этом не придётся использовать стороннее программное обеспечение.

1.5 Вывод по главе 1

При ознакомлении со структурой схемой информационных систем были выявлены их главные аспекты конструирования и функционирования. В результате обзора средств разработки был определен набор инструментов, способствующих проектированию и созданию информационной системы. Средством разработки был выбран язык С#. Выбор основывается на том, что он является наиболее подходящим для программируемого приложения. Также этот язык позволит наиболее быстро создать нужную информационную систему. Программное обеспечение Visual Studio было выбрано в качестве среды разработки, так как его инструментарий и различные дополнения

сильно упрощают процесс разработки интерфейса и написания кода. NET Framework выбран в качестве платформы разработки из-за возможности восприятия языка C#, обширной библиотеки классов и среды выполнения Windows Forms. Так же были рассмотрены различные виды баз данных, способы их создания и варианты доступа к ним.

Глава 2 Проектирование ИС

2.1 Формирование требований

В ходе проектирования автоматизированной информационной системы управления общежитием, были сформированы следующие задачи:

- Упростить процесс управления общежитием;
- Упростить процесс взаимодействия управляющего общежитием с сотрудниками;
- Упростить процесс взаимодействие с проживающими в общежитие студентами;
- Упростить процесс извлечения и нахождения нужной информации.

2.1.1 Требования к функциональности

Требования к функциональности такие:

- Реализовать процесс просмотра информации о проживающих студентов в общежитие;
- Создать инструмент организации плана мероприятий;
- Реализовать механизм оповещения о запланированном мероприятии;
- Создать инструмент отслеживания оплаты за проживание;
- Реализовать «канал» оповещения студентов о задолженности за проживание, как в автоматическом режиме, так и в ручном;
- Создать инструмент управления комнатами общежитий (просмотр имущества комнаты, просмотр жильцов комнаты);
- Создать в приложении механизм переселения студента в другую комнату;

- Реализовать поиск по полному и частичному совпадению, а также поиск должников;
- Создать инструмент перевода информации из любой таблицы в файл и обратно;
- Реализовать процесс сохранения в базе данных всех ранее проживающих в общежитие студентов и ранее работающих сотрудников;
- Реализовать контролируемый доступ в приложение;
- Реализовать инструмент для восстановления пароля;
- Создать механизм просмотра подключённых студентом услуг;
- Создать механизм оповещения и вызова сотрудников;
- Реализовать инструмент для навигации по приложению.

2.1.2 Требования к интерфейсу

Интерфейс должен удовлетворять таким требованиям:

- Он должен быть интуитивно понятным любому пользователю;
- Он должен быть с минимальной загруженностью;
- Он должен быть привлекательным;
- Он должен быть эффективным, т.е. в данном разделе должны быть функции только для этого раздела, без всех функций проекта;

2.1.3 Технические требования

Технические требования ко всему проекту:

- Использовать язык программирования C#;
- Небольшая нагрузка на операционную систему;
- Конечный продукт не должен занимать много места на жёстком диске;
- Конечный продукт должен работать быстро и стабильно.

2.2 Проектирование структуры приложения и разработка пользовательского интерфейса

В процессе проектирования интерфейса перед программистом ставится ряд целей, которые связаны с увеличением восприятия и эффективности некоторого продукта.

Интерфейс является одним из главных составляющих успешного программного продукта. Именно это определяет насколько система будет понятна, а также комфортна для пользователя, то есть будет определять дальнейшую судьбу созданного проекта.

Пользовательский интерфейс — это система средств и правил, которые регламентируют и обеспечивают взаимодействие приложения с пользователем. Понятие пользовательского интерфейса — это не только трёхмерная, анимированная или выполненная в модном дизайне картинка на экране, это способ взаимодействия пользователя с информационной системой.

Интерфейс приложения включает:

- Средства с помощью которых отображается информация;
- Режимы команд, то есть язык «пользователь — интерфейс»;
- Устройства ввода данных;
- Способы осуществляющие взаимодействия пользователя с компьютером, а также обратную связь;
- Поддержку принятия решений в данной предметной области;
- Особый порядок в использование программы. [16]

Интерфейс приложения спроектирован таким образом, чтобы он не содержал никаких вкладок. Это позволяет видеть все возможности программы сразу.

В процессе разработки структуры и пользовательского интерфейса приложения было создано 8 главных форм: «регистрация», «авторизация»,

«меню», «студенты», «комнаты», «персонал», «услуги» и «мероприятия». Каждая из них имеет собственную структуру.

Для раскрытия сущности каждого из разделов, внутри них будут использоваться различные элементы управления конструктора Windows Form. Хотя пользовательский интерфейс можно создать, написав код вручную, с помощью конструкторов это можно сделать намного быстрее и удобнее.

Windows Forms — это интерфейс программирования приложений, который отвечает за интерфейс пользователя, который в свою очередь является частью Microsoft .NET Framework. Данный пользовательский интерфейс сильно упрощает доступ к элементам интерфейса Microsoft Windows. Это происходит за счет создания некой обёртки для существующего Win32 API в управляемом коде. [11]

В Windows Forms при создании пользовательского интерфейса существуют три этапа:

- Добавление различных элементов управления на поверхность разработки системы;
- Установка определённых свойств для элементов управления;
- Написание обработчиков для всех заданных событий. [17]

При создании и изменении интерфейса программы нужно добавлять и выравнивать различные элементы управления.

Элементом управления называется объект, который находится внутри объектов формы. Все типы элементов управления имеют собственный набор свойств, методов и событий, которые соответствуют определенному назначению. С элементами управления можно работать в конструкторе, а также их можно добавлять динамически при выполнении с помощью кода. [18]

В качестве примера структур, можно привести некоторые окна.

Первым примером будет окно авторизации. Она имеет очень простую структуру (рисунок 4). В окне имеется два поля для ввода логина и пароля и

три кнопки для совершения различных действий, таких как вход в программу, отмена входа в программу и восстановления пароля.

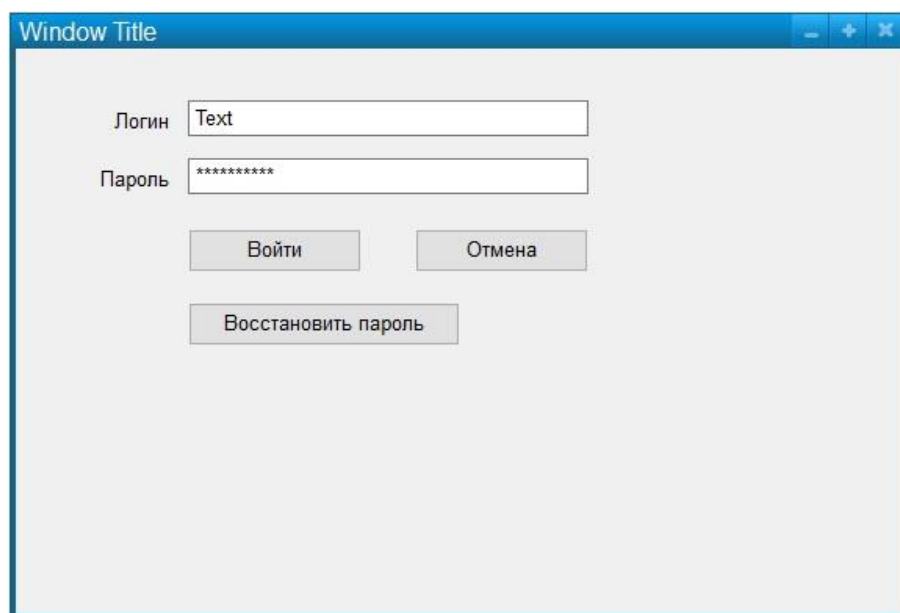


Рисунок 4 - Структура окна авторизации

При разработке этого окна используются такие элементы управления, как: «Button», «Label» и «TextBox».

Элемент управления Windows Forms «Button» даёт возможность пользователю приложения щелкнуть по нему для выполнения необходимого действия. На данном элементе могут отображаться как обычный текст, так и различные изображения. Когда происходит щелчок мышью элемент выглядит также, как и все стандартные кнопки в Windows, а это так будто его нажимают и потом отпускают. [19]

Элементы управления Windows Forms «Label» необходимы для выведения изображений или текста, не изменяемых обычным пользователем. Их используют для идентификации тех или иных объектов в форме. Например, для описания того что произойдет с элементом после того как на него кликнуть мышью, или для выведения сведений, отвечающих на процесс или событие количества времени выполнения в данном приложении. Так как элемент управления «Label» не способен получать фокус, его можно использовать при создании клавиш доступа к другим элементам. [20]

Текстовые поля форм Windows Forms используют для приема данных, которые вводит пользователь, а также для вывода текста. Элемент управления «TextBox» чаще всего используют для текста, который можно редактировать, но ещё его можно сделать доступным только для прочтения. В таких текстовых полях можно отображать несколько строк текста, располагать текст в соответствии с размером элемента и использовать основные элементы форматирования. В заданном элементе управления можно вводить и выводить текст только одного формата. Но ещё у этого элемента есть свойства, позволяющие сделать из него поле для ввода пароля. Так, для этого необходимо использовать PasswordChar и UseSystemPasswordChar. [21]

Второе окно — это «Главное меню». Так же, как и в первом случае структура окна очень проста (рисунок 5). Окно содержит такие разделы как: «Помощь», «Студенты», «Комнаты», «Персонал», «Услуги» и «Мероприятия».

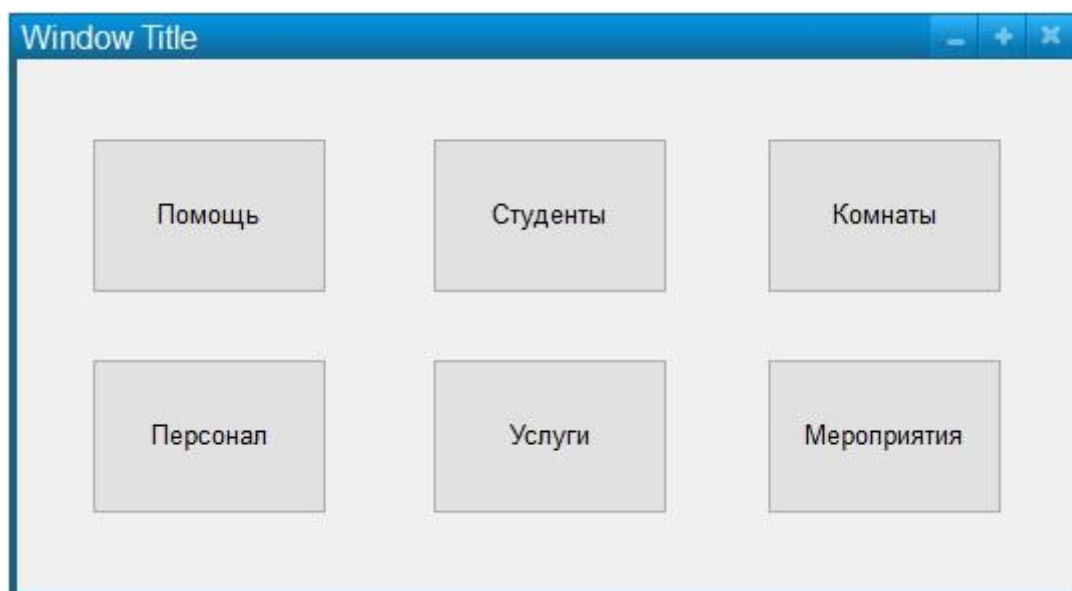


Рисунок 5 - Структура главного меню

Для создания этого окна использовался всего один элемент управления. Этим элементом является «Button».

Третьим примером будет окно «Студенты». У данного окна самая сложная структура во всём приложении (рисунок 6).

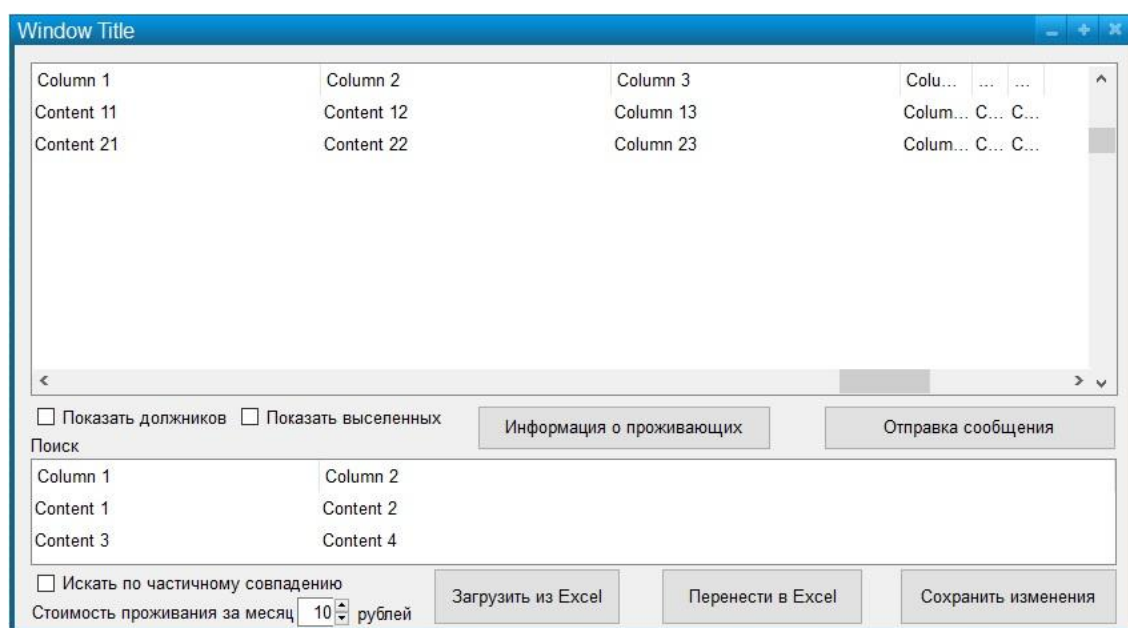


Рисунок 6 - Структура окна «Студенты»

Элементы управления, которые использовались для создания данного окна: «TableLayoutPanel», «HScrollBar», «VScrollBar», «Button», «CheckBox», «NumericUpDown» и «Label».

TableLayoutPanel — это элемент управления который может упорядочивать своё содержимое в сетке. Макет выполняется и во время разработки, и во время выполнения. Это позволяет изменять его динамически в процессе изменения среды приложения. Также из этого следует возможность изменения размера элементов управления в панели, так чтобы учитывались различные изменения родительского элемента управления или длины вставленного текста при локализации. [22]

Элементы управления HScrollBar и VScrollBar представляют из себя полосы прокрутки. Они используются для того чтобы обеспечить удобный просмотр длинных списков различных элементов или большого объема информации при помощи вертикальной и горизонтальной прокрутки окна

или элемента управления. В Windows очень распространены данные элементы интерфейса. [23]

CheckBox является элементом управления, который указывает, включено или нет некоторое условие. В большинстве случаев элемент используют для предоставления пользователю такого выбора, как: «Да/Нет» или «Истина/Ложь». Элементы типа «флажок» можно объединить в группы для того чтобы предоставить несколько вариантов ответа, из которых пользователь выберет один или несколько. Это напоминает элемент управления RadioButton, но разница в том, что можно выбрать любое количество объединенных в группу элементов CheckBox. [24]

Элемент Windows Forms NumericUpDown представляет из себя сочетание текстового поля и пары кнопок со стрелками для выбора значения. Он отображает и задает отдельное числовое значение в списке вариантов выбора. Есть возможность увеличения и уменьшения числа при нажатии кнопки со стрелками вверх и вниз соответственно или клавиши с такими же стрелками, а также это можно сделать, введя число в поле. Нажимая клавиши со стрелкой «вверх» значение можно увеличить до максимума и нажимая клавиши со стрелкой «вниз» число можно уменьшить до минимума. Управление громкостью в большинстве музыкальных проигрывателей производится с использованием этого элемента. Также многие числовые элементы управления со шкалой перемещения бегунка вверх и вниз используются в приложениях панели управления Windows. [25]

Ну и четвёртым примером будет структура окна мероприятий.

