

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт Космических и Информационных Технологий
Информационные Системы

УТВЕРЖДАЮ
Зав. кафедрой ИС
_____ С. А. Виденин
подпись инициалы, фамилия
« _____ » _____ 2017 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 Информационные системы и технологии

АРМ ответственного по заселению в общежития

Руководитель

подпись, дата

Ю. В. Шмагрис
инициалы, фамилия

Выпускник

подпись, дата

И. А. Русак
инициалы, фамилия

Консультант

подпись, дата

И. А. Легалов
инициалы, фамилия

Нормоконтролер

подпись, дата

Ю. В. Шмагрис
инициалы, фамилия

Красноярск 2017

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

«02» марта 2017 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту Русаку Илье Андреевичу
фамилия, имя, отчество

Группа КИ13-13Б Направление 09.03.02
номер код

Информационные системы и технологии
наименование

Тема выпускной квалификационной работы: АРМ ответственного по заселению в общежития

Утверждена приказом по университету № 2517/с от 01.03.2017г.

Руководитель ВКР: Ю. В. Шмагрис, старший преподаватель кафедры «Информационные системы» ИКИТ СФУ

Консультант: Легалов И. А. к.т.н., доцент кафедры «Информационные системы» ИКИТ СФУ
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР: Техническое задание, рекомендации руководителя

Перечень разделов ВКР: Введение, теоритическая часть, проектирование АРМ, описание и сравнение АРМ, заключение, список использованных источников

Перечень графического материала: Презентация, выполненная в Microsoft Office PowerPoint 2016.

Руководитель ВКР

подпись

Ю. В. Шмагрис
инициалы и фамилия

Консультант

подпись

И. А. Легалов
инициалы и фамилия

Задание принял к исполнению

подпись

И. А. Русак
инициалы и фамилия студента

« 02 » марта 2017 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «АРМ ответственного по заселению в общежития» содержит 53 страницы текстового документа, 27 иллюстраций, 21 использованный источник и одну таблицу.

Цель работы – разработать автоматизированное рабочее место для ответственного по заселению в общежития, с помощью которого будет осуществляться взаимодействие сотрудника с различными информационными аспектами его деятельности.

Задачи работы:

- Анализировать предметную область;
- Выбрать средства и среду разработки;
- Спроектировать АРМ;
- Разработать АРМ с помощью программных средств.

В результате ВКР было спроектировано и разработано web-приложение, предназначенное для управления и контроля процесса расселения. В нём содержатся функции составления документов. А также реализован процесс автоматического редактирования базы данных, после подписания распоряжения. Спроектированная система поможет сократить время работы и повысить производительность труда ответственного по заселению в общежития.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Теоретическая часть	6
1.1 Общие сведения об АРМ	6
1.2 Возможные решения	8
1.3 Выбор средств разработки.....	10
1.4 Базы данных	12
1.5 Способ доступа к базе данных	14
1.6 Вывод по главе 1	15
2 Проектирование АРМ.....	16
2.1 Формирование требований.....	16
2.2 Проектирование структуры приложения и разработка пользовательского интерфейса.....	17
2.3 Проектирование функциональной части приложения	22
2.4 Проектирование базы данных	30
2.5 Вывод по главе 2.....	34
3 Описание и внедрение АРМ	36
3.1 Сценарий работы пользователя с приложением	36
3.2 Информационная безопасность АРМ.....	43
3.3 Оценка качества ИС	44
3.4 Внедрение.....	46
3.5 Результат внедрения.....	47
3.6 Вывод по главе 3	49
ЗАКЛЮЧЕНИЕ	50
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ:	51

ВВЕДЕНИЕ

Современное человеческое общество живет в период, характеризующийся небывалым увеличением информационных потоков, и этот процесс всеобщей гиперинформатизации охватывает все области человеческой деятельности.

Одной из целей применения технологий автоматизации является повышение эффективности деятельности человека, для этой цели предназначены различные формы и методы управления организацией с применением вычислительной техники и средств связи. Комплекс технических и программных средств, выполняющих автоматизированную обработку данных в процессе управления организацией, образует автоматизированное рабочее место(АРМ).

Ответственный по заселению — профессия, связанная с работой и обработкой больших объёмов информации. Сегодня, во многих учебных заведениях, 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.

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 автоматизированного рабочего места для ответственного по заселению в общежития, с помощью которого будет осуществляться взаимодействие сотрудника с различными информационными аспектами его деятельности.

Для достижения поставленной цели были выделены и выполнены следующие задачи:

- Анализировать предметную область;
- Выбрать средства и среду разработки;
- Спроектировать АРМ;
- Разработать АРМ с помощью программных средств.

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

Этот продукт представляет собой базу данных Microsoft SQL Server под управлением web приложения на языке Си#.

Данное АРМ будет использоваться сотрудниками учебного заведения. Вход в систему будет производиться после авторизации пользователя непосредственно в самом приложении.

В качестве направлений автоматизации информационной системы можно выделить такие аспекты:

- Возможность составления нормативных документов;
- Возможность отслеживания места проживания студентов;
- Возможность автоматического распределения мест среди студентов;
- Удобство работы с системой.

1 Теоретическая часть

1.1 Общие сведения об АРМ

Возможности современных компьютерных технологий для автоматизации обработки информации могут повысить производительность труда, эффективность управления документами и ускорить обмен управления информацией.

На данный момент большое распространение имеет концепция распределенных автоматизированных систем управления, направленных на обработку информации. Это позволяет организовать разделение управления труда и персонала для автоматизации своих функций. Для реализации данной идеи необходимо создать для каждой предметной области автоматизированные рабочие места на базе персональных компьютеров.

Автоматизированное рабочее место (далее по тексту АРМ) – совокупность аппаратных и программных средств, которые находятся непосредственно в рабочем месте сотрудника и предназначены для автоматизации их работы по специальности.

АРМ имеют проблемы профессиональной направленности в конкретной предметной области и являются средством связи специалиста с автоматизированными информационными системами.

Автоматизированное рабочее место специалиста обеспечивает:

- простота, удобство и дружественное отношение к пользователю;
- легкая адаптация к конкретным функциям пользователя;
- компактное расположение и низкие требования к условиям труда;
- высокая надежность и выживание;
- сравнительно простую организацию технического обслуживания;
- мобильность работников;
- повышение производительности.

Эффективным режимом работы АРМ есть его функционирование в рамках локальной вычислительной сети. Созданные АРМ специалистов дают возможность пользователю работать в диалоговом режиме, оперативно решать текущие задачи, удобно вводить данные, вести контроль, обработку информации, определять достоверность результатной информации, выводить и передавать каналами связи. Информационное обеспечение АРМ ориентируется на конкретную, обычную для пользователя, предметную сферу [1].

Автоматизированные рабочие места должны быть установлены в соответствии с их предполагаемой функциональностью. Тем не менее, общие принципы остаются неизменными

- системность;
- гибкость;
- устойчивость;
- эффективность.

Под системностью стоит понимать следующее: рабочая станция должна представлять собой систему взаимосвязанных компонентов. Эта структура должна четко выполнять функции, для выполнения которых создается эта рабочая станция.

Принцип гибкости имеет первостепенное значение в создании современного и эффективного функционирования рабочих мест. Этот принцип означает, что АРМ может быть адаптирован к предлагаемому обновлению программного обеспечения и оборудования. В настоящее время, скорость старения оборудования и программного обеспечения продолжает расти, соблюдение этого правила становится одним из наиболее важных условий для создания новых автоматизированных рабочих мест.

Для обеспечения принципа гибкости в работающих АРМ все подсистемы отдельно взятого автоматизированного рабочего места должны выполняться в виде отдельных модулей, которые легко заменить.

Стандартизация всех элементов необходима для того, чтобы при замене модулей не возникало несовместимости.

Высокое значение имеет принцип развития устойчивости. Он заключается в выполнении функций, заложенных в АРМ, независимо от влияния как внешних факторов, так и внутренних. В случае выхода из строя работоспособность системы должна быстро восстанавливаться, а неполадки у отдельных элементов должны скорейшим образом быть устранены.

1.2 Возможные решения

Аналогичных решений на рынке решающих проблему расселения не так уж и много, вот тот список программных продуктов, которые помогают выполнять свои служебные обязанности работнику склада:

- ExaHostel
- БИТ.Общежитие.

