

Муромский институт (филиал) федерального государственного
бюджетного образовательного учреждения высшего образования
«Владимирский государственный университет имени Александра
Григорьевича и Николая Григорьевича Столетовых»

Факультет Информационных технологий и радиоэлектроники
Кафедра Программной инженерии

КУРСОВАЯ РАБОТА

по Разработке кроссплатформенных приложений
(наименование дисциплины)

Тема Приложение «АИС общежития»

Руководитель

Кульков Я. Ю.
(фамилия, инициалы)

(оценка)

(подпись) (дата)

Члены комиссии

Студент ПИН-119
(группа)

(подпись) (Ф.И.О.)

Емельянов В. А.
(фамилия, инициалы)

(подпись) (Ф.И.О.)

(подпись) (дата)

Муром 2022

В данной курсовой работе разработана автоматизированная информационная система для общежития. В ходе выполнения курсовой работы произведен анализ и сбор требований к разрабатываемой программе. При разработке применены язык Java, платформа JavaFX, реляционная система управления базами данных Firebird, JDBC-драйвер Jaybird и система администрирования баз данных IBExpert. Приложение протестировано и полностью работоспособно. Полученный программный продукт можно применять по его прямому назначению.

In this course work, the automated information system for a dormitory has been developed. In the process of creating a course work, the collection and analysis of the requirements for the projected program was made. The development used the Java language, the JavaFX platform, the Firebird relational database management system, the Jaybird JDBC driver and the IBExpert database administration system. The program has been tested and fully functional. The resulting product can be used for its intended purpose.

Содержание

Введение	6
1 Анализ технического задания.....	7
2 Разработка алгоритмов.....	9
2.1 Разработка моделей данных.....	9
2.1.1 Концептуальная модель	9
2.1.2 Логическая модель	10
2.1.3 Физическая модель.....	10
2.2 Реализация комплексных процессов	11
2.3 Прочие алгоритмы.....	13
3 Руководство программиста.....	15
4 Руководство пользователя	18
5 Тестирование	20
Заключение	23
Список используемой литературы	24
Приложение 1. Модели данных.....	25
Приложение 2. Текст программы	28
Приложение 3. Снимки окон программы (скриншоты программы)	29
Приложение 4. Диаграмма состояний процесса заселения студента	36

					МИВУ.09.03.04-02.000 ПЗ		
Изм.	Лист	№ докум.	Подп.	Дата			
Разраб.		Емельянов В. А.			Приложение «АИС общежития»		
Пров.		Кульков Я. Ю.					
Н. контр.							
Утв.					МИВУ ПИ _Н -119		
					Лит.	Лист	Листов
						5	36

Введение

В современном мире существует большая тенденция на автоматизацию. Это связано с тем, что подобные мероприятия повышают удобство и скорость работы. Кроме того, отпадает необходимость в хранении большого количества аналоговых носителей информации, ведь в итоге всё помещается внутри даже самого обычного компьютера. Для данных целей разрабатываются специальные программы – автоматизированные информационные системы.

Темой курсовой работы является разработка автоматизированной информационной системы общежития.

Общежитие, как никакое другое предприятие нуждается в подобной автоматизации. Ведь помимо работников, которые могут устраиваться на работу и увольняться с неё, присутствует гораздо большее количество студентов, которых необходимо заселять, выселять и перемещать по комнатам. Также необходим учёт заселённости этих самых жилых помещений, учёт проведения технических работ и многое другое. Как видно, даже приблизительный объём информации достаточно велик, но при помощи автоматизации работа с ним упростится в разы.

Таким образом, целью курсовой работы является создание приложения, занимающегося хранением, сортировкой и удобным предоставлением данных, производимых общежитием, а также автоматизацией самых часто используемых процессов. Главной задачей становится качественная реализация вышеперечисленного.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подп.	Дата		

1 Анализ технического задания

Целью данной курсовой работы является создание автоматизированной информационной системы для общежития, следовательно, разработка должна обеспечивать следующие функциональные возможности:

- ведение учёта персонала;
- ведение учёта студентов;
- ведение учёта заселённости комнат;
- ведение учёта посещений;
- автоматизацию заселения и выселения студентов;
- автоматизацию выдачи и сдачи инвентаря;
- автоматизацию учёта нарушений;
- автоматизацию контроля за уборкой помещений.

Из этих требований можно сделать несколько выводов.

Во-первых, разработка будет использоваться строго сотрудниками, следовательно, нет необходимости в создании и поддержке интерфейса для обычных (совсем несведущих) пользователей.

Во-вторых, программе не нужен доступ в интернет, так как СУБД может быть установлена на том же устройстве, что и АИС, или же связь с ней может обеспечиваться по локальной сети (например, с сервером, расположенным в одном из помещений). Клиентская часть, располагающаяся на компьютере сотрудника, должна быть реализована в виде приложения. Его интерфейс должен быть прост и понятен, а также, по возможности, содержать подсказки для самых неочевидных и сложных функций.

Таким образом, мы приходим к выводу, что для хранения будет очень удобно использовать несколько различных связанных между собой таблиц. Это приводит нас к реляционным СУБД. И лучшим выбором для разработки будет являться ПО Firebird 4.0. Эта система поддерживает все необходимые для реализации вышеперечисленного функции, является бесплатной, обладает хорошей

поддержкой как различного программного и аппаратного обеспечения, так и среди специалистов по базам данных.

Клиентское приложение будет разработано на языке Java с использованием платформы JavaFX. Это позволит обеспечить поддержку мультиплатформенности при полном сохранении функционала.

2 Разработка алгоритмов

2.1 Разработка моделей данных

2.1.1 Концептуальная модель

Перед тем, как приступить к разработке, необходимо составить проект планируемой базы данных. Первым делом составляется концептуальная модель. Для этого необходимо выделить сущности, которые будут встречаться в проектируемой АИС, их атрибуты, а также установить связи между ними.

Одной из основных сущностей будет являться Студент. Его атрибутами будут Фамилия, Имя, Отчество, Фотография, Дата рождения, Номер зачётной книжки и Пол.

Студенты должны проживать в Комнатах. Комнаты должны характеризоваться Номером и Количеством мест. У связи Проживание можно выделить атрибуты Дата заселения и Дата выселения.

В общежитие могут приходить Посетители. Их атрибуты следующие: Фамилия, Имя, Отчество, Номер телефона, Дата и время посещения. Они должны быть связаны со Студентами, то есть, приходить к кому-то конкретному.

Студенты могут совершать Нарушения. Для них стоит записывать Дату и время совершения, Описание и Мету наказания. Связь между этими сущностями показывает, кто именно совершил проступок.

Общежитие выдаёт своим постояльцам некоторый Инвентарь. Эта сущность должна содержать информацию о Виде и Количестве. На связи со Студентом образуются атрибуты Дата выдачи, Дата сдачи, Количество.