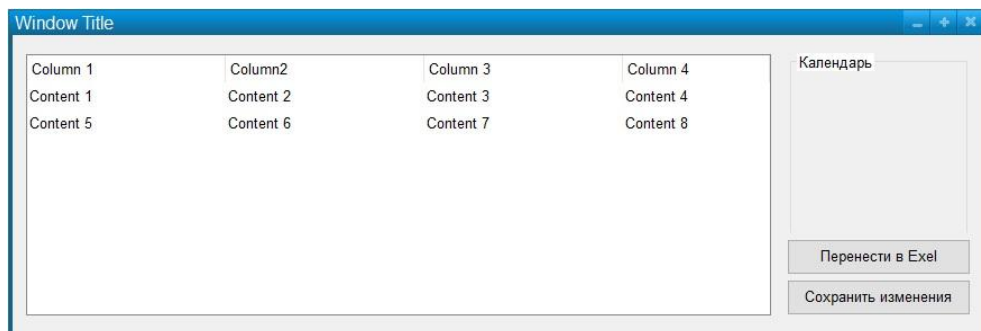


Рисунок 7 - Структура окна «Мероприятия»

Используемые элементы управления: «TableLayoutPanel», «Button» и «MonthCalendar».

Такой элемент управления, как MonthCalendar предоставляет всем пользователям очень понятный и привычный графический интерфейс, при помощи которого можно просмотреть и задать сведения о дате. Этот элемент выводит сетку, которая содержит пронумерованные дни месяца, в свою очередь разбитые на столбцы по дням недели. Переход к другому месяцу осуществляется путём нажатия кнопки со стрелками вправо и влево, находящимися по обе стороны от заголовка месяца. Но в отличие от почти аналогичного по функциональности элемента DateTimePicker, данный элемент управления может позволить выделение диапазона дат; однако элемент DateTimePicker может быть использован не только для задания дат, но и времени в том числе. [26]

Структура всех остальных окон строится на такой же основе и с такими же элементами управления, как и приведённые выше.

После того, как была спроектирована структура всех окон приложения и написан код, необходимо было разработать приятный графический интерфейс.

Внешний вид всех элементов приложения можно настроить при помощи панели «Свойства» (рисунок 8).

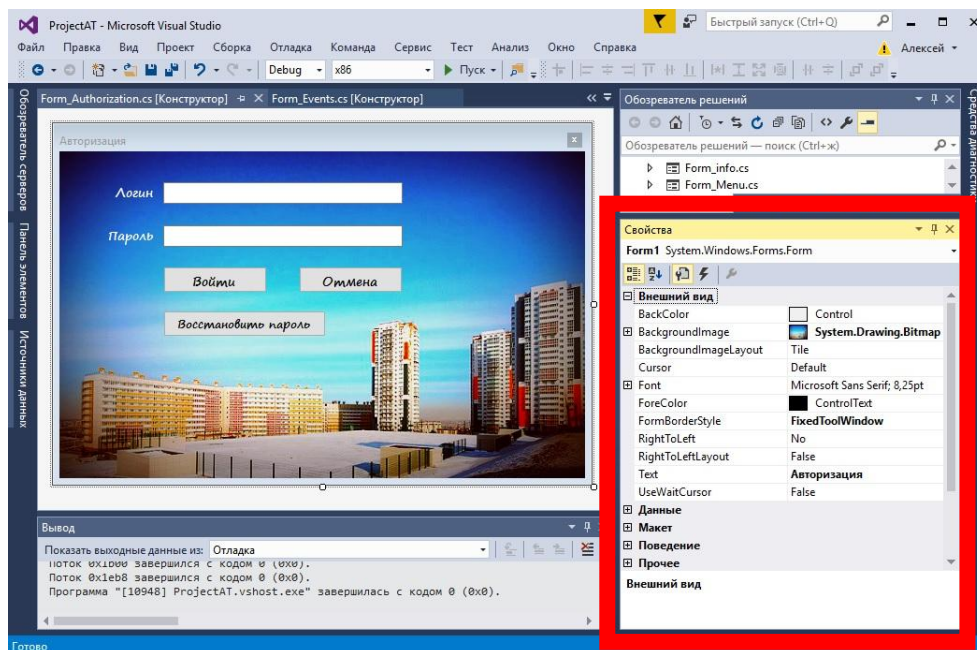


Рисунок 8 – Панель свойств формы

Эта панель позволяет просто и быстро изменить стиль любого элемента.

С её помощью был настроен стиль всех окон приложения. Рассмотреть получившийся интерфейс можно уже по знакомым нам примерам, применявшимся выше.

В окно авторизации было добавлено фоновое заранее обработанное изображение (рисунок 9). Это было сделано при помощи свойства `BackgroundImage`. Как уже стало понятно, свойство `BackgroundImage` используется для размещения графического изображения на элементе управления.

Также для всего текста в окне был применён полужирный шрифт `Segoe Print` размера 9 и 9,75. За это отвечает свойство `Font`, которое, как уже понятно отвечает за шрифт, используемый для отображения текста на любом элементе управления. Шрифт был выбран из-за того, что напоминает красивый рукописный текст и располагает к себе.

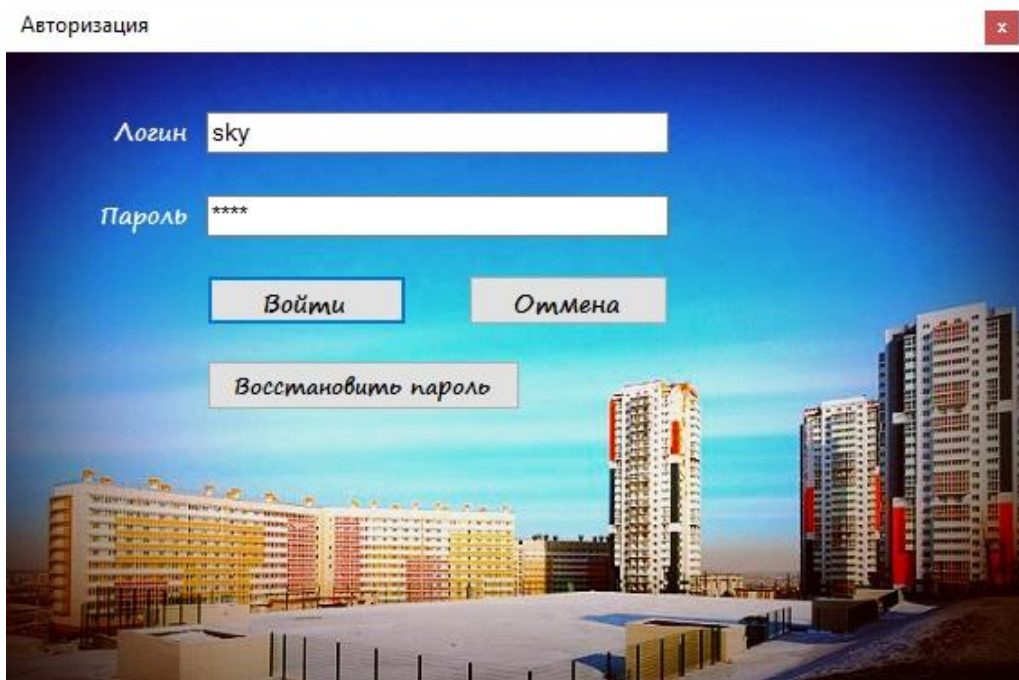


Рисунок 9 – Окно авторизации

В окне главного меню на кнопках также использовался шрифт Segoe Print размера 9,75. Ещё к данным элементам управления были добавлены иконки при помощи свойства Image. Кроме этого был добавлен фон (рисунок 10). Этот фон представляет из себя градиент. В отличии от предыдущего окна фон был прописан вручную (рисунок 11).

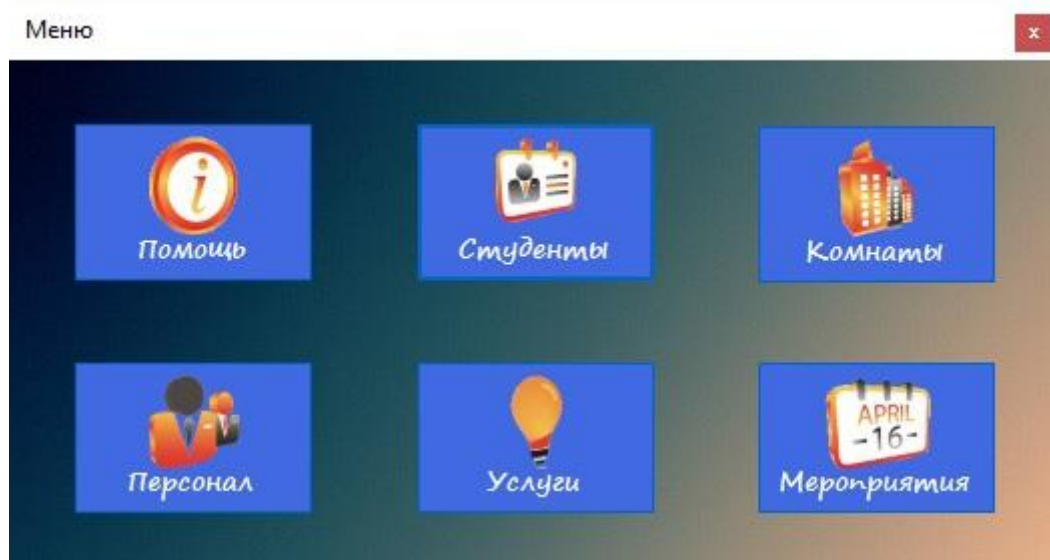


Рисунок 10 – Главное меню


```

//Градиент
private void Form_Menu_Paint(object sender, PaintEventArgs e)
{
    Helper.Gradient(e, 0, 0, this.ClientSize.Width,
        this.ClientSize.Height,
        Color.FromArgb(0, 0, 80),
        Color.FromArgb(230, 230, 230), 90F);
}

```

Рисунок 11 – Программный код реализации градиента

В следующих двух окнах также был добавлен фон. Добавлен, как и в первом случае в виде фоновой картинки (рисунок 12, рисунок 13).



Рисунок 12 - Структура окна «Мероприятия»

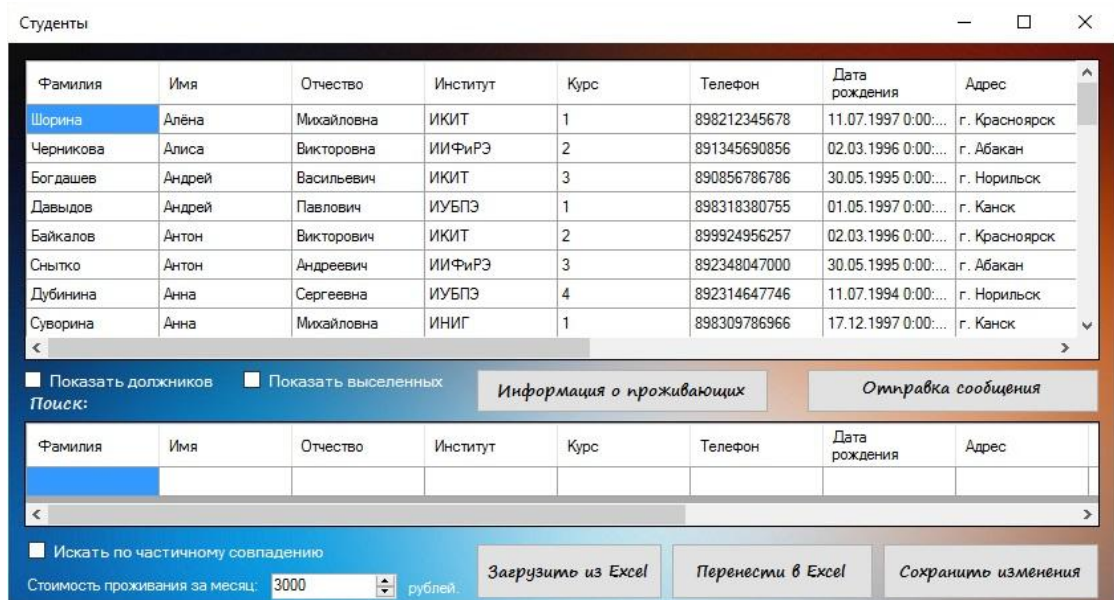


Рисунок 13 – Окно раздела «Студенты»

Был использован тот же самый шрифт, что и везде.

В ходе разработки необходимо обращать особое внимание на различные физические характеристики изображений, которые выводятся на экран, а именно на цветовой тон, цветовой баланс, контраст, яркость и насыщенность.

Цвет является мощным визуализатором, который способен как улучшить интерфейс и сделать его наиболее интересным, так и очень понизить заинтересованность пользователя в использовании приложения и просмотре информации. Самые важные части страницы, а именно блоки и элементы навигации, можно выделить при помощи цветовой гаммы. Также цвета могут показывать назначение программного продукта и его цели. Следствием этого, умение правильно работать с цветом является одним из главных атрибутов во всех сферах разработки, будь это веб-дизайн или разработка программного обеспечения. Такое умение позволяет правильно задать атмосферу и вызвать некий интерес к информационной системе, а также создать нужный эмоциональный настрой и прибавит желания работать в данном приложении. [27]

Сейчас существует много различных концепций выбора цветовой гаммы для разработки приложений, но самой простой и в это же время эффективной является цветовой круг Иттена (рисунок 14). Этот круг делится на теплую и холодную половины:

- Тёплые цвета: красный, оранжевый, жёлтый и промежуточные оттенки;
- Холодные цвета: синий, голубой, зелёный, и переходные между ними — сине-фиолетовый, сине-зелёный и т.д.. [28]



Рисунок 14 – Цветовой круг Иттена

Именно с помощью цветового круга Иттена можно подобрать наиболее успешное сочетание цветов. Для этого было сформировано шесть видов цветовых моделей, так называемые цветовые гармонии:

- Монохромные цвета — берётся один цвет, но с различными вариациями его насыщенности и прозрачности;
- Близкие цвета — это цвета, находящиеся рядом на круге;
- Комплементарные — цвета, расположенные друг напротив друга;
- Комплементарные и близкие — противоположные цвета, к примеру, А и Б. У Б выделяются соседи Б(л) слева и Б(п) справа;
- Комплементарные сдвоенные — комплементарные цвета для Б(л) и Б(п);
- Триадные цвета — они находятся через три сектора. В основном контрастные цвета сложно сочетаются, но в чистом виде, смотрятся достаточно эффектно. [28]

В нашем случае были выбраны комплементарные цвета.

Программа была полностью выполнена в синем и оранжевом цвете. Это было сделано что бы не переполнять программу всеми цветами радуги. Минимализм в таких программах приветствуется, так как целью программы является донести до пользователя информацию, а не пугать его пёстростью. Цвета были выбраны таким образом, чтобы они не отвлекали от основной работы, но в то же время привлекали к программе.

Цветовая гамма выдержана. На основных страницах цвета не так много. Он разбавлен серыми цветами вставок с данными и серыми кнопками. Это сочетание делает интерфейс очень «живым» и позволяет сосредоточиться на основной работе.

Текст на всех кнопках всех окон чёрного цвета, а текст вставленный с использованием элемента управления «Label» белый. Это позволяет хорошо считывать информацию и отлично поддерживает общий стиль программы.

