

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

**АВТОМАТИЗАЦИЯ ФУНКЦИОНИРОВАНИЯ ГОСТИНИЧНОГО
КОМПЛЕКСА**

Выпускная квалификационная работа

обучающегося по направлению подготовки 02.03.01 Математика и
компьютерные науки
очной формы обучения, группы 07001403
Слюта Наталии Сергеевны

Научный руководитель
Бурданова Е.В.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. АНАЛИЗ СОВРЕМЕННОГО СОСТОЯНИЯ ОБЪЕКТА ИССЛЕДОВАНИЯ	6
1.1 Обобщенная характеристика предметной области.....	6
1.1.1 Характеристика объекта исследования.....	6
1.1.2 Описание функции управления гостиничного комплекса	11
1.2 Актуальность разработки ИС	14
1.3 Постановка задачи на разработку ИС.....	16
2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ ГОСТИНИЧНОГО КОМПЛЕКСА	18
2.1 Проектирование базы данных.....	18
2.2 Создание и заполнение базы данных	20
2.3 Используемые классификаторы и система кодирования.....	32
2.4 Требования программного обеспечения.....	37
3. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ВЕБ - ПРИЛОЖЕНИЯ	48
3.1 Навигационная карта сайта.....	48
3.2 Описание интерфейсов реализации гостиничного комплекса.....	49
3.1 Тестирование гостиничного комплекса.....	51
ЗАКЛЮЧЕНИЕ	58
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	59
Приложение 1. Подключение базы данных	61
Приложение 2. Авторизация (фрагмент)	66
Приложение 3. Статус заявки и форма регистрации клиента (фрагмент)	68

ВВЕДЕНИЕ

В настоящее время автоматизация затрагивает все более широкие области. Новейшие компьютерные программы помогают выполнять большую часть работы, эффективно управляя основными и вспомогательными производственными процессами. Особенно актуально применять системы автоматизации на торговых предприятиях, причем одинаково полезными они могут стать как для больших, так и для малых форм торгового бизнеса. Если для крупной торговой компании необходимость автоматизации бизнеса уже рассматривается как единственно возможная и правильная стратегия успеха, то для небольшой компании внедрение новых технологий управления — способ информационной стабилизации, повышения производительности и усиления позиций предприятия на рынке.

Благодаря развитию информационных технологий изменились и способы ведения бизнеса.

За последние несколько лет объем и оборот информационных данных в различных сферах деятельности человека — экономической, политической, духовной, экономической, значительно возрос. Сам же процесс обработки, использования и накопления информации значительно ускоряется с каждым годом. Многие ученые говорят, что через десять лет объем используемой информации увеличится вдвое. Именно поэтому возникает необходимость использования автоматизированных средств обработки, хранения и распределения данных и информации.

Руководствуясь требованиями, которые предъявляются к работе финансовой службы крупной компании нужно отметить, что ее эффективная работа практически полностью зависит от уровня оснащенности информационными средствами на базе компьютерных систем конкретной компании.

Важно понимать, что компьютерный учет значительно отличается от обычного. Компьютер позволяет облегчить учет, сокращая время на оформление и создание документов и обработку всех хранящейся информации, касающейся торговой деятельности. Да, все эти отчеты можно сделать и без помощи ЭВМ, ведь особой математики в них не содержится, но вопрос в другом – сколько времени уйдет на их подготовку. В тот момент они уже будут не нужны. Либо же нужно будет расширять штат сотрудников, что увеличит расходы на зарплату и приведет к нулевой или отвратительной прибыли для самой компании. Таким образом, применяя компьютер количество переходит в качество, т.е. увеличивая скорость расчетов улучшается и качество самих готовых отчетов, что дает возможность улучшать схем построения бизнеса.

Сегодня практически всем компаниям приходится работать в условиях сложного финансово-экономического кризиса. И залогом успешной дальнейшей работы является правильно организованная внутренняя деятельность компании на всех без исключения уровнях, поскольку информационные технологии сегодня могут без проблем предоставить такие возможности. Следуя вышесказанному, становится ясно, что автоматизация деятельности является одной из основополагающих задач для руководства компании, которая требует решения для повышения эффективности производства.