ExaHostel — программное средство для автоматизации учета хостела и общежития. Это база данных, содержащая в себе всю информацию об номерах, гостях, услугах, контрагентах и не только это в режиме реального времени, онлайн, из любой точки мира. Программа связывает руководителя, администратора, бухгалтера и других сотрудников, обеспечивая всех нужной, точной и своевременной информацией.

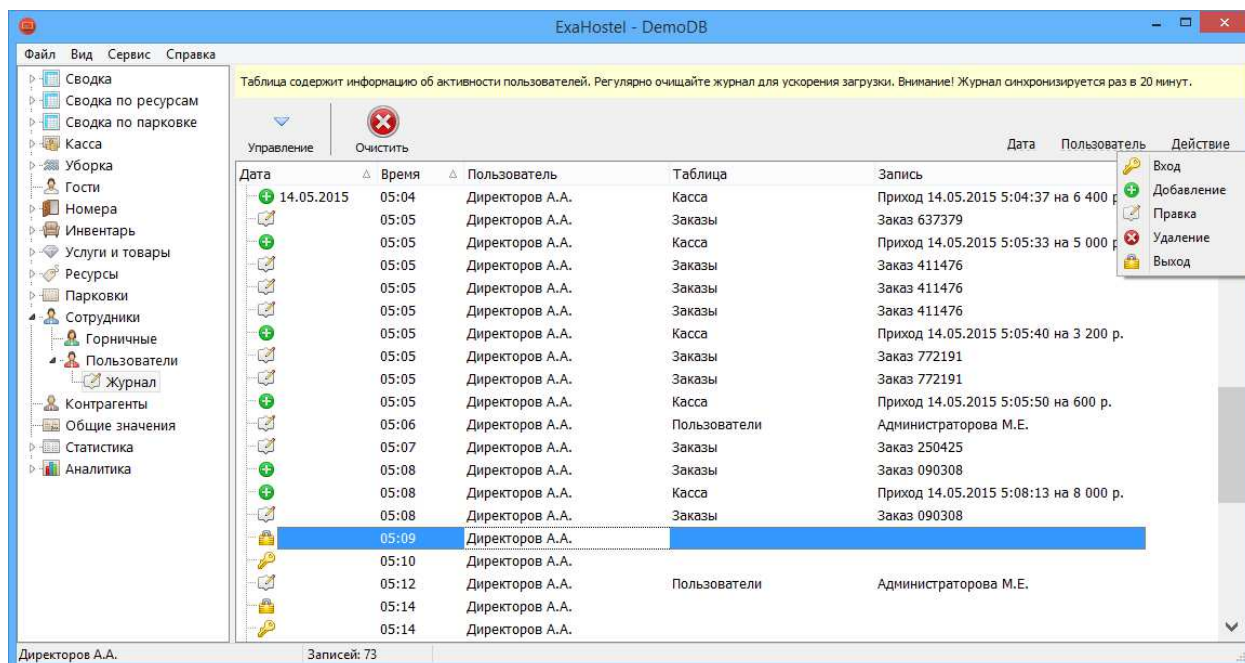


Рисунок 1 – ExaHostel

Однако ExaHostel имеет ряд недостатков:

- Данная система более предназначена для коммерческих структур и не рассчитана на изменения в статусе проживающих;
- Высокая цена системы. Приобретение всех продуктов линейки обходится в большую сумму денег.

БИТ.Общежитие представляет собой полнофункциональное решение для автоматизации учета в общежитии. Программа предназначена для автоматизации учета движения денежных средств при оказании услуг проживания общежитиями вузов, финансирование которых осуществляется как за счет бюджетных средств, так и за счет собственных средств вуза.

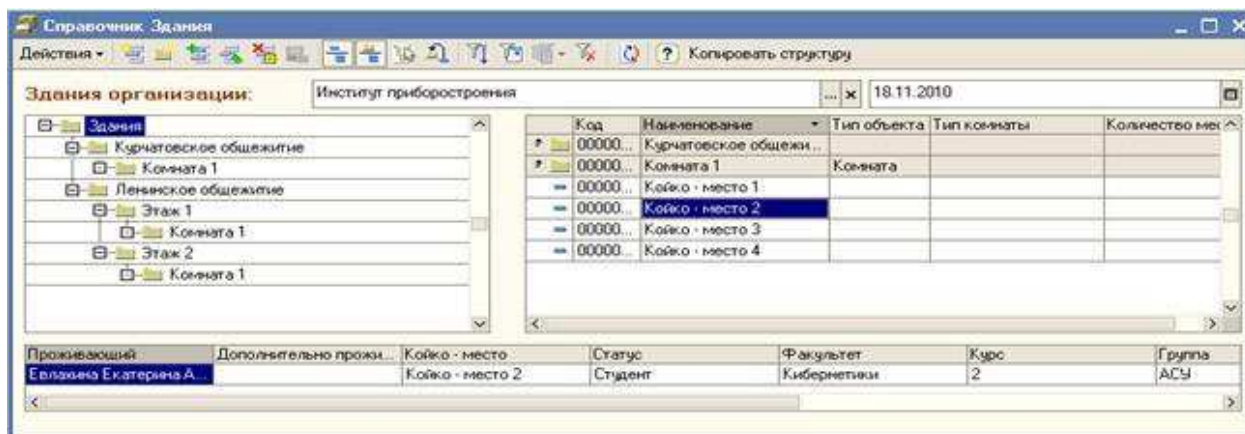


Рисунок 2 - БИТ.Общежитие

У этого продукта также имеются недостатки:

- Данный продукт предназначен заведующей общежития и не совпадает с порядком заселения в данном учреждении;
- Для реализации данного решения необходимо приобрести "1С:Предприятие 8.2", который также обходится приличную сумму денег.

Рассмотрев и проанализировав аналогичные системы, мы пришли к выводу, что лучшим решением поставленной цели станет разработка собственной системы которая покроет все бизнес-процессы, присущие данной предметной области. А также, предусмотрит возможность обновления и расширения системы новыми модулями.

1.3 Выбор средств разработки

В качестве среды для разработки проекта был выбран Visual Studio (рисунок 3). Это линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех

платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight [2].

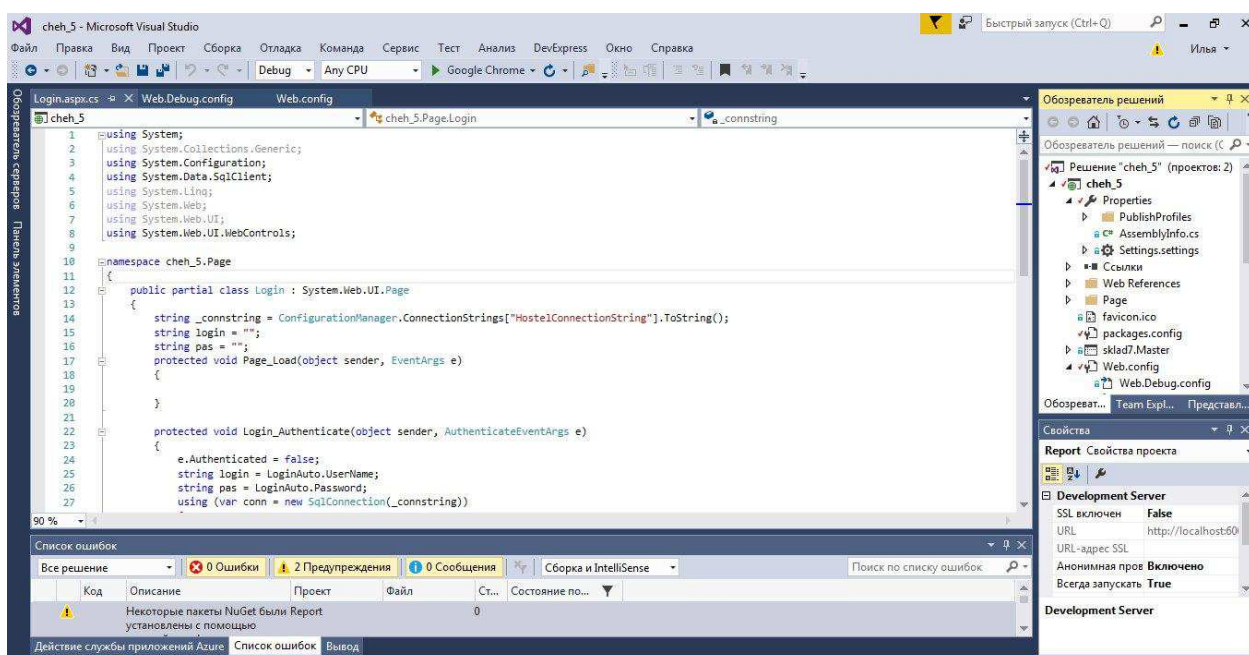


Рисунок 3 - Интерфейс Visual Studio

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.