2.3 Проектирование функциональной части приложения

Зависимо от области применения, информационные системы обычно различаются по своим функциональным особенностям, архитектуре построения или реализации. Однако, они обладают некоторыми общими свойствами:

- Любая из информационных систем предназначена для своевременного сбора, обработки и хранения информации. Следствием этого является то, что в основе всех таких систем лежат среды хранения и доступа к данным;
- Любая информационная система ориентирована на конечного пользователя, который не обладает достаточной квалификацией или нужным умением в области применяемой вычислительной техники. Исходя из этого, созданное приложение должно обладать простым, наиболее удобным и легко осваиваемым пользовательским интерфейсом. Интерфейс обязан

предоставить конечному пользователю всё необходимое для осуществления работы с функциями, но в это же время не должен давать ему возможность выполнять лишние действия. [29]

Исходя из вышеописанных свойств информационных систем, помимо программной составляющей, для проекта необходимо:

- Спроектированная база данных, которая будет использоваться для хранения некоторой информации;
- Разработанный графический интерфейс пользователя.

Для раскрытия функций системы была начата разработка диаграммы вариантов использования, также называемая Use-case диаграммой. Данная диаграмма рассматривается как одно из главных средств для начального этапа моделирования динамики системы. Её используют для подробного выяснения всех требований заказчика к системе и дальнейшей фиксации этих требований в некой форме, позволяющей проводить дальнейшую разработку.

Use-case диаграмма описывает отношения и зависимости между группами вариантов использования и действующими лицами, которые участвуют в процессе. [30]

Диаграмма вариантов использования не подходит для отображения проекта и описания внутреннего устройства информационной системы. Она предназначена для упрощения взаимодействия с будущими пользователями и клиентами системы. И особенно она пригодится для определения необходимых и нужных заказчику характеристик создаваемой системы. То есть, другими словами, Use-case диаграмма говорит, что система должна делать и не указывает на применяемые при этом методы.

Вариант использования создаёт описание группы действий в системе, приводящих к конкретному конечному результату. Делается это с точки зрения действующего лица. [30]

Варианты использования — это описания типичных взаимодействий пользователей системы с самой системой. Также они отображают внешний

интерфейс всей системы и указывают на форму того, что система обязана сделать. [30]

Смысловая архитектура диаграммы вариантов использования заключается в том, что проектируемая информационная система отображается в виде большого количества сущностей или актеров, взаимодействующих с системой при помощи различных вариантов использования.

Актер или действующее лицо — это любая сущность, которая взаимодействует с данной системой извне.

Система предоставляет актёру сервисы, описанием которых, также занимают варианты использования. Варианты использования системы управления прежде всего основываются на определенных видах действий. [30]

Действующее лицо — это внешний источник, то есть оно не является элементом системы. Взаимодействует с системой происходит через вариант использования. Действующими лицами не обязательно должны быть реальные люди. Ими могут быть как компьютерные системы, так и любые внешние события.

Ещё очень важным является то, что действующие лица представляют не физических людей или системы, а всего лишь их роли. Это значит, что при взаимодействии с системой какими-либо способами человек отображается несколькими действующими лицами.

В результате согласования с представителем организации, была создана диаграмма вариантов использования (рисунок 16).



Рисунок 16 – Use-case диаграмма использования

Существует множество методологий моделирования и представления бизнес-процессов. Самые популярные из них IDEF0, IDEF3 и DFD.

Моделирование бизнес-процессов (Business Process Modeling). Большее применение на данный момент имеет стандарт IDEF0, позволяющий описать бизнес-процесс. Предназначение модели в нотации IDEF0 состоит в высокоуровневом описании бизнеса организации в функциональном аспекте. [31]

Описание потоков работ (Work Flow Modeling). Основой Стандарта IDEF3 является описание действующих процессов и подобен алгоритмическим методам создания блок-схем. [31]

Описание потоков данных (Data Flow Modeling). Существует нотация DFD (Data Flow Diagramming), отображающая очередность работ, выполненные в ходе данного процесса, и объем информации, циркулирующий между текущими работами. [31]

В качестве методологии функционального проектирования был выбран стандарт IDEF0, так как эта методология сейчас имеет наибольшее распространение и в данном случае она более подходящая.

Стандарт IDEF0 отображает организацию как совокупность модулей. В нём существует правило — наиболее доминирующая функция располагается

вверху диаграммы в левом углу, а наименее доминирующая - в правом нижнем. Помимо этого, существует правило стороны:

- Стрелка входа находится всегда в кромке активности, расположенной слева;
- Стрелка управления — в кромку, расположенную вверху;
- Стрелка механизма — в кромку, расположенную снизу;
- Стрелка выхода — в кромку, расположенную справа. [32]

Существует 5 типов стрелок в методологии IDEF0:

- **Вход** — множество объектов, которые применяются и преобразуются работой с целью достижения какого-либо результата (выхода). Возможен и тот вариант, что в работе не существует ни одной стрелки, обозначающей вход. Стрелка входа рисуется как примыкающая к левой грани работы;

- **Управление** — объем информации, регулирующий действия работы. Также управляющие стрелки содержат информацию, указывающую действия для выполнения работе. Любой работе требуется как минимум одна стрелка управления, которая рисуется как примыкающая к верхней грани работы;

- **Выход** — те объекты, в которые трансформируются во входы. Любой работе необходимо как минимум одну такую стрелку, которая рисуется как выходящая из правой грани работы;

- **Механизм** — те ресурсы, за счет которых происходит выполнение работы. Данная стрелка представляется как примыкающая в нижнюю грань работы. Изображение стрелок механизма является не обязательным;

- **Вызов** — стрелка, имеющая особое предназначение - указывать на иную модель работы. Данная стрелка представляется как выходящая из нижней части работы и применяющаяся с целью указания на некоторую работу, которая выполняется вне моделируемой системы. [32]

Как уже стало понятно описание представляет собой «чёрный ящик», содержащий в себе входы, выходы, управление и механизм, который со временем конкретизируется до нужного уровня. Помимо этого, существуют словари описания активностей и стрелок, где даны описания того, какой смысл вложен в заданную активность или стрелку.

В дополнение демонстрируются сигналы управления, которые на DFD не были отображены. Такая модель применяется при создании бизнес-процессов и проектов, базирующихся на моделировании как административных, так и организационных процессов.

При создании информационной системы основной задачей является анализ работы предприятия, для которого создается данная система. Требуется создать модель, которая описывает процесс работы в организации, в соответствии с заданной тематикой. Наряду с этим, модель должна содержать весь спектр нужной информации о функциях бизнес-процессов и принципах работы организации.

Процесс моделирования системы базируется на формировании контекстной диаграммы абстрактного уровня описания системы, точно определяющей элементы моделирования, ее миссию и точки зрения на эту модель (рисунок 17).

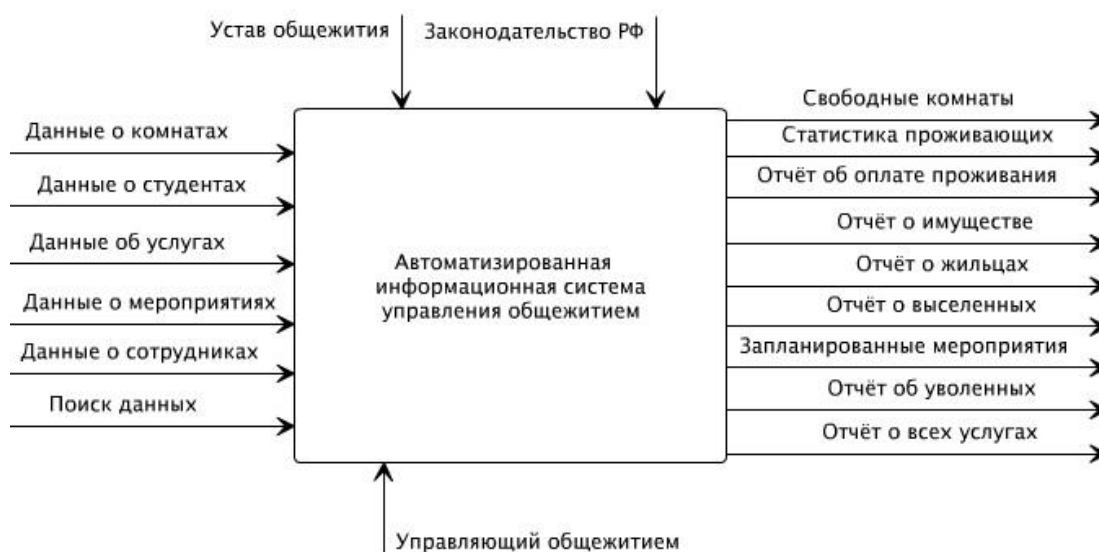


Рисунок 17 – Контекстная диаграмма

Она представляет обобщенное описание системы и взаимное сотрудничество системы и внешней среды.

Данная контекстная диаграмма демонстрирует главный процесс «Управления общежитием», обозначенный в блоке прямоугольного вида. Входной информацией для данного процесса являются: «Данные о комнатах», «Данные о студентах», «Данные об услугах», «Данные о мероприятиях», «Данные о сотрудниках» и «Поиск данных», вводимые в систему управляющим, полученный результат работы процесса - выходная информация: «Свободные комнаты», «Статистика проживающих», «Отчёт об оплате проживания», «Отчёт о имуществе», «Отчёт о жильцах», «Отчёт о выселенных», «Запланированные мероприятия», «Отчёт об уволенных», «Отчёт о всех услугах». Также были выделены направления «Устав общежития» и «Законодательство РФ», на базе которых осуществляются все выходные процессы, и механизм «Управляющий общежитием», занимающийся основной работой по управлению и контролю за этими процессами.

Позднее, как произошло описание контекстной диаграммы, осуществляется функциональная декомпозиция. Она предполагает деление на подсистемы. Дальше каждая подсистема описывается индивидуально. Затем, если таковое требуется, происходит деление ещё на более мелкие составляющие и так далее до того момента, пока не будет достигнут нужный уровень подробности. В конечном счете, всякий фрагмент системы отображается на специальной диаграмме декомпозиции (рисунок 18).

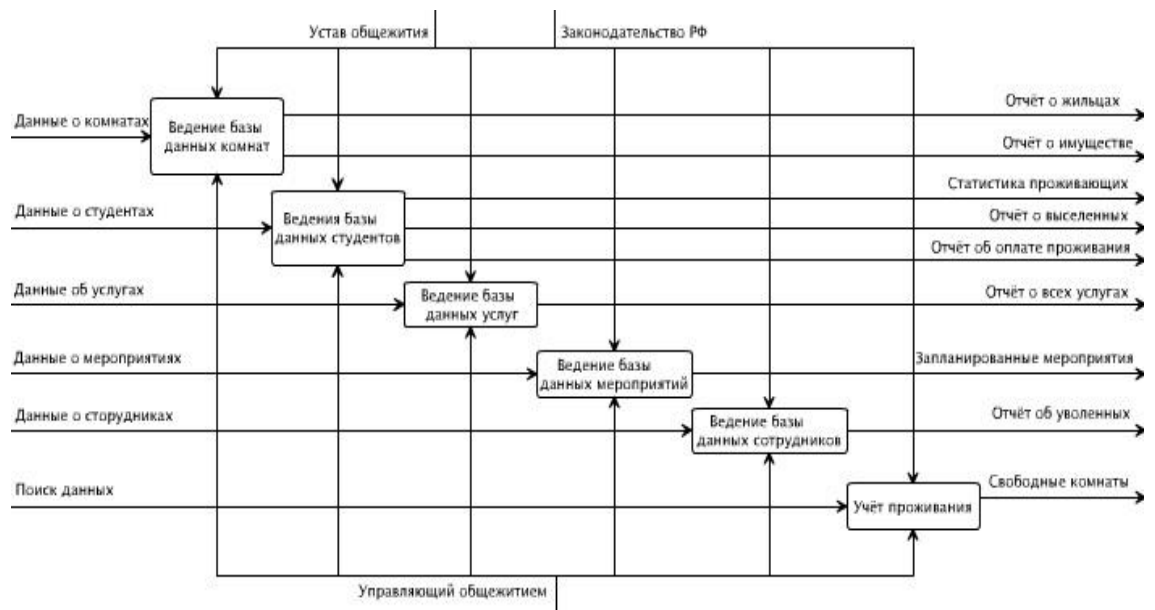


Рисунок 18 – Диаграмма декомпозиции

2.4 Проектирование структуры взаимодействия базы данных с приложением

Процесс взаимодействия компонентов программы, а именно способ связи базы данных с приложением, можно представить при помощи диаграммы развёртывания.

Данную диаграмму используют для визуализации элементов и компонентов программы. Это представление, в основном, обращено на распределение, постановку и установку частей, составляющих физическую систему в целом. [33] Но ещё её можно использовать для общей демонстрации взаимодействия частей приложения (рисунок 19).

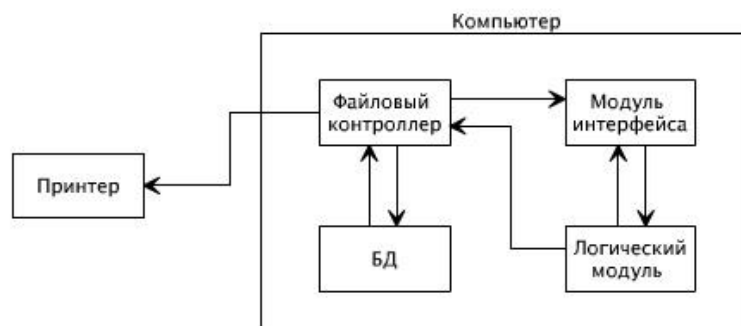


Рисунок 19– Диаграмма развёртывания

Для замены СУБД был вручную написан специальный файловый контроллер — FileControllе. Он занимается шифрованием, расшифровкой, преобразованием форматов, переводом текста в байты и обратно, в общем всем что связано с данными.

Чтение и запись данных производится при помощи пакетов StreamReader и StreamWriter соответственно.

StreamReader реализует объект TextReader, который считывает символы из потока байтов в определенной кодировке, а StreamWriter реализует TextWriter для записи символов в поток в определенной кодировке.

Также в файловом контроллере используется библиотека взаимодействия. Библиотека Interoperability позволяет использовать существующие вложения в неуправляемый код. При помощи неё C# взаимодействует с Microsoft Excel.

На вход контроллеру подаётся название таблицы и ключ шифрования. После этого нужный файл по таблице соответствия таблиц и файлов грузит, расшифровывает внутри себя и возвращает вот в таком виде: «Dictionary<int, Dictionary<string, string>»

Это ассоциативный массив в другом ассоциативном массиве. Сперва идёт id строки, но не всегда с нуля и не всегда последовательно. Затем идёт название поля как в базе данных и сами данные этого поля и строки.

В случае если бы данные не шифровались, то файл с данными выглядел бы таким образом: «name:Пётр&phone:214242984&institut:ikit».

Кроме обычных данных в каждой строке есть id. При загрузке id считывается и становится ключом ассоциативного массива.

2.5 Проектирование базы данных

Проектирование базы данных — процесс распознавания данных, их соотношения и размещение итогов в некоторую компьютерную программу. Данное проектирование существует вне зависимости от вида управления.

Этапы проектирования бывают:

- Концептуальное проектирование — процесс, в котором осуществляется сбор, исследование и редактирование требований, обращенных к данным. Существуют такие мероприятия: исследование предметной области, выявление фрагментов пользовательского представления (каждый характеризуется информационными объектами и взаимосвязями), моделирование и разделение всех представлений. После этого образовывается концептуальную модель, соразмерную структуре базы данных;
- Логическое проектирование — изменение требований к данным в структуры данных. В итоге имеем СУБД-упорядоченную структуру базы данных и спецификации прикладных программ;
- Физическое проектирование — определение основных особенностей хранения данных и методов доступа. [34]

Правила реляционной модели контролируют организацию информации в таблицах, а также их взаимосвязь.

Для ведения базы и организации поиска требуемой информации в базе должны храниться сведения о всех студентах, жильцах отдельных комнат, работающем персонале, комнатах, всех услугах, подключённых студентами услугах, имуществе комнат и мероприятиях.

