

Министерство просвещения РФ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информатики, информационных технологий
и методики обучения информатике

Информационная система для заселения студентов в общежитие

*Выпускная квалификационная работа
бакалавра по направлению подготовки
09.03.02 – Информационные системы и технологии*

Работа допущена к защите
«___» _____ 2022 г.
Зав. кафедрой _____

Исполнитель: студентка группы ИСиТ 1801
Института математики, физики,
информатики и технологий
Перевозникова О.С.

Руководитель: старший преподаватель
кафедры ИИТ и МОИ
Шимов И.В.

Екатеринбург – 2022

Реферат

Перевозникова О.С. ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ ЗАСЕЛЕНИЯ СТУДЕНТОВ В ОБЩЕЖИТИЕ, выпускная квалификационная работа: 56 стр., 33 рис., 42 библи. назв., 1 приложение.

Ключевые слова: информационная система, общежитие, автоматизация процессов, базы данных, заселение студентов.

Объект исследования: автоматизация процесса ввода данных о студентах в систему и автоматизация процесса заселения их в общежитие.

Цель работы: разработать информационную систему для заселения студентов в общежитие и сделать ее полностью автоматизированной.

В выпускной квалификационной работе описаны результаты анализа нормативной документации, регламентирующей специфику работы общежития и процедуру заселения студентов в общежитие.

На основе выбранных технологий и инструментов были построены функциональная диаграмма, диаграммы декомпозиции и UML диаграмма классов, для формирования структуры разрабатываемой информационной системы. Также был реализован программный продукт, обеспечивающий ввод данных в базу общежития и расселение студентов по комнатам в автоматизированном режиме.

Данный продукт реализован в среде разработки IntelliJ IDEA на языке программирования Java с использованием инструмента для разработки дизайна Scene Builder и приложения для работы с базами данных MySQL Workbench.

Оглавление

Реферат	2
Введение.....	4
Глава 1. Теоретико-аналитическая часть.....	6
1.1 Специфика работы	6
1.2 Анализ существующих программных решений и аналитический обзор технологий для разработки информационной системы	11
1.3 Формализованное описание технического задания.....	24
Глава 2. Разработка информационной системы.....	29
2.1 Модельные представления информационной системы	29
2.2 Описание работы информационной системы	33
2.3 Результаты апробации, техническая документация	45
Заключение	50
Список информационных источников.....	51
Приложения	57

Введение

Выпускники после школы, поступающие в вузы, получают особые права непосредственно после поступления и во время обучения в учебном заведении [37]. Одним из таких особых прав является предоставление вузом места в общежитии [38].

Жизнь в общежитии имеет массу преимуществ. Стоимость проживания в общежитии разительно отличается от стоимости аренды квартиры или комнаты в городе пребывания студента для учёбы. Так же общежитие удобно располагается рядом с вузом, что позволяет уменьшить расходы на транспорт и время, потраченное на путь. Дополнительным бонусом в общежитии является комфортная атмосфера. Имеется возможность проживать с одногруппниками, однокурсниками либо же просто с друзьями из университета, заводить новые знакомства и находить людей со схожими интересами.

И все же для предоставления комнаты в общежитии существуют характеристики студентов, которым место предоставляется в первую очередь. Данные характеристики описаны в Федеральном законе «Об образовании в Российской Федерации» [39]. Из этого следует, что заселение студентов, не подходящих под характеристики, возможно лишь при условии, что в общежитии имеются в наличии свободные места.

Рассмотрим спецификацию работы общежития, взяв за основу общежитие №4 УрГПУ, в котором за заселение студентов отвечает тьютор общежития, если руководствоваться положением о заселении [32, с.8].

Во избежание возможных конфликтов и предугадывая наличие общих интересов, тьютор заселяет студентов взяв за основу специфику их обучения. Также тьютор может учитывать желания студентов, в которые входит возможность проживания с определенным студентом, либо же проживание в блоке повышенной комфортности, чтобы сделать проживание в общежитии комфортнее для студента.

Сам процесс заселения является длительным и энергозатратным, так как сводится к ручному вводу данных в систему и ручному распределению студентов по местам, учитывая все критерии.

Трудоемкость процесса заключается в высокой вероятности допустить ошибку из-за человеческого фактора, что в итоге может повлиять на исход данного процесса. Ошибку можно допустить при вводе данных, а также утрата или непреднамеренная подмена информации может застопорить процесс заселения. Именно поэтому автоматизированное решение для ввода данных о студентах и их заселении очень необходимо для общежития №4 УрГПУ.

Продукт разработки: информационная система для заселения студентов в общежитие.

Цель работы: автоматизация процесса ввода данных о студентах в систему и автоматизация процесса заселения их в общежитие.

Задачи:

1. Произвести анализ нормативной документации, регламентирующей специфику заселения студентов в общежитие.
2. Произвести патентный поиск существующих решений; проанализировать и обосновать выбор технологий реализации и необходимых программных платформ.
3. В соответствии с техническим заданием провести разработку информационной системы для заселения студентов в общежитие.
4. Подготовить руководство пользователя и провести апробацию.

Глава 1. Теоретико-аналитическая часть

1.1 Специфика работы

Перед началом выполнения работы нам необходимо определиться с трактовкой ключевого понятия «общежитие».

В свободной энциклопедии «Википедия» приводится следующее определение понятия: «общежитие – оборудованное для жилья помещение, предоставляемое учебным заведением или предприятием для проживания работников предприятия либо учащихся» [29].

В положении о студенческом общежитии трактуется специфика работы общежития №4 УрГПУ. В данном документе прописаны характеристики студентов, которым предоставляется место в общежитии:

«Студенческое общежитие УрГПУ предоставляется для временного проживания и размещения:

- на период обучения иногородних студентов, аспирантов, докторантов, обучающихся на очной форме обучения;
- на период очного обучения слушателей подготовительного отделения;
- на период прохождения промежуточной и итоговой аттестации обучающихся по заочной форме обучения, при наличии свободных мест;
- абитуриентов на период прохождения вступительных испытаний» [32, с.4].

Из этого следует, что общежитие предоставляется иногородним студентам УрГПУ на определенный период, зависящий от формы обучения или период прохождения вступительных испытаний.

Обратимся к положению о заселении, чтобы понять кто отвечает за процесс распределения студентов по комнатам в общежитии:

«Ежегодно для распределения мест в общежитии для нуждающихся обучающихся создается комиссия по поселению в составе:

- ведущий тьютор;
- тьюторы общежитий;
- председатели ЖБК учебных подразделений;
- председатели студенческих советов общежитий.

После завершения зачислений, поступающих на очную форму обучения, приемная комиссия предоставляет данные о нуждающихся в поселении в общежитие студентах в Комиссию для подготовки протоколов по распределению мест.

На основании протоколов распределения мест ведущий тьютор готовит проект приказа о предоставлении мест в общежитии... <...>» [32, с.31]

Так становится понятно, что за процесс заселения студентов в общежитие отвечает комиссия по поселению. Информация о нуждающихся в заселении студентов попадает к тьютору общежития №4 УрГПУ. Обратимся к свободной энциклопедии «Википедия», чтобы определиться с понятием «тьютор»:

«Тьютор – это преподаватель, который проводит дополнительные занятия с учеником или с несколькими учениками ежедневно, еженедельно или ежемесячно с целью передать им знания или навыки по предмету <...> Иногда тьютор обеспечивает разработку индивидуальных образовательных программ учащихся и сопровождает процесс индивидуализации и индивидуального образования в школе, вузе, в системах дополнительного образования» [34].

После того как стало известно, кто такой тьютор, необходимо узнать его обязанности и роль в процессе заселения в общежитие, что приводится в положении о заселении:

«Тьютор общежития обязан:

- составлять годовой и текущие планы мероприятий работы с обучающимися, проживающими в общежитии, и утвердить их у ведущего тьютора;
- организовать работу органов студенческого самоуправления общежития, оказывать методическую помощь и курировать их деятельность;

- участвовать в работах по массовому заселению обучающихся в начале учебного года;
- контролировать работу объектов социальной сферы общежития;
- осуществлять контроль за поселением студентов в соответствии с предоставленными местами, текущий контроль фактического проживания;
- осуществлять контроль за своевременным выселением проживающих;
- осуществлять контроль за соблюдением требований настоящего положения;
- участвовать в работе общественной жилищной комиссии общежитий;
- организовывать информационную работу с проживающими, проводить информационные собрания;
- организовать работу по привлечению обучающихся и других проживающих к деятельности по самообслуживанию;
- содействовать психологической адаптации обучающихся, проживающих в общежитии, способствовать созданию психологически комфортной среды, прогнозировать ситуации психологической несовместимости проживающих;
- консультировать проживающих в общежитиях УрГПУ по жилищно-бытовым вопросам, связанным с их проживанием» [32, с.8].

Проанализировав обязанности, можно сделать вывод, что тьютор является важным звеном в процессе заселения студентов в общежитие. Тьютор осуществляет поиск оптимальных комнат для студентов, предугадывая совместимость соседей. Также тьютор контролирует поселение и своевременное выселение студентов общежития.

Заселение студентов происходит в определенном порядке, который регламентирует положение о студенческом общежитии:

«В обязательном порядке места в общежитии предоставляются обучающимся, имеющим льготно-социальный статус:

- сироты и оставшиеся без попечения родителей (категория 1);
- инвалиды 1, 2, 3 и категории ребенок-инвалид (категория 2);
- студенты, имеющие права на получение государственной социальной помощи, обучающиеся на бюджетной основе (категория 3);
- студенты, имеющие права на получение государственной социальной помощи, обучающиеся по договору, предусматривающему оплату (категория 4);
- ветеранам боевых действий и участники локальных конфликтов (категория 5);
- студенты, подвергшиеся воздействию радиации вследствие аварии на Чернобыльской АЭС и других радиационных катастроф, а также приравненных к ним территорий (категория 6);
- <...>
- обучающимся, являющимся именными стипендиатами (категория 7);
- обучающимся, активно занимающимся общественной работой (категория 8.1);
- обучающимся из числа победителей и призеров соревнований в личном зачете городского, областного, регионального, всероссийского, европейского, мирового уровня (категория 8.2);
- обучающимся, поступившим по целевому направлению Министерства образования Свердловской области (категория 9);
- из числа доноров, сдавших кровь не менее 2 раз в течение календарного года и на днях донора, организуемых в УрГПУ (категория 10);
- из многодетных семей (категория 11);
- остальным на бюджетной основе (категория 12);
- остальным на внебюджетной основе (категория 13)» [32, с.31-33].

Порядок предоставления комнаты связан с категориями студентов. В первую очередь места предоставляются студентам, имеющим льготно-

социальный статус. Далее категории студентов классифицируются по форме обучения, основе обучения, поступлении и активной деятельности.

После того как был обозначен порядок заселения студентов, основанный на их характеристиках и категориях, необходимо определить каким образом подбираются соседи по комнате.

Возвращаясь к обязанностям тьютора, можно связать определение психологической совместимости, которой он занимается, с поиском подходящих соседей для студентов. Так как методы для определения совместимости никак не регламентируются, то данный процесс полностью зависит от тьютора.

Распределение студентов по комнатам организовано тьютором подобным образом:

- заселить в свободную комнату на определенном этаже, который связан с обучением на определенном направлении;
- найти комнату, в которой проживает студент из той же группы, на определенном этаже;
- найти комнату, в которой проживает студент с того же направления, на определенном этаже;
- найти комнату, в которой проживает студент обучающийся на том же факультете, на определенном этаже;
- заселить в свободную комнату на любом этаже в общежитии;
- заселить в любую комнату, в которой есть свободное место.

Список порядка распределения студентов по комнатам основан на их критериях. Поиск комнаты происходит всегда начиная с первого пункта. Если пункт не дает результата в виде найденной комнаты, удовлетворяющей всем требованиям, то совершается переход к следующему пункту, пока не будет найдено место.

Главным критерием является половая принадлежность студента или другими словами сегрегация по признаку пола. Данное понятие рассмотрено в

свободной энциклопедии «Википедия», так что обратимся к ней, чтобы определить значимость данного критерия:

«Сегрегация по признаку пола, разделение по признаку пола или гендерная сегрегация – это физическое, юридическое или культурное разделение людей в соответствии с их биологическим полом. <...> ...некоторые сторонники утверждают, что он занимает центральное место в определенных религиозных законах и социальных и культурных историях и традициях» [17].

Критерии выявленные выше следующие: причастность к факультету, причастность к направлению, привязанному к номеру этажа, и причастность к группе, которая ссылается на направление и факультет. Так же из критериев можно отметить половую принадлежность студента, его желание быть заселенным в блок повышенного комфорта или с каким-то определенным студентом.

На сайте УрГПУ в разделе общежития описаны жилищно-бытовые условия, где указано, что разрешенное количество проживаемых студентов в комнате – до четырёх человек [30].

Проанализировав специфику работы общежития №4 УрГПУ, становится понятен механизм, критерии и порядок заселения, а также роль тьютора в данной системе и его обязанности.

1.2 Анализ существующих программных решений и аналитический обзор технологий для разработки информационной системы

Несколько лет назад, если судить по имеющимся данным в системе общежития, то в 2011 году, студент УрГПУ разработал систему для ведения записей в общежитиях УрГПУ на платформе 1С:Предприятие и с тех пор в общежитии №4 не было изменений.

Данная система предназначена для хранения данных о студентах и работает по следующему принципу: тьютор получает от приемной комиссии таблицу данных в формате (*.xlsx) и вручную вводит данные в форму для создания новой записи. Рассмотрим интерфейс программы и ее возможности.

Учет электроприборов (1С-Предприятие) Поиск Ctrl+Shift+F

Главное Константы Работа со студентами Электрооборудование

Студенты Электроприборы Доп. соглашения студентов Поступление денежных средств Отчеты Сервис

Студенты (создание)

Записать и закрыть Записать Основной Договор Печать направления Договор с визой Симоновой Еще

Договор №: Дата договора: Форма обучения:

ФИО:

Удостоверение личности

Вид: ☐ Серия: Номер: Дата выдачи: Кем выдан:

Дата рождения: Пол: Категории: ☐ Телефон: E-mail:

Адрес по регистрации:

Телефон родителей:

Адрес родителей:

Общеквартирное: ☐ Комната: Группа: ☐

Номер направления:

Комментарий:

Добавить Еще

N	Статус	ФИО	Дата рождения

Документ уд. личности:

размещение (папка):

Рис. 1. Окно с формой для создания элемента и добавления его в таблицу «Студенты»

Главным минусом такой системы является невозможность автоматического заполнения формы данными из документа, что очень замедляет работу, ведь приходится вводить данные вручную.

Стоит отметить и плюсы данной системы, например в форме существуют возможности, которые уменьшают вероятность ошибки при добавлении данных в таблицу и немного ускоряют процесс ввода информации. Одна из возможностей – контроль данных, например ввод даты контролируется специальным форматом, либо же дату можно выбрать из календаря. Следующая

возможность – выбор некоторых данных из перечня, представленного в виде списка, например форма обучения или категория.

К данным в таблицах можно применить сортировку и/или фильтрацию, что стало бы несомненным плюсом данного решения, но фильтрация, ограниченная только одним параметром, ставит пользователя перед выбором. Пользователю приходится искать вариант, при котором полученные в ходе фильтрации данные были бы максимально приближены к желаемому результату.

