

ТЕХНИЧЕСКИЕ НАУКИ / ENGINEERING

Научная статья

УДК 656.072.1:004.438Java Swing

<https://doi.org/10.35266/1999-7604-2025-3-1>



**Разработка системы бронирования железнодорожных билетов
с использованием Java Swing**

Александр Алексеевич Богнюков[✉], Дмитрий Юрьевич Зорькин,

Лариса Валентиновна Самофалова

Волгоградский государственный технический университет, Волгоград, Россия

Аннотация. В данной статье представлена разработка информационной системы для онлайн-бронирования железнодорожных билетов на языке Java с использованием библиотеки Swing. Цель работы – решение задач цифровизации транспортных услуг, обеспечивающей пользователям удобство, скорость и надежность. В статье сформулированы функциональные требования к системе (регистрация, поиск маршрутов, управление бронированиями) и детально описана архитектура на основе объектно-ориентированного программирования. Проектирование включает интуитивный графический интерфейс с валидацией данных, динамическим обновлением списка мест и модулем просмотра фотоальбома вокзалов. Визуализация структуры системы выполнена с помощью диаграмм использования вариантов. Модульность и кросс-платформенность обеспечивается благодаря компонентам Swing. Данные хранятся в файлах для хранения структурированных табличных данных, что упрощает интеграцию. Тестирование подтвердило корректность работы модулей, включая обработку исключений и синхронизацию операций. Практическая значимость заключается в возможности внедрения системы в реальные сервисы. Перспективы развития: интеграция с интерфейсом прикладного программирования, мобильная версия, рекомендательные алгоритмы и переход на облачные базы данных. Результаты демонстрируют эффективность Java Swing для создания сложных приложений с графическим интерфейсом в транспортной логистике.

Ключевые слова: Java Swing, бронирование билетов, железнодорожный транспорт, GUI, информационные системы, программирование, UML

Для цитирования: Богнюков А. А., Зорькин Д. Ю., Самофалова Л. В. Разработка системы бронирования железнодорожных билетов с использованием Java Swing // Вестник кибернетики. 2025. Т. 24, № 3. С. 6–16. <https://doi.org/10.35266/1999-7604-2025-3-1>.

Original article

Development of railway ticket booking system using Java Swing

Aleksandr A. Bognyukov[✉], Dmitry Yu. Zorkin, Larisa V. Samofalova

Volgograd State Technical University, Volgograd, Russia

Abstract. The paper presents the development of an information system for online railway tickets booking in Java Swing library. The goal is to provide users with convenient, fast and reliable service by solving problems in transport services' digitalization. The authors indicate the functional specifications for the system (registration, route search, booking management) and specify the object-oriented programming architecture. The design includes an intuitive graphical interface with data validation, dynamic updating of the locations' list and a module for viewing the train stations' photo album. Unified modeling language (UML) diagrams create a visualization of the system structure. The modularity and cross-platform compatibility are provided through the use of Swing components. Comma-separated values (CSV) store the data, thus simplifying integration. Testing confirms the correct operation of the modules, including exception handling and operations' synchronization.

The system's integration into real services has practical significance. Prospects involve integrating with Application programming interface (API) payment methods, having a mobile version, employing recommendation algorithms, and switching to cloud databases. The results demonstrate the Java Swing effectiveness for developing graphical user interface (GUI) applications in transport logistics.

Keywords: Java Swing, ticket booking, railway transport, GUI, information systems, programming, UML

For citation: Bognyukov A. A., Zorkin D. Yu., Samofalova L. V. Development of railway ticket booking system using Java Swing. *Proceedings in Cybernetics*. 2025;24(3):6–16. <https://doi.org/10.35266/1999-7604-2025-3-1>.

ВВЕДЕНИЕ

Нынешняя эпоха информационных технологий диктует высокие требования к качеству и удобству пользовательских приложений, особенно в сфере автоматизации повседневных сервисов. Одной из актуальных областей является организация онлайн-бронирования транспортных средств, где удобный и интуитивно понятный интерфейс играет решающую роль в повышении удовлетворенности пользователей и оптимизации рабочих процессов. Разработка графического пользовательского интерфейса для бронирования железнодорожных билетов представляет собой важную задачу, которая требует объединения современных методов проектирования программного обеспечения с практическими потребностями транспортной отрасли [1].

Постановка задачи заключается в разработке приложения для бронирования железнодорожных билетов, обеспечивающего удобство и надежность работы пользователей. Система реализует регистрацию и авторизацию, выбор маршрута, даты поездки, типа вагона и места, а также управление бронированиями.

Для создания графического интерфейса используется Java Swing, обеспечивая гибкость и масштабируемость системы. Особое внимание уделяется навигации между основными разделами: главным меню, формами входа и регистрации, экраном бронирования и личным кабинетом. Также предусмотрены функции просмотра забронированных билетов, их отмены и проверки статуса поездов.

В ходе разработки создаются диаграммы использования вариантов (UML-диаграммы) для визуализации архитектуры системы,

а также анализируются существующие решения в сфере бронирования. Разработка демонстрирует применение принципов объектно-ориентированного программирования (ООП) и интеграцию различных модулей, что позволит адаптировать систему к изменяющимся требованиям рынка.