Как итог всему выше сказанному следует отметить, что в современных условиях рыночной экономики и в условиях жесткой конкуренции проблема автоматизации и индустриализации в торговле является весьма актуальной.

Объектом исследования - гостиничный комплекс. Предмет исследования - процесс автоматизации учета работы с клиентами гостиницы по оказанию услуг.

В процессе работы была применена совокупность методов экономико-статистического анализа, методы синтеза и анализа экономической информации.

Целью работы является создание автоматизированной системы взаимоотношений с клиентами, внедрение которой бы позволило снизить трудоёмкость учета процесса и результатов взаимодействия с клиентами гостиницы и построения отчетов.

В качестве средства достижения поставленной цели будет использоваться моделирование бизнес-процессов и разработка порядка их автоматизации.

Практическая ценность результатов работы заключается в открывающейся возможности широкого использования технологии управления разработанного в сфере гостиничного бизнеса.

Кроме этого полученные методы могут быть использованы в иных сферах как при проектировании новых информационных систем управления сетью филиалов, так и для повышения эффективности существующих систем.

Выпускная квалификационная работа содержит 82 страницы из которых три основных главы (39 рисунков, 4 таблицы и 3 приложения).

В первой главе работы рассматриваются вопросы такие как, анализ современного состояния объекта исследования, обобщенные характеристики предметной области, произведена постановка задачи, выбор средств разработки, сформулированы требования по видам обеспечения ИС.

Во второй главе проектирование и реализация автоматизированной ИС гостиничного комплекса.

Третья глава посвящена тестированию разработанного проекта.

1. АНАЛИЗ СОВРЕМЕННОГО СОСТОЯНИЯ ОБЪЕКТА ИССЛЕДОВАНИЯ

1.1 Обобщенная характеристика предметной области

1.1.1 Характеристика объекта исследования

Промышленность гостиничного бизнеса исторически сложилась и выросла из раздела средств размещения, представленного разными типами гостиничных компаний. В традиционном осознании отель - это жилище с меблированными комнатами для приезжающих. В передовых критериях отель - это предприятие, предназначенное для гостиничного сервиса людей, а еще персональных туристов и санкционированных групп.

Практически гостиничные фирмы делают главные функции, например, как создают и предлагают покупателям полный гостиничный продукт, в формировании и продвижении которого принимают роль все секторы и составляющие радушия.

Исходя из сего, правомерно отметить гостиничную промышленность или же гостиничный бизнес как основную всеохватывающую компонент радушия и рассматривать ее автономно.

Сплошное инструкция предприятием воплотит в жизнь генеральный директор. Он же определяет рекламную политическому деятелю акционерного общества, направленности его становления, штатную количество, размеры и ассортимент предложений.

В реальное время увеличивается заинтересованность развитии промышленности радушия как на интернациональном, например и государственном и региональном уровнях, например как туризм и

гостиничный бизнес считаются стимулом становления всех разделов экономики.

Значимо лестные социально-экономические влияния промышленности радушия на государственном и региональном уровнях:

- свежие трудящиеся пространства, разработанные благодаря туризму и гостиничному бизнесу;

- мультипликативный эффект;

- лепта в муниципальные и региональные доходы.

Социокультурные выгоды:

- увеличение актуального значения населения;

- сбережение культурного наследия;

- помощь созданию и помощи музеев, сохранению и финансированию культурно-исторических памятников;

- ужесточение ощущения гордости населения за собственную культуру и самобытность;

- обеспечение вероятностей для межкультурных обменов.

По-этому, промышленность туризма и радушия считается одной из наиглавнейших частей просторного рынка предложений, и дает собой быстрорастущую и высокодоходную промышленность, способную напрямик и косвенно воздействовать на составление критерий стойкого социально-экономического подъема крупного и государственного хозяйства.

Конструкция гостиничного комплекса.