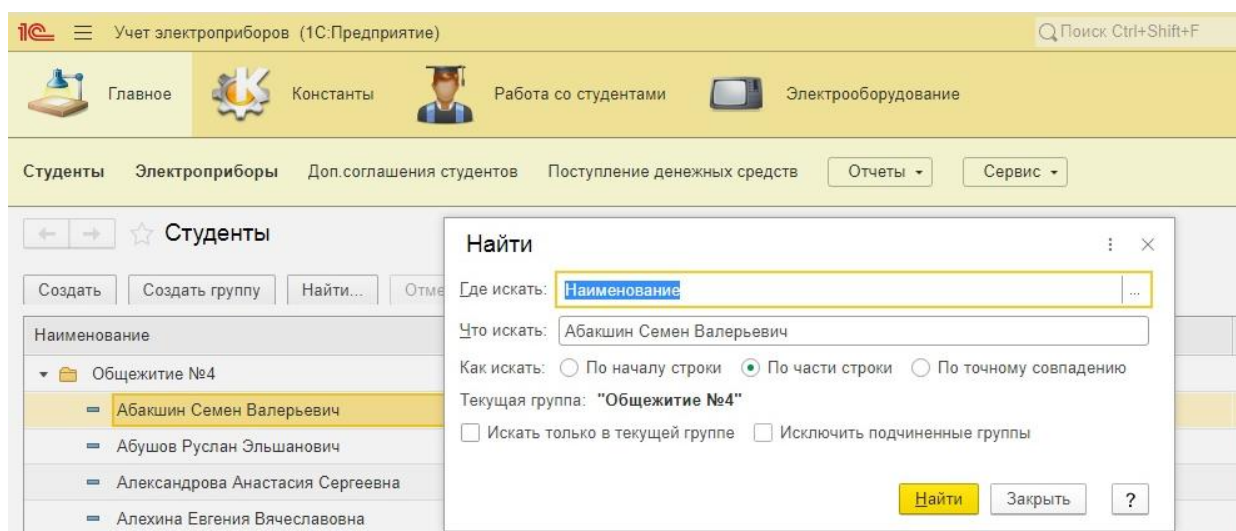


Рис. 2. Окно для фильтрации данных в системе

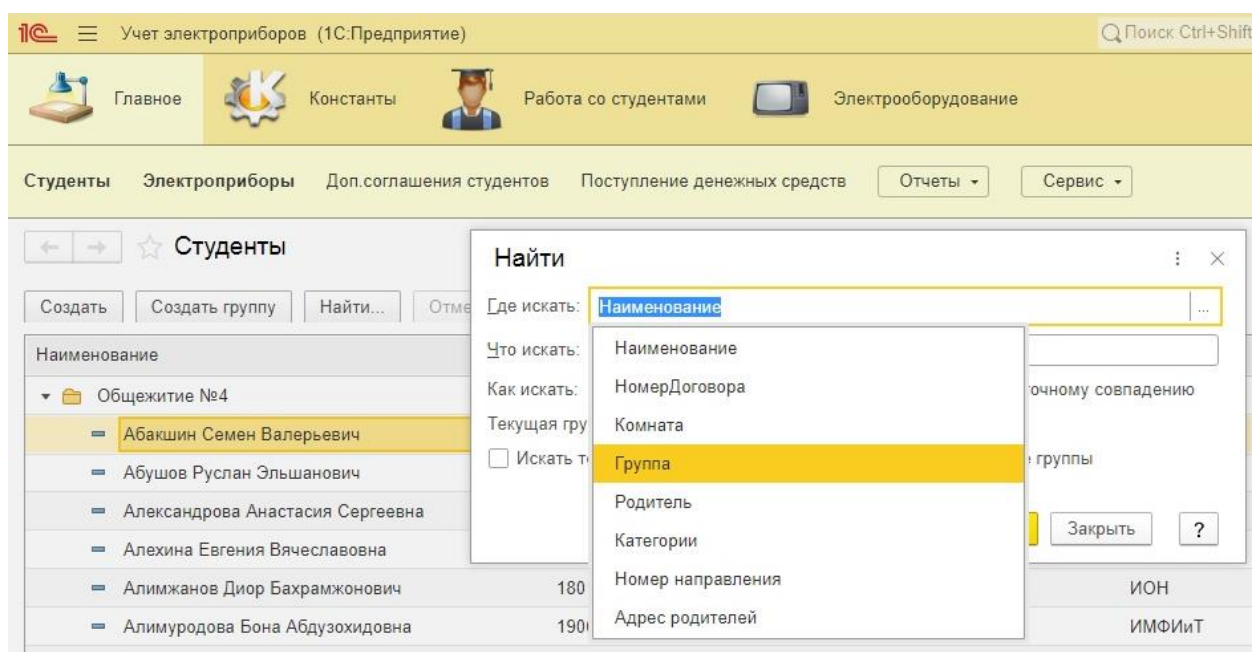


Рис. 3. Возможные параметры фильтрации данных в системе

При дальнейшем тестировании продукта были выявлены еще несколько минусов:

- отсутствие автоматического распределения студентов по комнатам;
- отсутствие формирования отчёта с данными о заселенных студентах.

После выявления минусов имеющейся системы стало понятно, что для упрощения работы тьютора и повышения производительности общежития, требуется новое программное решение.

Чтобы исключить вероятность создания уже готовой системы, перед началом работы имеет смысл произвести патентный поиск существующих решений в сети Интернет.

Выявим критерии для патентного поиска:

- соответствие нормативным документам;
- простота использования;
- автоматизация процесса.

В ходе поиска было найдено единственное решение позволяющее автоматизировать процессы в общежитии, которым является программа «БИТ.Общежитие» [22].

Данный программный продукт является полнофункциональной системой для автоматизации работы общежития. Целью данного решения является ведение учета проживающих, койко-мест и денежных средств при оказании услуг проживания общежитиями [35].

По своему функционалу она превосходит систему, имеющуюся в общежитии тем, что включает в себя расчет и контроль оплаты за проживание (Рис. 4), а также характеристики и наличие свободных мест (Рис. 5).

Квитанция 000000006 от 31.03.2017 12:00:01

Провести и закрыть | Провести | Создать на основании | Форма Н ПД-4 сб(налог) | Все действия ?

Номер: 000000006 от: 31.03.2017 12:00:01 | Общежитие: Общежитие № 2

Платательщик: | Оплата за период с: 01.01.2017 по: 31.03.2017

Содержание: | ☒ Оплата произведена Дата оплаты: 31.03.2017 0:00:00

Валюта: RUB; НДС включен в сумму | ☐ Аннулирована

Отборы | Проживание (6) | Доп. услуги

Добавить | Заполнить | Подобрать | Все действия ?

N	Проживающий	Договор	Сумма	Ставка НДС	Сумма НДС	Всего	Содержание
1	Истомина Эмма Оле...	Договор 3	3 000,00	18%	457,63	3 000,00	
2	Маркелов Алексан...	Договор 624	9 350,00	18%	1 426,27	9 350,00	
3	Морозов Анатолий С...	Договор от 01.01.20...	1 200,00	18%	183,05	1 200,00	
4	Москвин Игорь Конс...	Договор 156 от 01.0...	49 500,00	18%	7 550,85	49 500,00	
5	Москвина Валентина...	Договор 156 от 01.0...	3 300,00	18%	503,39	3 300,00	
6	Степанова Юлия Ива...	Договор 2	22 500,00	18%	3 432,20	22 500,00	

Всего: 88 850,00 RUB НДС (в т.ч.): 13 553,39 RUB

Ответственный: Администратор

Комментарий:

Рис. 4. Взаиморасчеты с проживающими программа «БИТ.Общежитие» [22]

Комната 104 (Номерной фонд)

Записать и закрыть | Все действия ?

Тип: Комната

Наименование: Комната 104

В эксплуатации с: 01.01.2015 | Снято с учета с: 01.01.2020

Номер комнаты: 4 | Площадь помещения: 0,00

Описание:

Тип комнаты: Жилая | Категория комнаты: 2-комнатный 4-местный

Родитель: Этаж 1 | Код: 00000009

Общежитие: Общежитие № 2

Рис. 5. Учет номерного фонда программа «БИТ.Общежитие» [22]

Программа поддерживается разработчиком, имеется возможность поставки на 5 рабочих мест и заказ бесплатной демонстрации. Основным недостатком программы является ее стоимость, которая за поставку пяти рабочих мест составляет 40000 рублей. Так же из недостатков стоит отметить, что ввод данных и процесс распределения комнат не автоматизирован, как и в системе, имеющейся на данный момент.

После проведения поиска и анализа найденных решений можно сделать вывод, что автоматизированных систем для общежитий нет в общем доступе, а найденные программы не до конца удовлетворяют всем требованиям, так как специфика заселения в каждом общежитии построена уже своим методом. Исходя из этого, самым оптимальным вариантом будет создать свою автоматизированную информационную систему для заселения студентов в общежитие.

Обзор технологий для разработки информационной системы начался с анализа выдвинутых требований. Так как система будет работать с данными и выполнять с ними большое количество операций, то для работы с ними лучшим вариантом будет использование баз данных. После анализа систем управления базами данных, было принято решение остановиться на MySQL [14]. Преимущества данной СУБД:

- лёгкость в использовании;
- большое разнообразие функций;
- высокий уровень защиты данных;
- высокоскоростная обработка данных;
- является бесплатной.

Существует множество программ для разработки и администрирования MySQL. Рассмотрим некоторые: Workbench, Navicat, PHPMyAdmin, HeidiSQL, SQLyog.

В свободной энциклопедии «Википедия» поясняется что такое «MySQL Workbench»:

«MySQL Workbench – инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL» [12].

Следующая программа «Navicat for MySQL» описана на официальном сайте «Navicat»:

«Navicat for MySQL является идеальным решением для администрирования баз данных MySQL / MariaDB. Программа позволяет подключаться к MySQL и MariaDB одновременно в одном приложении, совместима с облачными базами данных Amazon RDS, Amazon Aurora, Oracle Cloud, Google Cloud и Microsoft Azure. Navicat for MySQL имеет интуитивно понятный и мощный графический интерфейс для управления, разработки и обслуживания баз данных» [15].

Программа «phpMyAdmin» описана на официальном сайте «phpMyAdmin»:

«phpMyAdmin – это бесплатный программный инструмент, написанный на PHP, предназначенный для администрирования MySQL через Интернет. phpMyAdmin поддерживает широкий спектр операций на MySQL и MariaDB. Часто используемые операции (управление базами данных, таблицами, столбцами, отношениями, индексами, пользователями, разрешениями и т.д.) могут выполняться через пользовательский интерфейс, в то время как у вас все еще есть возможность напрямую выполнить любую инструкцию SQL» [4].

Программа «HeidiSQL» описана на официальном сайте «HeidiSQL»:

«HeidiSQL является свободным программным обеспечением и имеет цель быть простым в освоении. «Хайди» позволяет просматривать и редактировать данные и структуры с компьютеров под управлением одной из систем баз данных MariaDB, MySQL, Microsoft SQL, PostgreSQL и SQLite.» [20].

И последняя из перечня программа «SQLyog» описана в свободной энциклопедии «Википедия»:

«SQLyog – это графический интерфейс пользователя для популярной системы реляционных баз данных MySQL» [18].

Navicat является платной программой, а у PHPMyAdmin неудобный пользовательский интерфейс. Сравнивая остальные программы между собой, можно сделать вывод, что самым удачным решением для будущей системы является Workbench [11]. Преимущества программы:

- возможность проектировать, моделировать, создавать базы данных и управлять ими визуально;
- удобный редактор SQL запросов;
- быстрая и стабильная работа;
- является бесплатной.

На сайте «MySQL» более подробно описаны возможности проектирования баз данных в программе Workbench:

«MySQL Workbench упрощает проектирование и обслуживание баз данных, автоматизирует трудоемкие и подверженные ошибкам задачи... <...> Это позволяет проектировать базы данных на основе моделей, что является наиболее эффективной методологией для создания действительных и хорошо работающих баз данных, обеспечивая при этом гибкость для реагирования на меняющиеся бизнес-требования. Утилиты проверки моделей и схем применяют стандарты наилучшей практики для моделирования данных, а также применяют стандарты физического проектирования, специфичные для MySQL, чтобы не допускать ошибок при создании новых диаграмм ER или создании физических баз данных MySQL» [13].

На Рис. 6 представлен пользовательский интерфейс программы MySQL Workbench.

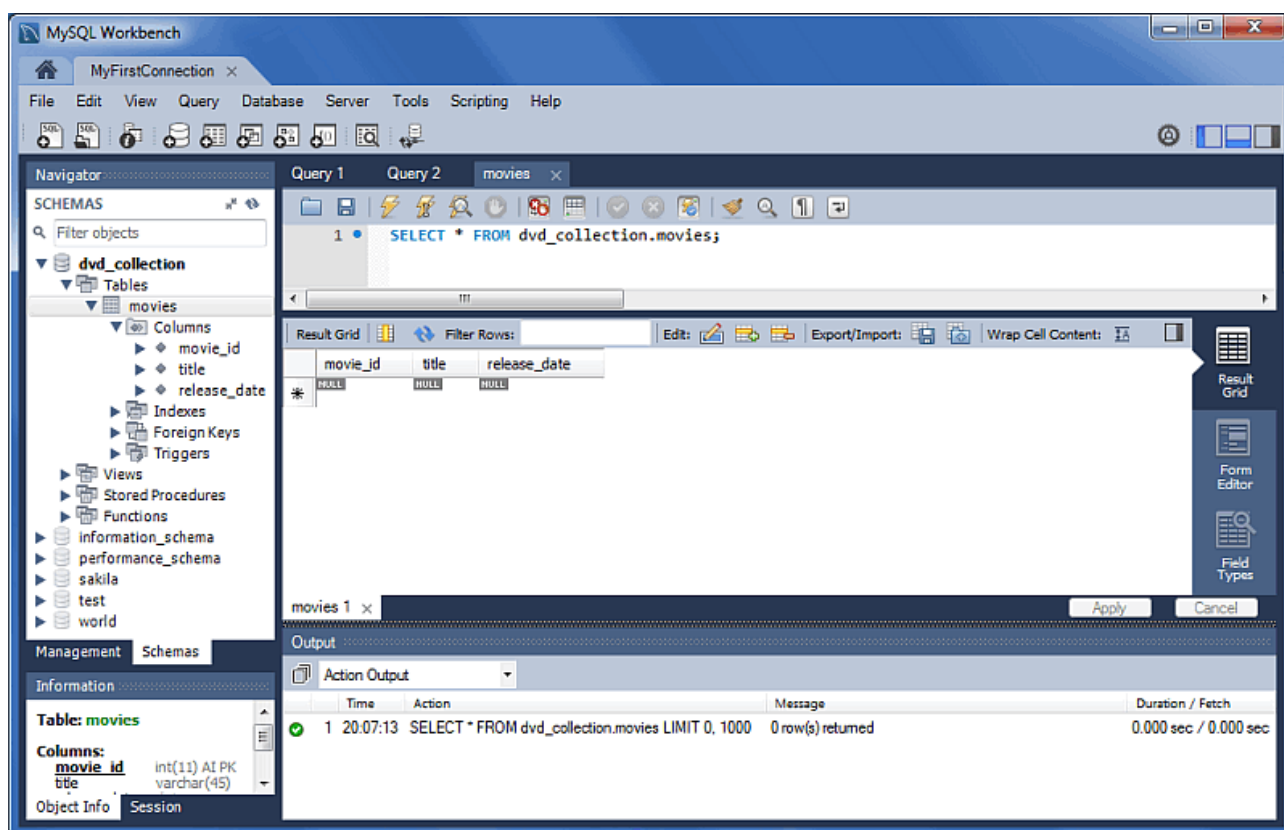


Рис. 6. Пользовательский интерфейс программы MySQL Workbench [7]

MySQL имеет API и коннекторы для нескольких языков программирования, одни из них: C++, Java, PHP, Python.

В свободной библиотеке «Википедия» предоставляется информация о языке C++:

«C++ — компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности» [5].

На официальном сайте «Java» дается понимание того, что такое язык Java и ее применение:

«Java представляет собой язык программирования и платформу вычислений, которая была впервые выпущена компанией Sun Microsystems в 1995 г. Технология эволюционировала из скромной разработки до инструмента, который играет серьезную роль в современном цифровом мире, предоставляя надежную платформу для множества сервисов и приложений» [42].