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 [3].

.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.

ASP.NET (Active Server Pages для .NET) — технология создания веб-приложений и веб-сервисов от компании Майкрософт. Она является составной частью платформы Microsoft .NET и развитием более старой технологии Microsoft ASP [4].

1.4 Базы данных

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

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.

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 [5].

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.

Microsoft SQL Server система управления реляционными базами данных, разработанная корпорацией Microsoft. Основным используемый язык запросов Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

Благодаря более низкой совокупной стоимости владения, повышенной гибкости и обширной сети партнеров Windows Server соответствует возможностям или даже превосходит возможности систем для мэйнфреймов и средних компьютеров, а также систем UNIX и Linux.

В комплектацию SQL Server входит набор расширенных средств управления, таких как Management Studio, Performance Studio Во всех

областях, связанных с управлением данными и являющихся существенными для бизнеса. Непревзойденный уровень безопасности и надежности, беспримерные возможности интеграции данных и не имеющий равных показатель совокупной стоимости владения (TCO).

Был выбран MSSQL Server, т.к. мы используем систему .NET для реализации нашего проекта, которая является разработкой Microsoft, т.е. интеграция между .NET и MSSQL Server будет проще, чем с другими СУБД [6].

1.5 Способ доступа к базе данных

В проекте было решено использовать ADO.NET (ActiveX Data Object для .NET). Это обусловлено тем, что в проекте небольшое количество запросов и при написании их вручную повысится скорость работы с БД.

ADO.NET — это набор классов, предоставляющих службы доступа к данным программистам, которые используют платформу .NET Framework. ADO.NET имеет богатый набор компонентов для создания распределенных приложений, совместно использующих данные. Это неотъемлемая часть платформы .NET Framework, которая предоставляет доступ к реляционным данным, XML-данным и данным приложений. ADO.NET удовлетворяет различные потребности разработчиков, включая создание клиентских приложений баз данных, а также бизнес-объектов среднего уровня, используемых приложениями, средствами, языками и браузерами [7].

У ADO.NET есть ряд преимуществ:

- Взаимодействие
- Простота модификации
- Простота программирования
- Производительность
- Масштабируемость

1.6 Вывод по главе 1

В данной части была рассмотрена общая информация об автоматизированных рабочих станциях, их актуальности, основных плюсах и главных принципах использования системы. Изучены подобные готовые аналоги данной системы, а также проведено сравнение и выявление у них ряда недостатков перед разрабатываемой системой. В результате обзора средств разработки был определен набор инструментов, способствующих проектированию и созданию информационной системы. Средством разработки был выбран язык C#. Выбор основывается на том, что он является наиболее подходящим для программируемого приложения. Так же 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 Web Forms. Так же была выбрана виды баз данных, способы их создания и варианты доступа к ним. Также было написано техническое задание для проекта с описанием функционала и основных этапов для создания автоматизированного рабочего места.

2 Проектирование АРМ

2.1 Формирование требований

В ходе проектирования автоматизированного рабочего места ответственного по заселению в общежитие, были сформированы следующие задачи:

- Упростить процесс заселения в общежитие;
- Упростить процесс взаимодействия сотрудника с работниками управления общежитиями;
- Упростить процесс взаимодействие с проживающими в общежитие студентами;
- Упростить процесс извлечения и нахождения нужной информации.
- Приложение должно работать на всех современных браузерах (Google Chrome, Mozilla Firefox, Opera), а также поддерживать Internet Explorer 6.0.

Требования к функциональности

Требования к функциональности такие:

- Реализовать процесс просмотра информации о проживающих студентов в общежитие;
- Создать кэш для распоряжений;
- Создать инструмент автоматического расселения по комнатам;
- Создать в приложении механизм переселения студента в другую комнату;
- Реализовать поиск по полному и частичному совпадению;

- Создать инструмент перевода информации из таблицы в pdf файл;
- Реализовать контролируемый доступ в приложение;
- Реализовать инструмент для навигации по приложению.

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;

Технические требования

Технические требования ко всему проекту:

- Использовать язык программирования C#;
- Небольшая нагрузка на сервер;
- 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, т.е. 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 [8].

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

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

Технологию ASP.NET WebForms можно рассматривать как некоторую надстройку над классическим принципом web-программирования «запрос-ответ». Данная технология является примером концепции RAD (rapid application development — быстрая разработка приложений), позволяя разработчикам максимально быстро и комфортно создавать рабочие приложения [9].

В Web Forms при создании пользовательского интерфейса существуют три этапа:

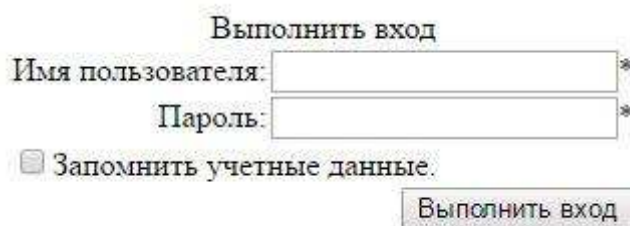
- Добавление различных элементов управления на поверхность разработки системы;
- Установка определённых свойств для элементов управления;
- Написание обработчиков для всех заданных событий [8].

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.

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, одна кнопка для совершения действия вход в программу.



Выполнить вход

Имя пользователя: *

Пароль: *

☐ Запомнить учетные данные.

Выполнить вход

Рисунок 4 - Структура окна авторизации

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

Элемент управления «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 [10].

Элементы управления Web Forms «Label» необходимы для вывода 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 вывода 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 [11].

Текстовые поля форм Web 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 [12].

Второе окно — это «Главное меню». Так же, как и в первом случае структура окна очень проста (рисунок 5). Окно содержит такие разделы как: «Распоряжение», «Отчёты» и «Загрузка массивов».

<p>Распоряжение</p> <p>Абитуриенты</p> <p>Студенты</p>	
<p>Отчёты</p> <p>Свободные места</p> <p>Занятые места</p> <p>Все места закрепленные за ИКИТ СФУ</p>	
<p>Загрузка массивов</p> <p>Абитуриенты</p> <p>Студенты</p>	

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

#	Фамилия	Имя	Отчество	Группа	Курс	Полная	Общ./Ком.	Срок проживания
						<input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="text"/>

Нет данных для отображения

Заселить

#	Имя	Фамилия	Отчество	Группа	Курс	Тип обучения	Общ./комната
						<input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>

Нет данных для отображения

Выселить

#	Имя	Фамилия	Отчество	Группа	Курс	Тип обучения	От курс (общ./ком.)	Курд (общ./ком.)	Срок ожидания
						<input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="text"/>

Нет данных для отображения

Перезелить

#	Записка	Фамилия	Имя	Отчество	Группа	Курс	Полная	Общая/комн.	Кан-во мест	Сотрудник	Дата заселения	Ген
Редактировать		61403	Сергеевич	Михайл	Александрович	KH14-166					3	<input checked="" type="checkbox"/>
Редактировать		120226	Долгов	Денис	Олегович	KH16-01-10M					1	<input checked="" type="checkbox"/>
Редактировать		9151879	Дубинский	Иван	Борисович	KH15-086					2	<input checked="" type="checkbox"/>
Редактировать		174-94072	Марков	Кирилл	Робертович	KH14-126					3	<input checked="" type="checkbox"/>
Редактировать		30903077	Лисицкий	Дмитрий	Владимирович	KH11-01					6	<input checked="" type="checkbox"/>
Редактировать		30903186	Падусова	Владислав	Олегович	KH11-01					6	<input checked="" type="checkbox"/>
Редактировать		30904699	Лозова	Вероника	Демисович	KH12-01					5	<input checked="" type="checkbox"/>
Редактировать		31030037	Аввакумович	Николай	Николаевич	KH14-116					3	<input checked="" type="checkbox"/>
Редактировать		31014933	Стариков	Григорий	Григорьевич	KH13-106					4	<input checked="" type="checkbox"/>
Редактировать		31014958	Маркина	Евгений	Андреевич	KH13-116					4	<input checked="" type="checkbox"/>

Страница 1 из 172 (Всего элементов: 1720) < Предыдущая 1 2 3 4 5 6 7 ... 170 171 172 Следующая >

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

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 [13].

