

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.03.01 – «Информатика и вычислительная техника»
Направленность (профиль) образовательной программы Автоматизированные
системы обработки информации и управления

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«___» _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка автоматизированной системы для личного кабинета студента Амурского государственного университета «Общежитие»

Исполнитель
студент группы 953об

(подпись, дата)

М.Е. Чирич

Руководитель
доцент, канд. физ.-мат.
наук

(подпись, дата)

В.В. Ерёмина

Консультант:
по части безопасности
и экологичности
доцент, канд. техн. наук

(подпись, дата)

А.Б. Булгаков

Нормоконтроль
инженер кафедры

(подпись, дата)

В.Н. Адаменко

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет Математики и информатики

Кафедра Информационных и управляющих систем

УТВЕРЖДАЮ

Зав. Кафедрой

_____ А.В. Бушманов

« ____ » _____ 2023г

ЗАДАНИЕ

К выпускной квалификационной работе студента: Чирич М.Е.

1. Тема выпускной квалификационной работы: Разработка автоматизированной системы для личного кабинета студента Амурского государственного университета «Общежитие» (Утверждена приказом от 24.04.2023 №974-уч)

2. Срок сдачи студентом законченной работы (проекта): _____

3. Содержание выпускной квалификационной работы: анализ предметной области; анализ программно-технических средств; анализ требований к разработке; разработка алгоритмов решения;

4. Перечень материалов приложения: Техническое задание

5. Дата выдачи задания: 07.09.2023

Руководитель выпускной квалификационной работы: _____

Ерёмина В.В. доцент кафедры ИиУС, канд. физ.-мат. наук

(фамилия, имя, отчество, должность, уч. степень, уч. звание)

Задание принял к исполнению (07.09.2023): _____

(Подпись студента)

РЕФЕРАТ

Дипломная (бакалаврская) работа содержит 107 с., 51 рисунок, 26 таблиц, 1 приложение, 18 источника.

ВЕБ-ПРИЛОЖЕНИЕ, ПРОЕКТИРОВАНИЕ БД, ПРЕДМЕТНАЯ ОБЛАСТЬ, RUBY, RUBY ON RAILS, JAVASCRIPT, MVC, РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Объект исследования – деятельность студенческого городка и сотрудников ФГБОУ ВО «АмГУ» по заселению (размещению) обучающихся в общежития.

Цель работы заключается в разработке модуля «Общежития» позволяющего дистанционно подать заявление на предоставление койко-место для проживания в общежитиях «Амурского государственного университета».

Модуль полностью интегрирован в информационную среду сайта ФГБОУ ВО «АмГУ» в личном кабинете, в разделе «Общежитие». На ряду с этим модуль производит обмен данными с 1С Университет посредством веб-сервисов, разработанных в индивидуальном порядке для решения конкретных целей общей разработки. При разработке используется ряд актуальных технологий веб-программирования – фреймворки Ruby on Rails и Vue.js.

СОДЕРЖАНИЕ

Введение	8
1 Общая характеристика предметной области и исследование внутренних процессов студенческого городка по заселению в общежития фгбоу во «амгу»	10
1.1 Цели и задачи комиссии по заселению	10
1.2 Организационная структура комиссии по заселению	11
1.3 Организационная структура университета	12
1.4 Основные положения регулирующие процессы разрабатываемого модуля	15
1.5 Анализ использования программно-технических средств	18
1.6 Обзор и анализ существующих методов решения рассматриваемой задачи	18
1.7 Формулировка задачи исследования и общей методики ее решения	21
1.8 Обоснование актуальности темы квалификационного исследования	22
2 Проектирование модуля «Общежитие»	24
2.1 Анализ требований к модулю «Общежитие»	24
2.2 Структурно-функциональная модель модуля «Общежитие»	26
2.3 Обоснование используемых обеспечивающих подсистем модуля «Общежитие»	29
2.4 Обоснование выбора средств разработки	31
2.4.1 Выбор языка программирования	31
2.4.2 Выбор среды разработки	33
2.4.3 Выбор методики разработки	34
2.5 Анализ данных	40
2.6 Проектирование структуры модуля и взаимодействующих систем	41
2.7 Разработка структуры пользовательского интерфейса	46
2.8 Анализ системных требований	51
3 Программная реализация	53

3.1 Проектирование базы данных	53
3.1.1 Обоснование выбора СУБД и инструментария разработки базы данных	53
3.1.2 Инфологическое проектирование базы данных	56
3.1.3 Логическое проектирование базы данных	64
3.1.4 Физическое проектирование базы данных	68
3.2 Создание модуля «Общежитие»	73
3.2.1 Создание функциональной части для студентов	73
3.2.2 Создание функциональной части для комиссии	78
4 Безопасность и экологичность	86
4.1 Безопасность	86
4.1.1 Необходимые условия для работы с ПЭВМ в помещении	87
4.1.2 Необходимые условия шума и вибрации в процессе работы	88
4.1.3 Необходимые требования к организации и эргономике рабочего места	88
4.2 Экологичность	90
4.3 Чрезвычайные ситуации	91
Заключение	94
Библиографический список	95
Приложение А	98

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты и нормативные документы:

СП 2.2.3670-20. Санитарно-эпидемиологические требования к условиям труда.

СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания.

ГОСТ Р 50948-2001. Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности.

ГОСТ Р 50949-2001. Средства отображения информации индивидуального пользования. Методы измерений и оценки эргономических параметров и параметров безопасности.

ГОСТ 28406-89. Персональные электронные вычислительные машины. Интерфейсы видеомониторов. Общие требования

ГОСТ Р 51645-2017. Рабочее место для инвалида по зрению типовое специальное компьютерное. Технические требования к оборудованию и производственной среде.

ГОСТ Р ИСО 1503-2014. Эргономика. Требования к пространственной ориентации и направлениям движения органов управления.

ГОСТ 12.1.007-76 Система стандартов безопасности труда (ССБТ). Вредные вещества. Классификация и общие требования безопасности.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ТЗ – Техническое задание

БД – База данных

СФМ – Структурно-функциональная модель

ИС – Информационная система

АСОИУ – Автоматизированная система обработки информации и управления

СУБД – Система управления базами данных

АмГУ – Амурский государственный университет

RoR – Ruby on Rails

JS – JavaScript

API – Application Programming Interface

SQL – Structured Query Language

MVC – Model-View-Controller

VS – Microsoft Visual Studio

SPA – Single Page Application

DOM – Document Object Model

ORM – Object-relational mapping

UML – Unified Modeling Language

ВВЕДЕНИЕ

Современный мир стремительно развивается, и технологии становятся неотъемлемой частью нашей жизни. В этом контексте цифровизация играет важную роль, упрощая жизнь и повышая эффективность действий человека. Важным направлением цифровизации является автоматизация, которая позволяет улучшить качество жизни и работы людей в различных сферах деятельности. Одной из сфер применения является – развитие современных вузов.

В целом автоматизация вуза реализуется на основе концепции единой информационной среды, что под собой подразумевает: однозначное представление информации об объектах предметной области; интеграция в единую информационную модель; комплексное, авторизованное использование информации.

Методом достижения таких свойств организации является использование веб-технологий, которые несут в себе ряд основных преимуществ:

- упрощение трудоёмкого процесса, путем понятной и удобной визуализации информации;
- повышение точности и организованности. Данные собираются и хранятся более организованным и эффективным образом;
- повышенное удобство. Доступ к веб-приложению возможен с любого устройства, подключенного к Интернету.

В выпускной квалификационной работе рассмотрен процесс проектирования и разработки модуля «Общежития» для информационной системы ФБОУ ВО «АмГУ», направленный на проведение процесса подачи заявления студентом на заселение и последующем его одобрением комиссией. Модуль представляет из себя некую подсистему в личном кабинете «АмГУ» студента, расположенную в сети интернет с предоставлением возможности подать заявление на проживание в общежитии и следить за его статусом, а также со стороны комиссии распределять студентов по местам указанным в заявлении.

Цель разработки автоматизированной системы «Общежитие» для

студентов университета состоит в создании эффективного и автоматизированного модуля, который позволит студентам легко и удобно подавать заявления на проживание в общежитии, а проверяющим (членам комиссии) – быстро и эффективно обрабатывать эти заявления, основываясь на академическом рейтинге успеваемости студентов и их социальном статусе. Это позволит уменьшить время, затрачиваемое на обработку заявлений, увеличить точность и объективность процесса, а также улучшить качество принимаемых решений относительно проживания студентов в общежитии. Результатом данной разработки будет повышение эффективности работы общежития и связанных с ним подразделений, таких как студгородок, факультеты и бухгалтерия.

В данном случае, предметом разработки является автоматизированная система «Общежитие» для студентов «Амурского Государственного Университета» и членов комиссии по заселению, а объектом исследования – процесс обработки заявлений на проживание в общежитии.

Основные задачи исследования:

- выполнить анализ предметной области;
 - разработать структуру модуля информационной системы;
 - разработать модель и структуру базы данных, включающую как новые созданные, так и уже используемые таблицы и структуры в целевой системе;
 - разработать клиентскую часть системы – спроектировать интерфейс модуля «Общежитие» и организовать взаимодействия пользователя с интерфейсом для обновления данных используя язык JavaScript;
 - разработать серверную часть сайта с использованием языка Ruby;
- интегрировать разработанный модуль в личный кабинет «АмГУ».

1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДМЕТНОЙ ОБЛАСТИ И ИССЛЕДОВАНИЕ ВНУТРЕННИХ ПРОЦЕССОВ СТУДЕНЧЕСКОГО ГОРОДКА ПО ЗАСЕЛЕНИЮ В ОБЩЕЖИТИЯ ФГБОУ ВО «АМГУ»

1.1 Цели и задачи комиссии по заселению

Комиссия по заселению является организационной структурой, созданной для распределения студентов по их заявлениям на койко-место в общежитии. Основной род деятельности – это планирование, координация и контроль за процедурой заселения, при этом обеспечивает соблюдение соответствующих правил, непосредственно касающихся заселения и проживания студентов в общежитиях вуза.

Основными целями комиссии являются:

- обеспечение оптимального заселения студентов в общежития с учетом доступных ресурсов и потребностей студенческого сообщества;
- создание комфортных и безопасных условий проживания для студентов, способствующих успешной учебной деятельности;
- соблюдение установленных норм и правил при заселении студентов в общежития в соответствии с законодательством и политикой университета.

Задачи, решаемые данной структурой:

- разработка и согласование критериев и принципов приоритетности заселения студентов в общежития, учитывающих социальные, академические и иные факторы;
- планирование и организация процесса заселения студентов, включая подготовку необходимых документов, распределение мест и составление списка проживающих;
- обеспечение своевременного информирования студентов о процедурах и сроках заселения;
- ведение учета и контроль за свободными местами в общежитиях, обновление базы данных студентов, проживающих в общежитиях;
- разрешение спорных ситуаций и конфликтов, возникающих в процессе заселения и проживания студентов в общежитии.

Числе организация культурно-массовых и образовательных мероприятий.

Работа комиссии по заселению студентов в общежития вуза заключается в проверке соблюдения правил подачи и верификации заявления, а также в эффективном распределении мест, обеспечении комфортных и безопасных условий проживания, оказании поддержки студентам и сотрудничестве с другими структурными подразделениями университета.

1.2 Организационная структура комиссии по заселению

Описание объекта исследования включает общую информацию о компании или предприятии, такую как название, основные и дополнительные виды деятельности, правовой статус, позиционирование на рынке, границы исследования, задачи, которые требуют автоматизации.

Организационная структура является документом, который графически отображает состав и иерархию подразделений организации. Она формируется на основе целей деятельности и необходимых для их достижения подразделений, которые выполняют функции, составляющие бизнес-процессы компании.

Для рассмотрения предметной области необходимо рассмотреть организационную структуру комиссии по заселению, которая представлена на рисунке 1.

Основные участники комиссии:

– главный бухгалтер – отвечает за финансовые аспекты заселения студентов, включая учет средств за обслуживание общежитий;

– директор студгородка – ответственен за общее управление и эксплуатацию общежитий, обеспечение их нормального функционирования и ремонтно-строительные работы;

– заместитель декана по воспитательной работе со всех факультетов – представляет интересы студентов и учитывает социальные аспекты при заселении, осуществляет координирование распределения студентов в места общежитий в рамках факультета;

– директор учебно-студенческого бюро (УСБ) – занимается организационными вопросами, связанными с заселением студентов, подготовкой документов и взаимодействием с учебными подразделениями;

– комендант – отвечает за контроль соблюдения правил и режима проживания в общежитии, урегулирование конфликтных ситуаций и поддержание дисциплины.

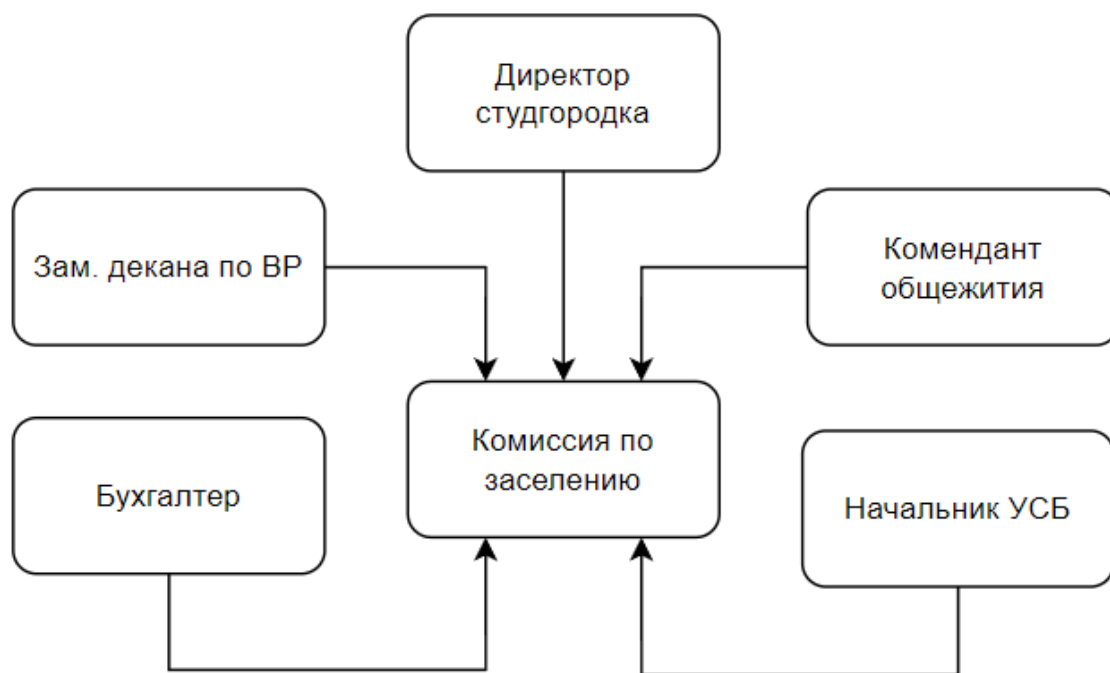


Рисунок 1 – Организационная структура комиссии

Комиссия по заселению в общежитие играет важную роль в процессе принятия и размещения студентов. Ее основная задача заключается в организации и проведении процедуры заселения, обеспечивая справедливость и эффективность этого процесса. Наряду с этим рассматривает различные факторы, такие как доступность мест, приоритеты, установленные регламентом или политикой общежития, и особые обстоятельства, как правило привилегий заявителей.

1.3 Организационная структура университета

Разрабатываемый модуль (система или подсистема) будет частью общей информационной системы (Личного кабинета «АмГУ»), лежащей в основе федерального государственного образовательного учреждения высшего

образования «Амурский государственного университета» (ФГБОУ ВО «АмГУ»). Структура подразделений, связанных с проектируемым модулем, данной учебной организации представлена на рисунке 2.

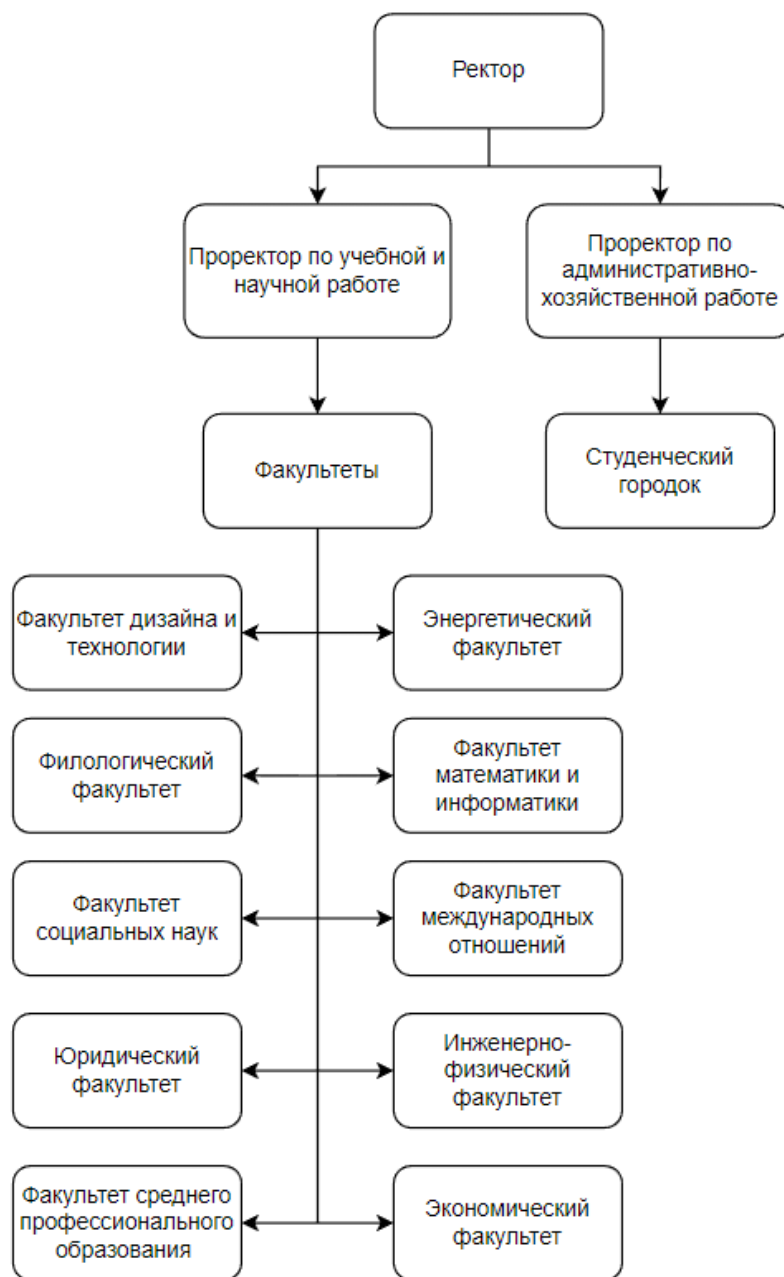


Рисунок 2 – Организационная структура

Руководство высшим учебным заведением осуществляет ректор. Он управляет образовательной и научной деятельностью, формирует программу стратегического развития и координирует административную и финансовую деятельность организации. Основными функциями, выполняющимися ректором, являются:

- планирование стратегического развития;
- разработка и согласование образовательных программ;
- подготовка приказов и расписаний;
- поддержание имиджа вуза;
- организация взаимодействия сотрудников учреждения;
- руководство ученым советом;
- прием и увольнение сотрудников;
- утверждение приказов о приеме и отчислении студентов;
- обеспечение учета, сохранности и пополнения материально-технической базы;
- управление административно-хозяйственной деятельностью высшего учебного заведения.

Подразумевается, что ректор в связи с большим количеством процессов в университете, делегирует часть своих обязанностей на другие функционально-образовательные подразделения. В данном случае рассматриваются проректор по научной и научной работе и проректор по административно-хозяйственной работе.

Проректор по научной работе осуществляет непосредственное руководство научно-исследовательской частью, а также организует, координирует и руководит научной работой в университете. Организует проведение научных исследований в тесной связи с образовательным процессом и отвечает за выполнение и качество научных работ.

Проректор по административно-хозяйственной работе руководит работой отдела, осуществляет контроль над выполнением обязанностей сотрудниками, в пределах установленных прав отвечает за подбор, расстановку и повышение квалификации сотрудников отдела, своевременное и качественное выполнение задач и функций отдела, за соблюдение производственной и трудовой дисциплины. Также в его обязанности входит управление «Студенческим городком».

Студенческий городок «АмГУ» в свою очередь организует

рациональное использование общежитий, поселение и проживание в студенческом городке университета и соблюдение проживающими установленных нормативными документами правил, также создание комфортных жилищно-бытовых условий для проживания.

1.4 Основные положения регулирующие процессы разрабатываемого модуля

Положения о структурных подразделениях образовательной организации является локальными нормативными актами. Положения о структурных подразделениях помогают решать многие вопросы, связанные с организацией потока действующих правил, так как они закрепляют распределение ролей и функций между подразделениями внутри организации.

Положение о жилищно-бытовой комиссии студенческого городка «АмГУ» и порядке ее работы ПКО СМК 22-2013, общее положение которое основывается на жилищное законодательство, Правил внутреннего распорядка в общежитиях «АмГУ», Положения о студенческом совете общежитий, Положения о студенческом городке «АмГУ», а также устанавливает состав, цели, задачи и функции проверяющей Комиссии.

Основной целью Комиссии является:

Контроль выполнения проживающими Правил внутреннего распорядка в общежитиях» АмГУ».

Председателем комиссии является проректор по экономике и перспективному развитию «АмГУ».

В состав членов Комиссии могут входить:

- главный бухгалтер «АмГУ»;
- директор студгородка «АмГУ»;
- заместитель директора студгородка по ВР «АмГУ»;
- зам. декана по ВР юридического факультета «АмГУ»;
- зам декана по ВР филологического факультета «АмГУ»;
- зам декана по ВР факультета дизайна и технологии «АмГУ»;
- зам декана по ВР факультета математике и информатике «АмГУ»;

- зам декана по ВР экономического факультета «АмГУ»;
- зам декана по ВР факультета социальных наук «АмГУ»;
- зам декана по ВР энергетического факультета «АмГУ»;
- зам декана по ВР факультета международных отношений «АмГУ»;
- зам декана по ВР инженерно физического факультета «АмГУ»;
- начальник управления собственной безопасности «АмГУ»;
- начальник отдела международных проектов «АмГУ»;
- начальник юридического отдела «АмГУ»;
- председатель студенческого совета общежитий «АмГУ»;
- начальник СКЦ «АмГУ»;
- председатель объединенного профкома «АмГУ».

Данный список является гибким, в связи с ненужностью участия большого количества ответственных лиц, поэтому его состав при разработке модуля поменяется.

Далее рассматриваются непосредственно правила внутреннего распорядка в общежитиях «АмГУ» ПР СМК 7.1.01-2013.

Сфера применения настоящих правил внутреннего распорядка в общежитиях «АмГУ» применяются во всех общежитиях федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Амурский государственный университет».

В правилах регулируется порядок предоставления мест и выселения из общежития, оплата услуг:

- размещение студентов в общежитиях производится с соблюдением установленных санитарных норм (6 кв.м. на 1 человека) в соответствии с настоящим Положением;
- распределение мест в общежитиях между факультетами и другими структурными подразделениями «АмГУ» осуществляется студгородком. Заселение студентов из числа обучающихся в общежитие должно производиться с учетом их материального положения;
- заселение в общежитие «АмГУ» обеспечивается деканатом

соответствующего факультета в соответствии с представленным в деканат личным заявлением обучающегося;

- деканат соответствующего факультета обеспечивает подготовку приказа ректора «АмГУ» на заселение обучающегося;

- управление собственной безопасности «АмГУ» на период оформления обучающемуся временной регистрации выдает ему временный пропуск в общежитие на срок 10 дней;

- после оформления временной регистрации обучающемуся выдается пропуск в общежитие на время проживания в нем;

- проживающие в общежитиях «АмГУ» заключают договор найма жилого помещения, в котором указывается номер студенческого общежития, секции и комнаты, и представляют в администрацию студгородка справку о состоянии здоровья обучающегося (из здравпункта «АмГУ»).

Кроме того, данные правила являются ознакомительной частью для понимания процесса заселения студентам.

Дополнением к этому рассматривается положение о порядке применения к обучающимся и снятия с обучающихся мер дисциплинарного взыскания, в котором рассматриваются общие положения:

- меры дисциплинарного взыскания применяются за неисполнение или нарушение Устава Университета, правил внутреннего распорядка, правил проживания в общежитиях и иных локальных нормативных актов по вопросам организации и осуществления образовательной деятельности;

- меры дисциплинарного взыскания не применяются к обучающимся с ограниченными возможностями здоровья (с задержкой психического развития и различными формами умственной отсталости);

- не допускается применение мер дисциплинарного взыскания к обучающимся во время их болезни, каникул, академического отпуска, отпуска по беременности и родам или отпуска по уходу за ребенком;

- за каждый дисциплинарный проступок может быть применена одна мера дисциплинарного взыскания.

1.5 Анализ использования программно-технических средств

В университете «АмГУ» используется информационная система 1С, называемая «1С Университет». Эта система предоставляет комплексное решение для автоматизации учета, хранения, обработки и анализа информации о различных процессах вуза. Она включает в себя функциональность для учета поступления абитуриентов, обучения студентов, оплаты за обучение, выпуска и трудоустройства выпускников, а также для расчета и распределения нагрузки преподавателей и других важных процессов.

Также в составе информационной системы есть личный кабинет «АмГУ», предназначенный для студентов и преподавателей. В этой системе они могут вести учет своих личных достижений, взаимодействовать с преподавателями и одногруппниками через чат, контролировать свою посещаемость и успеваемость, заниматься научной деятельностью и получать последние новости университета. Кроме того, система позволяет проводить обучение в дистанционном формате.

Обе информационные системы полностью интегрированы между собой, что обеспечивает информатизацию образовательного процесса. 1С Университет автоматизирует управление процессом обучения, а доступ к этой системе имеют деканаты, кафедры и руководители структурных подразделений вуза. Личный кабинет «АмГУ» доступен для всех сотрудников и студентов университета, и каждый пользователь может добавлять и изменять информацию в 1С Университет из Личного кабинета «АмГУ». Это упрощает работу руководства и обеспечивает надежную систему защиты персональных данных.

1.6 Обзор существующих методов решения рассматриваемой задачи

На данный момент в рассматриваемой организации функционирует следующий алгоритм заселения в общежития, учитывающий некоторые условия, которые понадобятся и при реализации предлагаемого решения в рамках выполнения индивидуального задания. Условия при заселении:

Заселение производится согласно приказам деканатов о заселении.

Заселение будет производиться по курсам, согласно графику, утвержденному приказом ректора;

Заселение производится на основании личных заявлений и Договора найма жилого помещения в общежитии (далее – Договор найма жилого помещения).