МАТЕРИАЛЫ И МЕТОДЫ

Проведенный обзор подтвердил эффективность объектно-ориентированного подхода и Java Swing для разработки приложений с графическим интерфейсом (GUI-приложений) в транспортной сфере [2, 3]. Исследования К. Коузена (2018) и И. А. Баркова (2023) демонстрируют, что ООП обеспечивает гибкость и масштабируемость систем бронирования. Данная научная работа [4] выделяет потенциал интеграции с облачными сервисами для обработки больших данных, что актуально для цифровизации транспортных услуг. Указанные методы легли в основу архитектуры системы, сочетающей классические принципы ООП с современными требованиями к модульности.

Разработан интуитивный интерфейс с ключевыми модулями:

1. Главное меню (навигация по разделам).
2. Формы входа/регистрации (с валидацией данных).
3. Экран бронирования (динамический выбор параметров).
4. Личный кабинет (управление заказами).

Использованы стандартные компоненты Swing для обеспечения кросс-платформенности и отзывчивости.

Диаграмма вариантов использования (или use case diagram) является одним из важных инструментов моделирования в методологии

Unified Modeling Language (UML). Она используется для описания функциональности системы с точки зрения ее взаимодействия с внешними сущностями, называемыми акторами. Акторы представляют роли, которые взаимодействуют с системой, и могут быть как людьми, так и внешними системами.

Диаграмма вариантов использования помогает установить, как различные виды пользователей будут взаимодействовать с системой и какие действия они совершат [4]. Это позволяет лучше понять требования к системе, определить основные функциональные возможности и обеспечить их соответствие потребностям пользователей [5] (рис. 1).

Основной поток:

1. Вариант использования начинается, когда Гость заходит в приложение.

2. Если Гость не зарегистрирован, он выполняет регистрацию, вводя e-mail, имя пользователя и пароль. Если же Гость уже имеет аккаунт, этот шаг пропускается, и пользова-

тель переходит к авторизации (см. Альтернативный поток A1).

3. После успешной регистрации или авторизации система предоставляет доступ к основному функционалу [6].

4. Пользователь выбирает одно из доступных действий: просмотр фотоальбома вокзалов или проверку наличия мест в поезде.

5. При выборе просмотра фотоальбома система отображает галерею изображений вокзалов, после чего пользователь возвращается к выбору дальнейших действий (см. Альтернативный поток A4).

6. При выборе проверки наличия мест пользователь переходит к бронированию билета: указывает дату, время и место, после чего система оформляет бронирование.

7. Если в дальнейшем потребуется отменить бронь, пользователь переходит в раздел просмотра забронированных билетов и активирует отмену бронирования (см. Альтернативный поток A3).

8. Завершив работу с системой, пользователь выбирает опцию «Выход из системы»