Помимо Студентов, также стоит задача хранить список Сотрудников. Их атрибутами будут Фамилия, Имя, Отчество, Фотография, Должность и Трудоустроенность.

В общежитии должна проходить уборка общих помещений. Информация о ней должна быть помещена в сущность Уборка с атрибутами Дата и время уборки, Убранные помещения. Так как уборщики являются Сотрудниками, имеет смысл создать связь между этой сущностью и Уборкой.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подп.	Дата		

В результате была подготовлена полная концептуальная модель. Её можно найти в Приложении 1.1.

2.1.2 Логическая модель

На основе концептуальной модели данных можно построить логическую модель. Но первым шагом будет нормализация таблиц, полученных на предыдущем этапе. Для соответствия второй нормальной форме в каждую таблицу, отражающую сущность, были добавлены идентификаторы, а в каждую таблицу, отражающую связь – идентификаторы связанных таблиц. В остальном все таблицы изначально были созданы с учётом требований всех нормальных форм вплоть до четвёртой включительно.

Для создания логической модели данных необходимо отобразить все сущности и связи, значимые для заданной предметной области и введённые в предыдущей схеме. Модель должна отражать внутреннее устройство базы данных. Результат проектирования расположен в Приложении 1.2.

2.1.3 Физическая модель

Из логической модели данных напрямую произрастает физическая. Она должна увязать созданную логику и возможности СУБД Firebird. Именно в ней происходит выбор внутренних названий таблиц и полей, типов используемых данных и занимаемого каждой записью объёма памяти. Разработанная модель находится в Приложении 1.3. Курсивом на схеме выделены ключевые поля таблиц.

Кроме того, имеет смысл произвести оценку максимального размера занимаемой памяти. Имеющиеся данные представлены в таблице 1.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подп.	Дата		

Таблица 1

Расчет объема ПЗУ для хранения данных

Таблица	Размер записи, байт	Максимальное (оценочное) количество записей	Всего, Кбайт
violations	1014	3600	3650,4
committed_violations	8	10800	86,4
students	1000111	1200	1200133,2
inventory	39	1000	39
issued_inventory	28	10800	302,4
check_ins_outs	24	1200	28,8
rooms	12	500	6
visitors	121	3600	435,6
employees	1000148	500	500074
performed_cleanings	8	1095	8,76
cleanings	513	365	187,245
Итого:			1704951,805

Таким образом, при максимальном заполнении БД объем таблиц составит приблизительно 1,7 Гб. Расчёты в таблице велись для одного года функционирования приложения.

2.2 Реализация комплексных процессов

АИС разрабатываются для упрощения совершения каких-либо действий, занимающих большое количество времени при выполнении их вручную. Данный программный продукт не является исключением и содержит в себе реализации некоторых комплексных процессов.

Разработка предоставляет большое количество функций, связанных с работой со студентами. Первое действие, которое необходимо совершить со студентом – заселение. Его диаграмма состояний находится в Приложении 4.

Помимо заселения, студент также может быть удалён из базы данных. Хотя, делать это не рекомендуется, такая возможность присутствует, а диаграмма её состояний приведена на рисунке 1.

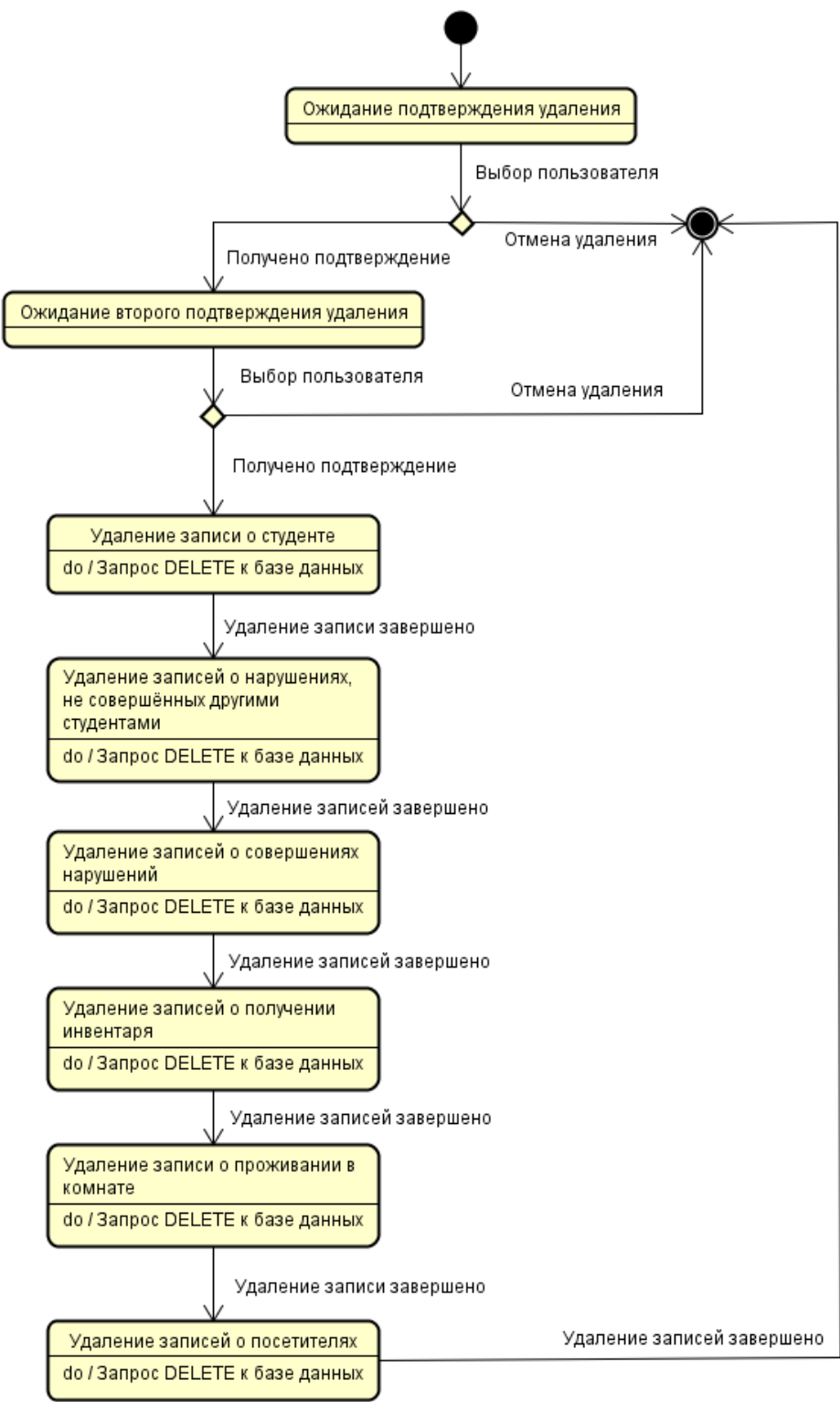


Рис. 1. Диаграмма состояний процесса удаления студента

2.3 Прочие алгоритмы

Интерфейс проектировался с учётом того, что им сможет пользоваться практически любой человек. Для этого кнопки имеют максимально развёрнутые описания происходящего при нажатии на них, текстовые поля содержат динамически изменяемые указания на выставленные ограничения по количеству символов, а в самых неочевидных местах расположены специальные элементы, при наведении на которые будет отображена справочная информация. Подробно ознакомиться с интерфейсом можно на скриншотах в Приложении 3.

