У Инцзе , Магистрант,
Амурский государственный университет

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА УПРАВЛЕНИЯ ОТЕЛЯМИ НА ОСНОВЕ АРХИТЕКТУРЫ РАЗДЕЛЕНИЯ СПЕРЕДИ И СЗАДИ

Аннотация. С углублением цифровой трансформации глобальной индустрии гостеприимства традиционные модели управления сталкиваются с серьезными проблемами с точки зрения эффективности, интеграции данных и опыта работы с клиентами. Цель этой статьи – предложить и реализовать интеллектуальную систему управления отелями, основанную на зрелом технологическом стеке. Модуль поддержки системы построен с использованием фреймворков Spring, Spring MVC и MyBatis (SSM) для обеспечения стабильной и эффективной бизнес - логики и услуг передачи данных; Фронтальный интерфейс использует Vue.js 3.0 для создания динамического, отзывчивого пользовательского интерфейса, обеспечивающего полное разделение переднего и заднего концов. Благодаря внедрению ролевых моделей, основанных на управлении доступом (RBAC), система достигла тонкого управления правами. В документе подробно описывается общая архитектура системы, разделение основных функциональных модулей, концепции и физические модели базы данных, а также описывается конкретная реализация системы и всесторонняя проверка тестирования. Результаты показывают, что система не только полностью функциональна, стабильна в производительности, но и обладает хорошей масштабируемостью и ремонтопригодностью, может эффективно повысить эффективность работы отеля и качество обслуживания.

Ключевые слова: Система управления отелем, Рамки SSM; Vue.js, Разделение передней и задней частей, RBAC, Системные испытания.

Сегодня, когда глобализация и цифровая интеграция глубоко интегрированы, индустрия гостеприимства переживает беспрецедентные изменения. Ожидания путешественников вышли за рамки базовых потребностей в жилье и переключились на индивидуальный, интеллектуальный и бесшовный опыт проживания. В то же время растущая конкуренция внутри отрасли, как эффективно управлять все более сложными рабочими процессами, оптимизировать распределение ресурсов, снизить эксплуатационные расходы и обеспечить устойчивый рост доходов, стала основной проблемой, с которой должны иметь дело современные менеджеры отелей. Традиционная зависимость от ручной, децентрализованной и независимой модели управления затрудняет адаптацию к быстро меняющимся рыночным потребностям из - за присущих ей недостатков, таких как островки информации, избыточность процессов и отставание данных. Поэтому проведение цифровой трансформации, использование технических средств для управления и расширения возможностей, стало неизбежным выбором для индустрии гостеприимства для продвижения к высококачественному развитию. Именно в этом контексте возникла комплексная система управления гостиницами. Это больше не просто инструмент для поддержки бронирования, а интеллектуальный нейронный центр, который объединяет прием на стойке регистрации, управление жильем, управление пользователями, маркетинг членов и анализ больших данных. Система, о которой идет речь в этой статье, представляет собой комплексную практику проектирования и разработки, основанную на этой концепции и предназначенную для предоставления всеобъемлющего, эффективного и надежного цифрового решения для современных отелей.

Успешное построение системы зависит прежде всего от четкого, надежного и масштабируемого общего дизайна. В философии архитектуры система строго следует классическому шаблону - представлению - контроллеру (MVC) и использует эту идею во всех аспектах от организации обслуживания до рендеринга интерфейса интерфейса интерфейса.

При выборе технологии в качестве основного компонента системы используется широко признанная и хорошо зарекомендовавшая себя комбинация фреймворков SSM (Spring + Spring MVC + MyBatis). Будучи основным контейнером, фреймворк Spring с его мощными возможностями инжекции зависимости (DI) и программирования, ориентированного на разрез (AOP), унифицирует жизненный цикл и зависимость всех компонентов системы управления, эффективно снижая степень связи между модулями и обеспечивая ключевые функции корпоративного уровня, такие как декларативное управление транзакциями. Spring MVC выполняет функции веб - уровня, обеспечивая четкую реализацию MVC для обработки запросов интерфейса. С другой стороны, MyBatis, как устойчивая иерархическая структура, упрощает громоздкие операции JDBC с помощью гибкой конфигурации XML или аннотаций, сохраняя при этом способность разработчиков писать сложные SQL - операторы, обеспечивая эффективность и гибкость доступа к данным. Это сочетание гарантирует надежность, масштабируемость и обслуживание архитектуры системы.