Место в общежитие не будет предоставлено студентам:

– имеющим нарушение Правил внутреннего распорядка в общежитиях «АмГУ»;

– имеющим задолженности по оплате за проживание более чем один месяц;

– имеющим прописку в городе непосредственного обучения.

Существующий порядок заселения в общежития «АмГУ» для студента от 26.08.2020:

При себе необходимо иметь следующие документы:

– Паспорт и его копии (1 и 2 разворот с пропиской);

– Комплект фотографий (2 шт. 3x4 для личного дела);

– Результаты флюорографии, полис;

– Копию военного билета (для юношей).

В деканате проверить, есть ли Ваша фамилия в списках на заселение, и в какое общежитие Вы заселяетесь по приказу (там же указан номер секции и комнаты).

Пройти медосмотр в медицинском кабинете общежития № 3 (Ул. Институтская 26/2) и получить справку-разрешение на вселение (при себе иметь флюорографию и полис).

Пропуска для прохода в общежитие для 2-5 курсов активируются при выходе приказа на заселение в общежитие.

Заполнить договор найма жилого помещения у коменданта общежития и расписаться в нем, ознакомиться с Правилами внутреннего распорядка в общежитиях «АмГУ», пройти инструктаж по противопожарной безопасности и расписаться в журнале, ознакомиться с положением о жилищно-бытовой

комиссии, ознакомиться с положением о студгородке (с данными положениями можно заранее ознакомиться на сайте «АмГУ»). Данный пункт включается в себя еще несколько действий со стороны студента, представленные схематично на рисунке 6.

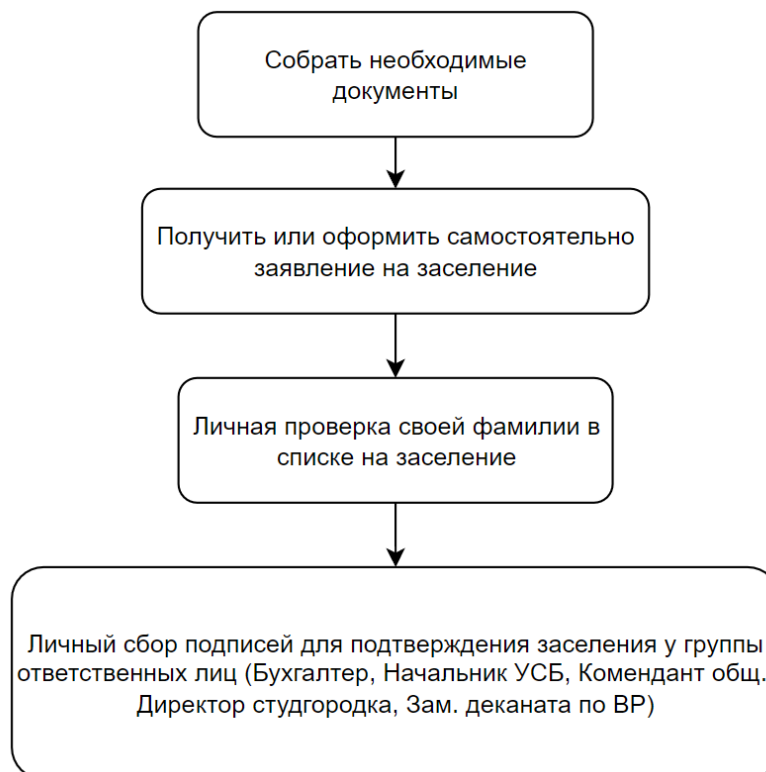


Рисунок 3 – Схематичное представление этапов формирования заявления

Студенты первого года обучения получают кампусную карту (Электронный пропуск) в главном корпусе «АмГУ» в каб.118 (УСБ);

Произвести оплату за проживание в кассе бухгалтерии в главном корпусе «АмГУ» (2 этаж) в течение 15 календарных дней;

Получить ключи у коменданта, осмотреть комнату, заполнить арматурный лист, ознакомиться с приложением № 1 договора найма жилого помещения «Стоимость имущества при заселении в общежитие по договору найма жилого помещения»;

Получить постельные принадлежности у коменданта или у кастелянши.

Для группы лиц, подтверждающих заявление студента необходимо:

– ручное распределение студентов в места их будущего проживания;

- ручное формирование приказа о распределении мест в общежитиях по факультетам;
- личное подтверждение личности заявителя и его данных.

Основываясь на существующем методе решения можно сделать вывод о необходимости автоматизации рассматриваемого бизнес-процесса, путем внедрения в имеющуюся информационную систему программного модуля – «Общежитие», который будет функционировать на основе ключевых положений с соблюдением условий на заселение.

1.7 Формулировка задачи исследования и общей методики ее решения

Система «Общежитие» предназначена для автоматизации подачи заявления в дистанционном формате на переселения или заселения студентов в общежития с точностью до секции и комнаты, в которой он намерен проживать. Каждый студент, у которого нет прописки в городе обучения, имеет доступ к программному модулю – «Общежитие», где в соответствии со своим рейтингом академической успеваемости и, если имеется, наличием особой квоты может выбрать место для проживания. В модуле существует три типа пользователя – студент, модератор и администратор. В первую очередь программно проверяется имеется ли у подающего актуальной информации о прохождении медицинского обследования, а также прописка в городе обучения. При наличии данных нарушений подача заявления на место в общежитии невозможна. Далее определяется тип пользователя и открывает соответствующий интерфейс со своим функционалом.

В процессе подготовки была также сформирована дополнительная задача, которая заключалась в создании системы для интерактивного распределения мест общежитий по факультетам вуза. Данная задача исходила из непосредственной необходимости программного учета и закрепления мест в общежитиях для факультетов. Решив ее каждый факультет сможет получить, то количество мест в общежитии, которое прописано в документе о распределении мест среди факультетов, а также упростит сам процесс распределения.

1.8 Обоснование актуальности темы квалификационного исследования

В настоящее время новые технологии развиваются молниеносными темпами. Технологии помогают более эффективно управлять производством. Интенсивный процесс развития не обошел стороной и сферу образования. Новые информационные технологии позволяют значительно повысить эффективность работы учебного заведения. Чтобы обеспечить оперативный автоматизированный учет информации внедряются и разрабатываются механизмы, способствующие успешной деятельности учебных учреждений. Автоматизация ключевых направлений деятельности высшего учебного заведения является одним из приоритетных направлений развития современного вуза.

Многообразие и сложная взаимосвязь данных внутри бизнес-процессов определяют функциональные и структурные особенности реализации вузовских автоматизированных систем. Каждое образовательное учреждение нуждается в создании специализированных и узконаправленных программных решениях и в принципе полной информационной системой в целом, где необходимо обеспечить тесное взаимодействие обособленных подсистем между собой.

Размещение обучающихся в общежития является важной задачей организации высшего учебного заведения. Задача включает в себя ряд подзадач: размещение жильцов, выселение и переселение жильцов, выдача жесткого и мягкого инвентаря, контроль платы за проживание, контроль выполнения проживающими правил внутреннего распорядка в общежитиях (применения, снятия и обжалования мер дисциплинарного взыскания за нарушения).

Особое внимание, в связи с нехваткой свободных мест в общежитиях, стоит уделить именно процессу заселения обучающихся. Сложность в том, что необходимо вести учет всех нуждающихся в общежитии, которых необходимо заселять в установленном порядке, зависящем от множества условий.

Необходимо учитывать то, что информация способна довольно быстро меняться, особенно в периодах массового заселения. Также ведутся

достаточные затраты денежных средств и времени на печать заявлений, сканов и последующей их транспортировкой в места подписания ответственными лицами.

На основе данных условий появляется конкуренция между студентами. И для того, чтобы правильным образом оценить или классифицировать положение студентов используется система рейтинга. Для его составления необходимо достаточная работа системы мониторинга информации.

Рейтинг предоставляет пользователям полезную и легко сопоставимую информацию, собранную и представленную в одном виде.

Под рейтингом можно также понимать величину, характеризующую субъект и позволяющую при сопоставлении с другим таким же субъектом оценить вероятность превосходства в конкурентной среде либо превосходство в прошедшие определенные периоды времени.

Таким образом разработка автоматизированной системы «Общежитие» для студентов «Амурского Государственного Университета» является актуальной в современном образовательном пространстве, это обуславливается практической потребностью в повышении эффективности управления процессом заселения, рассматриваемом в исследовании.

Каждый университет, независимо от выбранной стратегии, должен пройти цифровую трансформацию, т.е. осуществить внедрение более гибких и бесшовных процессов, изменение корпоративной культуры, оптимизацию процессов [8]. Данная разрабатываемая система подразумевает цифровизацию внутренних процессов университета для увеличения эффективности взаимодействия подразделения, а точнее студгородка, на уровне всего учебного заведения.

Наряду с этим, главным фактором для развития вуза остается единая информационная среда. Она является концепцией которой поддерживается разрабатываемая система так, как полностью интегрируется в информационную и цифровую среду вуза.

2 ПРОЕКТИРОВАНИЕ МОДУЛЯ «ОБЩЕЖИТИЕ»

2.1 Анализ требований к модулю «Общежитие»

В контексте требований к системе представляют собой совокупность условий, которые определяют необходимые функциональные и нефункциональные возможности, а также критерии и ограничения, которые должны быть учтены при разработке и внедрении информационной системы.

Требования к системе включают в себя описание функциональности, интерфейсов, производительности, безопасности, надежности, масштабируемости, а также другие аспекты, необходимые для успешного функционирования системы.

К условиям, влияющим при рассмотрении заявления студента, относятся:

- отличительный признак заселяющегося;
- на каком этапе обучения находится студент;
- наличие льгот;
- учебная успеваемость;
- выполнения внутреннего распорядка в общежитиях (сведения о нарушениях);
- проверка задолженности по оплате места проживания;
- проверка множества документов, подтверждающих личные данные;
- наличие каких-либо достижений по учебной и научной деятельности.

Разрабатываемая система должна включать в себя следующие функциональные требования, при чем они делятся на три стороны.

Со стороны студента предоставляются следующие возможности:

- просмотра текущего места жительства;
- просмотра рейтинга академической успеваемости за последние два семестра, как собственного, так и остальных студентов текущего факультета;
- просмотр нормативных документов;

– подачи заявления, в котором присутствует выбор общежития, секции, комнаты.

Подача заявления осуществляется (при этом сопровождается помощью программы, которая в соответствии с категориями приоритета и индивидуальным признаком) путем выбора студентом непосредственно места жительства, без просмотра остальных заявок студентов факультета.

После подачи заявления имеется информация о статусе заявления:

- заявление находится в рассмотрении;
- заявление подтверждено;
- заявление отклонено с описанием причины.

Со стороны участников комиссии предоставляется отдельный интерфейс, включающий в себя возможности:

- просмотра списка заявлений студентов по факультетам, общежитиям, секциям и комнатам (в случае Зам. декана по ВР, только список студентов по факультетам);
- просмотра актуальных заявок на проживание;
- одобрение или отклонение заявления с описанием причины. Все результаты решения отображаются непосредственно в списках заявлений;
- в случае «Зам. деканов по ВР» основная деятельность – распределение студентов в рамках своего факультета по заранее предоставленным местам в общежитии;
- предоставление необходимой сортировки заявлений и предпросмотра распределенных студентов.

Кроме того, учитываются оценки по практикам, по курсовым работам и проектам, а также зачетам с оценкой.

Администратору предоставляется полный доступ к функционалу, который доступен как модератору, так и студенту. Кроме того, администратор может создавать заявления от имени заявителя без его прямого участия, однако такая возможность доступна только администратору и применяется только в исключительных случаях.

Учитывая, что модуль обрабатывает персональные данные студентов, необходимы меры для обеспечения безопасности, сохранения и проверки достоверности этих данных.

Формируя требования к подсистеме для вычисления потенциальной рабочей нагрузки полезно выявить наиболее важные с точки зрения бизнеса и пользовательских сценариев, а для каждого из них определить профиль нагрузки, то есть организовать низкие требования к производительности системы компьютера и возможных его компонентов.

2.2 Структурно-функциональная модель модуля «Общежитие»

Структурно-функциональная модель является методом анализа и описания организации бизнес-процессов на предприятии. Она помогает понять структуру и взаимосвязи между различными функциями и подразделениями организации, а также определить, как эти функции выполняются и какие ресурсы требуются для их реализации. СФМ представляет собой графическую модель, которая отображает структуру организации, роли и обязанности сотрудников, а также поток информации и материалов между различными компонентами системы.

Основная цель СФМ заключается в оптимизации бизнес-процессов предприятия и повышении его эффективности. Она помогает выявить слабые места в организации, улучшить взаимодействие между различными отделами и процессами, а также улучшить управление и контроль над выполнением задач.

При описании бизнес-процессов предприятия в рамках СФМ можно использовать различные стандарты и методологии. Основные из них:

– BPMN (Business Process Model and Notation) – это стандартная графическая нотация для моделирования бизнес-процессов. Она предоставляет набор символов и правил для создания диаграмм, которые отображают последовательность шагов, потоки данных и участников процесса;

– IDEF0 (Integrated Definition for Function Modeling) – это методология, которая позволяет описывать функции и связи между ними в организации. Она

предоставляет нотацию для создания функциональных диаграмм, которые показывают входы, выходы и управляющие механизмы функций;

– UML (Unified Modeling Language) – это унифицированный язык моделирования, который широко используется в различных областях, включая бизнес-моделирование. UML предоставляет нотацию и инструменты для создания различных диаграмм, таких как диаграммы классов, диаграммы активностей и диаграммы последовательностей, которые можно использовать для описания бизнес-процессов.

Для функционального моделирования был выбран стандарт IDEF0 так, как он в свою очередь наглядно позволит отобразить механизмы деятельности структуры или части организации в виде иерархии взаимосвязанных ней функций.

Далее изображена диаграмма рассматриваемого процесса, которая представляет внешний уровень обрабатываемых данных, рисунок 4. На ней представлен перечень входных и выходных данных. Также указаны основные исполнители: сотрудники комиссии по заселению, одобряющие, распределяющие или отклоняющие заявления; студенты первого и выше курса, которые осуществляют выбор комнаты и подают заявления.

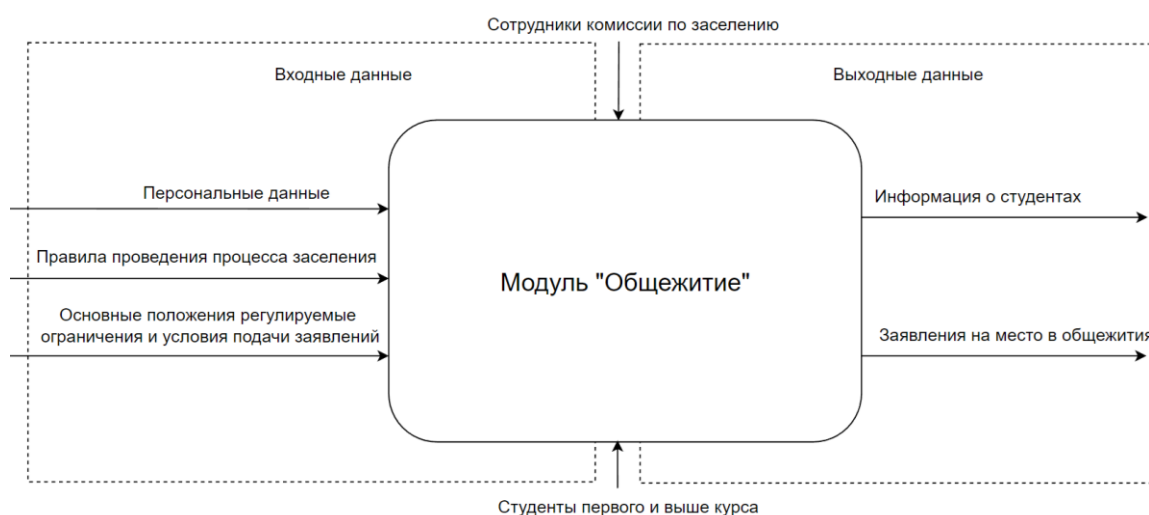


Рисунок 4 – Структурно-функциональная модель модуля «Общежитие»

Декомпозиция системы, изображенная на рисунке 5, описывает внутренние процессы модуля, которые связаны с его основной функциональностью.

Она также представляет сущности, которые взаимодействуют с внешними входными данными.



Рисунок 5 – Декомпозиция модуля «Общежитие»

На ряду с этим необходимо рассмотреть основные описанные сущности, «Подача заявления» и «Обработка заявления», для полного раскрытия сути автоматизации используется декомпозиция, представленная на рисунках 6 и 7.

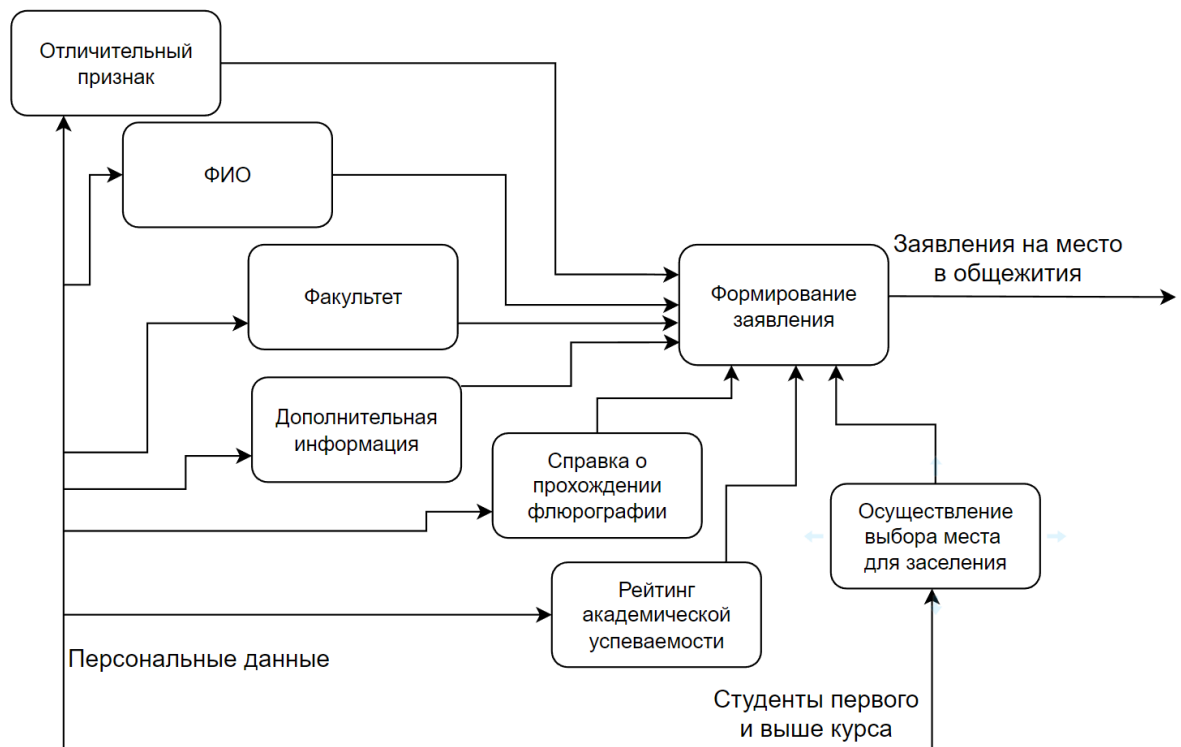


Рисунок 6 – Декомпозиция модуля «Подача заявления»

Декомпозиция модуля «Поддача заявления», описывает какие манипуляции совершаются студентами, а также какую информацию он получает от системы.

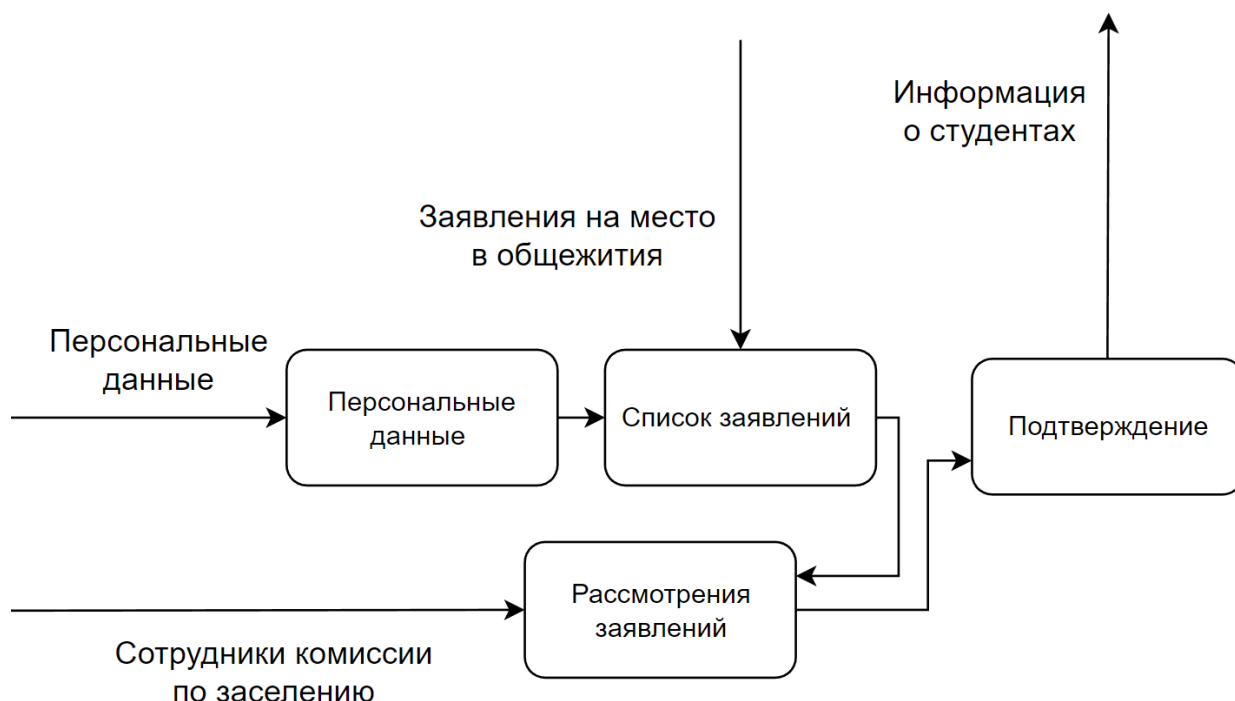


Рисунок 7 – Декомпозиция модуля «Обработка заявления»

Декомпозиция модуля «Обработка заявления» представляет информацию и функциональность, доступные сотрудникам комиссии, а также описывает входные и выходные данные данного модуля.

2.3 Обоснование используемых обеспечивающих подсистем модуля «Общежитие»

Обеспечивающие подсистемы – это компоненты или элементы, которые поддерживают работу основных систем или проектов, обеспечивая им необходимую функциональность, ресурсы и условия для эффективной работы. Они играют важную роль в обеспечении оптимальной производительности и надежности системы в целом.

В частности, выявляют такие подсистемы как: лингвистические, информационные, технические, программные, математические, правовые и организационные.

Для модуля «Общежитие» включающего расчет рейтинга успеваемости

студентов и строго регламентированные правила заселения, могут быть использованы следующие обеспечивающие подсистемы:

– Лингвистическая подсистема. Система коммуникации с пользователями на территории Российской Федерации использует русский язык, обеспечивающая понятное взаимодействие и передачу информации студентам, персоналу общежития и сотрудникам учебного заведения;

– Информационная подсистема. База данных, содержащая информацию о студентах, их успеваемости, заявках на заселение и других сопутствующих данных. Система учета и обработки данных для расчета рейтинга успеваемости студентов и составления сводной информации для администрации;

– Техническая подсистема. Компьютеры, серверы и сетевое оборудование для обработки информации и обеспечения связи между различными компонентами системы. Средства безопасности, такие как системы контроля доступа, видеонаблюдение и пожарная сигнализация, для обеспечения безопасности студентов и их имущества;

– Программная подсистема. Специализированное программное обеспечение для автоматизации процессов, связанных с расчетом рейтинга успеваемости, обработкой заявок на заселение, генерацией отчетов и уведомлений для студентов и персонала общежития;

– Математическая подсистема. Алгоритмы и модели для расчета рейтинга успеваемости, дальнейшего распределения студентов по комнатам и оптимизации заселения в соответствии с заданными правилами и ограничениями;

– Правовая подсистема. Нормативно-правовая база, включающая положения и правила, регламентирующие заселение студентов в общежитие, установление прав и обязанностей студентов, а также процедуры разрешения возможных конфликтных ситуаций;

– Организационная подсистема. Управление персоналом общежития, включая распределение задач и обязанностей, организацию процесса заселения и контроль за соблюдением правил. Процедуры и документация для учета

заявок на заселение, проведения сроков, правил выселения и рассмотрения жалоб.

Все эти подсистемы работают в совокупности, обеспечивая эффективное и надежное функционирование модуля «Общежития», включающего расчет рейтинга успеваемости студентов, распределение студентов и строго регламентированные правила заселения.

2.4 Обоснование выбора средств разработки

На начальном этапе разработки необходимо выбрать язык программирования, определиться со средой разработки и разработать стратегию.

2.4.1 Выбор языка программирования

При выборе языка понадобится сравнение главных представителей веб-индустрии, такие как: Python, Ruby, JavaScript и PHP.

Python является интерпретируемым, высокоуровневым языком программирования, разработанным Гвидо ван Россумом. Он был создан с упором на читаемость и простоту кода, что делает его очень популярным среди начинающих программистов. Язык имеет строгую синтаксическую структуру, основанную на отступах, что делает код более читабельным. Python поддерживает ООП и имеет обширную стандартную библиотеку, которая обеспечивает множество функциональных возможностей.

Ruby – это интерпретируемый, объектно-ориентированный язык программирования, созданный Юкихио Мацумото. Он был разработан с упором на простоту и удобство использования, с акцентом на ясность и выразительность. Язык имеет гибкую синтаксическую структуру, которая позволяет программистам писать код в естественном языке [13]. Он также поддерживает множество парадигм программирования, включая функциональное и императивное программирование. Ruby имеет богатую экосистему библиотек и фреймворков, таких как Ruby on Rails.

JavaScript – это интерпретируемый язык программирования, который обычно используется для создания интерактивных веб-страниц [10]. Он был разработан Бренданом Айком в 1995 году. JS является языком с динамической

типизацией и может выполняться непосредственно в браузере. Он широко используется для создания клиентской части веб-приложений, включая динамическое изменение содержимого страницы, взаимодействие с пользователем и отправку данных на сервер. JS также используется в серверной разработке с использованием платформы Node.js.

