



МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖИ РЕСПУБЛИКИ КРЫМ
Государственное бюджетное образовательное учреждение высшего
образования
Республики Крым
«КРЫМСКИЙ ИНЖЕНЕРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ГБОУВО РК КИПУ)

Факультет экономики, менеджмента и информационных технологий
Кафедра прикладной информатики

Направление подготовки 09.03.03 Прикладная информатика
Профиль «Прикладная информатика в информационной сфере»

БАКАЛАВРСКАЯ РАБОТА

Разработка плагина для управления гостиничных номеров

Студентки IV курса
группы И-2-15
очной формы обучения
Авдиль Салие Ленуровны

(подпись)

Научный руководитель:
ст.преп. Сейдаметов Г.С.

(подпись)

Работа допущена к защите
зав. кафедрой прикладной информатики,
д.пед.н., проф. _____ Сейдаметова З.С.
«__» _____ 2019 г.

Симферополь – 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ В СФЕРЕ РАЗРАБОТКИ ПЛАГИНА ДЛЯ УПРАВЛЕНИЯ ГОСТИНИЧНЫМИ НОМЕРАМИ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	7
1.1. Плагин, как средство модификации программ	7
1.2. Исследование предметной области управления гостиничными номерами	10
1.2.1. Классификация и разновидность гостиниц.....	10
1.2.2. Системы управления гостиничными номерами.....	12
1.2.3. Сравнительный анализ гостиниц	15
Вывод по главе 1	25
ГЛАВА 2. ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПЛАГИНА ДЛЯ УПРАВЛЕНИЯ ГОСТИНИЧНЫМИ НОМЕРАМИ И ИССЛЕДОВАНИЕ СРЕДСТВ РАЗРАБОТКИ.....	26
2.1. Унифицированный язык UML как средство визуального моделирования	26
2.2. Обзор современных средств разработки с целью реализации плагина для управления гостиничными номерами	28
2.2.1. Обзор используемых в работе языков программирования	29
2.2.2. Детальный анализ фреймворков	32
2.2.3. Системы управления базами данных, используемые в веб-разработке	35
2.2.4. Сравнительный анализ редакторов исходного кода	37
Вывод по главе 2	38
ГЛАВА 3. РАЗРАБОТКА ПЛАГИНА ДЛЯ УПРАВЛЕНИЯ ГОСТИНИЧНЫМИ НОМЕРАМИ	40
3.1. Создание базы данных в СУБД MySQL.....	40
3.2. Реализация компонентов веб-приложения	42
3.3. Проектирование плагина для управления гостиничными номерами ..	48
3.4. Проектирование административной панели при разработке плагина для управления гостиничными номерами	51
Вывод по главе 3	55
ЗАКЛЮЧЕНИЕ	56

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	58
ПРИЛОЖЕНИЕ А.....	61
ПРИЛОЖЕНИЕ Б.....	63
ПРИЛОЖЕНИЕ В.....	64
ПРИЛОЖЕНИЕ Г	65
ПРИЛОЖЕНИЕ Д.....	66
ПРИЛОЖЕНИЕ Е.....	67
ПРИЛОЖЕНИЕ Ж.....	70
ПРИЛОЖЕНИЕ З	71

ВВЕДЕНИЕ

Актуальность темы.

В современном мире приобретают большие обороты развитие новейших технологий и их широкое использование среди населения. Особенно это касается гостиничного бизнеса, который в современном мире уже не смог бы существовать без инновационных технологий. С каждым годом все больше людей пользуются услугами гостиниц, растет уровень обслуживания, обусловленный потребностями самих же гостей. Поэтому на сегодняшний день использование современных технологий в гостиничной индустрии является неотъемлемой частью ее развития.

В настоящее время гостиничный бизнес – одна из наиболее перспективных и быстроразвивающихся отраслей мировой экономики. Гостиничный бизнес является ключевой подотраслью туризма, который в свою очередь входит в число трёх крупнейших экспортных отраслей, уступая нефтедобывающей промышленности и автомобилестроению. Значение гостиничной индустрии в мире постоянно увеличивается в связи с ежегодным ростом количества гостиниц и отелей в различных странах.

Актуальность данной темы обусловлена стремительным развитием рынка гостиничных услуг, характерными чертами которого на сегодняшний день являются динамичные изменения туристического спроса и предложений. Коммерческий успех гостиничного бизнеса зависит также от того, какие услуги оказывает та или иная гостиница и насколько качественно все это делается.

Безусловно, успешные гостиницы и отели стремятся оказать как можно больше разнообразных услуг, как можно большему количеству клиентов. Данная модель коммерческого успеха является характерной для большинства предоставляемых услуг на нынешнем рынке. Многие отели стремятся максимально разнообразить свой ассортимент услуг для того, чтобы удовлетворить как можно больше потребностей клиента.

Успех данной отрасли в первую очередь зависит от того, насколько качественно развита ассортиментная политика и использование современных средств IT-технологий.

Известно, что IT-технологии получили глобальное распространение во всех областях, в частности и в сфере гостиничного бизнеса. Современный успех развития сферы гостиничной индустрии – это внедрение и использование в данной области информационных технологий. Соответственно, **актуальностью** данной бакалаврской работы является разработка расширения (плагина) для управления гостиничными номерами.

Целью данной бакалаврской работы является разработка программного модуля для управления гостиничными номерами, который позволит автоматизировать работу сферы гостиничного бизнеса.

Для достижения цели в рамках бакалаврской работы были поставлены следующие **задачи**:

- теоретически обосновать необходимость и востребованность плагинов;
- провести описание предметной области;
- провести сравнительный анализ систем управления гостиничными номерами;
- осуществить визуальное моделирование с помощью языка UML;
- выполнить обзор современных средств разработки;
- создать базу данных в СУБД MySQL;
- реализовать компоненты веб-приложения;
- спроектировать и разработать программный модуль;
- разработать административную панель приложения.

Объектом бакалаврской работы является процесс разработки плагина.

Предметом бакалаврской работы является разработка плагина для управления гостиничными номерами.

Практическая значимость бакалаврской работы заключается в том, что разработан плагин для управления гостиничными номерами и административная панель для этого плагина.

Структура и объем бакалаврской работы. Пояснительная записка бакалаврской работы состоит из введения, трех разделов, выводов к ним, общего заключения, списка использованных источников и приложений. Полный объем пояснительной записки – 68 страниц печатного текста, из них 58 страниц основного текста. Пояснительная записка курсового проекта проиллюстрирована 48 рисунками. Список использованных источников охватывает 25 наименований, объем приложений составляет 10 страниц.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ В СФЕРЕ РАЗРАБОТКИ ПЛАГИНА ДЛЯ УПРАВЛЕНИЯ ГОСТИНИЧНЫМИ НОМЕРАМИ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Плагин, как средство модификации программ

В современном мире область информационных технологий считается наиболее динамически развивающейся. Учитывая глобальную компьютеризацию и постоянное увеличение доступа к определенной информации это является естественным ходом развития событий. Создается огромное количество новых приложений и требуется совершенствование уже существующих. Однако, сегодня нет необходимости в написании сверхсложных приложений, предусматривающих все потребности пользователя в базовой программе, поскольку существует возможность создания дополнительных модулей, которые пользователь сможет подключить при необходимости. Такими дополнительными модулями являются плагины. Таким образом, базовая программа будет упрощенной и компактной, а пользователи смогут подключать только те плагины, которые будут им необходимы. Это будет способствовать комфортному использованию того или иного приложения, при этом не усложняя работу основного и исключая подключения ненужных модулей.

Плагин (происходит от англ. plug in – «подключать») – это независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей[1].

Другими словами, плагином называют небольшое дополнение к программе, которое создается отдельно и в случае необходимости может быть подключено к уже работающему приложению. Основной его задачей является расширение функциональности основной программы.

Плагины не могут работать сами по себе отдельно и обязательно относятся к какому-то приложению. При этом основное программное обеспечение (ПО) функционирует самостоятельно, в нем реализована возможность

добавлять один или несколько плагинов, обновлять или удалять их. И это никак не сказывается на работоспособности программы.

Стоит отметить, что существует огромное количество плагинов практически для всех видов программного обеспечения для компьютера[2]. Рассмотрим самые популярные и востребованные из них:

- Плагины для сайтов. Любой, удобный для пользователя сайт использует огромное количество плагинов для того, чтобы он функционировал необходимым образом. В этой категории чаще всего применяются плагины для безопасности сайта, для резервного копирования, настройки и исправления ссылок (транслитерации), удобного функционирования веб-сайта и т.д.
- Плагины для браузера. Наиболее популярным плагином для браузеров является всем известный Adobe Flash Player, который воспроизводит flash-анимацию и видеоролики на различных сайтах. Также к этому виду относятся плагины, позволяющие скачивать видео с YouTube или музыку из Вконтакте, обходить блокировку сайтов и т.д. Все они работают только в паре с браузером и не могут работать отдельно.
- Плагины для редакторов изображений. Данные плагины помогают использовать различные эффекты для обработки и новые фильтры.
- Плагины для программ по работе со звуком. Они подключают библиотеки сэмплов, то есть образцов звука и позволяют использовать инструменты для накладки эффектов на звук (эхо, эффекты зала, маленькой комнаты, и т.д.).
- Плагины для игр. Данный вид плагинов вносит новые возможности в игровой процесс (новые уровни, персонажи) и они не могут работать отдельно от исходной игры.

Рассмотрим более подробно основные задачи, которые помогают решить плагины[3]:

1. Упрощение разработки базовой версии программы. То есть изначально можно выпустить небольшую компактную программу, которая будет предусматривать подключение дополнительных модулей, таким образом, чтобы все доработки функций в дальнейшем можно было подключить отдельно. Это поможет оптимизировать как ресурсные, так и временные затраты. Также это сэкономит объемы памяти устройств пользователей, на которых будет развернуто приложение, что положительно повлияет на скорость его работы. Аналогичным образом возможность выбора нужного набора модулей позволит сэкономить рабочее пространство, поскольку не всем нужно ежеминутно иметь доступ к просмотру погоды, афиши и прочей полезной информации.
2. Учет индивидуальных потребностей. Данный пункт касается личных предпочтений каждого пользователя. То есть человек может установить себе только тот набор дополнительных модулей, который будет соответствовать его потребностям и вкусам.
3. Расширение функционала программ. Дополнительные функции позволяют делать ПО более гибким. Если возникает необходимость в каком-либо специфическом модуле, его можно подключить, доработать или даже написать самостоятельно, после чего вы получите необходимую работающую систему.

Существуют также официальные и неофициальные плагины:

1. Официальные плагины — это модули, разработанные компанией, которая создала исходную программу, либо компанией-партнёром с согласия разработчика программы и под его контролем. Такие плагины наиболее безопасны для пользования, однако за их установку или ежемесячное использование может взиматься дополнительная плата.
2. Неофициальные плагины. В отличие от официальных плагинов, неофициальные может создать каждый. Однако они не всегда бывают такими же полезными, практичными и конструктивными как

официальные. Зачастую они оказываются недоработанными, что приводит к затруднению их использования, либо делает его в принципе невозможным. Также нередко неофициальные плагины содержат в себе вредоносный код, который может нанести ущерб либо исходной программе, либо компьютеру в целом. Поэтому, используя лицензионную программу стоит доверять только крупным и проверенным разработчикам дополнительных модулей.

1.2. Исследование предметной области управления гостиничными номерами

1.2.1. Классификация и разновидность гостиниц

На сегодняшний день гостиничный бизнес развит достаточно широко и повсеместно. Огромное количество людей пользуются данными услугами, а некоторые категории населения, к примеру, такие как: отдыхающие на курорте, «командировочники», представители шоу-бизнеса и т.п. пользуются этими услугами довольно часто. И, безусловно, каждый выбирает отель или гостиницу в зависимости от её ценовой политики, расположения и т.д.

Гостиницей называют имущественный комплекс (здание, дом, часть здания, иные постройки) с меблированными комнатами, то есть номерами, для временного проживания[4].

Все эти факторы, а именно ценовая политика, расположение и многое другое, помогающие людям выбрать подходящий лично для них отель, напрямую зависят от типа управления гостиничными номерами и гостиницей в целом – ведь именно система управления номерами и персоналом гостиницы обеспечивает её востребованность на рынке услуг и качественное функционирование. Долгое существование на рынке услуг и достойное конкурентное преимущество с предприятиями аналогичного направления может обеспечить только грамотное управление предприятием.

Качество и стоимость предоставляемых услуг в гостинице зависит от её типа. Благодаря огромной востребованности гостиничных услуг на сегодняшний день, эталоны качества сервиса всюду значительно возросли. Вследствие этого ввели звездную систему классификации (от одного до пяти) для того, чтобы было проще определить уровень гостиницы.

Данная классификация выглядит следующим образом[5]:

- Одна звезда (Cat D) – недорогая гостиница, в которой предусмотрено минимум услуг, к примеру уборка номера может быть не повседневная. Номера все одного типа.
- Две звезды (Cat C) – гостиница с малым бюджетом, предусматривающая минимум услуг, в который обязательно входит ежедневная уборка номеров. Предполагает не более двух различных типа номеров.
- Три звезды (Cat B) – считается гостиницей среднего класса, включающей стандартный комплекс услуг, а именно: повседневную уборку номеров, телевизор, мини-бар или холодильник во всех номерах; на территории гостиницы предусмотрена прачечная для постояльцев, бизнес-центр, тренажёрный зал, бассейн, а также место для предоставляемого гостиницей завтрака. В тоже время некоторые из этих услуг могут отсутствовать или могут быть другие. Предполагаются различные виды номеров (одноместные, двухкомнатные, для курящих, не курящих и т. д.).
- Четыре звезды (Cat A) – гостиница достаточно высокого класса, предусматривающая как все вышеуказанные услуги, так и имеющая дополнительные, как например: спа, массаж, несколько баров и ресторанов на территории, конференц-залы). В свою очередь цены на номера в данных гостиницах гораздо выше средних.
- Пять звёзд (Cat De Luxe) – «люксовая» гостиница. Характерной особенностью таких гостиниц является обширный комплект услуг, исключительно эксклюзивный, к примеру: частный гольф-

клуб на территории отеля, многокомнатные апартаменты с прислугой и т.д. Гостю уделяется персональное внимание, учитываются все его пожелания. Эти гостиницы предполагают наличие различных типов номеров. Как следствие, цены на проживание и на дополнительные услуги, в номерах этих отелей крайне высоки.

Исходя из вышеприведенной классификации также можно определить, что с точки зрения структуры и функционирования отеля он предполагает наличие номеров с различным уровнем комфортности, сервиса и, соответственно, оплаты.

В основном номера бывают следующих типов:

- люкс – многокомнатный номер, предполагающий высокий уровень комфортности, обслуживания и сервиса;
- полулюкс – номер по площади меньший, чем люкс, однако с достаточно высоким уровнем сервиса и комфортности;
- стандарт – предполагает минимальный уровень сервиса.

При вселении всем клиентам размещаемым в гостинице необходимо заполнить карту регистрации, в которой указывается фамилия, имя, отчество, дата рождения, адрес места жительства, паспортные данные, время заселения, время отъезда и т.д. Все номера отелей имеют уникальный номер, с помощью которого ведется учет клиентов, проживающих в нём.

1.2.2. Системы управления гостиничными номерами

Система управления представляет собой совокупность подсистем, каждая из которых отвечает за управление определенным участком, а во взаимосвязи друг с другом – за работу всей системы. Руководству приходится иметь дело со сложной и разнообразной информацией, которую нельзя обработать и упорядочить, не имея в своем распоряжении специального алгоритма.

Существует огромное количество различных систем управления гостиничными номерами. Проведем сравнительный анализ основных из них.

