Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

Рязанский государственный радиотехнический университет  
имени В.Ф. Уткина

Кафедра ЭВМ

К защите

Руководитель работы:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ**

по дисциплине

**«Клиент серверные приложения баз данных»**

Тема:

«Разработка информационной системы общежития»

Выполнил студент группы 145

Жупин С.Ю.

дата сдачи на проверку, подпись

Руководитель работы

ассистент каф. ЭВМ

Баранова С.Н.

оценка дата защиты, подпись

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc167057093)

[1 Выявление задач автоматизации 4](#_Toc167057094)

[1.1 Описание автоматизированной области 4](#_Toc167057095)

[1.2 Обоснование актуальности разработки 5](#_Toc167057096)

[1.3 Постановка задачи 5](#_Toc167057097)

[1.4 Разработка архитектуры ИС 6](#_Toc167057098)

[2 Разработка серверной части ИС 8](#_Toc167057099)

[2.1 Инфологическое проектирование БД 8](#_Toc167057100)

[2.2 Даталогическое проектирование БД 8](#_Toc167057101)

[2.3 Разработка объектов поддержания целостности данных 8](#_Toc167057102)

[3 Разработка клиентской части ИС 9](#_Toc167057103)

[3.1 Разработка прототипа интерфейса пользователя 9](#_Toc167057104)

[3.2 Реализация интерфейса пользователя 9](#_Toc167057105)

[4 Тестирование основных функций приложения 10](#_Toc167057106)

[ЗАКЛЮЧЕНИЕ 11](#_Toc167057107)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 12](#_Toc167057108)

[ПРИЛОЖЕНИЕ А: сценарий создания объектов БД 13](#_Toc167057109)

[ПРИЛОЖЕНИЕ Б: сценарий заполнения таблиц БД 14](#_Toc167057110)

[ПРИЛОЖЕНИЕ В:исходный код клиентского приложения 15](#_Toc167057111)

# ВВЕДЕНИЕ

На сегодняшний день информационные системы (далее ИС) применяются во множестве различных сферах деятельности человека. ИС позволяют автоматизировать и ускорить процессы обработки, передачи и создания различных данных тем самым позволяя оптимизировать различные прикладные процессы. Благодаря цифровому представлению данных можно избавиться от большого количества бумажных архивов, значительно ускорить передачу информации и упростить обработку информации, а также обеспечить предоставление информации в удобной форме.

В данном курсовом проекте будет создана информационная система общежития, которая будет состоять из сервера состоящем из базы данных MS SQL Server и клиентского приложения на платформе Windows. Данная ИС позволит хранить, добавлять и обрабатывать информацию, требуемую для администрирования общежития в удобном виде, а также позволит делать это удаленно.

# Выявление задач автоматизации

## Описание автоматизированной области

Необходимо спроектировать ИС для студенческого общежития РГРТУ, предоставляющей проживание для студентов как самого РГРТУ, так и других вузов и профессиональных училищ. ИС. ИС разрабатывается для одного общежития, но при необходимости ее можно будет улучшить для администрирования сразу множества общежитий, например, в пределе одного студенческого городка.

Система хранит информацию о проживающих в общежитии студентах с указанием их персональных данных, местом их обучения, а также о тарифе и долгах за проживание.

Система хранит информацию о структуре общежития, такую как список блоков с указанием их типов и назначениях.

Система хранит информацию о всех помещениях общежития с указанием типа комнаты, ее принадлежности к блоку и количеству доступных для заселения мест.

Система хранит информацию о работниках общежития (вахтеры, уборщики, слесари и др.) с указанием их персональных данных и занимаемой должности.

Система хранит информацию о существующих в общежитии должностях. Должность определяет базовый оклад работника.

Ведется отслеживание различного рода инвентаря (мебель, фурнитура и т.д.) и хранение информации о нем, такой как название и описание, стоимость, дата поставки и комната его размещения.

При заселении нового или переселении уже существующего в системе студента администратор общежития вносит соответствующие изменения.

Проживающие в общежитии могут создать заявку с жалобой или предложением, на которую администратором будет назначен ответственный за ее решение из работников общежития.

При устройстве на работу в общежитии администратор вносит нового работника в систему с указанием его персональных данных и должности.

Администратор назначает дежурных по блокам общежития из работников, а также вахтеров на вахты из ранее размеченного расписания вахт.

Администратор составляет расписание вахт в общежитии с указанием даты захода и длительностью смены.

## Обоснование актуальности разработки

При изучении предметной области было замечено что многая информация о проживающих в общежитии хранится в бумажном виде, а также отсутствует доступ о данных студентах из деканатов образовательных организаций, из-за этого возникают ситуации беготни с бумагами. Также хранение информации в бумажном виде сильно ограничивает скорость ее обработки и увеличивает шансы допуска ошибок, которые могут нарушить целостность данных.

Также электронное представление данных значительно оптимизирует работу администратора общежития что позволит повысить производительность труда.

## Постановка задачи

Для данной предметной области основными задачами автоматизации являются:

– Автоматизация операций заселения/переселения проживающих.