PHP – это скриптовый язык программирования, который обычно используется для разработки веб-приложений и динамических веб-страниц. Он был создан Расмусом Лердорфом и впервые выпущен в 1995 году. PHP является языком сценариев на стороне сервера и обычно используется вместе с HTML для создания динамического содержимого веб-страниц. Он обладает широкой функциональностью и интеграцией с различными базами данных. PHP также поддерживает объектно-ориентированное программирование. Все рассматриваемые языки программирования широко применяются в веб-индустрии и имеют свои особенности, а также положительные и отрицательные стороны. Выбор языка программирования зависит от требований проекта, личных предпочтений программиста и экосистемы инструментов, доступных для каждого языка.

Для более подробного анализа необходимо применить метод морфологического разбора, основанном на целях проекта, используемых ранее технологий. Представление данного разбора представлен в таблице 1.

Таблица 1 – Анализ языков программирования

Ключевые параметры для оценки	Python	JS	Ruby	PHP
Скорость обработки	7	6	8	6
Простота обучения	8	7	7	7
Читабельность	8	8	6	6
Расширяемость	8	7	9	8
Экосистема	8	8	9	8
Поддержка сообщества	8	8	9	8
Наличие библиотек и фреймворков	10	8	10	9
Итоговое значение	57	52	58	52

После проведения анализа видно, что лидирующее место занимает язык Python, но итоговый выбор стоит за Ruby так, как при разработке необходимо учитывать, что модуль интегрируется в уже имеющуюся систему, где используется Ruby on Rails. Благодаря встроенной библиотеки ActiveRecord организуется быстрая и надежная работа с БД [14].

2.4.2 Выбор среды разработки

При выборе среды разработки необходимо учитывать какие будут использоваться в работе языки программирования, их функциональность и цели разработки. Самые популярные среды для веб-программирования: Visual Studio и RubyMine [11].

Visual Studio – это интегрированная среда разработки, разработанная компанией Microsoft. Она широко используется для разработки веб-приложений, включая веб-сервисы, на разных языках, включая Ruby и JavaScript. Некоторые особенности Visual Studio для работы с Ruby и JavaScript включают:

VS имеет расширения и плагины, которые обеспечивают поддержку Ruby, такие как RubyMine, Ruby in Steel и другие. Они предоставляют средства для синтаксического подсветки, автодополнения, отладки и других функций, упрощающих разработку на Ruby. VS предлагает мощную поддержку JavaScript с возможностями отладки, автодополнения, средствами проверки синтаксиса, интеграцией с пакетными менеджерами, такими как npm, и многими другими функциями. VS имеет встроенную поддержку системы контроля версий Git, что делает управление версиями проекта более удобным и позволяет командам разработчиков совместно работать над проектом. Visual Studio имеет большое количество плагинов и расширений, которые могут значительно расширить его функциональность и поддержку различных языков и фреймворков.

RubyMine – это интегрированная среда разработки, разработанная компанией JetBrains, специализирующаяся на инструментах разработки для различных языков программирования. RubyMine создана специально для разработки на Ruby и Ruby on Rails, но также поддерживает JavaScript. RubyMine

предлагает расширенные инструменты для разработки на Ruby, включая автодополнение кода, подсветку синтаксиса, отладчик, интегрированные тестовые фреймворки и другие полезные функции, специфичные для Ruby. RubyMine обеспечивает поддержку JavaScript средствами автодополнения, проверки синтаксиса, отладки и других инструментов, необходимых для эффективной разработки на этом языке. На ряду с этим среда разработки предлагает интеграцию с фреймворком Ruby on Rails, обеспечивая автоматическую генерацию кода, навигацию по проекту, поддержку миграций базы данных и другие функции, специфичные для Rails. RubyMine включает инструменты статического анализа кода, которые помогают выявлять потенциальные проблемы и предлагают рекомендации по улучшению кода. Он также предоставляет широкие возможности для рефакторинга кода. Интеграция с Git и другими системами контроля версий: RubyMine интегрируется с популярными системами контроля версий, включая Git, SVN и Mercurial, обеспечивая удобный интерфейс для работы с репозиториями.

По итогу, как Visual Studio, так и RubyMine предоставляют разработчикам инструменты и функции, которые помогают разрабатывать веб-сервисы на языках Ruby и JavaScript. Однако необходимо учитывать, что RubyMine ориентирована на разработку на Ruby и Ruby on Rails и предлагает более специализированные функции для этой цели, в то время как VS является более универсальной средой разработки с широкой поддержкой различных языков программирования и фреймворков.

2.4.3 Выбор программных платформ разработки

На данный момент существует множество платформ разработки под веб-приложения, обычно они носят названия фреймворков.

Фреймворк – программная платформа, определяющая структуру программной системы или программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

Все они предназначены как для различных узконаправленных задач, так и для их комбинирования.

Исходя из выбранных ранее языков программирования и целей разработки выявляются несколько кандидатов для анализа:

- со стороны Бэкэнд (Ruby): Ruby on Rails, Sinatra, Hanami;
- со стороны Фронтэнд (JS): React.js, Angular.js, Vue.js.

Такое разделение обусловлено тем, что любое веб-приложение делится на две основные части, фронтэнд и бэкэнд.

Фронтэнд – это часть веб-приложения, которую видят и с которой взаимодействуют пользователи. Он отвечает за представление и визуальное отображение данных, интерфейс пользователя и пользовательский опыт. Фронтэнд обычно состоит из HTML, CSS и JavaScript.

Далее важно провести оценку и анализ фреймворком JS для того, чтобы выделить наиболее подходящий. Результаты дангого разбора представлены в таблице 2.

Таблица 2 – Анализ фреймворков JS

Ключевые параметры для оценки	React.js	Angular.js	Vue.js
Производительность	8	7	7
Гибкость и масштабируемость	7	8	9
Сообщество и поддержка	9	9	8
Обучаемость	7	6	9
Экосистема и интеграция	8	9	7
Совместимость	8	8	8
Итоговый результат	47	47	48

Плюсы React.js [17]:

- React использует виртуальную DOM, что позволяет эффективно обновлять только необходимые компоненты и элементы интерфейса, что улучшает производительность;
- React имеет широкое сообщество разработчиков и огромное количество ресурсов, учебных материалов и сторонних библиотек, что облегчает разработку и решение проблем;
- React можно использовать как библиотеку для разработки отдельных

компонентов, так и как полноценный фреймворк для создания сложных веб-приложений.

Минус: необходимость использования сторонних библиотек. React предоставляет только базовые функциональности, поэтому для некоторых распространенных задач, таких как управление состоянием (state management), маршрутизация и другие, вам потребуется выбрать и настроить дополнительные библиотеки;

Плюсы Angular.js [16]:

– Angular предоставляет обширный набор инструментов и функций для разработки веб-приложений, включая маршрутизацию, управление состоянием, валидацию форм и другие;

– Angular разрабатывается и поддерживается командой Google, что обеспечивает стабильность, обновления и безопасность;

– у Angular есть большое сообщество разработчиков, которое активно обсуждает проблемы, делится знаниями и создает сторонние пакеты и библиотеки.

Минусы:

– Angular имеет большой размер, поэтому загрузка и выполнение приложения может занять больше времени по сравнению с другими фреймворками;

– из-за своего обширного набора функций и концепций, Angular может быть сложным для изучения и понимания, особенно для новичков.

Плюсы Vue.js:

– Vue.js обладает очень простым и понятным синтаксисом, что делает его легко изучаемым и использованием для начинающих разработчиков;

– Vue.js предоставляет реактивность из коробки, что означает, что данные и представление автоматически обновляются при изменении состояния данных;

– Vue.js может использоваться как библиотека для добавления интерактивности на отдельные страницы, так и как полноценный фреймворк для создания сложных приложений.

Минусы:

– в сравнении с React и Angular, сообщество Vue.js немного меньше, хотя оно все равно активно и быстро растет;

– возможно, вам понадобится использовать дополнительные библиотеки или инструменты для некоторых распространенных задач, таких как управление состоянием, что может потребовать больше настройки.

Исходя из анализа и выявленных положительных и отрицательных черт выдвинутых фреймворков, выбор останавливается на Vue.js. Его ядро в первую очередь решает задачу уровня представления (view), что упрощает интеграцию с другими библиотеками или существующими проектами. Данная технология применяется для создания сложных одностраничных приложений (SPA). Компонентного подхода, который использует фрейворк Vue.js представлен на рисунке 8.

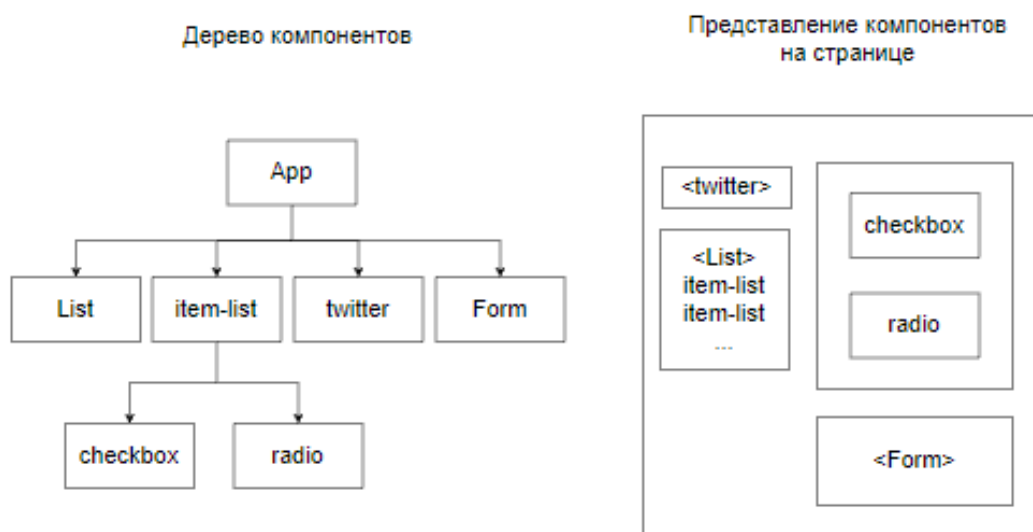


Рисунок 8 – компонентный подход Vue.js

Бэкенд – это часть веб-приложения, которая отвечает за обработку данных и бизнес-логику. Он работает на серверной стороне и взаимодействует с базами данных, обрабатывает запросы от клиента и возвращает соответствующие данные.

Используя те же методы и параметры сравнения выявляется лучший фреймворк для функционального бэкенда, разбор представлен на таблице 3.

Таблица 3 – Анализ фреймворков Ruby

Ключевые параметры для оценки	Ruby on Rails	Sinatra	Hanami
Производительность	8	7	8
Гибкость и масштабируемость	9	7	9
Сообщество и поддержка	10	8	7
Обучаемость	8	9	7
Экосистема и интеграция	9	7	8
Совместимость	9	8	8
Итоговый результат	53	46	47

Плюсы Ruby on Rails:

- Rails предлагает соглашения по конфигурации, что позволяет разработчикам быстро приступить к разработке и сосредоточиться на бизнес-логике приложения;

- Rails предлагает богатый стек функциональности, включая ORM (Active Record), систему маршрутизации, шаблонизацию (Action View), автоматическую проверку моделей и многое другое;

- Ruby on Rails имеет огромное и активное сообщество разработчиков, что обеспечивает доступ к обучающим материалам, инструментам, библиотекам и поддержке.

Минусы:

- В некоторых случаях Ruby on Rails может столкнуться с проблемами масштабирования на очень больших проектах или при высоких нагрузках. Тем не менее, с правильной оптимизацией и настройками можно достичь хорошей производительности;

- Для новичков в Ruby on Rails, может потребоваться время для освоения фреймворка и его особенностей. Кроме того, поскольку Rails активно развивается, изменения в новых версиях могут потребовать обновления кода в существующих проектах.

Плюсы Sinatra:

- Sinatra является минималистичным фреймворком, который

предоставляет только необходимый минимум инструментов. Это позволяет разработчикам больше свободы в организации кода и выборе используемых библиотек;

– Sinatra позволяет создавать приложения с более гибкой структурой, поскольку он не налагает строгих соглашений о структуре проекта. Это особенно полезно, когда требуется разработка небольших или прототипных приложений.

Минусы:

– отсутствие готовых решений: В отличие от Rails, Sinatra предоставляет только базовые инструменты для обработки HTTP-запросов. Для некоторых функциональностей, таких как ORM или шаблонизация, вам придется выбрать и настроить дополнительные библиотеки самостоятельно;

– из-за своей простоты и легковесности, Sinatra может оказаться неэффективным при разработке крупных и сложных проектов, где требуется более структурированный и организованный подход.

Плюсы Hanami:

– Hanami предлагает модульную архитектуру, позволяющую разделять приложение на независимые компоненты. Это способствует созданию расширяемых и легко поддерживаемых приложений;

– Hanami разработан с учетом принципов функционального программирования, что может быть привлекательным для разработчиков, предпочитающих такой стиль кодирования.

Минусы:

– Hanami является относительно новым фреймворком и не имеет такого большого и активного сообщества, как Ruby on Rails. Это может ограничить доступность ресурсов и поддержку;

– из-за своих отличий от традиционных фреймворков, Hanami может потребовать дополнительного времени на изучение и привыкание к его подходам и практикам.

Многоуровневый фреймворк Ruby on Rails для построения веб-

приложений, использующих базы данных, который основан на паттерне Модель-Представление-Контроллер (MVC), представленная на рисунке 9 [18].

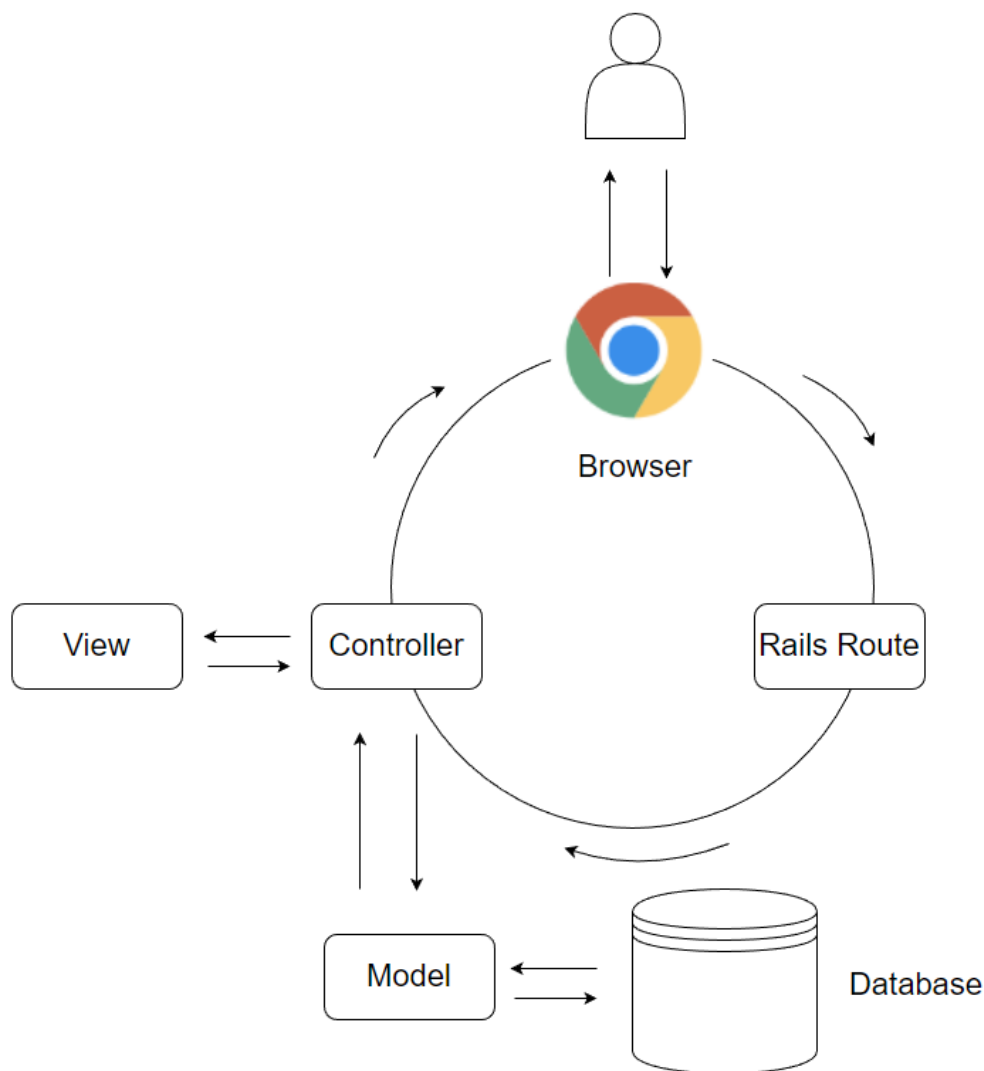


Рисунок 9 – Шаблон MVC

Исходя из анализа выбор остается на Ruby on Rails так, как резюмируя, он предлагает мощные инструменты и конвенции, которые помогут создать и поддерживать качественное веб-приложение.

2.5 Анализ данных

В основе выдвинутого технического задания, представленного в приложении А, выявлены ключевые данные для обработки их в системе:

- ФИО;
- пол;
- регистрационный адрес;

- номер телефона;
- отличительный признак студента;
- факультет обучающегося;
- специальность;
- рейтинг академической успеваемости за последние два учебных семестра;
- время подачи заявления;
- уровень образования;
- семейное положение;
- места в общежитиях (секции, комнаты).

Модуль «Общежитие» использует данные, представленные выше как в целях верификации, так и в обычном их представлении для пользователя.

2.6 Проектирование структуры модуля и взаимодействующих систем

Для функционального проектирования структуры модуля «Общежитие» и взаимодействующих с ним систем было выбрано программное обеспечение ArgoUML использующий унифицированный язык моделирования UML.

ArgoUML – это свободное программное обеспечение с открытым исходным кодом, предназначенное для моделирования и проектирования программных систем с использованием диаграмм UML (Unified Modeling Language).

ArgoUML разрабатывается на языке программирования Java и использует платформу Java для своей работы. Он поддерживается на различных операционных системах, таких как Windows, macOS и Linux. Java обеспечивает кроссплатформенность и позволяет ArgoUML работать на различных окружениях без необходимости изменений в исходном коде.

На ряду с этим предоставляется возможности для создания, редактирования, анализа и документирования моделей UML, а также поддерживает функциональность импорта и экспорта моделей в различные форматы файлов.

ArgoUML имеет интуитивно понятный интерфейс, который облегчает работу с диаграммами и моделями, поддерживает расширяемость с помощью

плагинов, что позволяет добавлять дополнительные функции и интегрировать его с другими инструментами и системами.

Также данное программное обеспечение предоставляет набор инструментов для создания и визуализации различных видов диаграмм UML, которые будут использоваться и для проектирования модуля «Общежитие»:

- а) диаграммы прецедентов;
- б) диаграммы последовательностей;
- в) диаграммы состояний;
- г) диаграммы классов.

Диаграммы прецедентов позволяют описать функциональность системы с точки зрения взаимодействия между «актерами» (пользователи, внешние системы и т.д.) и основными действиями (прецедентами), которые система выполняет (рассмотрено в пункте а). Они помогают определить основные «актеры» и их взаимодействие с системой, а также обеспечивают общее представление о функциональных требованиях системы.

Для разрабатываемой системы были выявлены ключевые «актеры»: студент, бухгалтер, директор студгородка, начальник УСБ, комендант общежития, зам. декан по ВР.

Соответственно студент имеет возможность просмотра правил по подаче заявления, просмотр мест в общежитиях в рамках факультета, просмотр личного рейтинга академически успеваемости, подача заявления и редактирование заявления. Данные возможности выступают в роли вариантов использования данной схемы. Дополнительно существует связь двух вариантов, которая описывает то, что само поданное заявление в последствии попадает в общий список заявлений.

Бухгалтер, директор студгородка, начальник УСБ, комендант общежития осуществляют проверку, просмотр заявлений в рамках факультета и одобрение его одобрение, дополнительно к этому могут просматривать персональную информацию, сосредоточенную в заявлении. Зам. декана по ВР имеет точно такие же возможности за исключением распределения студентов по

местам, указанным в заявлении. Данная диаграмма представлена на рисунке 10.

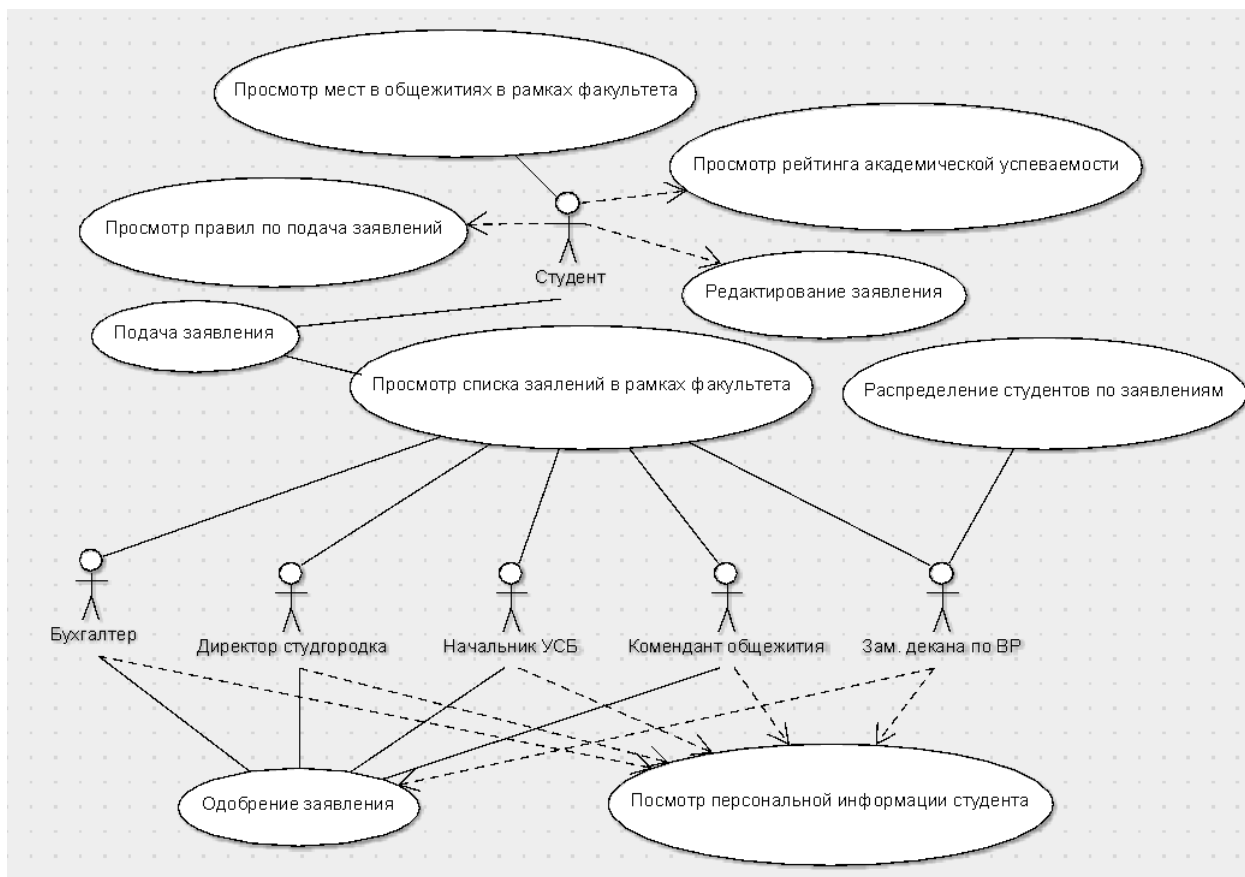


Рисунок 10 – Диаграмма вариантов использования

Далее следуют диаграммы последовательности используемые для визуализации взаимодействия между различными объектами или компонентами в рамках рассматриваемой системы и процессов (рассмотрено в пункте б).

Первая диаграмма будет описывать процесс подачи заявления студентом на место в общежитие с последующим его сохранением в БД и определением в общий список заявлений.

Рассматривая подробнее, студент просматривает правила подачи, места в общежитиях, выбирает их и указывает, формируется заявление, отправляется на сервер, там валидируется.

При успешной проверке методами валидации сервера, обратно пользователю возвращается текущее поданное заявление с возможностью его редактирования. Данная диаграмма представлена на рисунке 11.

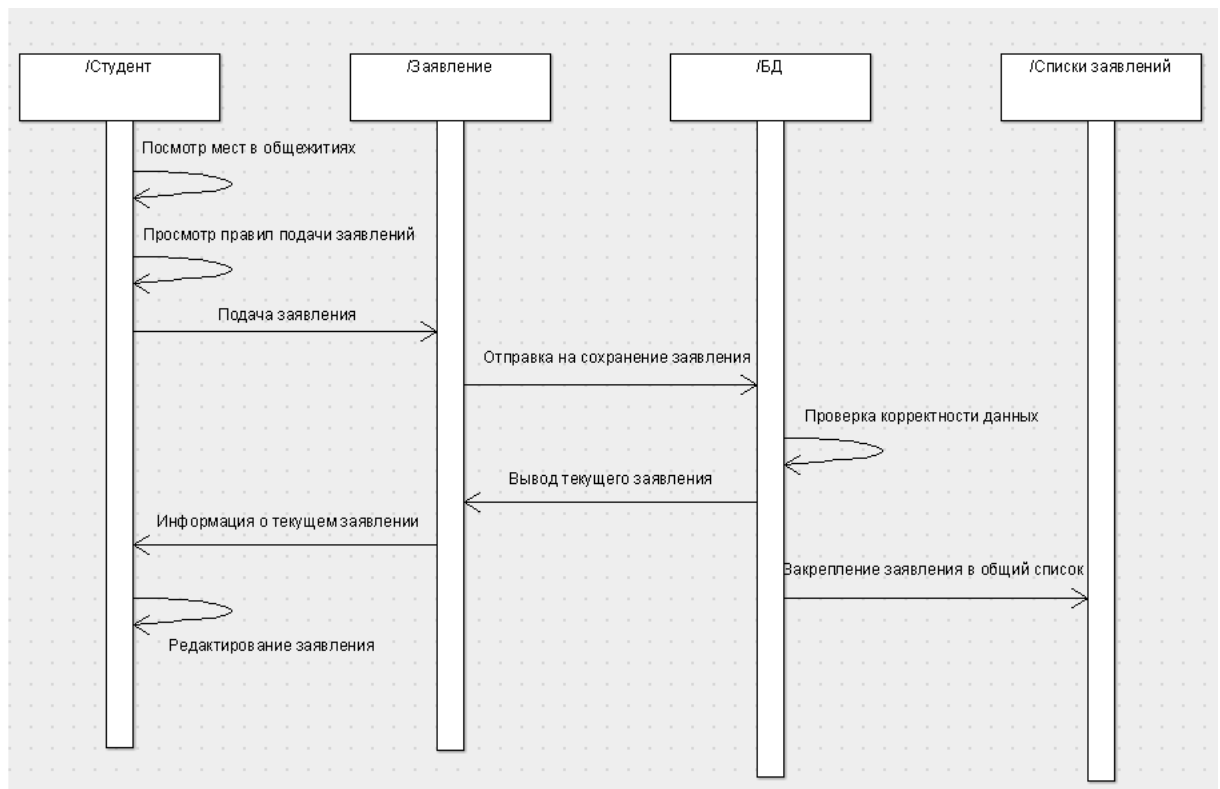


Рисунок 11 – Диаграмма последовательности студента

Вторая диаграмма описывает процесс получения списка заявлений для членов комиссии по проверке заявлений, а также последующую верификацию этих заявлений. Давайте рассмотрим этот процесс более подробно.