Исходя из сказанного выше и будет создано восемь таблиц, в каждой из которых будет храниться определённая информация.

Первая таблица — это таблица, содержащая всех проживающих в общежитие студентов — «Студенты». В ней, в обязательном порядке, должна быть такая информация, как: ФИО, институт, курс, номер телефона, дата рождения, адрес прописки, электронная почта, номер комнаты, дата окончания договора, оплата за проживание, подключённые услуги и поле для различных пометок о студенте. Вся информация будет вводится вручную или загружаться в базу с другого источника.

Вторая таблица — это таблица комнат — «Комнаты». В ней будут содержаться номера самих комнат, тип, вместимость, количество свободных мест, имущество и проживающие в ней студенты. Комнаты будут добавляться вручную или в автоматическом режиме при занесении их номера в таблицу «Студенты». Количество свободных мест в комнате будет высчитываться исходя из вместимости и проживающих в ней людей. Имущество будет добавляться через другую таблицу, таблицу «Имущество». Проживающих можно будет просмотреть исходя из таблицы «Жильцы комнаты».

Таблица «Имущество», то есть третья таблица должна содержать наименование имущества, его описание, количество и состояние. Вся информация вводится вручную.

«Жильцы комнаты» — четвёртая таблица. Она должна содержать ФИО, дату рождения, адрес прописки, подключённые услуги и пометки. Вся информация, кроме услуг, берётся из таблицы «Студенты». Услуги каждого студента берутся из таблицы «Подключённые услуги».

Пятая таблица «Подключённые услуги». Она содержит наименование услуги, её стоимость, описание, дату окончания договора и сумму оплаты за её подключение. Дата окончания договора и сумма оплаты вводится вручную. Остальная информация берётся из таблицы всех услуг.

«Все услуги» — таблица шесть. Она содержит Наименование услуги, её стоимость и описание. Всё вводится вручную.

Таблица номер семь — это таблица «Персонал». Она должна содержать такую информацию: ФИО, дата рождения, адрес прописки, электронная почта, дата найма на работу или заключения договора, должность и график работы. Вся информация вводится вручную.

Последняя, восьмая таблица — это таблица «Мероприятия». Она должна содержать название мероприятия, его описание и дату проведения. Вся информация, так же, как и в предыдущей таблице, будет вводится вручную.

На равне с информацией, которая будет в каждой таблице, должны быть операции по взаимодействию с ней.

Восемь видов использованных операций:

- Поиск — find ();
- Добавить — add ();
- Открыть — open ();
- Удалить — delete ();
- Сохранить — save ();
- Изменить — amend ();
- Переселить — to resettle ();
- Упорядочить — regularize ().

Каждая операция будет поставлена в соответствие с заданной функциональностью таблицы.

В конечном счете результат можно представить в виде диаграммы базы данных (рисунок 20).

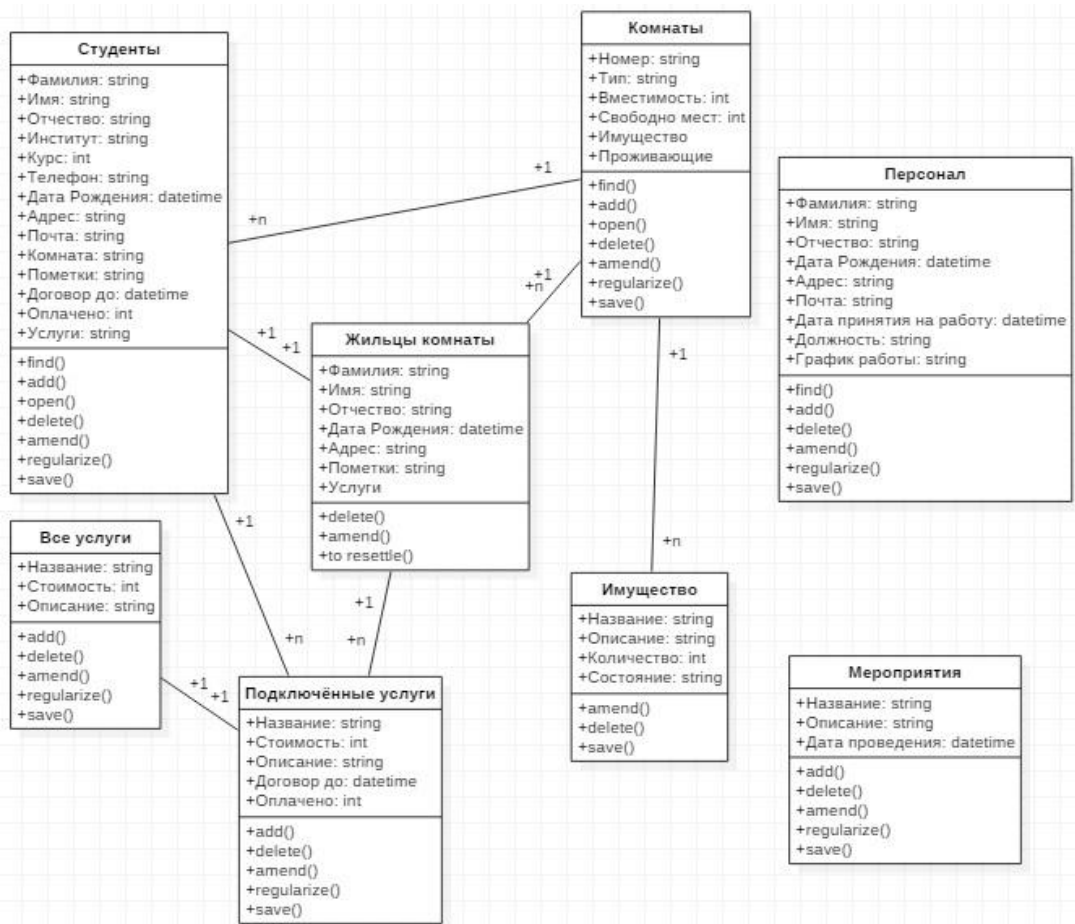


Рисунок 20 – Диаграмма базы данных

2.6 Вывод по главе 2

В данной главе был рассмотрен процесс проектирования информационной системы управления общежитием.

На первом этапе были определены основные требования к проекту.

Следующим этапом стала разработка структуры desktop-приложения, через которую комендант или тьютор смогут заходить и работать с базой данных.

Для определения функциональной части системы, а также для разработки упрощённой версии взаимодействия с пользователем, была построена диаграмма вариантов использования.

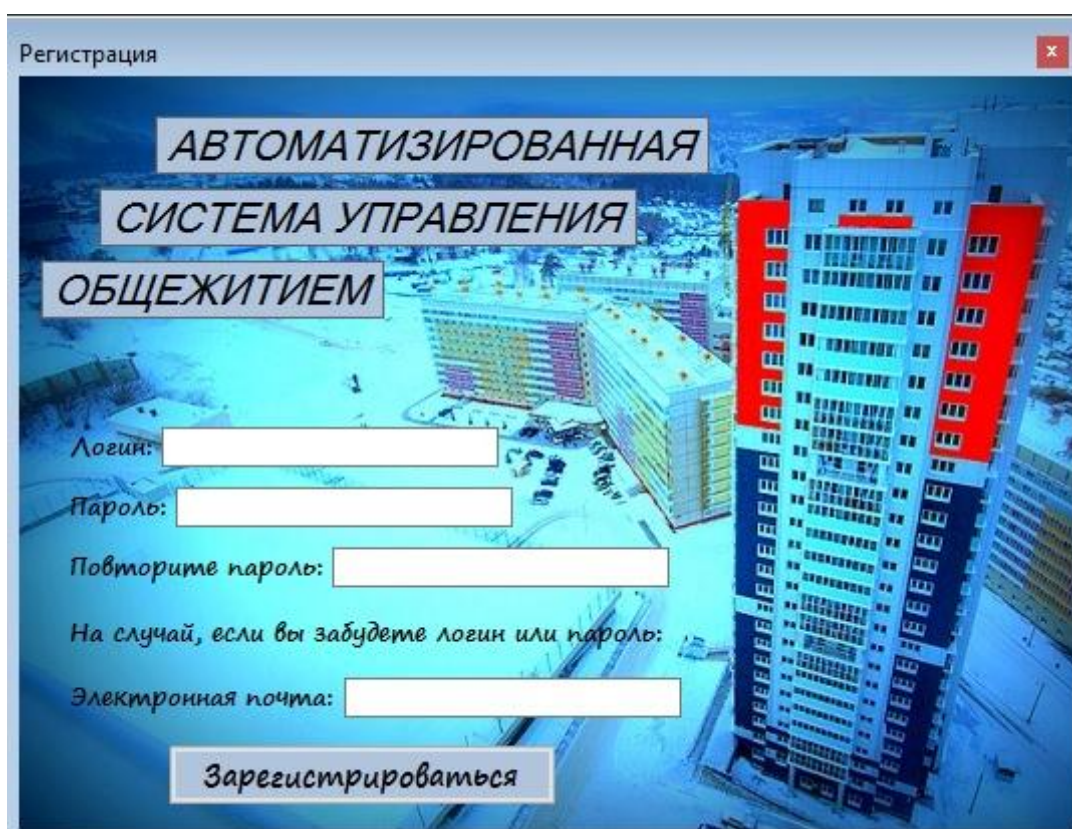
Далее для представления обобщенного описания системы и взаимодействие системы с внешней средой при использовании методологии IDEF0 была создана контекстная диаграмма. Следствием предыдущей диаграммы стала диаграмма функциональной декомпозиции.

На этапе проектирования базы данных для определения взаимодействия базы данных с приложением была создана диаграмма развёртывания. Также на этом этапе была создана диаграмма базы данных, определяющая ключевые атрибуты и операции, используемые в системе.

Глава 3 Описание и сравнение ИС

3.1 Сценарий работы пользователя с приложением

При первом входе в программу пользователь обязан зарегистрироваться (рисунок 21). На случай, если в дальнейшем при авторизации пользователь забудет логин или пароль, нужно ввести свою электронную почту.



The image shows a registration window titled "Регистрация" (Registration). The background features a 3D rendering of a modern apartment complex with several buildings in blue, red, and yellow. Overlaid on this background are several text boxes and input fields. At the top, three stacked boxes contain the text: "АВТОМАТИЗИРОВАННАЯ", "СИСТЕМА УПРАВЛЕНИЯ", and "ОБЩЕЖИТИЕМ". Below these are four input fields: "Логин:" (Login), "Пароль:" (Password), "Повторите пароль:" (Repeat password), and "Электронная почта:" (Email). A "Зарегистрироваться" (Register) button is located at the bottom center. The window has a standard Windows-style title bar with a close button in the top right corner.

Рисунок 21 – Окно регистрации

После этого происходит переход в окно авторизации (рисунок 22). Для перехода далее нужно ввести свои данные. В том случае, если вы забыли логин или пароль, их можно восстановить при помощи электронной почты, которую вводили при регистрации (рисунок 23).

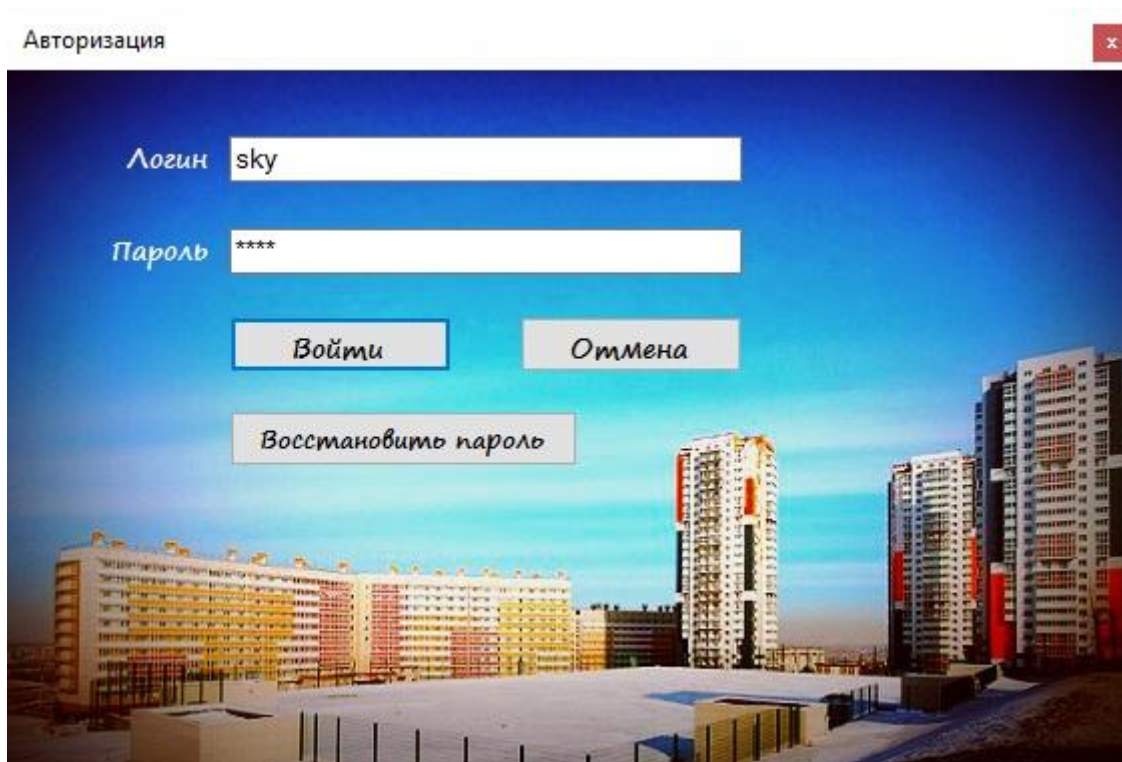


Рисунок 22 - Окно авторизации

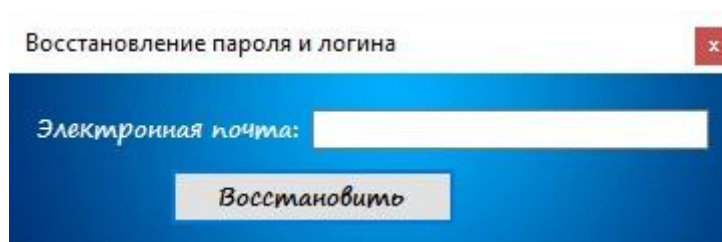


Рисунок 23 - Форма восстановления логина и пароля

В том случае, если пользователь попытается зайти в систему, используя при этом несуществующее имя или неправильный пароль, то он увидит оповещение о том, что введены неверные данные (рисунок 24).

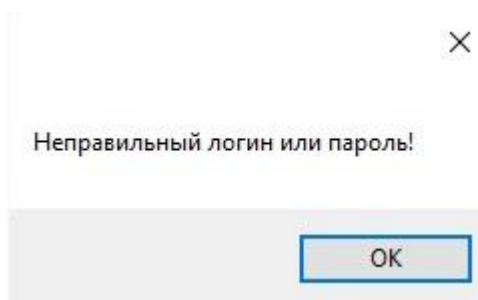


Рисунок 24 – Оповещение об ошибке

После ввода правильных данных при авторизации приложение пускает пользователя в главное меню (рисунок 25).