Элемент Web Forms NumericUpDown представляет собой 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 проигрывателях производится с использованием этого элемента. Также многие числовые элементы управления со шкалой перемещения бегунка вверх и вниз используются в приложениях панели управления Windows [14].

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.

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 [15].

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 [16].

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 [16].

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 [16].

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.

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

Система предоставляет актёру сервисы, описанием которых, также занимаются варианты использования. Варианты использования системы управления, прежде всего, основываются на определенных видах действий [16].

Действующее лицо — это внешний источник, то есть оно не является элементом системы. Взаимодействие 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 7).

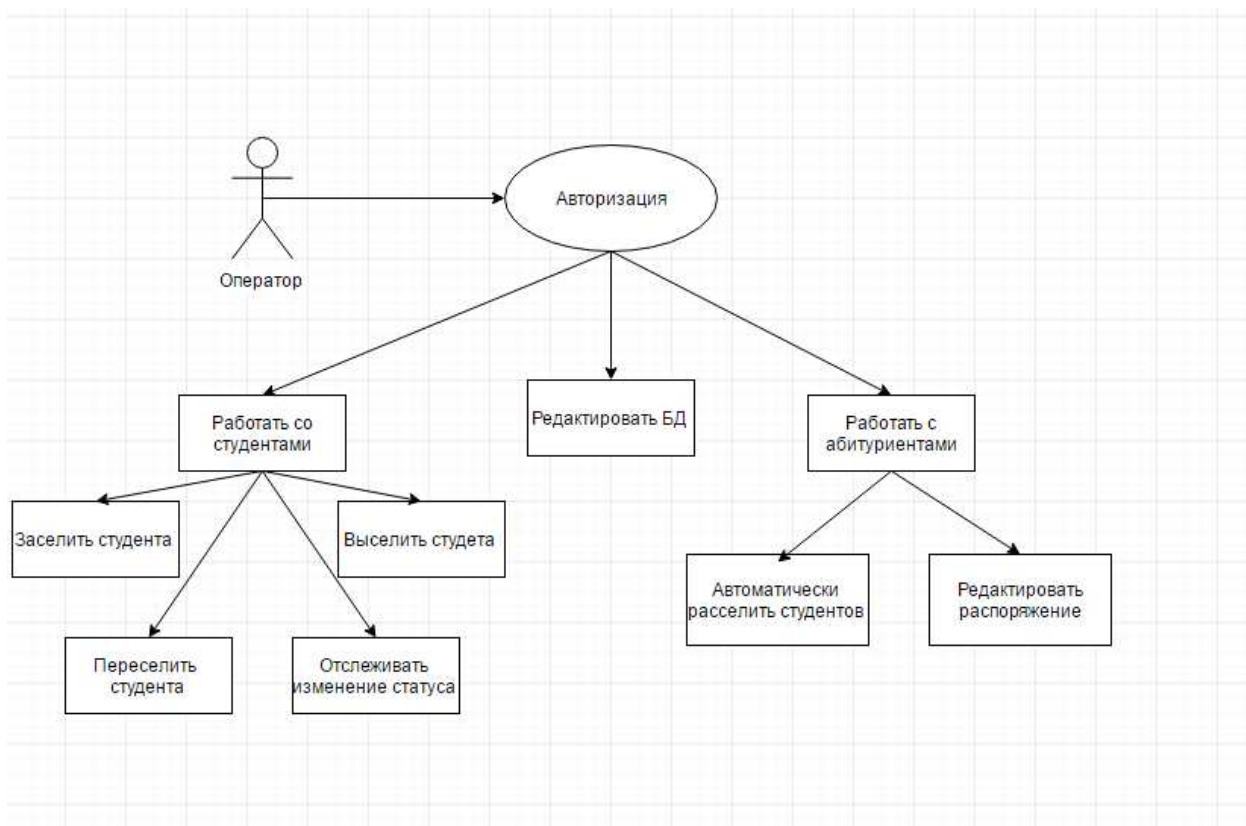


Рисунок 7 - Use case diagram (диаграмма вариантов использования)

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 [17].

Описание потоков работ (Work Flow Modeling). Основой Стандарта IDEF3 является описание действующих процессов, и он подобен алгоритмическим методам создания блок-схем [18].

Описание потоков данных (Data Flow Modeling). Существует нотация DFD (Data Flow Diagramming), отображающая очередность работ,

выполненных в ходе данного процесса, и объем информации, циркулирующий между текущими работами [18].

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 [18].

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 [18].

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 8).

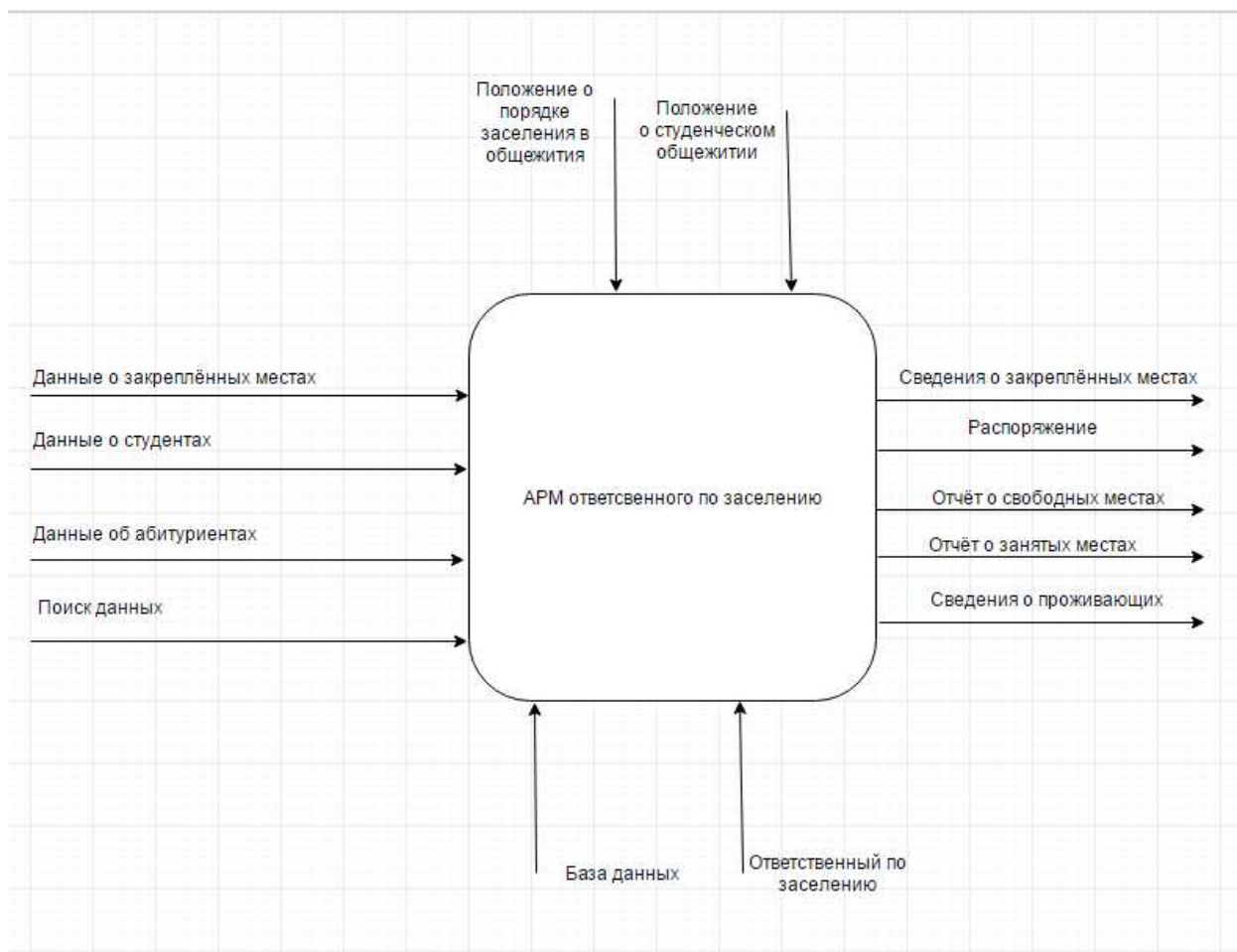


Рисунок 8 – Контекстная диаграмма

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

Данная контекстная диаграмма демонстрирует главный процесс «Заселения в общежитие», обозначенный в блоке прямоугольного вида. Входной информацией для данного процесса являются: «Данные о закреплённых местах», «Данные о студентах», «Данные об абитуриентах», и «Поиск данных», вводимые в систему управляющим, полученный результат работы процесса - выходная информация: «Сведения о закреплённых местах», «Сведения о проживающих», «Отчёт о свободных местах», «Отчёт о занятых местах», «Распоряжение». Также были выделены направления «Положение о порядке заселения в общежития» и «Положение о студенческом общежитии», на базе которых осуществляются все выходные

процессы, и механизмы «Ответственный по заселению», занимающийся работой по заселению с помощью базы данных.