После сбора заявлений они передаются представителям комиссии для рассмотрения. Каждый представитель получает список заявлений, которые ему нужно проверить и верифицировать. Представители комиссии просматривают предоставленные данные и проверяют соответствие заявлений требуемым критериям. Они анализируют информацию, проводят необходимые проверки и решают, следует ли одобрить заявление или нет. После завершения проверки представитель комиссии вносит результаты своей работы в общий список заявлений.

Одобренные заявления, находящиеся в общем списке, затем распределяются определенным представителем комиссии. Этот представитель отвечает за запись одобренных заявлений в базу данных и дальнейшее закрепление их в виде документов в системе 1С Университет. Данная диаграмма, представленная на рисунке 12.

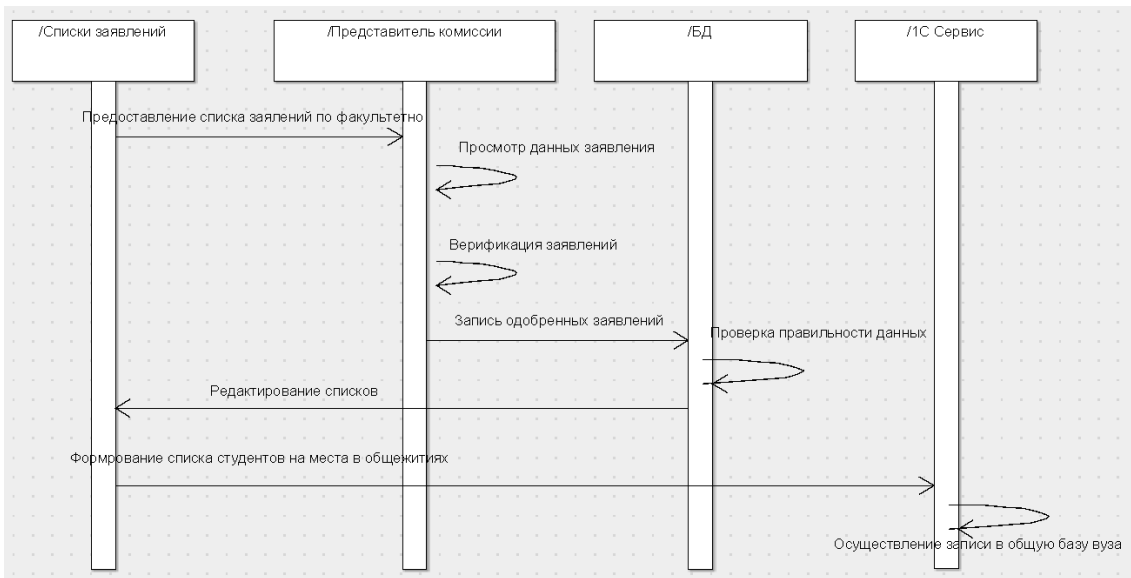


Рисунок 12 – Диаграмма последовательности верификации заявлений

После описания последовательности выполнения действий определенными объектами в системе проводится описание поведений их состояний и событий, которые могут с ними происходить (рассмотрено в пункте в). Диаграмма, представленная на рисунке 13, описывает последовательные действия завязанные на перемещение по системе заявления от начала его подачи и до верификации.

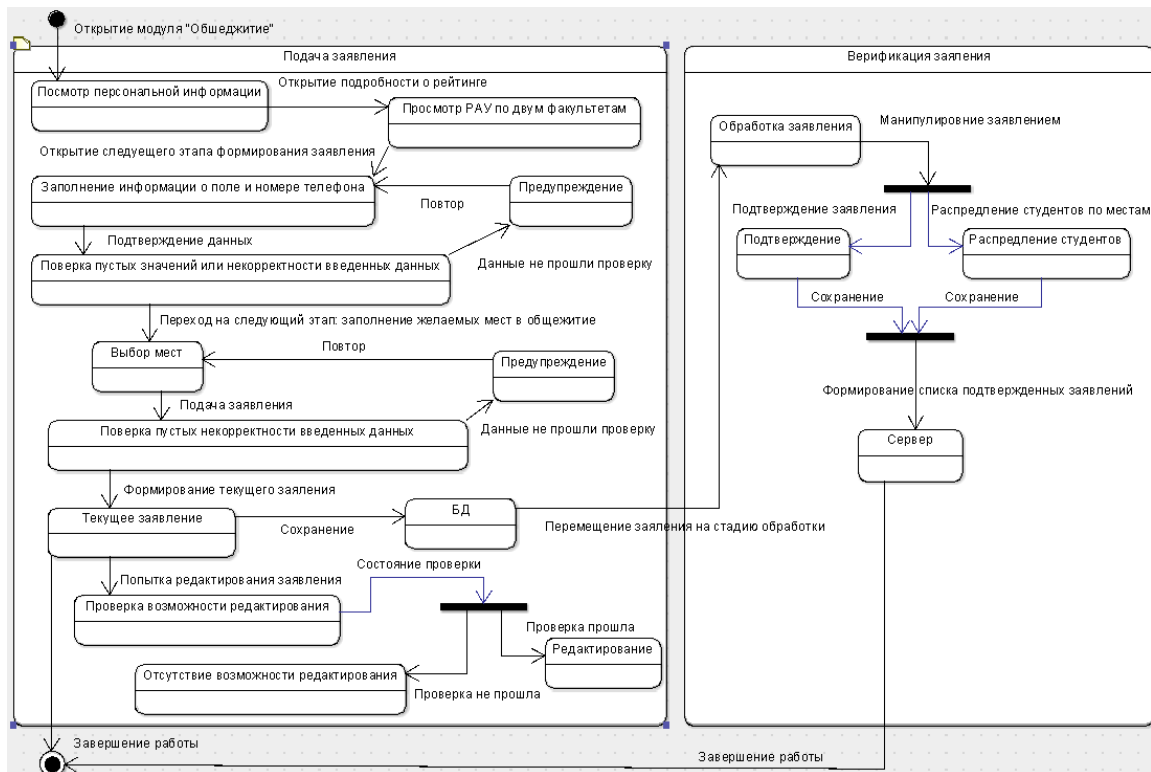


Рисунок 13 – Диаграмма состояний

Следующей заключительным этапом проектирования структуры разрабатываемой системы является построение диаграммы классов (рассмотрено в пункте г). Она используется для визуализации структуры системы, ее классов и связей между ними.

Диаграмма включает основные классы, наполненные основными наборами методов и полей для максимально эффективного функционирования.

Выделяются такие классы как: роли; пользователь; заявления; факультеты; места для факультетов; комментарии к заявлению и структура общежития. Взаимодействие между классами, их функции и поля представлены на рисунке 14.

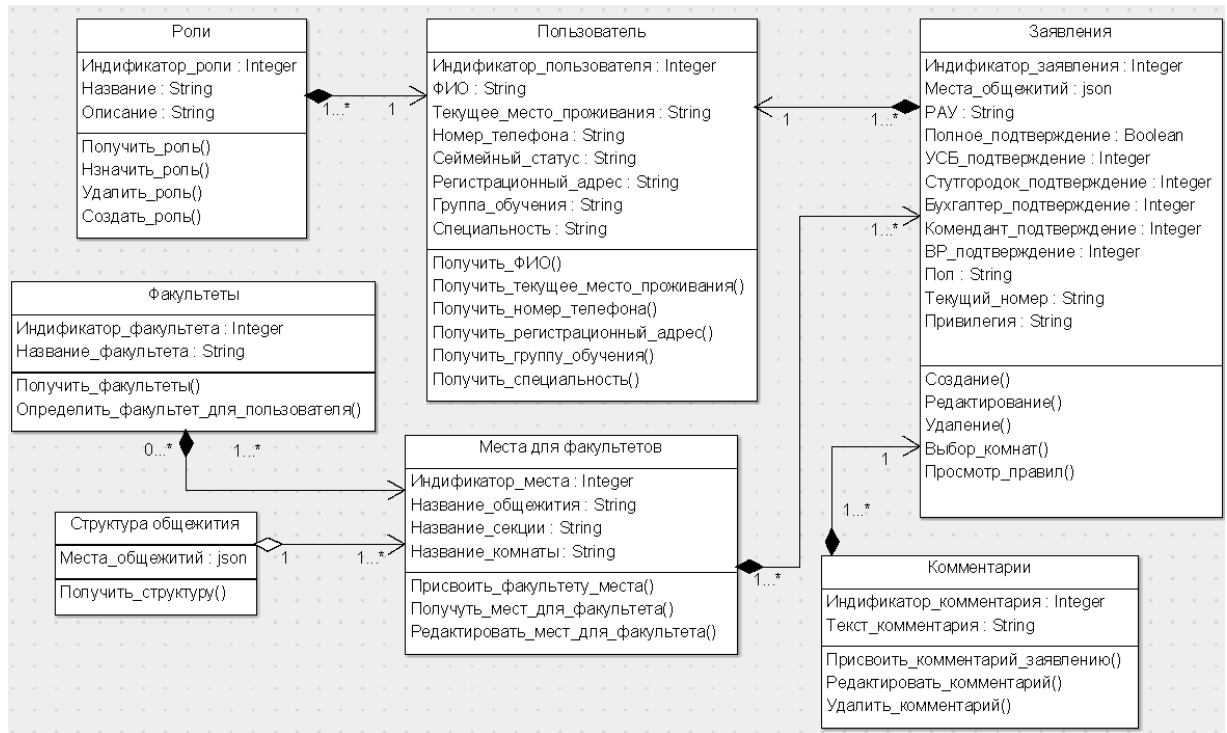


Рисунок 14 – Диаграмма классов

Данная диаграмма классов позволит обеспечить более четкую и структурированную архитектуру приложения, что в свою очередь облегчит его разработку, тестирование, масштабируемость и поддержку.

2.7 Разработка структуры пользовательского интерфейса

Исходя из выбора фреймворка для фронтэнд части, а точнее для визуальной составляющей модуля «Общежитие», необходимо выделить ключевые моменты при его использовании.

На ряду с Vue.js используется Vuetify – это библиотека пользовательского интерфейса, не требующая навыков проектирования, с прекрасно выполненными компонентами Vue.

Данные технологии образуют мощную комбинацию для разработки интерфейса веб-приложения. В их основе стоят такие преимущества и особенности как:

- возможность описывать пользовательский интерфейс с помощью декларативного синтаксиса, которой понятен и легко поддерживается;
- завязка на компонентах, что приводит к созданию модульных и переиспользуемых элементов интерфейса;
- поддержка реактивной системы для автоматического отслеживания зависимости между данными и их представлением. Двухнаправленная связь, позволяющая создавать динамическое приложение;
- отзывчивость дизайна, представление широкого спектра готовых стилей и компонентов, позволяющие создавать красивые и отзывчивые пользовательские интерфейсы;
- гибкая настройка позволяющая конфигурировать внешний вид и поведение компонент под определенные нужды;
- наличие механизмов виртуального DOM и эффективной системой рендеринга, что способствует оптимизации производительности интерфейса;
- предоставление широкого спектра возможностей для расширения и интеграции сторонних пакетов, библиотек и плагинов.

В сочетании с идеально подобранными технологиями, одно из главных значений, влияющих на качество интерфейса, является проработка ключевых аспектов при разработке пользовательского интерфейса. В их состав входит:

- понятность и простота использования;
- консистентность (поддержание одного стиля);
- отзывчивость и скорость;
- визуальное оформление;
- адаптивность (переопределение вида, исходя из используемого

устройства);

– доступность.

Аккумулируя вышеописанное и ориентируясь на требования для реализации модуля, выстраивается определенное представление внешнего функционального вида. Он делится на несколько частей: часть, ориентированная на студента; часть, ориентированная на участников комиссии.

Часть для студентов представляет собой три функциональных блока (представлены на рисунке 15):

- персональная информация, содержащая рейтинг академической успеваемости с возможностью открытия его подробной расшифровки, информацию о ФИО, факультете специальности и отличительного признака студента;
- информация о текущем месте проживания в рамках общежития;
- кнопка подачи заявления, при нажатии на которую появляется отдельный поэтапный процесс проверки личных данных и подачи заявления.

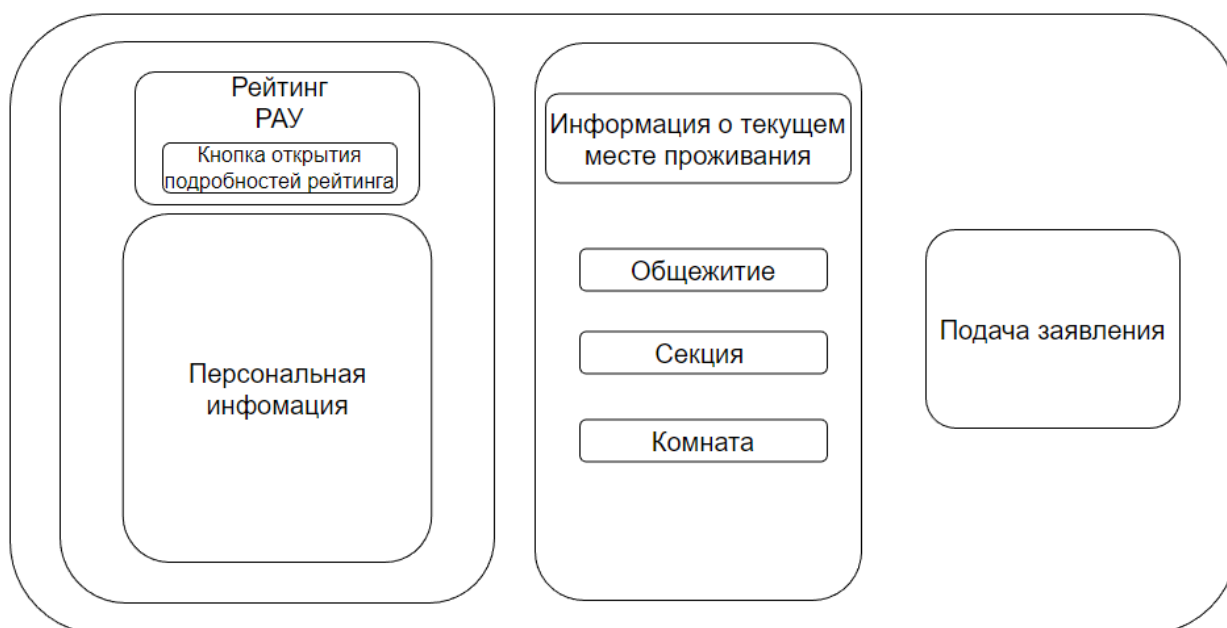


Рисунок 15 – Эскизный общий вид интерфейса стороны студента

На рисунке 16, представлен вид расшифровки рейтинга академической успеваемости с указанием семестров, взаимодействующих в рейтинге и видом таблице типа, дисциплина – оценка.

Расшифровка рейтинг за два последних семестра

Дисциплина	Оценка
.....
.....
.....
.....

Кнопка закрытия

Рисунок 16 – Эскизный вид расшифровки рейтинга

При нажатии на кнопку «Подать заявление», интерфейс открывает диалоговое окно, которое позволяет пользователю ознакомиться с правилами подачи заявления. Далее пользователю предлагается вписать некоторую персональную информацию, а с остальной ознакомиться. Итогом осуществляется выбор места для проживания и подача заявления.

Первый этап заключается в ознакомление с правилами и основной информации о подаче заявления. Пользователь может прочитать и убедиться, что он понимает требования и процедуру.

Второй этап основывается на проверке персональной информации, которая извлекается из общей информационной базы вуза, а также предлагается заполнить свою персональную информацию, такую как пол и актуальный номер телефона на момент подачи заявления.

Третий этап заключается в выборе мест для проживания, а конкретнее предоставляется возможность выбрать до трех мест в общежитии из списка доступных вариантов. Здесь же производится подача заявления. После проверки персональной информации и выбора мест в общежитии, пользователь

может нажать на кнопку «Подать». Заявление отправляется в систему и далее попадает в виде списка персоналу из комиссии по верификации заявлений для дальнейшей обработки и рассмотрения. Данные этапы представлены на рисунке 17.

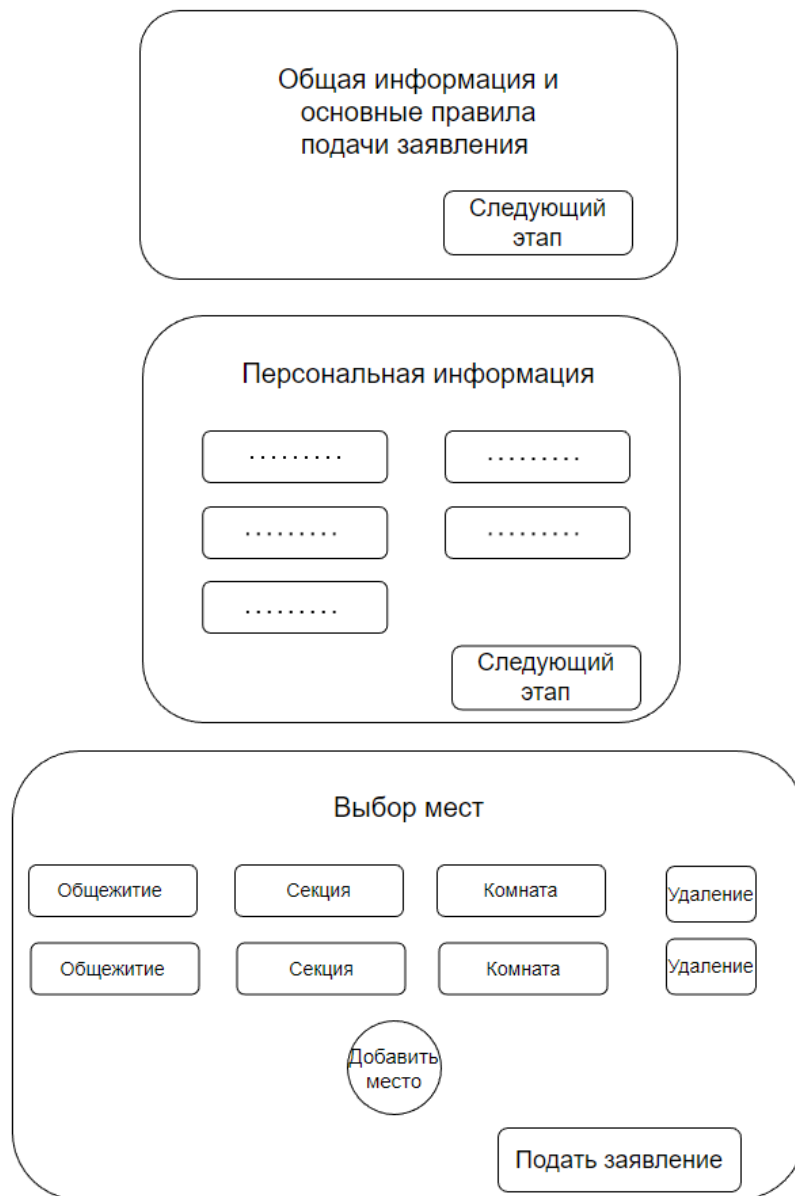


Рисунок 17 – Эскизный вид диалогового окна с выявленными этапами

Часть для комиссии для проверки должна делиться на две вида так, как все кроме одного участника должны просто одобрять или отклонять заявления с описанием причины, т.е. оставлять комментарий о своем решении.

Вид для одобрения или отклонения будет выглядеть в виде большого списка заявлений в рамках общежитий или всех без исключений, также

функциональная кнопка, описывающая вердикт для заявления.

Основной и главный интерфейс должен присутствовать у пользователя с ролью «Зам. декана по ВР». Он должен выглядеть таким образом чтобы удобно взаимодействовать с структурой общежития, заявлениями и непосредственно самого распределения. Данный эскизный вид представлен на рисунке 18.

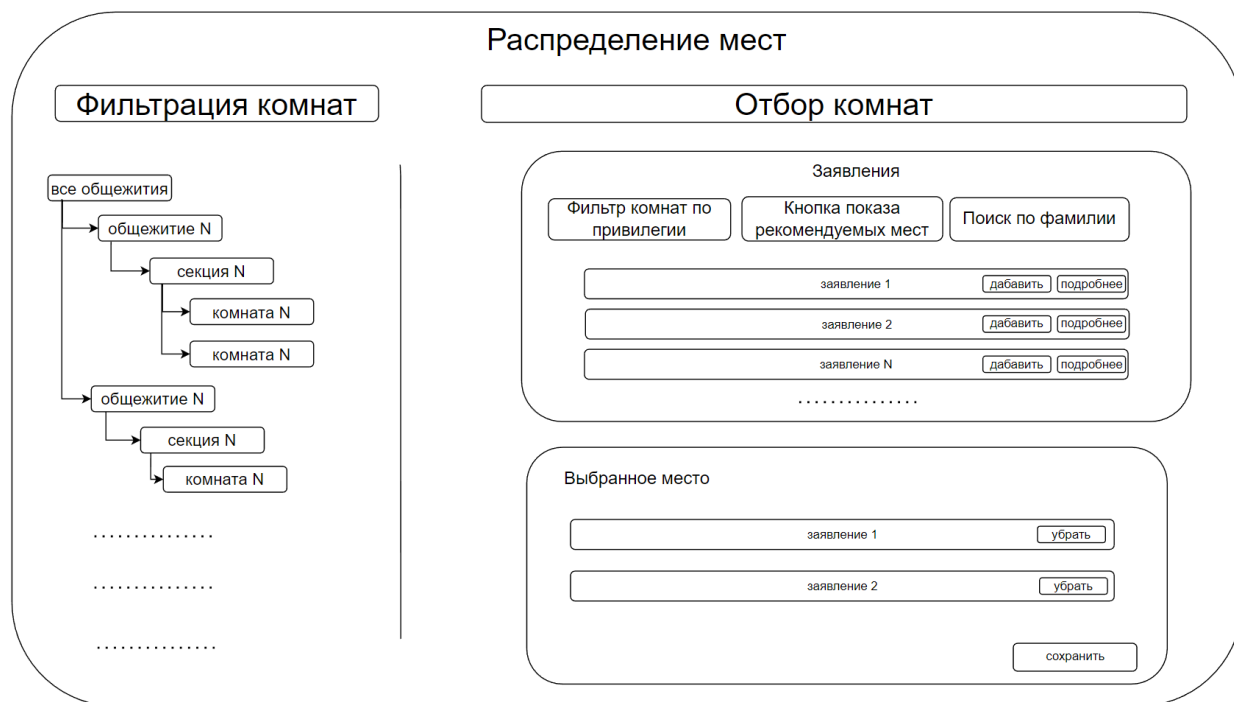


Рисунок 18 – Эскизный вид диалогового окна с выявленными этапами

После распределения существует интерфейс показывающий список мест в секционной форме с уже зафиксированными людьми в них.

В ходе проработки пользовательского интерфейса были созданы его эскизы, которые представляют собой визуальное представление будущего пользовательского опыта. Создание эскизов интерфейса является важным шагом в процессе разработки, поскольку позволяет представить и протестировать различные концепции и дизайн-решения до фактической реализации.

2.8 Анализ системных требований

Программа размещена на сервере вуза и доступна непосредственно из меню личного кабинета АмГУ под названием «Общежитие» и имеет доступ по адресу: https://cabinet.amursu.ru/statement_hostel.

Для обеспечения эффективной работы системы при одновременной работе 100 пользователей необходимо иметь сервер со следующими техническими характеристиками:

Процессор:

- Количество ядер: 8 или более.
- Частота процессора: 2.5 ГГц или выше.
- Кэш-память: 20 МБ или больше.

Оперативная память: 16 ГБ или больше.

RAID: 0+1 с объемом хранилища 500 ГБ или больше.

Операционная система: Linux версии 20.04 или новее.

Дополнительное ПО:

- Ruby: версия 3.0.1 и выше.
- Rails: версия 7.0.2.5 и выше.
- Vuetify: версия 2.
- Semantic-UI: версия 2.4.1.0.

Кроме того, сервер должен иметь надежное соединение с Интернетом, пропускной способностью канала не менее 1000/1000 МБ/с. Также требуется поддержка протоколов DNS и DHCP, а также связь с системой 1С Университет.

3 ПРОГРАМНАЯ РЕАЛИЗАЦИЯ

3.1 Проектирование базы данных

3.1.1 Обоснование выбора СУБД и инструментария разработки базы данных

Были рассмотрены несколько самых популярных систем управления базами данных для веб-приложений включая PostgreSQL, MySQL и MongoDB.

Краткое описание выдвинутых кандидатов на внедрение в разработку:

– MySQL является одной из наиболее популярных СУБД веб-приложений. Она отличается своей простотой использования, высокой производительностью и широкой поддержкой. СУБД обеспечивает широкий набор функций и хорошо масштабируется.

– PostgreSQL также является очень популярной СУБД для веб-приложений. Она предлагает богатый набор функций, поддержку сложных запросов и транзакций, а также хорошую масштабируемость. PostgreSQL часто используется для разработки больших и сложных проектов.

– MongoDB является документоориентированной NoSQL СУБД и пользуется большой популярностью в веб-приложениях. Она предоставляет гибкую схему данных и хранит информацию в формате BSON (расширение JSON). MongoDB хорошо подходит для работы с большими объемами неструктурированных данных.

Далее необходимо провести оценку и анализ средств управления базой данных для того, чтобы выделить наиболее подходящее. Результаты анализа представлены в таблице 4.

Таблица 4 – Анализ СУБД

Ключевые параметры для оценки	MySQL	PostgreSQL	MongoDB
1	2	3	4
Скорость обработки	8	7	6
Масштабируемость	9	8	9
Гибкость схемы данных	7	9	10

1	2	3	4
Сложность запросов	7	9	6
Поддержка транзакций	8	10	4
Безопасность данных:	8	9	6
Удобство использования	9	8	7
Экосистема и поддержка сообщества	9	9	8
Итоговый результат	65	69	56

Исходя из анализа и оценки ключевых параметров, а также дополнительных факторов и требований исходящих из цели проектирования и в целом разработки, можно определить фаворита – PostgreSQL.