Ответ на вопрос «что такое РНР» предоставляется на официальном сайте «РНР»:

«РНР (рекурсивный акроним словосочетания РНР: Hypertext Preprocessor) – это распространённый язык программирования общего назначения с открытым исходным кодом. РНР специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML» [41].

И последний язык программирования, который мы рассмотрим – Python:

«Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов» [16].

Произведя анализ характеристик языков программирования, сравнивая нужные качества для разработки нашей автоматизированной системы, был выбран язык программирования Java. Важные качества для разработки:

- есть стандарт взаимодействия Java-приложений с различными СУБД (стандарт JDBC) [10];
- имеются библиотеки, благодаря которым есть возможность работать с файлами формата (*.xlsx, *.docx);

- надежный код, благодаря строгой статистической типизации;
- высокая скорость работы.

После того, как был выбран язык программирования, необходимо выбрать интегрированную среду разработки. Сравним некоторые среды разработки для программирования на языке Java: IntelliJ IDEA, Eclipse, NetBeans, Xcode.

Рассмотрим каждую из них и начнем с IntelliJ IDEA:

«IntelliJ IDEA – это интеллектуальная IDE, учитывающая контекст. Она предназначена для разработки разнообразных приложений на Java и других языках JVM, например Kotlin, Scala и Groovy. Кроме того, IntelliJ IDEA Ultimate поможет в разработке веб-приложений: она предлагает эффективные встроенные инструменты, поддержку JavaScript и связанных с ним технологий, а также расширенную поддержку таких популярных фреймворков, как Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus и Helidon. А бесплатные плагины, разработанные JetBrains, позволяют дополнительно расширить возможности IntelliJ IDEA и использовать ее для работы с другими языками программирования, в том числе Go, Python, SQL, Ruby и PHP» [40].

Рассмотрим среду разработки Eclipse:

«Eclipse – свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation» [6].

Платформа NetBeans рассмотрена на официальном сайте «Apache NetBeans»:

«Платформа NetBeans – это широкая платформа Java, на которой можно создавать большие настольные приложения. Сама среда IDE NetBeans является одним из сотен приложений, основанных на платформе NetBeans. Платформа NetBeans содержит API, которые упрощают обработку окон, действий, файлов и многих других вещей, типичных для приложений» [36].

Последнюю среду разработки Xcode рассмотрим, обращаясь к свободной библиотеке «Википедия»:

«Xcode – интегрированная среда разработки (IDE) программного обеспечения для платформ macOS, iOS, watchOS и tvOS, разработанная корпорацией Apple. Первая версия выпущена в 2003 году. Стабильные версии распространяются бесплатно через Mac App Store» [21].

Разработка продукта будет происходить на устройстве под ОС Windows 10, поэтому данный параметр должен быть учтён. Так же при сравнении сред разработки между собой, нужно учесть разработку на определенном языке программирования. Так что самым оптимальным вариантом является IntelliJ IDEA [33]. Преимущества среды разработки:

- автодополнение кода;
- анализ качества кода;
- быстрота написания и изменения кода;
- может подсказать наиболее подходящие методы, имена для них и параметры.

На Рис. 7 продемонстрирован пользовательский интерфейс интегрированной среды разработки IntelliJ IDEA.

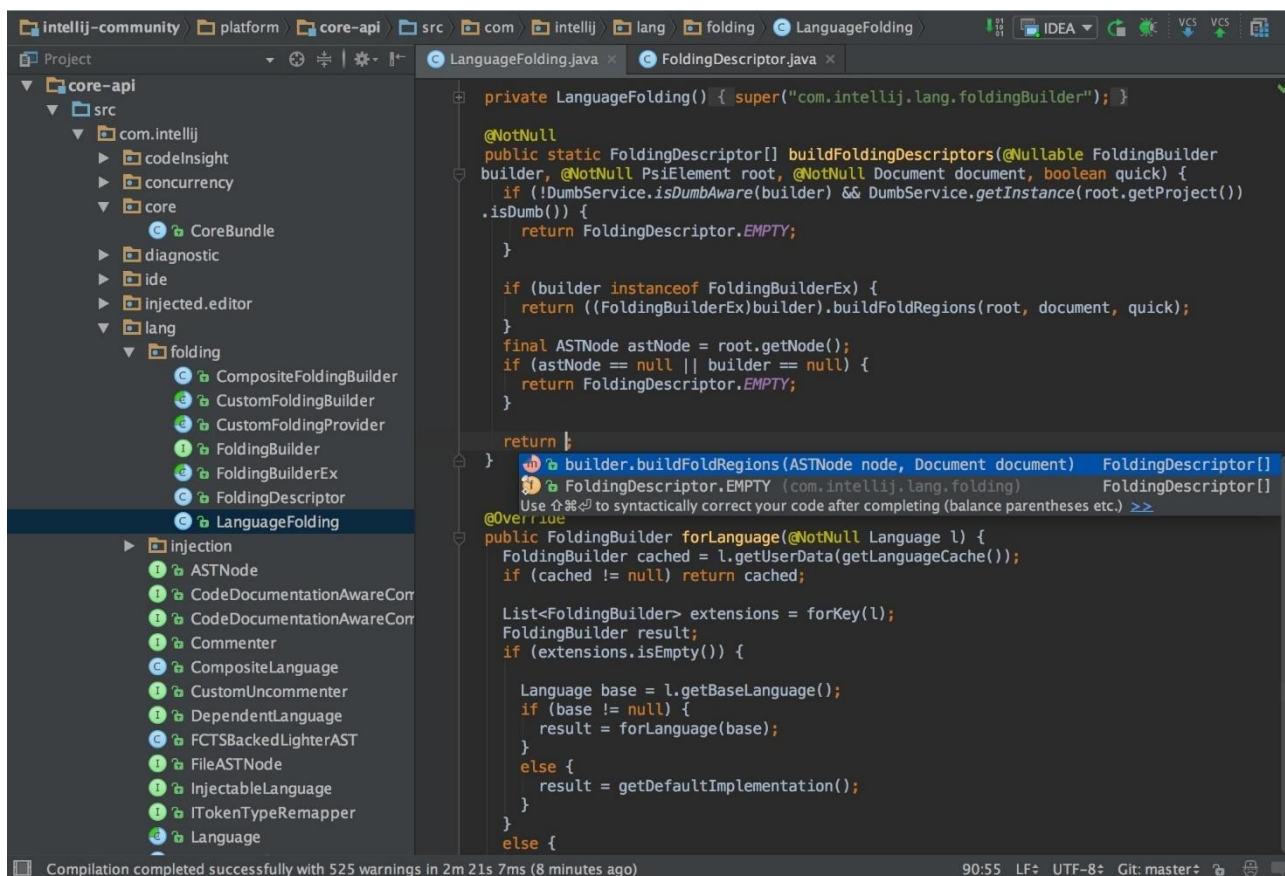


Рис. 7. Пользовательский интерфейс интегрированной среды разработки IntelliJ IDEA [8]

Для разработки пользовательского интерфейса на языке программирования Java подойдет инструмент визуального макета Scene Builder [29].

Этот инструмент позволяет пользователям быстро и просто разрабатывать интерфейсы программ. За счёт перетаскивания компонентов пользовательского интерфейса на рабочую область облегчается процесс создания дизайна программы. Так же есть возможность изменять свойства компонентов и применять таблицы стилей без кодирования, ведь код FXML [9] для создаваемого макета автоматически создается в фоновом режиме, который после можно объединить с проектом Java.

Интерфейс программы продемонстрирован на Рис. 8.

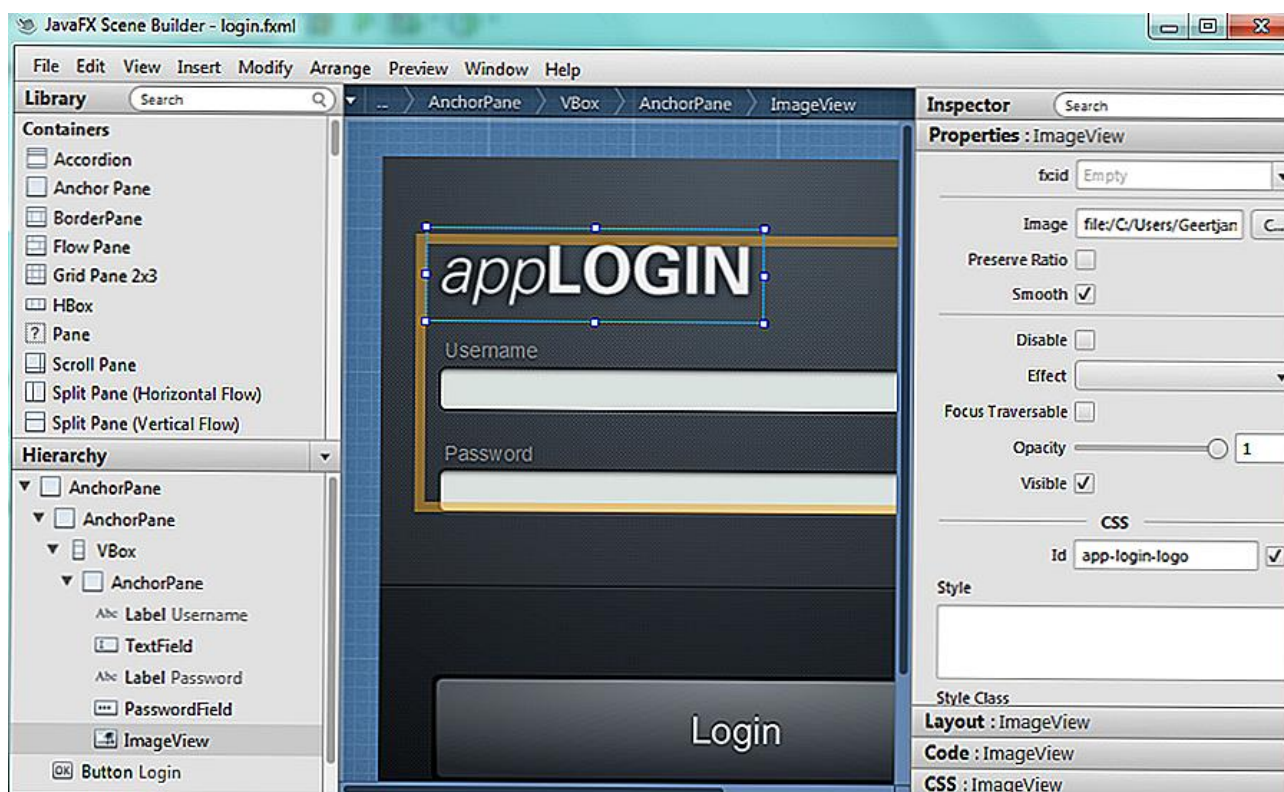


Рис. 8. Пользовательский интерфейс программы Scene Builder [19]

На основе проведённого анализа существующих программных решений в конечном итоге для разработки информационной системы были выбраны следующие решения: интегрированная среда разработки IntelliJ IDEA, язык программирования Java, приложение для работы с базами данных MySQL Workbench и инструмент для разработки дизайна программы Scene Builder.

1.3 Формализованное описание технического задания

1. Общие сведения.

1.1. Название организации-заказчика.

ФГБОУ ВО «УрГПУ».

1.2. Название продукта разработки (проектирования).

«Hostel Tutor».

1.3. Назначение продукта.

Для тьютора общежития, который входит в комиссию по заселению, отвечающей за заселение студентов в общежитие №4 УрГПУ. Система

вычисляет оптимальный номер комнаты для заселения, основываясь на данных о студенте. В дальнейшем может служить справочником информации о заселенных студентах.

1.4. Плановые сроки начала и окончания работ.

В соответствии с планом выполнения ВКР (01.09.2021 – 21.05.2022).

2. Характеристика области применения продукта.

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Автоматизация процесса распределения студентов по комнатам при заселении в общежитии №4 УрГПУ.

2.2. Характеристика персонала (количество, квалификация, степень готовности).

Тьютор – уровень владения IT на уровне уверенного пользователя.

3. Требования к продукту разработки.

3.1. Требования к продукту в целом.

Приложение под управлением ОС Windows с адаптивным дизайном.

3.2. Аппаратные требования.

Процессор: Pentium 2 266МГц;

Оперативная память: 2 ГБ;

Разрешение экрана: минимум 1280 × 1024.

3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).

Операционные системы: Microsoft Windows 7 и выше.

Базовое ПО: Microsoft Excel, Microsoft Word.

Системные программы: MySQL Workbench 8.0.

3.4. Указание программного обеспечения, используемого для реализации.

MySQL Workbench 8.0, IntelliJ IDEA Community Edition 2021.3.2, SceneBuilder.

3.5. Особенности реализации серверной и клиентской частей.

Не предусмотрено.

3.6. Форматы входных и выходных данных

Входные данные: Таблица данных в формате Microsoft Excel, База Данных «Hostel».

Выходные данные: Документ в формате Microsoft Word.

3.7. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения.

Данные импортируются из файла (*.xlsx) в БД или вводятся через диалоговое окно информационной системы.

Данные выводятся на экран системы из БД, так же есть возможность формирования отчёта (*.docx).

Хранятся данные в БД.

3.8. Порядок взаимодействия с другими системами, возможности обмена информацией.

Взаимодействует с приложением MS Excel и с приложением MySQL Workbench.

3.9. Меры защиты информации.

Не предусмотрено.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

WIMP интерфейс.

4.2. Размещение информации на экране, дизайн экрана.

Заявки

Автозаселение

Информация

Настройки

Выход

Рис. 9 Эскиз главного меню системы

Назад Информация Найти Отчёт

Колонка ▾ Текст Колонка ▾ Текст Колонка ▾ Текст

Комната	ФИО	Пол	Категория	Форма	Договор

Рис. 10 Эскиз окна с информацией о заселенных студентах

Назад

Подробная информация

ФИО

Пол

Категория

Форма

Группа

Договор

Комната

Изменить

Удалить

Рис. 11 Эскиз формы для изменения данных студента

4.3. Особенности ввода информации пользователем, представление выходных данных.

Данные импортируются из файла (*.xlsx) в БД или вводятся через диалоговое окно информационной системы с помощью клавиатуры.

Данные выводятся на экран системы из БД, так же есть возможность формирования отчёта (*.docx).

5. Требования к документированию.

5.1. Перечень сопроводительной документации.

Руководство пользователя информационной системой «Hostel Tutor».

5.2. Требования к содержанию отдельных документов.

Не предусмотрено.

6. Порядок сдачи-приемки продукта.

Продукт разработки считается принятым при его положительной оценке экспертами-преподавателями ВУЗа.

Глава 2. Разработка информационной системы

2.1 Модельные представления информационной системы

Перед началом разработки информационной системы следует спроектировать ее функциональную модель, чтобы точно обозначить все входные и выходные данные, управления и механизмы.