1. Система управления VDA Micromaster включает в себя различные функции[6]:

- *Функция энергосбережения.* После того, как система определяет появления гостя в номере (с помощью специальных магнитноконтактных датчиков), функция начинает выполнять свою работу – номер активируется, температура комнаты изменяется до комфортной, а гость получает доступ к управлению освещением в номере посредством сенсорных панелей.
- *Функция управления микроклиматом номера.* В гостиничном номере устанавливается специальная панель для управления климатом со встроенным термодатчиком. Данная система устанавливает определенные климат-статусы как комнаты или нескольких комнат, так и всего номерного фонда по определенным критериям и сценариям: 1. По присутствию гостя; 2. По сезонному сценарию; 3. По полученной информации от PMS.
- *Функция управления освещением в номере гостиницы.* Чтобы управлять освещением, в номере предполагается установка сенсорных панелей управления настенного исполнения, а также прикроватные панели управления.
- *Функция управления и мониторинга периферийного оборудования.* Система Micromaster предполагает управление и отслеживание состояния периферийного оборудования, которое может включать в себя:
 - Управление электрическим звонком;
 - Детекцию сигналов открытия окон/дверей;
 - Управление механизацией номера, например, управление шторами, раздвижными ролл-ставнями в номерах повышенной комфортности и т.п.
- *Функция управления номерным фондом.* Система Micromaster предполагает реализацию следующего функционала:

- Определение наличие гостя в номере;
- Определение состояния микроклимата в номере;
- Просмотр статуса комнат: «Не беспокоить» и «Убрать комнату»;
- Отображение сигналов аварийных кнопок;
- Интеграцию с PMS гостиницы.

2. Система управления гостиничными номерами Hotel Solution решает следующие задачи[7]:

- Интеграция контроля доступа, контроля температуры, энергопотребления и специальных функций гостевых номеров (т.к. «Не беспокоить» или «Вызов сервиса»);
- Станция управления Hotel Solution на рецепции имеет связь в режиме реального времени с гостевыми номерами;
- Интеграция в главную систему управления через стандартный интерфейс;
- Контроль систем отопления и охлаждения;
- Контроль микроклимата во всех общих зонах отеля, таких как вестибюль, рестораны, кухня, фитнес и т.д.;
- Интеграция всех систем здания, таких как освещение, жалюзи, безопасность, распределение энергии и т.д.

3. Система управления гостиничными номерами EcoStruxure™[8] предполагает обеспечение комфорта, удобства управления и средств связи для постояльцев, также полностью интегрируется с системами управления гостиницы. Помогает удовлетворить все потребности клиентов и оптимизировать работу и потребление энергии.

Также система EcoStruxure™ помогает решить следующие задачи:

- Управление и контроль энергопотреблением;
- Выявление и прогнозирование необходимости технического обслуживания;
- Упрощение запросов на услуги по уборке;

- Дистанционный поиск и устранение неисправностей.

1.2.3. Сравнительный анализ гостиниц

Данная бакалаврская работа предполагает разработку плагина для управления гостиничными номерами. Чтобы реализовать удобный и качественный плагин необходимо продумать его структуру, формализацию и внешний вид. Для этого проведем небольшой сравнительный анализ нескольких гостиниц. В качестве примера возьмем официальный сайт этно-отеля «Jeval»[9], а также официальный сайт отеля «Donna Rosa»[10].

Сайты располагают полной информацией о номерах и их расценках, различных специальных предложениях, информацией об имеющихся сервисах, а также предоставляют контакты и отзывы. Одним из преимуществ этих отелей является поиск необходимых номеров прямо на главной странице сайта (Рис.1.2.1, Рис.1.2.2). То есть указав период проживания и количество человек осуществляется автоматический подбор списка всех доступных номеров под указанные данные (Приложение А).

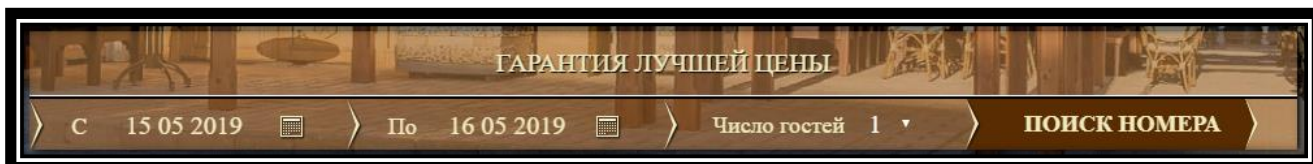


Рис.1.2.1 Поиск номера на сайте этно-отеля «Jeval»

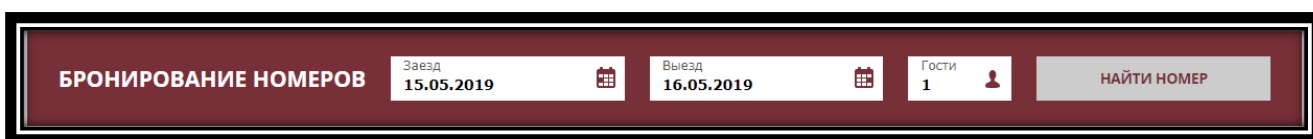


Рис.1.2.2 Поиск номера на сайте отеля «Donna Rosa»

Также пользователю предоставляется каталог различных номеров (Рис.1.2.3, Рис.1.2.4). В случае, если пользователя заинтересует какой-либо номер он может посмотреть всю подробную информацию о нем, нажав на кнопку *Подробнее* (Рис.1.2.5, Рис.1.2.6).



Рис.1.2.3 Каталог номеров этно-отеля «Jeval»

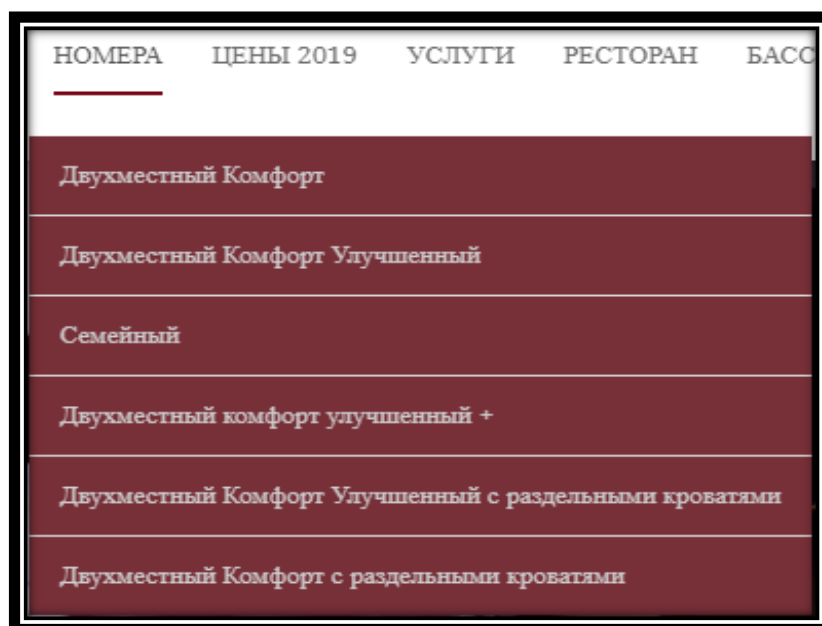


Рис.1.2.4 Каталог номеров отеля «Donna Rosa»



Рис.1.2.5 Общая информация о номере этно-отеля «Jeval»

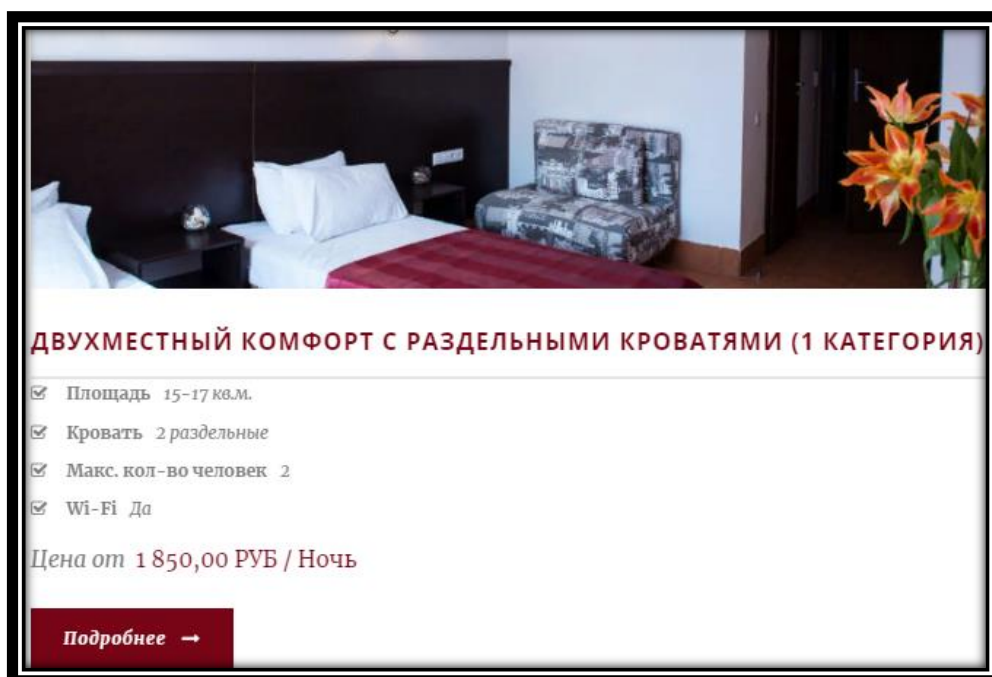


Рис.1.2.6 Общая информация о номере отеля «Donna Rosa»

Рассмотрим более детально информацию о номерах отелей. В качестве примера номера отеля «Джеваль» возьмем «Стандарт 2-х местный, без балкона». Итак, при переходе по вкладке *Подробнее* пользователь может увидеть детальную информацию об этом номере (количество кроватей, наличие телевизора, сейфа, кондиционера и многое другое) (Рис.1.2.5), ему представлена сезонная тарификация номера (Рис.1.2.7) и фотогалерея (Рис.1.2.8), а также клиент имеет возможность сразу забронировать этот номер (Рис.1.2.9).

ЦЕНА ЗА ПРОЖИВАНИЕ В НОМЕРЕ						
	01.01 - 28.02	01.03 - 31.05	01.06 - 31.08	01.09 - 15.10	16.10 - 30.11	01.01 - 28.02
Варианты	Завтрак	Завтрак	Завтрак	Завтрак	Завтрак	Завтрак
Основное место - 2 взрослых в номере	4000	5000	8000	7000	4500	4000
Дополнительное место - 1 взрослый	1500	1500	2000	1500	1500	1500
Дополнительное место - 1 ребенок (от 0 до 3-х лет)	0	0	0	0	0	0
Дополнительное место - 1 ребенок (от 3-х лет до 12-ти)	1000	1000	1500	1000	1000	1000

Рис.1.2.7 Сезонная тарификация номера «Стандарт 2-х местный, без балкона» в отеле «Джеваль»



Рис.1.2.8 Фотогалерея номера «Стандарт 2-х местный, без балкона» в отеле «Джеваль»

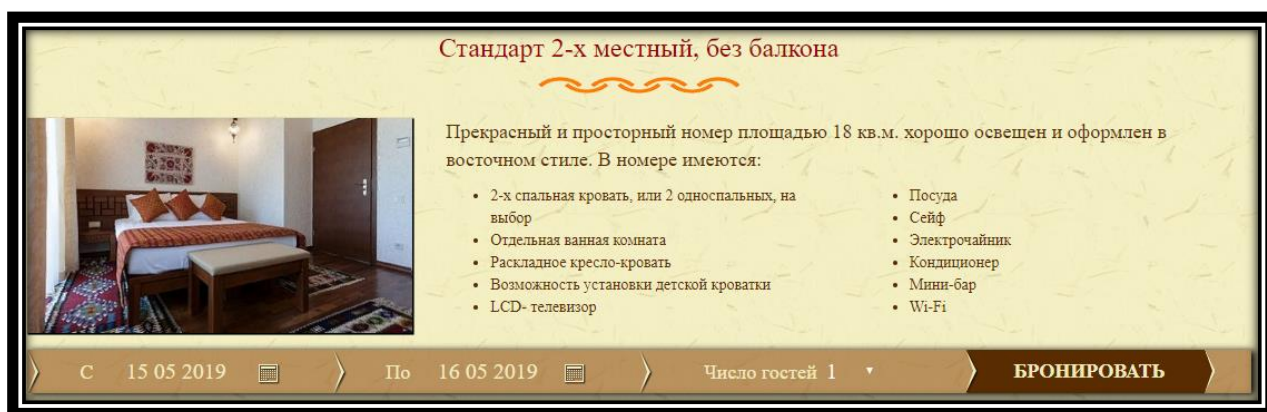








Рис.1.2.9 Панель бронирования номера «Стандарт 2-х местный, без балкона» в отеле «Джеваль»

Номер «Двухместный комфорт» будет выступать в качестве примера в отеле «Донна Роза». Итак, перейдя по вкладке *Подробнее* пользователь может наблюдать общее описание номера, а также всю детальную информацию о нем (Рис.1.2.10). Как и отель «Джеваль», отель «Донна Роза» предполагает сезонную тарификацию номеров (Рис.1.2.11) и имеет фотогалерею каждого номера (Рис.1.2.12). Здесь же предусмотрена панель бронирования этого номера (Рис.1.2.13).



ДВУХМЕСТНЫЙ КОМФОРТ (1 КАТЕГОРИЯ) Цена от **1 850,00 РУБ / Ночь**

 15-17 кв.м.
  ДВУХМЕСТНАЯ
  2
  ДА
  ДА

Описание
 Номера категории «Двухместный Комфорт» – наиболее экономичный вариант размещения в нашем отеле для молодой семьи.


Площадь номера: 15-17 кв.м. В номере: двуспальная кровать, возможна установка детской кроватки для малыша до 2-х лет.

Также в номере:


- встроенный шкаф;
- кондиционер;
- сейф;
- холодильник - минибар;
- LCD-телевизор с русскоязычными каналами;
- доступ в Интернет по Wi-Fi;
- электрочайник и набор посуды для приготовления кофе и чая;
- оборудованная ванная комната;
- французский балкон.


Рис.1.2.10 Детальная информация номера «Двухместный комфорт» в отеле «Донна Роза»


Сезонная тарификация номера

 01.01.19 – 27.04.19, 13.05.19 – 31.05.19, 01.10.19 – 31.12.2019

Р
 Тур «Завтрак» – 1850; Тур «Полный пансион» – 2700

 28.04.19 – 12.05.19

 01.06.2019 – 30.06.2019, 16.09.2019 – 30.09.2019

 01.07.2019 – 31.08.2019


 01.09.2019 – 15.09.2019

Рис.1.2.11 Сезонная тарификация номера «Двухместный комфорт» в отеле «Донна Роза»

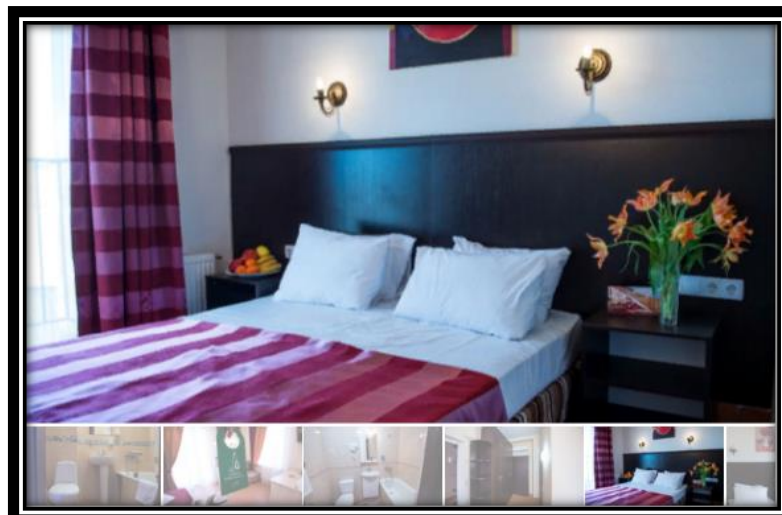


Рис.1.2.12 Фотогалерея номера «Двухместный комфорт» в отеле «Донна Роза»

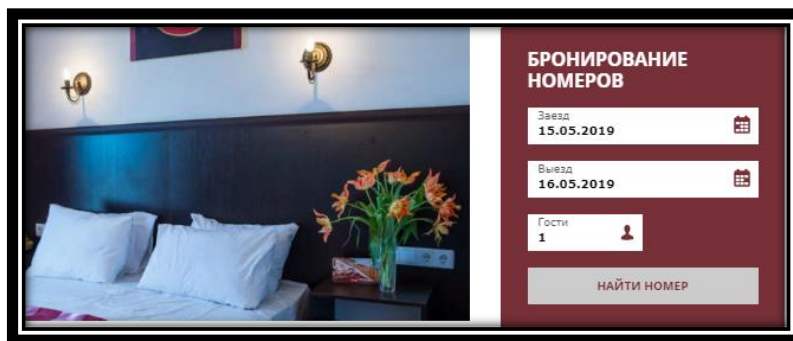


Рис.1.2.13 Панель бронирования номера «Двухместный комфорт» в отеле «Донна Роза»

Забронировать номера на сайтах отелей «Джеваль» и «Донна Роза» можно, как уже было упомянуто выше, либо с главной страницы сайта, либо в разделе «Номера», перейдя по вкладке «Подробнее». Для осуществления бронирования пользователю необходимо заполнить форму, предоставляя более подробную информацию о себе.