Позже, как произошло описание контекстной диаграммы, осуществляется функциональная декомпозиция. Она предполагает деление на подсистемы. Далее 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 9).

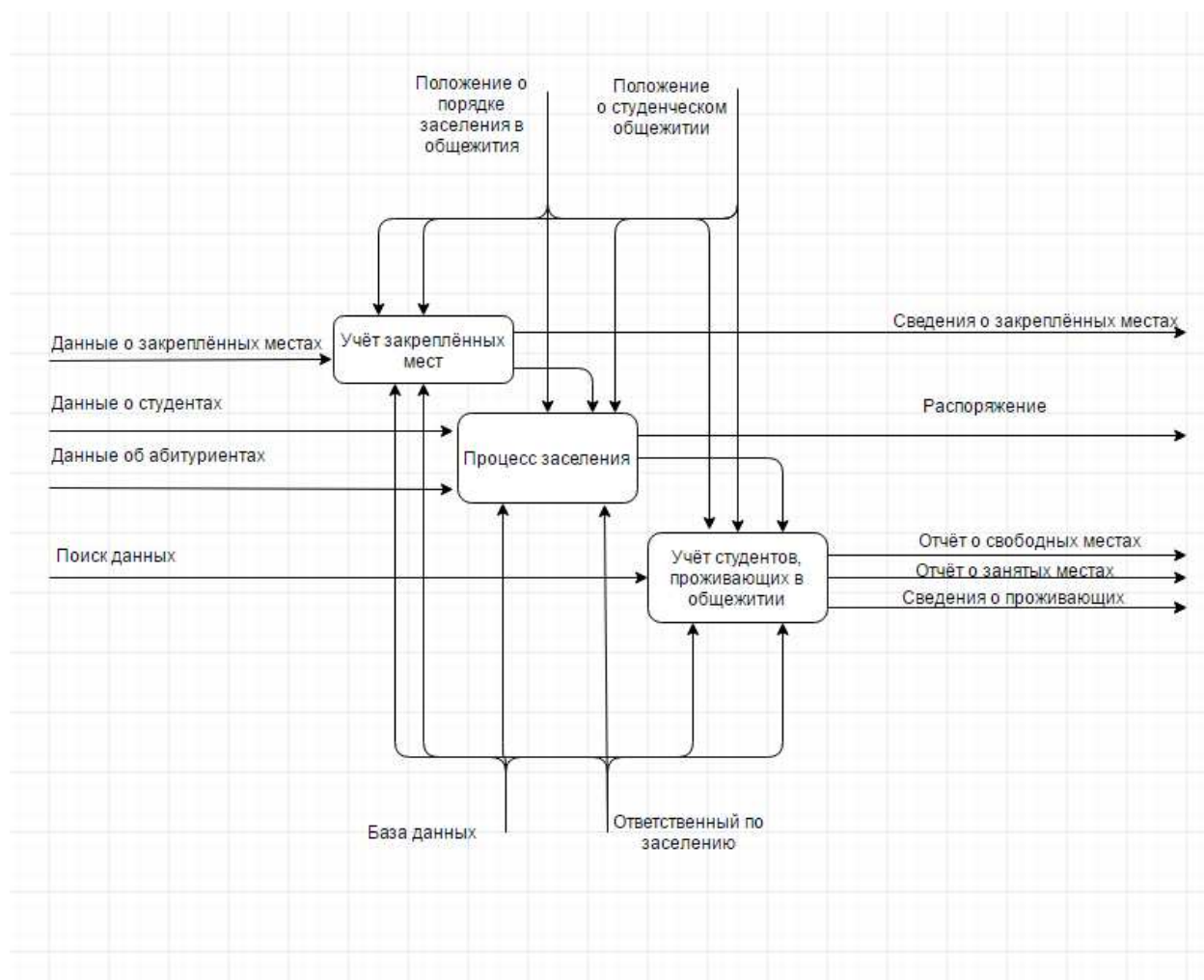


Рисунок 9 – Диаграмма декомпозиции

2.4 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 [19].

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 обо всех студентах, абитуриентах, койко-местах, пользователей, неподписанных распоряжениях, действий пользователей и правах.

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

Первая таблица — это таблица, содержащая всех студентов «Student». В ней, в обязательном порядке, должна быть такая информация, как:

- номер зачётной книжки(id студента);

- фамилия;
- имя;
- отчество;
- учебная группа;
- курс;
- сведения о платном обучении.

Вся информация загружаться в базу с другого источника (АИС web-деканат) или вводится вручную.

Вторая таблица — это таблица комнат — «EnrolleAIS». В ней будут содержаться информация об абитуриентах для их дальнейшего заселения. В этой таблице содержатся следующая информация:

- id абитуриента;
- фамилия;
- имя;
- отчество;
- балл ЕГЭ (Информатика);
- балл ЕГЭ (Математика);
- балл ЕГЭ (Русский язык);
- общий балл ЕГЭ;
- адрес прописки;
- адрес фактический;
- номер телефона;
- подан ли оригинал документов;
- согласие на платное обучение.

Вся информация загружаться в базу с другого источника (АИС web-деканат) или вводится вручную.

Учёт койко-мест будет вестись через другую таблицу, таблицу «Bed», которая содержит информацию:

- id койко-места;

- id общежития;
- id комнаты;
- количество мест в комнате;
- занята ли комната;
- id студента;
- id распоряжения;
- дата заселения;
- пол;
- Дата выселения.

По техническому заданию, необходимо хранить информацию из распоряжения во временной таблице, таблице «TempInstructions», в которой содержатся следующие данные:

- Id студента;
- Id распоряжения;
- Тип распоряжения;
- Дата конца действия распоряжения;
- Койко-место (от куда переселяется студент);
- Койко-место (куда переселяется студент).

Таблица «User», то есть пятая таблица должна содержать данные о пользователях системой и содержит в себе:

- Id пользователя;
- Фамилию;
- Имя;
- Id должности;
- Логин
- Пароль.

Пятая таблица «Login». Она содержит информацию о деятельности пользователя и в ней содержится:

- Id сеанса;

- Время сеанса;
- Id пользователя.

Данные в таблицу вносятся автоматически, во время авторизации в АРМ.

В последней таблице «Rights» содержатся права пользователя, она содержит всего 2 столбца:

- Id должности;
- Заметка.

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

Восемь видов использованных операций:

- Поиск — find ();
- Добавить — add ();
- Заселить — to settle ();
- Удалить — delete ();
- Сохранить — save ();
- Изменить — amend ();
- Переселить — to resettle ();
- Выселить — to evict ().

Каждая операция будет поставлена в соответствие с заданной функциональностью таблицы.

В конечном счете, результат можно представить в виде диаграммы базы данных (рисунок 10).