В функциональной организации, благодаря углубленному анализу бизнес - процессов отеля, система научно разделена на 13 основных функциональных модулей с высокой степенью сцепления и низкой связью. Эти модули охватывают все сценарии работы магазина от пользователя до администратора. В частности, модуль регистрации и модуль регистрации пользователей для обеспечения безопасного входа в систему; Модуль отображения номеров для обслуживания клиентов, модуль бронирования, модуль пользовательских сообщений и модуль системных объявлений; Кроме того, модуль управления администраторами для внутреннего управления, модуль управления информацией о пользователях, модуль управления помещениями, модуль управления бронированием, модуль управления размещением, модуль управления сообщениями и модуль управления объявлениями. Каждый модуль имеет четкие обязанности и взаимодействует через хорошо определенные интерфейсы, которые вместе составляют органическое целое для совместной работы. Особого упоминания заслуживает то, что система использует ролевую модель управления доступом (RBAC) для управления правами, благодаря четкой цепочке зависимостей « пользователь - роль - права - меню», которая обеспечивает детализацию и гибкий контроль функционального доступа и диапазона данных пользователей разных ролей и закладывает прочную основу для безопасности системы.

Данные являются кровью системы, поэтому дизайн базы данных является ключевым элементом в построении системы. Система использует реляционную базу данных MySQL 8.0 для сохранения данных и строго следует третьей парадигме (3NF) в процессе проектирования, чтобы свести к минимуму избыточность данных и обновления аномалий. Построив детальную модель отношений сущность - отношение (E - R), были уточнены основные субъекты, такие как пользователь, администратор, номер, тип комнаты, заказ, регистрация проживания и другие, и их деловые связи между ними. На основе этой модели была разработана структура нескольких физических таблиц, включая таблицу администратора, таблицу пользователя, таблицу номеров, таблицу классификации номеров, таблицу бронирования, таблицу заказов, таблицу сообщений и таблицу объявлений. Каждая таблица определяет соответствующие основные и внешние клавиши, а также индексы для полей высокочастотных запросов, обеспечивая тем самым базовую поддержку данных для эффективной работы системы, обеспечивая при этом согласованность и целостность данных.

После завершения тщательного проектирования система вступила в стадию реализации. Система использует модель разработки с разделением спереди и сзади, в которой модуль обеспечивает чистый RESTful API, а интерфейс отвечает за рендеринг и взаимодействие с пользователями, которые передают данные через протокол HTTP, что значительно повышает эффективность разработки и техническое обслуживание системы.

Модульная реализация основана на Spring Boot, интегрирует структуру SSM и упрощает процесс конфигурации. Бизнес - логика инкапсулируется на уровне сервиса (Service Layer), уровень контроллера (Controller) отвечает за получение фронтальных запросов и возвращение ответов, в то время как постоянный уровень (Mapper) отвечает за взаимодействие с базой данных MyBatis. Что касается контроля разрешений, то с помощью фреймворка Spring Security или настраиваемого перехватчика реализован контроль безопасности на уровне интерфейса API, обеспечивающий доступ к определенным ресурсам только пользователям с соответствующими правами. Кроме того, декларативное управление транзакциями Spring гарантирует атомность и согласованность данных для основных бизнес - операций, таких как создание заказов, обработка проживания и т. Д. Система также интегрирована с протоколом WebSocket для реализации таких функций, как напоминание о новых заказах и push - рассылка объявлений в режиме реального времени, которые требуют от сервера активной отправки сообщений клиенту, что повышает производительность системы в режиме реального времени.

Фронтальная реализация является ключом к пользовательскому опыту современных веб - приложений. Эта система построена с помощью Vue.js 3.0 и его интерфейса Composition API и в сочетании с библиотекой компонентов Element Plus UI обеспечивает быстрый и интерактивный пользовательский интерфейс. Vue Router управляет маршрутизацией всех страниц и проверяет права доступа в глобальной передней страже. Библиотека управления статусом Pinia используется для централизованного управления глобальными состояниями, такими как информация о входе пользователя в систему, состояние корзины покупок (бронирование) и т. д., что делает обмен состояниями и отзывчивые обновления между компонентами простыми и управляемыми. Интерфейс интерфейса в полной мере учитывает многоуровневую адаптацию, адаптивный дизайн с использованием технологии компоновки Flex / Grid CSS3, обеспечивающей хороший визуальный и рабочий опыт в различных размерах экрана от настольных компьютеров до планшетов. Конкретно для различных функциональных интерфейсов, таких как страница показа номеров, встроенная в ротацию изображений и 3D панорамный просмотр; Процесс бронирования с помощью четких шагов направляет пользователя на выбор даты завершения, подтверждение типа номера, заполнение информации и онлайн - платежи; Функциональный интерфейс администрирования предоставляет панели просмотра данных, визуальные диаграммы и эффективные инструменты пакетной работы, которые значительно облегчают повседневную работу менеджеров.

Чтобы обеспечить качество доставки системы, мы провели всесторонние и строгие системные испытания, охватывающие различные измерения, такие как функциональность, производительность и совместимость.