Так как выбранная СУБД является реляционной, то для отображения данных из неё используется компонент TableView. Для предотвращения ошибок ввода реализовано множество проверок, как при помощи выбранного языка программирования, так и при помощи SQL-запросов. Так, во всех окнах, в которых можно вводить какие-либо данные, используются ограничивающие ввод компоненты (например, Spinner для чисел или DatePicker для дат). При загрузке изображений происходит проверка типа загружаемого файла по его заголовочным (первым четырём) байтам. В начале попытки заселения студента проверяется наличие свободных комнат для мужского и женского полов. Если свободные комнаты не найдены, то выдаётся предупреждение, а возможность заселения студента такого пола блокируется. Если же свободных комнат нет совсем, то заселение студента делается совершенно невозможным.

Для предотвращения попытки загрузки какой-либо другой БД, при запуске ведётся трёхступенчатая проверка загружаемого файла. На первом шаге проверяется, точно ли это файл, который можно открыть при помощи СУБД Firebird. Это позволяет отсеять повреждённые файлы или файлы с изменённым расширением. Вторым шагом проверяется наличие таблицы Students. Третьим шагом проверяется структура этой таблицы. При этом колонки должны идти в строго определённом порядке, а размеры полей, для которых указаны размеры типов данных, должны им соответствовать. Эти два шага в совокупности позволяют отсеять файлы, не относящиеся к данной разработке или, по какой-то

					МИВУ.09.03.04-02.000 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подп.	Дата		

причине, пытающиеся безуспешно маскироваться под них.

Для облегчения читаемости кода самые часто используемые вызовы диалоговых окон вынесены в отдельные классы со статическими методами.

При обычной работе у пользователя не должна возникать потребность в удалении записей из базы данных. Однако такая возможность присутствует в разработке, но АИС предупреждает пользователя о том, что такие действия нежелательны, а также показывает типы связанных данных, которые будут удалены вместе с выбранной записью.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подп.	Дата		

3 Руководство программиста

Приложение создано на языке Java версии 17 с использованием платформы JavaFX для создания интерфейса. В качестве СУБД используется Firebird 4.0. Для работы с ней использованы система администрирования IVExpert и JDBC-драйвер Jaybird 4.

При разработке использовался подход, схожий с паттерном MVC, что означает структурное разделение программы на четыре вида компонентов: модели, контроллеры и обычные скрипты, являющиеся классами и представления, являющиеся файлами разметки.

Контроллеры реализуют логику работы с представлениями. Следовательно, для каждого контроллера реализовано отдельное представление. Перечень контроллеров приведён в таблице 2.

Таблица 2

Перечень контроллеров

Имя контроллера	Краткое описание
CleaningInfoController	Реализует работу окна, позволяющего добавлять, изменять и удалять данные об уборках
EmployeeInfoController	Реализует работу окна, позволяющего добавлять, изменять и удалять данные о сотрудниках
InventoryGivenInfoController	Реализует работу окна, позволяющего добавлять, изменять и удалять данные о выданном инвентаре
InventoryInfoController	Реализует работу окна, позволяющего добавлять, изменять и удалять данные об инвентаре на складе
MainWindowController	Реализует работу главного окна
PhotoViewController	Реализует работу окна, показывающего фотографии, расположенные в БД

Таблица 2 (продолжение)

Имя контроллера	Краткое описание
MultifunctionalTable WindowController	Реализует работу окна, содержащего таблицу и элементы управления для отображения в дополнительных окнах (таких как, например, окна получения информации или выбора для изменения одного из множества элементов, связанных с выбранным на предыдущем окне элементом)
RoomInfoController	Реализует работу окна, позволяющего добавлять, изменять и удалять данные о комнатах
SettingsWindow Controller	Реализует работу окна с настройками приложения
StudentInfoController	Реализует работу окна, позволяющего добавлять, изменять и удалять данные о студентах
ViolationInfoController	Реализует работу окна, позволяющего добавлять, изменять и удалять данные о нарушениях студентов
VisitorInfoController	Реализует работу окна, позволяющего добавлять, изменять и удалять данные о посетителях

Контроллерам, реализующим окна для CRUD-операций, соответствуют одноимённые модели: например, контроллер – VisitorInfoController, модель – Visitors. Помимо этого, для удобства отображения данных в контроллере CleaningInfoController, была создана модель Cleaners (убиравшиеся сотрудники). Эти классы хранят данные об одной из основных сущностей базы данных (имеющей поле с уникальным идентификатором). В моделях Students и Employees также реализованы методы открытия окна с фотографией.

Помимо перечисленного выше, были созданы различные вспомогательные классы. Их перечень приведён в таблице 3.

Перечень вспомогательных классов

Имя класса	Краткое описание
DatabaseWorker	Класс, осуществляющий подключение к базе данных (с проведением проверок на соответствие подключаемого файла) и отключение от неё
DeleteMessage	Класс, отображающий сообщения, возникающие при удалении записи из БД
ErrorMessage	Класс, отображающий переданное сообщение об ошибке
FormatCheck	Класс, осуществляющий проверку загружаемого изображения на соответствие декларируемому типу файла
Program	Базовый класс, осуществляющий запуск программы
TableState	Перечисление, определяет действие, совершаемое в окне многофункциональной таблицы
TableType	Перечисление, определяет данные, добавляемые в окно многофункциональной таблицы

Программа является полностью функциональной, однако при желании может быть доработана. Например, можно ввести авторизацию различных пользователей, ведь в текущий момент администрированием занимается один человек, но некоторую часть работы по заполнению можно было бы делегировать непосредственным исполнителям работы (внесение информации об уборках доверить уборщикам, внесение информации об инвентаре – кладовщику и т. д.). Полезной будет реализация поиска по таблицам. Благодаря возможностям СУБД Firebird реальна реализация сетевого подключения к базе данных. Также при необходимости АИС может быть дополнена совершенно новыми функциями.