Рассмотрим более подробно форму бронирования на сайте этно-отеля «Jeval» (Рис.1.2.14). Данная форма требует следующие поля для заполнения:

- С (в этом поле пользователь указывает дату заезда);
- По (поле, в котором пользователь указывает дату выезда);
- Взрослых (вводится количество взрослых гостей);
- Детей (количество детей (необязательно для заполнения));
- Контактное лицо (данное поле предназначено для заполнения имени клиента, осуществляющего бронирование;
- Страна (в этом поле указывается страна в которой проживает пользователь (также необязательно для заполнения));
- E-mail (поле, для указания электронной почты гостя);
- Телефон (данное поле предназначено для номера телефона клиента.
- Сообщение (поле, предназначенное для заполнения дополнительной информации или пожеланий будущего гостя (необязательно для заполнения))

Рис.1.2.14 Форма бронирования номера этно-отеля «Джеваль»

Поля «С» и «По» в форме бронирования имеют Календарь (Рис.1.2.15), который позволяет выбрать дату заезда и выезда, при этом автоматически деактивируя прошедшие даты.

Рис.1.2.15 Календарь бронирования на сайте этно-отеля «Джеваль»

Бронирование номеров на сайте отеля «Donna Rosa» делится на несколько этапов. Первым этапом является заполнение основной формы (Рис.1.2.16), в которую входят следующие поля и пункты:

- Дата заезда (пользователь отмечает дату заезда);
- Дата выезда (пользователь отмечает дату выезда);
- Размещение в номере (данный пункт предполагает выбор количества взрослых гостей и детей);

- Нужен еще один номер (этот пункт позволяет добавить дополнительные номера при необходимости);
- Выбор языка и валюты (также при необходимости пользователь может выбрать язык и валюту).

Рис.1.2.16 Основная форма бронирования номера отеля «Донна Роза»

Даты заезда и выезда также как и на сайте отеля «Джеваль» заполняются с помощью календаря бронирования (Рис.1.2.17).

Рис.1.2.17 Календарь бронирования на сайте отеля «Donna Rosa»

Следующим этапом бронирования на сайте отеля «Donna Rosa» является выбор подходящего номера под введенные данные. После чего предлагается заказать дополнительные услуги (например трансферы) и далее появляется еще одна форма бронирования, которая предполагает заполнение информации о гостях (Приложение Б). Данная форма в свою очередь также делится на 3 подпункта:

- Как с вами связаться (поля для ввода электронного адреса и номера телефона) (Рис.1.2.18);
- Информация о гостях (поля для ввода ФИО клиента, его гражданства, времени заезда и выезда, а также при необходимости добавления пожеланий клиента) (Рис.1.2.19);
- Выберите способ оплаты (для удобства клиента предоставляется несколько различных способов оплаты: оплата при заселении, банковский перевод, безналичный расчет, банковская карта, электронный платеж) (Рис.1.2.20).

Как с вами связаться

На этот адрес будет выслано подтверждение бронирования с возможностью отмены брони

Пожалуйста, введите номер телефона. Пример: +79010000000

☐ Я хочу получить **бесплатное** SMS-подтверждение бронирования

☐ Я хочу узнавать о специальных предложениях и новостях отеля по email или SMS

Рис.1.2.18 Раздел «Как с вами связаться» на сайте отеля «Донна Роза»

Информация о гостях

Номер: Двухместный Комфорт с раздельными кроватями (твин). 1 взрослый

Время заезда и выезда

Вы можете выбрать удобное для вас время заезда и выезда.
Услуга позднего выезда предоставляется за дополнительную плату.
Стандартное время заезда — 14:00, выезда — 12:00.

Заезд

Выезд

Ранний заезд и поздний выезд возможны только при наличии свободных номеров

Дополнительные комментарии

Здесь вы можете написать, что вам необходимы дополнительная кровать для ребёнка или другие услуги

Рис.1.2.19 Раздел «Информация о гостях» на сайте отеля «Донна Роза»

Выберите способ оплаты

Фактом бронирования вы соглашаетесь с [обработкой персональных данных](#) и [политикой конфиденциальности](#) и [пользовательским соглашением](#)

Самый удобный способ **Оплатить при заселении**

Выбирая этот способ оплаты, вы НЕ ВНОСИТЕ ПРЕДОПЛАТУ за бронь. Получаете ВАУЧЕР и предъявляете его при заселении.

Без предоплаты
ЗАБРОНИРОВАТЬ

Банковский перевод
(для физических лиц)

Оплачивается **вся сумма** брони.

Вы получаете подтверждение бронирования и квитанцию для оплаты в банке. Квитанцию необходимо оплатить в течение 3 банковских дней.

Предоплата
3 500 Р
ЗАБРОНИРОВАТЬ

Безналичный расчет
(для юридических лиц)

Оплачивается **вся сумма** брони.

Вы получаете подтверждение бронирования и счет на оплату. Счет необходимо оплатить в течение 3 банковских дней.

Предоплата
3 500 Р
ЗАБРОНИРОВАТЬ

Банковская карта
(Visa, MasterCard, Maestro, MIR, JCB, UnionPay)

Оплачивается **вся сумма** брони.

Кассовый чек в электронном виде будет предоставлен для вас компанией TravelLine на адрес электронной почты, указанный вами в процессе бронирования. Обращаем ваше внимание на то, что средство размещения, в котором вы бронируете номера (услуги) и будете проживать, выдаст вам документы, подтверждающие факт проживания за весь период, в том числе оплаченный с использованием банковской карты через интернет.

Предоплата
3 500 Р
ЗАБРОНИРОВАТЬ

Рис.1.2.20 Раздел «Выберите способ оплаты» на сайте отеля «Донна Роза»

Проведя сравнительный анализ сайтов этно-отеля «Jeval» и отеля «Donna Rosa» были рассмотрены различные виды номеров, а также подробная информация о каждом из них с возможностью последующего бронирования. Вследствие этого были выделены следующие категории номеров, которые будут реализованы при разработке плагина для управления гостиничными номерами:

- Номер «Стандарт»;
- Номер «Люкс»;
- Номер представительского класса.

Также в процессе сравнительного анализа, как упоминалось выше, было выявлено такое полезное преимущество, как форма поиска свободных номеров по указанной информации на главной странице, которое значительно экономит временные затраты пользователя и облегчает процесс поиска необходимых номеров. Таким образом, при разработке плагина управления гостиничными номерами будет реализована такая же форма для сайта гостиничного комплекса «Blu hotels».

Вывод по главе 1

В ходе написания первой главы бакалаврской работы был подробно изучен основной теоретический материал в области программных модулей. Выявлены различные плагины практически для всех видов программного обеспечения для компьютера, а также рассмотрены основные задачи, которые выполняют программные модули.

Следующим этапом был проведен общий анализ предметной области, а именно: приведена звездная классификация, которая помогает в определении уровня отелей; выявлены основные типы номеров (люкс, полулюкс, стандарт). После чего отдельно были рассмотрены различные системы управления гостиничными номерами и проведен их сравнительный анализ. Последним этапом в написании данной главы являлось проведение сравнительного анализа работы отелей и управления ими на примерах этно-отеля «Jeval» и отеля «Donna Rosa», которые в дальнейшем послужили для определения основных критериев разрабатываемого плагина.

ГЛАВА 2. ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПЛАГИНА ДЛЯ УПРАВЛЕНИЯ ГОСТИНИЧНЫМИ НОМЕРАМИ И ИССЛЕДОВАНИЕ СРЕДСТВ РАЗРАБОТКИ

2.1. Унифицированный язык UML как средство визуального моделирования

В современных условиях возрастания информационной нагрузки требуются новые методы по оперативному и эффективному усвоению больших объемов информации. Одним из решений этой проблемы является визуальное моделирование. Этот вид моделирования для представления данных чаще всего использует изображения, потому как обычно оно является более информативным и менее абстрактным видом представления, то есть обладает более гибким каналом, чем текст или цифра[11].

Визуальное моделирование представляет собой сложное комплексное моделирование, требующее значительных вычислительных ресурсов. Источником визуального моделирования являются данные получаемые в информационном поле[12]. Для графического представления модели, зачастую используют язык UML (Unified Modeling Language – унифицированный язык моделирования).

UML – это стандартный инструмент, предназначенный для создания документированных каркасов («чертежей») программного обеспечения, а также помогающий визуализировать, конструировать, специфицировать и документировать процесс разработки программных систем[13].

Для реализации графического представления модели при разработке плагина управления гостиничными номерами, был выбран веб-сервис Draw.io, предназначенный для формирования различных диаграмм и схем.

Диаграмма вариантов использования (use case diagram) предназначена для описания функционального назначения системы или, иначе говоря, для описания того, что система будет делать в процессе своего функционирования[14]. Диаграмма вариантов использования является исходным концептуальным представлением, или концептуальной моделью системы, в процессе

ее проектирования и разработки. Суть такой диаграммы заключается в том, что проектируемая система представляется в виде множества сущностей, или актеров, которые взаимодействуют с системой посредством так называемых вариантов использования.

При разработке плагина для управления гостиничными номерами были выявлены следующие актеры (сущности): гость, администратор, номерной фонд.

Для сущности гость был определен следующий набор действий (Рис.2.1.1):

- Поиск номера
- Выбор номера
- Изучение номера
- Бронирование номера
- Заполнение формы бронирования

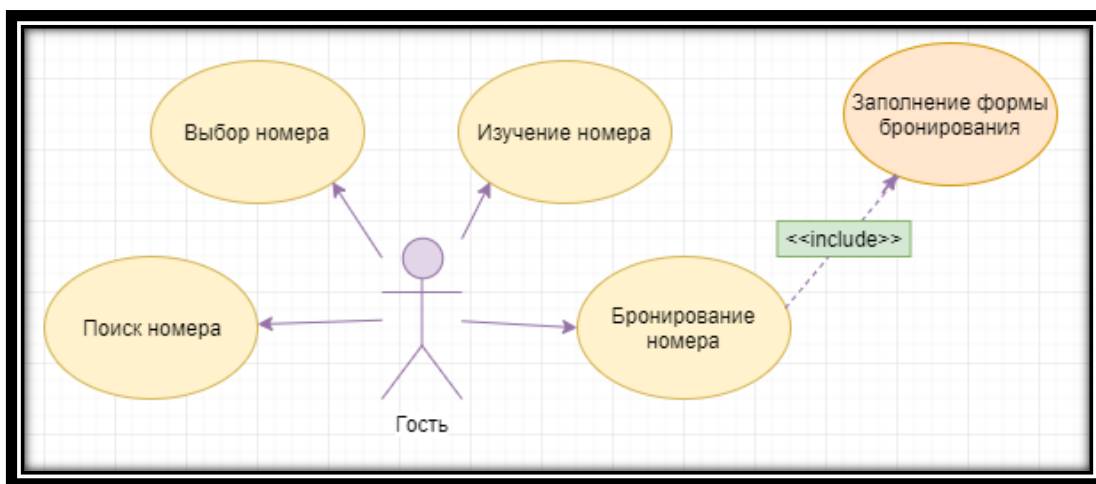


Рис.2.1.1 Диаграмма вариантов использования для сущности *Гость*

Функциональными возможностями сущности администратор являются (Рис.2.1.2):

- Добавление номера
- Удаление номера
- Просмотр листа бронирования
- Добавление новости
- Изменение новости

- Удаление новости

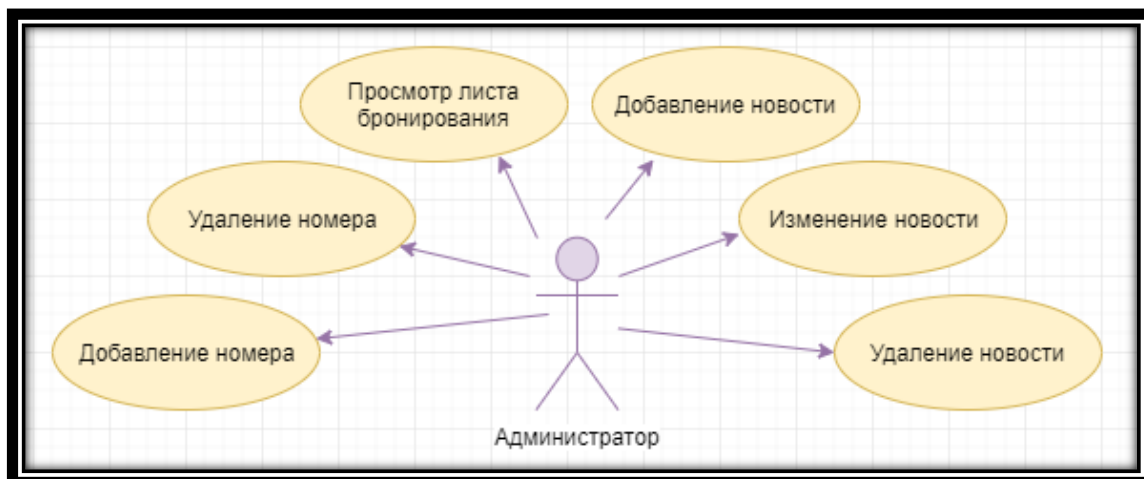


Рис.2.1.2 Диаграмма вариантов использования для сущности *Администратор*

Сущность номерной фонд содержит такие варианты использования как (Рис.2.1.3):

- Есть номер
- Нет номера

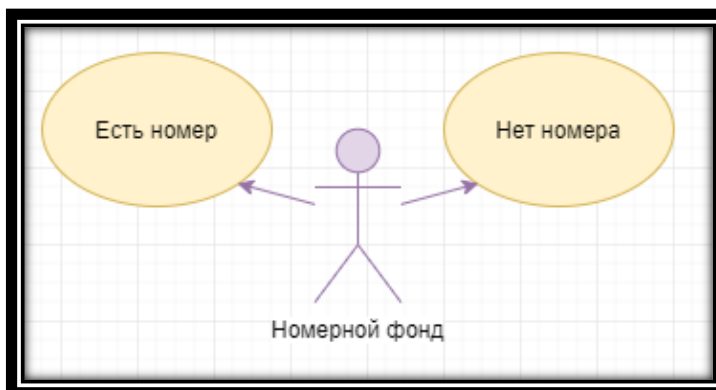


Рис.2.1.3 Диаграмма вариантов использования для сущности *Номерной фонд*

В результате визуального моделирования средствами Draw.io были построены три диаграммы вариантов использования для сущностей: гость, администратор, номерной фонд.

2.2. Обзор современных средств разработки с целью реализации плагина для управления гостиничными номерами

На сегодняшний день для создания веб-сайтов и веб-приложений существует огромное количество средств разработки как для клиентской, так и для серверной части. В данной работе рассмотрим следующий инструментарий: языки веб-программирования, системы управления базами данных, фреймворки и текстовые редакторы.

2.2.1. Обзор используемых в работе языков программирования

Сегодня практически все компании, организации и магазины имеют свой собственный сайт, который выполняет функцию рекламы, а также, что немаловажно, упрощает работу с клиентами. Это говорит о том, что веб-программированием является создание приложений и сайтов, которые работают в сети. Связующим звеном между программистом и программами является язык программирования, а между веб-разработчиком и сайтами – *язык веб-программирования*.