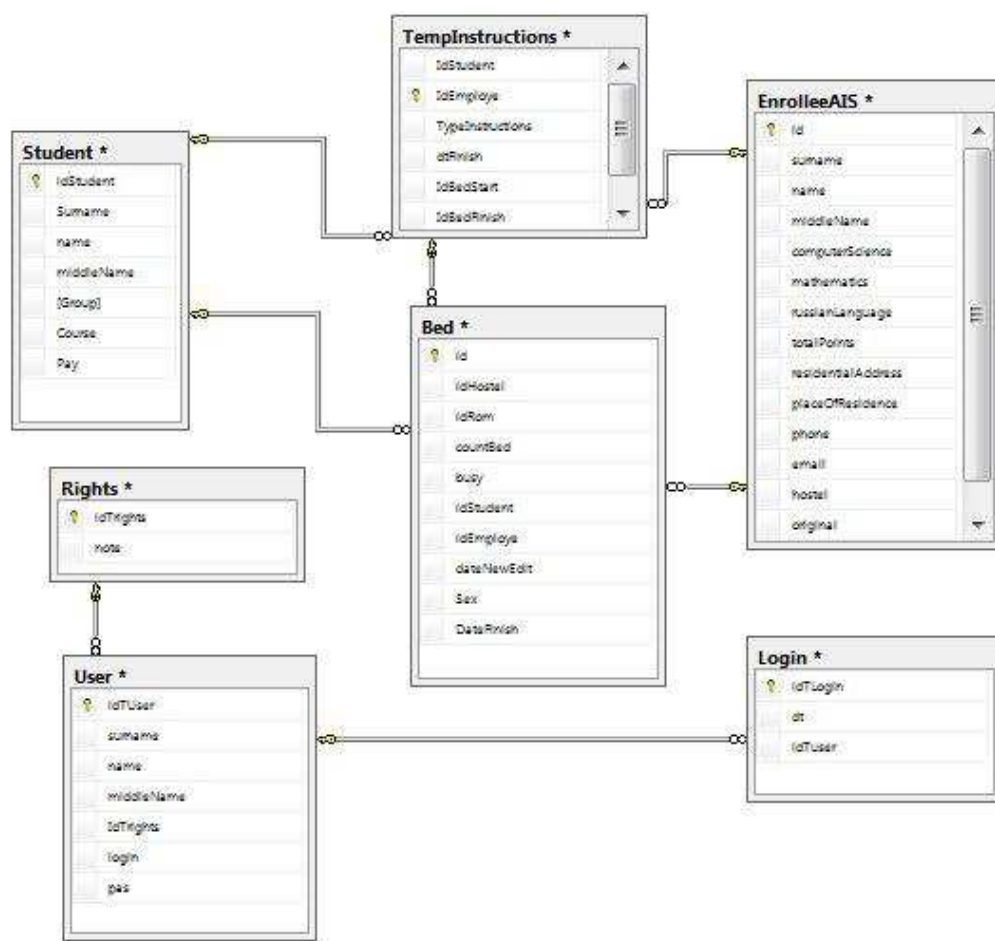


Рисунок 10 – Диаграмма базы данных

2.5 Вывод по главе 2

В данной главе был рассмотрен процесс проектирования автоматизированного рабочего места ответственного по заселению в общежитие.

На первом этапе были определены основные требования к проекту.

Следующим этапом стала разработка структуры web-приложения, через которую сотрудник сможет 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 бала создана диаграмма базы данных, определяющая ключевые атрибуты и операции, используемые в системе.

3 Описание и внедрение АРМ

3.1 Сценарий работы пользователя с приложением

При входе в программу пользователь видит окно авторизации (рисунок 11). Для перехода далее нужно ввести свои данные.

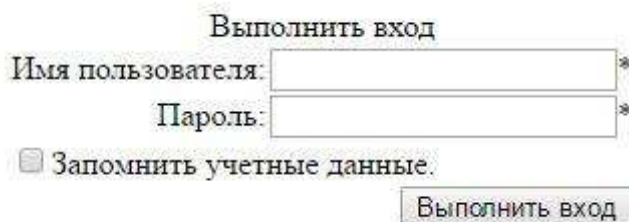


Рисунок 11 - Окно авторизации

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 12).

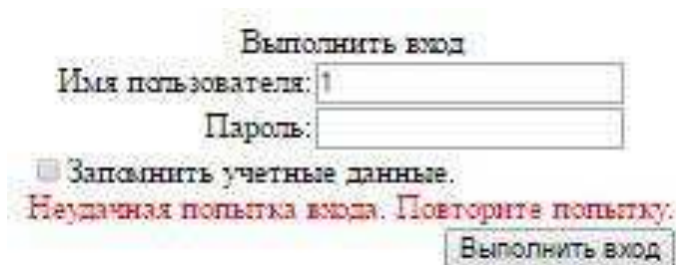


Рисунок 12 – Оповещение об ошибке

После ввода правильных данных при авторизации приложение пускает пользователя в главное меню (рисунок 13).

Распоряжение

Абитуриенты

Студенты

Отчёты

Свободные места

Занятые места

Все места закрепленные за ИКИТ СФУ

Загрузка массивов

Абитуриенты

Студенты

Рисунок 13 – Окно главного меню

Далее пользователю открываются все функции данного приложения. Первым разделом является составление распоряжения (рисунок 14).

№

Фамилия

Имя

Отчество

Группа

Курс

Платник

Общ.Ком.

Срок проживания

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Нет данных для отображения

Заселить

№

Имя

Фамилия

Отчество

Группа

Курс

Тип обучения

Общ./комната

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Нет данных для отображения

Выселить

№

Имя

Фамилия

Отчество

Группа

Курс

Тип обучения

От кур. (общ./ком.)

Кур. (общ./ком.)

Срок окончания

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Нет данных для отображения

Переселить

№

Зачета

Фамилия

Имя

Отчество

Группа

Курс

Платник

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор	61403	Сергеев	Максим	Александрович	КИП-165	3	
Выбор	120326	Долгов	Денис	Олегович	КИП-01-104	1	
Выбор	9151879	Дубинский	Иван	Борисович	КИП-085	2	
Выбор	1714072	Нарков	Кирилл	Робертович	КИП-125	3	
Выбор	30903077	Акимович	Дмитрий	Владимирович	КИП-01	6	
Выбор	30902185	Падеснов	Владислав	Олегович	КИП-01	6	
Выбор	30904699	Леонова	Вероника	Денисовна	КИП-01	5	
Выбор	31010037	Акимович	Никол	Михайлович	КИП-115	3	
Выбор	31014453	Степанов	Геннадий	Геннадьевич	КИП-105	4	
Выбор	31014458	Мартынов	Евгений	Андреевич	КИП-115	4	

Страница 1 из 172 (Всего элементов: 1720) < Предыдущая 2 3 4 5 6 7 ... 170 171 172 Следующая >

№

Фамилия

Имя

Отчество

Группа

Курс

Платник

Общ.Ком.

Комната

Кол-во мест

Сотрудник

Дата заселения

Пол

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор	Назундиков	Кирилл	Игоревич	КИП-095	1		22	1002	5	1	30.03.2017	
Выбор	Климов	Дмитрий	Александрович	КИП-135	3		8	519	3	1	30.03.2017	
Выбор	Худяков	Антон	Николаевич	КИП-115	4		20	809	1	1	30.03.2017	
Выбор	Сонина	Анастасия	Сергеевна	КИП-115	3		8	315	3	1	30.03.2017	

Страница 1 из 172 (Всего элементов: 1720) < Предыдущая 2 3 4 5 6 7 ... 170 171 172 Следующая >

№

Фамилия

Имя

Отчество

Группа

Курс

Платник

Общ.Ком.

Комната

Кол-во мест

Сотрудник

Дата заселения

Дата выезда

Пол

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор

Выбор	Назундиков	Кирилл	Игоревич	КИП-095	1		22	1002	5	1	30.03.2017	22.03.2017	
Выбор	Климов	Дмитрий	Александрович	КИП-135	3		8	519	3	1	30.03.2017	22.03.2017	
Выбор	Худяков	Антон	Николаевич	КИП-115	4		20	809	1	1	30.03.2017	01.04.2017	
Выбор	Сонина	Анастасия	Сергеевна	КИП-115	3		8	315	3	1	30.03.2017	04.04.2017	

Страница 1 из 172 (Всего элементов: 1720) < Предыдущая 2 3 4 5 6 7 ... 170 171 172 Следующая >

Распечатать распоряжение

Подписать распоряжение

Рисунок 14 – Окно составления распоряжения

В разделе составления распоряжения пять основных кнопок. Первая отвечает за заселение студента, вторая за выселение студента, третья за переселение студента, четвёртая за распечатку распоряжения, а пятая за внесение изменений в базу после подписание распоряжения.

37

Также в окне содержится удобное средство по созданию распоряжения, путём поиска нужного студента, в контролерах подключенным к разным таблицам.

Первый контролер связан с таблицей студенты. Оператор в окне справа ищет студента в БД, путем частичного поиска, добавляет его в кэш память для составления распоряжения(рисунок 15).