Полный код разработанной системы, включающий в себя реализацию всех описанных выше возможностей, расположен в Приложении 2.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						17
Изм.	Лист	№ докум.	Подп.	Дата		

4 Руководство пользователя

Минимальные системные требования:

1. Операционная система Windows 10 или любая UNIX-подобная;
2. Процессор с частотой 1 ГГц;
3. 2 Гб оперативной памяти;
4. Не менее 2 Гб места на жёстком диске;
5. Необходимо наличие монитора с разрешением не менее 1280x720 и устройств ввода (клавиатуры и мыши).

Требования к установленному ПО: необходимо наличие на компьютере JRE версии не ниже 17 (желательно с включенной в состав JavaFX) и СУБД Firebird 4.0.

Программа не требует установки. Для начала работы с приложением необходимо запустить файл DormitoryAIS.jar. Для этого в папке с программой необходимо создать файл с расширением «.cmd», внутри которого будет расположено следующее:

```
java --module-path . --add-modules javafx.controls,javafx.fxml  
-jar DormitoryAIS.jar
```

Далее необходимо два раза нажать на созданный файл, после чего программа запустится. На рисунке 2 показан внешний вид приложения после запуска.

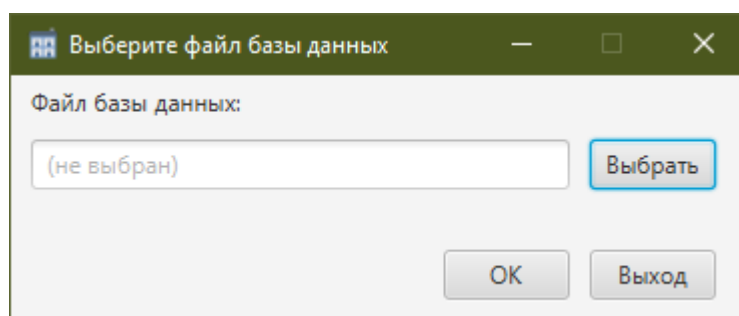


Рис. 2. Запущенное приложение

Для запуска основного функционала приложения необходимо загрузить в него файл базы данных с расширением «.fdb». После этого откроется главный

					МИВУ.09.03.04-02.000 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подп.	Дата		

экран, показанный на рисунке 3.

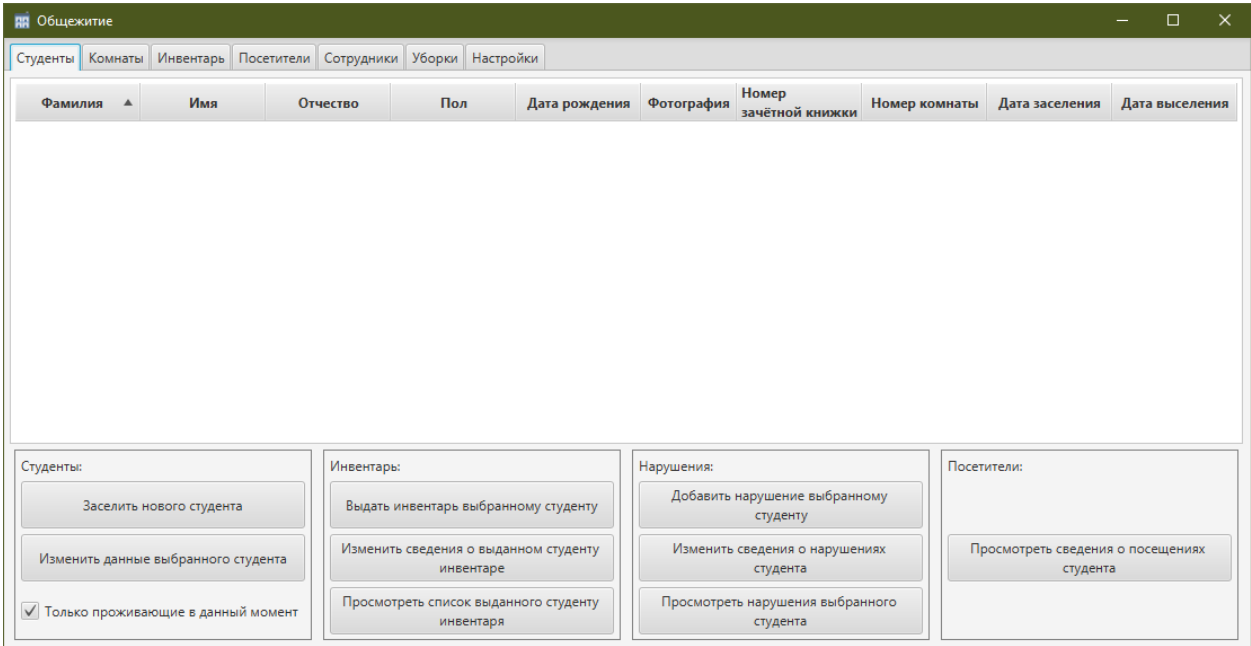


Рис. 3. Главный экран

Изначально приложение поставляется с пустой базой данных. Для добавления используются кнопки с соответствующими названиями. После нажатия на такую кнопку, открывается окно. В нём необходимо заполнить все поля, после чего нажать на кнопку подтверждения. Запись появится в таблице, после чего её можно будет изменять при помощи кнопки с соответствующим названием. Кнопки просмотра информации выводят дополнительное окно с запрошенной информацией или сообщение о её отсутствии.

В верхней части окна расположены вкладки, использующиеся для переключения между различными таблицами. При переключении на вкладку «Настройки» открывается окно с параметрами приложения.

5 Тестирование

Для проведения тестирования во все таблицы базы данных были добавлены записи. Имена и номера телефонов являются случайно сгенерированными, фотографии созданы при помощи нейросети, все использованные ресурсы находятся в открытом доступе.

Требования, подлежащие проверке во время испытаний, представлены в таблице 4.

Для проведения испытаний использовался компьютер с операционной системой Windows 10.

Таблица 4

Требования для проверки во время тестирования

Действие	Результат
Запуск программы	Появляется окно загрузки, после чего отображается основной интерфейс программы
Нажатие на кнопку «Заселить нового студента»	Открывается окно «Заселить нового студента»
Заселение студента с незаполненным полем «Имя»	Выводится сообщение об ошибке: «Не заполнено поле “Имя”»
Загрузка в раздел «Фотография» текстового файла, расширение которого изменено на «.png»	Выводится сообщение об ошибке: «Файл не является изображением»
Попытка инъекции SQL-запроса в поле «Мера наказания» окна «Добавить новое нарушение»	SQL-запрос воспринят как обычный текст и добавлен в новую запись в таблице нарушений

Таблица 4 (продолжение)