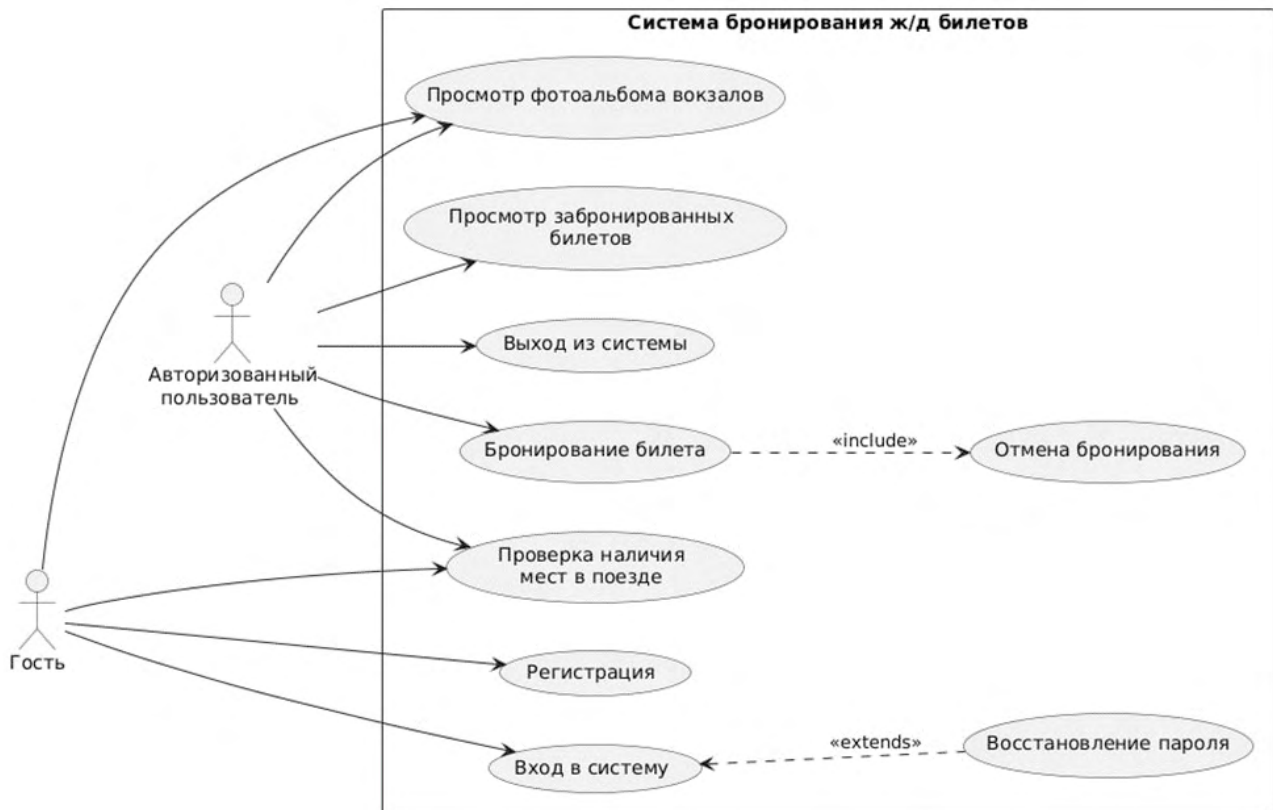


Рис. 1. Диаграмма вариантов использования для информационной системы по бронированию ж/д билетов
Примечание: составлено авторами.

(см. Альтернативный поток А5), после чего сеанс завершается.

Альтернативные сценарии:

1. А1 (Вход): Авторизация → восстановление пароля при ошибке (А2).
2. А2 (Восстановление пароля): Сброс пароля по e-mail/имени → возврат к А1.
3. А3 (Отмена брони): Выбор билета → подтверждение отмены.
4. А4 (Фотоальбом): Просмотр галереи → возврат в меню.
5. А5 (Выход): Завершение сеанса → стартовый экран.

Диаграмма классов (class diagram) – статическая диаграмма, являющаяся основным логическим представлением системы. Диаграмма классов позволяет понять структуру системы, определить классы объектов, их атрибуты и методы, а также взаимосвязи между классами (рис. 2). Это полезный инструмент для анализа, проектирования и документирования программных систем [7].

Сервис Аутентификации – отвечает за вход, регистрацию и сброс пароля. Методы возвращают или проверяют объекты типа «Пользователь».