Функциональное тестирование является основным звеном в проверке правильности системы в соответствии с требуемыми спецификациями. Мы разработали подробные тестовые примеры для всех 13 функциональных модулей. Процесс тестирования имитирует поведение реального пользователя, в том числе: пользователь от регистрации, входа в систему, запроса номера, представления бронирования, онлайн - оплаты, до фонового администратора для проверки заказа, обработки регистрации / выхода из магазина, обработки сообщений клиентов, публикации объявлений и другого полного набора бизнес - процессов. Результаты испытаний показывают, что все заданные функциональные точки могут нормально работать, бизнес -

процессы гладкие, данные передаются между модулями точно, а поведение системы соответствует ожидаемым целям проектирования.

Тесты производительности предназначены для оценки работы системы при высоких нагрузках. Мы уделяем особое внимание способности системы реагировать на сценарии высокочастотного доступа пользователей. Благодаря стресс - тестированию и мониторингу производительности ключевых интерфейсов, таких как загрузка списка номеров, представление и подтверждение заказов, рендеринг данных на главной странице и т. Д. Система демонстрирует хорошие показатели производительности. Данные тестирования показывают, что время загрузки основной страницы контролируется в течение 3 секунд, ключевые операции быстро реагируют, могут удовлетворить потребности большинства отелей в параллельном доступе в реальной работе, обеспечивая плавный опыт конечных пользователей.

Тестирование на совместимость гарантирует стабильную работу системы в разнообразной среде пользователей. Тесты охватывали различные платформы операционной системы (например, Windows 10 / 11) и основные веб - браузеры (включая Google Chrome, Mozilla Firefox, 360 Security Browser и т. Д.). В этих различных тестовых средах пользовательский интерфейс системы может нормально отображаться, все интерактивные функции работают безошибочно, без явных сбоев макета или функциональных сбоев. Окончательные результаты тестирования рекомендуют пользователям использовать комбинацию операционной системы Windows 10 с браузером Google Chrome для оптимального использования. Обобщая все результаты испытаний, система управления отелем была подтверждена высокой степенью целостности, стабильности и практичности, что соответствует стандартам доставки и ввода в эксплуатацию.

В этой статье систематически рассматривается полный жизненный цикл разработки интеллектуальной системы управления отелями, основанной на архитектуре SSM backend и передней фреймворке Vue.js, начиная с анализа потребностей, после общего проектирования, детальной реализации и, наконец, путем полного тестирования и проверки. Благодаря модульной и иерархической концепции дизайна система успешно построила современную платформу управления отелями с полным набором функций, четкими полномочиями и строгими моделями данных. Режим разработки с разделением спереди и сзади не только повышает эффективность разработки, но и делает систему на всех уровнях четкой ответственности, легко поддерживать и расширять. Строгий процесс тестирования доказывает, что система соответствует проектным ожиданиям с точки зрения функциональности, производительности и совместимости.

Заглядывая в будущее, система все еще имеет несколько направлений, которые заслуживают углубленного изучения и оптимизации. Во - первых, это глубокая адаптация мобильного устройства, и можно рассмотреть возможность разработки оригинального мобильного приложения или Progressive Web App (PWA) для удовлетворения потребностей пользователей в управлении своим маршрутом в любое время и в любом месте. Во - вторых, внедрение технологий искусственного интеллекта и анализа больших данных, путем анализа исторических данных о проживании, данных о поведении пользователей, создания интеллектуальной системы рекомендаций, предоставления клиентам персонализированных рекомендаций по типу жилья и обслуживанию, а также предоставления информации о ценах и маркетинговых стратегиях менеджеров. Наконец, с популяризацией технологий Интернета вещей (IoT) система может дополнительно интегрировать интеллектуальное оборудование для номеров, реализовать централизованный контроль и мониторинг состояния освещения, температуры, дверных замков и других объектов в номере, чтобы создать настоящий «умный номер», чтобы еще больше повысить интеллектуальный уровень отеля и конкурентоспособность на рынке.

Список литературы:

1. Иванов, А. В., & Петрова, С. К. (2020). Современные информационные системы в гостиничном бизнесе. М.: Издательство "ИНФРА-М.
2. Сидоров, Д. Л. (2022). *Разработка веб-приложений с использованием фреймворков Spring и Vue.js. // Программирование, (4), 45-54.
3. Кузнецова, Е. Н. (2021). Проектирование и реализация баз данных для корпоративных информационных систем. СПб.: Питер.
4. Федоров, М. П., & Васин, А. С. (2019). *Методы и средства тестирования программного обеспечения. // Информационные технологии, 25(3), 112-120.
5. Wallace, P., & Choi, J. (2021). Modern Hotel Property Management Systems: A Framework for Integration and Data Analytics. International Journal of Hospitality Management, 95, 102909.
6. Nelson, R. A. (2023). *Full-Stack Development with Spring Boot and Vue.js. (2nd ed.). O'Reilly Media.