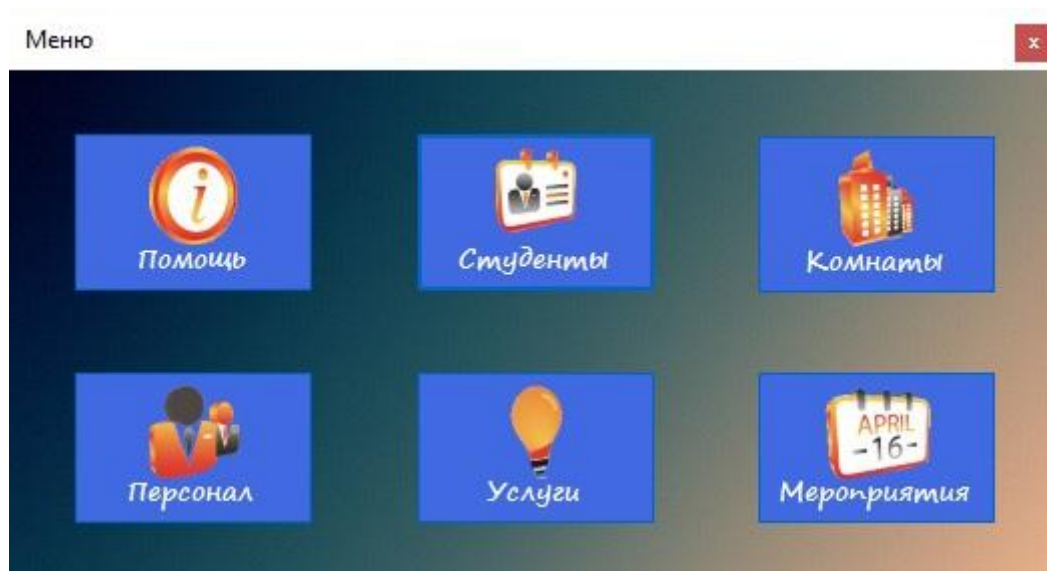


Рисунок 25 – Окно главного меню

Далее пользователю открываются все функции данного приложения. Первым разделом является помощь в управлении информационной системой (рисунок 26).

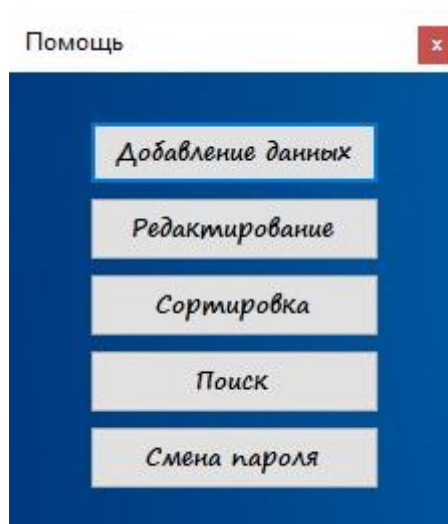


Рисунок 26 – Окно помощи

В разделе помощи существует пять кнопок. Первые четыре отвечают за различную помощь в работе с данными, а пятая отвечает за смену пароля.

После нажатия на одну из первых четырёх кнопок отображаются сообщения с выбранной вами помощью. Примером может служить оповещение о том, как действует функция поиска в данной системе (рисунок 27).

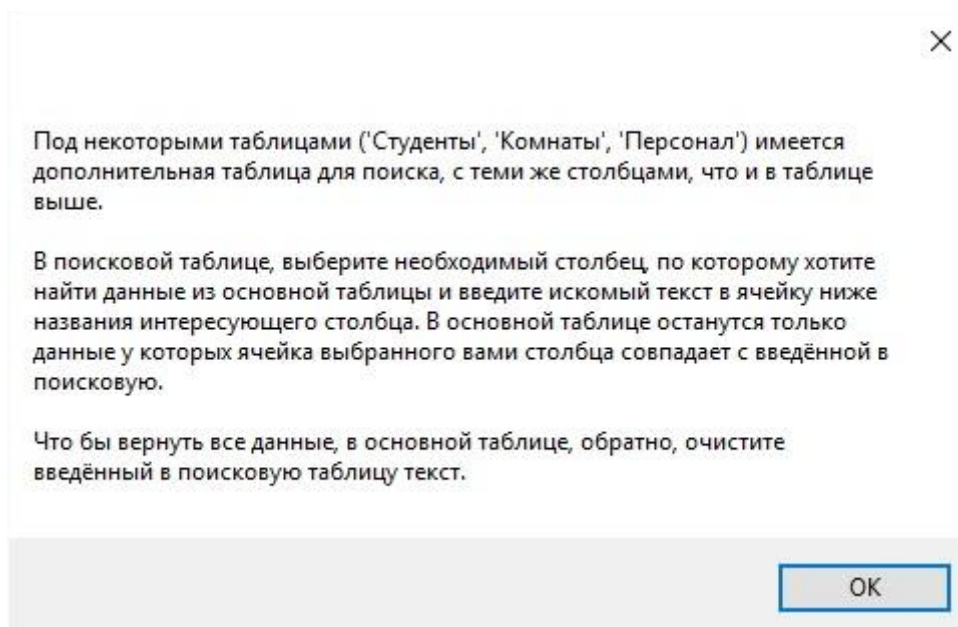


Рисунок 27 – Сообщение о помощи в поиске

При нажатии на пятую кнопку — «Смена пароля» отобразится форма с одноимённой функцией (рисунок 28). В данной форме нужно будет ввести старый и новый пароли.



Рисунок 28 – Окно смены пароля

Второй раздел в главном меню — это раздел управления студентами. После нажатия на кнопку «Студенты» откроется форма управления (рисунок 29).

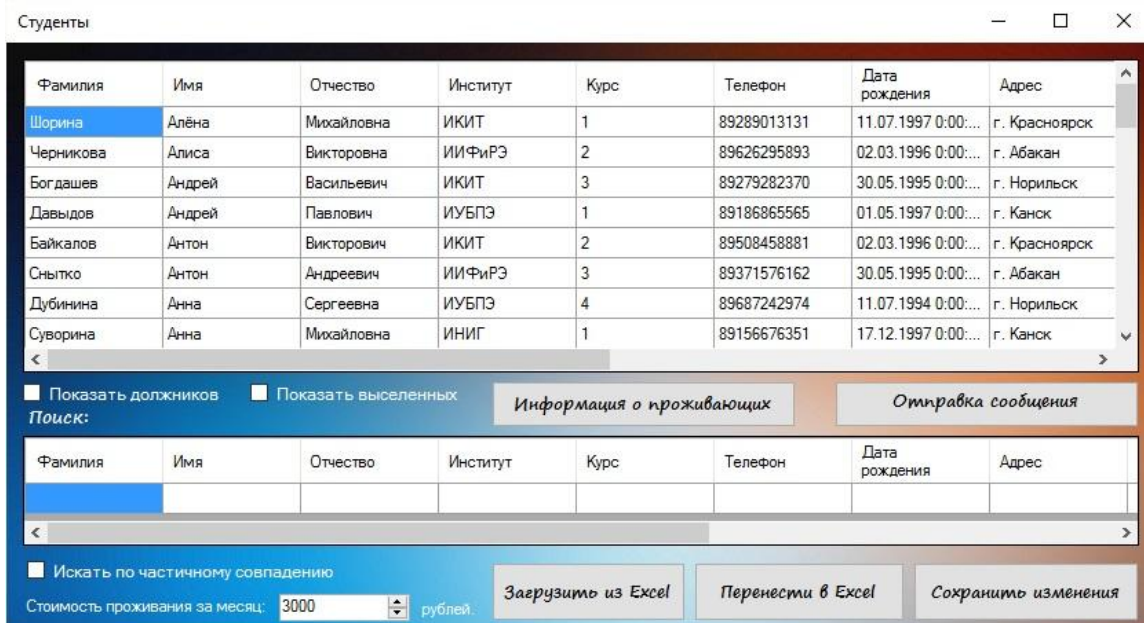


Рисунок 29 – Окно управления студентами

В данной форме имеется много разных функций.

Информацию можно упорядочить в алфавитном порядке или наоборот при помощи двойного клика на название столбца. Так же со столбцами, содержащими числа. В качестве примера можно привести столбец «Фамилия» (рисунок 30).

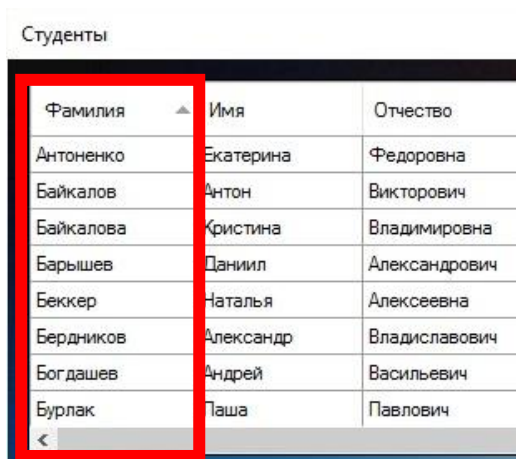
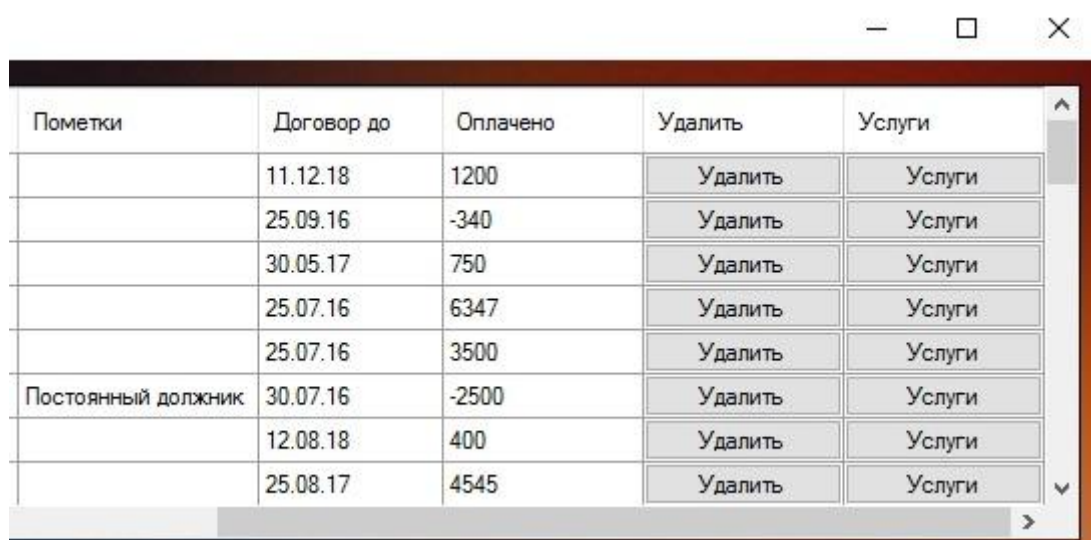


Рисунок 30 – Упорядоченный столбец по алфавиту

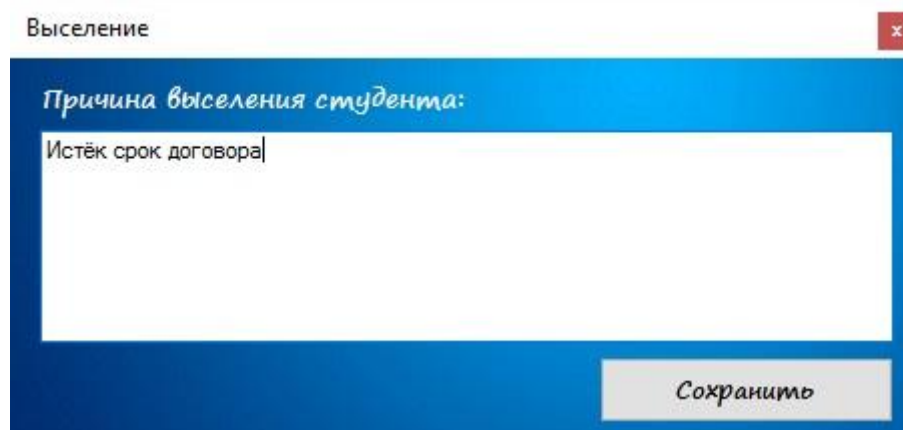
Последние два столбца дают возможность удалить/выселить студента и просмотреть услуги которые он подключил (рисунок 31).



Пометки	Договор до	Оплачено	Удалить	Услуги
	11.12.18	1200	Удалить	Услуги
	25.09.16	-340	Удалить	Услуги
	30.05.17	750	Удалить	Услуги
	25.07.16	6347	Удалить	Услуги
	25.07.16	3500	Удалить	Услуги
Постоянный должник	30.07.16	-2500	Удалить	Услуги
	12.08.18	400	Удалить	Услуги
	25.08.17	4545	Удалить	Услуги

Рисунок 31 – Столбцы удаления и услуг

При удалении/выселение студента открывается форма для заполнения причины выселения (рисунок 32). Это будет использоваться в случае повторного добавления студента в базу (рисунок 33).



Выселение

Причина выселения студента:

Истёк срок договора

Сохранить

Рисунок 32 – Окно причины выселения

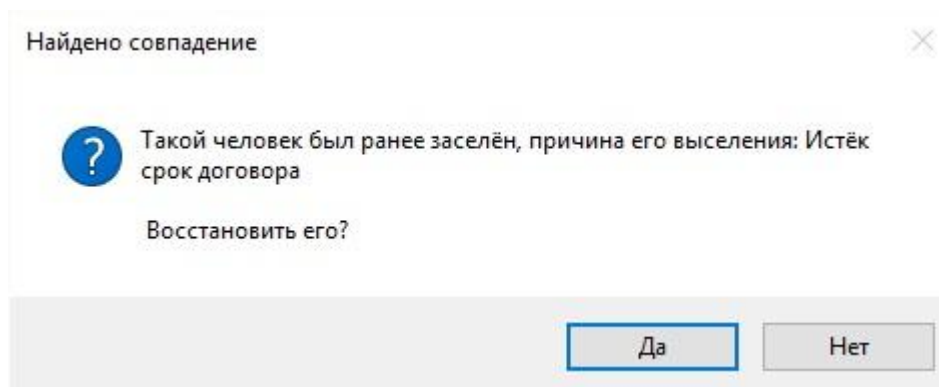


Рисунок 33 – Сообщение о том, что такой студент был жильцом общежития

При нажатии на кнопку «Услуги» открывается форма со всеми подключёнными студентом услугами (рисунок 34). Они выбираются из перечня услуг заданного в другой форме.



Рисунок 34 – Форма подключённых услуг

Выселенных людей можно просмотреть, поставив галочку на функцию «Показ выселенных» (рисунок 35). Таким же способом, только поставив галочку на функцию «Показать должников» можно просмотреть всех должников (рисунок 36).

Студенты

Фамилия	Имя	Отчество	Инс:
Шорина	Алёна	Михайловна	ИКИТ

Показать должников Показать высленных

Рисунок 35 – Пример отображения высленных

Студенты

Адрес	Почта	Комната	Пометки	Договор до	Оплачено
г. Абакан	GeForse93@yandex.ru	3		13.05.16	-340
г. Абакан		3	Постоянный должник	25.08.18	-2500

Показать должников Показать высленных

Поиск: [Информация о проживающих](#)

Рисунок 36 – Пример отображения должников за общежитие

Должникам можно отправить сообщение на электронную почту о сумме их долга нажав кнопку «Отправить сообщение». При нажатии открывается форма, в которой можно задать автоматическое отправление сообщений или отправить в данный момент (рисунок 37).



Рисунок 37 – Форма отправки сообщения о долге

В результате студенту на электронную почту приходит сообщение о задолженности (рисунок 38).



Ваш долг: -1234. Пожалуйста, оплатите его. Иначе, вас выелят :). P.S. За всеми подробностями к коменданту.

Рисунок 38 – Пример сообщение о долге приходящего студенту на почту

Если у должников не задана электронная почта, то отобразится сообщения об этом (рисунок 39).

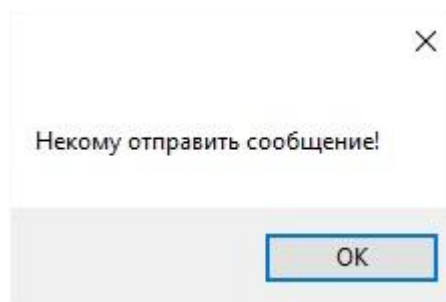


Рисунок 39 – Сообщение, отображающееся если не задана электронная почта

Также в форме по работе со студентами есть кнопка «Информация о проживающих». В результате её нажатия выводится форма, на которой отображена статистика проживающих в общежитие (рисунок 40).