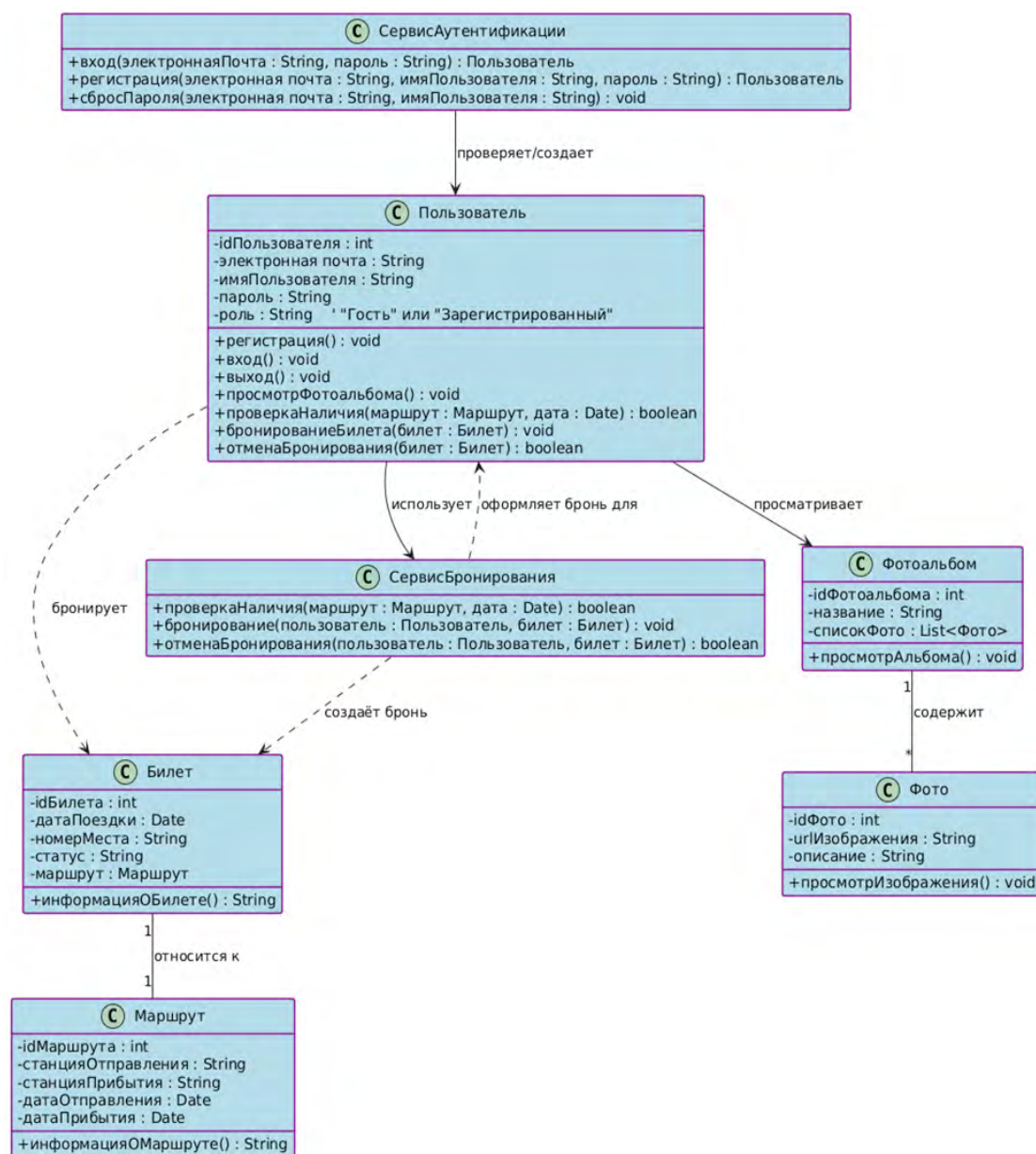


Рис. 2. Диаграмма классов для информационной системы по бронированию ж/д билетов
Примечание: составлено авторами.

Пользователь – представляет как Гостя, так и зарегистрированного пользователя. Помимо стандартных методов регистрации, входа и выхода, содержит методы для просмотра фотоальбома, проверки наличия мест, бронирования билета и отмены брони [8].

Билет – содержит информацию о поездке (дату, время, номер места, статус) и связан с конкретным маршрутом. Билет используется как бронь.

Маршрут – содержит сведения о маршруте (станции, время отправления/прибытия).

Сервис Бронирования – реализует логику проверки наличия мест, оформления брони и ее отмены. Методы работают с объектами «Пользователь» и «Билет».

Фотоальбом и Фото – отвечают за отображение галереи вокзалов.

Ассоциации отражают направленность связей:

1. Сервис аутентификации создает и проверяет Пользователя.

2. Пользователь использует Сервис бронирования для оформления брони, а также просматривает Фотоальбом.

3. Билет связан с Маршрутом, а Пользователь напрямую бронирует билет.

4. Сервис бронирования оформляет бронь для пользователя, создавая связь между Пользователем и Билетом.

5. Фотоальбом агрегирует объекты Фото.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Приложение разработано на Java с использованием Swing. Архитектура основана на принципе разделения ответственности, что обеспечило модульность и упростило поддержку системы. Ключевые компоненты (рис. 3):

1. Навигация и состояние:

– MainFrame: Главный контейнер (управление панелями через CardLayout, хранение глобальных данных).

– TopPanel: Отображение информации о пользователе и кнопки выхода.

2. Основные функциональные модули:

– MainPanel: Стартовое меню (вход, регистрация, проверка мест, фотоальбом).

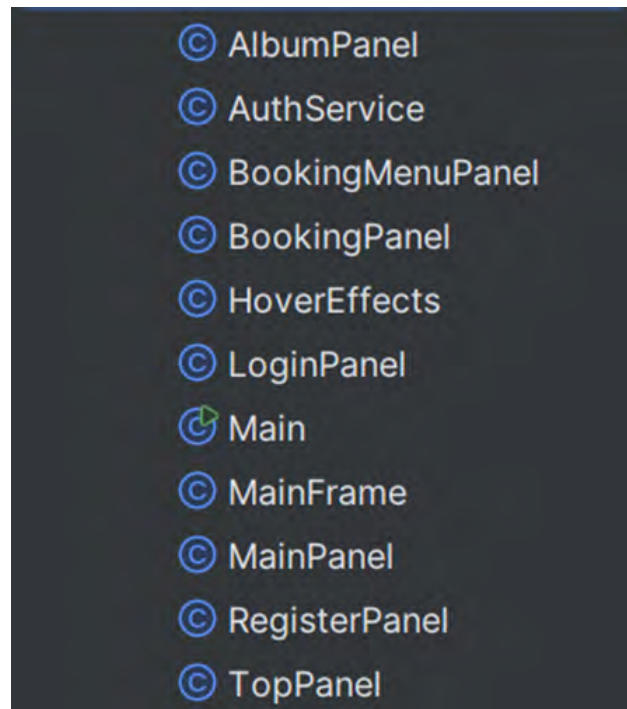


Рис. 3. Полученная структура классов графического пользовательского интерфейса

Примечание: составлено авторами.

– LoginPanel/RegisterPanel: Авторизация и регистрация с валидацией данных.

– BookingPanel: Бронирование билетов (выбор параметров, проверка мест, запись в файл).

– BookingMenuPanel: Управление бронированиями для авторизованных пользователей.

3. Вспомогательные сервисы:

– AlbumPanel: Просмотр фотоальбома вокзалов.

– AuthService: Логика работы с учетными данными (users.csv).

– HoverEffects: Стилизация интерфейса (эффекты наведения, скругленные контейнеры).

Главный класс MainFrame служит контейнером для всех панелей пользовательского интерфейса и реализует механизм навигации между ними посредством CardLayout. Он также хранит глобальное состояние (например, имя текущего пользователя) и обеспечивает корректное обновление верхней панели через вызов метода topPanel.updateProfile (). Особое внимание уделено обработке события нажатия клавиши ESC, чтобы в зависимости от состояния авторизации пользователь не мог вернуться к начальному экрану, если уже авторизован.

