

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 корпоративных приложений
(наименование дисциплины)

Тема Распределённая ИС Общежития

Руководитель

Кульков Я. Ю.

(фамилия, инициалы)

(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 распределённая информационная система для общежития. В ходе выполнения курсовой работы произведен анализ и сбор требований к разрабатываемой программе. При разработке применены языки Java, JavaScript, HTML и CSS, фреймворк Spring вместе с фреймворком Spring Security, шаблонизатор Thymeleaf, библиотека Hibernate и система управления базами данных MySQL 8.0. Приложение протестировано и полностью работоспособно. Полученный программный продукт можно применять по его прямому назначению.

In this course work, the distributed 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, JavaScript, HTML and CSS languages, the Spring framework along with the Spring Security framework, the Thymeleaf template engine, the Hibernate library and MySQL 8.0 database management 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 моделей данных	9
2.1 Концептуальная модель.....	9
2.2 Логическая модель	10
2.3 Физическая модель	10
3 Проектирование работы системы.....	11
3.1 Базовые понятия.....	11
3.2 Комплексные операции	11
4 Разработка и реализация системы	13
4.1 Создание моделей, контроллеров и представлений	13
4.2 Руководство администратора.....	15
5 Тестирование системы.....	16
Заключение	18
Список используемой литературы	19
Приложение 1. Модели данных	20
Приложение 2. Текст программы	23
Приложение 3. Снимки окон программы (скриншоты программы)	24

					МИВУ.09.03.04-02.000 ПЗ							
Изм.	Лист	№ докум.	Подп.	Дата	Распределённая ИС Общежития				Лит.	Лист	Листов	
Разраб.		Емельянов В. А.									5	26
Пров.		Кульков Я. Ю.										
Н. контр.												
Утв.					МИВУ ПИ _Н -119							

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. Зачастую с таким ПО необходимо работать сразу нескольким сотрудникам – для этого были созданы распределённые информационные системы.

Темой курсовой работы является разработка распределённой 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 распределённой 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 уборки помещений.

Помимо этого, также необходимо реализовать:

- авторизацию пользователей;
- разграничение функционала в зависимости от роли пользователя;
- адаптивный интерфейс;
- валидацию данных на стороне клиента (на предмет корректности формата данных) и сервера (на предмет соответствия данных логике работы).

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, система должна поддерживать несколько ролей пользователей. Это будет влиять на доступные возможности. Так, администратор будет иметь полный доступ к функционалу, тогда как обычный пользователь сможет только просматривать уже существующие данные.

В-третьих, программа будет размещена на сервере (локальном или удалённом), а пользователь будет получать доступ посредством веб-браузера, как к

					МИВУ.09.03.04-02.000 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подп.	Дата		

обычному сайту. В случае с использованием удалённого сервера будет необходимо постоянное подключение к интернету.

В результате изучения доступных технологий был выбран фреймворк Spring. Он позволяет использовать для разработки языки Java, JavaScript, HTML и CSS, а также реализует шаблон модель-представление-контроллер, что в совокупности поможет добиться всех поставленных задач наиболее эффективным способом, не теряя и в пользовательской функциональности. Для упрощения задачи генерации веб-страниц используется шаблонизатор Thymeleaf. В качестве СУБД используется MySQL (а также соответствующий ей JDBC-драйвер), для работы с ней при помощи технологии ORM – библиотека Hibernate.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подп.	Дата		

2 Разработка моделей данных

2.1 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, Имя файла с фотографией, 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 Меру 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, Имя файла с фотографией, 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 и территории. Так как уборщики являются Сотрудниками, имеет смысл создать связь между этой сущностью и Уборкой.

Стоит заметить, что фотографии не хранятся в базе данных, а хранятся в виде файлов на сервере. Именно этим обусловлено хранение только лишь имени файла фотографии.

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

2.2 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.

2.3 Физическая модель

Из логической модели данных напрямую произрастает физическая. Она должна увязать созданную логику и возможности выбранной СУБД. 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.

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

3 Проектирование работы системы

3.1 Базовые понятия

Система построена с использованием архитектурного паттерна MVC – Model-View-Controller или Модель-Представление-Контроллер. Он применяется для удобного разделения логики различных элементов проекта. В качестве СУБД используется MySQL 8.0, доступ к ней реализован посредством репозитория и сервисов. Подробнее о реализации вышеперечисленного можно прочитать в пункте 4.1.

Для авторизации используется часть фреймворка Spring – Spring Security. С его помощью будет создано две базовых роли – Пользователь и Администратор. Администратор будет иметь доступ ко всем возможностям системы, тогда как Пользователь сможет лишь просматривать имеющиеся в ней данные, без возможности добавлять, изменять или удалять их. Так как программа предназначена для внутреннего пользования, то вход в обе учётные записи будет защищён паролем для предотвращения утечки данных.