СУБД PostgreSQL [6]. Это объектно-реляционная система управления базами данных, разработанная на факультете компьютерных наук Калифорнийского университета в Беркли. POSTGRES стал пионером многих концепций, которые стали доступны в некоторых коммерческих системах баз данных намного позже. Он поддерживает большую часть стандарта SQL и предлагает множество современных функций [12]:

- сложные запросы;
- внешние ключи;
- триггеры;
- обновляемые представления;
- целостность транзакций;
- управление многоверсионным параллелизмом.

Взаимодействие с PostgreSQL в Rails обеспечивается с помощью интегрированного ORM фреймворка ActiveRecord. ORM позволяет вам работать с базой данных через объектно-ориентированный подход, что упрощает и ускоряет разработку.

Основные концепции и методы взаимодействия с PostgreSQL в Rails состоят из:

- настройки соединения с базой данных в файле конфигурации, где

настраивается соединение с PostgreSQL. Как правило указываются параметры, такие как имя базы данных, имя пользователя, пароль и хост;

- создания моделей, которые в свою очередь представляют таблицы в базе данных. Модель обычно наследуется от класса ApplicationRecord и содержит ассоциации и валидации данных;

- миграций используемые для создания и изменения структуры базы данных. Обычно создается миграция, содержащая таблицы, столбцы с описанием типа и ограничений, также при необходимости можно создать изменения в методе, а затем применить миграцию. Миграции автоматически создают таблицы, индексы, ограничения и другие объекты базы данных;

- запросов к базе данных при помощи ActiveRecord, который предоставляет множество методов для выполнения запросов к базе данных. Можно использовать методы такие как «find», «where», «create», «update», «destroy» и другие для получения данных, фильтрации, создания, обновления и удаления записей. ActiveRecord автоматически генерирует SQL-запросы на основе методов моделей, что довольно сильно упрощает работу с БД;

- ActiveRecord обеспечивающий механизм ассоциаций между моделями. Он позволяет определить связи такого типа как «один-к-одному», «один-ко-многим» и «многие-ко-многим»;

- запросов с использованием чистого SQL. При необходимости существует возможности написания сложного запроса на языке запросов;

- PostgreSQL предоставляет различные инструменты для оптимизации производительности базы данных, такие как создание индексов и оптимизация запросов.

Взаимодействие веб-приложения с сервером осуществляется с помощью REST API – код, который позволяет должным образом обмениваться данными с сервером. Основные принципы такого способа взаимодействия является:

- отделения клиента от сервера (Client-Server);
- единство интерфейса (Uniform Interface);
- кэшируемость (Cacheable).

Схема клиент серверной архитектуры представлена на рисунке 19.

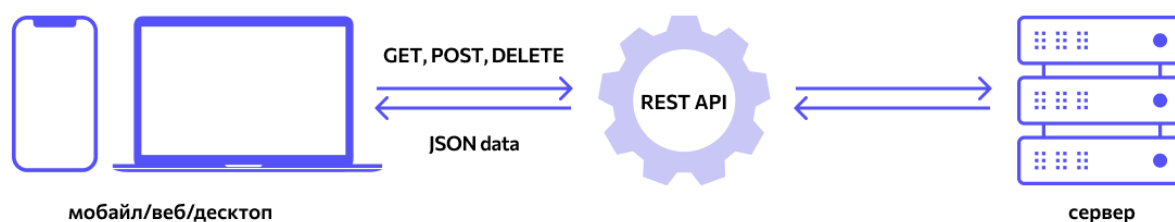


Рисунок 19 – Клиент серверная архитектура

Данной разработке необходимо использовать клиент-серверную архитектуру так, как взаимодействие происходит между клиентом и сервером. Клиент обращается с запросами к серверу. Роль клиента для сервера в том, чтобы сообщить серверу, что нужно сделать с данными, которые хранятся в базе, или с данными, которые он передаёт. Сервер принимает запросы от клиента. Его роль в том, чтобы сохранять информацию от клиента в базе данных, обрабатывать её и предоставлять к ней доступ по некоторым правилам. Такие правила называются бизнес-логикой.

3.1.2 Инфологическое проектирование базы данных

Инфологическое проектирование представляет собой процесс разработки модели данных, которая описывает информационные объекты и их взаимосвязи в определенной области. Это первый шаг в создании базы данных, где основное внимание уделяется логическому анализу и описанию информации, исключая конкретные технические детали.

Основные этапы инфологического проектирования включают:

- Анализ предметной области. Начало процесса инфологического проектирования связано с изучением области, для которой создается система. В этом этапе собираются и анализируются требования, изучаются бизнес-процессы и определяются ключевые элементы информации, их атрибуты и связи между ними;

- Создание концептуальной модели данных. На основе анализа предметной области разрабатывается абстрактная модель данных, которая описывает информационные объекты и их взаимосвязи. Для визуализации используются

графические инструменты, например, диаграммы сущность-связь или UML-диаграммы классов;

– Нормализация данных. На этом этапе осуществляется нормализация данных с целью устранения избыточности и противоречий в модели. Нормализация позволяет разделить модель на более мелкие, логически связанные компоненты, что повышает эффективность хранения и обработки данных;

– Определение связей. В этом шаге определяются типы связей между сущностями (один-к-одному, один-ко-многим, многие-ко-многим) и атрибуты сущностей. Также устанавливаются ограничения целостности данных;

– Проверка модели данных. После создания инфологической модели данных следует проверить ее соответствие требованиям и правильность построения. Это включает анализ на наличие ошибок, отсутствие противоречий и корректное представление данных.

Инфологическое проектирование не зависит от конкретных технологий хранения данных или платформы. Этот процесс является основой для физического проектирования базы данных, включающего выбор конкретной системы управления базами данных (СУБД), оптимизацию хранения данных и создание физической структуры базы данных.

Были выявлены следующие сущности:

– «Пользователи»: представляет информацию о зарегистрированных пользователях системы;

– «Заявления на общежитие»: содержит данные о заявках студентов на проживание в общежитии, включая необходимые документы;

– «Флюорографии»: включает информацию о результате медицинского обследования студентов с использованием флюорографии;

– «Комментарии»: Сущность, хранящая комментарии для заявлений, написанные пользователем;

– «Места общежитий для факультета»: содержит информацию о доступных местах в общежитии, специфичных для определенного факультета;

– «Отделы»: включает данные о различных отделах или подразделениях

в учебном учреждении;

– «Учебные планы»: содержит информацию о содержании и структуре учебных планов для разных образовательных программ;

– «Группы обучения»: представляет данные о группах студентов, которые обучаются вместе в рамках определенной программы или курса;

– «Уровни обучения»: включает информацию о разных уровнях обучения, таких как бакалавриат, магистратура или аспирантура;

– «Образовательные программы»: содержит данные о различных образовательных программах, предлагаемых учебным учреждением, включая их описание, требования и учебные планы. Все последующие спецификации атрибутов для выдвинутых сущностей представлены с 5 по 14 таблицу.

Таблица 5 – Спецификация атрибутов сущности «Пользователи»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
Идентификатор пользователя	Число, однозначно определяющее пользователя	integer	>0	1111
Идентификатор пользователя ИС	Число, однозначно определяющее пользователя в системе ИС Университет	string	1d00000001–id99999999	id1111
Имя	Имя пользователя	string	-	Рома
Фамилия	Фамилия пользователя	string	-	Махов
Отчество	Отчество пользователя	string	-	Евгеньевич
Пол	Пол	string	Мужской, женский	Мужской
Место в общежитии	Текущее занимаемое место в общежитии	json	-	Общежитие №1, секция 2, комната 2/2
Мобильный телефон	Мобильный номер, указанный при поступлении	string	-	89247778856
Сем. полож.	Семейное положение	string	-	Холост
Регистрационный адрес	Адрес регистрации	json	-	Прописка г. Благовещенск
Создание записи	Время создания записи	date	≤ текущее время	21.04.2023

1	2	3	4	5
Обновление записи	Время обновление записи	date	≤ текущее время	25.04.2023

Таблица 6 – Спецификация атрибутов сущности «Заявления на общежитие»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
Идентификатор заявления	Число, присвоенное заявлению для его идентификации	integer	>0	1111
Место в общежитии	Информация о поданных местах в общежитие	json	-	– Общежитие №1, секция 2, комната 2/2 – Общежитие №1, секция 3, комната 3/2
РАУ	Рейтинг академической успеваемости	string	-	4.8
Подтверждение начальника УСБ	Подтверждение от участника комиссии по проверке заявления	integer	От 0 до 2	1
Подтверждение коменданта общежития	Подтверждение от участника комиссии по проверке заявления	integer	От 0 до 2	1
Подтверждение директора студгородка	Подтверждение от участника комиссии по проверке заявления	integer	От 0 до 2	1
Подтверждение зам декана по ВР	Подтверждение от участника комиссии по проверке заявления	integer	От 0 до 2	1
Подтверждение Бухгалтер	Подтверждение от участника комиссии по проверке заявления	integer	От 0 до 2	1
Пол заявления	Пол, указанный в заявлении	string	Мужской, женский	Мужской
Мобильный телефон	Текущий мобильный телефон на момент подачи заявления	string	-	89247778856
Привилегия	Отличительный признак подающего	string	–	Дети-инвалиды
Результирующее место	Однозначно определенное место для студента в общежитии	json	-	Общежитие №1, секция 2, комната 2/2

Продолжение таблицы 6

1	2	3	4	5
Полное подтверждение	Полное подтверждение заявления, при котором однозначно можно определить статус заявления	integer	От 0 до 2	1
Создание записи	Время создания записи	date	≤ текущее время	21.04.2023
Обновление записи	Время обновление записи	date	≤ текущее время	25.04.2023

Таблица 7 – Спецификация атрибутов сущности «Флюорографии»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
Идентификатор флюорографии	Число, присвоенное записи флюорографии	integer	>0	1111
Дата	Дата прохождения флюорографии	date	≤ текущее время	16.02.2023
Название поликлиники	Название поликлиники, где проходил просмотр	string	-	Городская больница № 3
Статус патологий	Статус, определяющий выявлены ли при осмотре патологии	boolean	выявлены / не выявлены	Выявлены
Создание записи	Время создания записи	date	≤ текущее время	21.04.2023
Обновление записи	Время обновление записи	date	≤ текущее время	25.04.2023

Таблица 8 – Спецификация атрибутов сущности «Комментарии»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
Идентификатор комментария	Число, присвоенное комментарию	integer	>0	1111
Текст комментария	Текст комментария	text	≤ 300 символов	16.02.2023
Тип комментария	Тип комментария	string	-	Рекомендация
Создание записи	Время создания записи	date	≤ текущее время	21.04.2023
Обновление записи	Время обновление записи	date	≤ текущее время	25.04.2023

Таблица 9 – Спецификация атрибутов сущности «Места общежитий для факультета»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
Идентификатор места общежития	Число, присвоенное места общежития	integer	>0	1111
Название общежития	Название общежития	string	-	Общежитие №1
Название секции	Название поликлиники, где проходил просмотр	string	-	Секция 2
Название комнаты	Статус, определяющий выявлены ли при осмотре патологии	string	-	Комната 2/2
Идентификатор общежития 1С	Число, однозначно определяющее общежития в системе 1С Университет	string	1d00000001–id99999999	id1111
Идентификатор комнаты 1С	Число, однозначно определяющее комнаты в системе 1С Университет	string	1d00000001–id99999999	id1111
Создание записи	Время создания записи	date	≤ текущее время	21.04.2023
Обновление записи	Время обновление записи	date	≤ текущее время	25.04.2023

Таблица 10 – Спецификация атрибутов сущности «Отделы»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
Идентификатор отдела	Число, присвоенное отделу	integer	>0	1111
Идентификатор отдела в 1С	Число, однозначно определяющее отдел в системе 1С Университет	string	1d00000001–id99999999	id1111
Название отдела	Название отдела	string	-	Факультет математики и информатики
Аббревиатура	Аббревиатура отдела	string	-	ФМиИ
Адрес	Адрес, по которому располагается отдел	string	-	Амурская область, г. Благовещенск, ул. Игнатъевское шоссе, д. 21, корпус 8, каб. 208
Номер	Номер отдела	string	–	1

1	2	3	4	5
Почта	Электронный почтовый адрес	string	–	if@amursu.ru
ФИО ответственного	ФИО ответственного	string	–	Максим Данилов Андреевич
Общая информация	Информация по отделу	string	–	В настоящее время обучается более 300 студентов ...
Пост ответственного	Пост ответственного	text	–	Декан

Таблица 11 – Спецификация атрибутов сущности «Учебные планы»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
Идентификатор учебного плана	Число, присвоенное составлению студентов	integer	>0	1111
Файл	Документ описывающий учебный план	file	Формат png, docx	Учебный план.png
Название учебного плана	Название учебного плана	string	-	РУП-09.03.01.01-О-2019-2022/2023
Создание записи	Время создания записи	date	≤ текущее время	21.04.2023
Обновление записи	Время обновление записи	date	≤ текущее время	25.04.2023

Таблица 12 – Спецификация атрибутов сущности «Группы обучения»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
Идентификатор группы обучения	Число, присвоенное группе обучения	integer	>0	1111
Названия группы	Названия группы обучения	string	-	953-об
Идентификатор группы в 1С	Число, однозначно определяющее группу обучения в системе 1С Университет	string	1d00000001–id99999999	id1111

Продолжение таблицы 12

1	2	3	4	5
Создание записи	Время создания записи	date	≤ текущее время	21.04.2023
Обновление записи	Время обновление записи	date	≤ текущее время	25.04.2023

Таблица 13 – Спецификация атрибутов сущности «Уровни обучения»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
Идентификатор уровня обучения	Число, присвоенное группе обучения	integer	>0	1111
Название уровня обучения	Названия уровня обучения	string	-	Бакалавриат
Идентификатор группы в 1С	Число, однозначно определяющее группу обучения в системе 1С Университет	string	1d00000001–id99999999	id1111
Создание записи	Время создания записи	date	≤ текущее время	21.04.2023
Обновление записи	Время обновление записи	date	≤ текущее время	25.04.2023

Таблица 14 – Спецификация атрибутов сущности «Образовательные программы»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
Идентификатор образовательной программы	Число, присвоенное группе обучения	integer	>0	1111
Название образовательной программы	Названия образовательной программы	string	-	09.03.01 Информатика и вычислительная техника
Код образовательной программы	Код образовательной программы	string	–	09.03.01
Время обучения	Время обучения	string	–	4 года
Создание записи	Время создания записи	date	≤ текущее время	21.04.2023
Обновление записи	Время обновление записи	date	≤ текущее время	25.04.2023

В заключении инфологического проектирования необходимо объединить спецификации в одну модель сущность-связь используя любую из нотаций: Мартина, Чена или IDEF1X.

Для построения модели БД в данной работе была выбрана нотация IDEF1X, она предоставляет удобный и понятный способ описания структуры базы данных, позволяя легко визуализировать и описать сущности, связи и атрибуты данных. ER-диаграмма показана на рисунке 20.

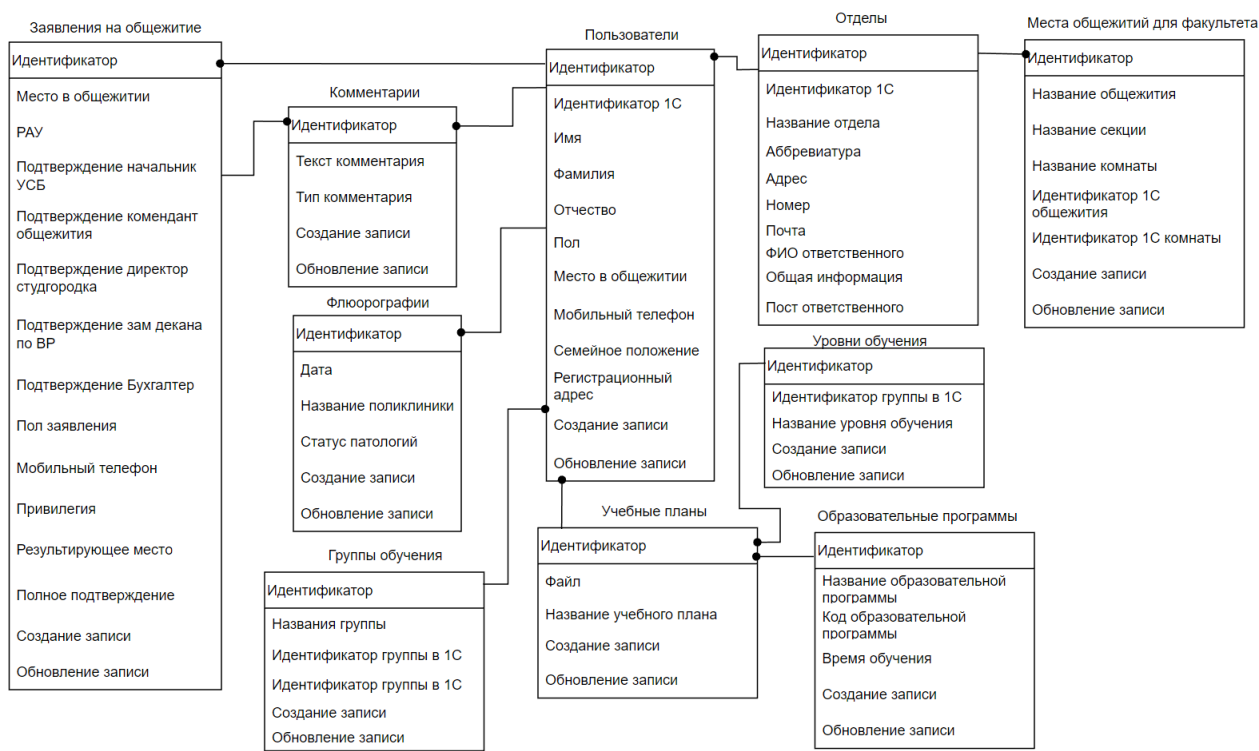


Рисунок 20 – ER-диаграмма в нотации IDEF1X

3.1.3 Логическое проектирование базы данных

Логическое проектирование – это процесс разработки структуры и организации данных в информационной системе, чтобы обеспечить эффективное хранение, доступ и обработку информации. Оно основывается на анализе требований и моделировании данных, чтобы создать логическую модель, которая описывает структуру и связи между сущностями.

В ходе логического проектирования определяются сущности, атрибуты и связи между ними на основе анализа предметной области. Сущности представляют реальные или концептуальные объекты, а атрибуты определяют

характеристики этих объектов. Связи определяют отношения между сущностями и определяются их типы (один-к-одному, один-ко-многим, многие-ко-многим).

Основная цель логического проектирования – создание независимой от конкретной технологии модели данных, которая точно отражает требования предметной области и обеспечивает эффективное хранение и доступ к информации. Логическая модель данных может быть представлена в виде диаграмм сущность-связь или других графических нотаций.

Для представленных сущностей предлагается несколько типов связи модели «сущность-связь», которые будут описывать принадлежность одной таблицы к другой.

Связь «Заявления на общежитие – Комментарии». Рассматривая сущности, участвующие в связи, выявляется связь между ними. Связь «один-ко-многим», множество комментариев могут относиться к одному заявлению, оно представлено на рисунке 21.



Рисунок 21 – Связь «Заявления на общежитие – Комментарии»

Связь «Пользователи – Комментарии». Рассматривая сущности, участвующие в связи, выявляется связь между ними. Связь «один-ко-многим», один пользователь может написать множество комментариев, оно представлено на рисунке 22.



Рисунок 22 – Связь «Пользователи– Комментарии»

Связь «Пользователи – Учебные планы». Рассматривая сущности,

участвующие в связи, выявляется связь между ними. Связь «один-ко-многим», один пользователь может относиться ко многим учебным планам, оно представлено на рисунке 23.



Рисунок 23 – Связь «Пользователи – Учебные планы»

Связь «Пользователи – Флюорографии». Рассматривая сущности, участвующие в связи, выявляется связь между ними. Связь «один-ко-многим», один пользователь может добавить за все время обучения множество документов о прохождении флюорографии, оно представлено на рисунке 24.



Рисунок 24 – Связь «Пользователи – Флюорографии»

Связь «Пользователи – Группы обучения». Рассматривая сущности, участвующие в связи, выявляется связь между ними. Связь «один-ко-многим», множество пользователей может иметь одну группу обучения, оно представлено на рисунке 25.



Рисунок 25 – Связь «Пользователи – Группы обучения»

Связь «Учебные планы – Уровни обучения». Рассматривая сущности, участвующие в связи, выявляется связь между ними. Связь «один-ко-многим», множество учебных планов могут относиться к одному уровню обучения, оно представлено на рисунке 26.



Рисунок 26 – «Учебные планы – Уровни обучения»

Связь «Отделы – Места общежитий для факультета». Рассматривая сущности, участвующие в связи, выявляется связь между ними. Связь «один-ко-многим», один отдел может иметь множество мест из общежитий, оно представлено на рисунке 27.



Рисунок 27 – Связь «Отделы – Места общежитий для факультета»

Связь «Отделы – Места общежитий для факультета». Рассматривая сущности, участвующие в связи, выявляется связь между ними. Связь «один-ко-многим», одна образовательная программа может содержать множество учебных планов, оно представлено на рисунке 28.



Рисунок 28 – Связь «Заявления на общежитие – Комментарии»

После определения всех отношений между сущностями, необходимо убедиться в соответствии каждой из таблиц третьей нормальной форме.

Таблицы находятся в первой нормальной форме (1NF), то есть каждая ячейка содержит только одно значение, также они соответствуют второй нормальной форме (2NF) так, как каждый не ключевой столбец зависит только от первичного ключа, что и было приказано на ранее представленных рисунках.

Далее убеждаемся в том, что таблицы находятся в третьей нормальной форме (3NF), то есть каждый не ключевой столбец зависит только от первичного ключа или других не ключевых столбцов, но не зависит от других не ключевых столбцов. Если есть столбцы, зависящие от других не ключевых столбцов, они также должны быть разделены на отдельные таблицы.

Результатом логического проектирования и нормализации отношений является логическая модель, представленная на рисунке 29.

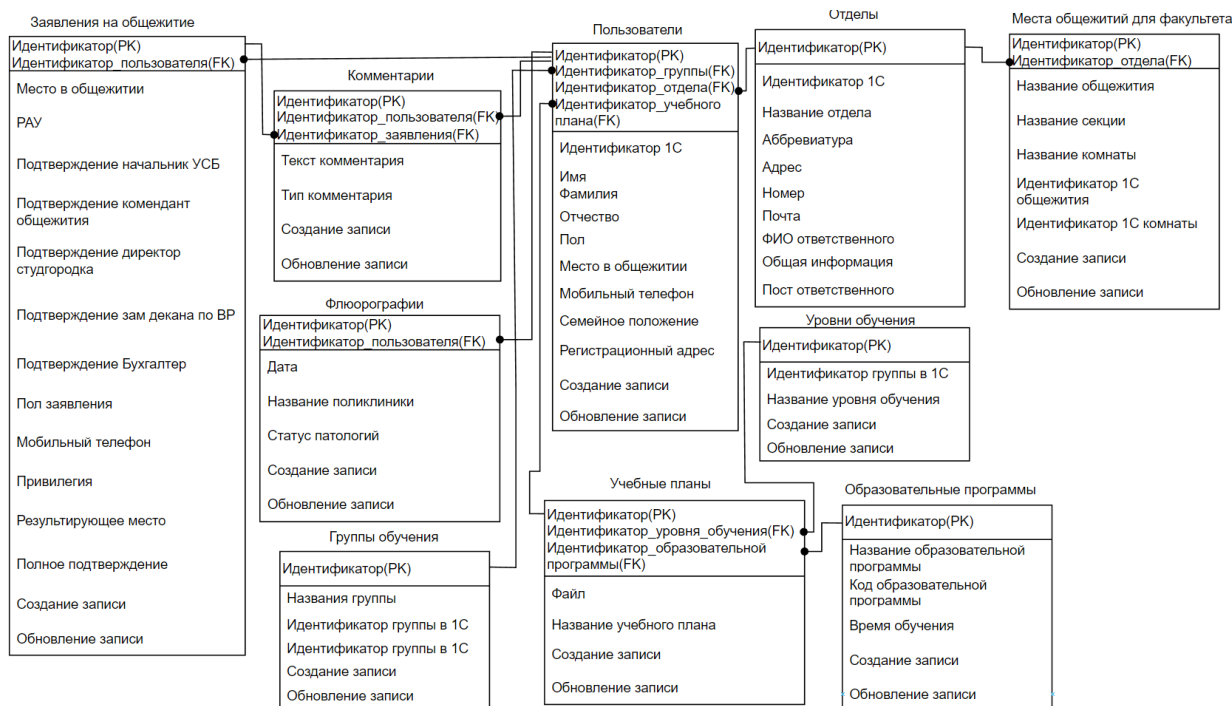


Рисунок 29 – Логическая модель БД

3.1.4 Физическое проектирование базы данных

После завершения логического проектирования БД, а точнее описание сущностей, их атрибутов, связей и определение ограничений целостности и требований к данным, можно приступить к физическому проектированию.

Суть физического проектирования – преобразование логической модели в конкретную реализацию базы данных, включая выбор конкретных типов данных, определение индексов, физических структур таблиц, оптимизацию производительности и т.д.

Далее представлены физические структуры сущностей с 15 по 24 таблицу.