В конкретном руководстве у генерального директора есть подгруппы правления. Интересующей нас группой считается группа гостиничного обслуживания, в следствие этого разглядим ее организационно-штатную структуру (рис 1.1):

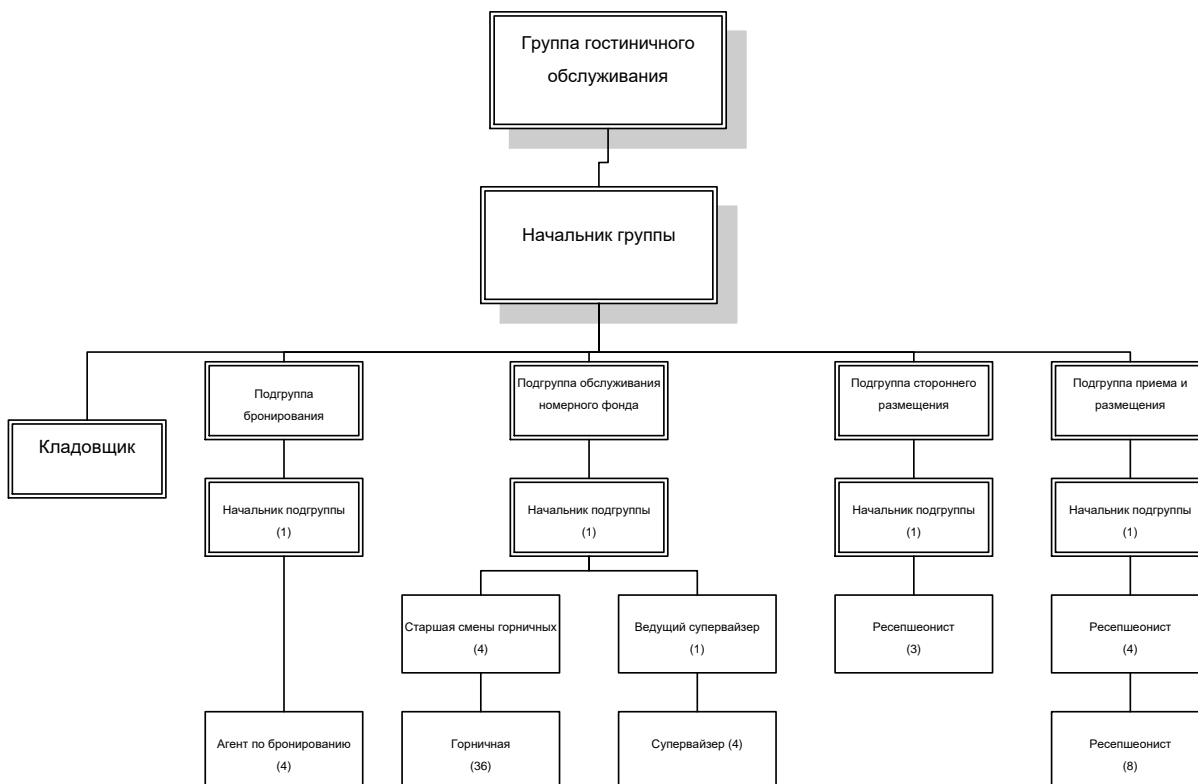


Рис. 1.1 Организационно-штатная структура группы гостиничного обслуживания

Кратко рассмотрим обязанности основных должностных лиц.

Начальник группы гостиничного обслуживания является непосредственным начальником для всего персонала группы. Он выполняет следующие обязанности:

- осуществляет контроль за качеством обслуживания проживающих клиентов в соответствии с классом обслуживания гостиницы, правильным использованием, учетом и распределением жилых номеров и мест, а также соблюдением паспортного режима;
- направляет работу персонала и служб гостиницы на обеспечение сохранности и содержания помещений и имущества в исправном состоянии в соответствии с правилами и нормами эксплуатации, бесперебойной работы оборудования, внешнего благоустройства, соблюдение санитарно - технических и противопожарных правил;

- обеспечивает рентабельное ведение гостиничного хозяйства, своевременное и качественное предоставление проживающим комплекса услуг;
- организует работу по профилактическому осмотру жилых номеров, подсобных и других помещений гостиницы, по капитальному и текущему ремонту;
- принимает меры к укреплению и расширению материально - технической базы гостиницы, повышению уровня ее комфортабельности;
- обеспечивает ведение и своевременное представление отчетности о хозяйственно - финансовой деятельности гостиницы;
- обеспечивает правильное сочетание экономических и административных методов руководства;
- осуществляет мероприятия по внедрению прогрессивных форм организации обслуживания;
- способствует развитию коммерческой деятельности.