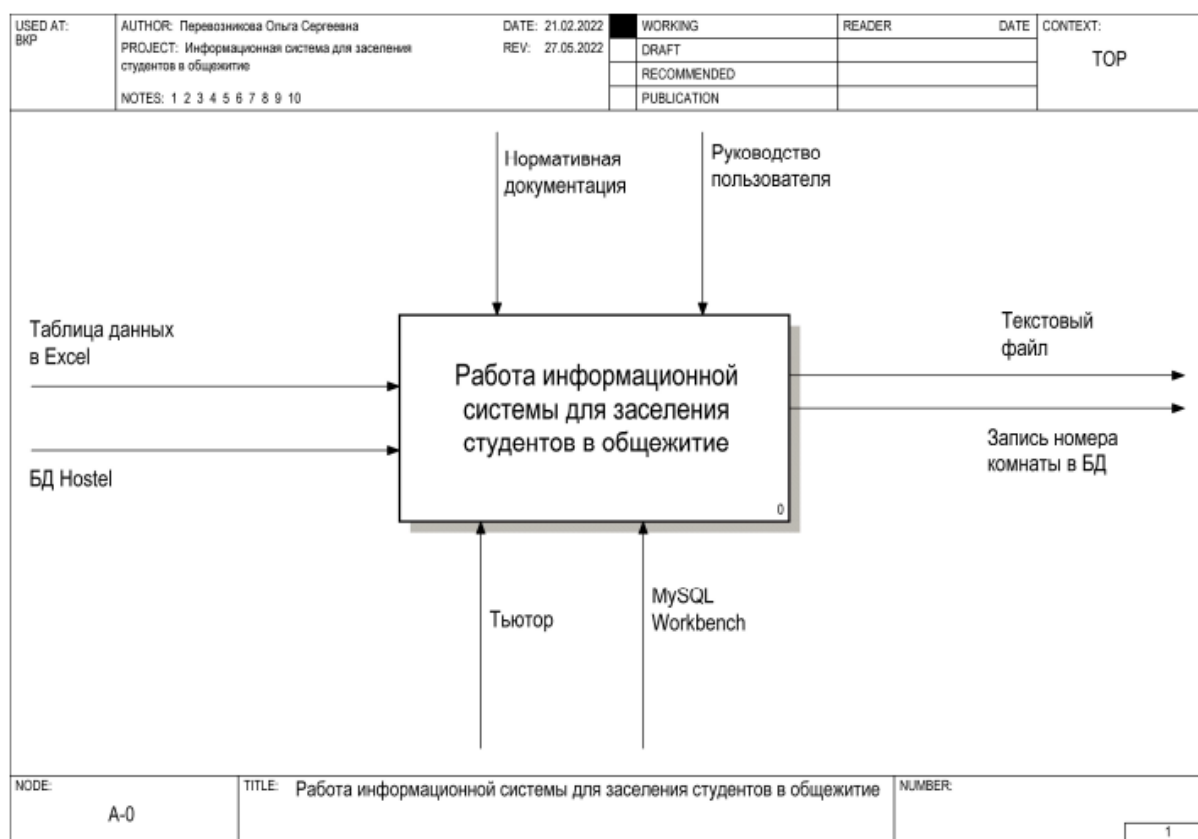


Рис. 12. Функциональная модель информационной системы

Спроектированная функциональная модель (Рис. 12) показывает работу информационной системы для заселения студентов в общежитие: на вход подаются таблица данных в формате Microsoft Excel и БД «Hostel», на выходе система записывает в БД номер комнаты и может представить их в виде отчёта в формате текстового файла (*.docx), механизмами выступают тьютор и приложение MySQL Workbench, управлениями – нормативная документация и руководство пользователя.

Следующим этапом будет построение диаграммы декомпозиции, основанной на построенной ранее модели, чтобы определить организацию функционального взаимодействия внутри системы.

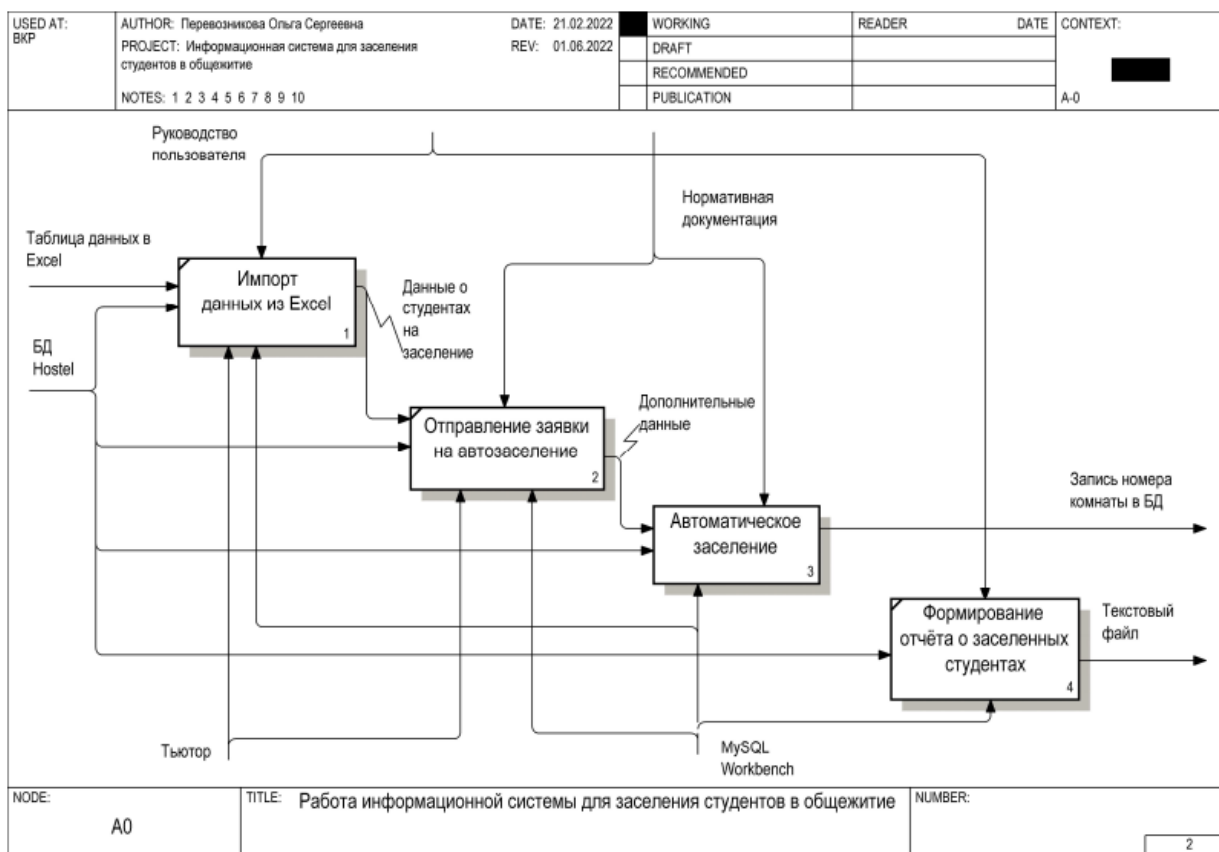


Рис. 13. Диаграмма декомпозиции первого уровня

На диаграмме декомпозиции (Рис. 13) было выявлено четыре основных функций:

- импорт данных из файла;
- отправление заявки на автозаселение;
- автоматическое заселение;
- формирование отчёта.

Можно заметить, что диаграмма демонстрирует связь всех функций и определяет передаваемые данные между ними, поэтому их стоит рассмотреть подробнее. После подачи тьютором на вход таблицы данных в формате (*.xlsx) происходит импортирование данных. Далее информация о студентах на

заселение формируется в заявки и отправляется на этап проверки. После того, как тьютор проверит и дополнит данные, он отправит их на автозаселение. На этапе автозаселения происходит поиск комнаты, который присваивается студенту и как следствие измененная информация отправится обратно в БД. Последним этапом является формирование отчёта, который составляется по желанию тьютора.

Функция автоматического заселения требует подробного описания происходящих в ней процессов, поэтому было принято решение детализировать функциональную диаграмму до второго уровня.

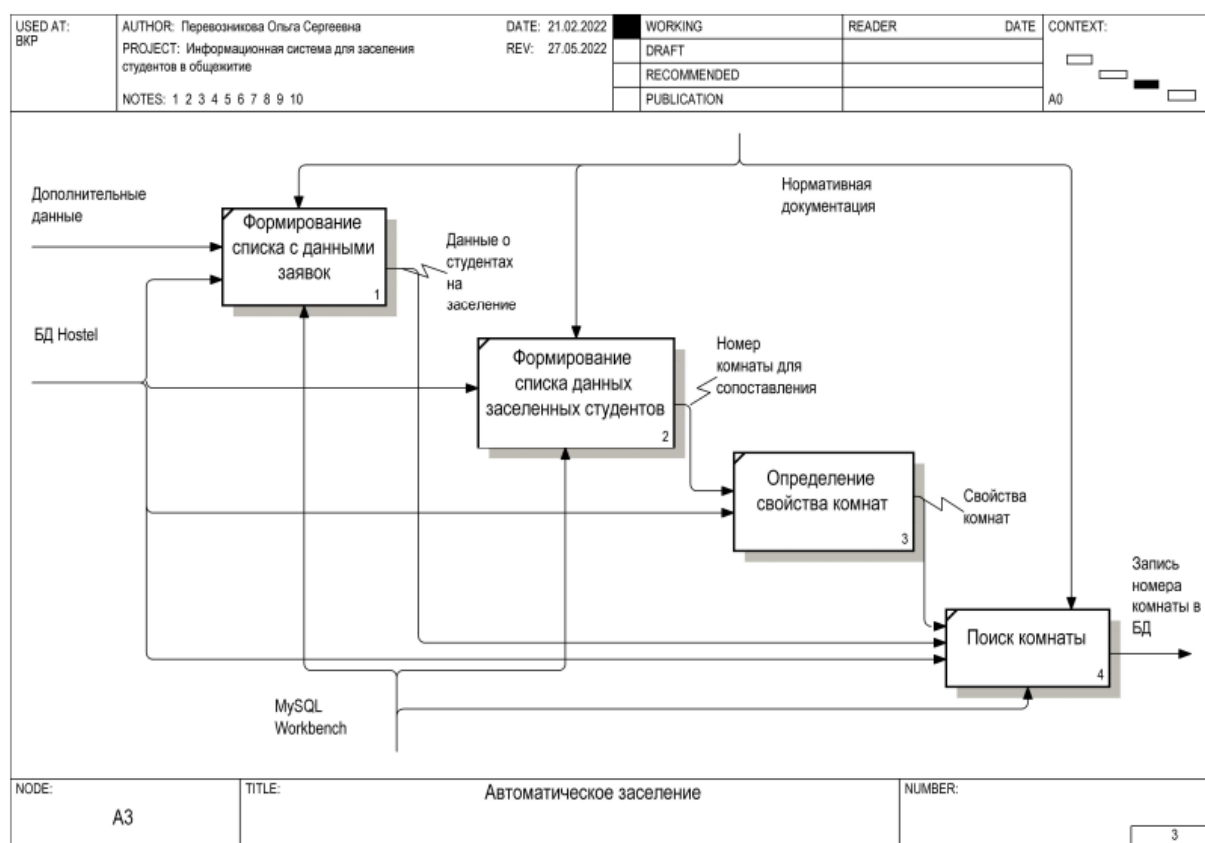


Рис. 14. Диаграмма декомпозиции второго уровня

Диаграмма декомпозиции второго уровня (Рис. 14) отображает работу автоматического заселения. Работа состоит из четырёх задач:

- формирование списка с данными из заявок;
- формирование списка данных заселённых студентов;

- определение свойств комнат;
- поиск комнаты.

После того, как тьютор передаст дополненные и изменённые заявки, сформируется список с важными для заселения данными. Затем список отправляется на этап для поиска комнаты. Одновременно с этим, формируется список информации о студентах, которые уже заселены в общежитие. Второй список передается на этап определения свойств комнат, где номер комнаты студента сравнивается с существующими номерами в общежитии и при совпадении – свойства студента присваиваются комнате. В последнем этапе поиска комнат сопоставляются данные студентов из заявок и свойства комнат. Номер комнаты, свойства которой будут максимально приближены к данным студента, присваивается данному студенту, и измененная информация отправляется в БД.

После построения диаграмм, на их основе, можно сформировать диаграмму классов UML, чтобы показать множество классов, необходимых для реализации функций, их связь, атрибуты и методы.

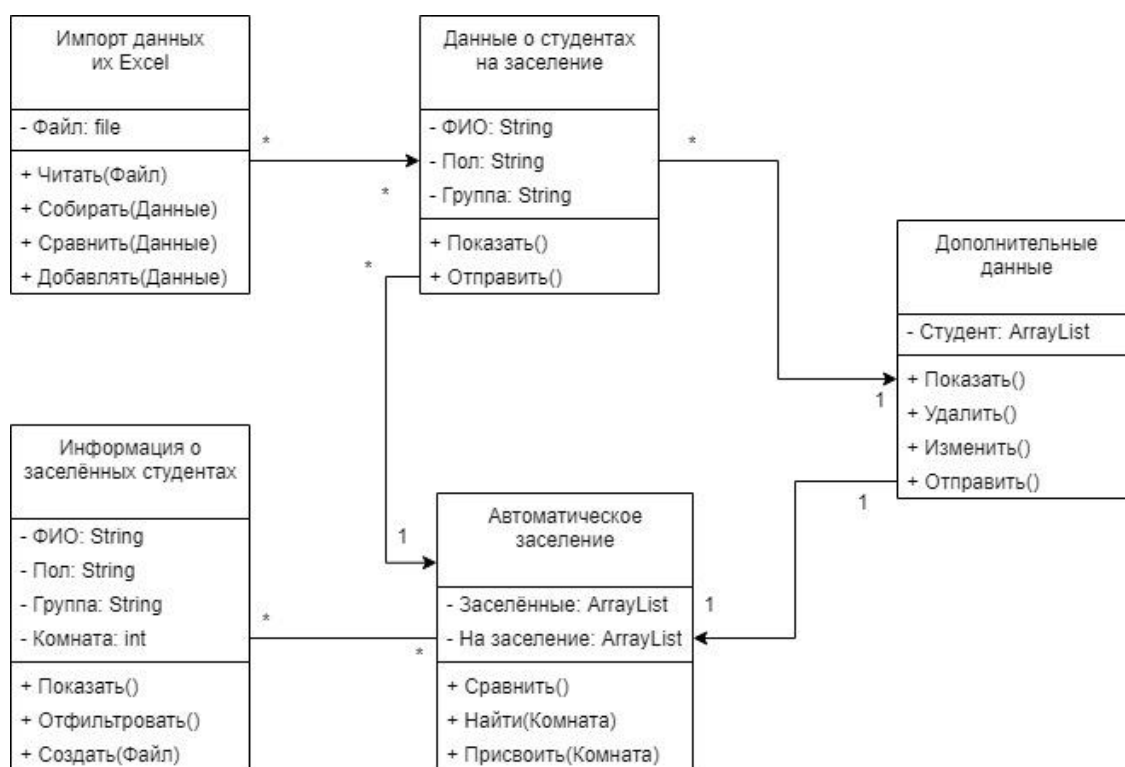


Рис. 15. UML Диаграмма классов

Диаграмма классов (Рис. 15) помогает понять структуру будущего кода программы. После построения диаграммы удалось определить методы, которые необходимо добавить в классы для увеличения вероятности достижения цели разработки. Так в класс «Импорт данных» был добавлен метод «Сравнить (Данные)», для того чтобы исключить ошибку случайного добавления уже имеющихся данных. Для формирования отчёта в классе «Информация о заселённых студентах» был добавлен метод «Создать (Файл)». Метод «Отфильтровать» упростит вывод данных о конкретных студентах и, к тому же, ускорит поиск информации о них.

Таким образом, благодаря функциональной диаграмме, диаграммам декомпозиции и UML диаграмме классов, была сформирована структура разрабатываемой информационной системы.

2.2 Описание работы информационной системы

На основе построенных диаграмм, представленных в п.2.1 Модельные представления информационной системы, была разработана система для заселения студентов в общежитие. Для работы с системой была создана база данных общежития «Hostel», в которую были добавлены таблицы, с необходимыми данными. На Рис. 16 можно увидеть базу данных, таблицы в ней и пример заполнения данных. На Рис. 17 можно увидеть пример связи таблицы «Группа» с таблицей «Направление».