3.2 Комплексные операции

Для занесения в базу некоторых данных требуется использование нескольких таблиц, как правило, двух или трёх. В таких случаях, чтобы не загромождать страницу добавления информацией, целесообразно разделить этот процесс на несколько этапов. На рисунке 1 в качестве примера показана диаграмма последовательностей процесса заселения студента.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подп.	Дата		

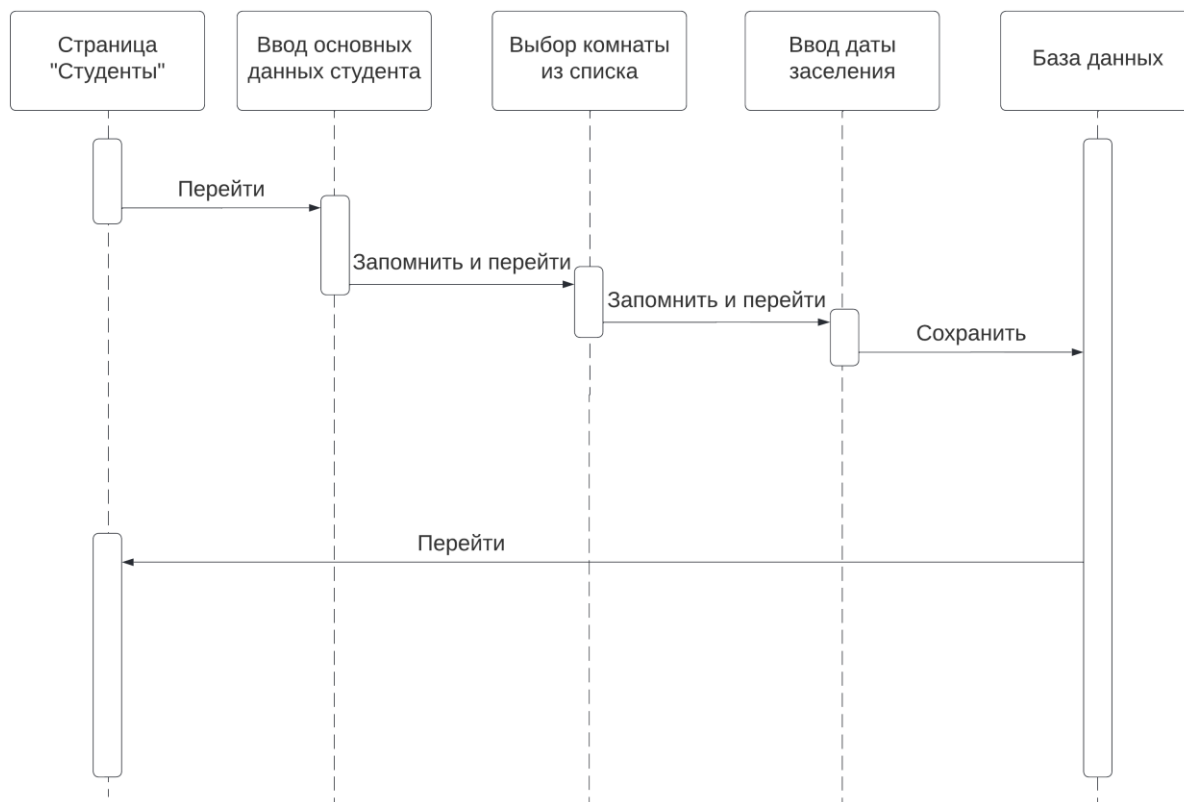


Рис. 1. Заселение студента

4 Разработка и реализация системы

4.1 Создание моделей, контроллеров и представлений

Так как разрабатываемое приложение должно обеспечивать работу с базой данных, первым шагом, основываясь на физической модели, была создана её внутренняя структура. Для создания использовался подход ORM. Его суть заключается в том, что для создания базы данных необходимо написать специальные классы – модели – в которых описываются типы полей, хранимые в них данные и различные ограничения. Далее на основании этих классов генерируются таблицы, а также, при необходимости, связывающие таблицы. Перечень моделей приведён в таблице 1.

Таблица 1

Перечень моделей

Имя модели	Краткое описание
CheckInsOuts	Информация о заселении студента в комнату
Cleanings	Информация о проведённых уборках
Employees	Информация о сотрудниках
Inventory	Информация об инвентаре
IssuedInventory	Информация о выдачах инвентаря
Rooms	Информация о комнатах
Students	Информация о студентах
Violations	Информация о совершённых нарушениях
Visitors	Информация о посетителях

Перед созданием пользовательского интерфейса необходимо создать логику его работы. Для этой цели были созданы классы контроллеров. В них реализовано запрашивание информации из базы данных и логика обработки информации, полученной от пользователя. Контроллеры созданы для всех моделей, кроме CheckInsOuts, так как доступ к ней реализован в контроллере Students. Для вывода

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

информации об авторе создан контроллер AboutController.