Начальник подгруппы бронирования отвечает за работу группы, обеспечения бронирования номеров, корректность документооборота, увеличение свойства сервиса, урегулирование неоднозначных вопросов.

Начальник подгруппы сервиса номерного фонда отвечает за работу подгруппы, силами которой гарантируется обслуживание постояльцев в номерах, поддерживается важное санитарно-гигиеническое положение номеров и степень комфорта в жилых помещениях, оказываются домашние предложения живущим.

Подгруппы постороннегостороннего размещения и способа и размещения воплотят в жизнь регистрацию, рассредотачивание номеров, предложение постояльцам всевозможных информационных предложений, взимание оплаты за размещение и ведение важной документации. В прямые обязанности данных подгрупп заходит еще вступление и поддержание в

рабочем состоянии базы данных, отображающих информацию о постояльцах и состоянии номерного фонда, сбор заказов и бронирование пространств в отеле, то есть обслуживает вероятных покупателей.

В прямые обязанности кладовщика входят способ на склад, взвешивание, сбережение и выдача со склада всевозможных вещественных ценностей.

При приеме на хранение кладовщик проверяет соответствие принимаемых материальных ценностей сопроводительным документам.

В работе рассматривается деятельность администратора гостиницы.

Администратор гостиницы:

- обеспечивает работу по эффективному и культурному обслуживанию клиентов, созданию для них комфортных условий;
- осуществляет контроль над своевременной подготовкой номеров к приему прибывающих в гостиницу, соблюдением чистоты в гостинице, регулярной сменой белья в номерах, сохранностью имущества и оборудования;
- информирует проживающих в гостинице о предоставляемых дополнительных платных услугах, принимает заказы на их выполнение и контролирует их исполнение;
- дает устные справки, касающиеся гостиницы, расположения городских достопримечательностей, зрелищных, спортивных сооружений и т.д.;
- принимает и оформляет необходимые документы;
- осуществляет контроль над исполнением работниками указаний руководства организации;
- контролирует соблюдение работниками организации трудовой и производственной дисциплины, правил и норм охраны труда, техники безопасности, требований производственной санитарии и гигиены;

- принимает меры к разрешению конфликтов, возникающих при обслуживании проживающих;
- рассматривает претензии, связанные с неудовлетворительным обслуживанием клиентов и проводит соответствующие организационно-технические мероприятия;
- информирует руководство организации об имеющихся недостатках в обслуживании клиентов, принимает меры к их ликвидации.

1.1.2 Описание функции управления гостиничного комплекса

Гостиничный комплекс состоит из двух корпусов А и В. Каждый корпус имеет в своем составе по 3 здания. Здание состоит из 20 этажей. На каждом этаже располагаются по 30 номеров. В зависимости от состава номера ему присваивается класс (эконом-класс, бизнес-класс, vip-класс).

Эконом-класс: 1 комната, кол-во мест 2, 2-х спальная кровать, совмещенные ванна и туалет, балкон, шкаф, телевизор, стол, тумбочка, кондиционер.

Бизнес-класс: 2 комнаты, кол-во мест 3, диван, 2-х спальная кровать, отдельные ванна и туалет, балкон, шкаф, телевизор, стол, тумбочка, 2 кондиционера.

Vip-класс: 3 комнаты, кол-во мест 5, 2 2-х спальные кровати, отдельные ванна и туалет, 2 балкона, 2 шкафа, 2 телевизора, 2 стола, 2 тумбочки, 2 кондиционера.

В зависимости от класса номера имеется стандартный набор оказываемых услуг. Клиент может заказать дополнительные услуги.

Описание процесса работы с посетителями гостиницы (рис 1.2):



Рис. 1.2 Описание процесса работы с клиентами гостиницы

Декомпозиция данного процесса (рис 1.3):