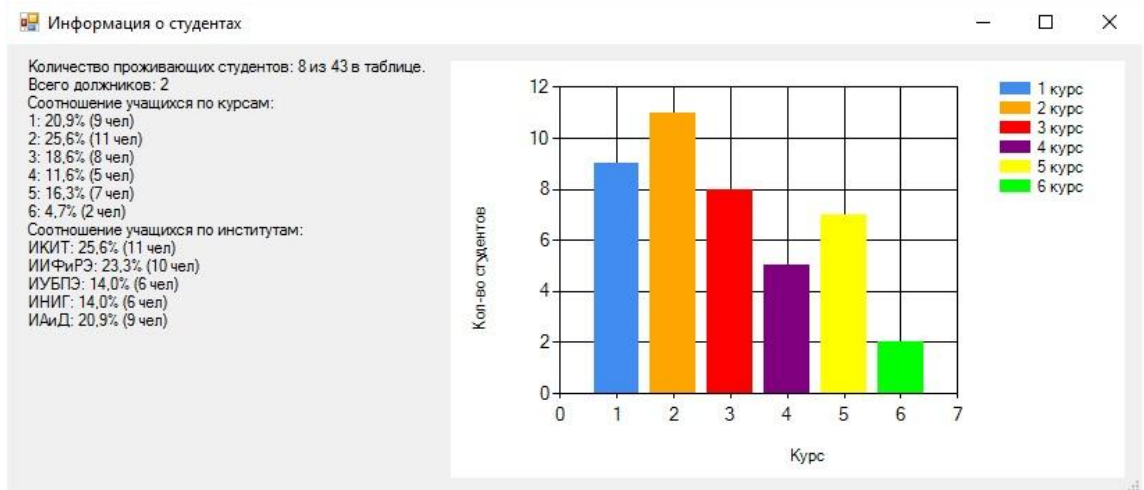


Рисунок 40 – Окно статистики проживающих

Ещё в разделе студенты есть возможность не только загружать данные в Excel документы, но и переносить в них информацию. Примером может служить форма, которая открывается при нажатии кнопки «Перенести в Excel» (рисунок 41).

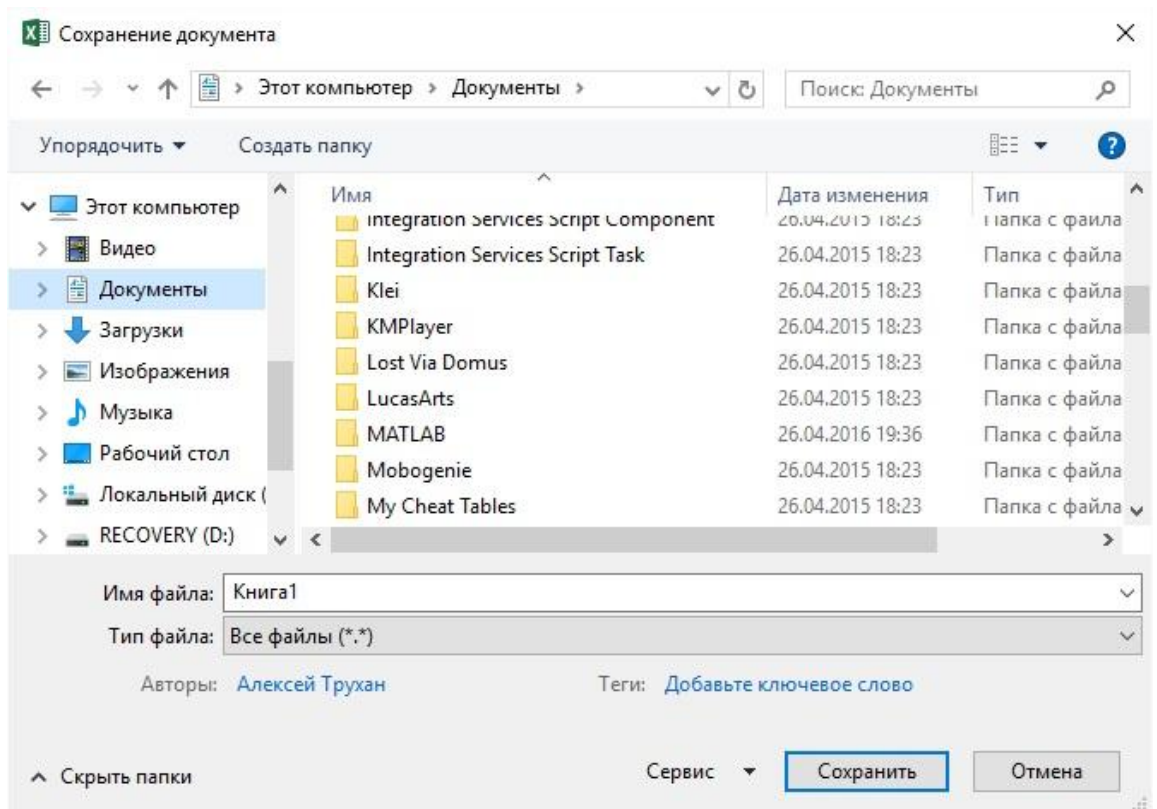


Рисунок 41 – Форма сохранения документа Excel

Если отменить сохранение данных в Excel файл, то отобразится сообщение о неудаче или об отмене (рисунок 42).

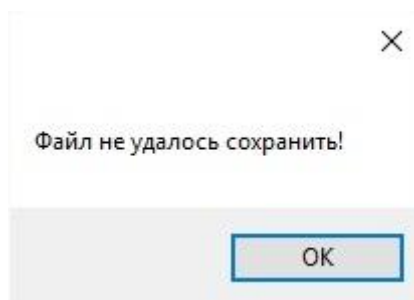


Рисунок 42 - Сообщение о неудачном сохранение

В данной форме, также подразумевается функция поиска, как обычная там и по частичному совпадению. Нужно написать, то что нужно найти в любом поле или нескольких полях сразу. Примером может служить поиск по частичному совпадению по столбцу «Фамилия» (рисунок 43).

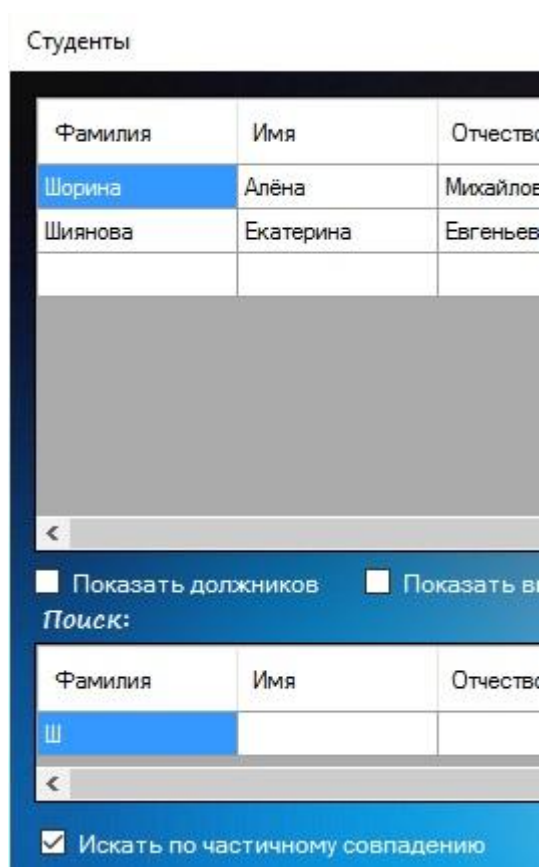


Рисунок 43 – Пример поиска по частичному совпадению

В форме существует столбец оплаты. В этом столбце автоматически производится ежемесячное списание средств за общежитие. Для задания суммы списания нужно найти строку «Стоимость проживания за месяц» и задать нужную сумму (рисунок 44).

Студенты

Адрес	Почта	Комната	Пометки	Договор до	Оплачено
г. Красноярск		1			1200
г. Абакан	GeForse93@yandex.ru	3		13.05.16	-340
г. Норильск		5			750
г. Канск		8			6347
г. Красноярск		1			3500
г. Абакан	руру	3	Постоянный должник	25.08.18	-2500
г. Норильск		5			400
г. Канск		1			4545

Показать должников Показать выселенных [Информация о проживающих](#)

Поиск:

Фамилия	Имя	Отчество	Институт	Курс	Телефон

Искать по частичному совпадению

Стоимость проживания за месяц: рублей [Загрузить из Excel](#) [Перенести в 1](#)

Рисунок 44 – Пример столбца «Оплачено» и функции списания средств

При любом изменении данных существует необходимость их сохранения. Делается это при помощи кнопки «Сохранить изменения». Если не сохранить, то при закрытии формы появится другая форма, с помощью которой можно будет сохранить изменения, не сохранить или отменить выход (рисунок 45).

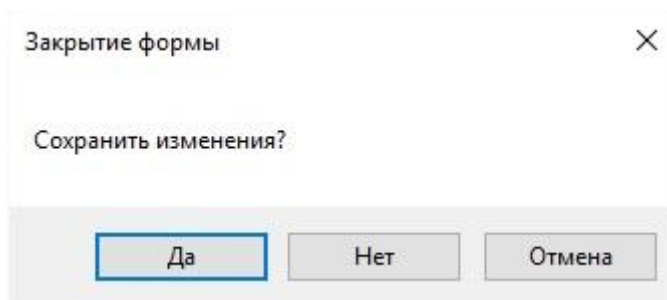


Рисунок 45 - Сообщение о сохранение изменений при закрытии формы

Третий раздел главного меню — это «Комнаты» (рисунок 46). Отвечает за управление всеми комнатами общежития.

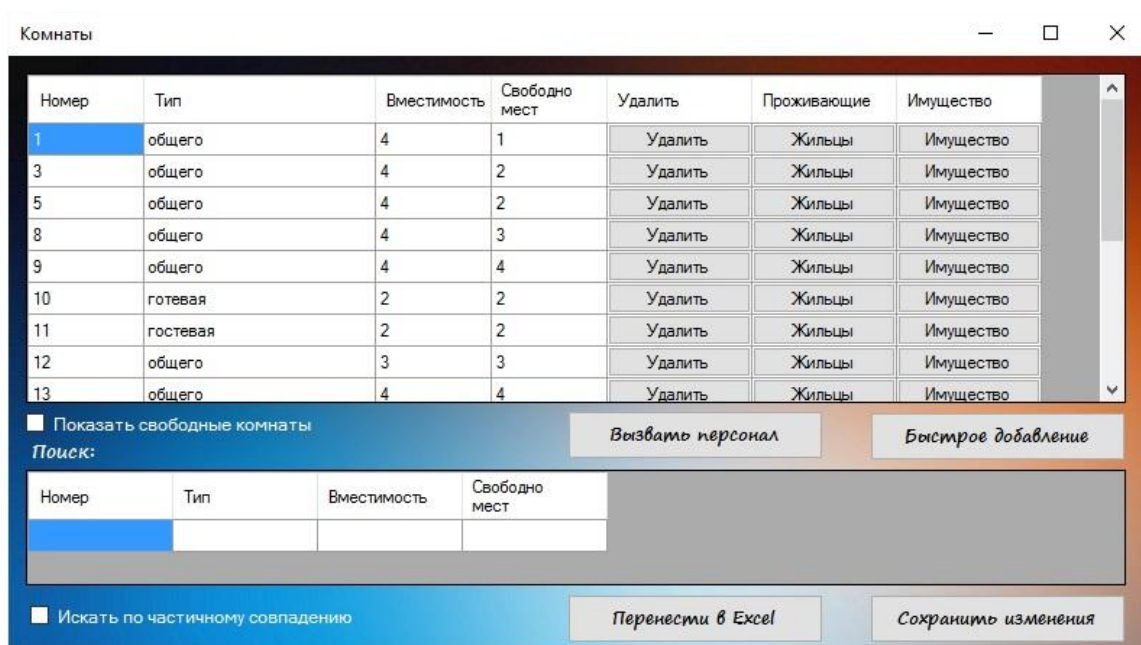


Рисунок 46 – Окно управления комнатами

Также, как и в форме «Студенты» тут имеются функции поиска, упорядочивания и переноса данных в Excel.

Кроме этого, в данную форму входят столбцы жильцов и имущества.

При нажатии на кнопку жильцов открывается окно со списком студентов проживающих в данной комнате (рисунок 47).

Жильцы комнаты 1

Комната 1

Фамилия	Имя	Отчество	Дата рождения	Адрес	Пометки	Услуги	Переселить	Удалить
Шорина	Алёна	Михайловна	11.07.1997 0:00:...	г. Красноярск		Услуги	Переселить	Удалить
Байкалов	Антон	Викторович	02.03.1996 0:00:...	г. Красноярск		Услуги	Переселить	Удалить
Суворина	Анна	Михайловна	17.12.1997 0:00:...	г. Канск		Услуги	Переселить	Удалить

Рисунок 47 – Окно отображения жильцов комнаты

В свою очередь в форме жильцов комнат есть столбцы «Удалить», «Услуги» и «Переселить».

Удаление и перечень услуг были описаны ранее.

При нажатии на кнопку переселения откроется форма, с помощью которой можно легко переселить студента в другую комнату (рисунок 48).

Переселить

ФИО: Шорина Алёна Михайловна

Комната: 1

Переселить в комнату

Переселить

Рисунок 48 – Окно переселения студента

Ещё в форме «Комнаты» есть возможность просмотреть свободные комнаты. Для этого нужно поставить галочку напротив функции «Показать свободные комнаты». В результате в списке останутся комнаты, в которых есть хотя бы одно свободное место (рисунок 49).

Комнаты

Номер	Тип	Вместимость	Свободно мест
1	общего	4	1
3	общего	4	1
5	общего	4	2
8	общего	4	3
9	общего	4	4
10	гостевая	2	2
11	гостевая	2	2
12	общего	3	3

Показать свободные комнаты

Рисунок 49 – Пример перечня свободных комнат

Также в данной форме есть возможность отправить персонал в какую-либо комнату для обслуживания. Происходит это при нажатии кнопки «Вызвать персонал» (рисунок 50). В открывшейся форме нужно будет выбрать сотрудника, из перечня существующих в базе данных, выбрать комнату и написать сообщение о проблеме в той или иной комнате.

Отправка сообщения сотруднику

Сотрудник:

Комната:

Сообщение:

Отправить

Рисунок 50 – Окно отправки сообщения сотруднику

В результате отправления сообщения сотруднику о работе ему приходит сообщение на электронную почту, в котором указана комната и сообщение с дополнительной информацией (рисунок 51).



Рисунок 51 – Пример сообщение, приходящего сотруднику

Рядом с кнопкой отправки персонала есть кнопка «Быстрое добавление». Эта функция отвечает за добавление сразу нескольких комнат. В данной форме нужно будет вести тип комнат, их количество и вместимость, а также сразу можно добавить имеющееся имущество (рисунок 52).