Для работы с базой данных используются специальные объекты – сервисы и репозитории. Выстраивается следующая иерархия: Контроллеры → Сервисы → Репозитории → База данных. Из этого следует, что в репозиториях реализуются функции непосредственного взаимодействия с базой данных. В текущем проекте для этой цели используется библиотека Hibernate. Благодаря ей становится возможна автоматическая генерация запросов при использовании правильного стиля наименования заголовков методов в репозитории, реализованном как интерфейс. Сервисы же обеспечивают доступ к данным методам, а также могут проводить дополнительную валидацию передаваемых данных или же вовсе передавать некоторые данные самостоятельно. Репозитории и сервисы созданы для всех моделей.

Для отображения информации из контроллеров используются представления. В отличие от предыдущих элементов, они написаны не на языке Java, а на языке HTML. Для передачи данных из контроллеров применяется шаблонизатор Thymeleaf, который принимает переданные из кода переменные (при помощи класса Model) и разбирает их в соответствии с инструкциями, написанными внутри представления, при помощи собственного синтаксиса, подобного тому, что встречается в языках программирования, в результате чего получается обычная веб-страница, с которой могут работать браузеры. Для обеспечения дополнительной функциональности (например, проверок данных) на стороне клиента использовался язык JavaScript, для доработки интерфейса использовался язык CSS, дополненный библиотекой Bootstrap, которая также помогла в создании адаптивного интерфейса.

Физически представления объединены в папки, названные в соответствии с именем использующегося контроллера. Помимо этого, существует папка Shared, в которой содержатся общие файлы представлений. Большая часть контроллеров реализует следующие представления: Create (создание), Delete (удаление), Details (подробности), Edit (изменение), Index (главная страница). Многие также содержат дополнительные файлы, реализующие многоступенчатое создание и изменение.

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

4.2 Руководство администратора

Требования к системе:

1. Поддержка последних версий браузеров Google Chrome, Mozilla Firefox, Microsoft Edge или других, выпущенных не ранее 2020 года.
2. Необходимо наличие монитора с разрешением не менее 1366x768 и устройств ввода (клавиатуры и мыши).

Требования к установленному ПО: веб-браузер, выпущенный не ранее 2020 года.

Доступ к системе осуществляется посредством веб-браузера. Система поддерживает роли Пользователя и Администратора, которые различаются возможностью доступа к методам добавления, изменения и удаления информации. Для получения полного функционала требуется роль Администратора.

Текст программы приведён в Приложении 2.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подп.	Дата		

5 Тестирование системы

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 2.

Для проведения испытаний использовался компьютер с операционной системой Windows 11 и браузером Google Chrome (версия 108).

Таблица 2

Требования для проверки во время тестирования

Действие	Результат
Вход на сайт	Отображается страница авторизации
Авторизация с некорректным паролем	Ошибка «Неверные учетные данные пользователя»
Нажатие на кнопку «Заселить» на странице «Студенты»	Открывается страница «Данные студента»
Заселение студента с незаполненным полем «Имя»	Выводится сообщение об ошибке: «Поле должно быть заполнено»
Загрузка в раздел «Фотография» текстового файла, расширение которого изменено на «.png»	Выводится сообщение об ошибке: «Файл не является изображением»
Открепление всех студентов от нарушения	Выводится сообщение об ошибке: «Необходимо выбрать хотя бы одного студента»

Таблица 2 (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»
Изменение количества мест в полностью заселённой комнате с 3-х до 2-х	Выводится сообщение об ошибке: «Количество мест не может быть меньше количества проживающих студентов»
Нажатие на кнопку в столбце «Фотография»	Открывается новая вкладка с изображением
Выход из заселения студента после второй стадии	Информация не добавляется в базу данных
Попытка входа по адресу, недоступному для данной роли	Открывается страница входа в систему для входа под учётной записью с достаточными правами

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 распределённой 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;
- авторизация пользователей;
- разграничение функционала в зависимости от роли пользователя;
- адаптивный интерфейс;
- валидация данных на стороне клиента (на предмет корректности формата данных) и сервера (на предмет соответствия данных логике работы).

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
						18
Изм.	Лист	№ докум.	Подп.	Дата		

Список используемой литературы

1. Spring Data JPA - Reference Documentation: сайт. – URL: <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/> (дата обращения: 20.10.2022). – Текст: электронный.
2. Руководство по MySQL: сайт. – URL: <https://metanit.com/sql/mysql/> (дата обращения: 22.11.2022). – Текст: электронный.
3. Hibernate Validator 8.0.0.Final - Jakarta Bean Validation Reference Implementation: Reference Guide: сайт. – URL: https://docs.jboss.org/hibernate/stable/validator/reference/en-US/html_single/?v=8.0 (дата обращения: 28.11.2022). – Текст: электронный.
4. Stack Overflow: сайт. – URL: <https://stackoverflow.com/> (дата обращения: 09.12.2022). – Текст: электронный.