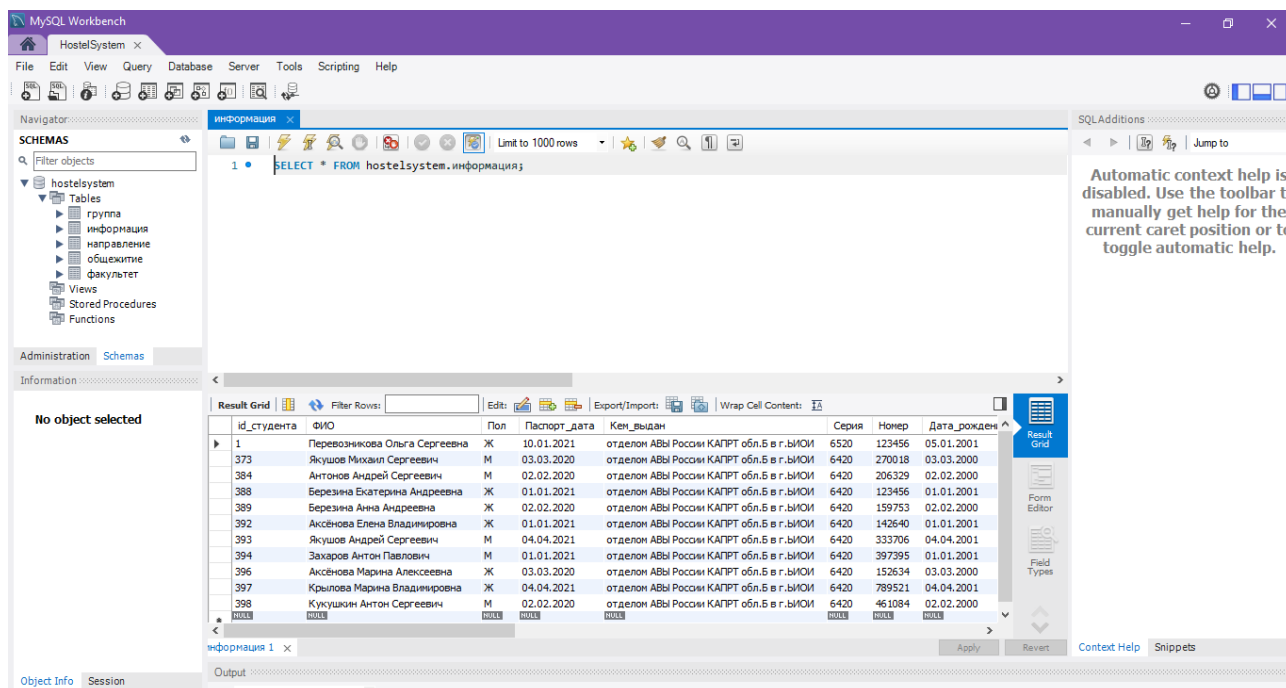


Рис. 16. База данных «Hostel». Таблицы «группа», «информация», «направление», «общежитие», «факультет»

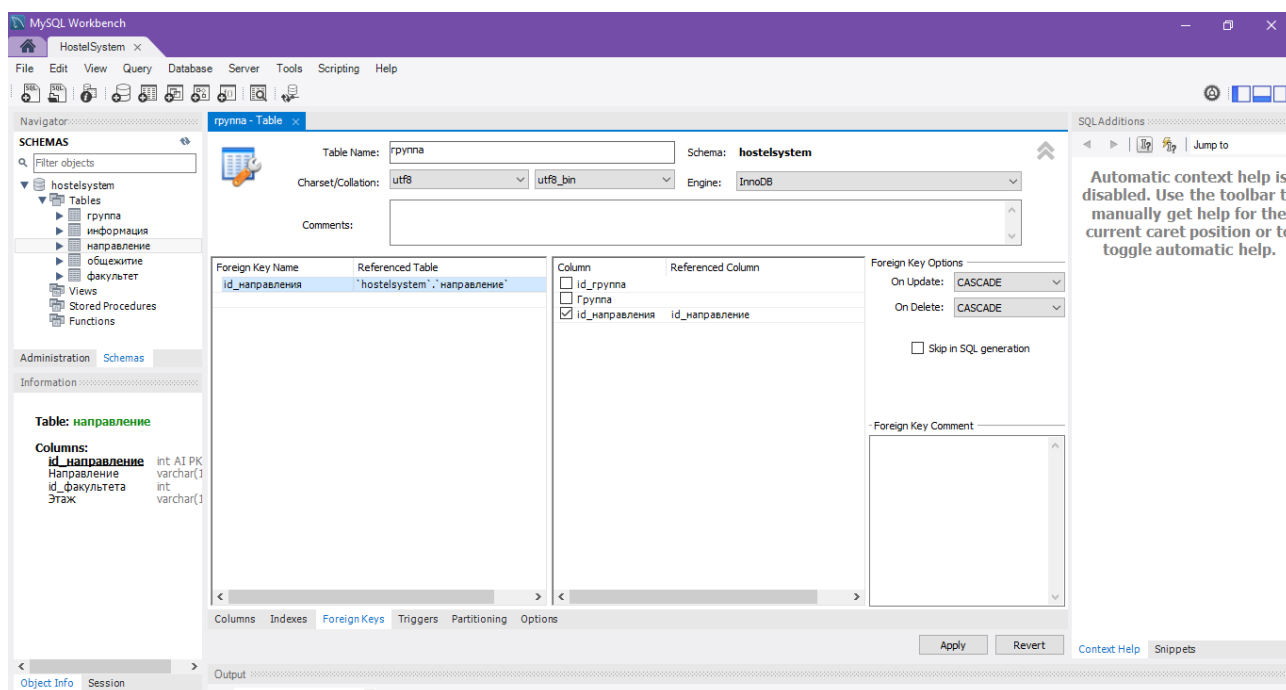


Рис. 17. Связь таблицы «группа» с таблицей «направление»

Рассмотрим пользовательский интерфейс информационной системы. В Scene Builder был разработан дизайн приложения, основанный на эскизах,

представленных в техническом задании. На Рис. 18 показано главное меню системы.

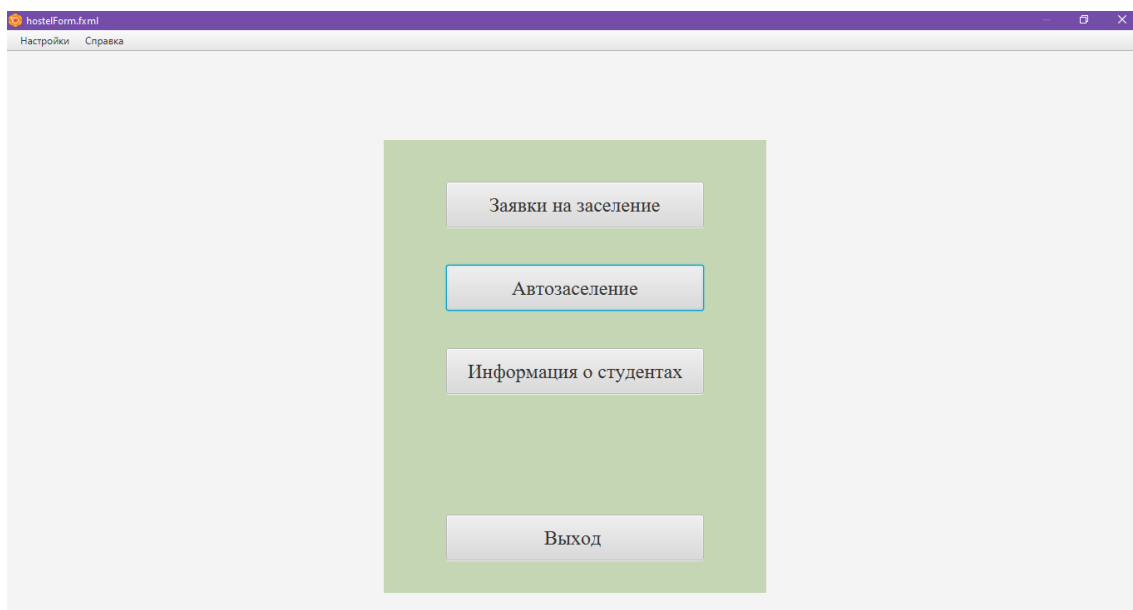


Рис. 18. Дизайн главного меню системы

Здесь можно:

- проверить состояние заявок на заселение;
- запустить процесс автозаселения;
- просмотреть информацию о заселённых студентах;
- выйти из системы.

Из названия последней кнопки становится понятно, что она производит выход из программы, а функционал остальных кнопок рассмотрим далее.

Первая кнопка «Заявки на заселение» открывает окно с информацией о заявках, представленных в виде таблицы (Рис. 19). После нажатия на строку с информацией открывается форма для отправления заявки на заселение, с возможностью добавить дополнительные данные или удалить заявку из таблицы. Кнопка «Назад» расположена на экране для возврата в главное меню.

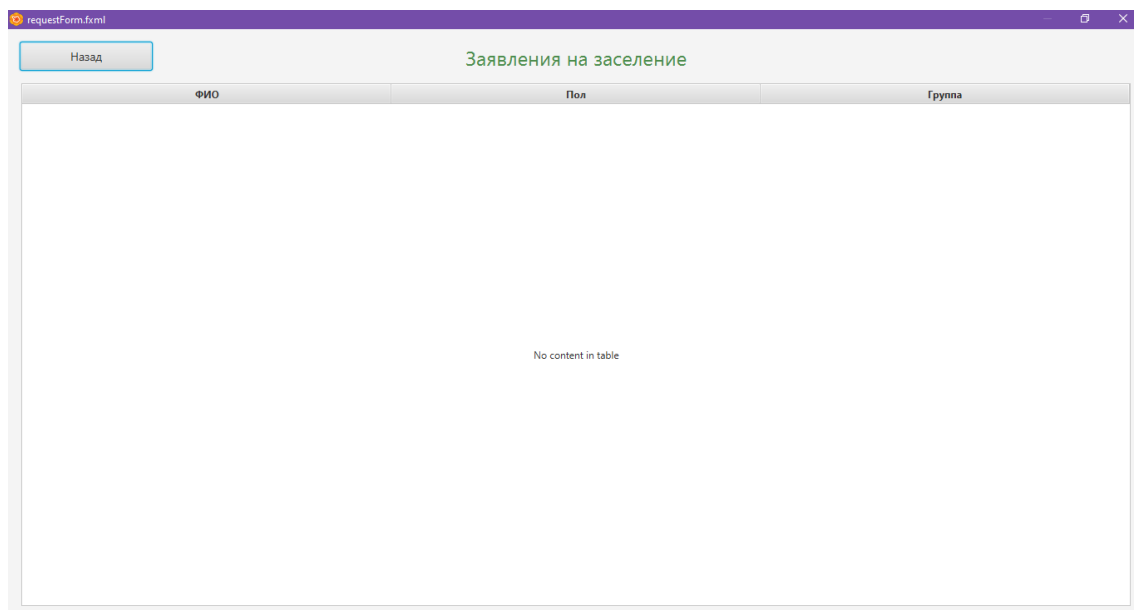


Рис. 19. Окно с информацией о заявках, представленных в виде таблицы

Фрагмент кода реализующий показ формы и переход данных, выбранной заявки, в нее:

```
«tableUsers.getSelectionModel().selectedItemProperty().addListener((
observable, oldValue, newValue) -> {
    try {
        showUserEditDialog(newValue);
    } catch (IOException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
});»
```

Форма для отправки заявки на заселение (Рис. 20) предоставляет возможность отметить место в блоке повышенного комфорта, куда возможно заселить студентов только женского пола. Также можно выбрать студентов из списка, с которыми, по желанию владельца заявки, стоит заселить при возможности. После добавления дополнительных данных заявку следует отправить на заселение, выбрав кнопку «Заселить», либо же удалить заявку,

выбрав кнопку «Удалить». Так же в форме есть кнопка для возврата к таблице заявок – «Назад».

Рис. 20. Форма для дополнения данных и отправки заявки на заселение

Фрагмент кода, регламентирующий возможность выбора места в блоке повышенного комфорта, основанный на половой принадлежности владельца заявки:

```
«if(user.getSx().compareTo("M")==0){
    lux.setDisable(true);    }»
```

Когда некоторое количество заявок будет отправлено за автоматическое заселение, в главном меню следует выбрать кнопку «Автозаселение». Рассмотрим функционал данной кнопки.

Сначала создаётся список, состоящий из некоторых данных о студентах на заселение, которые потребуются, чтобы подобрать более подходящую комнату. Список сортируется по порядку предоставлению комнаты, описанному в п.1.1 Специфика работы, который основан на принадлежности к какой-либо категории и/или форме обучения. Далее создаётся список комнат, в которые заселены студенты, чтобы определить для данных комнат свойства и внести их в БД, либо изменить, ранее присвоенные свойства. После этого происходит

подбор комнаты для каждого студента из списка на заселение, сравнивая свойства и учитывая критерии для заселения.

Рассмотрим фрагмент кода для подбора комнаты по свойствам, в данном случае по половой принадлежности и принадлежности к группе:

```
«for (int i = 0; i < strRoom.length; i++) {  
    String str1 = strRoom[i];  
    String[] strFind = str1.split(":");  
    String numFind = strFind[0];  
    String placeFind = strFind[1];  
    int kMax = Integer.parseInt(placeFind);  
    String sxFind = strFind[2];  
    String groupFind = strFind[3];  
    String[] group = groupFind.split(";");  
    ArrayList<String> groupS = new ArrayList<>();  
    for (int j = 0; j < group.length; j++) {  
        groupS.add(group[j]);    }  
    if ((sxFind.compareTo(sxNew) == 0) && (groupS.indexOf(groupNew)  
> 0) && (kMax >= d)) {  
        room = numFind;  
        numPeople++;  
        return room;}    }»
```

Когда находится подходящая комната, ее номер присваивается студенту и передается в БД.

Заселенных студентов можно увидеть в окне с информацией о заселённых студентах (Рис. 21). Чтобы это сделать необходимо выбрать кнопку «Информация о студентах» в главном меню. Данные в окне представлены в виде таблицы.

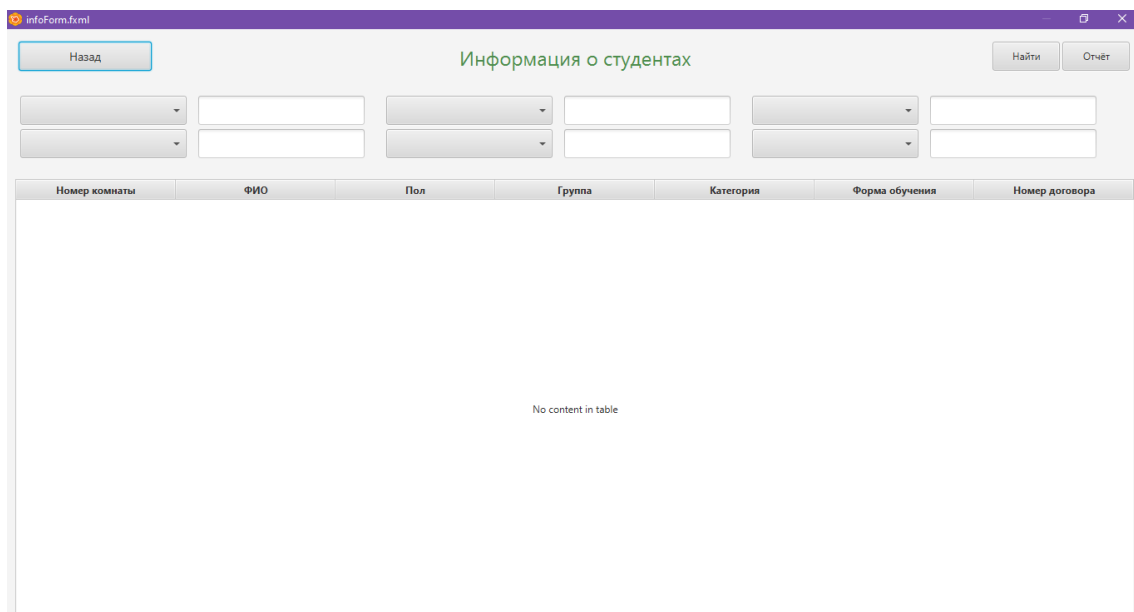


Рис. 21. Окно с информацией о заселённых студентах

Также есть возможность сортировки данных в таблице через нажатие по шапке столбца, по которому необходимо отсортировать. Отфильтровать данные на экране можно, выбрав из списка на экране параметр фильтрации и вписав в строку данные, по которым нужно отфильтровать.