#	Зачетка	Фамилия	Имя	Отчество	Группа	Курс	Платник
		Рус					
Выбрать	31311112	Русак	Илья	Андреевич	КИ13-135	4	<input type="checkbox"/>
Выбрать	31403334	Русин	Михаил	Сергеевич	КИ14-145	3	<input type="checkbox"/>
Выбрать	31620308	Русинов	Алексей	Эдуардович	КИ16-045	1	<input type="checkbox"/>
Выбрать	31620309	Руссоина	Алена	Алексеевна	КИ16-115	1	<input type="checkbox"/>

Рисунок 15 – Поиск по части данных.

После нажатия на кнопку заселить запись переносится на поле слева (рисунок 16)

#	Фамилия	Имя	Отчество	Группа	Курс	Платник	Общ./Ком.	Срок проживания
Изменить Удалить	Русак	Илья	Андреевич	КИ13-135	4	<input type="checkbox"/>		
Изменить Удалить	Белозерский	Николай	Игоревич	КИ13-135	4	<input type="checkbox"/>		
Изменить Удалить	Бирюков	Денис	Дмитриевич	КИ13-135	4	<input type="checkbox"/>		

Рисунок 16 –Записи, готовые к заселению

Дальше оператору предоставляется возможность выбрать ещё студентов для заселения (рисунок 17).

#	Фамилия	Имя	Отчество	Группа	Курс	Платник	Общ./Ком.	Срок проживания
Изменить Удалить	Русак	Илья	Андреевич	КИ13-135	4	<input type="checkbox"/>		
Изменить Удалить	Белозерский	Николай	Игоревич	КИ13-135	4	<input type="checkbox"/>		
Изменить Удалить	Бирюков	Денис	Дмитриевич	КИ13-135	4	<input type="checkbox"/>		

Рисунок 17 –Выбранные студенты для заселения

При нажатии на ссылку изменить, окно предложит оператору выбрать общежитие, комнату и срок проживания для студента (рисунок 18).

#	Фамилия	Имя	Отчество	Группа	Курс	Платник	Общ./Ком.	Срок проживания
	Русак	Илья	Андреевич	КИ13-135	4			

Общ./Ком.: 22 общ. 707 ком. 2 хол. 0 зан. пол

Срок проживания: 30.05.2017

[Заселить](#) [Сохранить](#) [Отмена](#)

Изменить	Удалить	Белозерский	Николай	Игоревич	КИ13-135	4		
Изменить	Удалить	Бирюков	Денис	Дмитриевич	КИ13-135	4		

Рисунок 18 – Назначение комнаты и срока проживания

При указании срока проживания появляется календарь, который делает эту опцию немного приятней (рисунок 19)

#	Фамилия	Имя	Отчество	Группа	Курс	Платник	Общ./Ком.	Срок проживания
Изменить	Удалить	Русак	Илья	Андреевич	КИ13-135	4	22 общ. 707 ком. 2 хол. 0 зан. пол	30.05.2017
		Белозерский	Николай	Игоревич	КИ13-135	4		

Общ./Ком.: 22 общ. 707 ком. 2 хол. 0 зан. пол

Срок проживания:

Май 2017

Пн	Вт	Ср	Чт	Пт	Сб	Вс
18	1	2	3	4	5	6
19	8	9	10	11	12	13
20	15	16	17	18	19	20
21	22	23	24	25	26	27
22	29	30	31	1	2	3
23	5	6	7	8	9	10

[Сегодня](#) [Очистить](#)

[Изменить](#) [Отмена](#)

Рисунок 19 –Указание срока проживания

После назначения комнаты и срока проживания (рисунок 20) оператор может составить распоряжение на заселение или продолжить работу с другими модулями.

#	Фамилия	Имя	Отчество	Группа	Курс	Платник	Общ./ком.	Срок проживания
ВЫСЕЛИТЬ	Васильев	Иван	Александрович	КИ13-135	4	<input type="checkbox"/>	22 общ. 707 ком. 2 хол. 0 зан. хол.	30.06.2017
ВЫСЕЛИТЬ	Белозеров	Николай	Игоревич	КИ13-135	4	<input type="checkbox"/>	22 общ. 707 ком. 2 хол. 0 зан. хол.	30.06.2017
ВЫСЕЛИТЬ	Берков	Денис	Дмитриевич	КИ13-135	4	<input type="checkbox"/>	15 общ. 707 ком. 2 хол. 0 зан. хол.	30.06.2017

Рисунок 20 – Готовый блок «заселить»

Модуль, посередине – это модуль «Выселить», он связан с таблицей «Bed». Студент также как и в модуле выше может быть выбран путём частичного поиска (рисунок 21).

#	Фамилия	Имя	Отчество	Группа	Курс	Платник	Общедомовое	Комната	Кол-во мест	Сотрудник	Дата заселения	Пол
ВЫСЕЛИТЬ	ВЕР					<input type="checkbox"/>						
ВЫБРАТЬ	Веротин	Виталий	Александрович	КИ15-145	2	<input type="checkbox"/>	22	714	5	1	30.05.2017	М

Рисунок 21 – Выбор студента в модуле выселить

По аналогии с предыдущим модулем сотруднику надо нажать на кнопку «Выселить», что бы студент попал в поле для распоряжения (рисунок 22)

#	Имя	Фамилия	Отчество	Группа	Курс	Тип обучения	Общ / комната
УДАЛИТЬ	Веротин	Виталий	Александрович	КИ15-145	2	<input type="checkbox"/>	227

Рисунок 22 – Модуль выселить для составления распоряжения

После выбора студентов для заселения/переселения, оператор АРМ может составить распоряжение о заселении/переселении или продолжить работу в 3 модуле.

Третий модуль называется «Переселить» (рисунок 23)

#	Имя	Фамилия	Отчество	Группа	Курс	Тип обучения	От куда (общ./ком.)	Куда (общ./ком.)	Срок заселения
ПЕРЕСЕЛИТЬ						<input type="checkbox"/>			
ВЫБРАТЬ	Наумов	Карел	Игоревич	КИ15-095	1	<input type="checkbox"/>	15 707	2	1 30.05.2017 30.06.2017
ВЫБРАТЬ	Климов	Дмитрий	Александрович	КИ14-135	3	<input type="checkbox"/>	16 705	2	1 30.05.2017 30.06.2017
ВЫБРАТЬ	Худяков	Антон	Николаевич	КИ13-115	4	<input type="checkbox"/>	30 609	1	1 30.03.2017 01.04.2017
ВЫБРАТЬ	Савина	Анастасия	Сергеевна	КИ14-115	3	<input type="checkbox"/>	8 315	3	1 30.03.2017 04.04.2017
ВЫБРАТЬ	Русаков	Илья	Александрович	КИ13-135	4	<input type="checkbox"/>	22 707	2	1 30.05.2017 30.06.2017

Рисунок 23 – Модуль «переселить»

Объекты выбираются также как и в первых двух модулях (рисунок 24)

Переселить

#	Фамилия	Имя	Отчество	Группа	Курс	Платник	Общжитие	Комната	Кол-во мест	Сотрудник	Дата заселения	Date Finish	Пол
Выбрать	Климов	Дмитрий	Александрович	КИ14-135	3	<input type="checkbox"/>	8	519	3	1	30.03.2017	22.03.2017	<input type="checkbox"/>
Выбрать	Худоголов	Антон	Николаевич	КИ13-115	4	<input type="checkbox"/>	20	809	1	1	30.03.2017	01.04.2017	<input type="checkbox"/>
Выбрать	Сомова	Анастасия	Сергеевна	КИ14-115	3	<input type="checkbox"/>	8	315	3	1	30.03.2017	04.04.2017	<input type="checkbox"/>

Рисунок 24 – Выбор объектов в модуле «Переселить»

После нажатия кнопки «переселить» объекты попадают в левую таблицу для составления распоряжения (рисунок 25)

#	Имя	Фамилия	Отчество	Группа	Курс	Тип обучения	От куда (общ./ком.)	Куда (общ./ком.)	Срок окончания
	Назмутдинов	Кирилл	Игоревич	КИ16-095	1	<input type="checkbox"/>	22 общ. 1002 ком. 5 кол. 1 зан. пол		
							Куда (общ./ком.): 16 общ. 706 ком. 2 кол. 0 зан. пол	Срок окончания: 30.06.2017	
<div> <div>Изм.</div> <div>Удалить</div> <div>Климов</div> <div>Дмитрий</div> <div>Александрович</div> <div>КИ14-135</div> <div>3</div> <div><input type="checkbox"/></div> </div>									

Переселить

Распечатать распоряжение

Подписать распоряжение

Изм.