Таблица 15 – Физическая структура данных сущности «Пользователи»

Название атрибута	Условия	Тип данных	Формат данных	Индексация
1	2	3	4	5
Идентификатор пользователя	>0	Число	bigint	Primary key
Идентификатор группы	>0	Число	bigint	Foreign key
Идентификатор отдела	>0	Число	bigint	Foreign key
Идентификатор учебного плана	>0	Число	bigint	Foreign key
Идентификатор пользователя IC	Не пустое	Строка	string	–
Имя	Не пустое	Строка	string	–
Фамилия	Не пустое	Строка	string	–
Отчество	Не пустое	Строка	string	–
Пол	Не пустое	Строка	string	–
Место в общежитии	Не пустое	Объект	json	–
Мобильный телефон	Не пустое	Строка	string	–
Семейное положение	Не пустое	Строка	string	–
Регистрационный адрес	Не пустое	Объект	json	–
Создание записи	≤ текущее время	Дата	date	–
Обновление записи	≤ текущее время	Дата	date	–

Таблица 16 – Физическая структура данных сущности «Заявления на общежитие»

Название атрибута	Условия	Тип данных	Формат данных	Индексация
1	2	3	4	5
Идентификатор заявления	>0	Число	bigint	Primary key
Идентификатор пользователя	>0	Число	bigint	Foreign key
Место в общежитии	Не пустое	Объект	json	–
РАУ	Не пустое	Строка	string	–
Подтверждение УСБ	≥ 0 и ≤ 2	Число	integer	–
Подтверждение комендант общежития	≥ 0 и ≤ 2	Число	integer	–
Подтверждение директор студгородка	≥ 0 и ≤ 2	Число	integer	–
Подтверждение зам декана по ВР	≥ 0 и ≤ 2	Число	integer	–
Подтверждение Бухгалтер	≥ 0 и ≤ 2	Число	integer	–
Пол заявления	Не пустое	Строка	string	–
Мобильный телефон	Не пустое	Строка	string	–

Продолжение таблицы 16

1	2	3	4	5
Привилегия	Не пустое	Строка	string	–
Результирующее место	Не пустое	Объект	json	–
Полное подтверждение	true или false	true или false	boolean	–
Создание записи	\leq текущее время	Дата	date	–
Обновление записи	\leq текущее время	Дата	date	–

Таблица 17 – Физическая структура данных сущности «Флюорографии»

Название атрибута	Условия	Тип данных	Формат данных	Индексация
1	2	3	4	5
Идентификатор флюорографии	>0	Число	bigint	Primary key
Идентификатор пользователя	>0	Число	bigint	Foreign key
Дата	\leq текущее время	Дата	date	–
Название поликлиники	Не пустое	Строка	string	–
Статус патологий	true или false	true или false	boolean	–
Создание записи	\leq текущее время	Дата	date	–
Обновление записи	\leq текущее время	Дата	date	–

Таблица 18 – Физическая структура данных сущности «Комментарии»

Название атрибута	Условия	Тип данных	Формат данных	Индексация
1	2	3	4	5
Идентификатор комментария	>0	Число	bigint	Primary key
Идентификатор пользователя	>0	Число	bigint	Foreign key
Идентификатор заявления	>0	Число	bigint	Foreign key
Текст комментария	≤ 300 символов	Текст	text	–
Тип комментария	Не пустое	Строка	string	–
Создание записи	\leq текущее время	Дата	date	–
Обновление записи	\leq текущее время	Дата	date	–

Таблица 19 – Физическая структура данных сущности «Места общежитий для факультета»

Название атрибута	Условия	Тип данных	Формат данных	Индексация
1	2	3	4	5
Идентификатор места общежития	>0	Число	bigint	Primary key
Идентификатор отдела	>0	Число	bigint	Foreign key
Название общежития	Не пустое	Строка	string	–
Название секции	Не пустое	Строка	string	–
Название комнаты	Не пустое	Строка	string	–
Идентификатор общежития 1С	1d00000001–id99999999	Строка	string	–
Идентификатор комнаты 1С	1d00000001–id99999999	Строка	string	–
Создание записи	≤ текущее время	Дата	date	–
Обновление записи	≤ текущее время	Дата	date	–

Таблица 20 – Физическая структура данных сущности «Отделы»

Название атрибута	Условия	Тип данных	Формат данных	Индексация
1	2	3	4	5
Идентификатор отдела	>0	Число	bigint	Primary key
Идентификатор отдела в 1С	1d00000001–id99999999	Строка	string	–
Название отдела	–	Строка	string	–
Аббревиатура	–	Строка	string	–
Адрес	–	Строка	string	–
Номер	–	Строка	string	–
Почта	–	Строка	string	–
ФИО ответственного	–	Строка	string	–
Общая информация	–	Строка	string	–
Пост ответственного	–	Текст	text	–

Таблица 21 – Физическая структура данных сущности «Учебные планы»

Название атрибута	Условия	Тип данных	Формат данных	Индексация
1	2	3	4	5
Идентификатор учебного плана	>0	Число	bigint	Primary key
Идентификатор уровня обучения	>0	Число	bigint	Foreign key
Идентификатор образовательной программы	>0	Число	bigint	Foreign key

Продолжение таблицы 21

1	2	3	4	5
Файл	–	Файл	file	–
Название учебного плана	–	Строка	string	–
Создание записи	≤ текущее время	Дата	date	–
Обновление записи	≤ текущее время	Дата	date	–

Таблица 22 – Физическая структура данных сущности «Группы обучения»

Название атрибута	Условия	Тип данных	Формат данных	Индексация
1	2	3	4	5
Идентификатор группы обучения	>0	Число	bigint	Primary key
Названия группы	Не пустое	Строка	string	–
Идентификатор группы в 1С	1d00000001–id99999999	Строка	string	–
Создание записи	≤ текущее время	Дата	date	–
Обновление записи	≤ текущее время	Дата	date	–

Таблица 23 – Физическая структура данных сущности «Уровни обучения»

Название атрибута	Условия	Тип данных	Формат данных	Индексация
1	2	3	4	5
Идентификатор уровня обучения	>0	Число	bigint	Primary key
Название уровня обучения	Не пустое	Строка	string	–
Идентификатор группы в 1С	1d00000001–id99999999	Строка	string	–
Создание записи	≤ текущее время	Дата	date	–
Обновление записи	≤ текущее время	Дата	date	–

Таблица 24 – Физическая структура данных сущности «Образовательные программы»

Название атрибута	Условия	Тип данных	Формат данных	Индексация
1	2	3	4	5
Идентификатор образовательной программы	>0	Число	bigint	Primary key
Название образовательной программы	Не пустое	Строка	string	–

1	2	3	4	5
Код образовательной программы	–	Строка	string	–
Время обучения	–	Строка	string	–
Создание записи	≤ текущее время	Дата	date	–
Обновление записи	≤ текущее время	Дата	date	–

Таким образом было проведено физическое проектирование базы данных, выявлены для всех атрибутов типы данных их ограничения и форматы данных, результирующий вид физической структуры БД представлен на рисунке 30.

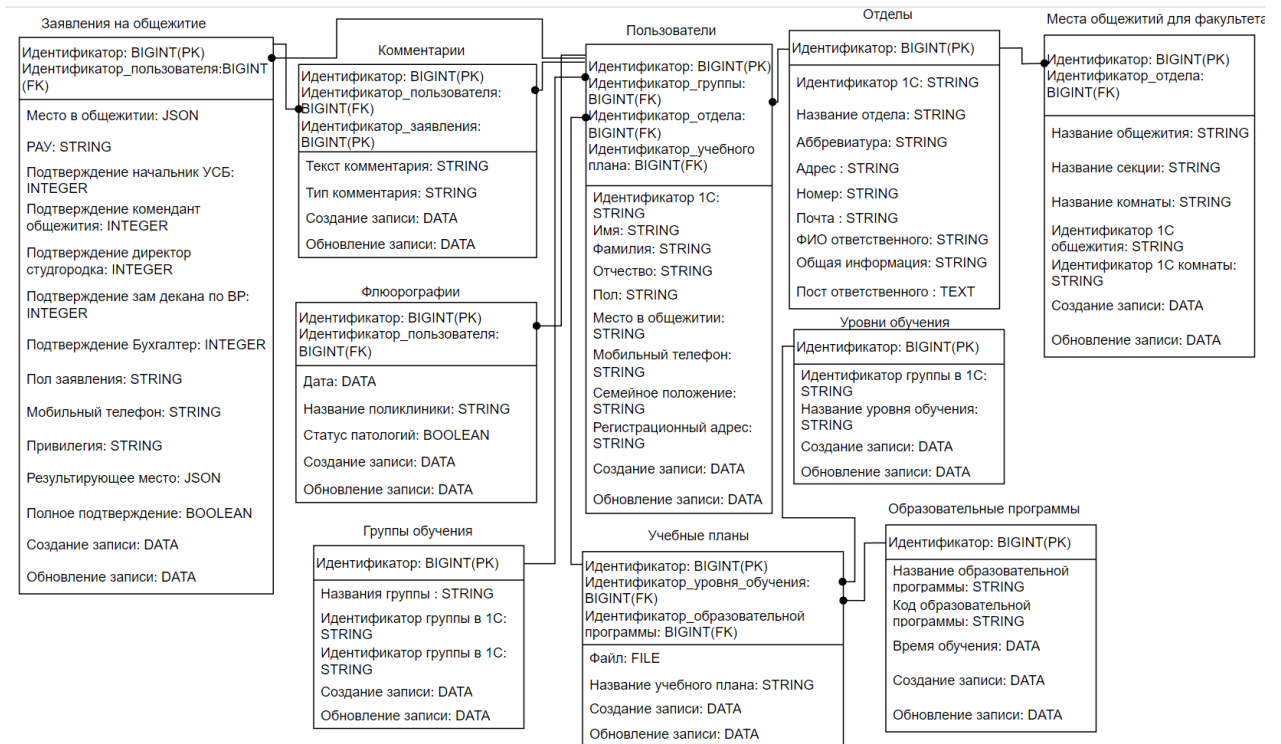


Рисунок 30 – Физическая структура базы данных

3.2 Создание модуля «Общежитие»

3.2.1 Создание функциональной части для студентов

При входе в раздел «Общежитие», пользователь видит главное окно, которое может иметь стандартный вид или же вид исходя из его ограничений или недостающей информации.

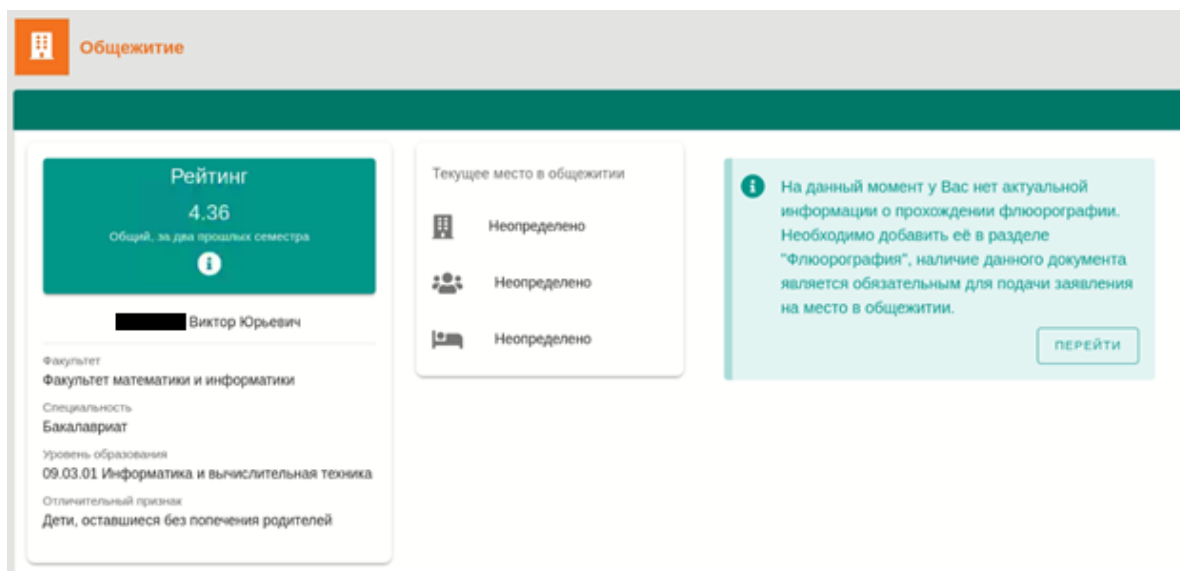


Рисунок 31 – Вид главного окна, отсутствует флюорография

Зачастую возникнет ситуация, когда необходимо обновить информацию о прохождении флюорографии, это является одним из главных пунктов при котором подача заявлений возможна. Главное окно для пользователя, который не обновил информацию о флюорографию изображено на рисунке 31.

Далее пользователь нажимает на кнопку «Перейти» и заполняет все актуальные данные о прохождении флюорографии, после чего главное окно принимает вид, показанный на рисунке 32.

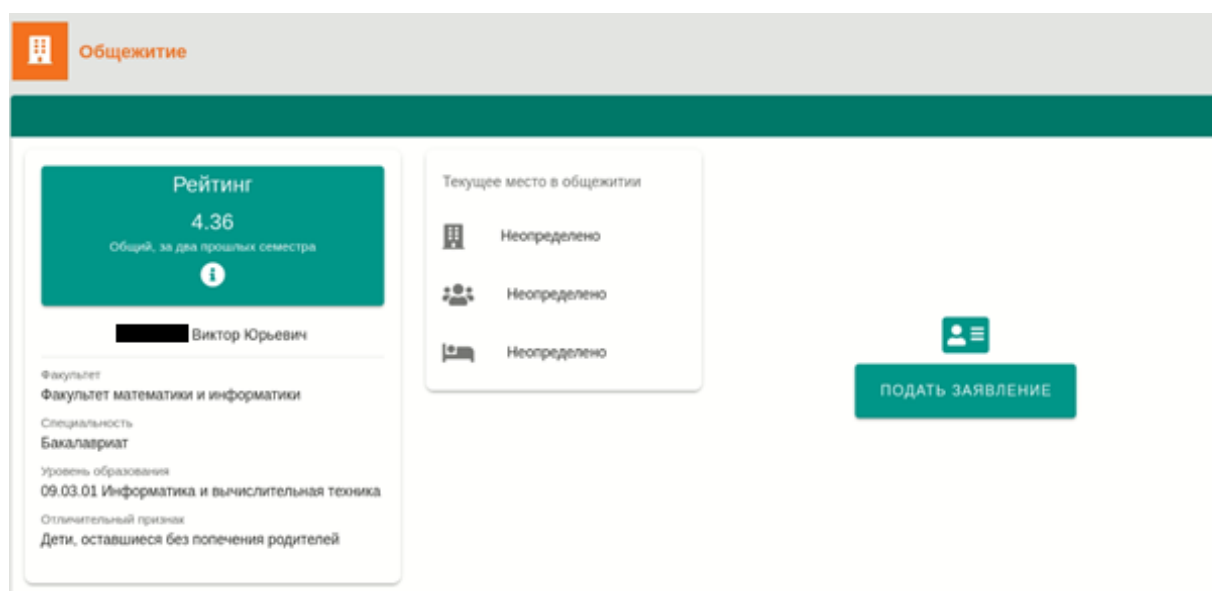


Рисунок 32 – Вид главного окна, отсутствуют ограничения

Но может и возникнуть ситуация, когда пользователь имеет прописку «Благовещенского района», тогда доступ для подачи заявления закрыт, вид

данной ошибки представлен на рисунке 33. Еще одним из главных требований для подачи заявления является иногородняя прописка потому, что только такие пользователи могут пользоваться системой.

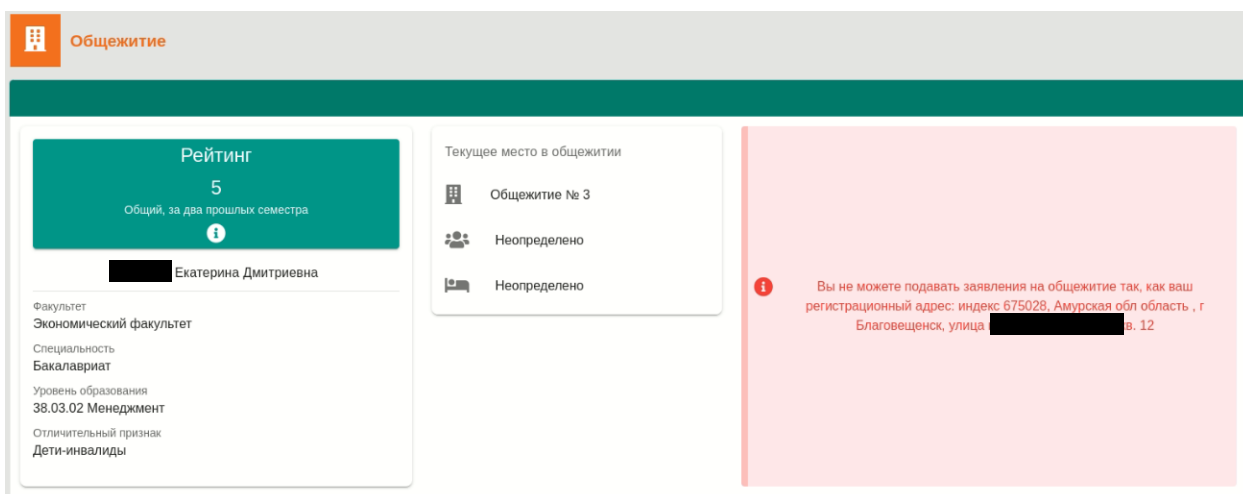


Рисунок 33 – Вид главного окна, ограничение по прописке

На ряду с этим пользователь может просмотреть свой рейтинг академической успеваемости и его непосредственно расшифровку за два последних семестра, вид всплывающего окна показан на рисунке 34.

Расшифровка РАУ	
Седьмой семестр, Восьмой семестр	
Дисциплина	Оценка
Безопасность жизнедеятельности	Отлично
Верификация программного обеспечения	Отлично
Инженерно-техническая защита информации	Удовлетворительно
Информационная безопасность	Отлично

Рисунок 34 – Всплывающее окно, расшифровка рейтинга

Далее пользователь может подать заявление. При нажатии на кнопку «Подать заявление» всплывает окно с несколькими этапами прохождения процесса оформления заявления.

На первом этапе пользователю предоставляется общая информация о том каким образом работает системы и правила ее функционирования, данный вид отражает рисунок 35.

Поддача заявления ×

ОБЩАЯ ИНФОРМАЦИЯ
ПЕРСОНАЛЬНАЯ ИНФОРМАЦИЯ
ВЫБОР МЕСТА

ПЕРСОНАЛЬНАЯ ИНФОРМАЦИЯ

Вам необходимо проверить ваши персональные данные, указать пол и текущий номер телефона на момент подачи заявления. Остальные данные для изменения или ввода неактивны, так как они извлекаются из общей информационной базы, где хранится ваша персональная информация, заполненная в момент поступления в вуз или измененная в период обучения. Если вы заметили ошибку или неточность данных, то на первом шаге необходимо убедиться в том, что актуальная информация о вас хранится в информационной базе университета. Для этого обратитесь в деканат, к которому вы относитесь, запросите и уточните информацию о вас. Если после выполнения первого шага проблема не была решена, обратитесь за помощью непосредственно к разработчикам системы.

ВЫБОР МЕСТА

Вам предоставляется возможность выбрать до трех комнат для рассмотрения в целях заселения. Приоритет выбора комнат определяется порядком, установленным вами. Таким образом, первая выбранная комната считается наивысшим приоритетом, и с учетом последующих комнат приоритет постепенно снижается. Ваши выбранные комнаты рассматриваются как предпочтения и принимаются во внимание комиссией. Однако существует возможность, что вас могут разместить в комнату, которую вы не выбрали.

Рисунок 35 – Первый этап при подаче заявления

Далее на втором этапе необходимо проверить персональную информацию и заполнить поля с полом и актуальным телефоном, вид представлен на рисунке 36. Большинство данных автоматически извлекаются из общей информационной базе вуза о пользователях поэтому у него нет необходимости все вписывать самому.

Поддача заявления ×

ОБЩАЯ ИНФОРМАЦИЯ
ПЕРСОНАЛЬНАЯ ИНФОРМАЦИЯ
ВЫБОР МЕСТА

ФИО

████████ Виктор Юрьевич

Пол

Мужской ▼

Телефон

8924 ██████████

Семейное положение

Холост/Не замужем

Отличительные признаки

Дети, оставшиеся без попечения родителей

Специальность

09.03.01 Информатика и вычислительная техника

Факультет

Факультет математики и информатики

НАЗАД
ДАЛЕЕ

Рисунок 36 – Второй этап при подаче заявления

На следующем этапе необходимо выбрать желаемые места в общежитии. Их количество ограничивается до трех и всегда должны располагаться по выстроенному пользователем приоритете. Наивысшим приоритетом является самая первая выбранная комната и далее с их добавлением уменьшается.

Места в общежитии выводятся непосредственно по факультетно, отталкиваясь от факультета, на котором находится данный пользователь.

При максимальном выборе комнат показывается сообщение о заполненности комнатами заявления. После нажатия кнопки «Подать» поля с местами валидируются так, как они не должны быть пустыми. Вид данного этапа показан на рисунке 37.

Поддача заявления

ОБЩАЯ ИНФОРМАЦИЯ ПЕРСОНАЛЬНАЯ ИНФОРМАЦИЯ **ВЫБОР МЕСТА**

Остаться на текущем месте проживания

Общежитие Общежитие № 2	Секция 5	Комната 5/1	
Общежитие Общежитие № 2	Секция 5	Комната 5/2	
Общежитие Общежитие № 2	Секция 6	Комната 6/2	

Вы выбрали максимальное количество мест

ПОДАТЬ

Рисунок 37 – Последний этап при подаче заявления

После успешного процесса подачи заявления, оно непосредственно показывается пользователю с краткой информацией. Дополнительно пользователь может его редактировать или удалять, но если хоть один из участников комиссии начнет его рассматривать, то редактирование и удаление будет невозможным. Вид главной страницы при успешно поданном заявлении представлен на рисунке 38.

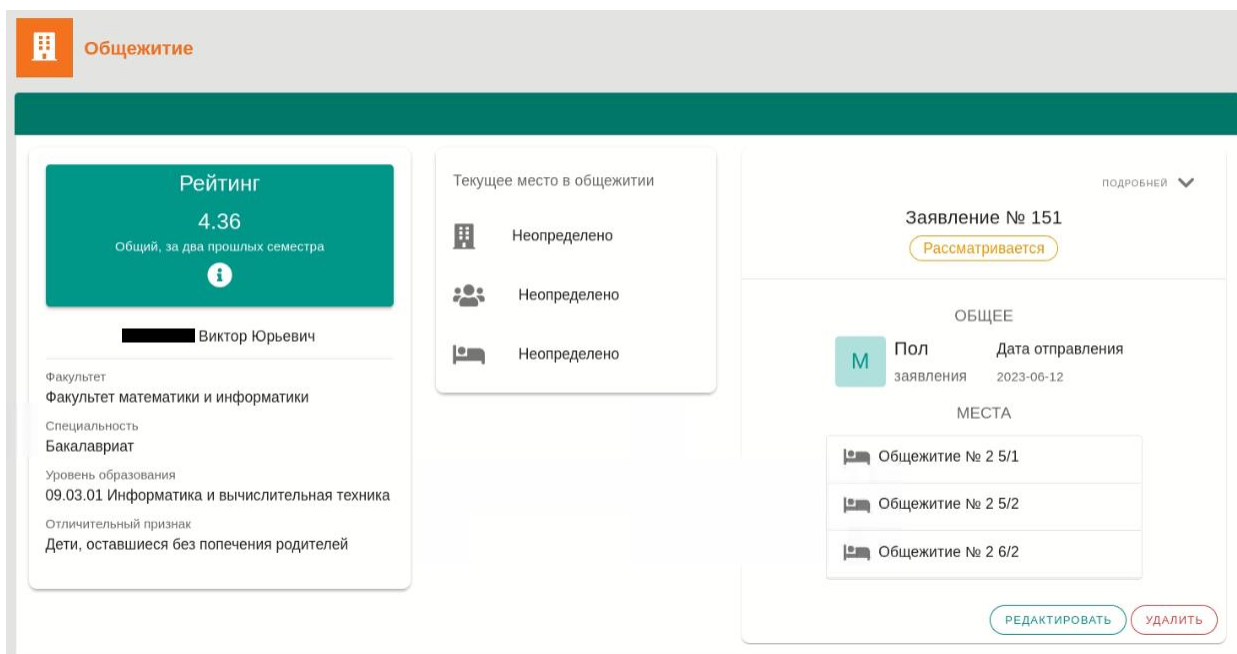


Рисунок 38 – Последний этап при подаче заявления

На ряду с этим пользователь может посмотреть подробное состояние своего заявления нажав на кнопку «Подробнее», вид данного события представлен на рисунке 39.

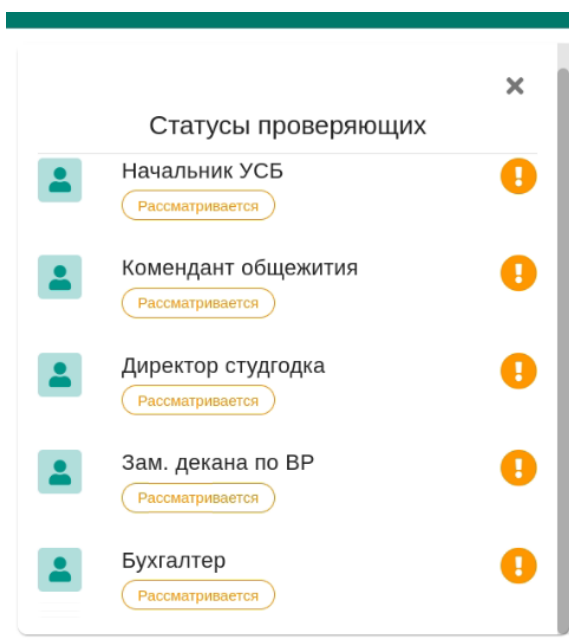


Рисунок 39 – Подробный просмотр статусов проверяющих

3.2.2 Создание функциональной части для комиссии

При авторизации пользователя «зам. декана по ВР» из комиссии проверяющих, интерфейс меняется и принимает функциональный вид, представленный на рисунке 40.

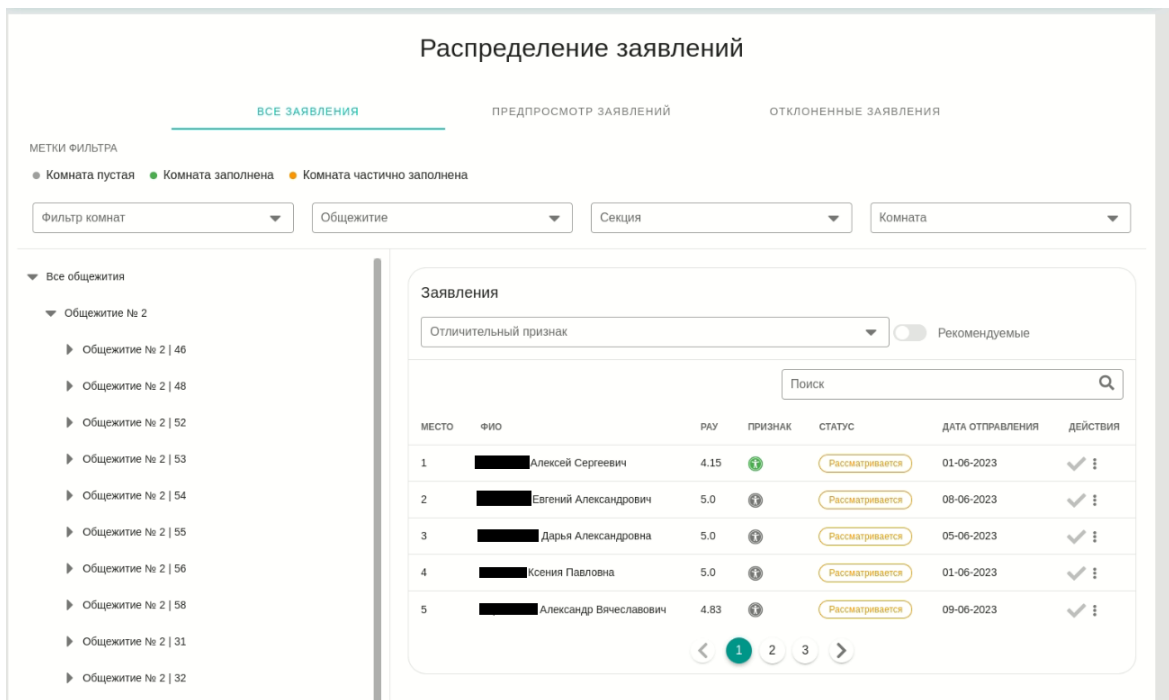


Рисунок 40 – Главный вид системы для комиссии

Данный интерфейс наполнен функциональными элементами такими как древовидный список структуры общежития, выведенный в рамках факультета, сортировка данного дерева и отбор конкретных мест. Также наряду с этим выводятся все заявления опять же в рамках факультета с возможностью сортировки данных заявлений. Древовидная структура интерактивна и можно из нее выбрать комнаты, что показывает рисунок 41.