Язык веб-программирования – это множество операторов, посредством которых создаются коды веб-программ, также их называют скриптами и сценариями[15]. Язык программирования передает понятные компьютеру инструкции для выполнения определенных операций. Таким образом, посредством языков программирования человек «общается» с машиной. Зачастую коды, которые написаны на веб-языках, читаются браузерами. К наиболее распространенным языкам веб-программирования можно отнести: HTML, CSS, JavaScript, PHP, SQL. Все они используются при реализации плагина для управления гостиничными номерами.

HTML (*HyperText Markup Language* — “язык разметки гипертекста”) – самый известный язык веб-браузеров, с помощью которого создаются сайты. HTML-страницы, которые еще называют гипертекстовыми документами, интерпретируются браузером клиента в удобное для них представление, позволяющее просматривать информацию различного вида, такую как графика или текст[16]. Средства форматирования HTML помогают построить сложную иерархическую структуру, состоящую из HTML-страниц. HTML является теговым языком разметки. Все документы на этом языке представляют собой набор элементов, начало и конец каждого из которых обозначаются специальными пометками – тегами. Элементы могут быть самыми различными: пустыми, вложенными или иметь атрибуты, которые определяют какие-либо их свойства. Помимо элементов HTML-документы содержат также сущности, которые являются специальными символами и начинаются с амперсанта.

CSS (*Cascading Style Sheets*- “каскадные таблицы стилей”) – это язык описания внешнего вида документа, который написан с помощью языка гипертекстовой разметки, то есть HTML[17]. На сегодняшний день CSS используется практически на всех сайтах. Целью его создания являлось разделение описания логической структуры документа, осуществляемой посредством HTML, от описания внешнего вида документа, осуществляемого посредством CSS. Развитие компьютерных технологий и интернета послужило тому, что люди старались создавать более функциональные и яркие сайты, однако все параметры внешнего вида представлялись в html-документе посредством тегов. Чтобы решить данную проблему и был создан CSS.

JavaScript – язык программирования, созданный для «оживления и придания динамичности» веб-сайтам. Программы, написанные на языке JavaScript, называются *скриптами*, которые выполняются совместно с HTML-документами[18]. С помощью JavaScript программисты создают некоторые функции, как например открытие нового окошка с выводом в нем сообщения об ошибке после некоторых действий пользователя. Язык JavaScript способен выполнять свои скрипты спустя заданные интервалы времени. В общем, JavaScript это и самостоятельный язык, но также его можно назвать вспомогательным для остальных, так как с помощью него легко сделать сайт более функциональным и интересным для пользователя.

PHP (*HypertextPreprocessor*– “процессор гипертекста”) – язык разработки программного обеспечения для веб-приложений, исполняющихся на веб-сервере. Первоначально он создавался для обработки данных и подготовки HTML-страниц. Однако, со временем, его области применения значительно расширились.

Основная задача, выполняемая PHP – создание динамических веб-сайтов. То есть содержание страницы сайта может меняться в зависимости от заданных условий (динамические страницы).

Существует три основные области, где используется PHP[19]:

- Написание скриптов для приложений, которые выполняются на серверах. Язык PHP наиболее удобен для этих задач.
- Написание скриптов для приложения, которые прописываются в командной строке. Данные скрипты, будут выполняться на конкретном компьютере.
- Создание графических интерфейсов, выполняемых на компьютере клиента. Так называемые GUI-приложения.

SQL (Structured Query Language) — это специализированный непроведенный язык, позволяющий описывать данные, осуществлять выборку и обработку информации из реляционных СУБД, то есть он предназначен для взаимодействия веб-сайта с базой данных и хранящейся в ней информацией. Специализированность означает, что sql предназначен лишь для работы с БД; нельзя создать полноценную прикладную систему только средствами этого языка — для этого потребуются использовать другие языки, в которые можно встраивать sql-команды. Поэтому sql еще называют вспомогательным языковым средством для обработки данных.

SQL состоит из четырех отдельных частей[20]:

1. Язык определения данных (DDL) используется для определения структур данных, хранящихся в базе данных. Операторы DDL позволяют создавать, изменять и удалять отдельные объекты в БД.
2. Язык манипуляции данными (DML) используется для извлечения и изменения данных в БД. Операторы DML позволяют извлекать, вставлять, изменять и удалять данные в таблицах.
3. Язык определения доступа к данным (DCL) используется для контроля доступа к данным в БД. Операторы DCL применяются к привилегиям и позволяют выдавать и отбирать права на применение определенных операторов DDL и DML к определенным объектам БД.
4. Язык управления транзакциями (TCL) используется для контроля обработки транзакций в БД.

Для реализации плагина управления гостиничными номерами были определены задачи для каждого из вышеперечисленных языков программирования.

HTML и CSS были использованы для функционального разделения отдельных частей документа, а также для описания его внешнего вида. Язык программирования PHP использовался для обеспечения качественной разработки серверной части, а для качественной разработки клиентской части был использован язык программирования JavaScript. Чтобы обеспечить взаимодействие с данными, которые хранятся в базе данных был применен SQL.

2.2.2. Детальный анализ фреймворков

Фреймворк является программной оболочкой, которая позволяет упростить и ускорить решение типовых задач, характерных для конкретного языка программирования. В переводе с английского фреймворк означает «каркас».

В ходе сравнительного анализа, были выделены несколько наиболее востребованных фреймворков: Vue, Angular и React. Рассмотрим подробнее каждый из них.

Vue.js – это прогрессивный JavaScript фреймворк с открытым исходным кодом, предназначенный для разработки пользовательского интерфейса. Он является одним из самых популярных фреймворков для упрощения веб-разработки. Vue.js работает в основном с уровнем представления. Его с лёгкостью можно интегрировать в большие проекты для frontend-разработки.

Основные преимущества Vue.js:

- Библиотека довольно проста и функциональна;
- Требования к стеку отсутствуют, поэтому Vue.js можно использовать на любом проекте;
- Фреймворк совсем немного весит. Мы экономим время загрузки страниц и получаем целую кучу возможностей: плюс к конверсии, плюс к UX, плюс к поисковому ранжированию и т.д.;

- Довольно высокая скорость разработки;
- Возможность найти и подключить к проекту практически любого разработчика, кто хоть немного знаком с frontend-разработкой.

Vue позволяет создавать функциональные приложения, соответствующие всем современным стандартам, при минимальном подключении новых ресурсов.

Как и любой другой фреймворк, Vue.js тоже имеет свои недостатки:

- Риск чрезмерной гибкости. Иногда у Vue.js могут возникать проблемы при интеграции в огромные проекты;
- Отсутствие полной англоязычной документации.

Angular.js - open source-фреймворк, который позволяет создавать популярные сегодня одностраничные приложения. Основные преимущества:

- Декларативный стиль кода. При создании шаблонов в Angular.js применяется декларативная парадигма программирования. Это делает код более легковесным, облегчает его чтение и поддержку;
- Использование директив. В качестве языка шаблонов в Angular.js используется HTML. Директивы позволяют вам сконцентрироваться на проработке логики и работать более продуктивно;
- MVC из коробки. В AngularJS используется схема MVC, разделяющая логику, представление и данные приложения. Это позволяет создавать одностраничные веб-приложения (Single Page Application);
- Модульность. В Angular.js можно организовывать приложения из отдельных модулей. Такие модули могут как зависеть друг от друга, так и быть автономными.

Недостатки:

- Довольно сложный язык программирования;

- Интеграционные ошибки, которые могут возникать при переходе от старой версии к новой.

React.js – это своего рода модифицированная библиотека, «заточенная» под MVC (Model-View-Controller, где Модель отвечает за предоставление данных, Вид – отображает данные Модели пользователю, а Контроллер интерпретирует действия пользователя и заставляет Модель вносить изменения).

Преимущества:

- Легкое освоение;
- Высокий уровень гибкости и максимальной отзывчивости;
- Открытая библиотека данных JavaScript ежедневно совершенствуется благодаря открытому исходному коду;
- Легкий вес базы данных, процессы, происходящие со стороны пользователя, могут быть легко задействованы на сервере.

Недостатками React.js являются:

- Отсутствие упорядоченной документации;
- Каждый раз, много обновлений вносятся в его документы, а новые инструменты продолжают выпускаться, что делает его более сложным для разработчиков при работе с кодом;
- Всегда есть разница в архитектуре при новых обновлениях, что затрудняет работу с управлением.

Исследуя три различных фреймворка, для реализации плагина управления гостиничными номерами был выбран Vue.js, поскольку он является очень простым в освоении, гибким, высоко интегрируемым со сторонними технологиями фреймворком, а также Vue.js реализует все современные подходы к веб-разработке пользовательского интерфейса.

2.2.3. Системы управления базами данных, используемые в веб-разработке

Сейчас, при разработке серьезных веб-приложений, практически всегда используются базы данных, в которых хранится информация, необходимая для работы сайта – начиная от самого содержимого (контента), и заканчивая логинами, паролями пользователей и различными настройками сайта. А управление базами данных в программировании сайтов осуществляется при помощи клиент-серверных систем управления базами данных (СУБД).

СУБД – это комплекс языковых и программных средств, предназначенных для создания, ведения и совместного использования базы данных одним или несколькими пользователями[21].

В данной бакалаврской работе мы рассмотрим одни из наиболее востребованных СУБД на сегодняшний день, а именно: MySQL, PostgreSQL и MS SQL Server.

MySQL – это реляционная система управления базами данных с открытым исходным кодом. То есть данные в ее базах хранятся в виде логически связанных между собой таблиц, доступ к которым осуществляется с помощью языка запросов SQL. Эта серверная система способна эффективно функционировать во взаимодействии с интернет-сайтами и веб-приложениями. При этом она проста в освоении, что лишь увеличивает ее востребованность. Из преимуществ СУБД MySQL также стоит отметить обширный функционал, высокую безопасность, масштабируемость и высокую скорость выполнения запросов[22].

Работать с MySQL можно не только в текстовом режиме, но и в графическом. Существует очень популярный визуальный интерфейс (написанный на PHP) для работы с этой СУБД, который называется PhpMyAdmin. Этот интерфейс позволяет значительно упростить работу с базами данных в MySQL.

PostgreSQL - свободная объектно-реляционная система управления базами данных. PostgreSQL предоставляет множество различных

возможностей, достаточно надежна и имеет хорошие характеристики по производительности[23]. Кроме того, она свободно распространяется и имеет открытый исходный код. Одной из сильных сторон PostgreSQL является ее архитектура. Как и многие коммерческие СУБД, она может применяться в среде клиент-сервер, что дает массу преимуществ как пользователям, так и разработчикам. Основа PostgreSQL составляет серверный процесс базы данных. Он выполняется на одном сервере.

Доступ из приложений к данным базы осуществляется посредством процесса базы данных. Клиентские программы не могут получить доступ к данным самостоятельно, даже если они работают на том же компьютере, на котором выполняется серверный процесс. Такое разделение клиентов и сервера позволяет построить распределенную систему. Можно отделить клиентов от сервера посредством сети и разрабатывать клиентские приложения в среде, удобной для пользователя.

MS SQL Server - система управления реляционными базами данных, разработанная корпорацией Microsoft. Основными характеристиками в данной СУБД можно выделить[24]: простоту администрирования, возможность подключения к Web, быстроедействие и функциональные возможности механизма сервера СУБД, наличие средств удаленного доступа.

В целом данная СУБД представляет собой современную полнофункциональную базу данных, которая отлично подходит для малых и средних организаций. Необходимо заметить, что SQL Server уступает другим рассматриваемым СУБД по двум важным показателям: программируемость и средства работы. При разработке клиентских БД приложений на основе языков Java, HTML часто возникает проблема недостаточности программных средств SQL Server и пользоваться этой СУБД будет труднее, чем другими системами.

Проведя анализ наиболее востребованных СУБД, для реализации плагина управления гостиничными номерами была выбрана СУБД MySQL, поскольку она наилучшим образом реализует необходимые задачи.

2.2.4. Сравнительный анализ редакторов исходного кода

Работа программистов заключается в создании приложений, предназначенных для различных целей и удовлетворяющих самым различным запросам пользователей. Сегодня существуют средства, способные облегчить работу разработчиков, делающие написание программы и ее последующую отладку как можно более простыми и доступными. Эти приложения способны одинаково хорошо работать с кодами на самых разных языках программирования, обеспечив при этом удобный интерфейс, продуманность опций, быстроту работы и надёжность. Такие средства называются редакторами исходного кода. Наиболее востребованными на сегодняшний день являются: Sublime Text, Notepad++, PSPad.

Sublime Text — мультиплатформенный редактор (Linux, Mac OS, Windows). Из первичных возможностей нужно отметить базовую поддержку нескольких десятков языков программирования и разметки: HTML, CSS, Ruby, PHP, SQL, Javascript, C и других. Для поддерживаемых форматов предусмотрена подсветка синтаксиса и автодополнение. Одна из наиболее интересных функций Sublime Text — мультिवыделение, что позволяет редактировать несколько участков кода одновременно.

Работая в Sublime Text, написание кода также можно ускорить за счет сниппетов и макросов. Сниппеты — фрагменты часто используемого кода, которые вставляются посредством автодополнения[25]. Макросы применяются для воспроизведения последовательности действий. Из других особенностей интерфейса — боковая панель, которая ускоряет доступ к файлам и проектам. Основные неудобства в Sublime Text: невозможность локализации и отсутствие графического окна с настройками.

Notepad++ — свободный текстовый редактор с открытым исходным кодом. Имеет огромное количество возможностей, основные из которых автодополнение функций, параметров и слов, а также подсветка кода. Сюда же входят операции со строками, в том числе комментирование, преобразование текста. Характерная особенность Notepad++ — возможность применения

токенов (шаблонов) стилей кода[25]. Помимо стандартной нумерации строк, предполагается сворачивание кода (фолдинг).

С точки зрения удобства, Notepad++ является одним из наиболее привычных и понятных редакторов. Дополнительные возможности данного редактора: сохранение сессий, возможность смены режима интерфейса, поддержка огромного количества кодировок, поиск с использованием регулярных выражений, запись и воспроизведение макросов.

PSPad – редактор текста и кода для программистов. Основными возможностями данного редактора являются[25]: макросы, шаблоны, подсветка парных скобок и автодополнение, хотя последнее реализовано не полностью. Также в PSPad встроен CSS-редактор TopStyle Lite и известная библиотека Tidy, позволяющая оптимизировать код. Боковая панель PSPad — наиболее функциональная среди множества других редакторов. С ее помощью можно не только управлять проектами, но и настроить синхронизацию. Также в панели доступны элементы избранного, менеджер файлов и даже FTP-менеджер. Дополнительный плюс при работе с графической оболочкой – иконка в меню практически для каждой команды.

Анализируя одни из наиболее популярных редакторов исходного кода, для написания плагина управления гостиничными номерами был выбран редактор Sublime Text, поскольку он содержит в себе больше удобных возможностей, чем другие редакторы и не имеет ощутимых недостатков.

Вывод по главе 2

В процессе написания второй главы бакалаврской работы был исследован унифицированный язык UML как средство визуального моделирования при разработке плагина для управления гостиничными номерами. В процессе разработки были выделены такие сущности как: гость, администратор, номерной фонд. Для каждой из этих сущностей был определен набор действий после чего средствами Draw.io были построены три диаграммы вариантов использования.

Следующим этапом был проведен подробный обзор современных средств разработки плагина. Описаны все используемые в данной работе языки веб-программирования, проведен детальный анализ наиболее востребованных фреймворков и систем управления базами данных. Затем, последним этапом был проведен сравнительный анализ редакторов исходного кода.

ГЛАВА 3. РАЗРАБОТКА ПЛАГИНА ДЛЯ УПРАВЛЕНИЯ ГОСТИНИЧНЫМИ НОМЕРАМИ

3.1. Создание базы данных в СУБД MySQL

Физическое проектирование предполагает создание базы данных с использованием определенной СУБД. База данных сайта «Blu hotels» реализована посредством СУБД MySQL (см. пункт 2.2.1.).