					МИВУ.09.03.04-02.000 ПЗ	Лист
						19
Изм.	Лист	№ докум.	Подп.	Дата		

Приложение 1. Модели данных

1.1 Концептуальная модель

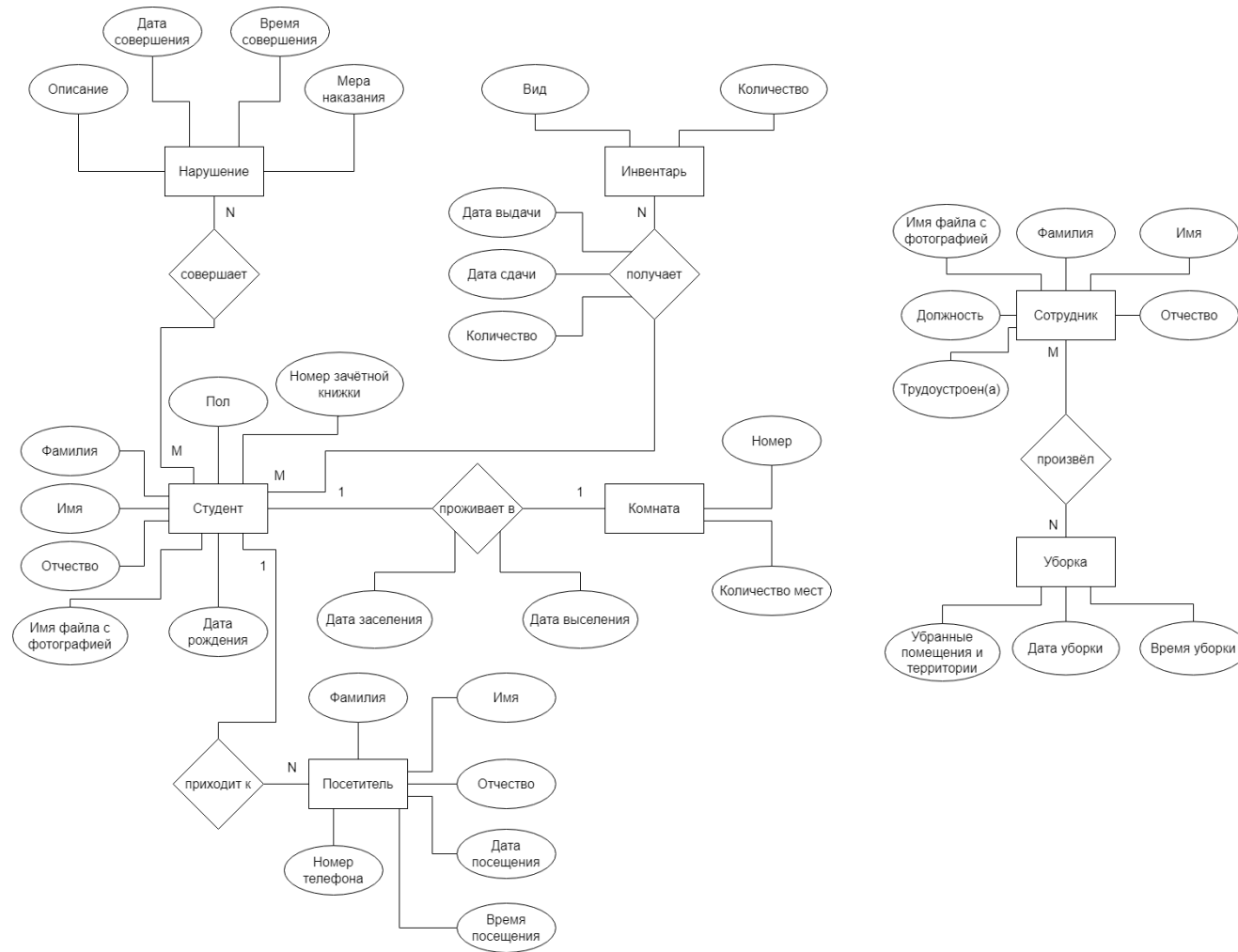