Фрагмент кода, отвечающий за фильтр данных в таблице по нескольким критериям:

```
«if ((filter.compareTo(filterText) != 0)|| (filter1.compareTo(filterText1) !=
0)|| (filter2.compareTo(filterText2) != 0)|| (filter3.compareTo(filterText3) !=
0)|| (filter4.compareTo(filterText4) != 0)|| (filter5.compareTo(filterText5) != 0)) {
    select += " WHERE ";
    if (filter.compareTo(filterText) != 0) {
        if(a==1){
            select += "AND ";
        }
        select += filter + " LIKE '%" + filterText + "%'";
        a=1;
    }
    if (filter1.compareTo(filterText1) != 0) {
        if(a==1){
```

```

        select += "AND ";          }
select += filter1 + " LIKE '%" + filterText1 + "%'";
a=1;          }
if (filter2.compareTo(filterText2) != 0) {
    if(a==1){
        select += "AND ";          }
select += filter2 + " LIKE '%" + filterText2 + "%'";
a=1;          }
if (filter3.compareTo(filterText3) != 0) {
    if(a==1){
        select += "AND ";          }
select += filter3 + " LIKE '%" + filterText3 + "%'";
a=1;          }
if (filter4.compareTo(filterText4) != 0) {
    if(a==1){
        select += "AND ";          }
select += filter4 + " LIKE '%" + filterText4 + "%'";
a=1;          }
if (filter5.compareTo(filterText5) != 0) {
    if(a==1){
        select += "AND ";          }
select += filter5 + " LIKE '%" + filterText5 + "%'";
a=1;          } }»

```

Кнопка «Назад» возвращает пользователя в главное меню. Кнопка «Отчёт» формирует отчёт (Рис. 22), включающий в себя отфильтрованные данные из таблицы, продемонстрированные на экране в форматах (*.docx).

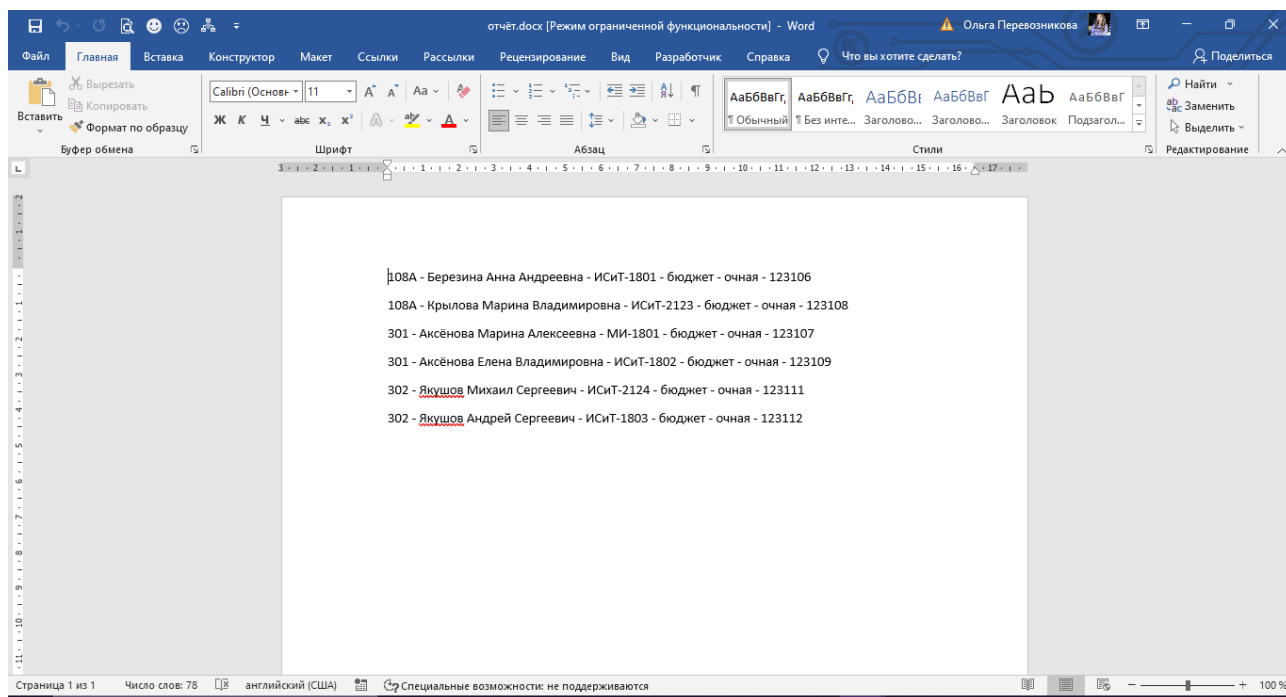


Рис. 22. Пример отчета в формате (*.docx)

Фрагмент кода, реализующий запись данных в файл для отчёта:

```
«File file = fileChooser.showSaveDialog(infoStage);
String name= String.valueOf(file);
try {
    XWPFDocument document = new XWPFDocument();
    for(int i=0; i<report.size(); i++){
        XWPFParagraph xwpfParagraph = document.createParagraph();
        xwpfParagraph.setAlignment(ParagraphAlignment.LEFT);
        XWPFRun xwpfRun = xwpfParagraph.createRun();
        xwpfRun.setText(report.get(i));    }
    FileOutputStream outputStream = new FileOutputStream(new
File(name));
    document.write(outputStream);
    outputStream.close();
    JOptionPane.showMessageDialog(null, "Отчёт успешно
сформирован и сохранён!", "Отчёт", JOptionPane.INFORMATION_MESSAGE);
}catch (Exception e){
```

```
e.printStackTrace();    }»
```

По нажатию на строку с информацией о студенте, открывается форма для изменения данных (Рис. 23). Форма несет в себе функцию не только изменения информации о выбранном студенте, а также просто ее демонстрацию в полном объеме. Кнопки, показанные в форме, имеют следующий функционал: «Изменить» – отправляет измененную информацию о данном студенте из формы в БД, «Удалить» – удаляет запись о студенте из БД, «Назад» – возвращает пользователя к таблице с информацией о заселённых студентах.

The screenshot shows a JavaFX window titled 'updateForm.fxml'. Inside, there's a header bar with a 'Назад' (Back) button and the title 'Информация о студенте'. The form is organized into several sections, each with a label on the left and input fields on the right:

- Личное** (Personal): Fields for 'ФИО' (Full Name), 'Дата рождения' (Date of Birth), 'Пол' (Gender) with a dropdown, and 'Адрес' (Address).
- Паспорт** (Passport): Fields for 'Серия' (Series), 'Дата выдачи' (Issue Date), 'Номер' (Number), and 'Кем выдан' (Issued by).
- Связь** (Contact): Fields for 'Номер телефона личный' (Personal Phone Number), 'Почта email' (Email), and 'Номер телефона родителей' (Parents' Phone Number).
- Общешитие** (General): Fields for '№ договора' (Contract Number) with the value '123456', 'Дата выдачи' (Issue Date), and 'Комната' (Room) with the value 'комната'.
- Обучение** (Education): Fields for 'Категория' (Category) and 'Группа' (Group) with dropdown menus, and 'Форма обучения' (Form of Education) with a dropdown menu.

At the bottom right of the form, there are two buttons: 'Изменить' (Update) in green and 'Удалить' (Delete) in red.

Рис. 23. Форма для изменения информации о заселённых студентах

Вернемся к окну главного меню и обратим внимание на графическое меню с настройками программы и справкой для пользователя. В настройках есть возможность импорта заявок из выбранного файла с расширением (*.xlsx), что упрощает ввод данных, делая его полностью автоматизированным. Также пункт «Добавить...» предоставляет возможность пользователю добавить некоторые необходимые данные (Рис. 24).

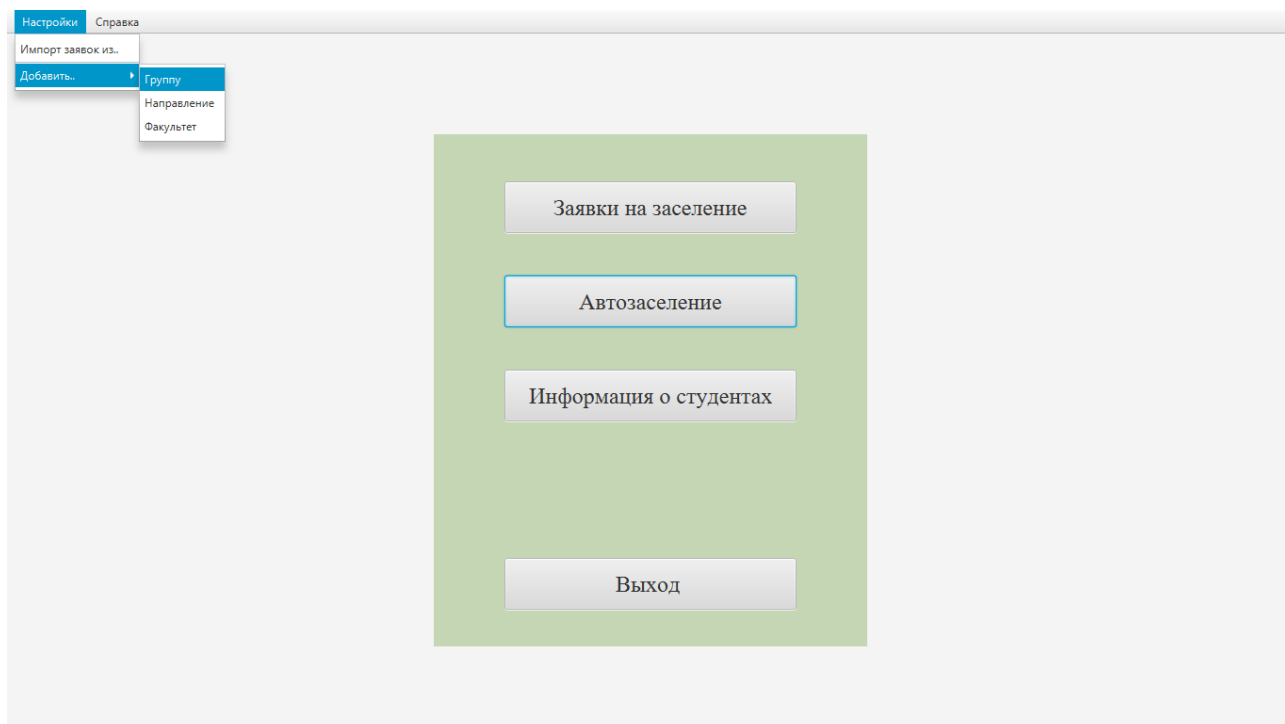


Рис. 24. Подпункт меню «Добавить..» и его составляющие

Таковыми данными являются: факультет, направление и группа. Окна для добавления данных параметров продемонстрированы ниже на рисунках (Рис. 25, Рис. 26, Рис. 27):

Рис. 25. Дизайн формы для добавления параметра «факультет» в систему

npForm.fxml

Добавить направление

Направление

Факультет

Этаж

Добавить

Рис. 26. Дизайн формы для добавления параметра «направление» в систему

groupForm.fxml

Добавить группу

Группа

Направление

Добавить

Рис. 27. Дизайн формы для добавления параметра «группа» в систему

В пункте меню «Справка» имеется подпункт «О программе...», при нажатии на который выводится окно с информацией о данной информационной системе (Рис. 28), а также подпункт «Руководство пользователя», при нажатии на который открывается руководство пользователя информационной системой «Hostel Tutor» в окне браузера.

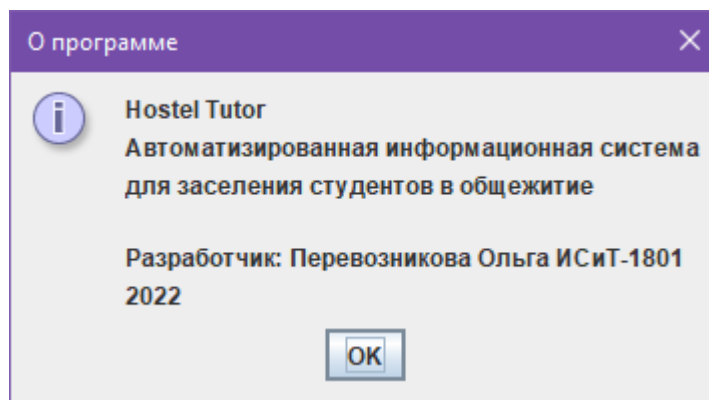


Рис. 28. Окно с информацией о данной информационной системе

Фрагмент кода реализующий показ руководства пользователя:

```
«if(Desktop.isDesktopSupported()){  
    try{  
        File dok = new File("C:/hostel/руководство.pdf");  
        Desktop.getDesktop().open(dok);  
    }catch (IOException ex){        } }»
```

Для написания кода были задействованы следующие библиотеки:

- Apache Poi – необходима для работы с файлами MS Office [3]. Из всего разнообразия компонентов использовался компонент «XSSF», он позволяет читать и записывать файлы нужного нам формата [2];
- JavaFX – необходима для разработки пользовательского интерфейса, включает в себя мультимедийные и графические пакеты [9].

2.3 Результаты апробации, техническая документация

Разработанная автоматизированная информационная система для заселения студентов в общежитие требовала создания руководства пользователя. Данный этап необходим для того, чтобы пользователь мог узнать точную последовательность использования продукта и, в случае ошибки, найти рекомендации по её устранению (Приложение 1).

Для того, чтобы открыть руководство пользователя, в системе нужно выбрать пункт меню «Справка», затем выбрать подпункт «Руководство пользователя».

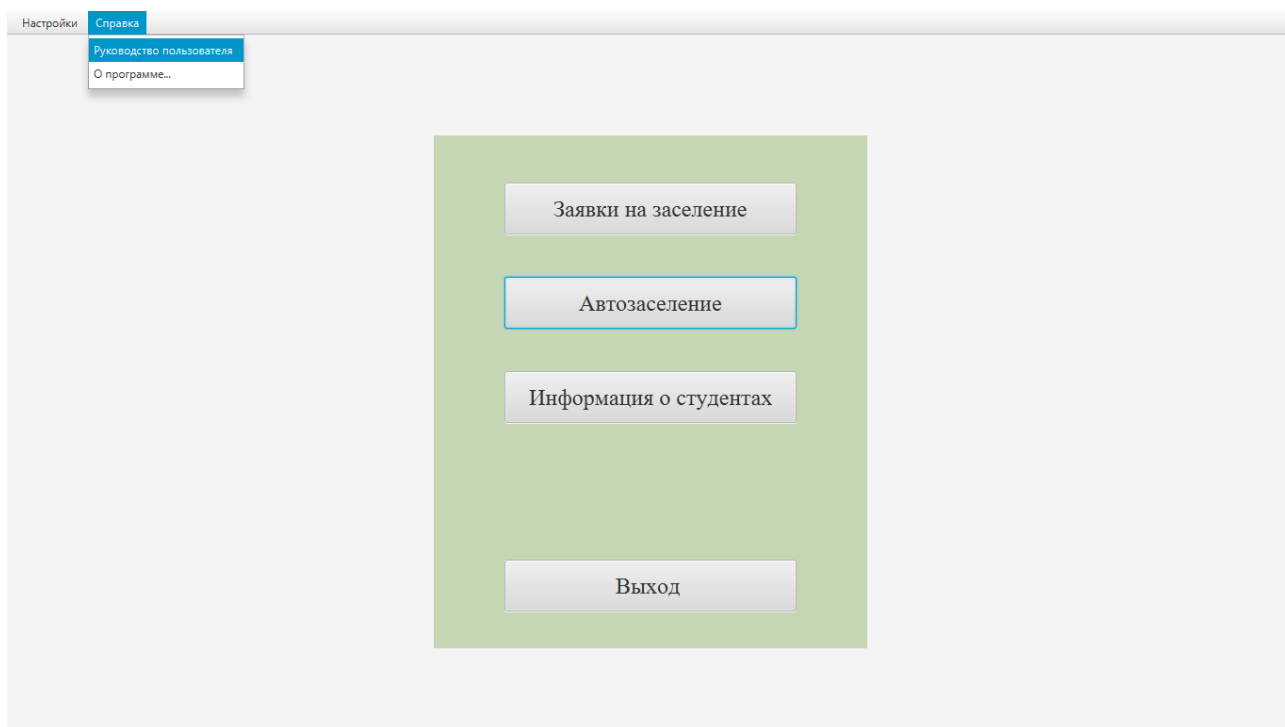


Рис. 29. Выбор подпункта «Руководство пользователя»

Руководство пользователя в виде документа формата (*.pdf) откроется в окне браузера.