Действие	Результат
Удаление нарушения	Удалена запись о нарушении, а также все записи о его совершении
Добавление комнаты с номером, кратным 100	Выводится сообщение об ошибке: «Комната с таким номером не может быть добавлена»
Получение информации об обладателях инвентаря, который никому не выдавался	Выводится информационное сообщение: «Весь имеющийся инвентарь данного вида находится на складе»
Нажатие на кнопку «Найти посещаемого студента» на вкладке «Посетители»	Программа переходит на вкладку «Студенты» и выделяет запись со сведениями о посещаемом студенте
Нажатие на кнопку «Найти посещаемого студента» на вкладке «Посетители» при условии, что студент выселен	Выводится сообщение об ошибке, сообщающее о том, что студент выселен и для его поиска следует совершить указанные дополнительные действия
Двойное нажатие на ячейку таблицы основного окна	Открывается окно изменения, соответствующее вкладке
Нажатие на кнопку в столбце «Фотография»	Открывается окно с изображением
Запуск программы без файла базы данных	Функционал программы заблокирован до указания корректного пути к файлу базы данных; выход из программы доступен
Запуск программы на компьютере с СУБД Firebird версии, младшей, чем 4.0	Выводится сообщение об ошибке, сообщающее о неподдерживаемой структуре файла базы данных

Таблица 4 (продолжение)

Действие	Результат
Указание пути к аудиофайлу, расширение которого изменено на «.fdb»	Выводится сообщение об ошибке, сообщающее о том, что из данного файла невозможно прочитать базу данных
Нажатие на любую кнопку, кроме кнопки добавления, на вкладке с пустой таблицей	Выводится информационное сообщение: «Добавьте данные в таблицу на выбранной вкладке, чтобы воспользоваться этой функцией»
Попытка выдать инвентарь выселенному студенту	Выводится информационное сообщение: «Выбранный студент выселен, действие недоступно»
Попытка заселения студента при полном отсутствии свободных комнат	Выводится сообщение об ошибке: «В общежитии нет свободных комнат»
Закрытие программы	Программа закрывает соединение с базой данных, после чего все окна закрываются

Результаты, полученные в ходе тестирования, позволяют сделать заключение о том, что разработанная программа соответствует требованиям технического задания.

Заключение

В ходе данной курсовой работы была выполнена разработка автоматизированной информационной системы общежития.

В ходе выполнения были решены следующие задачи:

- возможность учёта студентов;
- возможность учёта персонала;
- возможность учёта заселённости комнат;
- возможность учёта посещений;
- возможность автоматизированного заселения и выселения студентов;
- возможность автоматизированной выдачи и сдачи инвентаря;
- возможность автоматизированного учёта нарушений;
- возможность автоматизированного контроля за уборкой помещений.

Разработанная программа может быть усовершенствована. Однако это не обязательно, ведь на выходе имеется полностью рабочий продукт, который уже можно использовать.

Таким образом, в курсовой работе были реализованы все пункты технического задания.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						23
Изм.	Лист	№ докум.	Подп.	Дата		

Список используемой литературы

1. Руководство по языку SQL СУБД Firebird 4.0 – URL: https://www.ibase.ru/files/firebird/Firebird_4_0_Language_Reference_RUS.pdf (дата обращения: 22.02.2022). – Текст: электронный.
2. Раздел о языке программирования Java на Metanit.com: сайт. – URL: <https://metanit.com/java/> (дата обращения: 01.03.2022). – Текст: электронный.
3. Типы данных [Словарик по FireBird]: сайт. – URL: http://firebirdsql.su/doku.php?id=tipy_dannyx (дата обращения: 06.04.2022). – Текст: электронный.
4. Stack Overflow: сайт. – URL: <https://stackoverflow.com/> (дата обращения: 26.05.2022). – Текст: электронный.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						24
Изм.	Лист	№ докум.	Подп.	Дата		