Для создания новой базы данных необходимо перейти в PhpMyAdmin (Рис.3.1), дать название базе и нажать *Создать* (Рис.3.2)

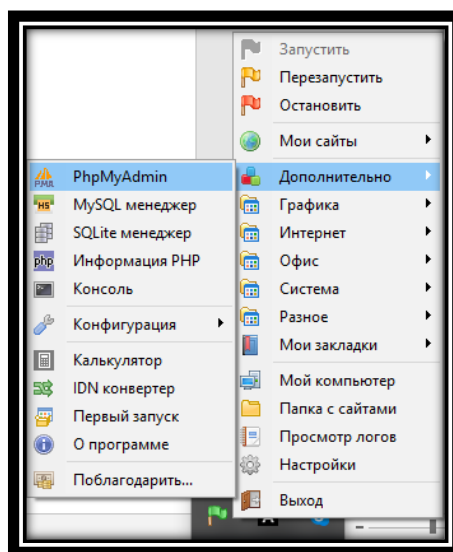


Рис.3.1 Переход в PhpMyAdmin

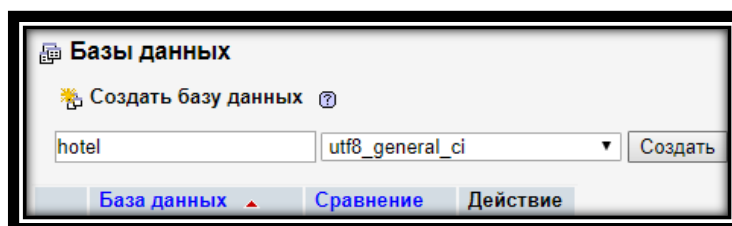


Рис.3.2 Создание БД «hotel»

Следующим этапом является создание необходимых таблиц в базе данных (Рис.3.3): booking, news, rooms, room_categories.

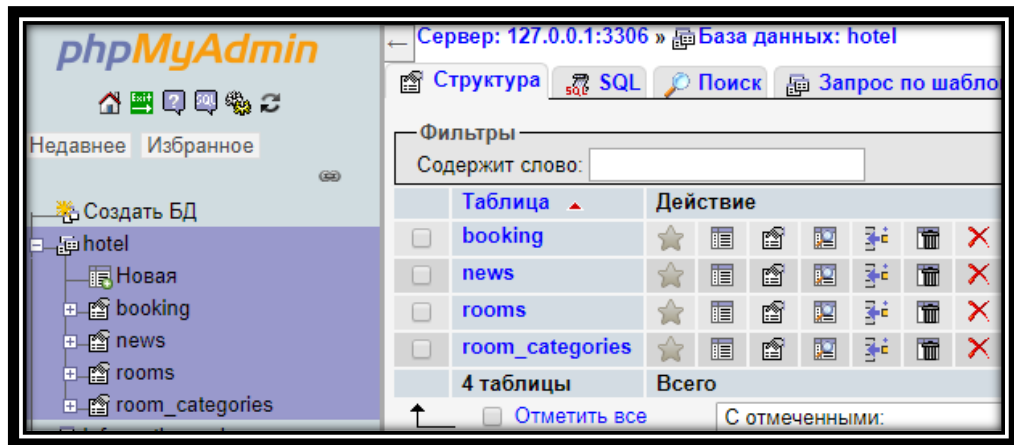


Рис.3.3 Созданные таблицы в базе данных «hotel»

Таблица *booking* создана для хранения информации о забронированных номерах для последующего отображения на календаре бронирования и содержит в себе 9 полей:

- *id_booking* – уникальное автоинкрементируемое поле;
- *date_from* – дата заезда клиента;
- *date_to* – дата отъезда клиента;
- *total_cost* – хранит общую стоимость номера за выбранное количество дней;
- *id_room* – номер выбранной категории;
- *room_name* – наименование выбранной категории номера;
- *user_name* – имя пользователя;
- *user_phone* – номер пользователя;
- *user_comment* – комментарий от пользователя (возможно личные пожелания).

Таблица *news* предназначена для хранения новостей на сайте гостиничного комплекса «Blu hotels» и содержит в себе четыре поля:

- *id_news* – уникальное автоинкрементируемое поле;
- *title* – предназначен для хранения заголовка новости;
- *description* – предназначен для хранения описания новости;
- *date* – хранит дату публикации номера.

Таблица *rooms* предназначена для создания номеров гостиницы и хранения информации о них. Содержит в себе 8 полей:

- *id_room* – уникальное автоинкрементируемое поле;
- *id_category* – номер выбранной категории;
- *name* – название номера;
- *short_description* – краткое описание номера;
- *description* – полное описание номера;
- *cost_per_day* – стоимость номера;
- *num_of_people* – максимальное количество человек в номере;
- *img_main* – изображение номера.

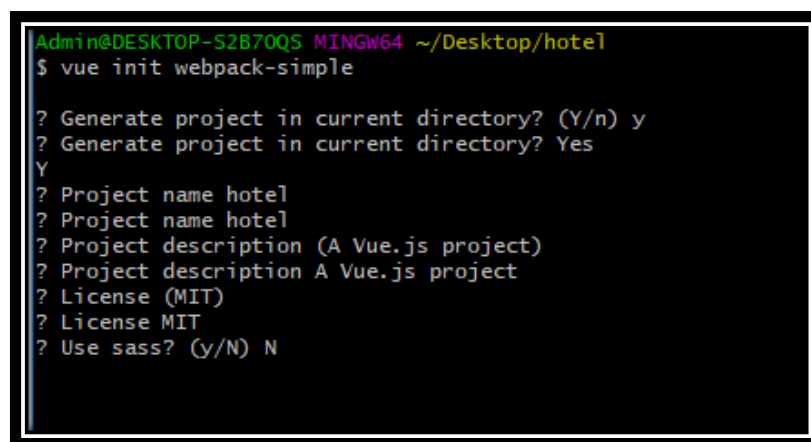
Таблица *room_categories* предназначена для создания категорий номеров и для хранения информации о них. Содержит 3 поля:

- *id_category* – уникальное автоинкрементируемое поле;
- *name* – название категории номера;
- *description* – описание категории.

3.2. Реализация компонентов веб-приложения

С помощью интерфейса командной строки (CLI) для фреймворка Vue – VUE CLI необходимо создать проект.

В консоли требуется ввести следующую команду: `vue init webpack-simple`. Далее будут пройдены несколько шагов установки (Рис.3.4).



```

Admin@DESKTOP-S2B70QS MINGW64 ~/Desktop/hotel
$ vue init webpack-simple

? Generate project in current directory? (Y/n) y
? Generate project in current directory? Yes
Y
? Project name hotel
? Project name hotel
? Project description (A Vue.js project)
? Project description A Vue.js project
? License (MIT)
? License MIT
? Use sass? (y/N) N
  
```

Рис.3.4 Процесс установки в командной строке

Для дальнейшей разработки необходимо установить зависимости. Для этого в консоли вводится следующая команда: `npm install`.

После успешной установки всех необходимых библиотек директория проекта выглядит следующим образом (Рис.3.5):

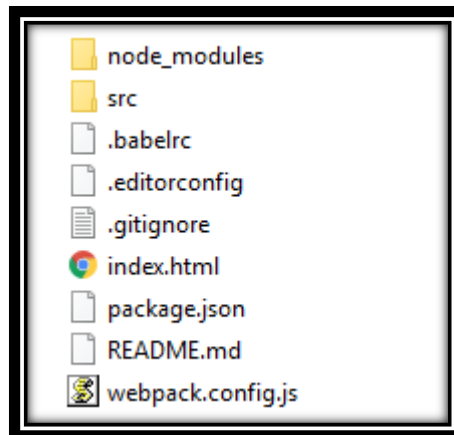


Рис.3.5 Директория проекта

Для запуска проекта в режиме разработки необходимо ввести следующую команду в командной строке: `npm run dev`. Это поможет запустить сервер с так называемой «горячей перезагрузкой», то есть при каждом изменении файлов приложение будет автоматически перезагружаться. Данный подход является общепринятой практикой и очень сильно облегчает разработку приложений.

Следующим этапом открывается стартовая страница (Рис.3.6), которая поставляется при развертывании проекта с помощью `vue cli`.

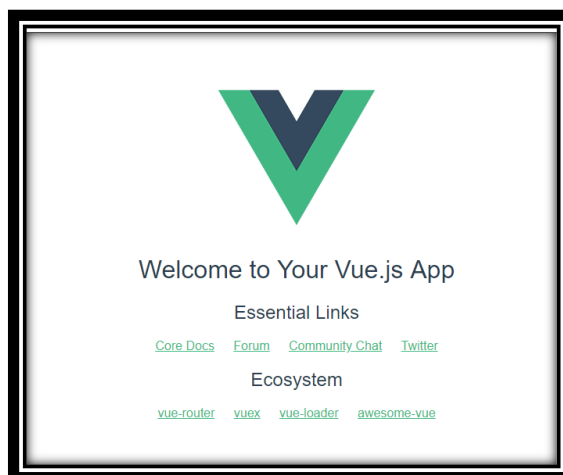


Рис.3.6 Стартовая страница Vue.js

Далее необходимо добавить библиотеку «vue-router» для осуществления перехода по страницам (Рис.3.7). В проект добавляем папку «views» и помещаем в нее однофайловые компоненты Vue, то есть файлы с расширением `.vue`.

```

import Vue from 'Vue'
import App from './App.vue'

import VueRouter from "Vue-router";
Vue.use(VueRouter);

// страницы
import Index from "./views/Index.vue";
import News from "./views/News.vue";
import NewsView from "./views/News-view.vue";
import Rooms from "./views/Rooms.vue";
import About from "./views/About.vue";
import Contacts from "./views/Contacts.vue";

import Categories from "./views/Categories.vue";
import Category from "./views/Category.vue";

import QuicklyBooking from "./views/Quickly-booking.vue";

import Admin from "./views/Admin.vue";

```

Рис.3.7 Добавление библиотеки vue-router

Следующим пунктом является добавление глобального объекта «router» и описания в нем маршрутов (переходов по страницам) (Рис.3.8).

```

let router = new VueRouter({
  routes: [
    { path: "/", component: Index },
    { path: "/news", component: News },
    { path: "/news-view/:id", name: "news", component: NewsView },
    { path: "/rooms", component: Rooms },
    { path: "/about", component: About },
    { path: "/contacts", component: Contacts },
    { path: "/categories", component: Categories },
    { path: "/category/:id", name: "category", component: Category },

    { path: "/quickly-booking", name: "quickly-booking", component: QuicklyBooking },

    { path: "/admin", component: Admin },

    { path: "/room-view/:id", name: "room", component: RoomView }
  ],
  /* scrollBehavior (to, from, savedPosition) {
    return { x: 0, y: 0 }
  } */
  scrollBehavior (to, from, savedPosition) {
    //https://router.vuejs.org/guide/advanced/scroll-behavior.html
    if (to.hash) {
      return { selector: to.hash }
    } else if (savedPosition) {
      return savedPosition;
    } else {
      return { x: 0, y: 0 }
    }
  }
})

```

Рис.3.8 Добавление глобального объекта «router»

После выполнения вышеперечисленных действий созданы пустые страницы. Далее для добавления главной навигации была создана папка *components*, а в ней новый компонент *header-app.vue* (Рис.3.9).

```

<template>
  <header>
    <nav v-scroll="handleScroll">
      <ul class="nav-main">
        <li><router-link to="/" class="nav-main--link">Главная</router-link></li>
        <li><router-link to="/about" class="nav-main--link">О комплексе</router-link></li>
        <li><router-link to="/categories" class="nav-main--link">Категории номеров</router-link></li>
        <li><router-link to="/rooms" class="nav-main--link">Номера</router-link></li>
        <li><router-link to="/news" class="nav-main--link">Новости</router-link></li>
        <li><router-link to="/contacts" class="nav-main--link">Контакты</router-link></li>
      </ul>
    </nav>

    <div class="logo-text">
      Гостиничный комплекс «Blu hotels»
    </div>

    <form class="header-form" v-if="this.$router.history.current.path === '/'">
      <label>
        <span>Дата заезда</span>
        <input type="date" v-model="booking.dateFrom" :min="minDateForSelection">
      </label>
      <label>
        <span>Дата выезда</span>
        <input type="date" v-model="booking.dateTo" :min="minDateForSelection">
      </label>
      <label>
        <span>Кол-во человек</span>
        <select v-model="booking.numOfPeople">
          <option value="1">1</option>
          <option value="2">2</option>
          <option value="3">3</option>
          <option value="4">4</option>
        </select>
      </label>
      <button @click="quicklyBooking">Подобрать</button>
    </form>

    <parallax class="parallax" :speed-factor="0.3">
      
    </parallax>

    <a href="#" class="scroll-link"><span class="scroll-link_icon"></span><span class="scroll-link_text">Крутите вниз</span></a>
  </header>
</template>

```

Рис.3.9 Разметка навигации

Чтобы визуально оформить меню и «шапку» сайта, необходимо в секции *style* описать стили (Рис.3.10).

```

168 nav {
169   position: fixed;
170   z-index: 12;
171   top: 0;
172   width: 100%;
173   padding: 20px 0;
174   background-color: rgba(19,47,97, 0);
175   transition: all .55s;
176 }
177
178 nav.scrolled {
179   padding: 14px 0;
180   background-color: rgba(19,47,97, 1);
181   box-shadow: 0 2px 15px 0 rgba(19,47,97, .5);
182 }
183
184 .nav-main {
185   display: flex;
186   justify-content: center;
187   list-style: none;
188 }
189
190 .scroll-link {
191   position: absolute;
192   right: 0;
193   bottom: 10px;
194   left: 0;
195   display: block;
196   width: 8.125rem;
197   margin: auto;
198   font-size: .75rem;
199   line-height: 2;
200   letter-spacing: .2em;
201   color: inherit;
202   text-decoration: none;
203   text-align: center;
204 }
205
206 #app > div:not(.index-page) .scroll-link {
207   display: none;
208 }

```

Рис.3.10 Описание стилей

После добавления стилей главная страница приложения выглядит следующим образом (Рис.3.11):

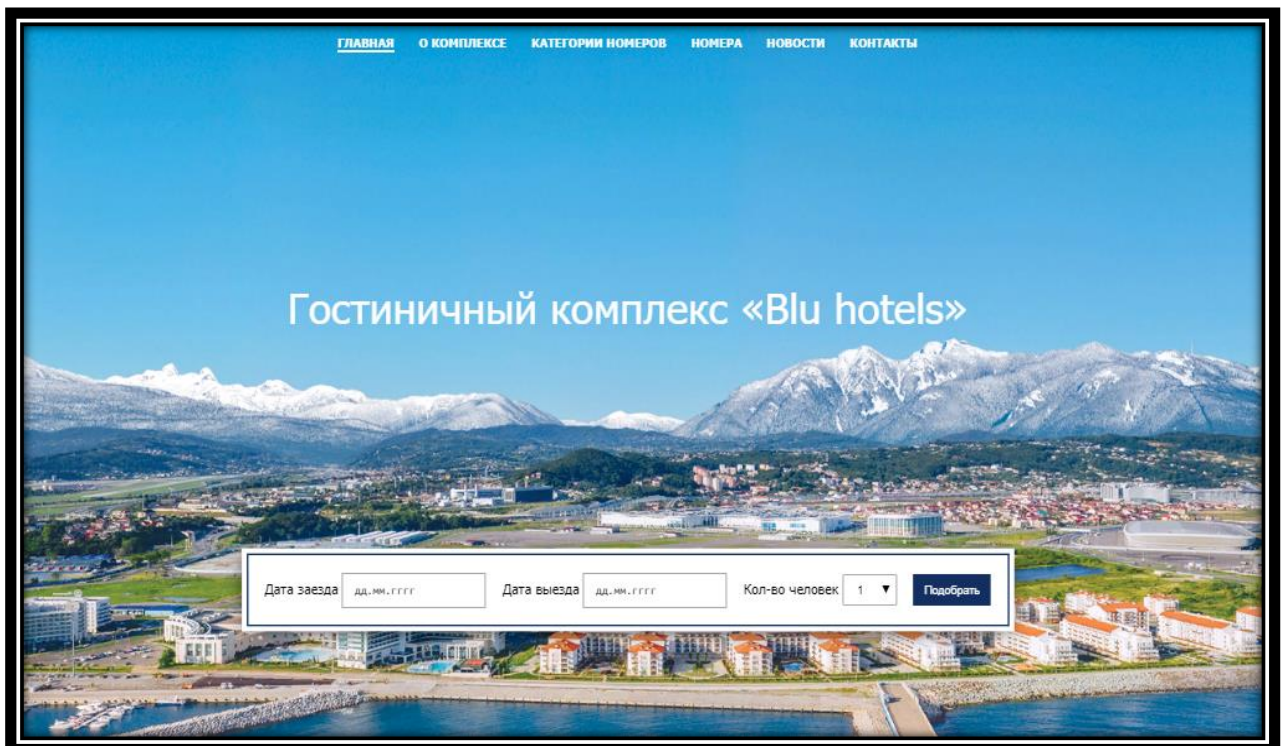


Рис.3.11 Главная страница

Далее, имеющуюся «шапку» необходимо подключить на каждую страницу, для этого в секции script подключается следующий файл: import

headerApp from "../components/header-app.vue". После чего нужно отобразить этот файл в нужном месте в разметке следующий образом: <headerApp/>.