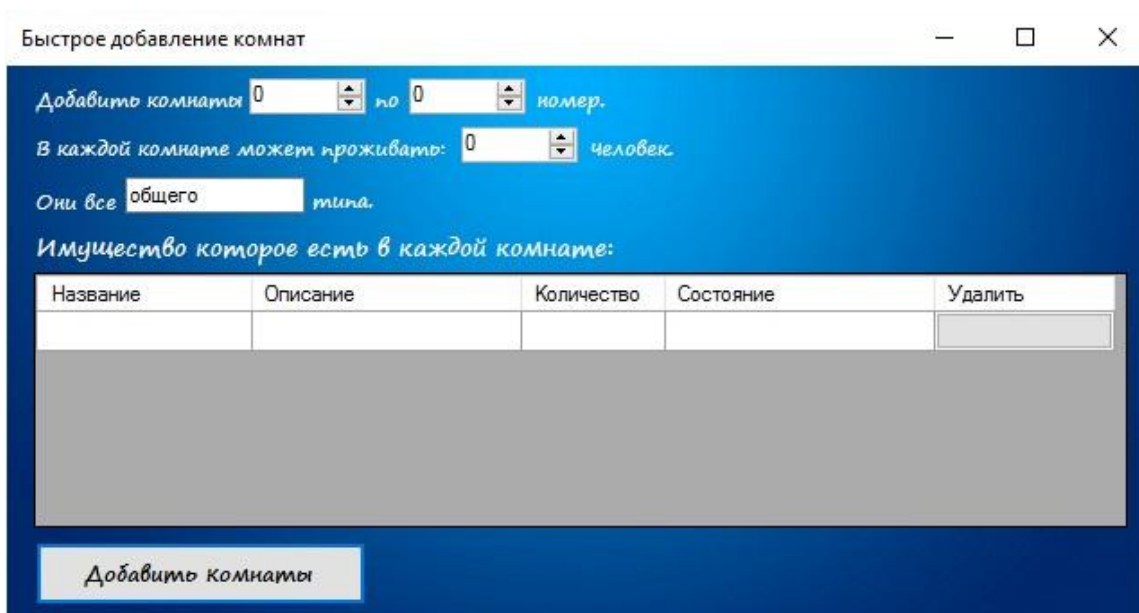


Рисунок 52 – Окно быстрого добавления комнат

Следующая форма доступная в меню это «Персонал» (рисунок 53).

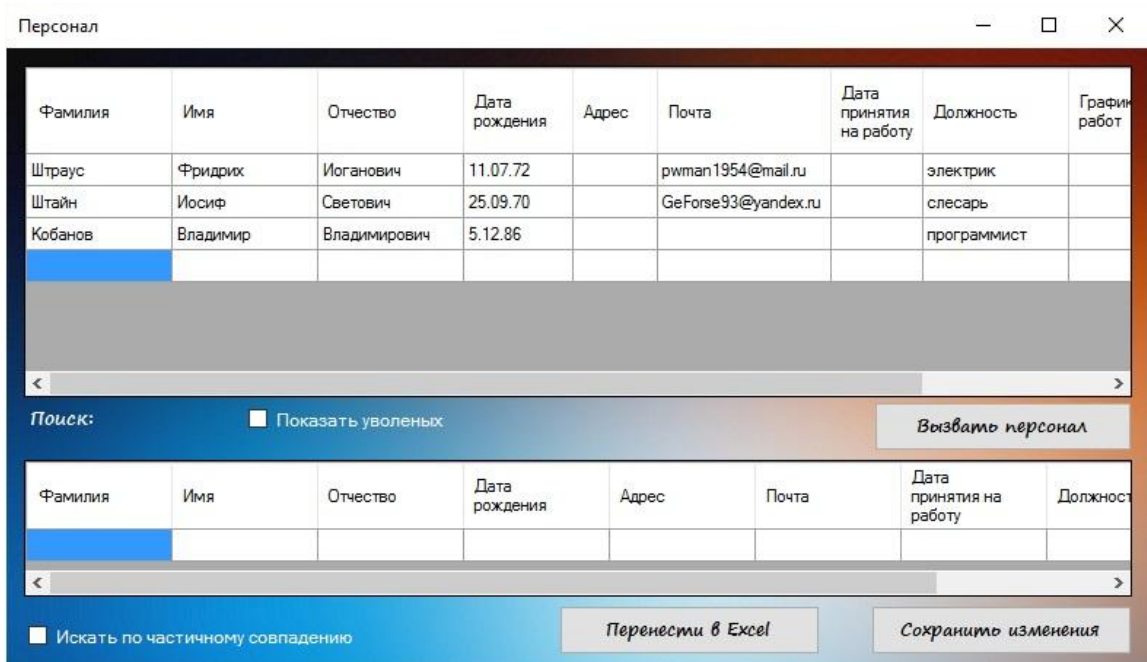


Рисунок 53 – Окно управления персоналом

Функции этой формы не отличаются от функций других форм. Также в здесь имеется функция для отображения уволенных сотрудников. Работает она также, как и функция показа выселенных студентов.

«Услуги» - это следующий раздел в главном меню (рисунок 54).

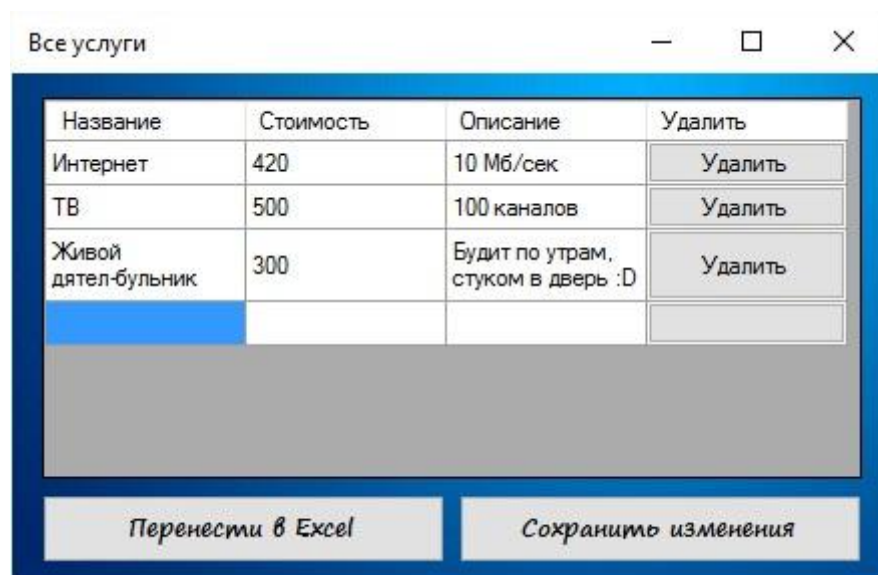


Рисунок 54 – Окно управления услугами

В этой форме создаются все доступные услуги в общежитие. Далее их можно будет выбрать в форме услуг для студента. Все доступные функции работают также, как и во всех остальных случаях.

Последняя доступная форма — форма управления мероприятиями (рисунок 55).



Рисунок 55 – Окно управления мероприятиями

Добавление мероприятий происходит также, как и добавление остальной информации в ранее приведённых формах. Здесь имеется календарь. Если нажать на нужное нам число и оно совпадёт с числом, на которое задано мероприятие, то оно загорится зелёным цветом (рисунок 56).

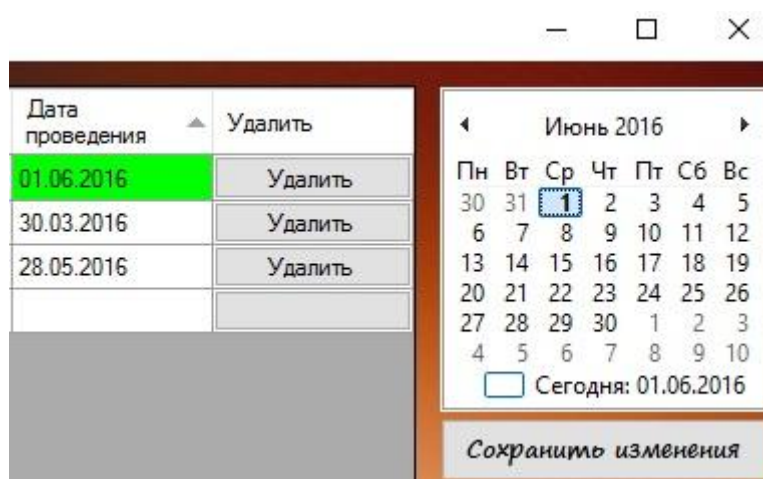


Рисунок 56 - Сообщение о неудачном сохранение

Если мероприятие задано на сегодняшнее число, то при входе в главное меню отобразится сообщение об этом (рисунок 57).

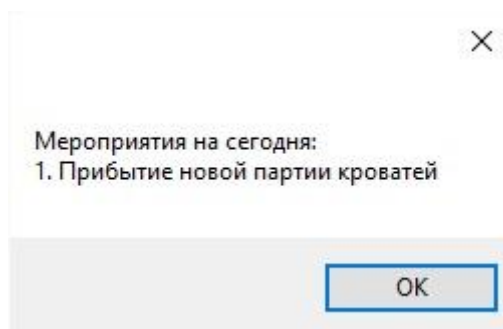


Рисунок 57 - Сообщение о запланированном мероприятии

3.2 Информационная безопасность ИС

Исходя из требований безопасности, необходимо организовать контролируемый доступ к данным.

Доступ к данным можно организовать при помощи логина и пароля.

При регистрации логин и пароль с помощью алгоритма SHA512 по отдельности будут хэшироваться и записываться в файл авторизации. Будет отсутствовать возможность просмотреть введённые данные. Также при хешировании пароля и логина их хэши складываются, каждый второй байт удаляется, и в результате получается ключ для симметричного шифрования. При включении приложения и ввода логина и пароля, они повторно будут хэшироваться и сравниваться с хэшем, который лежит в файле авторизации. При совпадении хэш-сумм приложение будет давать доступ пользователю к меню. При открытии любого пункта меню, в котором используются или отображаются зашифрованные данные, с помощью CryptoStream и ранее полученного ключа расшифровываются данные из соответствующего файла.

Также существует вариант того, что хэш может быть заменён или удалён в самом файле. В таком случае злоумышленник всё равно не сможет адекватно просмотреть или удалить данные, так как его ключ будет отличаться от того, которым данные были зашифрованы ранее.

SHA-2 является безопасным алгоритмом хеширования. Он относится к семейству криптографических алгоритмов, так называемых, однонаправленных хеш-функций, которые включают в себя такие алгоритмы: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256 и SHA-512/224. [35]

При помощи хеш-функций создаются «отпечатки» или «дайджесты» сообщения произвольной битовой длины, которые могут применяться в разных приложениях или элементах, связанных с защитой данных.

.NET Framework создан с поддержкой модели программирования в основе которого лежат потоки. Такие классы потоков, предоставляют информацию из разных хранилищ, например, текстовых файлов и дают возможность считывания данных или записи в них. CryptoStream, как раз, лежит в основе такой функциональности. Что бы защитить данные CryptoStream производит шифрование секретным ключом. Ключ для дешифрования не всегда один и тот же. Он выбирается в зависимости от алгоритма. CryptoStream входит в пространство имен System.Security.Cryptography. Этот класс используют для осуществления криптографических действий на объекте Stream.

Конструктор CryptoStream принимает такие параметры как: определение используемого потока; криптографическое преобразование; указание на доступ, чтение или запись к криптографическому потоку.

3.3 Сравнение ИС

Сравнение информационных систем может производиться по финансовым и не финансовым показателям.

Если рассматривать финансовые показатели, то можно выделить такие аспекты, как затраты на обучение, покупку лицензий и покупку технических средств.

При внедрении информационной системы в организацию или учреждение стоит необходимость обучения кадров, так как все новейшие информационные системы переполнены различными функциями, в которых необходимо разбираться. Также это сильно усложняет структуру пользовательского интерфейса. Но как показывает практика, такое обилие функций не всегда нужно пользователю. В разработанной информационной системе собраны только необходимые функции. Это позволяет поддерживать интуитивно понятный и доступный интерфейс, что в свою очередь делает необходимость обучения кадров только формальностью и от неё можно отказаться. На крайний случай, в разработанной системе существуют раздел помощи по управлению приложением.

Так же созданная информационная система значительно отличается от остальных систем тем, что не использует никакого стороннего программного обеспечения. То есть пользователю не придётся платить за лицензионный продукт для обеспечения непрерывной работы. В то время как большинство современных информационных систем используют такие программы. Обычно это программы для управления базами данных, так называемые СУБД. Наиболее используемыми являются Microsoft SQL Server, Oracle и MySQL. На все эти программные продукты довольно дорогая лицензия, что не очень хорошо влияет на финансовое состояние определённого учреждения или организации.

Ещё при использовании созданной информационной системы не потребуется мощной технической поддержки, что нельзя сказать о

большинстве современных продуктах. Загруженность операционной системы может происходить по той же причине использования дополнительных программ, что, как уже было сказано ранее, не является проблемой в случае использования разработанной мною системы.

При рассмотрении не финансовых показателей обычно учитывают такие критерии: функциональность, удобство и привлекательность пользовательского интерфейса, быстродействие системы в условиях, приближенных к реальному использованию, возможность доработки, то есть возможность добавления в код программы новых функций.

Что касается функциональности, разработанная программа не отличается какими-то сильными нововведениями от всех современных приложений, но в ней содержатся все необходимые для управляющего общежитием функции. Они выбраны с учётом пожеланий коменданта и тьютора общежития. Такая минимизация количества функций позволяет не загромождать программу и использовать только действительно нужное в работе.

Если рассматривать уникальность каждой функции в отдельности, то в разработанной мною системе имеются такие функции, которых нет в других информационных системах такого вида. Например, такие: функция автоматической отправки сообщения студентам о задолженности за проживание; функция ведения списка запланированных мероприятий и оповещения о них управляющему; функция перевода информации из базы данных системы в другие форматы, например, в формат программы Microsoft Office Excel.

Удобство и привлекательность пользовательского интерфейса может показаться не важной частью информационной системы, но это не так. От этого зависит отношение пользователя к программе. В разработанной программе интерфейс подобран так что бы у пользователя возникало желание работать и заходить в приложение и это было удобно. В то же время он не слишком отвлекает и не бросается в глаза (рисунок 58).

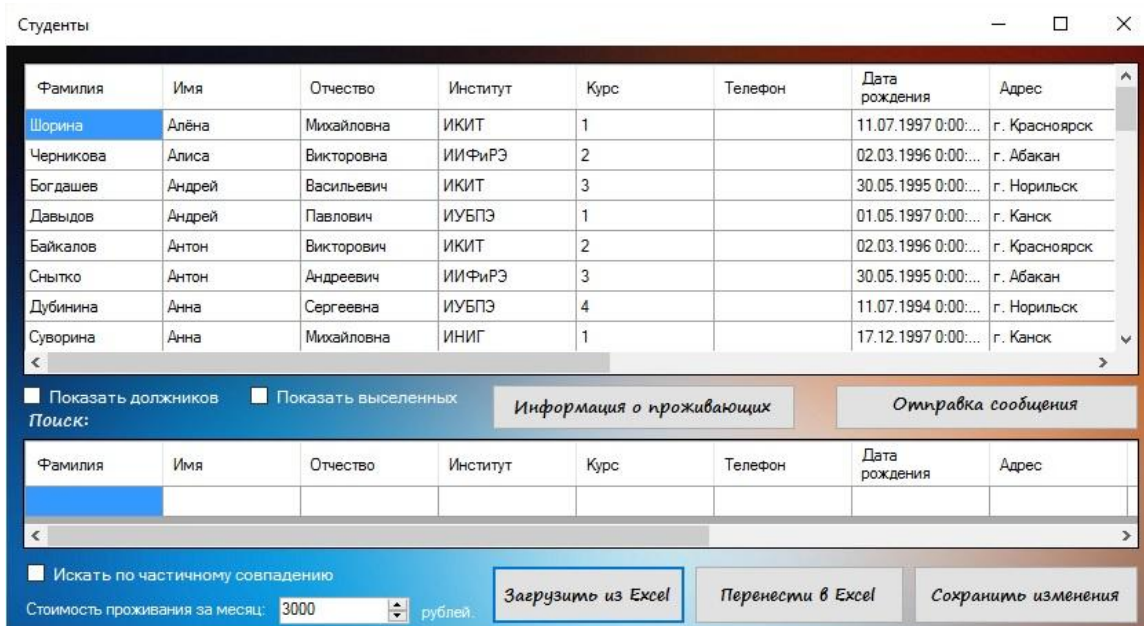


Рисунок 58 - Интерфейс одного из разделов приложения

Все основные кнопки и функции доступны и их не нужно искать. Всё просто, доступно и привлекательно для любого пользователя.