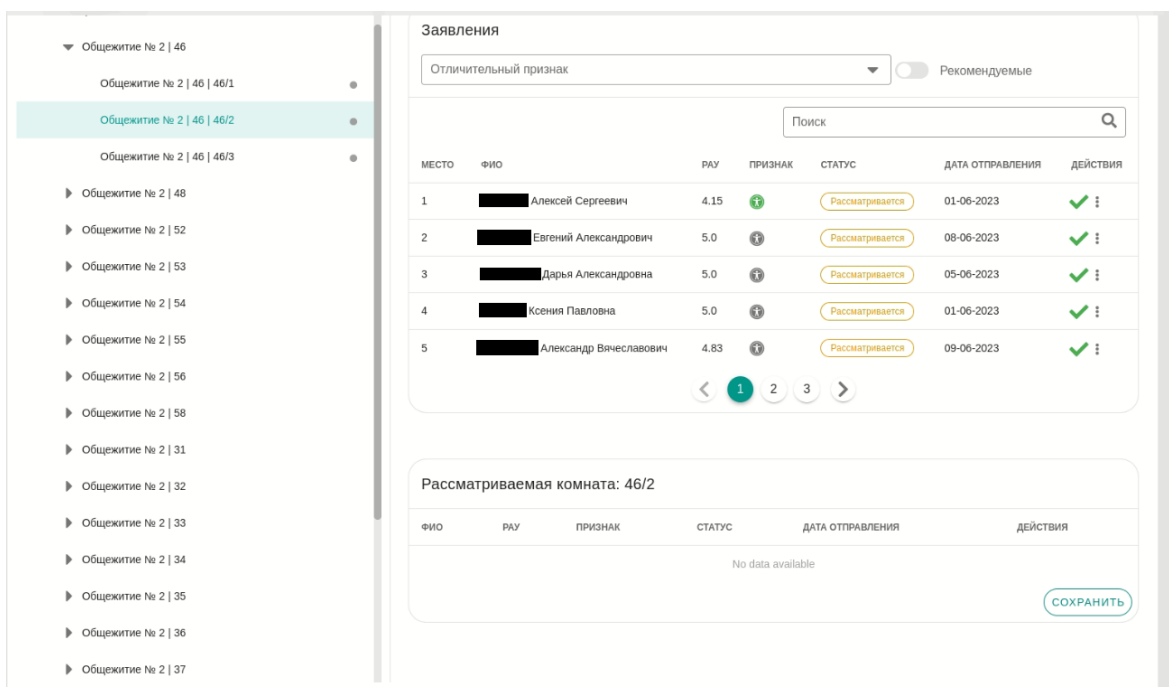


Рисунок 41 – Главный вид системы для комиссии при выборе комнаты

Существует функциональный элемент позволяющий показать на выбранную комнату рекомендованные заявления. Они строятся из рекомендации я самих пользователей чьи заявления рассматриваются.

На рисунке 42 показано как изменился список на комнату 36/2, при нажатии кнопки «Рекомендуемые».

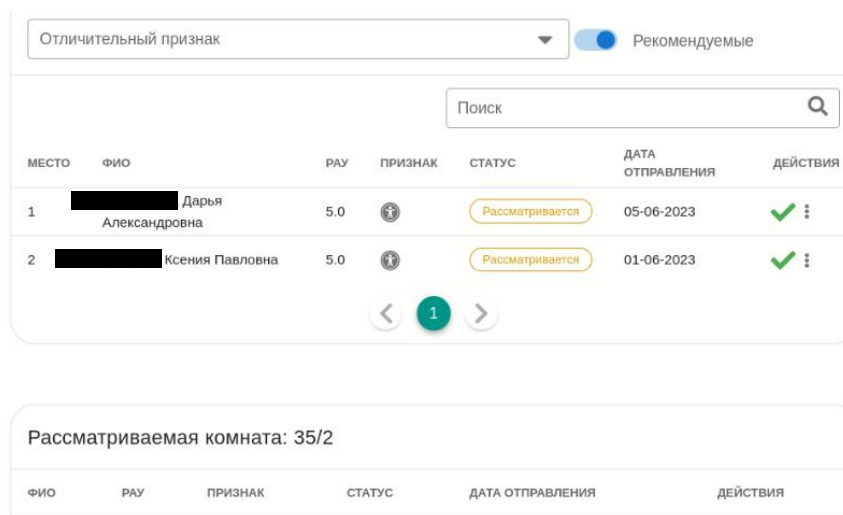


Рисунок 42 – Рекомендуемые заявления на комнату

Также присутствует сортировка по отличительному признаку, он влияет на подачу заявления, так как его наличие определяет наивысший порядок при рассмотрении списка заявлений, данный отбор представлен на рисунке 43 – 44.

Список заявлений заполняется программно по следующему правилу в виде сортированного списка, от самого высокого к самому низкому приоритету.

Для того чтобы была возможность определять на какое место проживания может претендовать студент, выявляются «категории приоритета» основывающиеся на ряде факторов, представленных в таблице 25.

Таблица 25 – Влияния различных факторов на ранжирование заявлений

Факторы	Степень важности от 1 до 3, от самого высшего до низшего
1	2
Отличительный признак (пример, дети, оставшиеся без попечительства родителей, инвалиды)	1

1	2
Рейтинг академической успеваемости (среднее значение оценок, набранных за два последних семестра)	2
Время подачи заявления	3

На основе факторов, работа алгоритма следующая: если студент имеет отличительный признак, то он имеет высокий приоритет на подаваемую комнату. При отсутствии или совпадении отличительного признака с другим человеком, учитывается рейтинг академической успеваемости (РАУ) и уже от его значения заявление занимает соответствующую позицию. Если же и РАУ совпадает с другим студентом, то сравнивается время подачи заявлений и соответственно на основании этого определяется приоритет на подаваемое место.

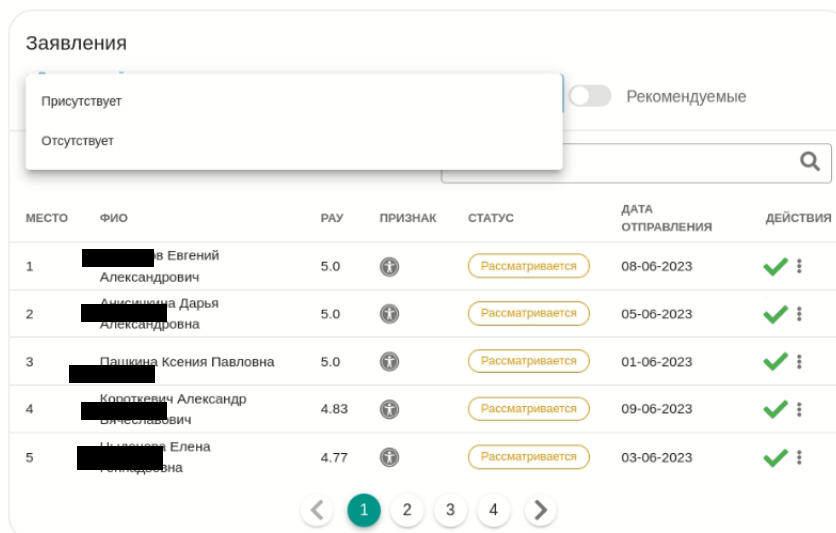


Рисунок 43 – Фильтрация по отличительному признаку

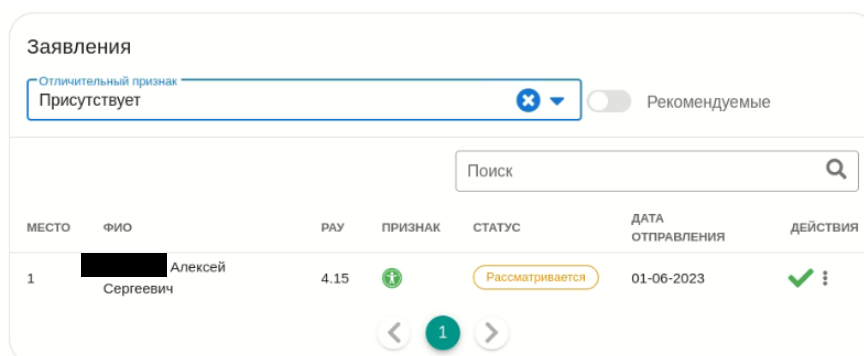


Рисунок 44 – Фильтрация по отличительному признаку

Дополнительно реализована возможность просмотра персональной информации о пользователе и выбранных им желаемых комнат, данный вид представлен на рисунке 45.

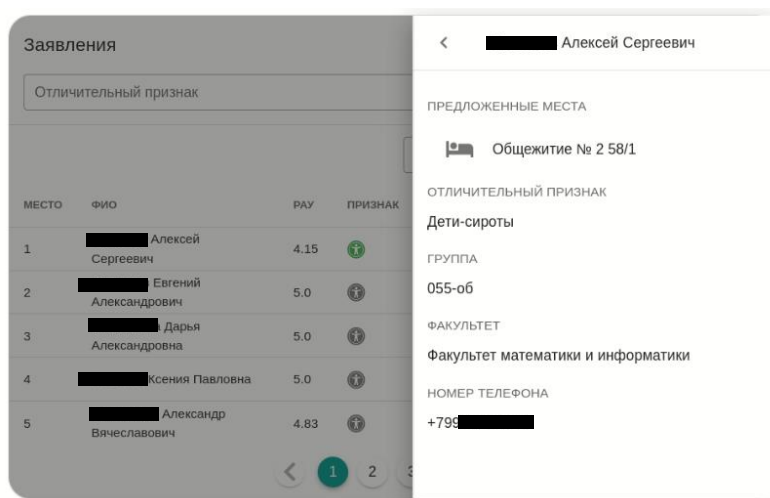


Рисунок 45 – Персональная информация о пользователе

Главный функционал данного вида пользователя заключается в распределении студентов по местам в общежитии на основе поданных ими заявлениями. Участник комиссии может интерактивно распределять студентов по местам, по мере заполненности комнаты будут подсвечиваться метками, какие полные, какие частично, а какие еще пустые. Также и само заявления при распределении меняет статус на «Одобрено» и уже не появляется в общем списке заявлений. Данный функционал показан на рисунке 46.

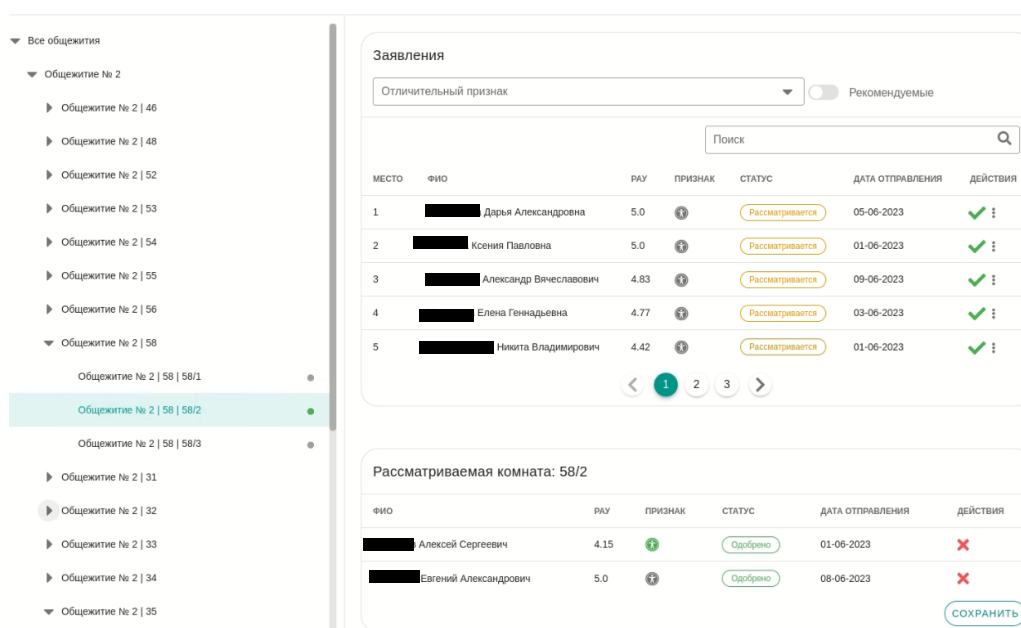


Рисунок 46 – Распределение студентов по местам

После успешного выполнения распределения предоставляется возможность просмотра общего состояния комнат, предоставленных для факультета, рисунок 47 отражает главный вид мест с назначенными для них студентами.

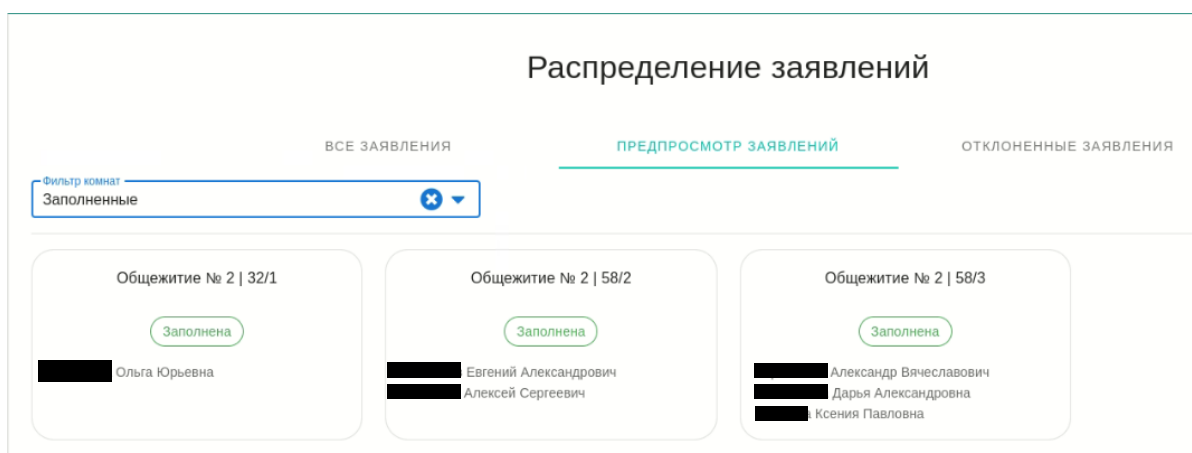


Рисунок 47 – Предпросмотр заявлений

При авторизации пользователей таких как «Бухгалтер», «Начальник УСБ», «Комендант общежития», «Директор студгородка» из комиссии проверяющих, интерфейс меняется и принимает функциональный вид, представленный на рисунке 48.

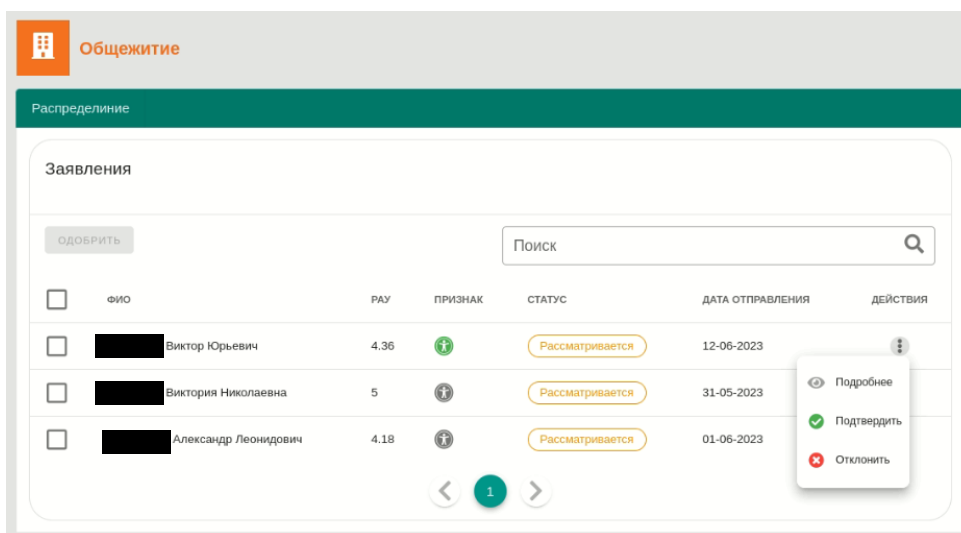


Рисунок 48 – Функциональный вид члена комиссии

Он предоставляет возможности поиска, сортировки, просмотра, одобрения или отклонения заявлений на заселение, поданными студентами. При подтверждении или отклонении заявления статус его меняется соответственно («Подтверждено», «Отклонено»). Для просмотра персональных данных из заявления необходимо нажать на пункт «Подробнее» из выпадающего меню

Для одобрения заявления существует пункт в меню «Подтвердить», по нажатию на который всплывает окно с предпросмотром основных данных заявления и непосредственно самой кнопкой «Одобрить», вид всплывающего окна представлен на рисунке 49.

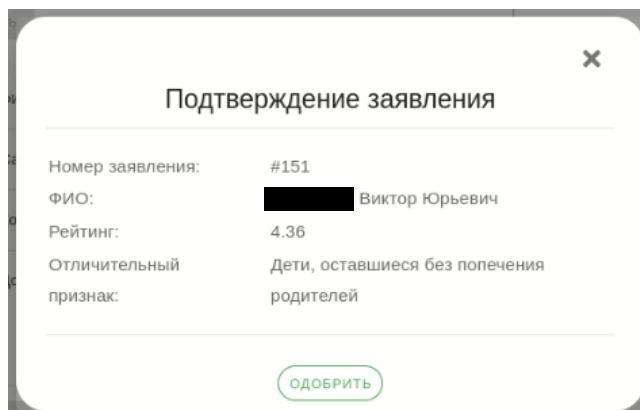


Рисунок 49 – Окно одобрения заявления

Для отклонения заявления существует пункт в меню «Отклонить», по нажатию на который всплывает окно с предпросмотром основных данных заявления и непосредственно полем для формирования комментария, обосновывающего отклонение выбранного заявления, вид всплывающего окна представлен на рисунке 50.

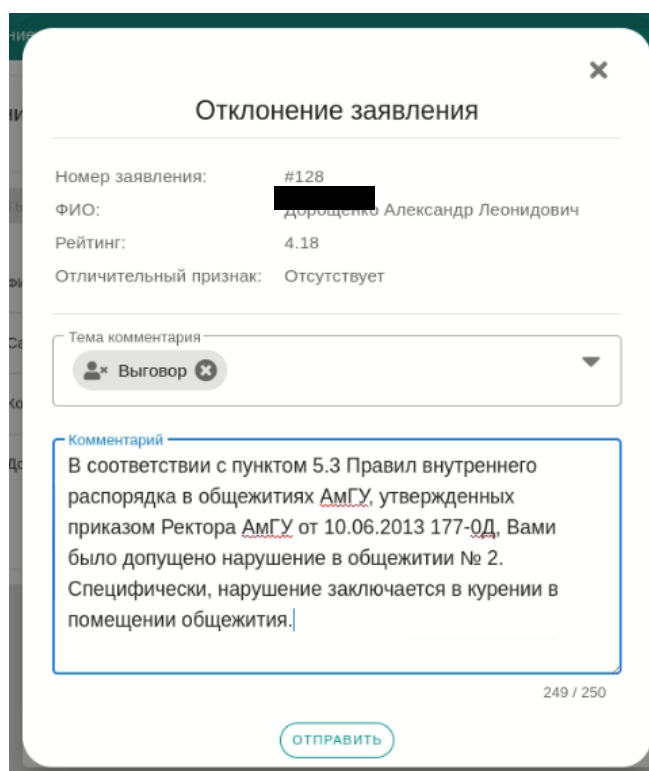


Рисунок 50 – Окно отклонения заявления

Далее если необходимо просмотреть комментарий, то можно воспользоваться пунктом «Подробнее» и увидеть ранее оставленный комментарий для выбранного заявления, рисунок 51.

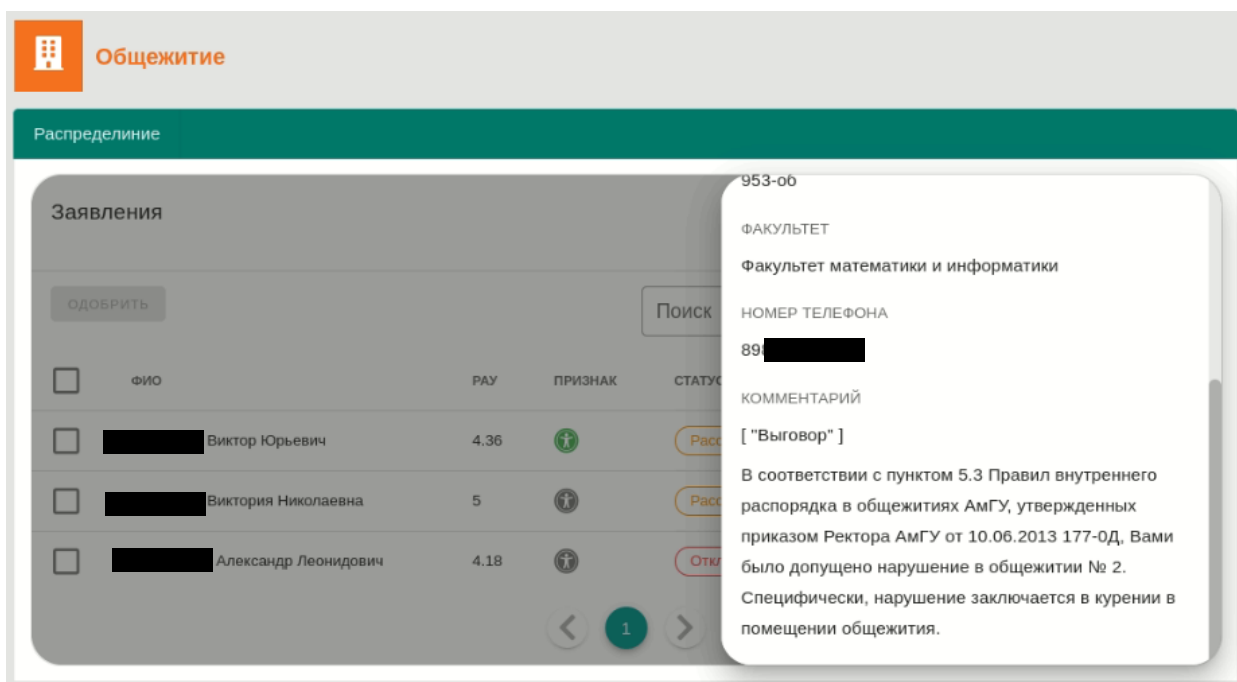


Рисунок 51 – Просмотр комментария

Также на ряду с представленным функционалом существует возможность выбора нескольких заявлений и их массовое одобрение.

Таким образом, на основе предварительно выдвинутых технических и функциональных требований к системе был разработан интерактивный и многопользовательский модуль «Общежитие». Система позволяет проводить верификацию заявлений студентов, отслеживать и устанавливать их статусы, а также осуществлять распределение по местам в общежитиях в соответствии с факультетами. Внешний вид модуля соответствует разработанным требованиям и способствует эффективной работе пользователей.

4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

В данном разделе подробно изучаются вопросы, связанные с обеспечением безопасности сотрудников комиссии, ответственные за рассмотрение заявлений на проживание в общежитиях ФГБОУ ВО «АмГУ». Здесь представлены основные правила работы с компьютерами и меры по обеспечению безопасности персонала, методы безопасной утилизации информационных носителей, оргтехники и компонентов ПК, а также рассмотрены проблемы противопожарной безопасности. Кроме того, уделяется внимание безопасности в процессе разработки модуля «Общежитие».

4.1 Безопасность

Безопасность жизнедеятельности – это комплекс мер и правил, направленных на предотвращение возникновения опасностей и минимизацию рисков, связанных с различными аспектами человеческой деятельности. Она является важным аспектом в обеспечении благоприятных условий для жизни, работы и отдыха людей.

В рамках работы с персональными электронно-вычислительными машинами, безопасность жизнедеятельности относится к набору мероприятий, направленных на защиту здоровья и безопасности пользователей при работе с компьютерами и связанными с ними устройствами. В данном контексте безопасность жизнедеятельности сосредоточена на предотвращении возникновения травм, заболеваний и прочих опасностей, связанных с использованием ПЭВМ.

БЖД охватывает различные области и находит применение в разных сферах человеческой деятельности. В производственной среде БЖД включает в себя разработку и соблюдение правил по безопасному использованию оборудования и технологий, предотвращение пожаров и аварий, обеспечение безопасных условий труда и эргономики. В образовательных учреждениях БЖД направлена на создание безопасной обстановки для учащихся и соблюдение правил пожарной безопасности.

4.1.1 Необходимые условия для работы с ПЭВМ в помещении

Для обеспечения безопасности жизнедеятельности и эффективной работы с персональными электронно-вычислительными машинами в помещении, необходимо учесть ряд важных факторов. Следует обратить внимание на организацию помещения, расположение компьютеров и столов, а также поддержание оптимальных параметров окружающей среды, таких как влажность и температура. Исходя из СанПиН 1.2.3685-21 можно выделить основные показатели для организации благоприятной среды, а также опираясь на ГОСТ Р 50948-2001 и ГОСТ Р 50949-2001 можно определить каким образом должно быть организовано рабочее помещение с ПЭВМ. Итогом анализа стандартов и нормативного документа является требования и рекомендации, описанные далее.

Организация помещения:

- помещение должно иметь достаточное пространство для размещения ПЭВМ, столов, стульев и другого необходимого оборудования;
- размещение компьютеров и столов должно быть организовано таким образом, чтобы обеспечить комфортное рабочее пространство с достаточной свободой движения;
- помещение должно быть хорошо освещено естественным или искусственным освещением, предотвращающим усталость глаз.

Расположение компьютеров и столов:

- компьютеры следует устанавливать на стабильных поверхностях, чтобы предотвратить их падение или перекося;
- столы должны быть достаточно прочными, чтобы выдерживать вес компьютеров и других устройств, а также обеспечивать удобную рабочую поверхность;
- размещение мониторов должно быть таким, чтобы экраны находились на уровне глаз пользователей, что способствует правильной осанке и уменьшает напряжение в шее и спине;
- клавиатуры и мыши должны быть размещены так, чтобы руки и

предплечья пользователей находились в комфортном положении, избегая перенапряжения мышц.