Аналогичным образом добавляется компонент для «футера» сайта, который будет располагаться на всех страницах приложения. Футером называют секцию или блок, находящуюся внизу сайта, в данном случае в нем будет размещен копирайт.

Компонент футера представлен на рисунке 3.12.

```
<template>
  <footer>
    <mapApp/>
    <footerContacts/>
    <div class="copyright">
      Salie, Anastasia, CEPU ©2018
    </div>
  </footer>
</template>
<script>
import mapApp from "../map-app.vue";
import footerContacts from "../footer-contacts.vue";

export default {
  components: {
    mapApp,
    footerContacts
  },
  data() {
    return {
    }
  }
}
</script>

<style>
footer {
  position: relative;
}

.copyright {
  padding: 15px 0;
  background-color: #132F61;
  text-align: center;
  color: #fff;
}
</style>
```

Рис.3.12 Компонент футера

Далее была разработана функциональная часть отображения номеров и новостей на главной странице. На сайте, помимо отдельных страниц номеров и новостей предусмотрены также блоки на главной странице, в которых размещены несколько номеров (Рис.3.13) и последние новости (Рис.3.14).

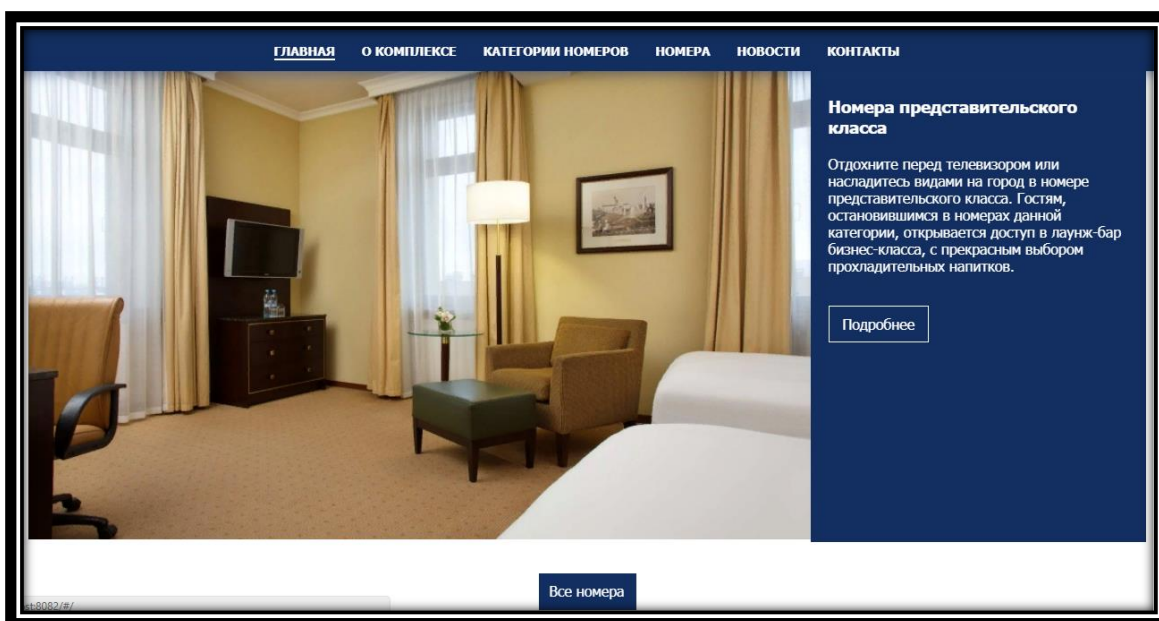


Рис.3.13 Блок номеров, расположенных на главной странице

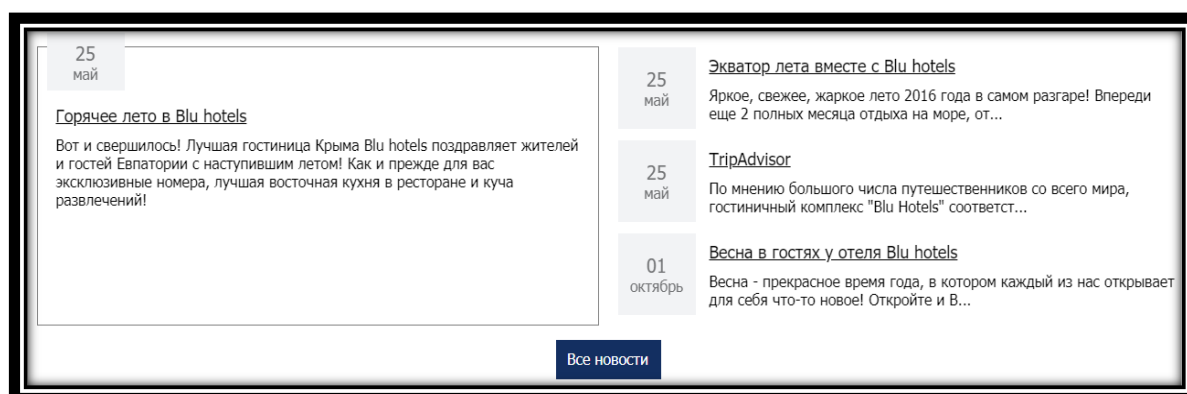


Рис.3.14 Блок новостей, расположенных на главной странице

3.3. Проектирование плагина для управления гостиничными номерами

При разработке плагина для управления гостиничными номерами в первую очередь учитывалось удобство пользователя. Первым этапом в ходе реализации плагина являлось создание категорий номеров: Номер «Стандартный», Номер «Люкс». Дополнительные категории можно добавить с помощью административной панели (см. ниже пункт 3.4). Чтобы увидеть имеющиеся виды номеров, необходимо на сайте перейти на вкладку категории (Рис.3.15) (Приложение В):

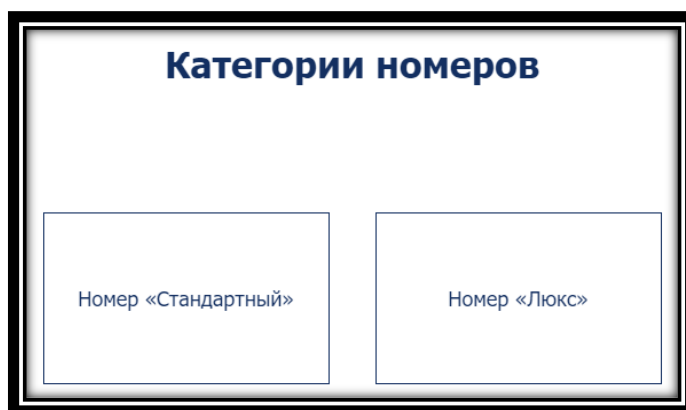


Рис.3.15 Категории номеров

Категории номеров в отелях и гостиницах необходимы для того, чтобы клиент в зависимости от его потребностей и финансов мог выбрать номер с необходимым уровнем комфортности. К примеру, если человек отправляется в командировку на пару дней, то ему вполне достаточно заказать стандартный тип номера или же если наоборот, планируется более длительная поездка с целью отдыха, то стоит обратить внимание на апартаменты с повышенным уровнем комфортности.

Далее, после того как гость выберет подходящую для себя категорию, ему откроются все номера, имеющиеся в ней (Рис.3.16) (Приложение Г) с кратким описанием, возможностью просмотра подробной информации о номере и, в случае необходимости, дальнейшего бронирования.

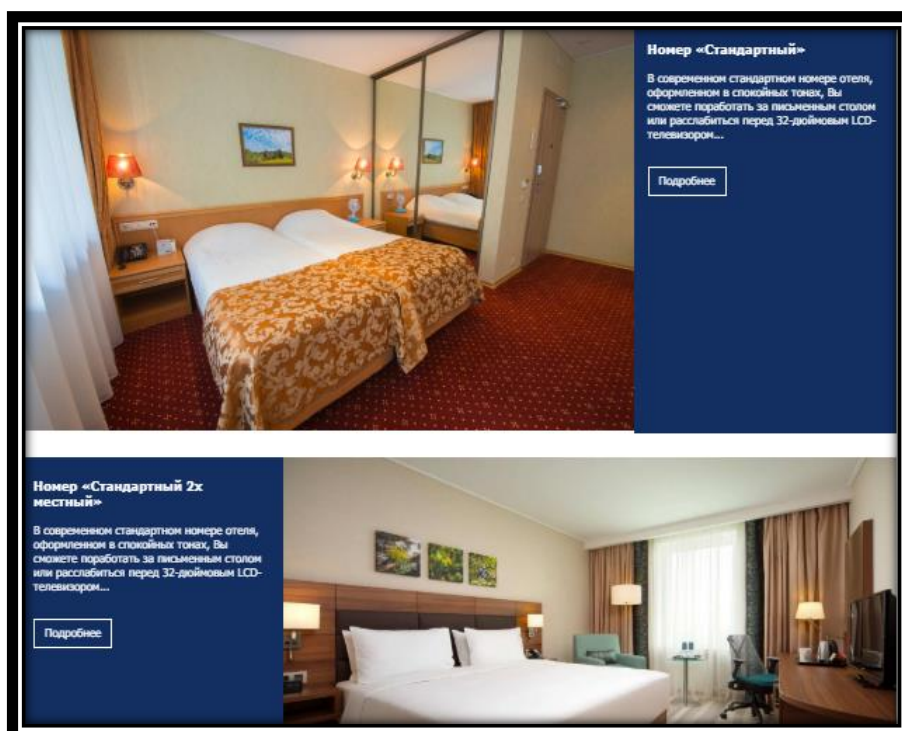
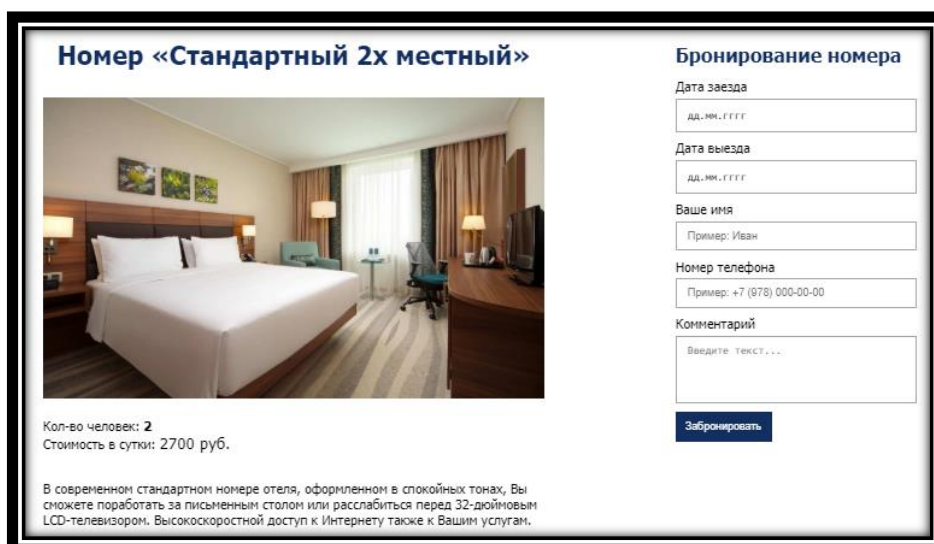


Рис.3.16 Фрагмент номеров, имеющихся в категории «Стандарт»

Если пользователь желает просмотреть все имеющиеся номера в гостинице он может перейти по вкладке *Номера*, где ему будут представлены номера с различным уровнем комфортности (Приложение Д). Полную информацию о них пользователь может посмотреть, нажав на кнопку *Подробнее* (Рис.3.17).



Номер «Стандартный 2х местный»

Кол-во человек: **2**
Стоимость в сутки: 2700 руб.

В современном стандартном номере отеля, оформленном в спокойных тонах, Вы сможете поработать за письменным столом или расслабиться перед 32-дюймовым LCD-телевизором. Высокоскоростной доступ к Интернету также к Вашим услугам.

Бронирование номера

Дата заезда
дд.мм.гггг

Дата выезда
дд.мм.гггг

Ваше имя
Пример: Иван

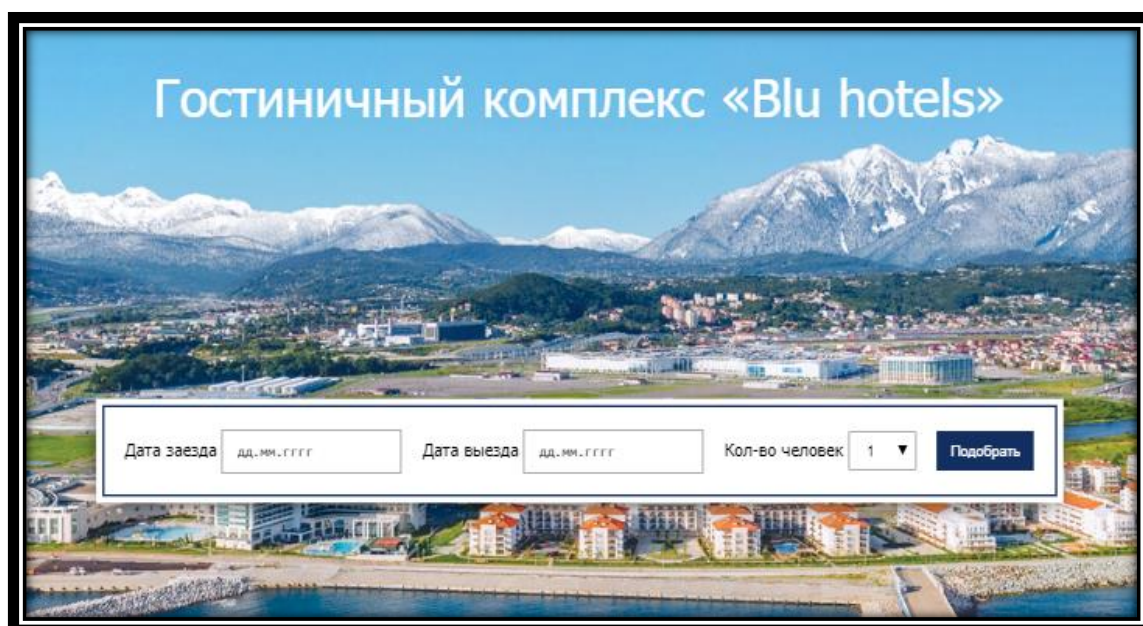
Номер телефона
Пример: +7 (878) 000-00-00

Комментарий
Введите текст...

Забронировать

Рис.3.17 Подробная информация о номере «Стандарт 2х местный»

Для удобства пользователей при разработке плагина для управления гостиничными номерами был реализован быстрый поиск свободных номеров по выбранным параметрам (Рис.3.18) (Приложение Е). То есть при вводе даты заезда, даты выезда и количества человек можно подобрать все имеющиеся свободные номера.