Модуль TopPanel отвечает за отображение верхней части интерфейса, включающей кнопку с иконкой аккаунта и приветственное сообщение. При обновлении профиля через метод updateProfile (String username) панель скрывает или отображает данные компоненты в зависимости от наличия имени пользователя. При выходе из аккаунта данные сбрасываются, что приводит к скрытию иконки и надписи (рис. 4).

MainPanel реализует стартовый экран приложения. Здесь расположены кнопки для перехода к авторизации, регистрации, проверки статуса поезда и просмотра фотоальбома.

Данный модуль предоставляет интуитивно понятный интерфейс для первичного взаимодействия пользователя с системой (рис. 5).

Панели LoginPanel и RegisterPanel обеспечивают сбор учетных данных пользователя для входа и регистрации соответственно. В LoginPanel реализована логика обращения к AuthService для проверки корректности введенных данных, а в RegisterPanel дополнительно производится валидация e-mail с использованием регулярного выражения. Оба модуля оповещают пользователя об успешном выполнении или ошибке посредством диалоговых окон (рис. 6, 7).

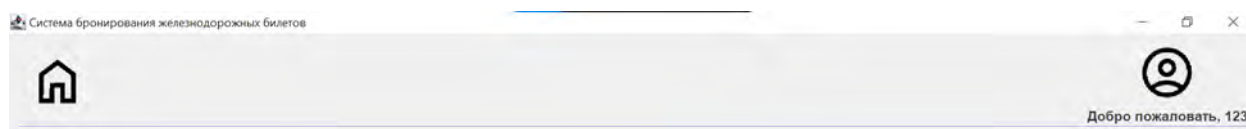


Рис. 4. Пример отображения верхней панели с элементами профиля
Примечание: составлено авторами.

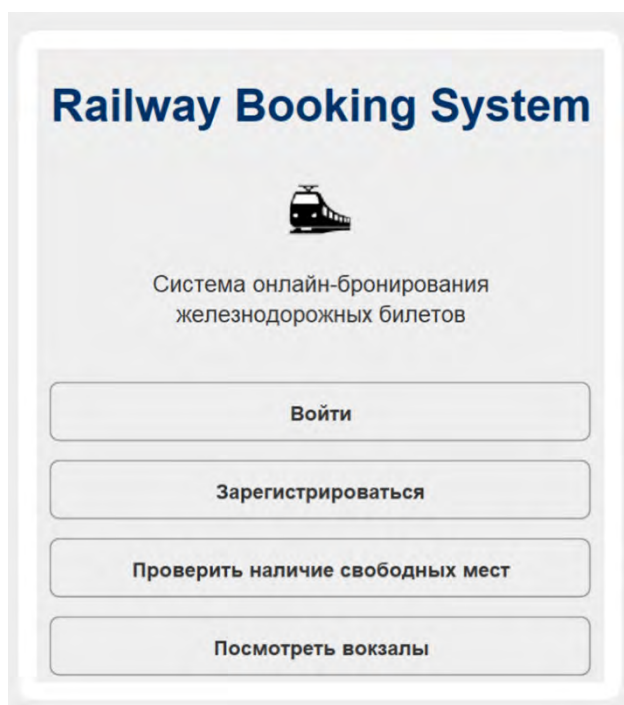


Рис. 5. Внешний вид главной панели приложения
Примечание: составлено авторами.

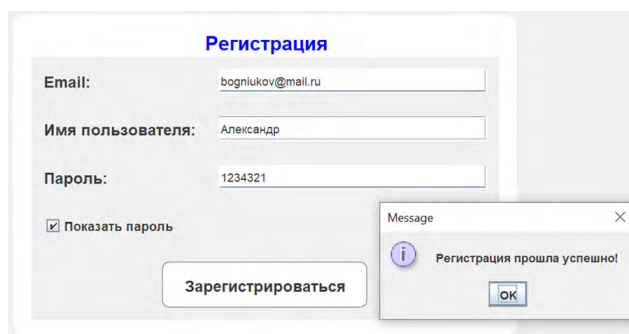


Рис. 6. Пример формы регистрации пользователя (RegisterPanel)
Примечание: составлено авторами.

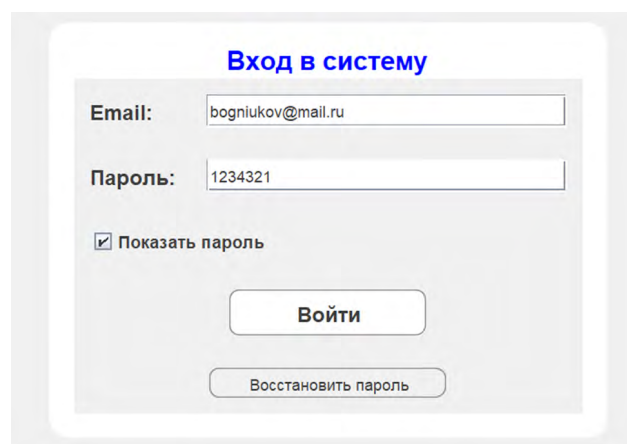


Рис. 7. Пример формы входа пользователя (LoginPanel)
Примечание: составлено авторами.

