

ИССЛЕДОВАНИЕ МАСШТАБНЫХ СЕРВИСНЫХ СИСТЕМ И ТЕХНОЛОГИЧЕСКОЙ АРХИТЕКТУРЫ В КОМПЛЕКСНОЙ СИСТЕМЕ УПРАВЛЕНИЯ ОТЕЛЕМ

Аннотация. В статье разрабатывается комплексная система управления отелем на основе технологического стека SSM (Spring + Spring MVC + MyBatis) и архитектурного шаблона MVC. Исследуется применение масштабных сервисных систем для повышения операционной эффективности и качества обслуживания в гостиничной индустрии. Система включает модули бронирования, управления номерами, клиентскими данными и статистики, с реализацией ролевого управления доступом (RBAC). Внедрение системы способствует автоматизации процессов и оптимизации ресурсов, а в перспективе возможно расширение за счёт интеграции искусственного интеллекта и блокчейн-технологий.

Ключевые слова: Система управления отелем, масштабные сервисные системы, SSM-фреймворк, архитектура MVC.

В условиях жесткой конкуренции и постоянно растущих ожиданий клиентов современная гостиничная индустрия рассматривает системы массового обслуживания (СМО) как ключевой инструмент для повышения качества обслуживания и операционной эффективности. В данной статье на основе технологической платформы SSM (Spring + Spring MVC + MyBatis) разработана и реализована комплексная система управления отелем, направленная на оптимизацию бизнес-процессов гостиничного хозяйства и повышение удовлетворенности клиентов с помощью автоматизации и информационных технологий.

Массовые сервисные системы, являясь краеугольным камнем современного управления отелями, повышают общую эффективность работы гостиницы за счет стандартизированных процессов, автоматизации и оптимизации ресурсов^[1]. Основные принципы включают: стандартизацию процессов - единые стандарты и процедуры обслуживания обеспечивают стабильность и предсказуемость качества услуг. Например, Marriott и Hilton стандартизировали процессы регистрации, уборки номеров и дополнительных услуг, добившись единообразия качества. Автоматизация и применение технологий - использование современных технологий, таких как автоматизированные системы бронирования, системы управления номерами и CRM, значительно повышает скорость и точность обслуживания. Например, электронные ключи и мобильные приложения позволяют гостям самостоятельно бронировать, заезжать и заказывать услуги. Оптимизация распределения ресурсов - динамическое ценообразование, прогнозирование спроса и планирование персонала обеспечивают эффективное использование ресурсов. Система управления доходами (RMS) анализирует рыночный спрос в реальном времени, корректирует цены и максимизирует доход от номеров. Контроль качества и обратная связь: постоянный мониторинг качества обслуживания и сбор отзывов клиентов помогают отелям своевременно выявлять и решать проблемы, повышая удовлетворенность клиентов^[2]. Например, онлайн-анкеты и системы репутационного управления (такие как TripAdvisor) позволяют оптимизировать процессы обслуживания. Конкретное применение СМО в отелях охватывает многочисленные аспекты, включая ресепшн, ресторанное обслуживание, управление номерами и управление взаимоотношениями с клиентами (CRM). Например, ресепшн использует систему управления имуществом (PMS) для быстрого заезда и выезда; рестораны оптимизируют процесс заказов и расчетов с помощью POS-систем; управление номерами



осуществляется с помощью системы управления уборкой (HMS), которая отслеживает состояние номеров в реальном времени, повышая эффективность уборки.