Гостиничный комплекс «Blu hotels»

Дата заезда дд.мм.гггг Дата выезда дд.мм.гггг Кол-во человек 1 ▼ **Подобрать**

Рис.3.18 Быстрый поиск на главной странице сайта

3.4. Проектирование административной панели при разработке плагина для управления гостиничными номерами

Для входа в административную панель плагина, который был спроектирован с целью внесения изменений в управления номерами и новостями, важным пунктом является создание формы для авторизации администратора (Рис.3.19, Рис.3.20). Это необходимо для контроля достоверности данных и обеспечения защиты от несанкционированного доступа. После авторизации администратора для него открываются дополнительные функции (Рис.3.21).

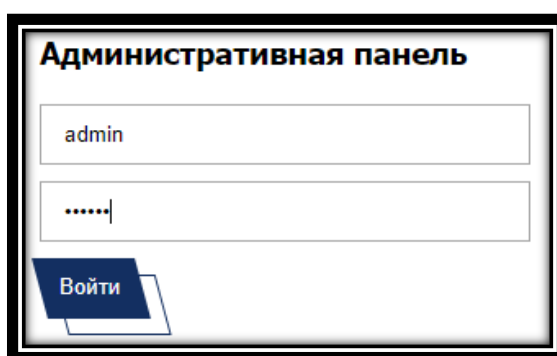
The image shows a web form for administrator login. It has a title 'Административная панель' at the top. Below the title are two input fields: the first contains the text 'admin', and the second contains masked characters '.....'. At the bottom left of the form is a blue button with the white text 'Войти'.

Рис.3.19 Форма авторизации администратора

```

<div v-if="!showAdminPanel" class="admin-form-wrap">
  <h3>Административная панель</h3>
  <form class="admin-form">
    <input type="text" v-model="admin.login">
    <input type="password" v-model="admin.password">
    <button type="button" class="btn" @click="login">Войти</button>
  </form>
</div>

data() {
  return {
    showDate: new Date(),

    newList: [],
    isShowChangeNewsForm: false,
    changedNews: null,
    errors: [],
    changedNewsIndex: null,

    showAdminPanel: false,
    admin: {
      login: "",
      password: ""
    },

    adminCorrectData: {
      login: "admin",
      password: "123456"
    },

    categories: [],
    newCategory: {
      name: "",
      description: ""
    }
  }
},

```

Рис.3.20 Фрагмент кода для формы авторизации администратора



Рис.3.21 Меню, доступное для администратора

Одной из основных возможностей администратора является добавление и удаление номеров (Приложение Ж). Для добавления номера администратору необходимо перейти по вкладке *Номера* и ввести необходимую информацию о нём (Рис.3.22):

- Название номера;
- Краткое описание;
- Описание;
- Стоимость в сутки (руб.);
- Количество человек в номере;
- Категория номера.

Последние два пункта представляют собой выпадающий список, где количество человек в номере не превышает четырех, а категории номера можно выбрать *Стандартный* или *Люкс*.

The image shows a web form on the left and its corresponding Vue.js code on the right. The form is titled "Номера" and "Добавить номер". It contains several input fields: "Название номера" (text), "Краткое описание" (text area), "Описание" (text area), "Стоимость в сутки (руб.)" (text), "Кол-во человек в номере" (dropdown menu with "1" selected), and "Категория номера" (dropdown menu). A "Добавить" button is at the bottom. The code on the right is a Vue.js template for a form, using v-model for data binding and v-for for the category dropdown options.

```
<h3>Добавить номер</h3>
<form>
  <label>
    <span>Название номера</span>
    <input type="text" v-model="room.name">
  </label>
  <label>
    <span>Краткое описание</span>
    <textarea v-model="room.shortDescription"></textarea>
  </label>
  <label>
    <span>Описание</span>
    <textarea v-model="room.description"></textarea>
  </label>
  <label>
    <span>Стоимость в сутки (руб.)</span>
    <input type="text" v-model="room.cost">
  </label>
  <label>
    <span>Кол-во человек в номере</span>
    <select v-model="room.numOfPeople">
      <option value="1">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4">4</option>
    </select>
  </label>
  <label>
    <span>Категория номера</span>
    <select v-model="room.category">
      <option value="0">-</option>
      <option v-for="(category, index) in categories" :key="index"
        :value="category.id_category">{{ category.name }}</option>
    </select>
  </label><br>
  <button type="button" class="btn" @click="addRoom">Добавить</button>
</form>
```

Рис.3.22 Форма для добавления номера и её код

Для того, чтобы удалить номер необходимо из списка номеров выбрать подходящий, нажать на кнопку *Удалить* (Рис.3.23) и подтвердить удаление во всплывающем окне.

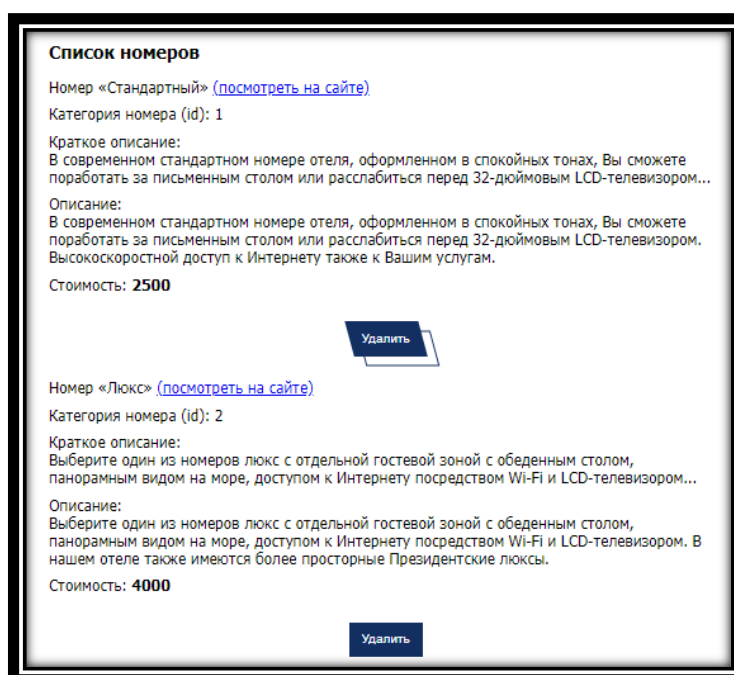


Рис.3.23 Удаление номера в административной панели

Еще одной полезной возможностью администратора является добавление категорий номеров (Приложение 3). Для этого ему нужно перейти во вкладку *Категории номеров* и заполнить необходимую информацию (Рис.3.24): название категории и её описание.

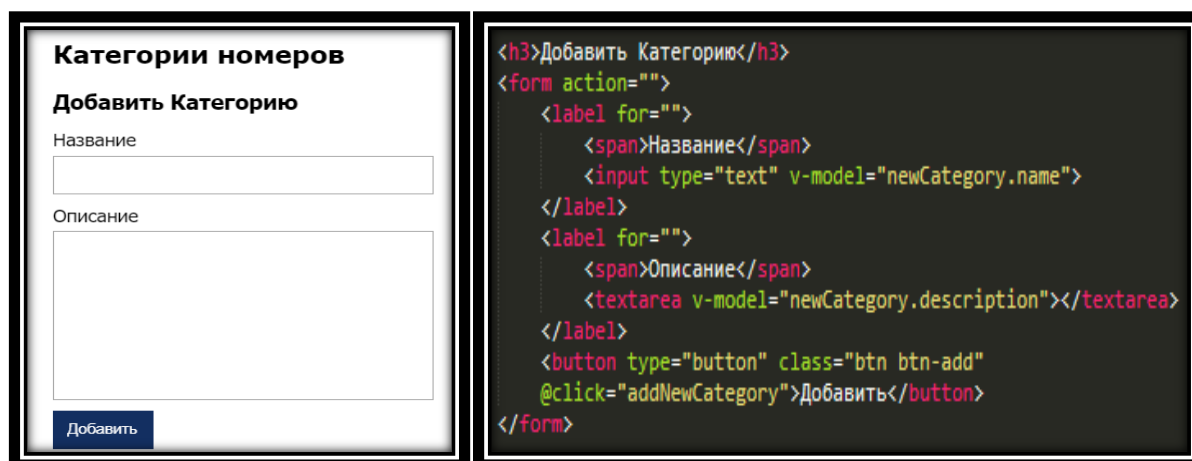


Рис.3.24 Форма добавления категорий номеров и её код

Также в административной панели предусмотрено добавление, изменение и удаление новостей, просматривание листа бронирования и общего календаря бронирования. Изучение листа бронирования также предназначено для управления номерами (Рис.3.25). То есть, изучив его данные, администратор может увидеть какие номера были забронированы, на какой промежуток времени, имя клиента и его номер телефона, комментарии клиента

(если таковые имеются), а также общую сумму на которую были забронированы номера за последние пол года.

Лист бронирования								
Заказ	Номер	Название	Дата заезда	Дата выезда	Стоимость	Имя клиента	Номер клиента	Комментарий
1	3	Номера представительского класса	2019-04-09	2019-04-12	18000	Иван	+79784568712	
5	3	Номера представительского класса	2019-04-15	2019-04-21	36000	Анастасия	+79786841232	
6	1	Номер «Стандартный»	2019-07-15	2019-07-20	12500	Николай	+79786547832	
7	2	Номер «Стандартный»	2019-04-09	2019-04-12	12000	Рефат	+79787531295	Необходим трансфер в аэропорт Симферополя
8	1	Номер «Стандартный»	2019-05-27	2019-05-31	10000	Константин	+79784562189	
9	1	Номер «Стандартный»	2019-05-20	2019-05-22	5000	Кристина	+79788643455	

Рис.3.25 Фрагмент листа бронирования

Вывод по главе 3

Итак, в процессе написания заключительной главы бакалаврской работы в первую очередь был описан процесс создания базы данных в MySQL и рассмотрены все таблицы, которые в ней имеются. После чего описывалась реализация всех использующихся в работе компонентов веб-приложения. Следующим этапом являлось описание процесса проектирования плагина для управления гостиничными номерами и отдельно проектирования административной панели для этого плагина.

ЗАКЛЮЧЕНИЕ

В ходе написания выпускной квалифицированной работы был подробно изучен основной теоретический материал в области программных модулей, который позволил выявить различные виды плагинов, а также определить основные задачи, которые выполняют различные расширения:

- упрощение разработки базовой версии программы;
- учет индивидуальных потребностей;
- расширение функционала программ.

Исходя из анализа предметной области можно сделать вывод, что услуги гостиничного бизнеса с каждым годом становятся более востребованными. Услугами гостиниц и отелей пользуются круглый год. Согласно прогнозам специалистов в гостиничной индустрии, развитие данной отрасли будет постоянно возрастать.

Основные характеристики программного модуля определены в ходе сравнительного анализа систем управления гостиничными номерами:

- VDA Micromaster;
- Hotel Solution;
- EcoStruxure™.

Для лучшего понимания того, как должен работать плагин для управления номерами в отелях, был также проведен сравнительный анализ этно-отеля «Jeval» и отеля «Donna Rosa», которые в дальнейшем и послужили для определения основных критериев разрабатываемого плагина.

В данной бакалаврской работе в качестве средства визуального моделирования рассматривался унифицированный язык UML. В процессе разработки были выделены такие сущности как: гость, администратор, номерной фонд, для каждой из которых был определен набор действий. Графическое представление модели при разработке плагина, было реализовано посредством веб-сервиса Draw.io, предназначенного для формирования различных диаграмм и схем, и содержащего набор элементов для UML диаграмм.

В ходе проведения подробного обзора современных средств разработки плагина сначала были описаны все используемые в данной работе языки веб-программирования. Затем, рассматривая и сравнивая между собой наиболее востребованные фреймворки, для разработки плагина управления гостиничными номерами был выбран фреймворк Vue.js, поскольку он является очень простым в освоении, гибким, высоко интегрируемым со сторонними технологиями фреймворком, а также реализует все современные подходы к веб-разработке пользовательского интерфейса.

Далее, в процессе сравнительного анализа различных систем управления базами данных, для реализации плагина была использована СУБД MySQL, потому как она наилучшим образом реализует поставленные задачи. После чего, анализ редакторов исходного кода показал, что Sublime Text, содержит в себе наибольшее количество удобных возможностей и не имеет ощутимых недостатков, поэтому он был выбран для реализации плагина управления номерами отеля.

В рамках реализации плагина была создана база данных *hotel* в MySQL, которая содержит в себе 4 таблицы для хранения различных данных. Далее был описан процесс реализации всех использующихся в работе компонентов веб-приложения.

Итогом написания выпускной квалифицированной работы являлось описание процесса проектирования плагина для управления гостиничными номерами и отдельно проектирования административной панели для него.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое плагин, его особенности [Электронный ресурс]. – Режим доступа: <http://www.novichkoff.ru/cto-takoe-plugin-ego-osobennosti.html> (дата обращения 19.02.2019)
2. Что такое плагин в компьютере и зачем он нужен [Электронный ресурс]. – Режим доступа: <https://lifeo.ru/cto-takoe-plugin-v-kompyutere-i-zachem-on-nuzhen/> (дата обращения 19.02.2019)
3. Плагин: что это такое, какие есть решения для SEO [Электронный ресурс]. – Режим доступа: <https://semantica.in/blog/cto-takoe-plugin.html> (дата обращения 19.02.2019)
4. Классификация и разновидность гостиниц [Электронный ресурс]. – Режим доступа: <http://mir-restoratora.ru/?p=8393> (дата обращения 11.03.2019)
5. Звездная система гостиниц | ГК МЕДИНАР [Электронный ресурс]. – Режим доступа: <https://medinardom.ru/informaciya/press-centr/stati/zvyozdnaya-sistema-gostinic/> (дата обращения 11.03.2019)
6. Система управления гостиничным номером VDA Micromaster. Room Management System [Электронный ресурс]. – Режим доступа: <http://pwv.ru/ru/systems/automatization/micromaster/> (дата обращения 18.04.2019)
7. Hotel Solution products in detail – overview & topologies [Электронный ресурс]. – Режим доступа: <http://ais.by/sites/ais.by/files/5.pdf> (дата обращения 18.04.2019)
8. Система управления номерами | Schneider Electric [Электронный ресурс]. – Режим доступа: <https://www.se.com/ru/ru/work/solutions/for-business/hotels/guest-room-management.jsp> (дата обращения 18.04.2019)
9. Гостиница в Евпатории – этно-отель Джеваль [Электронный ресурс]. – Режим доступа: <http://hotel-jeval.com> (дата обращения 04.05.2019)
10. Отель в Евпатории Донна Роза 3 звезды в Крыму [Электронный ресурс]. – Режим доступа: <http://dona-rosa.ru> (дата обращения 04.05.2019)

11. Шорыгин, С.М. Визуальное моделирование в информационных технологиях / С.М.Шорыгин. – Перспективы науки и образования, 2014, №6(12). – 19 с.
12. Tsvetkov, V.Ya. Information field. Life Science Journal / V.Ya.Tsvetkov. – 2014, 11(5). – 551-554 pp.
13. Буч, Г., Рамбо, Д., Якобсон, А. Язык UML. Руководство пользователя. Второе издание / Г.Буч, Д.Рамбо, А.Якобсон — ДМК, 2006. – 496 с.
14. Конушин А.В., Мазанова В.И. Язык визуального моделирования UML: методические указания к курсовой работе / А.В.Конушин, И.В.Мазанова. – 2012. – 6 с.
15. Панзеев. Е.А., Квитков А.А., Кулай С.В. Внедрение дисциплин по информационным технологиям / Е.А.Панзеев, А.А.Квитков, С.В.Кулай. – Экономика и социум, №1(32), 2017. – 4 с.
16. Описание языка HTML [Электронный ресурс]. – Режим доступа: <https://helpiks.org/7-8902.html> (дата обращения 17.05.2019)
17. Каскадные таблицы стилей CSS [Электронный ресурс]. – Режим доступа: <https://allrefrs.ru/2-36177.html> (дата обращения 17.05.2019)
18. Разновидности языков веб-программирования [Электронный ресурс]. – Режим доступа: <http://haknotdie.ru/viewtopic.php?id=2358> (дата обращения 17.05.2019)
19. Область применения языка программирования PHP [Электронный ресурс]. – Режим доступа: <http://gazetax.ru/statyi/pochemu-php.php> (дата обращения 17.05.2019)
20. SQL – Энциклопедия языков программирования [Электронный ресурс]. – Режим доступа: <http://progopedia.ru/language/sql/> (дата обращения 17.05.2019)
21. Базы данных и СУБД [Электронный ресурс]. – Режим доступа: <http://bd-subd.ru/lekci/bazi-dannih-i-subd.htm> (дата обращения 18.05.2019)

22. MySQL: особенности и сферы применения [Электронный ресурс]. – Режим доступа: <https://www.bytemag.ru/articles/detail.php?ID=6547> (дата обращения 18.05.2019)
23. Отличия, достоинства и недостатки базы данных PostgreSQL [Электронный ресурс]. – Режим доступа: <https://oracle-patches.com/common/3214-что-такое-postgresql> (дата обращения 18.05.2019)
24. СУБД SQL-Server основные особенности и ее применение [Электронный ресурс]. – Режим доступа: <https://www.kazedu.kz/referat/133608> (дата обращения 18.05.2019)
25. Путевод по редакторам исходного кода [Электронный ресурс]. – Режим доступа: <https://www.ixbt.com/soft/code-editors.shtml> (дата обращения 18.05.2019)

ПРИЛОЖЕНИЕ А

С 16 05 2019

По 18 05 2019

Взрослых 4

Детей 0

НАЙТИ



Апартаменты "Бадем"

Превосходный номер - Люкс с личным патио, выполненный в традиционном восточном стиле, имеет прекрасные панорамные виды и отличное внутреннее убранство. Мебель с резными вставками, изящные люстры и бра, традиционная восточная посуда для чая и кофе, просторная душевая комната.