Приложение 1. Модели данных

1.1 Концептуальная модель

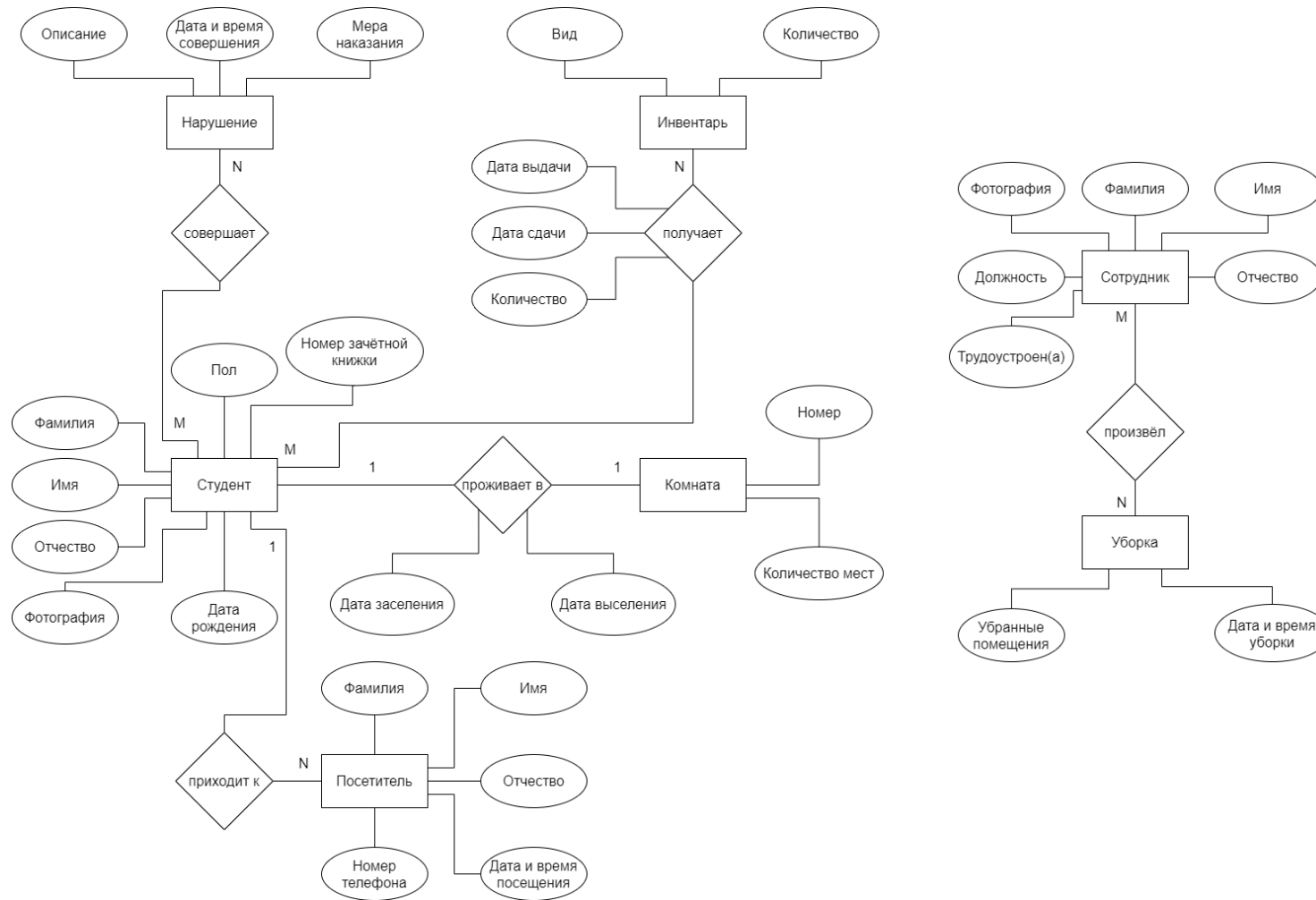


Рис. 1. Концептуальная модель

1.2 Логическая модель

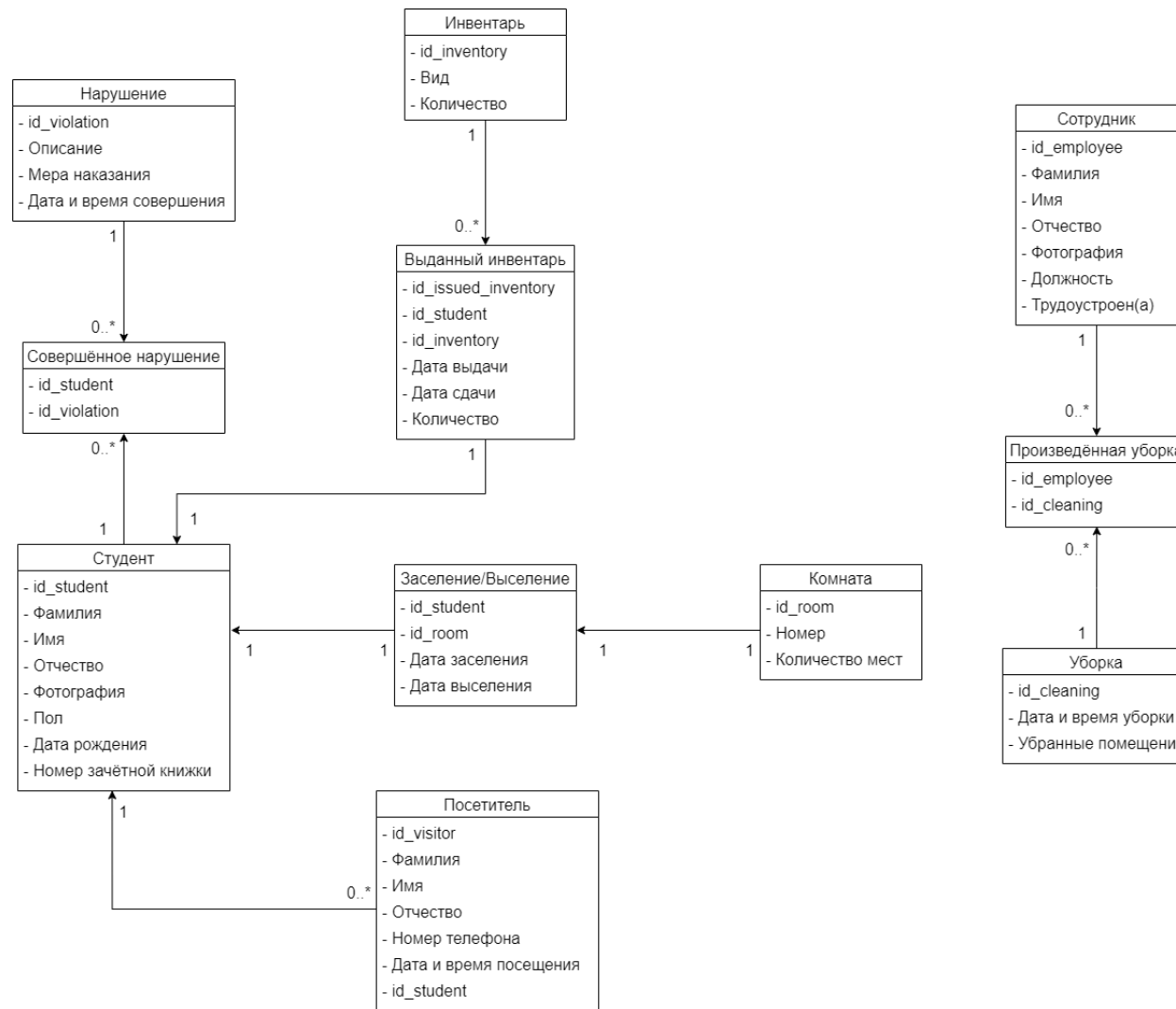


Рис. 2. Логическая модель

1.3 Физическая модель

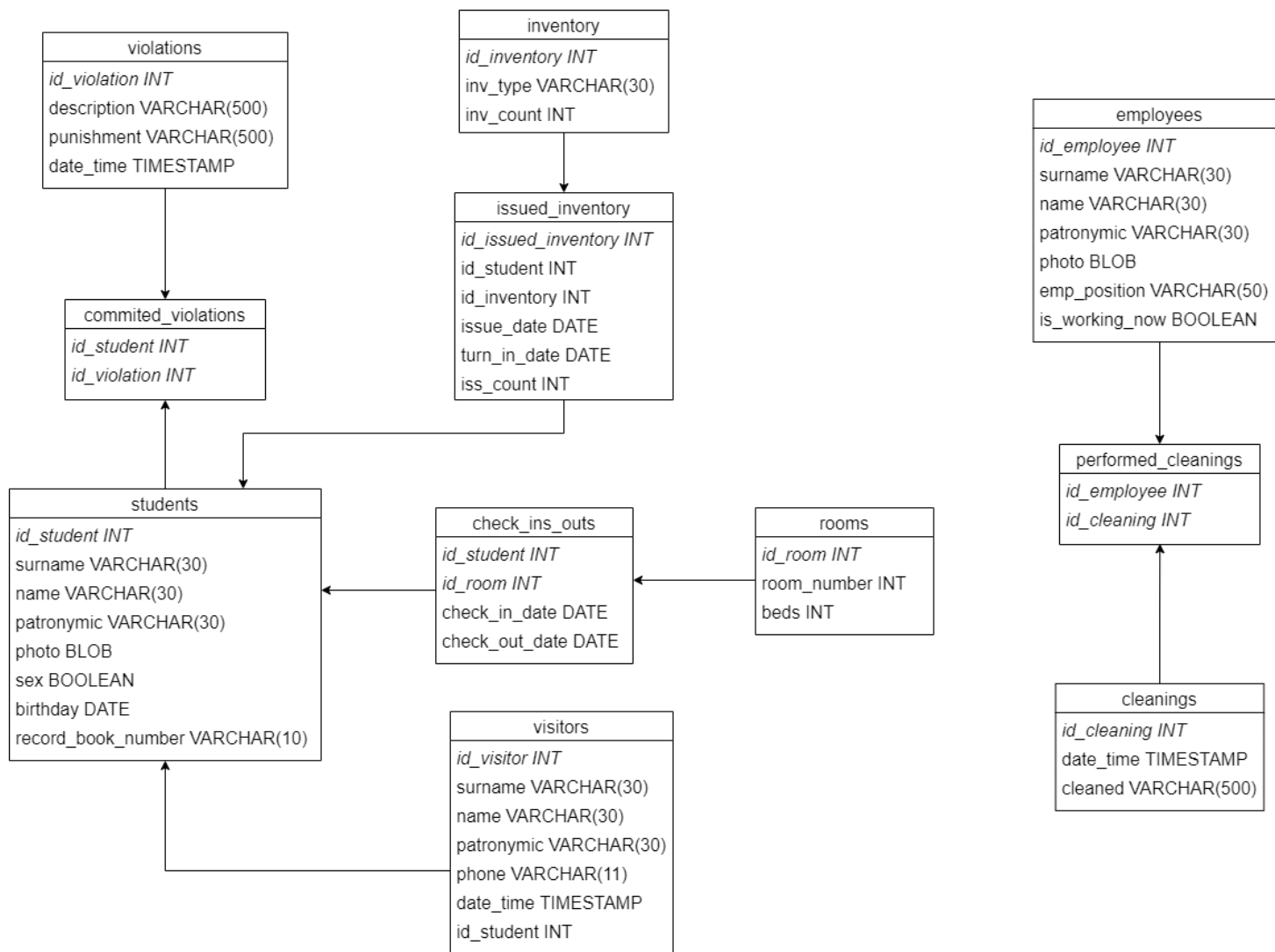


Рис. 3. Физическая модель

Приложение 2. Текст программы

Текст программы доступен по ссылке:

<https://github.com/FireBoT-er/DormitoryAIS>

Приложение 3. Снимки окон программы (скриншоты программы)

Фамилия	Имя	Отчество	Пол	Дата рождения	Фотография	Номер зачётной книжки	Номер комнаты	Дата заселения	Дата выселения
Авдеев	Артём	Арсеньевич	Мужской	24.04.2000	Открыть	ГЛД-118-03	104	29.08.2018	29.08.2022
Вавилов	Всеволод	Владиславович	Мужской	25.05.2000	Открыть	СМР-118-04	104	29.08.2018	29.08.2022
Докторская	Мария	Евгеньевна	Женский	03.05.2002	Открыть	ЧПМ-120-02	101	09.12.2021	09.12.2025
Исаева	Ксения	Егоровна	Женский	12.08.2001	Открыть	ЧСВ-119-01	101	31.08.2019	31.08.2023
Коновалов	Платон	Данилович	Мужской	21.02.2000	Открыть	ЧУА-118-01	104	29.08.2018	29.08.2022
Котов	Андрей	Русланович	Мужской	22.03.2000	Открыть	ВОА-118-02	104	29.08.2018	29.08.2022
Крючкова	Ева	Артёмовна	Женский	14.05.2001	Открыть	МРХ-219-14	102	03.09.2019	03.09.2023