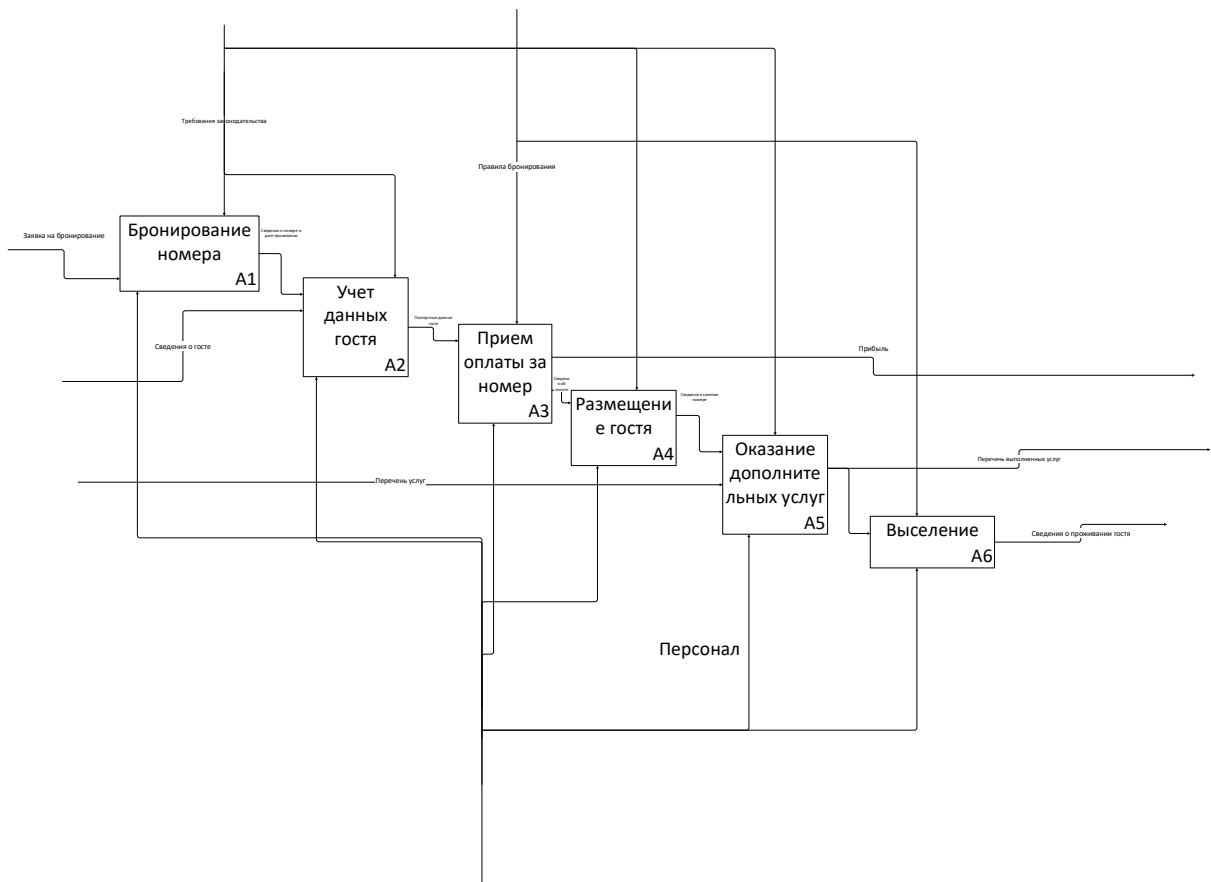


Рис. 1.3 Декомпозиция процесса работы с клиентами гостиницы

В согласовании с представленной схемой, процесс работы с покупателями заключается в регистрации прибывших гостей, распределении их по номеру фонда, планировании, проведении и учете заселения, а еще подготовке отчетности (бухгалтерской, управленческой и финансовой).

Информационные процессы цикла сервиса покупателей в отеле можно поделить на ряд групп [1]:

- «бронирование »;
- «заезд, регистрация и размещение клиентов»;
- «обслуживание постояльцев во время проживания»;
- «выезд постояльца (процедура выписки гостя)».

Разглядим возведение информационной модели - «Бронирование». Разработка бронирования подключает ряд ИП:

- сотрудник отдела бронирования заполняет бланк-заявку на бронирование;
- реквизиты клиента (Ф.И.О., государство, сроки проживания, численность номеров, категория номеров, контактный телефонный аппарат, дата приёма заявки) вводятся в электрическую систему бронирования;
- заказчик получает номер брони.