Рис. 1. Концептуальная модель

1.2 Логическая модель

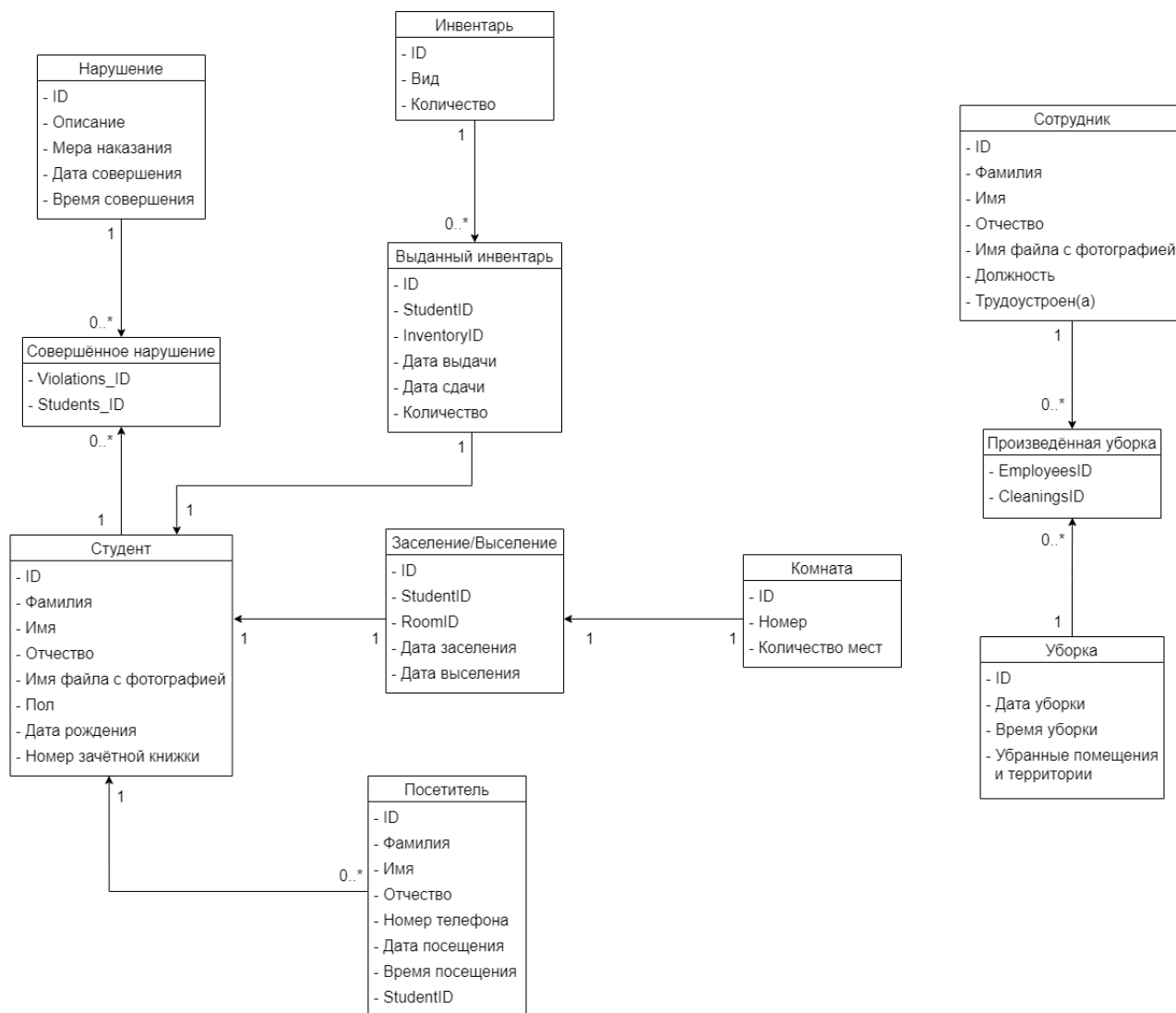


Рис. 2. Логическая модель

1.3 Физическая модель