- 2-х спальная кровать, или 2 односпальных, на выбор
- 2 отдельные ванные комнаты
- Раскладное кресло-кровать
- Возможность установки детской кроватки
- LCD- телевизор
- Посуда
- Сейф
- Электрочайник
- Кондиционер
- Мини-бар
- Wi-Fi

ПОДРОБНЕЕ



Тематический номер - Люкс "Нар"

Номер с прекрасным панорамным видом. Выполнен в традиционном восточном стиле и оснащен всем необходимым. В номере великолепная мебель с резьбой ручной работы, тканые ковры и салфетки, изящные бра, подушки с традиционной вышивкой. Для удобства гостей:

- 2-х спальная кровать, или 2 односпальных, на выбор
- 2 отдельные ванные комнаты
- Раскладное кресло-кровать
- Возможность установки детской кроватки
- LCD- телевизор
- Посуда
- Сейф
- Электрочайник
- Кондиционер
- Мини-бар
- Wi-Fi

ПОДРОБНЕЕ

Рис.1 Результат поиска номеров в отеле «Jeval» по введенным данным

16 мая — 17 мая

1 номер, 3 гостя

ИЗМЕНИТЬ ДАТЫ

У меня есть промокод

RUB

ПРОСМОТР ПО ТАРИФАМ

ПРОСМОТР ПО НОМЕРАМ

Двухместный Комфорт Улучшенный

подробнее

19 кв. м

В номере категории "Двухместный Комфорт Улучшенный" может удобно разместиться семья с двумя детьми разного возраста. Площадь номера: 17-22 кв.м. В номере:

Тариф с завтраком

подробнее

Стоимость за 1 ночь

1+1 3 800 Р

ЗАБРОНИРОВАТЬ

Завтрак **ВКЛЮЧЕН**

Условия отмены

Оплата: **ПРИ ЗАСЕЛЕНИИ**, банковской картой, по безналичному расчету, через терминалы

Полный пансион

подробнее

Стоимость за 1 ночь

1+1 5 300 Р

ЗАБРОНИРОВАТЬ

Полный пансион **ВКЛЮЧЕН**

Условия отмены

Оплата: **ПРИ ЗАСЕЛЕНИИ**, банковской картой, по безналичному расчету, через терминалы

Двухместный Комфорт Улучшенный с раздельными кроватями (твин)

подробнее

19 кв. м

Номер категории "Двухместный Комфорт Улучшенный с раздельными кроватями" - удобный, комфортабельный номер с двумя отдельными кроватями. Классический

Тариф с завтраком

подробнее

Стоимость за 1 ночь

1+1 3 800 Р

ЗАБРОНИРОВАТЬ

Завтрак **ВКЛЮЧЕН**

Условия отмены

Оплата: **ПРИ ЗАСЕЛЕНИИ**, банковской картой, по безналичному расчету, через терминалы

Полный пансион

подробнее

Стоимость за 1 ночь

1+1 5 300 Р

ЗАБРОНИРОВАТЬ

Полный пансион **ВКЛЮЧЕН**

Условия отмены

Оплата: **ПРИ ЗАСЕЛЕНИИ**, банковской картой, по безналичному расчету, через терминалы

Рис.2 Результат поиска номеров в отеле «Donna Rosa» по введенным данным

Рис.3 Окончательная форма бронирования номера в отеле «Donna Rosa»

ПРИЛОЖЕНИЕ В

```

<script>

import { instance } from "../axios_config.js";

import headerApp from "../components/header-app.vue";
import footerApp from "../components/footer-app.vue";

export default {
  components: {
    headerApp,
    footerApp
  },
  data() {
    return {
      categoryList: [],
      errors: []
    }
  },
  created() {
    instance.post("/", {
      requestType: "getAllCategories"
    })
    .then(response => {
      this.categoryList = response.data;

      console.log(this.categoryList);
    })
    .catch(e => {
      this.errors.push(e)
    });
  },
}

</script>

```

```

if($data->requestType === "getAllCategories") {
  $i = 0;
  $category = array();
  $category_query = mysqli_query($link, "SELECT * FROM `room_categories`");

  if($category_query && mysqli_num_rows($category_query) > 0) {
    $category_item = mysqli_fetch_array($category_query);

    do {
      $category[$i] = $category_item;
      $i++;
    }
    while($category_item = mysqli_fetch_array($category_query));

    print_r(json_encode($category));
  }
}

```

Рис.4 Фрагмент кода для вывода всех категорий

ПРИЛОЖЕНИЕ Г

```

<script>
import { instance } from "../axios_config.js";
import headerApp from "../components/header-app.vue";
import footerApp from "../components/footer-app.vue";

export default {
  components: {
    headerApp,
    footerApp
  },
  data() {
    return {
      rooms: [],
      errors: []
    }
  },
  created() {
    let categoryId = this.$route.params.id;

    console.log(categoryId)

    instance.post("/", {
      requestType: "getAllRoomsFromCategory",
      requestData: categoryId
    })
    .then(response => {
      // console.log( "-" + response.data + "-" )

      if(response.data && typeof response.data !== "string") {
        this.rooms = response.data;
        console.log( 123 );
      }

      console.log( 321 );
      console.log( this.rooms );
    })
    .catch(e => {
      this.errors.push(e)
    });
  }
}
</script>

```

```

if($data->requestType === "getAllRoomsFromCategory") {
  $category_id = $data->requestData;

  $i = 0;
  $room = array();
  $room_query = mysqli_query($link, "SELECT * FROM `rooms` WHERE `id_category` = '$category_id'");

  if($room_query && mysqli_num_rows($room_query) > 0) {
    $room_item = mysqli_fetch_array($room_query);

    do {
      $room[$i] = $room_item;
      $i++;
    } while($room_item = mysqli_fetch_array($room_query));

    print_r(json_encode($room));
  }
}

```

Рис.5 Фрагмент кода для вывода всех номеров, относящихся к определенной категории

ПРИЛОЖЕНИЕ Д

```

<script>

import { instance } from "../../axios_config.js";

import headerApp from "../../components/header-app.vue";
import footerApp from "../../components/footer-app.vue";

export default {
  components: {
    headerApp,
    footerApp
  },

  data() {
    return {
      rooms: [],
      errors: []
    }
  },

  created() {
    instance.post("/", {
      requestType: "getAllRooms"
    })
    .then(response => {
      if(response.data) {

        console.log( response.data )

        this.rooms = response.data;
      }
    })
    .catch(e => {
      this.errors.push(e)
    });
  }
}

</script>

```

```

if($data->requestType === "getAllRooms") {
  $i = 0;
  $room = array();
  $room_query = mysqli_query($link, "SELECT * FROM `rooms`");

  if($room_query && mysqli_num_rows($room_query) > 0) {
    $room_item = mysqli_fetch_array($room_query);

    do {
      $room[$i] = $room_item;
      $i++;
    }
    while($room_item = mysqli_fetch_array($room_query));

    print_r(json_encode($room));
  }
}

```

Рис.6 Фрагмент кода для вывода всех номеров

ПРИЛОЖЕНИЕ Е

```

1 <template>
2   <div class="">
3
4     <headerApp/>
5     <div class="container">
6       <h1>Свободные номера:</h1>
7
8       <div class="">Дата заезда: <b>{{ bookingData.dateFrom }}</b></div>
9       <div class="">Дата отъезда: <b>{{ bookingData.dateTo }}</b></div>
10      <div class="">Кол-во человек: <b>{{ bookingData.numOfPeople }}</b></div><br><br><br>
11
12      <!-- {{ getFreeRooms }} -->
13
14      <div class="rooms-wrap">
15        <div class="room-item" v-for="(room, index) in getFreeRooms" v-bind:key="index">
16          <div class="room-item--img-wrap">
17            
18          </div>
19          <div class="room-item--info">
20            <h3 class="room-item--title">{{ room.name }}</h3>
21            <p>{{ room.short_description }}</p>
22
23            <router-link :to="{name: 'room', params: {id: room.id_room, dateFrom: bookingData.dateFrom, dateTo: bookingData.dateTo }}" class="view-more">
24              Подробнее
25            </router-link>
26          </div>
27        </div>
28        <div v-show="getFreeRooms === []">
29          Номера не найдены!
30        </div>
31      </div>
32    </div>
33
34    <footerApp/>
35  </div>
36 </template>
37
38
39
40 <script>
41
42   import { instance } from "../axios_config.js";
43
44
45   import headerApp from "../components/header-app.vue";
46   import footerApp from "../components/footer-app.vue";
47
48
49   export default {
50     components: {
51       headerApp,
52       footerApp
53     },
54     data() {
55       return {
56         bookingData: {
57           dateFrom: "2019-05-19",
58           dateTo: "2019-05-25",
59           numOfPeople: ""
60         },
61
62         freeRooms: {
63
64         },
65         listOfBooking: [],
66
67         rooms: [],
68         errors: []
69       }
70     },
71   },
72
73   created() {
74
75     if(this.$route.params.booking !== undefined) {
76       this.bookingData = this.$route.params.booking;
77     }
78
79
80     instance.post("/", {
81       requestType: "getAllBookingList"
82     })
83     .then(response => {
84       if(response.data) {
85
86         instance.post("/", {
87           requestType: "getAllRooms"
88         })

```

```

89 .then(response => {
90     if(response.data) {
91         this.rooms = response.data;
92         console.log( this.rooms )
93     }
94 })
95 .catch(e => {
96     this.errors.push(e)
97 });
98
99
100 this.listOfBooking = response.data;
101
102 let userDateFrom = new Date(this.bookingData.datefrom);
103 let userDateTo = new Date(this.bookingData.dateto);
104
105 console.log(123)
106
107
108
109 this.listOfBooking.forEach(bookingItem => {
110
111     let bookingDateFrom = new Date(bookingItem.date_from);
112     let bookingDateTo = new Date(bookingItem.date_to);
113
114     let isRoomFree = true;
115
116
117     if((bookingDateFrom < userDateFrom && bookingDateTo > userDateFrom)) {
118         // console.log("случай 1");
119         this.freeRooms[bookingItem.id_room] = false;
120         return false;
121     }
122
123     if((bookingDateFrom < userDateTo && bookingDateTo > userDateTo)) {
124         // console.log("случай 2");
125         this.freeRooms[bookingItem.id_room] = false;
126         return false;
127     }
128
129     if((userDateFrom < bookingDateFrom && userDateTo > bookingDateTo)) {
130         // console.log("случай 3"); // брони номера между датой брони пользователя
131         this.freeRooms[bookingItem.id_room] = false;
132         return false;
133     }
134
135     if((userDateFrom.toLocaleDateString() === bookingDateFrom.toLocaleDateString() && userDateTo.toLocaleDateString() === bookingDateTo.
136         toLocaleDateString())) {
137         // console.log("случай 4"); // одинаковые даты
138         this.freeRooms[bookingItem.id_room] = false;
139         return false;
140     }
141
142     if(userDateFrom.toLocaleDateString() === bookingDateFrom.toLocaleDateString()) {
143         // console.log("случай 5"); // начало заезда одинаковое
144         this.freeRooms[bookingItem.id_room] = false;
145         return false;
146     }
147
148     if(this.freeRooms[bookingItem.id_room] !== false) {
149         this.freeRooms[bookingItem.id_room] = true;
150         return true;
151     }
152
153     });
154
155 })
156 .catch(e => {
157     this.errors.push(e)
158 });
159
160 },
161
162 computed: {
163     getFreeRooms() {
164         let rooms = [];
165
166         if( Object.keys(this.freeRooms) ) {
167             console.log( this.rooms.length );
168
169             this.rooms.forEach(room => {
170
171                 // console.log( room )
172
173                 if( ( this.freeRooms[room.id_room] === undefined || this.freeRooms[room.id_room] ) && room.num_of_people >= this.bookingData.numOfPeople ) {
174                     rooms.push(room);
175                 }
176             });
177         }
178
179         return rooms;
180     }
181 }
182
183 }
184
185 </script>

```

Рис.7 Листинг кода из файла Quick_search.vue для быстрого поиска номеров

ПРИЛОЖЕНИЕ Ж

```

methods: {
  removeRoom(roomIndex) {
    let res = confirm("Вы уверены?");

    if(res) {
      let room = this.rooms[roomIndex];

      instance.post("/", {
        requestType: "removeRoom",
        requestData: room
      })
      .then(response => {
        console.log(response.data)
      })
      .catch(e => {
        this.errors.push(e)
      });

      this.rooms.splice(roomIndex, 1);
    }
  },
  addRoom() {
    console.log(this.room.category);

    if(!this.room.category || this.room.category === "" || this.room.category === 0 || this.room.category === "-") {
      alert("Необходимо выбрать категорию номера!");
      return;
    }

    instance.post("/", {
      requestType: "addRoom",
      requestData: this.room
    })
    .then(response => {
      if(response.data) {
        alert("Номер успешно добавлен");
      }
    })
    .catch(e => {
      this.errors.push(e)
    });
  }
},

```

```

if($data->requestType === "addRoom") {
  $room = $data->requestData;

  $add = mysqli_query($link, "INSERT INTO `rooms` (`name`, `short_description`, `description`, `num_of_people`, `cost_per_day`, `id_category`)
  VALUES ('$room->name', '$room->shortDescription', '$room->description', '$room->numOfPeople', '$room->cost', '$room->category')");

  if($add) {
    print_r(json_encode($room));
  }
}

if($data->requestType === "removeRoom") {
  $room = $data->requestData;

  $remove = mysqli_query($link, "DELETE FROM `rooms` WHERE `id_room`='{$room->id_room}'");

  print_r(json_encode($room));
}

```

Рис.8 Фрагмент кода для добавления и удаления номеров

ПРИЛОЖЕНИЕ 3

```
addNewCategory() {  
    if(this.newCategory.name === "" || this.newCategory.description === "") {  
        alert("Заполните все поля!");  
        return;  
    }  
  
    instance.post("/", {  
        requestType: "addCategory",  
        requestData: this.newCategory  
    })  
  
    .then(response => {  
        if(response.data) {  
            alert("Категория успешно добавлена");  
        }  
    })  
  
    .catch(e => {  
        this.errors.push(e)  
    });  
},
```

```
if($data->requestType === "addCategory") {  
    $category = $data->requestData;  
  
    $add = mysqli_query($link, "INSERT INTO `room_categories` (`name`, `description`)  
        VALUES ('$category->name', '$category->description')");  
  
    if($add) {  
        print_r(json_encode($category));  
    }  
}
```

Рис.9 Фрагмент кода для добавления категории номеров