BookingPanel предназначена для оформления брони. Здесь реализовано:

- выбор станций отправления и прибытия;
- выбор даты отправления и, при необходимости, даты возврата;
- выбор типа вагона и номера вагона;
- выбор места с динамическим обновлением доступных мест (на основе данных, считанных из файла bookings.txt);
- логика сохранения информации о брони в файл и оповещение пользователя о результате операции.

Кроме того, в BookingPanel реализована функция проверки статуса поезда, которая отображает количество забронированных билетов и оставшиеся свободные места (рис. 8, 9).

Рис. 8. Пример формы меню бронирования билетов

Примечание: составлено авторами.

bookings.txt – блокнот
Файл Правка Формат Вид Справка
Пользователь: Александр, Откуда: Волгоград (Волгоград-1), Куда: Самара, Дата отправления: 03-02-2025,
Дата возврата: 05-02-2025, Тип вагона: Плацкарт, Номер вагона: 6, Место: 2

Рис. 9. Забронированный билет в файле bookings.txt

Примечание: составлено авторами.

Модуль BookingMenuPanel предназначен для предоставления дополнительных возможностей авторизованному пользователю. Здесь реализована функция просмотра забронированных билетов, которая считывает данные из файла и отображает их в виде таблицы с возможностью возврата брони.

Также с этой панели пользователь может перейти к оформлению новой брони, проверить статус поезда или просмотреть фотоальбом (рис. 10).

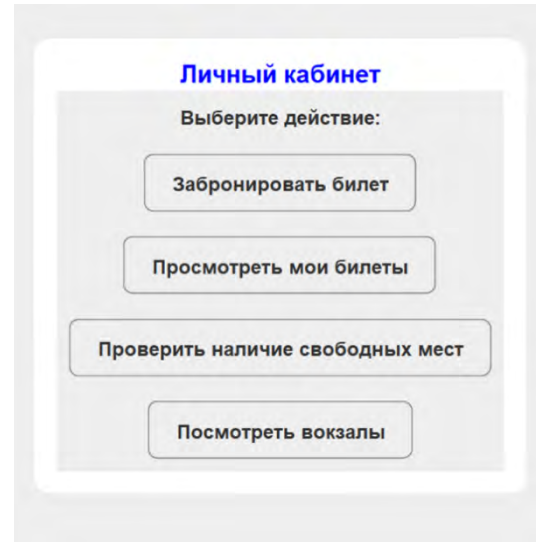


Рис. 10. Пример интерфейса панели меню бронирования

Примечание: составлено авторами.

AlbumPanel реализует функционал фотоальбома вокзалов. Модуль позволяет пользователю просматривать изображения вокзалов с поддержкой переключения с помощью кнопок (и клавиатурных стрелок). Для масштабирования изображений используется метод `scaleImageIcon()`, обеспечивающий плавное отображение фото в заданном разрешении (рис. 11).

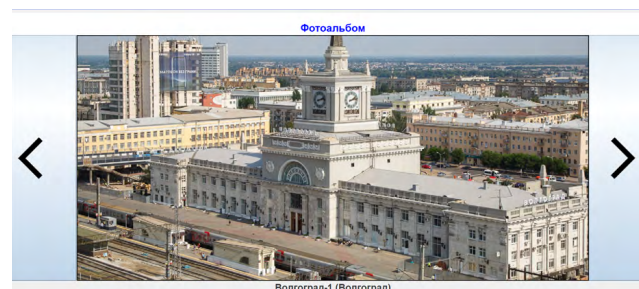


Рис. 11. Пример интерфейса фотоальбома

Примечание: составлено авторами.

AuthService представляет собой сервис для работы с учетными записями пользователей. Реализованные методы включают регистрацию, вход и сброс пароля, при этом

данные пользователей хранятся в файле users.csv. Логика проверки существования пользователя и корректности учетных данных реализована посредством чтения и записи в файл (рис. 12).

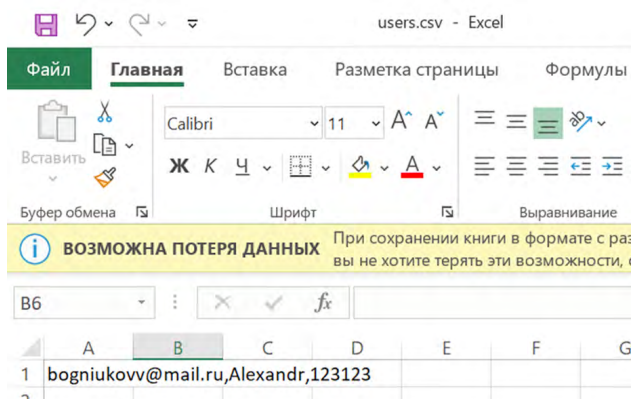


Рис. 12. Запись данных пользователя в файле users.csv

Примечание: составлено авторами.

Класс HoverEffects содержит утилитные методы для стилизации компонентов интерфейса, таких как добавление эффекта подчеркивания при наведении, создание кнопок с иконками, а также обертывание панелей в скругленные контейнеры для улучшения визуального восприятия (рис. 13).