– Автоматизация операций с заявками и предложениями.

– Автоматизация инвентаризации.

– Автоматизация назначения работ для работников общежития.

– Удобный и быстрый поиск и отображение информации.

– Удаленный просмотр информации.

## Разработка архитектуры ИС

Для данной ИС была выбрана двухуровневая клиент-серверная архитектура. Она позволяет значительно сэкономить на серверной части, так как в отличии от распределенных систем здесь требуется только один физический сервер.

Из других преимуществ такая архитектура способна поддерживать целостность данных на обоих уровнях и равномерно распределять нагрузку между клиентом и сервером. Также такая архитектура позволяет организовать централизованную защиту данных, т.к. все данные будут храниться на сервере. Клиентское приложение в свою очередь будет отправлять запросы на сервер и выводить пользователю полученную информацию в форматированном виде. Также с клиента можно отправлять параметризированные запросы с различными ограничениями на добавление/изменение/удаление данных, которые позволят исключить нарушение целостности данных и SQL-инъекции.

Централизованное хранилище данных является не только преимуществом, но и недостатком, так как при нарушении работы сервера все экземпляры приложений-клиентов потеряют доступ к данным, а также при несанкционированном доступе будет утечка сразу всех данных.

Схема архитектуры ИС представлена на рисунке 1. Виртуальный сервер БД физически располагается на устройстве-сервере, также сервер БД представляет часть слоя логики и выполняет большинство операций по обработке данных. Клиентское приложение содержит в себе остальную часть слоя логики который формирует запросы к серверу и слой представления данных полученных от сервера.

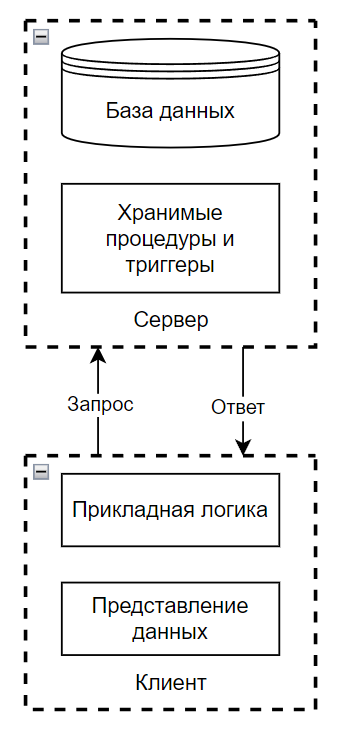


Рисунок 1 – Архитектура ИС

# Разработка серверной части ИС

## Инфологическое проектирование БД

### Требуемая информация

Исходя из предметной области в БД требуется хранить следующую информацию:

1. **Информацию о проживающих:**

– ФИО,

– Серия и номер паспорта,

– Пол (муж/жен),

– Дата рождения,

– Номер телефона для связи,

– Учебное заведение, в котором он обучается,

– Комната проживания,

– Тариф, который зависит от льгот и места обучения.

1. **Информацию о комнате:**

– Тип комнаты,

– Номер блока, в котором она находится,

– Количество койко-мест, которое зависит от типа комнаты.

1. **Информацию о блоках:**

– Тип блока, который определяет какие типы проживающих рекомендуется заселять в блок,

– Принадлежность блока к крылу и этажу,

– Информацию о работнике являющийся дежурным по блоку.

1. **Информацию об инвентаре**

– Наименование и описание,

– Дата поставки,

– Стоимость,

– Номер комнаты размещения.

1. **Информацию о вахтах:**

– Тип вахты,

– Ставка почасового оклада за вахту в зависимости от типа вахты

– Дата и время заступления на вахту,

– Длительность вахты в часах

– Назначенный на вахту работник общежития.

1. **Информацию о работниках общежития:**

– ФИО,

– Серия и номер паспорта,

– Телефон для связи,

– Должность,

– Базовый оклад, зависимый от должности.

1. **Информацию о заявках** –

– Тема для удобного поиска и сортировки,

– Основной текст,

– Статус заявки (создана, закрыта, в процессе),

– Дата создания,

– Составитель из проживающих,

– Ответственный за рассмотрение, назначенный из работников.

### Выделение сущностей

Для хранения вышеуказанной информации необходимо выделить следующие сущности:

* Проживающий (Код проживающего)
* Тип проживающего (Название типа)
* Комната (Номер комнаты)
* Тип комнаты (Название типа)
* Блок (Номер блока)
* Тип блока (Название типа)
* Инвентарь (Наименование)
* Работник (Код работника)
* Должность (Название должности)
* Вахта (Номер вахты)
* Тип вахты (Название типа)
* Заявка (Номер заявки)

### Выделение связей

Проживающий имеет тип проживающего;

Проживающий проживает в комнате;

Проживающий составляет заявки;

Комната имеет тип комнаты;

Комната содержит инвентарь;

Блок содержит комнаты;

Блок имеет тип блока;

Работник дежурит в блоке;

Работник дежурит на вахтах;