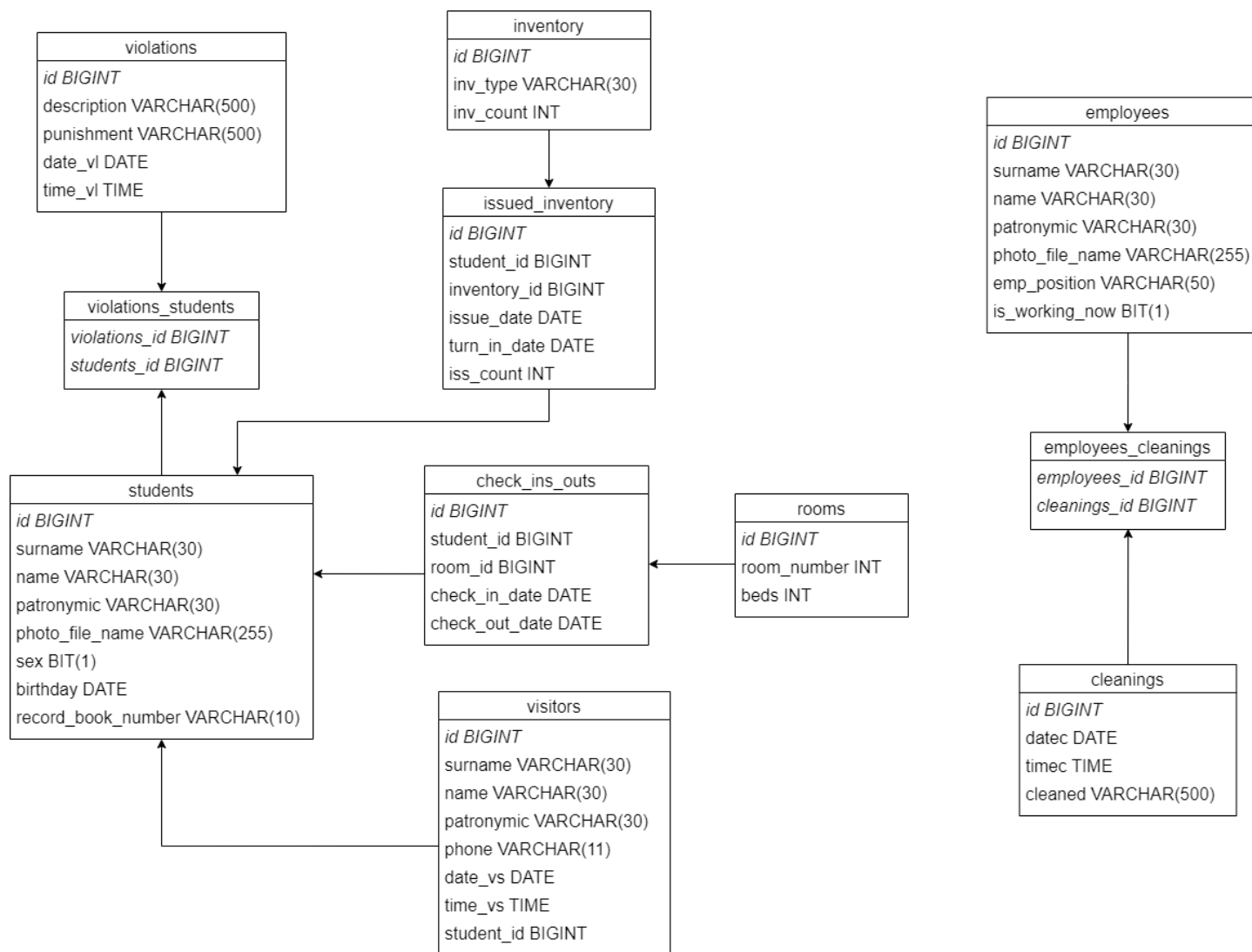


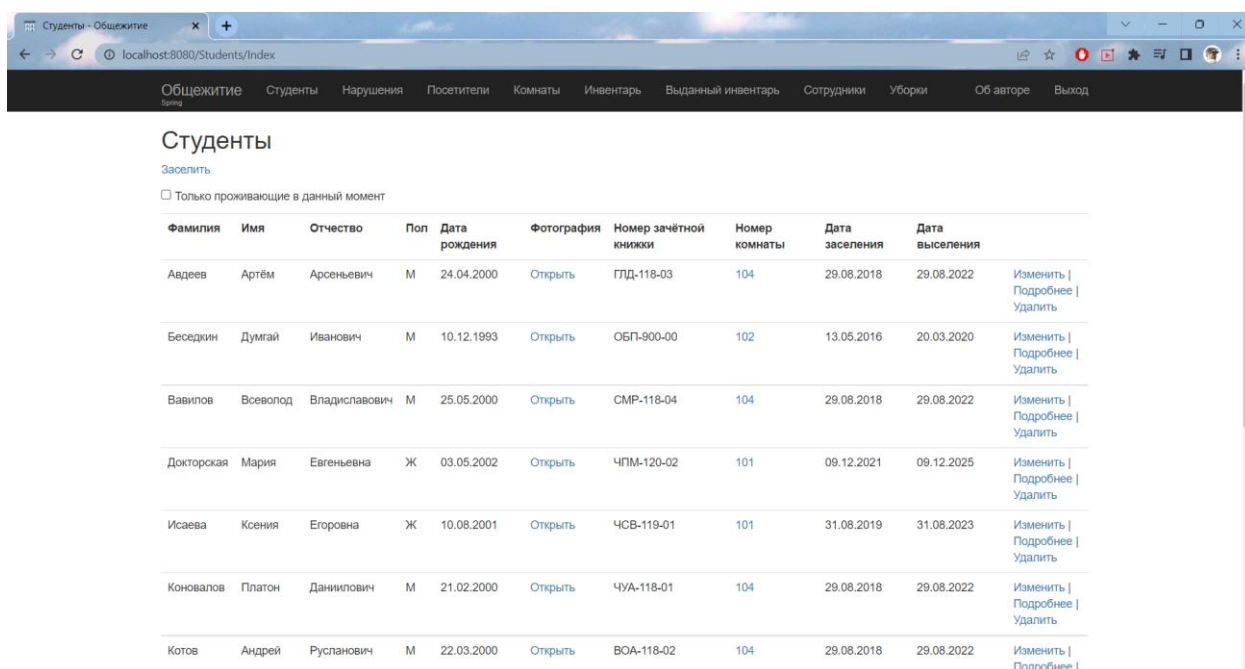
Рис. 3. Физическая модель

Приложение 2. Текст программы

Текст программы доступен по ссылке:

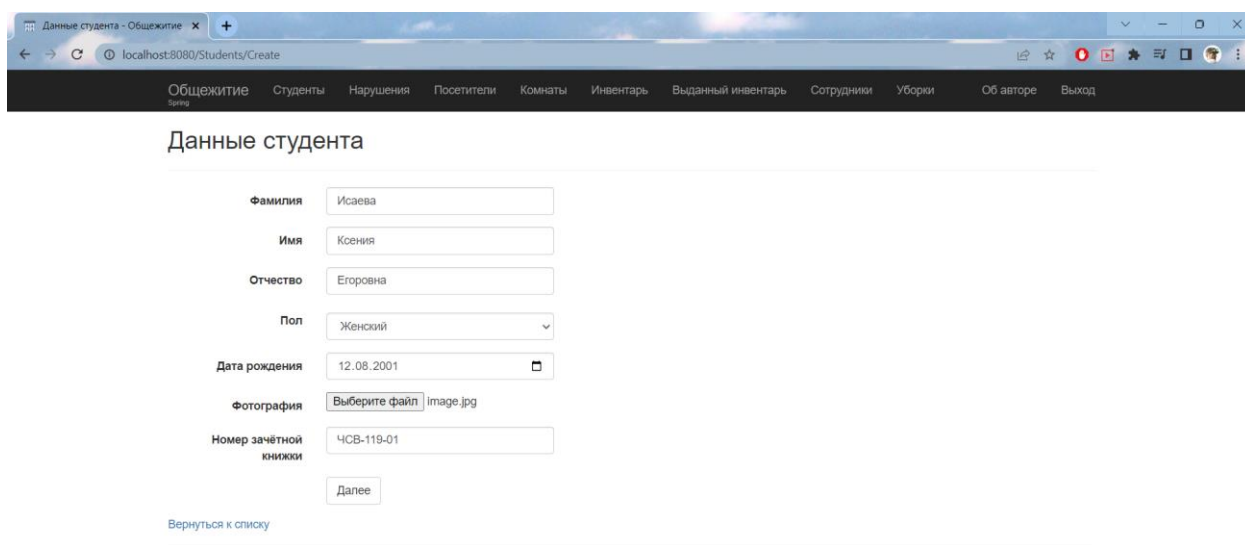
https://github.com/FireBoT-er/DormitoryAIS_Spring

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	М	24.04.2000	Открыть	ГЛД-118-03	104	29.08.2018	29.08.2022	Изменить Подробнее Удалить
Беседкин	Думгай	Иванович	М	10.12.1993	Открыть	ОБП-900-00	102	13.05.2016	20.03.2020	Изменить Подробнее Удалить
Бавилов	Всеволод	Владиславович	М	25.05.2000	Открыть	СМР-118-04	104	29.08.2018	29.08.2022	Изменить Подробнее Удалить
Докторская	Мария	Евгеньевна	Ж	03.05.2002	Открыть	ЧПМ-120-02	101	09.12.2021	09.12.2025	Изменить Подробнее Удалить
Исаева	Ксения	Егоровна	Ж	10.08.2001	Открыть	ЧСВ-119-01	101	31.08.2019	31.08.2023	Изменить Подробнее Удалить
Коновалов	Платон	Данилович	М	21.02.2000	Открыть	ЧУА-118-01	104	29.08.2018	29.08.2022	Изменить Подробнее Удалить
Котов	Андрей	Русланович	М	22.03.2000	Открыть	ВОА-118-02	104	29.08.2018	29.08.2022	Изменить Подробнее Удалить

Рис. 1. Вкладка «Студенты»



Данные студента

Фамилия:

Имя:

Отчество:

Пол:

Дата рождения:

Фотография: image.jpg

Номер зачётной книжки:

[Вернуться к списку](#)

Рис. 2. Заполненная первая часть формы заселения нового студента

Выберите нарушителей

	Фамилия	Имя	Отчество	Пол	Дата рождения	Номер зачётной книжки	Номер комнаты
<input type="checkbox"/>	Докторская	Мария	Евгеньевна	Ж	03.05.2002	ЧПМ-120-02	101
<input checked="" type="checkbox"/>	Исаева	Ксения	Егоровна	Ж	10.08.2001	ЧСВ-119-01	101
<input type="checkbox"/>	Крючкова	Ева	Артёмовна	Ж	14.05.2001	МРХ-219-14	102
<input type="checkbox"/>	Никулина	Карина	Максимовна	Ж	07.12.2002	АМФ-120-08	102
<input checked="" type="checkbox"/>	Попова	Анастасия	Дмитриевна	Ж	09.03.2001	ОГО-119-04	102

Готово

[Назад](#)