Данная система использует зрелую структуру SSM в качестве базовой архитектуры бэкенда, сочетая паттерн MVC (Model-View-Controller), чтобы создать иерархически четкую и слабосвязанную системную архитектуру. Техническая архитектура использует Spring в качестве ядра контейнера, который с помощью внедрения зависимостей (DI) и аспектно-ориентированного программирования (AOP) унифицирует управление жизненным циклом и зависимостями всех Java-компонентов в системе; Spring MVC отвечает за обработку запросов в веб-слое; MyBatis служит фреймворком для уровня постоянства, реализуя отображение объектов-отношений через XML или аннотации, упрощая операции с базой данных. Применение паттерна MVC в системе строго следует этому паттерну, разделяя приложение на модель (бизнес-логика и данные), представление (интерфейс пользователя) и контроллер (координация модели и представления)^[3]. Данная архитектура обеспечивает высокую сплоченность и низкую связанность системы, облегчая последующее обслуживание и техническое обновление. Процесс обработки запросов: пользователь отправляет HTTP-запрос через браузер, DispatcherServlet принимает его и маршрутизирует к соответствующему контроллеру; контроллер вызывает сервисный слой для обработки бизнес-логики, при необходимости взаимодействует с базой данных через MyBatis; результат обработки возвращается в представительский слой, рендерится в HTML-страницу и возвращается пользователю^[4].

Система, основанная на анализе требований, разделена на десять основных функциональных модулей, включая вход и регистрацию пользователей, бронирование и управление заселением, управление этажами и номерами, управление клиентской информацией, операционную статистику и т.д. Каждый модуль обеспечивает безопасные и контролируемые права доступа через единый механизм управления правами (ролевое управление доступом, RBAC).

Проектирование базы данных, как ключевой этап реализации системы, использует комбинированный метод концептуальной модели (диаграмма E-R) и физической структуры таблиц. Основные сущности включают пользователей, номера, заказы, этажи, роли и т.д., а полная сеть данных построена с помощью внешних ключей^[5]. Например, таблица пользователей (User) записывает основную информацию и учетные данные; таблица заказов (BookOrder) хранит информацию о бронировании, включая тип номера, время заезда, статус заказа и т.д.; таблица номеров (Room) фиксирует атрибуты, такие как номер, тип, статус и цена; таблица ролей и меню (Authority) управляет соответствием ролей и прав доступа к меню. В физической структуре таблиц с помощью первичных ключей, ограничений внешних ключей и оптимизации индексов обеспечивается согласованность данных и эффективность запросов. Например, таблица заказов и таблица номеров связаны через ID номера, что обеспечивает синхронизацию статуса заказа и статуса номера в реальном времени^[6].

В данной статье на основе фреймворка SSM и модели MVC разработана и реализована комплексная система управления отелем с полным набором функций и четкой архитектурой. Внедрение концепции крупномасштабных сервисных систем и современных информационных технологий позволило эффективно повысить операционную эффективность и качество обслуживания отеля. В будущем можно интегрировать новые технологии, такие как искусственный интеллект и блокчейн, для усиления интеллектуальности и безопасности системы.

Список литературы:

1. Волкова В.Н., Денисов А.А. Основы теории систем и системного анализа: учебник для вузов / Волкова В.Н., Денисов А.А. – СПб.: Издательство Политехнического университета, 2019. – 767 с.



2.Тимохин В.В., Жарков А.С. Современные информационные системы в гостиничном бизнесе: монография / Тимохин В.В., Жарков А.С. – М.: КУРС, 2021. – 218 с.

3.Шафранская Д.Э. Управление доходами в индустрии гостеприимства: стратегии и автоматизация / Д.Э. Шафранская // Российское предпринимательство. – 2022. – Т. 23. – № 4. – С. 112–128.

4.Гайдаржи Г.Б., Федоров И.В. Веб-технологии и фреймворки в разработке корпоративных приложений: учебное пособие / Гайдаржи Г.Б., Федоров И.В. – М.: ИНФРА-М, 2020. – 189 с.

5.Ivanov, A., & Petrov, S. (2023). Digital Transformation in Hospitality: Property Management Systems and Revenue Optimization. New York, NY: Hospitality Press.

6.Kozlov, M., & Sokolova, E. (2022). "Designing Scalable Enterprise Systems with SSM Framework: A Case Study in Hotel Management." Journal of Information Technology in Tourism, 15(3), 45-62.