Удалить

Климов

Дмитрий

Александрович

КИ14-135

3

☐

30.06.2017

Июнь 2017

Пн Вт Ср Чт Пт Сб Вс

22 29 30 31 1 2 3 4

25 5 6 7 8 9 10 11

26 12 13 14 15 16 17 18

23 19 20 21 22 23 24 25

26 26 27 28 29 30 1 2

27 3 4 5 6 7 8 9

Сегодня Очистить

Рисунок 25 – Выбор комнаты и срока проживания в модуле переселить

Таблицу, сформированную для распоряжения, можно увидеть на рисунке 26

#	Фамилия	Имя	Отчество	Группа	Курс	Платник	Общ./Ком.	Срок проживания
Изменить Удалить	Русах	Илья	Андреевич	КИ13-13Б	4	<input checked="" type="checkbox"/>	22 общ. 707 ком. 2 кол. 0 зан. пол.	30.06.2017
Изменить Удалить	Белозерский	Николай	Игоревич	КИ13-13Б	4	<input checked="" type="checkbox"/>	22 общ. 707 ком. 2 кол. 0 зан. пол.	30.06.2017
Изменить Удалить	Бирюков	Денис	Дмитриевич	КИ13-13Б	4	<input checked="" type="checkbox"/>	16 общ. 707 ком. 2 кол. 0 зан. пол.	30.06.2017

Заселить

#	Имя	Фамилия	Отчество	Группа	Курс	Тип обучения	Общ./комната
Удалить	Верютин	Виталий	Александрович	КИ15-14Б	2	<input checked="" type="checkbox"/>	227

Выселить

#	Имя	Фамилия	Отчество	Группа	Курс	Тип обучения	От куда (общ./ком.)	Куда (общ./ком.)	Срок окончания
Изменить Удалить	Назмутдинов	Кирилл	Игоревич	КИ16-09Б	1	<input checked="" type="checkbox"/>	22 общ. 1002 ком. 5 кол. 1 зан. пол.	16 общ. 707 ком. 2 кол. 0 зан. пол.	30.06.2017
Изменить Удалить	Климов	Дмитрий	Александрович	КИ14-13Б	3	<input checked="" type="checkbox"/>	8 общ. 519 ком. 3 кол. 1 зан. пол.	16 общ. 705 ком. 2 кол. 0 зан. пол.	30.06.2017

Переселить

Рисунок 26 – Таблица для распоряжения

Следующим шагом будет формирование распоряжения на заселение/выселение/переселение, это производится нажатием кнопки «Распечатать распоряжение», пример распоряжение находится на рисунке 27.

Министерство образования и науки РФ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ КОСМИЧЕСКИХ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

РАСПОРЯЖЕНИЕ

О выселении/заселении/переселении

от _____ № _____

Заселить с 30.5.2017г. следующих студентов:

№	ФИО			Институт	Группа	Общ.	Ком.	Форма обучения	Период заселения
1	Русах	Илья	Андреевич	ИКИТ	КИ13-13Б	22	707	Очное/бюджетное	2017-06-30
2	Белозерский	Николай	Игоревич	ИКИТ	КИ13-13Б	22	707	Очное/бюджетное	2017-06-30
3	Бирюков	Денис	Дмитриевич	ИКИТ	КИ13-13Б	16	707	Очное/бюджетное	2017-06-30

Выселить с 30.5.2017г. следующих студентов:

№	ФИО			Институт	Группа	Общ.	Ком.	Форма обучения	Период заселения
4	Верютин	Виталий	Александрович	ИКИТ	КИ15-14Б	22	714	Очное/бюджетное	2017-05-30

Переселить с 30.5.2017г. следующих студентов:

№	ФИО			Институт	Группа	Общ.	Ком.	Форма обучения	Период заселения
5	Назмутдинов	Кирилл	Игоревич	ИКИТ	КИ16-09Б	22/16	1002/707	Очное/бюджетное	30.5.2017г.
6	Климов	Дмитрий	Александрович	ИКИТ	КИ14-13Б	8/16	519/705	Очное/бюджетное	30.5.2017г.

Директор ИКИТ

Г.М. Цибульский

Согласованно:

Начальник управления общежитиями

А.В. Реводько

Рисунок 27 – Распоряжение о выселении/заселении/переселении

Далее это распоряжение отправляется на подпись ответственного по заселению и начальника управления общежитий. После утверждения этого распоряжения необходимо нажать кнопку подписать распоряжение, только после этого изменения внесутся в БД.

3.2 Информационная безопасность АРМ

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

Доступ к данным можно организовать при помощи логина и пароля.

.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 Оценка качества ИС

С точки зрения ИСО/МЭК 9126 [20], 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.

3.4 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.

3.5 Результат внедрения

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.

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 .pdf.

Таблица 1 – Результаты внедрения АРМ

Род работы	Ресурс	До внедрения	После внедрения
Абитуриенты	Персонал	4-7 человек	1 человек
	Время	72 - 120 часов	3 часа
Студенты	Время	60 минут	10 минут
Отслеживание закреплённых мест	Время	120 минут	5 минут

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. Также в главе был проведен анализ качества информационной системы, у большинства критериев качества преобладают высокие показатели. В следующем разделе – разделе «Внедрение» были представлены этапы внедрения системы в организацию. В разделе «Результат внедрения» была произведена сравнительная характеристика современных информационных систем с созданной.

ЗАКЛЮЧЕНИЕ

Результатом выполненной работы является web-приложение, предназначенное для управления и контролирования процесса расселения. В нём содержатся функции составления документов. А также реализован процесс автоматического редактирования базы данных, после подписания распоряжения.

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

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 ответственному по заселению в общежития.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ:

- 1 Венделева, М. А. Информационные технологии управления. / М. А. Венделева, Ю. В. Вертакова. – Москва: Юрайт, 2011. 34с.
- 2 Государство. Бизнес. ИТ [Электронный ресурс]: Microsoft Visual Studio. // «Тадвизер». - 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
- 3 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>
- 4 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)
- 5 Создаём и раскручиваем сайт самостоятельно [Электронный ресурс]: Урок 3. Реляционные базы данных. // «СайтДу». - Режим доступа: <http://www.site-do.ru/db/db3.php>
- 6 Бен-Ган, И. Microsoft SQL Server 2012. Основы T-SQL / И. Бен-Ган. – Москва : Эксмо, 2015. – 33с.
- 7 Ресурс для пользователей продукцией Microsoft. [Электронный ресурс]. ADO.NET. // «МсднМайкрософт». - Режим доступа: [https://msdn.microsoft.com/ru-ru/library/e80y5yhx\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/e80y5yhx(v=vs.110).aspx)
- 8 Вся помощь студенту и школьнику. [Электронный ресурс]: Понятие и виды интерфейсов. // «Студалл». - Режим доступа: <http://studall.org/all-95003.html>
- 9 Ресурс для IT-специалистов [Электронный ресурс]: ASP.NET MVC vs. WebForms// «Хабрахабр». - Режим доступа: <https://habrahabr.ru/post/47249/>

10 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)

11 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)

12 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)

13 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)

14 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)

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

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

17 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/>

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

19 Информационный ресурс [Электронный ресурс]: Этапы проектирования БД. Цели и виды работ на этапах концептуального, логического и физического проектирования. // «СтудопелиЯ». - Режим доступа: <http://studopedia.org/8-228774.html>

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

21 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
Информационные Системы

УТВЕРЖДАЮ
Зав. кафедрой ИС
С. А. Виденин
подпись инициалы, фамилия
« 13 » 06 2017 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 Информационные системы и технологии

АРМ ответственного по заселению в общежития

Руководитель	<u>Шмагрис</u> 13.06.17 подпись, дата	<u>Ю. В. Шмагрис</u> инициалы, фамилия
Выпускник	<u>Русаков</u> 13.06.17 подпись, дата	<u>И. А. Русак</u> инициалы, фамилия
Консультант	<u>Легалов</u> 13.06.17 подпись, дата	<u>И. А. Легалов</u> инициалы, фамилия
Нормоконтролер	<u>Шмагрис</u> 13.06.17 подпись, дата	<u>Ю. В. Шмагрис</u> инициалы, фамилия

Красноярск 2017