Рис. 13. Пример работы эффекта подчеркивания при наведении на иконку «Домой»

Примечание: составлено авторами.

Интеграция модулей обеспечивается через класс MainFrame, который связывает все панели с помощью CardLayout. Каждый модуль взаимодействует с MainFrame для получения общего состояния (например, имени пользователя) и перехода между экранами. Сервисы AuthService и BookingService вызываются из соответствующих панелей, что позволяет

реализовать чистую архитектуру и модульное тестирование.

Для тестирования были разработаны следующие сценарии:

1. Авторизация и регистрация: проверка ввода корректных и некорректных данных, тестирование восстановления пароля.

Проверим все возможности регистрации и авторизации. Если пользователь захочет зарегистрировать новый аккаунт, но этот адрес электронной почты уже есть в «базе данных», программа выдаст ошибку регистрации «Пользователь с таким адресом уже зарегистрирован» (рис. 14).

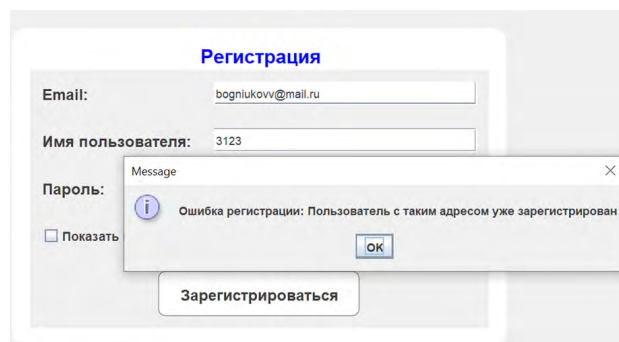


Рис. 14. Попытка зарегистрировать второй аккаунт с одним адресом электронной почты

Примечание: составлено авторами.

Если пользователь забыл пароль – ему покажется диалоговое окно «Неверный адрес электронной почты или пароль» (рис. 15).

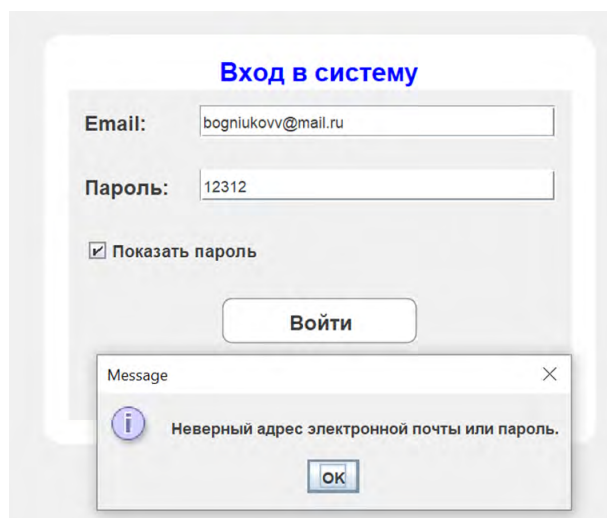


Рис. 15. Попытка авторизоваться с неверным паролем

Примечание: составлено авторами.

Для восстановления пароля необходимо ввести e-mail и имя пользователя, указанные при регистрации (рис. 16).

Рис. 16. Меню ввода данных для восстановления пароля

Примечание: составлено авторами.

Если данные корректны, система предоставляет возможность сменить пароль (рис. 17).

Рис. 17. Меню сброса пароля

Примечание: составлено авторами.

2. Оформление брони: выбор станций, дат, типа вагона, номера вагона и места, проверка корректности обновления списка свободных мест.

Если пользователь бронирует билет, то в следующий раз это забронированное место пропадет из списка (рис. 18).

Рис. 18. Список доступных мест после бронирования

Примечание: составлено авторами.

Если пользователи бронируют все билеты на конкретный поезд, то появится надпись «Все билеты забронированы» (рис. 19).

Рис. 19. Сообщение «Все билеты забронированы»

Примечание: составлено авторами.

3. Просмотр бронирований: считывание данных брони из файла, корректное отображение в виде таблицы, а также проверка механизма возврата брони (рис. 20).

В результате реализации системы был достигнут высокоуровневый уровень модульности, что позволило разделить функциональность на логически независимые блоки. Каждый модуль отвечает за свою область ответственности: от управления навигацией до обработки учетных данных и оформления брони. Проведенное функциональное тестирование подтвердило корректность работы системы, стабильность ее функционирования и удобство использования конечным пользователем.

ЗАКЛЮЧЕНИЕ

Результаты тестирования подтвердили корректность работы всех функциональных модулей. Были проверены сценарии регистрации, авторизации, оформления и отмены бронирования, а также взаимодействие с файлами данных. Приложение демонстрирует надежность и производительность, а его архитектура соответствует принципам модульности, что облегчает его поддержку и дальнейшее развитие.