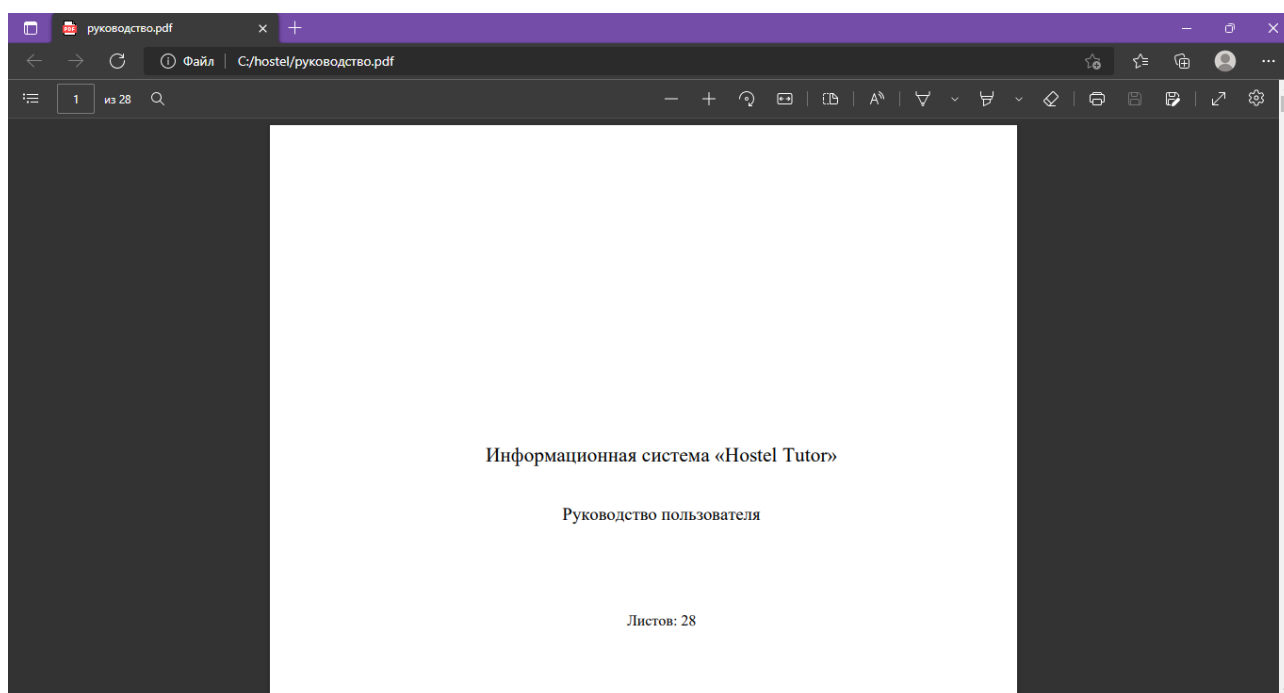
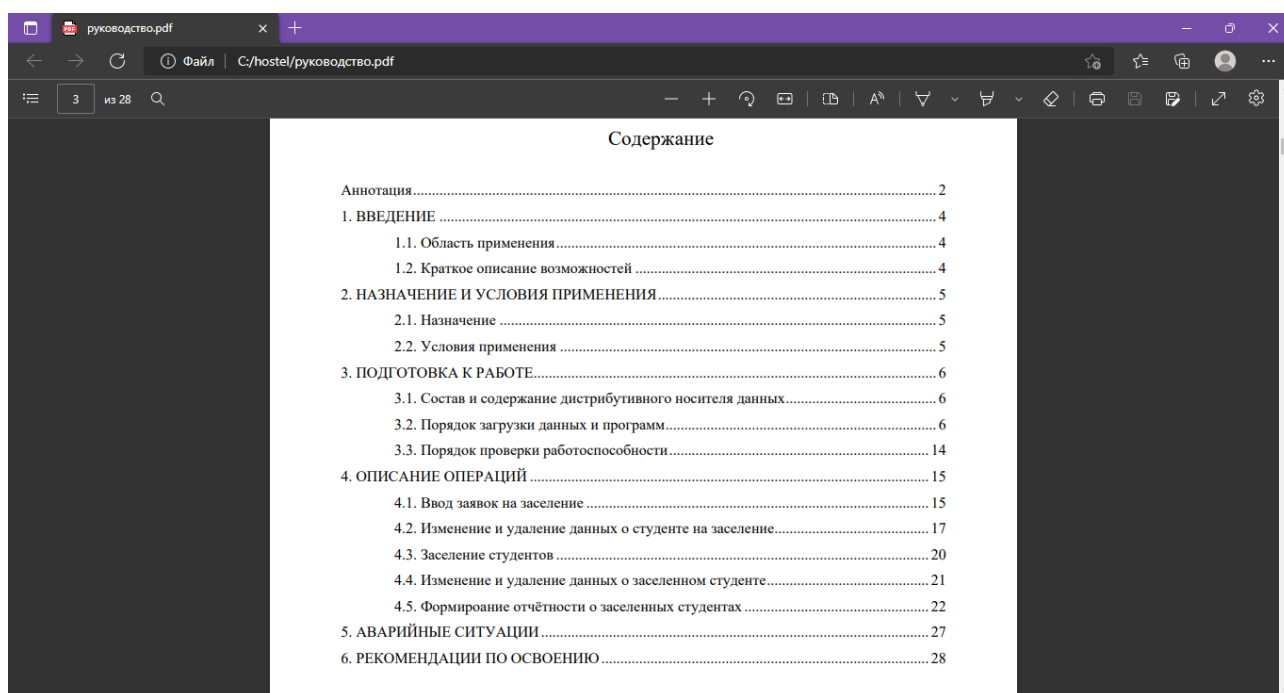


Рис. 30. Руководство пользователя информационной системы «Hostel Tutor»

Содержание включает в себя краткие сведения о системе, её возможности, назначение, условия применения. Так же подробно описана подготовка к работе, в которую входит состав дистрибутивного носителя и порядок загрузок программ для работы с системой. Описаны все операции и правила оформления документа для ввода его в систему. Руководство определяет порядок формирования отчетности о заселенных студентах, а также помогает найти решение при аварийной ситуации.



The image shows a PDF viewer window with a purple header. The title bar indicates the file is 'руководство.pdf'. The address bar shows the file path 'C:/hostel/руководство.pdf'. The viewer interface includes navigation buttons and a toolbar. The main content area displays the 'Содержание' (Table of Contents) of the document. The table lists sections and their corresponding page numbers.

Содержание	
Аннотация.....	2
1. ВВЕДЕНИЕ	4
1.1. Область применения.....	4
1.2. Краткое описание возможностей	4
2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ	5
2.1. Назначение	5
2.2. Условия применения	5
3. ПОДГОТОВКА К РАБОТЕ.....	6
3.1. Состав и содержание дистрибутивного носителя данных.....	6
3.2. Порядок загрузки данных и программ.....	6
3.3. Порядок проверки работоспособности.....	14
4. ОПИСАНИЕ ОПЕРАЦИЙ	15
4.1. Ввод заявок на заселение	15
4.2. Изменение и удаление данных о студенте на заселение.....	17
4.3. Заселение студентов	20
4.4. Изменение и удаление данных о заселенном студенте.....	21
4.5. Формирование отчетности о заселенных студентах	22
5. АВАРИЙНЫЕ СИТУАЦИИ.....	27
6. РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ	28

Рис. 31. Содержание Руководства пользователя информационной системы «Hostel Tutor»

Апробация продукта была произведена за счёт написания статьи в научный журнал [31]. В статье описывается план создания информационной системы в рамках выполнения выпускной квалификационной работы. За основу и как пример для статьи была взята созданная информационная система для заселения студентов в общежитие. В особенности разработанные диаграммы декомпозиции.

Министерство просвещения Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий

**АКТУАЛЬНЫЕ ВОПРОСЫ ПРЕПОДАВАНИЯ
МАТЕМАТИКИ, ИНФОРМАТИКИ
И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Межкузовский сборник научных работ

Екатеринбург 2022

УДК 37.016:004:37.016:51
ББК В1
А43

Рекомендовано Ученым советом федерального государственного бюджетного
образовательного учреждения высшего образования
«Уральский государственный педагогический университет»
в качестве научного издания (Решение № 46 от 23.05.2022)

Научный редактор
кандидат педагогических наук, доцент Л.В. Сардак
Уральский государственный педагогический университет

Члены редакционной коллегии
кандидат педагогических наук, доцент А.И. Газейкина
кандидат педагогических наук, доцент Т.И. Блинова
кандидат педагогических наук, доцент С.С. Арбузов
кандидат педагогических наук, доцент Н.А. Аппакумова
Уральский государственный педагогический университет

Рецензенты
доктор педагогических наук, профессор Б.Е. Стариченко,
кандидат педагогических наук, доцент И.И. Семенов
Уральский государственный педагогический университет

А43 Актуальные вопросы преподавания математики, информатики и информационных
технологий : межкузовский сборник научных работ / Уральский государственный
педагогический университет ; научный редактор Л.В. Сардак. – Электрон. дан. –
Екатеринбург : [б. и.], 2021. – 1 CD-ROM. – Текст : электронный.

ISBN 978-5-7186-1689-7

Сборник содержит материалы исследовательских и внедренческих работ, проведен-
ных преподавателями, сотрудниками, аспирантами, магистрантами, студентами российских
университетов и педагогов в области методики преподавания математики, информатики, а
также теории и практики применения информационно-коммуникационных технологий в об-
разовательном процессе вузов и школ.
Материалы адресованы преподавателям, аспирантам и студентам педагогических ву-
зов, а также учителям школ.

УДК 37.016:004:37.016:51
ББК В1

ISBN 978-5-7186-1689-7

© ФГБОУ ВО «УрГПУ», 2022

Рис. 32. Сборник научных работ «Актуальные вопросы преподавания математики, информатики и информационных технологий»

Веймер Н. В., Газейкина А. И. ВЫБОР ОНЛАЙН-СЕРВИСА ДЛЯ СОЗДАНИЯ ОБУЧАЮЩИХ ТЕСТОВ ПО ИНФОРМАТИКЕ.....	194
Газейкина А. И., Докрынина Е. Д. ОБЗОР ПЛАТФОРМ ДЛЯ СОЗДАНИЯ И РЕАЛИЗАЦИИ ОБРАЗОВАТЕЛЬНЫХ ВЕБ- КВЕСТОВ.....	200
Газейкина А. И., Мисюкованов А. Д. МЕТОДИКА ОБУЧЕНИЯ БУДУЩИХ УЧИТЕЛЕЙ ПРИМЕНЕНИЮ ТЕХНОЛОГИИ СКРАЙБИНГА В УЧЕБНОМ ПРОЦЕССЕ ШКОЛЬНИКОВ.....	206
Глухов И. А., Стариченко Б. Е. СИСТЕМА ГЕНЕРАЦИИ И ПРОВЕРКИ ВАРИАНТОВ ЗАДАНИЙ ПО ШКОЛЬНОМУ КУРСУ ФИЗИКИ.....	214
Голыгина Е. С., Семенов И. И. СРЕДСТВА ФОРМИРОВАНИЯ УМЕНИЯ ОПРЕДЕЛЯТЬ СВЯЗИ МЕЖДУ ЧАСТЯМИ В ПРОЦЕССЕ РАБОТЫ С СЮЖЕТНЫМИ ЗАДАЧАМИ.....	222
Дубовкин С. В. ДИДАКТИЧЕСКИЕ ВОЗМОЖНОСТИ СРЕДСТВ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ПРИ ОБУЧЕНИИ МАТЕМАТИКЕ СТУДЕНТОВ МЕДИЦИНСКОГО КОЛЛЕДЖА.....	227
Епаринцев М. Ю., Стариченко Б. Е., Шакирова А. А. ИСПОЛЬЗОВАНИЕ ЧАТ-БОТОВ ПРИ ОБУЧЕНИИ ИНФОРМАТИКЕ СТУДЕНТОВ МЕДИЦИНСКОГО КОЛЛЕДЖА.....	236
Епаринцев М. Ю., Стариченко Б. Е., Шакирова А. А. МЕТОДЫ ОБЕСПЕЧЕНИЯ ИНДИВИДУАЛИЗАЦИИ ОБУЧЕНИЯ МАТЕМАТИКЕ В МЕДКОЛЛЕДЖЕ НА ОСНОВЕ МОБИЛЬНЫХ ТЕХНОЛОГИЙ.....	242
Ерондаева Ю. В., Кошарова В. Е., Рожина И. В. ФОРМИРОВАНИЕ МЕТАПРЕДМЕТНЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ У ШКОЛЬНИКОВ 5-6 КЛАССОВ НА УРОКАХ ПО ПРОГРАММИРОВАНИЮ В СРЕДАХ KODU И SCRATCH.....	251
Зарыных Н. Ф., Шимов И. В. РАЗРАБОТКА УЧЕБНЫХ ЗАНЯТИЙ НА БАЗЕ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ВИРТУАЛЬНОГО ИСПОЛНИТЕЛЯ CONSTRUCT 2.....	258
Замкина А. Е., Шимов И. В. РАЗРАБОТКА УЧЕБНЫХ ЗАНЯТИЙ НА БАЗЕ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ВИРТУАЛЬНОГО ИСПОЛНИТЕЛЯ GAME EDITOR.....	265
Зубарева П. Д., Шимов И. В. ИСПОЛЬЗОВАНИЕ ЭЛЕМЕНТОВ РАЗРАБОТКИ ИГР В ЭЛЕКТИВНОМ КУРСЕ ИНФОРМАТИКИ.....	272
Киричкова О. А., Рухавишинова М. Д. ГОЛОВОЛОМКИ КАК СРЕДСТВО РАЗВИТИЯ ИНТЕРЕСА К МАТЕМАТИКЕ.....	279
Киричкова О. А., Сабуров Э. НОРМАТИВНЫЕ ДОКУМЕНТЫ, РЕГУЛИРУЮЩИЕ ОБРАЗОВАТЕЛЬНЫЙ ПРОЦЕСС ПО МАТЕМАТИКЕ В ТУРКМЕНИСТАНЕ.....	283
Кокеева С. В., Кудрявцев А. В. ФОРМИРОВАНИЕ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ У СТУДЕНТОВ ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ РАБОТЫ МИКРОКОНТРОЛЛЕРА В СООТВЕТСТВИИ С ЗАДАНИЯМИ РАСТРАСНИЕМ.....	289
Коралева В. В., Ободкина Т. А. ТЕХНОЛОГИЯ ОБУЧЕНИЯ РЕШЕНИЮ ПРАКТИКО-ОРИЕНТИРОВАННЫХ ОПТИМИЗАЦИОННЫХ ЗАДАЧ ПО МАТЕМАТИКЕ.....	295
Ладуга О. О., Лисицкая В. А., Тимирова А. М. ТЕХНОЛОГИЯ СОЗДАНИЯ ЭЛЕМЕНТОВ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ДЛЯ ОБУЧЕНИЯ ШКОЛЬНИКОВ МАТЕМАТИКЕ И ИНФОРМАТИКЕ.....	300