Есть некоторое количество направлений становления, содействующих подъему преданности покупателей отеля. К их количеству относятся: улучшение предоставляемых предложений, увеличение производительности рекламы и политические деятели продаж номеров, составление гибкой политические деятели ценообразования, а еще увеличение значения маневренности ансамблем в целом. Оценив конкурентную среду и личные способности, инструкция отеля надлежит избрать для собственного ансамбля подходящий баланс меж данными направленностями.

Увеличить эффективность всех повыше перечисленных направлений становления вполне вероятно с поддержкой информационных систем.

Ведомо, собственно что автоматизация бизнеса содействует существенному уменьшению трудовых и временных расходов, а итогом делается больше действенное внедрение ресурсов фирмы. В случае если фирма желает деятельно вырастать и развиваться, без использования информационных технологий в данном случае не стать.

В реальное время учет и дальнейшая работа с покупателями ведется с применением лишь только картонных форм документов и ручного труда. Гости предусматриваются в журнальчике учета, еще им заводится карта постояльца, дальше в которой изготавливаются отметки о проживании и оказываемых предложениях. Рассредотачивание гостей по номерному фонду ведется еще вручную на базе картонной схемы занятости номерного фонда. Эта постановка вопроса не разрешает действенно рулить работой с гостями, получать доклады, а еще задумывать загруженность номерного фонда.

В реальное время применяемая система организации бронирования владеет надлежащими дефектами:

1. низкая скорость обработки документов;
2. большое численность некорректностей и промахов, связанных с человеческим факторам;
3. сложность в построении докладов и розыске ситуации определенного документа;
4. сложность в отслеживании прохождения документа в фирмы и его рецензирования.

Данные дефекты возможно убрать методом внедрения автоматической системы бронирования номеров отеля.

1.2 Актуальность разработки ИС

Гостиничный комплекс содержит огромные потенциальные способности для увеличения прибыли. Высочайшее качество предлагаемых

предложений – это комфортабельное месторасположение комплекса, в одном из курортных населенных пунктов Рооси, дешевые стоимости – все это надлежит направлению к предстоящему развитию фирмы.

Не обращая внимания на эти имеющиеся вероятные способности, они не применяются в абсолютной мере. Это вызвано тем, собственно что фирма не популярна и не доступна в сети Онлайн, собственно что большущее численность покупателей в реальное время отыскивает партнеров для выполнения важных предложений.

Посодействовать в заключении данной трудности имеет возможность осуществление веб-представительства, которое, не считая нормальной информационной функции, станет владеть еще и функцию on-line наполнения заказов на бронирование номеров, благоприятный безналичный расплата, простота и доступность наполнения заказы на бронь. Спасибо веб-представительству, покупатели смогу оформить заявку на предложение предложения выделено, не приезжая в кабинет отеля, собственно что важно упрощает покупателю жизнь и сберегает собственное время.

В ходе собственной работы админ отеля пользуется всевозможные бумаги, между коих ведущими считаются предшествующая заказ покупателя и контракт реализации.

В ходе работы с покупателем админ отеля поочередно делает надлежащие воздействия:

- принимает заявку, уточнив неоднозначные вопросы у клиента;
- регистрирует клиента;
- идентифицирует клиента;
- выясняет особенности покупателя и степени соотношения притязаниям клиента;
- представляет покупателю данные о брони номера в комфортное ему время;
- при позитивном заключении покупателя оформляет бронь.

Перечень покупателей, предшествующая заказ, описания номеров присутствуют в всевозможных файлах, собственно что заставляет по очереди обращаться к любому из них. При оформлении соглашений админ обязан вручную выносить этот из файла перечнем покупателей в шаблон контракта, собственно что негативно воздействует на скорости его работы. В итоге возрастает время дизайна брони, собственно что в собственную очередь наращивает время сервиса 1-го покупателя и сокращает сплошное численность обслуженных покупателей за пролетарий денек.

При планируемой автоматизации нужно беречь все данные в единственной основе данных, собственно что дозволит прирастить скорость работы админа за счет самодействующего подбора номеров, регистрации и идентификации покупателей, дизайна выходных документов в частности, брони.

1.3 Постановка задачи на разработку ИС

Разрабатываемая система предназначена для автоматизации бронирования номера клиентами отеля. Система обязана быть исполнена в виде веб-сайта с вероятностью автономного выбора номеров покупателями и бронирования их.