Таким образом, поставленные задачи были успешно выполнены, а разработанное программное обеспечение может быть исполь-

Выбрать	Откуда	Куда	Дата отправления	Дата возврата	Тип вагона	Вагон	Место
☒	Волгоград (Волгоград-1)	Самара	04-02-2025	Нет	Плацкарт	1	1
☒	Волгоград (Волгоград-1)	Самара	04-02-2025	Нет	Плацкарт	1	2
☒	Волгоград (Волгоград-1)	Самара	04-02-2025	Нет	Плацкарт	1	3
☒	Волгоград (Волгоград-1)	Самара	04-02-2025	Нет	Плацкарт	1	4
☒	Волгоград (Волгоград-1)	Самара	04-02-2025	Нет	Плацкарт	1	5
☒	Волгоград (Волгоград-1)	Самара	04-02-2025	Нет	Плацкарт	1	6
☒	Волгоград (Волгоград-1)	Самара	04-02-2025	Нет	Плацкарт	1	7
☒	Волгоград (Волгоград-1)	Самара	04-02-2025	Нет	Плацкарт	1	8
☒	Волгоград (Волгоград-1)	Самара	04-02-2025	Нет	Плацкарт	1	9
☐	Волгоград (Волгоград-1)	Самара	04-02-2025	Нет	Плацкарт	1	10

Вернуть выбранные билеты

Рис. 20. Проверка механизма отображения и возврата брони

Примечание: составлено авторами.

зовано как основа для дальнейшего расширения системы, в том числе путем интеграции с внешними сервисами и добавления допол-

нительных функций, таких как онлайн-оплата и динамическое обновление данных о расписании поездов.

Список источников

1. Андреева Т. И. Частный железнодорожный транспорт Сибири : моногр. Барнаул : АлтГПУ, 2021. 276 с.
2. Коузен К. Современный Java: рецепты программирования. М. : ДМК Пресс, 2018. 275 с.
3. Москвитин А. А. Данные, информация, знания: методология, теория, технологии : моногр. 2-е изд., стер. СПб. : Лань, 2023. 236 с.
4. Lalović K. G., Bogdanoski M. Z. Java GUI application for comparing the levels of biometric security: Fingerprint vs. Iris // *Vojnotehnički glasnik*. 2021. Vol. 69, no. 3. P. 676–686. <https://doi.org/10.5937/vojtehg69-32007>.
5. Рамальо Л. Python – К вершинам мастерства. Лаконичное и эффективное программирование / пер. с англ. А. А. Слинкина. 2-е изд. М. : ДМК Пресс, 2022. 898 с.
6. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя : руководство / пер. с англ. Н. Мухина. 2-е изд. М. : ДМК Пресс, 2006. 496 с.
7. Харрис Д. М., Харрис С. Л. Цифровая схемотехника и архитектура компьютера / пер. с англ. Imagination Technologies. М. : ДМК Пресс, 2017. 792 с.
8. Бурмистров А. В. Программирования на языке JAVA. Методические указания к лабораторным работам : методические указания. Пенза : ПензГТУ, 2014. 150 с.

References

1. Andreeva T. I. Chastnyy zheleznodorozhnyy transport Sibiri. Monograph. Barnaul: Altai State Pedagogical University; 2021. 276 p. (In Russ.).
2. Kousen K. Modern Java recipes: Simple solutions to difficult problems in Java 8 and 9. Moscow: DMK Press; 2018. 275 p. (In Russ.).
3. Moskvitin A. A. Dannye, informatsiya, znaniya: metodologiya, teoriya, tekhnologii. Monograph. 2nd rev. ed. St. Petersburg: Lan; 2023. 236 p. (In Russ.).
4. Lalović K. G., Bogdanoski M. Z. Java GUI application for comparing the levels of biometric security: Fingerprint vs. Iris. *Vojnotehnički glasnik*. 2021;69(3):676–686. <https://doi.org/10.5937/vojtehg69-32007>.
5. Ramalho L. Fluent Python: Clear, concise and effective programming. Slinkin A. A., trans. 2nd ed. Moscow: DMK Press; 2022. 898 p. (In Russ.).
6. Booch G., Rumbaugh D., Jacobson I. The unified modeling language user guide. Mukhin N., trans. 2nd ed. Moscow: DMK Press; 2006. 496 p. (In Russ.).
7. Harris D. M., Harris S. L. Digital design and computer architecture. Imagination Technologies, trans. Moscow: DMK Press; 2017. 792 p. (In Russ.).
8. Burmistrov A. V. Programmirovaniya na yazyke JAVA. Metodicheskie ukazaniya k laboratornym rabotam. Methodical instructions. Penza: Penza State Technological University; 2014. 150 p. (In Russ.).

Информация об авторах

А. А. Богнюков – студент;
<https://orcid.org/0009-0007-4463-6050>,
bogniukovv@gmail.com✉

Д. Ю. Зорькин – преподаватель;
<https://orcid.org/0009-0002-3875-9285>,
mosh285@gmail.com

Л. В. Самофалова – старший преподаватель;
<https://orcid.org/0000-0003-4694-2645>,
samof13@mail.ru

About the authors

A. A. Bognyukov – Student;
<https://orcid.org/0009-0007-4463-6050>,
bogniukovv@gmail.com✉

D. Yu. Zorkin – Lecturer;
<https://orcid.org/0009-0002-3875-9285>,
mosh285@gmail.com

L. V. Samofalova – Senior Lecturer;
<https://orcid.org/0000-0003-4694-2645>,
samof13@mail.ru