Другие программы подобного рода не всегда отличаются такой привлекательностью и удобством (рисунок 59).

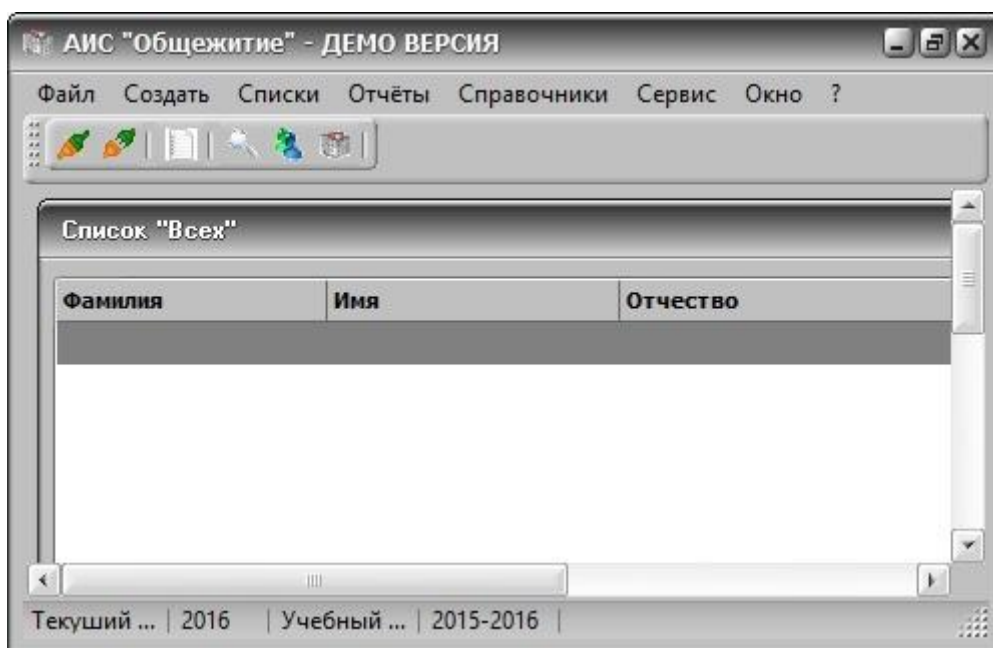


Рисунок 59 - Пример интерфейса АИС «Общежитие»

Данная программа не отличается привлекательным интерфейсом. Все функции скрыты во вкладках и их довольно трудно найти на первый взгляд. Есть кнопки, которые не никак не обозначены и не ясно какую функцию они выполняют.

Быстродействие системы осуществляется опять же за счёт отсутствие взаимодействия со сторонними программами, а также за счёт малой требовательности к техническому обеспечению.

3.4 Оценка качества ИС

С точки зрения ИСО/МЭК 9126 [36], качество информационной системы можно определить, как совокупность характеристик программного средства, при этом учитывая следующие составляющие:

- **Функциональные возможности** — это способность программного средства предоставлять решение задач, которые удовлетворяют необходимым потребностям пользователей при применении приложения в некоторых заданных условиях;
- **Функциональная пригодность** — это набор и описание субъективных характеристики и ее атрибутов, которые определяют назначение и номенклатуру, а также основные, необходимые и достаточные функции программного средства. Всё это должно соответствовать техническому заданию и спецификациям требований заказчика или потенциального пользователя системой;
- **Правильность** — это способность приложения обеспечивать правильные или приемлемые для заказчика или пользователя результаты;
- **Способность к взаимодействию** — это свойство программного средства и его компонентов взаимодействовать с одним или многими компонентами внутренней и внешней среды;

- Защищённость — это способность элементов программного средства защищать саму программу и содержащуюся в ней информацию от любого рода негативных воздействий;

- Надежность работы системы определяется надежностью её функциональных компонентов. Этими компонентами являются: программное обеспечение, совокупность технических и инженерных средств;

- Сопровождаемость — это предрасположенность программного средства к различным модификациям и изменению конфигураций и функций;

- Практичность — это свойства программного средства, которые определяют степень сложности его понимания, изучения и использования. Также практичность отвечает за привлекательность приложения для квалифицированных пользователей при применении в некоторых условиях;

- Эффективность — это свойства программного средства, обеспечивающие нужную производительность решения функциональных задач информационной системы, с обязательным учётом наличия используемых вычислительных ресурсов при установленных пользователем условиях;

- Мобильность — это подготовленность программного средства к переносу из одной аппаратно-операционной среды в какую-либо другую.

После анализа критериев качества информационных систем, можно сказать, что для спроектированной системы технической поддержки они будут следующими:

- Функциональные возможности. Функциональные возможности удовлетворяют потребностям, которые были сформулированы потенциальным пользователем, при применении информационной системы. Таким образом, функциональные возможности высокие.

- Функциональная пригодность. Исходя из технического задания, данная информационная система имеет все необходимые функциональные составляющие. Следовательно, функциональная пригодность высокая.

- **Правильность.** Программное средство выдаёт правильные и оптимальные для пользователя результаты. Таким образом, правильность высокая.
- **Защищенность.** Защита системы организована при помощи алгоритма шифрования SHA-2 и встроенного в .NET Framework конструктора CryptoStream, но при должном умении это можно обойти и испортить зашифрованные данные. Но всё равно нельзя их прочитать. Из этого следует, что защищенность можно определить, как среднюю.
- **Надежность.** Проект использует оптимальное количество функций, что положительно сказывается на надежности. Таким образом, надёжность средняя.
- **Сопровождаемость.** Так как система разработана с использованием фреймворка .NET Framework, то ее можно легко модифицировать и увеличивать ее функционал и масштаб. Следовательно, сопровождаемость высокая.
- **Практичность.** Система имеет простой и понятный пользовательский интерфейс. Благодаря этому, она будет удобна для любого пользователя. Таким образом, практичность высокая.
- **Эффективность.** Информационной системе для своевременного выполнения задач и функций требуется очень мало вычислительных ресурсов. Следовательно, эффективность высокая.
- **Мобильность.** Так как информационная система технической поддержки реализована с помощью средства Visual Studio, то проблем с ее возможным переносом не возникнет.

3.5 Внедрение

Для внедрения были выполнены следующие этапы:

- Утверждение проекта;
- Введение в представление о системе персонала технической поддержки (предназначение информационной системы, какие задачи реализует, где и как будет использоваться);
 - Уточняются цели и задачи проекта (беседа с управляющим общежития с целью формирования задач и требований к информационной системе);
 - Согласуются план работы, последовательность этапов и условия опытной эксплуатации.

3.6 Вывод по главе 3

В первом разделе главы был представлен сценарий работы пользователя с системой. Таким образом были продемонстрированы все возможности пользователя в системе. В разделе «Информационная безопасность ИС» были описаны способы обеспечения информационной безопасности с помощью фреймворка различных алгоритмов шифрования. В следующем разделе – разделе «Сравнение ИС» была произведена сравнительная характеристика современных информационных систем с созданной. Также в главе был проведен анализ качества информационной системы, у большинства критериев качества преобладают высокие показатели. В разделе «Внедрение» были представлены этапы внедрения системы в организацию.

ЗАКЛЮЧЕНИЕ

Результатом выполненной работы является desktop-приложение, предназначенное для управления общежитием. В нём содержатся функции отправки сообщений студентам и сотрудникам. А также реализован процесс отслеживания запланированных мероприятий.

В данной работе представлен процесс проектирования системы. В начале проектирования системы были поставлены определённые задачи. Их поэтапное решение привело к реализации системы.

При проектировании системы были определены средства разработки. Анализ этих средств помог установить преимущества разработки и определить особенности подобных средств.

При выполнении следующего этапа проектирования системы были сформулированы требования к проекту. Формулировка требований и их утверждение привели к началу этапа построения системы. На этом этапе был создан пользовательский интерфейс проекта, разработана база данных и построены различные диаграммы.

Последним этапом стало описание системы. На этом этапе описывается сценарий действий управляющего при взаимодействии с информационной системой. Также были определены средства обеспечения безопасности системы, произведено сравнение с другими системами и проведен анализ ее надежности.

Спроектированная система поможет сократить время работы и повысить производительность труда управляющему общежитием.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Свободная энциклопедия [Электронный ресурс]: Информационная система. // «Википедия». - Режим доступа: <https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0>
2. Студенческий информационный ресурс [Электронный ресурс]: Структура информационной системы. // «Ксиэит». - Режим доступа: http://www.xsieit.ru/download/design_of_information_systems/lectures/873.html
3. Всё для программиста [Электронный ресурс]: Язык С. // «КодНэт». - Режим доступа: <http://www.codenet.ru/progr/cpp/6/>
4. Энциклопедия языков программирования [Электронный ресурс]: С++. // «Прогопедия». - Режим доступа: <http://progopedia.ru/language/c-plus-plus/>
5. Онлайн библиотека [Электронный ресурс]: Язык и архитектура Java. // «СитФорум». - Режим доступа: <http://citforum.ru/internet/iinet96/17.shtml>
6. Всё о языках программирования [Электронный ресурс]: С#. // «АльтКод». - Режим доступа: <http://altcode.ru/c-sh/>
7. Государство. Бизнес. ИТ [Электронный ресурс]: Microsoft Visual Studio. // «Тадвизэр». - Режим доступа: http://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:Microsoft_Visual_Studio
8. Типичный программист [Электронный ресурс]: 5 полезных расширений для Visual Studio. // «ТПрогер». - Режим доступа: <https://tproger.ru/articles/5-useful-extensions-for-vs/>
9. Свободная энциклопедия [Электронный ресурс]: Фреймворк. // «Википедия». - Режим доступа:

<https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA>

10. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: .NET Framework. // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/cc738855\(v=ws.10\).aspx](https://msdn.microsoft.com/ru-ru/library/cc738855(v=ws.10).aspx)

11. Свободная энциклопедия [Электронный ресурс]: Windows Forms. // «Википедия». - Режим доступа: https://ru.wikipedia.org/wiki/Windows_Forms

12. Национальный Открытый Университет [Электронный ресурс]: База данных и СУБД. Внедрение в SQL. // «ИНТУИТ». - Режим доступа: <http://www.intuit.ru/studies/courses/42/42/lecture/27193>

13. Научная библиотека избранных естественно-научных изданий [Электронный ресурс]: Проектирование баз данных. // «Сернам». - Режим доступа: http://sernam.ru/book_cbd.php?id=2

14. Создаём и раскручиваем сайт самостоятельно [Электронный ресурс]: Урок 3. Реляционные базы данных. // «СайтДу». - Режим доступа: <http://www.site-do.ru/db/db3.php>

15. Национальный Открытый Университет [Электронный ресурс]: База данных: модели, разработка, реализация. // «ИНТУИТ». - Режим доступа: <http://www.intuit.ru/studies/courses/1001/297/lecture/7401>

16. Вся помощь студенту и школьнику. [Электронный ресурс]: Понятие и виды интерфейсов. // «Студалл». - Режим доступа: <http://studall.org/all-95003.html>

17. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Создание пользовательского интерфейса (Visual C#) // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/ms173080\(v=vs.90\).aspx](https://msdn.microsoft.com/ru-ru/library/ms173080(v=vs.90).aspx)

18. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Элементы управления Windows Forms // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/ettb6e2a\(v=vs.100\).aspx](https://msdn.microsoft.com/ru-ru/library/ettb6e2a(v=vs.100).aspx)

19. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Элемент управления Button (Windows Forms) // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/s0scsyfb\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/s0scsyfb(v=vs.110).aspx)

20. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Элемент управления Label (Windows Forms) // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/4k7d0bhs\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/4k7d0bhs(v=vs.110).aspx)

21. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Элемент управления TextBox (Windows Forms) // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/sbxka1kb\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/sbxka1kb(v=vs.110).aspx)

22. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Элемент управления TableLayoutPanel (Windows Forms) // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/3a1tbfwd\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/3a1tbfwd(v=vs.110).aspx)

23. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Элемент управления HScrollBar и VScrollBur (Windows Forms) // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/4zfkbw2e\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/4zfkbw2e(v=vs.110).aspx)

24. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Элемент управления CheckBox (Windows Forms) // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/dzddyc7z\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/dzddyc7z(v=vs.110).aspx)

25. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Элемент управления NumericUpDown (Windows Forms) // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/729xt55s\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/729xt55s(v=vs.110).aspx)

26. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Элемент управления MonthCalendar (Windows Forms) //

«МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/3ehzhyst\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/3ehzhyst(v=vs.110).aspx)

27. Студенческий научный форум [Электронный ресурс]: Цветовое решение интерфейса как средство повышения эффективности работы ИС. // «СайнсФорум». - Режим доступа: <http://www.scienceforum.ru/2015/pdf/12143.pdf>

28. Студенческий научный форум [Электронный ресурс]: Цветовое решение интерфейса как средство повышения эффективности работы ИС. // «СайнсФорум». - Режим доступа: <http://www.scienceforum.ru/2015/857/10761>

29. Национальный Открытый Университет [Электронный ресурс]: Категории информационных систем. // «ИНТУИТ». - Режим доступа: <http://www.intuit.ru/studies/courses/1055/271/lecture/6876>

30. Ресурс для IT-специалистов [Электронный ресурс]: UML-диаграмма вариантов использования (use case diagram). // «Хабрахабр». - Режим доступа: <https://habrahabr.ru/post/47940/>

31. Консалтинг и автоматизация [Электронный ресурс]: Основные типы методологий моделирования и анализа бизнес-процессов. // «СитиКонсалтингГруп». - Режим доступа: <http://www.citycg.ru/services/business-process/metody-opisaniya-processov/>

32. Справочные материалы по информационным технологиям. [Электронный ресурс]: Методология IDEF0. // «АйтиТич». - Режим доступа: <http://itteach.ru/bpwin/metodologiya-idef0/all-pages>

33. Проектирование и разработка автоматизированных, информационных и аналитических систем. [Электронный ресурс]: Диаграмма развёртывания (deployment diagram). // «ИнфоСистем». - Режим доступа: http://www.info-system.ru/designing/methodology/uml/theory/deployment_diagram_theory.html

34. Информационный ресурс [Электронный ресурс]: Этапы проектирования БД. Цели и виды работ на этапах концептуального,

логического и физического проектирования. // «Студопедия». - Режим доступа: <http://studopedia.org/8-228774.html>

35. Свободная энциклопедия [Электронный ресурс]: SHA-2. // «Википедия». - Режим доступа: <https://ru.wikipedia.org/wiki/SHA-2>

36. Единая база гостов РФ [Электронный ресурс]: Средства и языки разработки. // «ГостЭксперт». - Режим доступа: <http://gostexpert.ru/gost/gost-9126-93>

37. СТО 4.2–07–2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 9.01.2014 – Красноярск : ИПК СФУ, 2014. – 60 с.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Информационные системы»

УТВЕРЖДАЮ
Заведующий кафедрой ИС
_____ С.А.Виденин
подпись
«__» _____ 2016 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту Трухан Алексей Михайлович

Группа: КИ12-13Б Направление: 09.03.02 «Информационные системы и технологии»

Тема выпускной квалификационной работы: «Разработка автоматизированной информационной системы управления общежитием»

Утверждена приказом по университету №4729/с от 05.04.2016 г.

Руководитель ВКР: Е. В. Головчанская, старший преподаватель кафедры «Информационные системы» ИКИТ СФУ.

Исходные данные для ВКР: список требований к разрабатываемой системе, методические указания научного руководителя.

Перечень разделов ВКР: введение, общие сведения, проектирование ис, описание и сравнение ис, заключение, список использованных источников.

Перечень графического или иллюстрированного материала с указанием основных чертежей, плакатов, слайдов: презентация, выполненная в Microsoft Office PowerPoint 2013.

Руководитель ВКР

(подпись)

Е.В.Головчанская

Задание принял к исполнению

(подпись)

А.М.Трухан

«__»_____ 2016 г.