Для заказа бронирования пространства в отеле заказчик регится на веб-сайте. Избирает важный ему ос номер и делает заявку на бронирование пространства (дата заезда, дата выезда, численностьколичество людей) при данном в ней он лицезреет какие предложения считаются стереотипными для сего номера а какие вспомогательные. Вспомогательные предложения возможно до заказать, при данном возрастает цена проживания в день.

Впоследствии такого как заказ оформлена в течении 10 часов на номер покупателя поступает сигнал и СМС с информацией по его заказу. В случае если заказ на бронирование одобряется то заказчик оплачивает номер хоть каким легкодоступным методикой или по прибытии в гостиницу или по

карте и имеет возможность брать в долг номер в обозначенные часы. Когда заказчик приезжает в гостиницу он именуется номером заказа и впоследствии администратором ему выдается источник. В течение присутствия в отеле заказчик имеет возможность заказывать вспомогательные предложения, которые оплачиваются им порознь. Во время присутствия покупателя ему назначаются рейтинги по поведению, по порче имущества, по культуре общения, по чистоте номера, по выполнению правил внутреннего распорядка. Данные рейтинги проставляются администратором гостиничного строения способом собственного общения с покупателем, выборочного опроса персонала контактирующего с покупателем и зрительного осмотра номера покупателя.

Впоследствии выезда в зависимости от приобретенных с покупателя валютных средств, и рейтингов покупателю назначается статус. В зависимости от статуса покупателя ему выдается карточка на скидку в 5%, 10%, 15%.

При регистрации заказчик показывает: ФИО, телефонный аппарат, email.

Для сотворения заказы на бронирование дополнительно указываются паспортные данные: серия, номер, когда и кем выдан, код отряда, дата рождения.

Не считая обслуживания проживания наличествует: кафе, боулинг, фитнес-центр, баня, прачечная, салон красоты.

2 . ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ ГОСТИНИЧНОГО КОМПЛЕКСА

2.1 Проектирование базы данных

Для обеспечения целостности данных работы информационной системы гостиничного комплекса необходимо, чтобы таблицы базы данных были связаны между собой. В процессе загрузки и корректировки базы данных для получения информации по запросам и вывода отчетов осуществляется объединение данных из взаимосвязанных таблиц.

Связи между таблицами устанавливаются в соответствии с проектом логической структуры базы данных, определяются и запоминаются в схеме данных СУБД. Схема данных является не только средством графического отображения логической структуры базы данных, она активно используется системой в процессе обработки данных. Система при необходимости обработки данных двух взаимосвязанных таблиц автоматически использует связи, определенные в схеме данных. Таким образом, у разработчика нет необходимости специально сообщать системе о наличии той или иной связи. Однажды указанные в схеме данных связи используются системой автоматически. Создание схемы данных позволяет упростить конструирование многотабличных форм, запросов, отчетов и страниц доступа к данным, а также обеспечить поддержание целостности взаимосвязанных данных при корректировке таблиц.

В базе данных информационной системы гостиничного комплекса главной таблицей является «Пользователи». Если таблицу не разбивать на подтаблицы, то можно наблюдать избыточность данных. В соответствии с предметной областью была создана физическая таблица (рис 2.1):