Оптимальные параметры окружающей среды:

– влажность помещения должна поддерживаться в пределах рекомендуемых значений, обычно от 40 % до 60 %, чтобы предотвратить появление статического электричества и обеспечить комфортные условия для работы;

– температура помещения должна быть умеренной, в пределах 20-24 градусов Цельсия, чтобы избежать перегрева компьютерного оборудования и создать комфортные условия для пользователей;

– хорошая вентиляция помещения необходима для обеспечения свежего воздуха и удаления излишнего тепла, а также для предотвращения скопления пыли и других аллергенов.

4.1.2 Необходимые условия шума и вибрации в процессе работы

В соответствии с требованиями ГОСТ 12.1.003–99 «Шум. Общие требования безопасности», в помещениях с компьютерами, где работают инженерно-технические специалисты, уровень шума не должен превышать 60 дБ.

В компьютерах источниками шума являются вентиляторы, трансформаторы и жесткие диски. Для снижения уровня шума рекомендуется использовать специально разработанные корпуса, которые являются активной частью системы охлаждения. Все вентиляторы в системном блоке заменяются радиаторами с трубками для отвода тепла. Трубки имеют тепловые интерфейсы с корпусом, что позволяет эффективно удалять тепло от источников. Такой корпус не содержит вентиляторов, что практически устраняет шум, связанный с работой компьютера.

Чтобы исключить излишний шум от принтера, можно использовать лазерный принтер или разместить принтер вне рабочего помещения.

Также стоит отметить, что шум, создаваемый новыми моделями люминесцентных ламп, практически не заметен.

4.1.3 Необходимые требования к организации и эргономике рабочего места

Организация и эргономика рабочего места являются важными аспектами для обеспечения комфорта, безопасности и эффективности работы с ПЭВМ. При создании рабочего пространства необходимо учитывать ряд требований, включая размеры рабочего места, безопасное расстояние глаз от экрана и правильное положение тела пользователя [9].

Важным аспектом являются размеры рабочего места. Рабочее место должно обладать достаточными размерами для размещения всех необходимых элементов, включая компьютер, клавиатуру, мышь и при наличии необходимые документы. Ширина стола должна быть достаточной для удобного размещения компьютера и обеспечения свободы движения рук пользователя. Глубина стола должна позволять разместить клавиатуру и мышь перед пользователем с достаточным пространством для покоя рук [7].

Безопасное расстояние между глазами пользователя и экраном монитора также имеет важное значение. Рекомендуются, чтобы расстояние составляло примерно (50-70) см. Экран должен быть расположен прямо перед пользователем на уровне глаз или немного ниже, чтобы уменьшить напряжение глазных мышц и предотвратить наклон головы вперед или назад.

Правильное положение человека играет ключевую роль в предотвращении неприятных последствий для здоровья, данный факт подтверждается в ГОСТ Р ИСО 1503-2014. Стул должен обеспечивать поддержку спины и иметь возможность регулировки высоты. Оптимальная позиция – прямая спина и руки, опирающиеся на стол перед собой. Клавиатура и мышь должны быть размещены таким образом, чтобы предотвратить излишнее напряжение в руках и предплечьях. Клавиатура должна быть расположена на уровне локтей пользователя. Ноги пользователя должны находиться на плоской поверхности с углом наклона коленей около 90 градусов. При необходимости можно использовать подставку для ног.

Организация рабочего пространства также имеет важное значение. Оно должно быть свободным от лишних препятствий и перегруженности предметами. Это обеспечит свободу движения и уменьшит риск возникновения травм

или нежелательных последствий. Необходимые документы и материалы должны быть расположены таким образом, чтобы они были легко доступны без излишних поворотов и растяжений.

4.2 Экологичность

С увеличением числа ПЭВМ, возникают вопросы об их влиянии на окружающую среду и потреблении ресурсов. Экологичность в работе с персональными электронно-вычислительными машинами – это важный аспект, который требует внимания и принятия мер для снижения негативного воздействия на окружающую среду.

Согласно отчету Международного союза электросвязи, ПЭВМ потребляют огромное количество электроэнергии, внося значительный вклад в потребление энергии в мире. На самом деле, сектор информационных и коммуникационных технологий ответственен за примерно 2 % глобальных выбросов парниковых газов.

В последние годы индустрия персональных электронных вычислительных машин сделала значительные шаги в направлении повышения энергоэффективности. Производители ПЭВМ стремятся выпускать процессоры с более низким энергопотреблением, не ухудшая их производительности. Это достигается с помощью оптимизации архитектуры процессора и снижения рабочего напряжения. Вместо традиционных холодных катодных люминесцентных ламп (CCFL) в мониторах и ноутбуках все чаще используется светодиодная (LED) подсветка. Она потребляет меньше энергии и обеспечивает более яркое и резкое изображение. Новые технологии экранов, такие как OLED (органические светодиоды), позволяют снизить энергопотребление дисплеев, сохраняя при этом высокое качество изображения. ПЭВМ оснащены режимами сна и ожидания, которые позволяют снизить энергопотребление при неактивном использовании. Компоненты, такие как процессор, жесткий диск и монитор, переходят в спящий режим или снижают свою активность, когда ПЭВМ не используется в течение определенного времени. Современные операционные системы и программное обеспечение предоставляют инструменты для

управления энергопотреблением ПЭВМ. Это может включать настройки яркости экрана, автоматическое выключение устройства при неактивности и оптимизацию использования энергии различных компонентов.

ПЭВМ и другие электронные устройства вносят значительный вклад в проблему электронного мусора. Согласно докладу Группы экспертов по отходам Центра ООН по окружающей среде, в 2019 году генерировалось около 53,6 млн. тонн электронного мусора. Это вызывает серьезные проблемы с утилизацией и переработкой, поскольку в ПЭВМ содержатся различные вредные химические вещества, такие как свинец, ртуть, кадмий и полихлорированные бифенилы [1].

Переработка и вторичное использование: Правильная переработка электронного мусора является важным шагом для уменьшения негативного влияния ПЭВМ на окружающую среду. Многие компании предлагают программы по приему старых устройств для их переработки и вторичного использования.

ПЭВМ содержат различные химические вещества, которые могут быть вредными для окружающей среды и человеческого здоровья. К ним относятся свинец, ртуть, кадмий, бромированные пламеинги и другие. Количество этих веществ может варьироваться в зависимости от конкретных компонентов и материалов, используемых в ПЭВМ [15].

В последние годы были введены международные и региональные нормативы, такие как Директива Европейского союза о ограничении использования определенных опасных веществ в электронной и электротехнической продукции, для ограничения использования опасных веществ в ПЭВМ.

4.3 Чрезвычайные ситуации

На предприятиях могут возникать различные чрезвычайные ситуации, которые могут иметь серьезные последствия. Некоторые из наиболее распространенных и опасных ЧС включают:

– пожары, взрывы и угрозы взрывов. Пожары являются одними из наиболее часто встречающихся чрезвычайных ситуаций на предприятиях. Они могут привести к значительным материальным убыткам, потере жизней и

повреждению инфраструктуры. Угрозы взрывов также представляют серьезную опасность, особенно в промышленных объектах, где находятся взрывоопасные вещества;

– угрозы терроризма. Террористические акты могут произойти на предприятиях и вызвать большой ущерб как в плане материальных потерь, так и в отношении безопасности персонала. Террористы могут планировать нападения, в том числе использование взрывных устройств, оружия или химических веществ;

– угрозы выброса химических веществ. Неконтролируемые выбросы химических веществ на близлежащих промышленных предприятиях могут представлять опасность для окружающей среды и здоровья людей. Это может привести к отравлениям, авариям с химическими веществами и необходимости эвакуации.

Согласно Федеральному закону № 68-ФЗ «О защите населения и территорий от чрезвычайных ситуаций природного и техногенного характера», организации обязаны принимать меры для защиты своих сотрудников и обеспечения безопасности при возникновении чрезвычайных ситуаций. В рамках этого закона они должны:

– организации должны разрабатывать планы и принимать необходимые меры для обеспечения безопасности своих сотрудников в случае чрезвычайных ситуаций. Это включает проведение тренингов, инструктажей и специальных курсов для подготовки сотрудников к возможным чрезвычайным ситуациям;

– организации должны планировать и проводить мероприятия, направленные на обеспечение устойчивости своей деятельности в случае ЧС. Это включает создание систем предупреждения и ликвидации чрезвычайных ситуаций, а также теоретическую и практическую подготовку сотрудников;

– организации должны обеспечить наличие и поддержание систем оповещения о чрезвычайных ситуациях. Это позволит своевременно информировать сотрудников о возможных угрозах и ситуациях чрезвычайного характера;

– организации должны финансировать мероприятия, направленные на защиту своих сотрудников от чрезвычайных ситуаций. Это включает создание финансовых и материальных резервов для ликвидации последствий и обеспечение необходимых средств для защиты;

– организации должны информировать своих сотрудников о мерах защиты от чрезвычайных ситуаций и оповещать их о возможных угрозах. Они также должны сотрудничать с федеральными органами, ответственными за предупреждение и ликвидацию чрезвычайных ситуаций, в установке средств оповещения и распространения информации.

При угрозе или возникновении чрезвычайной ситуации на территории организации, ее руководитель обязан принять меры повышенной готовности, информировать органы управления и включить меры защиты для сотрудников и других лиц, находящихся на территории организации.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была изучена деятельность комиссии по рассмотрению заявлений на места в общежития ФГБОУ ВО «АмГУ» по работе также и с абитуриентами, на основании анализа которой составлено техническое задание, обоснована необходимость создания, спроектирован модуль «Общежития» для информационной системы ФГБОУ ВО «АмГУ».

На этапе проектирования выполнен анализ требований, определены структура модуля, взаимодействие его функциональных модулей и обеспечивающих подсистем. На основании анализа подходящих инструментов разработки, выбраны: язык JavaScript для описания визуальной части модуля, язык Ruby для разработки внутренней архитектуры модуля и связи с внешними источниками информации и СУБД PostgreSQL организации работы базы данных. Созданы макет модуля «Общежития» и структура базы данных.

Результатом работы является законченный модуль «Общежитие» для проведения приемной кампании ФГБОУ ВО «АмГУ». Модуль полностью интегрирован в личный кабинет АмГУ и используется в деятельности вуза, что подтверждается справкой о внедрении.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Безопасность жизнедеятельности в химической промышленности [Электронный ресурс]: учебник / Н. И. Акинин [и др.]. – Санкт-Петербург: Лань, 2019. – Режим доступа: <https://e.lanbook.com/book/116363>.

2 ГОСТ 19.101-77. Единая система программной документации (ЕСПД). Виды программ и программных документов = Unified system for program documentation. Types of programs and program documents : межгосударственный стандарт : издание официальное : утвержден и введен в действие Постановлением Государственного комитета стандартов Совета Министров СССР от 20 мая 1977 г. N 1268 : введен впервые : дата введения 1980-01-01 / Электронный текст документа подготовлен ЗАО "Кодекс" и сверен по: официальное издание Единая система программной документации: Сб. ГОСТов . - М.: Стандартинформ, 2010. - Текст : электронный.

3 ГОСТ 19.102-77. Единая система программной документации (ЕСПД). Стадии разработки = Unified system for program documentation. Development stages : межгосударственный стандарт : издание официальное : утвержден и введен в действие Постановлением Государственного комитета стандартов Совета Министров СССР от 20 мая 1977 г. N 1268 : введен впервые : дата введения 1980-01-01 / Электронный текст документа подготовлен ЗАО "Кодекс" и сверен по: официальное издание Единая система программной документации: Сб. ГОСТов . - М.: Стандартинформ, 2010. - Текст : электронный.

4 ГОСТ 19.201-78. Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению = Unified system for program documentation. Technical specification for development. Requirements for contents and form of presentation : межгосударственный стандарт : издание официальное : утвержден и введен в действие Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. N 3351 : введен впервые : дата введения 1980-01-01 / Электронный Изм. Лист № докум. Подпись Дата Лист 61 КР.195907.09.03.01.ПЗ текст документа

подготовлен АО "Кодекс" и сверен по: официальное издание Единая система программной документации: Сборник национальных стандартов. - М.: Стандартинформ, 2010. - Текст : электронный.

5 ГОСТ Р 7.0.100-2018. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления = System of standards on information, librarianship and publishing. Bibliographic record. Bibliographic description. General requirements and rules : национальный стандарт Российской Федерации : издание официальное : утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 3 декабря 2018 г. N 1050-ст : введен впервые : дата введения 2019-07-01 / разработан Федеральным государственным унитарным предприятием "Информационное телеграфное агентство России (ИТАРТАСС)", филиал "Российская книжная палата", Федеральным государственным бюджетным учреждением "Российская государственная библиотека", Федеральным государственным бюджетным учреждением "Российская национальная библиотека" - М.: Стандартинформ, 2018. - Текст : непосредственный.

6 Гутман, Г. Н. Объектно-реляционная СУБД PostgreSQL: учебное пособие / Г. Н. Гутман. – Самара: Самарский государственный технический университет, ЭБС АСВ, 2016. – 125 с.

7 Кардаш, Т. А. Эргономика рабочих мест служащих и инженерно-технических работников, оснащенных ПЭВМ [Текст] : учеб. пособие / Т. А. Кардаш ; АмГУ, ИФФ. - Благовещенск : Изд-во Амур. гос. ун-та, 2002. - 60 с

8 Направления автоматизации системы управления вузом [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/napravleniya-avtomatizatsii-sistemy-upravleniya-vuzom/viewer>.

9 Пособие по безопасной работе на персональных компьютерах [Текст] / разработ. В. К. Шумилин. - М. : НИЦ ЭНАС, 2005. - 28 с.

10 Прогрессивный JavaScript-фреймворк [Электронный ресурс]. – Режим доступа: <https://ru.vuejs.org/>.

11 Разработка мобильных и веб-приложений [Электронный ресурс]. – Режим доступа: <https://smartum.pro/ru/blog-ru/razrabotka-mobilnykh-i-web-prilozheniy/>.

12 Фиайли К. SQL М.: ДМК Пресс, 2017. – 456 с.

13 Фултон, Х. Программирование на языке Ruby / Х. Фултон; под редакцией Д. А. Мовчан; перевод А. А. Слинкин. – 2-е изд. – Саратов : Профобразование, 2019. — 685 с.

14 Хартл, М. Ruby on Rails для начинающих / М. Хартл. – Москва: ДМК Пресс, 2017. – 572 с.

15 Экология компьютерной техники [Электронный ресурс]: учебное пособие / И. Г. Гетия, В. К. Шумилин [и др.]. – Москва: Московский государственный университет приборостроения и информатики, 2007. – 25 с. –Режим доступа: https://studopedia.ru/14_131205_ekologiya-kompyuternoy-tehniki.html. – 05.05.2019.

16 Angular js [Электронный ресурс]. URL: <https://docs.angularjs.org/tutorial> (дата обращения 01.06.2021).

17 React [Электронный ресурс]. URL: <https://ru.reactjs.org/tutorial/tutorial.html> (дата обращения 01.06.2021).

18 Ruby on Rails по-русски. [Электронный ресурс]. URL: <https://http://rusrails.ru> (дата обращения 16.03.2023).

ПРИЛОЖЕНИЕ А

Техническое задание

1 Введение

1.1 Наименование программы

Наименование программного модуля – «Общежитие».

1.2 Краткая характеристика области применения

Система «Общежитие» предназначена для автоматизации подачи заявления в дистанционном формате на переселения или заселения студентов в общежития с точностью до секции и комнаты, в которой он намерен проживать. Каждый студент, у которого нет прописки в городе обучения, имеет доступ к программному модулю – «Общежитие», где в соответствии со своим рейтингом академической успеваемости и, если имеется, наличием особой квоты может выбрать место для проживания. Людям, имеющим роль «Модератор», позволяется утвердить или отклонить, с описанием результата премодерации.

2 Основания для разработки

Основанием для разработки является потребность в автоматизации бизнес-процесса – подача заявления студентами на места проживания в общежитиях Амурского государственного университета и дальнейшего сбора подписей для подтверждения на заселение.

3 Назначение разработки

Модуль будет использоваться в Амурском государственном университете несколькими группами пользователей: «Студент»; «Модератор» (зам. деканов по ВР, Начальник УСБ, директор студгородка, комендант общежития, главный бухгалтер), «Администратор».

3.1 Функциональное назначение

Для студентов предоставляется возможность дистанционной подачи заявления на проживание в желаемое место.

Для группы пользователей с ролью «Модератор» предоставляется возможность просмотра заявок студентов с различных факультетов, их утверждение или отклонение.

Для группы пользователей с ролью «Администратор» главной задачей

Продолжение ПРИЛОЖЕНИЕ А

будет являться контроль правильности выполнения всей внедренной программной логики, а также сопровождение и при крайней необходимости корректировка подсистемы. Дополнительной возможностью для данных пользователей станет распределение мест общежитий для различных факультетов согласно «Приказу о распределении мест в общежитиях».

3.2 Эксплуатационное назначение

Модуль должен эксплуатироваться в личном кабинете Амурского государственного университета. Имеющий права на доступ и его функциональность, ограниченные по ролям и требованиям к пользователям.

4 Требования к программе или программному изделию

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

В модуле существует три типа пользователя – студент, модератор и администратор. В первую очередь программно проверяется имеется ли у подающего нарушение правил внутреннего распорядка в общежитиях АмГУ, задолженность по оплате за проживания более чем один месяц и прописка в городе обучения. При наличии данных нарушений подача заявления на место в общежитии невозможна. Далее определяется тип пользователя и открывает соответствующий интерфейс со своим функционалом.

Со стороны студента предоставляются следующие возможности:

- просмотра текущего места жительства;
- просмотра рейтинга академической успеваемости за последние 2 семестра, как собственного, так и остальных студентов текущего факультета;
- просмотра «категории приоритета» с пояснением, что включает в себя каждая из категорий;
- просмотра нормативных документов;
- подачи заявления, в котором присутствует выбор общежития, секции, комнаты.

Подача заявления осуществляется двумя способами (при этом сопровождается помощью программы, которая в соответствии с категориями

Продолжение ПРИЛОЖЕНИЕ А

приоритета и индивидуальным признаком): Студент непосредственно выбирает места жительства или остаться в текущей комнате, без просмотра остальных заявок студентов факультета.

Список заявлений заполняется программно по следующему правилу в виде сортированного списка (от самого высокого к самому низкому приоритету):

1. Студент, имеющий отличительный признак (курс не учитывается).
2. Студент, ранее проживающий в общежитии:
 - 2.1. Студент, имеющий наивысшую категорию приоритета;
 - 2.2. Студент, имеющий высшую категорию приоритета;
 - 2.3. Студент, имеющий приоритет выше среднего.
3. Студент, ранее не проживающий в общежитии;
4. Студент, имеющий среднюю категорию приоритета;
5. Студент, имеющий низкую категорию приоритета.

Для того чтобы была возможность определять на какое место проживания может претендовать студент, выявляются «категории приоритета»

Кроме того, учитываются оценки по практикам, по курсовым работам и проектам, а также зачетам с оценкой.

Таблица 26 – Влияния различных факторов на ранжирование заявлений

Факторы	Степень важности от 1 до 3, от самого высшего до низшего
Отличительный признак (пример, дети, оставшиеся без попечительства родителей, инвалиды)	1
Рейтинг академической успеваемости (среднее значение оценок, набранных за два последних семестра)	2
Время подачи заявления	3

На основе факторов, работа алгоритма ранжирования следующая: если студент имеет отличительный признак, то он имеет высокий приоритет на подаваемую комнату. При отсутствии или совпадении отличительного признака с другим человеком, учитывается рейтинг академической успеваемости (РАУ) и уже от его значения заявление занимает соответствующую позицию. Если же и РАУ совпадает с другим студентом, то сравнивается время подачи

Продолжение ПРИЛОЖЕНИЕ А

заявлений и соответственно на основании этого определяется приоритет на подаваемое место.

После подачи заявления имеется информация о статусе заявления:

- заявление находится в рассмотрении;
- заявление подтверждено;
- заявление отклонено с описанием причины.

Алгоритм проверки заявления представляет поочередное его подтверждение группой ответственных лиц. В группу ответственных лиц (модераторы) входят:

- Бухгалтер;
- Начальник УСБ;
- Комендант общежития;
- Директор студгородка;
- Зам. декана по ВР.

Для таких типов пользователей как «Модератор» предоставляется отдельный интерфейс, включающий в себя возможности:

- просмотра списка заявлений студентов по факультетам, общежитиям, секциям и комнатам (в случае Зам. декана по ВР, только список студентов по факультетам);
- просмотра актуальных заявок на проживание;
- одобрение или отклонение заявления с описанием причины. Все результаты решения отображаются непосредственно в списках заявлений;
- самостоятельное определение студента в определенное место жительства.

Заявления, распределенные по факультетам, отображённые в соответствии с алгоритмом ранжирования, должны содержать следующую информацию:

- ФИО студента;
- Рейтинг академической успеваемости;
- Место проживания, на которое подал студент.

Продолжение ПРИЛОЖЕНИЕ А

Для типа пользователя администратор предоставляется весь функционал, предоставляемый модератору и студенту, а также возможность самостоятельно сформировать заявление подающего без его прямого участия может только администратор и только в исключительных случаях и возможность распределения мест общежитий для различных факультетов согласно «Приказу о распределении мест в общежитиях».

4.1.2 Требования к организации входных и выходных данных

Данные о поданных заявлениях и основной информации о студентах будут храниться в базе данных

«PostgreSQL». Доступ к БД будет реализован через «Человеко-машинный интерфейс» при помощи архитектуры запросов RESTful. Пользователь при помощи интерфейса будет слать запрос на сервер, где находится БД и СУБД, и получать ответ от сервера в виде данных, которые сразу же отображаются в представлении. Как студент, так и «Модератор» может отправлять данные для их записи в БД.

Валидация введённых данных происходит сперва на клиенте и проверяет:

- заполнены ли все поля с атрибутом «Обязательные»;
- введены корректные данные;
- не превышено количество заданных символов;

Вторая валидация происходит на сервере, и она во много дублирует валидацию на клиенте, если каким-то образом прошла невалидная информация на клиенте, а также:

- проверка на уникальность созданной записи для определенного пользователя;
- данные корректно сохранились на сервере и имеет валидное название.

4.1.3 Требования к временным характеристикам

Изменения в БД должны отображаться с задержкой не более 5 секунд, запись на сервер не должна превышать время ожидания равное 20-ти секундам.

4.2 Требования к надежности

Вероятность безотказной работы системы должна составлять не менее 99% при условии исправности сети.

4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы.

В связи с тем, что в базе данных хранятся личные данные пользователей. Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- организация хранения информации в виде raid 10;
- организацией бесперебойного питания технических средств;
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций министерства труда и социального развития рф, изложенных в постановлении от 23 июля 1998 г. «об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию пэвм и оргтехники и сопровождению программных средств»;
- выполнением требований гост р 53114-2008 защита информации. обеспечение информационной безопасности в организации;
- регулярным выполнением требований гост 51188-98. защита информации. испытания программных средств на наличие компьютерных вирусов.

4.2.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 10 минут при условии соблюдения условий эксплуатации технических и программных средств. Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать 1 день.

4.2.3 Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий

Продолжение ПРИЛОЖЕНИЕ А

пользователя при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

4.3 Условия эксплуатации

Система размещается на сервере вуза с устойчивой связью доступом в интернет, пропускной мощностью канала не менее 1000/1000 (Mbps), с описанным DNS и DHCP протоколами, а также связью с системой 1С Университет.

4.3.1 Климатические условия эксплуатации

Специальные условия не требуются.

4.3.2 Требования к видам обслуживания

Не реже чем в 6 месяцев производить обновление системы и связанных с ней компонентов.

4.3.3 Требования к численности и квалификации персонала

При установке, настройке и сопровождении системы необходим системный администратор, для организации бесперебойной работы сервера и устранения возникших технических проблем во время работы системы. В процессе эксплуатации с системой работает пользователь (студент, оператор).

Системный администратор должен иметь высшее профильное образование. В перечень задач, выполняемых системным администратором, должны входить:

- развертывание и обновление системы;
- настройка сетевого оборудования.

Пользователь программы должен обладать практическими навыками работы с графическим пользовательским интерфейсом, знать русский язык.

4.4 Требования к составу и параметрам технических средств

Состав технических средств:

- Сервер предприятия:
 - Дисковое пространство, организованное в формате Raid 10,

Продолжение ПРИЛОЖЕНИЕ А

объемом не менее 512 гб.

– Доступ к сети интернет и пропускной мощностью канала не менее 1000/1000 (Mbps).

4.5 Требования к информационной и программной совместимости

Для корректной работы системы необходимо программное обеспечение:

– Ruby 2.7.1 и выше.

– Rails 6.0.3.5 и выше.

– Vue 2 и выше.

– Vuetify 2 и выше.

– Axios 0.27.2 и выше.

– lodash 4.17.21 и выше.

4.6 Требование к маркировке и упаковке

Программное обеспечение распространяется по средство системы контроля версий GitLab. Специальных требований к маркировке не предъявляется.

4.7 Требования к транспортированию и хранению

Специальных требований не предъявляется.

4.8 Специальные требования

Специальных требований не предъявляется.

5 Требования к программной документации

Предварительный состав программной документации:

– Техническое задание.

– Руководство пользователя.

6 Технико-экономические показатели

Модуль разрабатывается для информационной системы ФГБОУ ВО «АмГУ» и не предназначен для извлечения прибыли. Достоинством данного программного продукта является возможность полной интеграции в уже существующую информационную сеть.

7 Стадии и этапы разработки

Разработка должна быть проведена в четыре стадии:

Продолжение ПРИЛОЖЕНИЕ А

- Анализ предметной области;
- Техническое задание;
- Технический (и рабочий) проекты;
- Внедрение.

На стадии «Анализ предметной области» должен быть выявлены потребности заказчика и определены задачи которые решает программа, её состав и функции.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- разработка модульного расширения для автоматической подачи заявлений на койко-место в общежитии;
- разработка программной документации для сопровождения;
- прохождения ряда программных испытаний;
- прохождения испытаний в реальных условиях.

На стадии «Внедрение» должен быть выполнен этап разработки «Подготовка и передача программы».

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к программе;
- определение стадий, этапов и сроков разработки программы и документации на нее;
- согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть

Продолжение ПРИЛОЖЕНИЕ А

выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

8 Порядок контроля и приемки

При приемке проводятся приемо-сдаточные испытания. Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний. На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

8.1 Исполнители

Руководитель разработки

Преподаватель

Исполнитель

Студент группы

Еремина В.В.

Чирич М.Е.

953-об