Работник обрабатывает заявки;

Работник имеет должность;

Вахта имеет тип вахты;

### Построение ER диаграммы (Рисунки 224-2343)

**Проживающий имеет тип**

Проживающий имеет только один тип

Проживающий обязательно имеет тип

Один тип может иметь множество проживающих

Тип не обязательно должен иметь проживающих

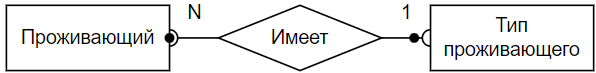


Рисунок 2 – Проживающий имеет тип

**Проживающий проживает в комнате**

Проживающий живет только в одной комнате

Проживающий обязательно должен проживать в комнате

В одной комнате могут жить несколько проживающих

В комнате не обязательно должны проживать

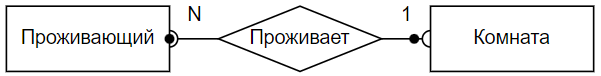


Рисунок 3 – Проживающий проживает в комнате

**Проживающий составляет заявки**

Проживающий может составить множество заявок

Проживающему не обязательно составлять заявки

Одну заявку составляет только один проживающий

Заявка обязательно содержит информацию о составителе

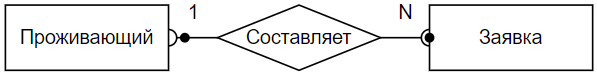


Рисунок 4 – Проживающий составляет заявки

**Комната имеет тип**

Комната обязательно имеет тип

Комната может иметь только один тип

Одному типу может соответствовать множество комнат

Типу не обязательно иметь соответствующую ему комнату

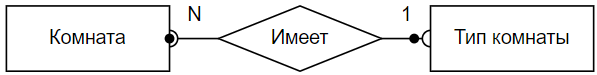


Рисунок 5 – Комната имеет тип

**Комната содержит инвентарь**

В одной комнате может содержаться множество инвентаря

В комнате может не быть инвентаря

Инвентарь обязательно имеет комнату, в которой он размещается

Единица инвентаря размещается только в одной комнате

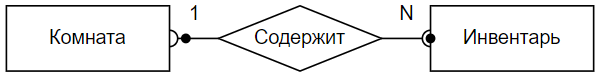


Рисунок 6 – Комната содержит инвентарь

**Блок содержит комнаты**

В одном блоке может быть множество комнат

Комната обязательно должна иметь блок, в котором она находится

Комната может находиться только в одном блоке

Блок необязательно должен иметь комнаты



Рисунок 7 – Блок содержит комнаты

**Блок имеет тип**

Блок обязательно имеет тип

Блок имеет только один тип

Одному типу может соответствовать множество блоков

Типу может не соответствовать ни одного блока



Рисунок 8 – Блок имеет тип

**Работник дежурит в блоке**

В одном блоке может дежурить только один работник

В блоке необязательно должны дежурить

Работник может дежурить в множестве блоках

Работнику необязательно дежурить



Рисунок 9 – Работник дежурит в блоке

**Работник дежурит на вахтах**

Работнику необязательно дежурить на вахте

Работник может дежурить на нескольких вахтах

На одну вахту назначается только один работник

Вахте обязательно должен быть назначен работник



Рисунок 10 – Работник дежурит на вахтах

**Работник обрабатывает заявки**

Работник может обрабатывать множество заявок

Работнику необязательно обрабатывать заявки

Одну заявку обрабатывает только один работник

Заявка может существовать без назначения на нее работника

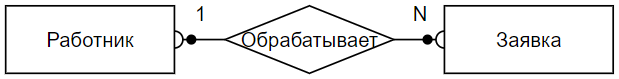


Рисунок 11 – Работник обрабатывает заявки

**Работник имеет должность**

Работник имеет только одну должность

Работник обязательно имеет должность

На одной должности может состоять множество работников

Должности может никто не соответствовать

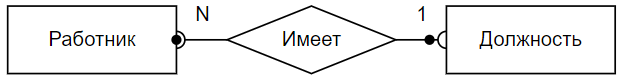


Рисунок 12 – Работник имеет должность

**Вахта имеет тип вахты**

Вахта имеет только один тип вахты

Вахта обязательно имеет тип

Одному типу может соответствовать множество вахт

Тип может не иметь соответствующих ему вахт

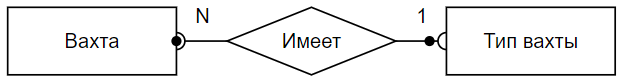


Рисунок 13 – Вахта имеет тип вахты

## Даталогическое проектирование БД

## Разработка объектов поддержания целостности данных

# Разработка клиентской части ИС

## Разработка прототипа интерфейса пользователя

## Реализация интерфейса пользователя

# Тестирование основных функций приложения

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

# ПРИЛОЖЕНИЕ А: сценарий создания объектов БД

# ПРИЛОЖЕНИЕ Б: сценарий заполнения таблиц БД

# ПРИЛОЖЕНИЕ В:исходный код клиентского приложения