Никулина	Карина	Максимовна	Женский	07.12.2002	Открыть	АМФ-120-08	102	31.08.2020	31.08.2024
Попова	Анастасия	Дмитриевна	Женский	09.03.2001	Открыть	ОГО-119-04	102	02.09.2019	02.09.2023

Рис. 1. Главное окно программы после запуска (отображение информации о заселённых студентах)

Фамилия: Исаева 6/30 ⓘ

Имя: Ксения 6/30 ⓘ

Отчество: Егоровна 8/30 ⓘ

Пол: ☐ Мужской ☒ Женский

Дата рождения: 12.08.2001

Фотография: Открыть image.jpg

Номер зачётной книжки: ЧСВ-119-01 ⓘ

Номер комнаты: Выбрать ✓

Дата заселения: 31.08.2019

Предпросмотр фотографии:

OK Отмена

Рис. 2. Заполненная форма заселения нового студента

Инвентарь, выданный студенту Исаева Ксения Егоровна			
Вид	Количество	Дата выдачи	Дата сдачи ▲
Матрас	1	31.08.2019	31.08.2023
Наволочка	1	31.08.2019	31.08.2023
Одеяло	1	31.08.2019	31.08.2023
Пододеяльник	1	31.08.2019	31.08.2023
Подушка	1	31.08.2019	31.08.2023
Простыня	1	31.08.2019	31.08.2023

Рис. 5. Просмотр списка выданного студенту инвентаря

Добавить новое нарушение

Описание:

0/500

Мера наказания:

0/500

Дата совершения:

Время совершения:

11 : 42

OK
Отмена

Рис. 6. Добавление нарушения

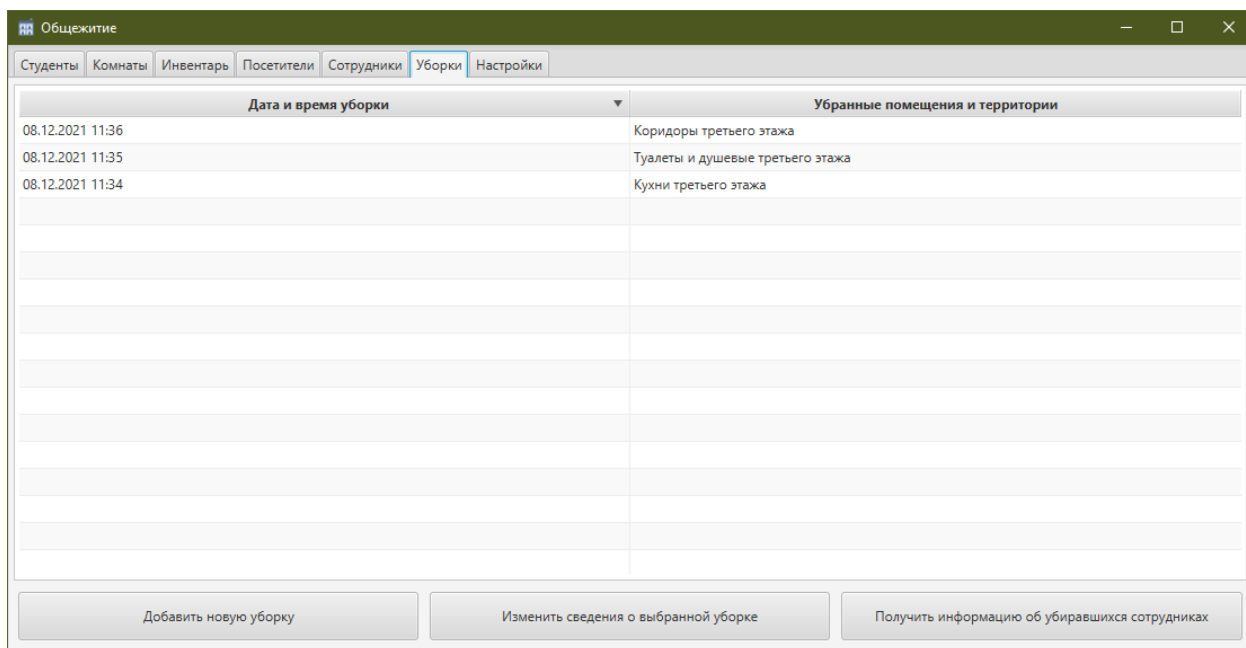


Рис. 9. Отображение информации обо всех совершённых уборках

Убранные помещения и территории:

0/500

Дата уборки:
02.06.2022

Время уборки:
11 : 43

Исполнители

Калинина Мия Александровна

Добавить Убрать

OK Отмена

Рис. 10. Частично заполненное окно добавления информации о совершённой уборке

Сотрудники, убиравшиеся 08.12.2021 в 11:36

Фамилия ▲	Имя	Отчество	Фотография	Должность	Трудоустроен(а)
Громова	Анна	Данииловна	Открыть	Уборщица	Да
Калинина	Мия	Александровна	Открыть	Уборщица	Да

Рис. 11. Просмотр списка убравшихся сотрудников

Общекитие

Студенты Комнаты **Инвентарь** Посетители Сотрудники Уборки Настройки

Вид ▲	Всего	На складе	Выдано
Лопата	10	10	0
Матрас	1000	998	2
Наволочка	3000	2999	1
Одеяло	1000	998	2
Пододеяльник	3000	2999	1
Подушка	1000	998	2
Простыня	3000	2999	1

Добавить новый вид инвентаря Изменить сведения о выбранном инвентаре Получить информацию о текущих обладателях инвентаря

Рис. 12. Отображение информации обо всех единицах инвентаря

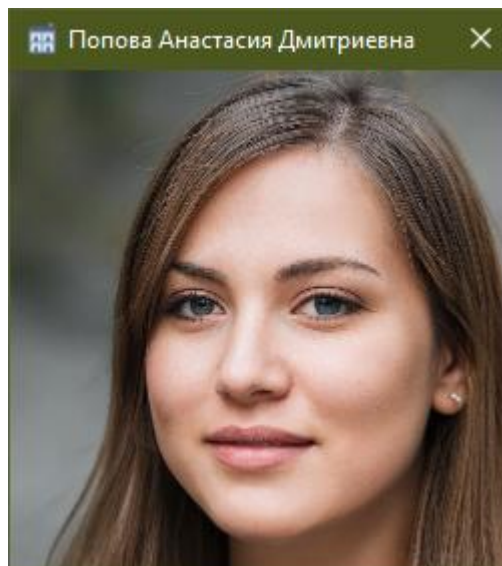


Рис. 13. Окно просмотра фотографии

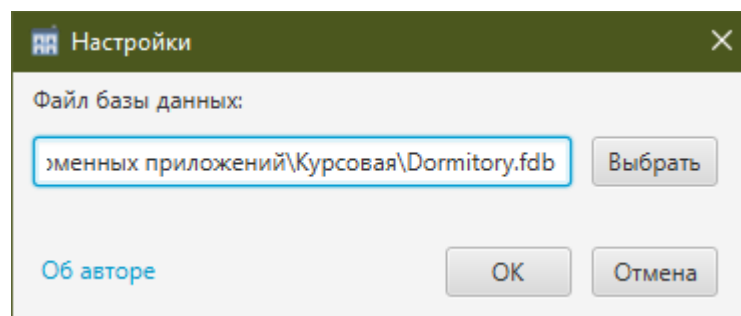


Рис. 14. Окно настроек

Приложение 4. Диаграмма состояний процесса заселения студента

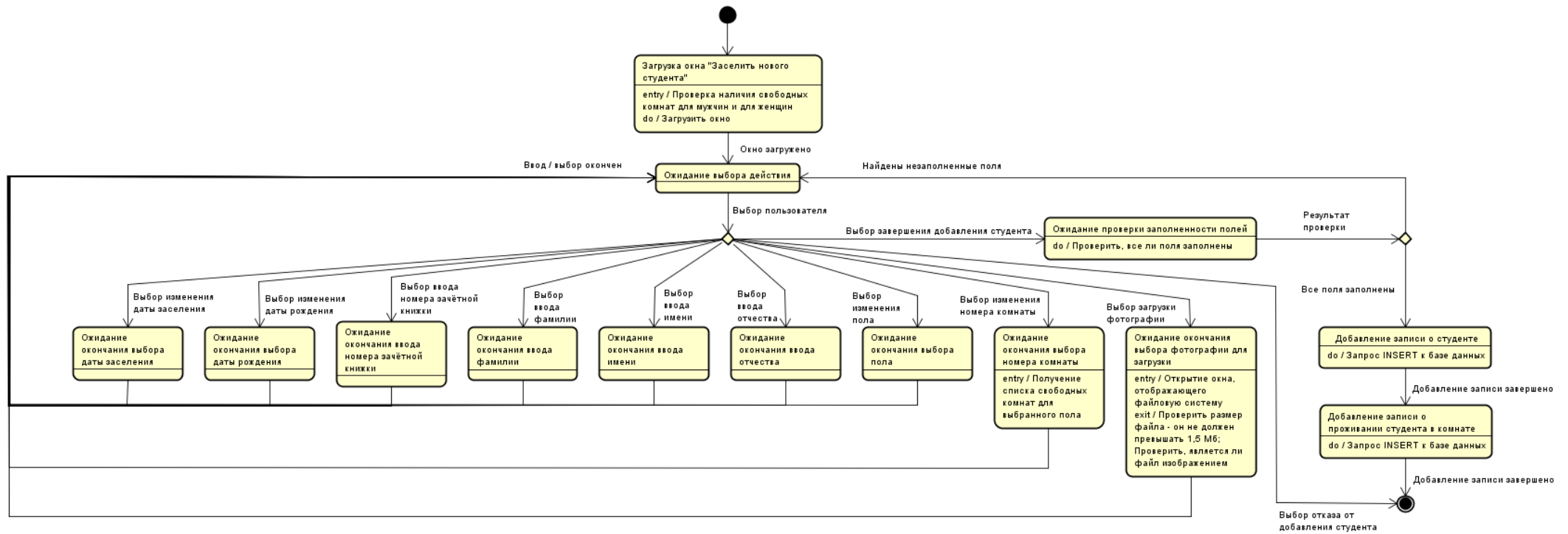


Рис. 1. Диаграмма состояний процесса заселения студента