6

Перевозникова О. С., Шимов И. В. РЕАЛИЗАЦИЯ ИНФОРМАЦИОННО-ТЕХНОЛОГИЧЕСКИХ ПРОЕКТОВ ПО ПОДГОТОВКЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ В РАМКАХ ВЫПОЛНЕНИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ.....	308
Петрова А. А., Степанова Т. А. ПРИМЕНЕНИЕ ТРАНСФОРМАЦИОННЫХ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ ДЛЯ РЕАЛИЗАЦИИ ПЕРВОНАЧАЛЬНОГО ОБУЧЕНИЯ ПРИ ИЗУЧЕНИИ ТЕМЫ «АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ» В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ.....	313
Плявини К. Д., Сардак Л. В. ОБУЧЕНИЕ БУДУЩИХ ДИЗАЙНЕРОВ ПРОТОТИПИРОВАНИЮ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ СРЕДСТВАМИ СПЕЦИАЛИЗИРОВАННЫХ ПРОГРАММ.....	321
Сардак Л. В., Соколов Е. В. МЕТОД ПРОЕКТОВ КАК ИНСТРУМЕНТ ФОРМИРОВАНИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ У СТУДЕНТОВ.....	326
Сардак Л. В., Терехова М. Т. ЦИФРОВАЯ ИНСТРУМЕНТАЛЬНАЯ ГРАМОТНОСТЬ МЕДИА-ДИЗАЙНЕРА.....	332
Сардак Л. В., Хохлова Е. А. ПОВЫШЕНИЕ ИНТЕРЕСА У УЧАЩИХСЯ ПРОФИЛЬНЫХ ХУДОЖЕСТВЕННЫХ КЛАССОВ К ИЗУЧЕНИЮ МАТЕМАТИКИ СРЕДСТВАМИ ЭЛЕКТИВНОГО КУРСА «МАТЕМАТИКА В КОМПЬЮТЕРНОЙ ГРАФИКЕ».....	338
Семенов И. В., Стариченко Б. Е. ГЕЙМИФИКАЦИЯ ПРИ ОБУЧЕНИИ ИНФОРМАТИКЕ В 7-8 КЛАССАХ ШКОЛЫ НА ОСНОВЕ ПЛАТФОРМЫ JOYTEKA.....	346
Тимирова А. М., Токмьянина М. В. ПРИМЕНЕНИЕ ДИДАКТИЧЕСКИХ ИГР ДЛЯ ФОРМИРОВАНИЯ ФУНКЦИОНАЛЬНОЙ МАТЕМАТИЧЕСКОЙ ГРАМОТНОСТИ ШКОЛЬНИКОВ.....	353
Тимирова А. М., Токмьянина М. В. ИСПОЛЬЗОВАНИЕ МУЛЬТИМЕДИЙНЫХ ТЕХНОЛОГИЙ В ОБУЧЕНИИ ШКОЛЬНИКОВ МАТЕМАТИКЕ.....	360

7

Рис. 33. Статья в сборнике «Реализация информационно-технологических проектов по подготовке информационной системы в рамках выполнения выпускной квалификационной работы»

Заключение

В рамках выпускной квалификационной работы была разработана автоматизированная информационная система для заселения студентов в общежитие. В процессе выполнения работы были выполнены следующие задачи:

1. Произведён анализ нормативной документации, регламентирующей специфику заселения студентов в общежитие.
2. Произведён патентный поиск существующих решений; проанализирован и обоснован выбор технологий реализации и необходимых программных платформ.
3. В соответствии с техническим заданием проведена разработка информационной системы для заселения студентов в общежитие.
4. Подготовлено руководство пользователя и проведена апробация.

Таким образом, следует считать, что результаты разработки соответствуют всем требованиям технического задания, все поставленные задачи выполнены, цель достигнута. Работа имеет законченный характер.

Список информационных источников

1. A Visual Layout Tool for JavaFX Applications. – Текст: электронный // Oracle: [сайт]. – URL: <https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html> (дата обращения: 10.04.2022).
2. Apache POI – Download Release Artifacts. – Текст: электронный // The Apache Software Foundation: [сайт]. – URL <https://poi.apache.org/download.html> (дата обращения: 19.04.2022).
3. Apache POI, взаимодействие с Excel. – Текст: электронный // Java онлайн для разработчиков: [сайт]. – URL: <https://java-online.ru/java-excel.xhtml> (дата обращения: 24.04.2022).
4. Bringing MySQL to the web. – Текст: электронный // phpMyAdmin: [сайт]. – URL: <https://www.phpmyadmin.net/> (дата обращения: 02.04.2022).
5. C++. – Текст: электронный // свободная энциклопедия Википедия: [сайт]. – URL: [https://ru.wikipedia.org/wiki/C%2B%2B#:~:text=C%2B%2B%20\(%D1%87%D0%B8%D1%82%D0%B0%D0%B5%D1%82%D1%81%D1%8F%20%D1%81%D0%B8%2D%D0%BF%D0%BB%D1%8E%D1%81%2D%D0%BF%D0%BB%D1%8E%D1%81\)%20%E2%80%94%D0%BA%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D1%8B%D0%B9%2C,%D0%BF%D0%BE%D0%B4%D0%B4%D0%B5%D1%80%D0%B6%D0%BA%D1%83%20%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%D0%BF%D0%BE%D1%82%D0%BE%D1%87%D0%BD%D0%BE%D1%81%D1%82%D0%B8%20%D0%B8%20%D0%B4%D1%80%D1%83%D0%B3%D0%B8%D0%B5%20%D0%B2%D0%BE%D0%B7%D0%BC%D0%BE%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D0%B8](https://ru.wikipedia.org/wiki/C%2B%2B#:~:text=C%2B%2B%20(%D1%87%D0%B8%D1%82%D0%B0%D0%B5%D1%82%D1%81%D1%8F%20%D1%81%D0%B8%2D%D0%BF%D0%BB%D1%8E%D1%81%2D%D0%BF%D0%BB%D1%8E%D1%81)%20%E2%80%94%D0%BA%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D1%8B%D0%B9%2C,%D0%BF%D0%BE%D0%B4%D0%B4%D0%B5%D1%80%D0%B6%D0%BA%D1%83%20%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%D0%BF%D0%BE%D1%82%D0%BE%D1%87%D0%BD%D0%BE%D1%81%D1%82%D0%B8%20%D0%B8%20%D0%B4%D1%80%D1%83%D0%B3%D0%B8%D0%B5%20%D0%B2%D0%BE%D0%B7%D0%BC%D0%BE%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D0%B8) (дата обращения: 10.04.2022).

6. Eclipse (среда разработки). – Текст: электронный // свободная энциклопедия Википедия: [сайт]. – URL: [https://ru.wikipedia.org/wiki/Eclipse_\(%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8\)](https://ru.wikipedia.org/wiki/Eclipse_(%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8)) (дата обращения: 10.04.2022).
7. How to view and edit table and column comments with MySQL Workbench ... – Текст и графика: электронный // Mavink: [сайт]. – URL: <https://mavink.com/post/99847904970F100CAA8559A430C544A0E1AM0705DD> (дата обращения: 26.05.2022)
8. IntelliJ Idea 2020.1 x64. – Текст и графика: электронный // electronics-workbench.ru: [сайт]. – URL: <https://electronics-workbench.ru/intellij-idea-2020-1-x64> (дата обращения: 26.05.2022)
9. Introduction to FXML. – Текст: электронный // Oracle: [сайт]. – URL: https://docs.oracle.com/javase/8/javafx/api/javafx/fxml/doc-files/introduction_to_fxml.html (дата обращения: 10.04.2022).
10. Java Database Connectivity. – Текст: электронный // свободная энциклопедия Википедия: [сайт]. – URL: https://ru.wikipedia.org/wiki/Java_Database_Connectivity (дата обращения: 20.04.2022).
11. MySQL Installer (Archived Versions). – Текст: электронный // MySQL Product Archives: [сайт]. – URL: <https://downloads.mysql.com/archives/installer/> (дата обращения: 21.02.2022).
12. MySQL Workbench. – Текст: электронный // свободная энциклопедия Википедия: [сайт]. – URL: https://ru.wikipedia.org/wiki/MySQL_Workbench (дата обращения: 10.04.2022).
13. MySQL Workbench: визуальный дизайн базы данных. – Текст: электронный // MySQL: [сайт]. – URL: <https://www.mysql.com/products/workbench/design/> (дата обращения: 20.04.2022).

14. MySQL. – Текст: электронный // MySQL: [сайт]. – URL: <https://www.mysql.com/> (дата обращения: 05.04.2022).
15. Navicat for MySQL. – Текст: электронный // Navicat: [сайт]. – URL: <https://www.navicat.com/ru/products/navicat-for-mysql> (дата обращения: 02.04.2022).
16. Python. – Текст: электронный // свободная энциклопедия Википедия: [сайт]. – URL: <https://ru.wikipedia.org/wiki/Python> (дата обращения: 02.04.2022).
17. Sex segregation. – Текст: электронный // The free encyclopedia Wikipedia: [сайт]. – URL: https://en.wikipedia.org/wiki/Sex_segregation (дата обращения: 28.05.2022).
18. SQLyog. – Текст: электронный // свободная энциклопедия Википедия: [сайт]. – URL: <https://ru.wikipedia.org/wiki/SQLyog> (дата обращения: 03.04.2022).
19. Swing против JavaFx для настольных приложений [закрыто]. – Текст и графика: электронный // Projectbackpack: [сайт]. – URL: <https://rus.projectbackpack.org/214975-swing-vs-javafx-for-desktop-GXITJZ> (дата обращения: 26.05.2022)
20. What's this? – Текст: электронный // HeidiSQL: [сайт]. – URL: <https://www.heidisql.com/> (дата обращения: 03.04.2022).
21. Xcode. – Текст: электронный // свободная энциклопедия Википедия: [сайт]. – URL: <https://ru.wikipedia.org/wiki/Xcode> (дата обращения: 25.05.2022).
22. БИТ.Общежитие. – Текст: электронный // первый.Бит: [сайт]. – URL: <https://www.pulsar.ru/progs/1916/?ysclid=l3oufs53px> (дата обращения: 28.05.2022).
23. ГОСТ 19.101-77: Единая система программной документации. – Текст: электронный // URL: <https://docs.cntd.ru/document/1200007627?ysclid=l4bq7mwv4f451120460> (дата обращения 27.05.2022).
24. ГОСТ 34 РД 50-34.698-90: Автоматизированные системы. Требования к содержанию документов. – Текст: электронный // URL:

<https://docs.cntd.ru/document/1200006978?ysclid=l4bq3kyu8f844776732> (дата обращения 27.05.2022).

25. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы: межгосударственный стандарт: утв. и введ. в действие Постановлением Государственного комитета СССР по стандартам от 24 марта 1989 г. No 661: дата введ. 1990-01-01: переиздание июнь 2009 г. / разработан и внесен Государственным комитетом СССР по стандартам, Министерством приборостроения, средств автоматизации и систем управления СССР. – М.: Стандартиформ, 2009. – Текст: электронный // Электронный фонд правовых и нормативно-технических документов [сайт]. – URL: <https://docs.cntd.ru/document/1200006924> (дата обращения 17.01.2022).
26. ГОСТ Р 7.0.100 – 2018. Библиографическая запись. Библиографическое описание: общие требования и правила составления: нац. стандарт Рос. Федерации: изд. офиц.: утв. и введ. в действие Приказом Федер. агентства по техн. регулированию и метрологии от 3 дек. 2018 г. No 1050-ст: введ. впервые: дата введ. 2019-07-01 / разраб. Федер. гос. унитар. предприятием «Информ. телеграф. агентство России (ИТАР-ТАСС)» фил. «Рос. кн. палата», Федер. гос. бюджет. учреждением «Рос. гос. б-ка», Фе-дер. гос. бюджет. учреждением «Рос. нац. б-ка». – Москва: Стандартиформ, 2018. – 128 с. – (Система стандартов по информации, библиотечному и издательскому делу). – Текст: непосредственный.
27. ГОСТ Р 7.0.100 – 2018: новые правила описания статей и книг. Примеры библиографического описания для списка литературы. Памятка. – Текст: электронный // URL: <http://spf-vshni.ru/upload/medialibrary/9df/PAMYATKA-i-PRIMERY-bibliograficheskogo-opisaniya-dlya-spiska-literatury.pdf> (дата обращения 21.01.2022).

28. Общая информация о JavaFX. – Текст: электронный // Java: [сайт]. – URL: <https://www.java.com/ru/download/help/javafx.html> (дата обращения: 10.04.2022).
29. Общежитие. – Текст: электронный // свободная энциклопедия Википедия: [сайт]. – URL: <https://ru.wikipedia.org/wiki/%D0%9E> (дата обращения: 20.05.2021).
30. Общежития. – Текст: электронный // Уральский государственный педагогический университет: [сайт]. – URL: <https://uspu.ru/student/studgorodok/> (дата обращения: 27.05.2022).
31. Перевозникова О.С., Шимов И.В. Реализация информационно-технологических проектов по подготовке информационной системы в рамках выполнения выпускной квалификационной работы // Актуальные вопросы преподавания математики, информатики и информационных технологий. – Текст: непосредственный: межвузовский сборник научных работ – Екатеринбург: [б. и.], 2022. – С. 308-313.
32. Положение о студенческом общежитии Екатеринбург 2019 год 66 стр. / Утверждено Приказом ректора от 24.06.2019. – Текст: непосредственный.
33. Скачать IntelliJ IDEA. – Текст: электронный // JetBrains: [сайт]. – URL: <https://www.jetbrains.com/idea/download/?ysclid=l4bpo2ubuk379613454#section=windows> (дата обращения: 23.02.2022).
34. Тьютор. – Текст: электронный // свободная энциклопедия Википедия: [сайт]. – URL: <https://ru.m.wikipedia.org/wiki/%D0%A2%D1%8C%D1%8E%D1%82%D0%BE%D1%80> (дата обращения: 05.04.2022).
35. Управляйте общежитием легко и просто вместе с «БИТ.Общежитие». – Текст: электронный // первый.Бит: [сайт]. – URL: <https://www.pulsar.ru/upload/iblock/214/2147c17bb916ff9a8637732ead69e7db.pdf> (дата обращения: 28.05.2022).

36. Учебная тетрадь по платформе NetBeans. – Текст: электронный // Apache NetBeans: [сайт]. – URL: <https://netbeans.apache.org/kb/docs/platform/> (дата обращения: 20.05.2022).
37. Федеральный закон "Об образовании в Российской Федерации" от 29.12.2012 № 273-ФЗ Ст. 34 в ред. от 16.04.2022.
38. Федеральный закон "Об образовании в Российской Федерации" от 29.12.2012 № 273-ФЗ Ст. 39 в ред. от 16.04.2022.
39. Федеральный закон "Об образовании в Российской Федерации" от 29.12.2012 № 273-ФЗ Ст. 36 в ред. от 16.04.2022.
40. Что такое IntelliJ IDEA? – Текст: электронный // JetBrains: [сайт]. – URL: <https://www.jetbrains.com/ru-ru/idea/features/> (дата обращения: 05.04.2022).
41. Что такое PHP? – Текст: электронный // PHP: [сайт]. – URL: <https://www.php.net/manual/ru/intro-what-is.php> (дата обращения: 10.04.2022).
42. Что такое технология Java и каково ее применение? – Текст: электронный // Java: [сайт]. – URL: https://www.java.com/ru/download/help/what-is_java.html#:~:text=Java%20%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D1%82%20%D1%81%D0%BE%D0%B1%D0%BE%D0%B9%20%D1%8F%D0%B7%D1%8B%D0%BA%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F,%D1%82%D0%B0%D0%BA%D0%B8%D1%85%20%D0%B2%D0%B5%D0%B1%2D%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2%20%D0%B8%20%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9%20%D1%83%D0%B2%D0%B5%D0%BB%D0%B8%D1%87%D0%B8%D0%B2%D0%B0%D0%B5%D1%82%D1%81%D1%8F (дата обращения: 02.04.2022).

Приложения

Приложение 1

Руководство пользователя