Рис. 3. Выбор нарушителей

Выберите посещаемого студента

Фамилия	Имя	Отчество	Пол	Дата рождения	Номер зачётной книжки	Номер комнаты	
Докторская	Мария	Евгеньевна	Ж	03.05.2002	ЧПМ-120-02	101	Выбрать
Исаева	Ксения	Егоровна	Ж	10.08.2001	ЧСВ-119-01	101	Выбрать
Крючкова	Ева	Артёмовна	Ж	14.05.2001	МРХ-219-14	102	Выбрать
Никулина	Карина	Максимовна	Ж	07.12.2002	АМФ-120-08	102	Выбрать
Попова	Анастасия	Дмитриевна	Ж	09.03.2001	ОГО-119-04	102	Выбрать

[Назад](#)

Рис. 4. Выбор посещаемого студента

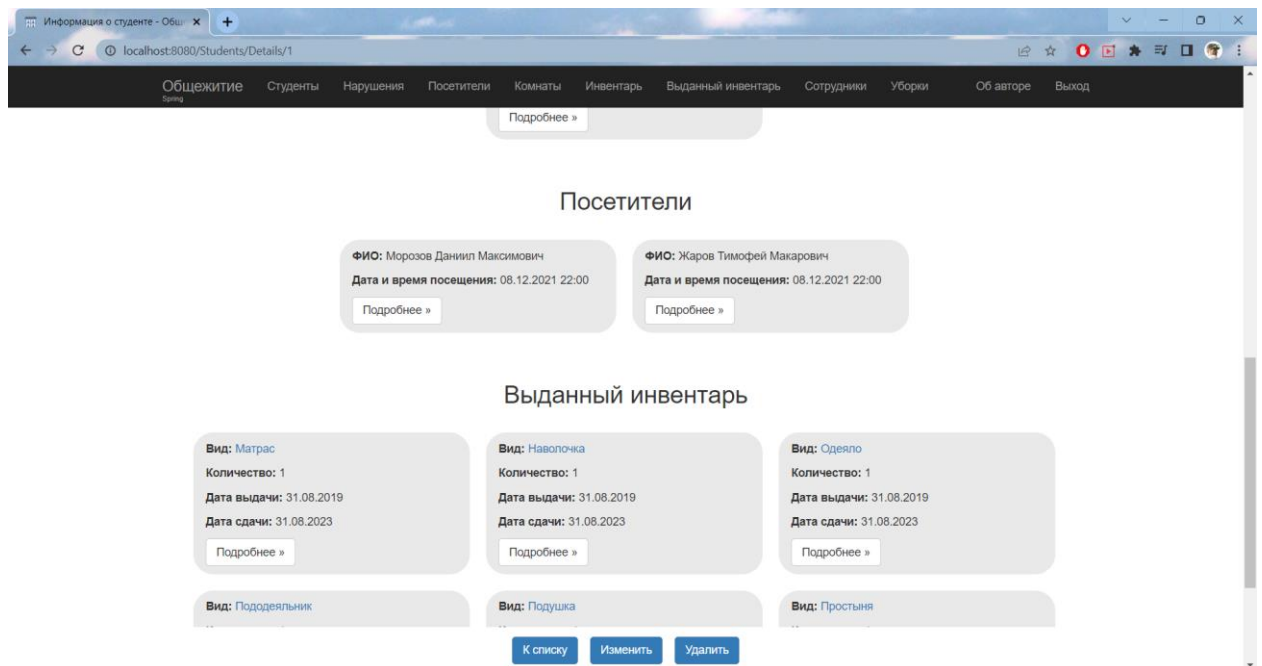


Рис. 5. Часть страницы с подробной информацией о студенте

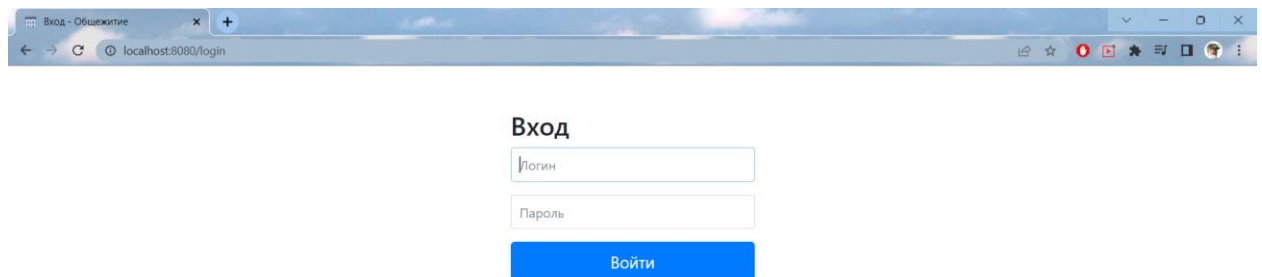


Рис. 6. Страница авторизации