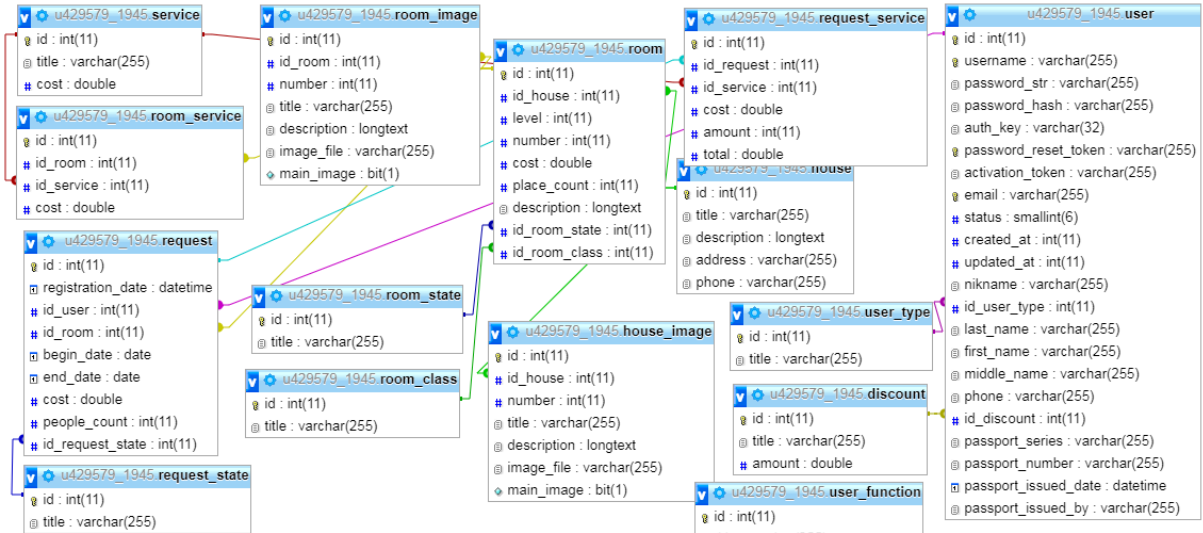


Рис 2.1 Таблицы связей базы данных (Физическая)

При реализации базы данных гостиничного комплекса было создано 15 таблиц (рис 2.2):

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
discount	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	16 КБ	-
house	Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8_general_ci	16 КБ	-
house_image	Обзор Структура Поиск Вставить Очистить Удалить	4	InnoDB	utf8_general_ci	32 КБ	-
request	Обзор Структура Поиск Вставить Очистить Удалить	6	InnoDB	utf8_general_ci	64 КБ	-
request_service	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	48 КБ	-
request_state	Обзор Структура Поиск Вставить Очистить Удалить	6	InnoDB	utf8_general_ci	16 КБ	-
room	Обзор Структура Поиск Вставить Очистить Удалить	6	InnoDB	utf8_general_ci	80 КБ	-
room_class	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	16 КБ	-
room_image	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	32 КБ	-
room_service	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8_general_ci	48 КБ	-
room_state	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	16 КБ	-
service	Обзор Структура Поиск Вставить Очистить Удалить	5	InnoDB	utf8_general_ci	32 КБ	-
user	Обзор Структура Поиск Вставить Очистить Удалить	4	InnoDB	utf8_unicode_ci	96 КБ	-
user_function	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	16 КБ	-
user_type	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	16 КБ	-
15 таблиц	Всего	53	InnoDB	utf8_general_ci	544 КБ	0 байт

Рис 2.2 Структура данных гостиничного комплекса

Таблица 2.1

Перечень таблиц

Название	Определение
discount	Скидка для пользователей
house	Таблица информации сайта на главной странице
house_image	Данные изображения гостиничного комплекса на главной странице
request	Регистрация пользователя
request_service	Данные парковочных мест и заявок
request_state	Данные заказов
room	Информация гостиничного комплекса
room_class	Виды номеров
room_image	Данные изображения гостиничного номера
room_state	Перечень услуг гостиничного комплекса
service	Пользователи
user	Данные должностей гостиничного комплекса
user_type	Тип пользователя

2.2 Создание и заполнения базы данных

Теперь разберем подробное создание и заполнение таблиц в базе данных. Первая таблица называется «discount - скидка пользователей», на рисунке 2.3 представлено создание таблицы.

```
-- Структура таблицы `discount`
--
CREATE TABLE IF NOT EXISTS `discount` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(255) DEFAULT NULL,
  `amount` double DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AVG_ROW_LENGTH=5461 AUTO_INCREMENT=4 ;

--
-- Дамп данных таблицы `discount`
--

INSERT INTO `discount` (`id`, `title`, `amount`) VALUES
(1, '5 процентов', 5),
(2, '10 процентов', 10),
(3, '15 процентов', 15);
```

Рис. 2.3 Создание таблицы «discount»

Таблица «Скидка» (discount), которая отображается в базе данных представлена на рисунке 2.4 и содержит следующие атрибуты.

- Id (int(11)) - генерация уникального идентификатора для новых строк.
- Title (varchar(255)) - наименование скидки, предполагает не более 255 символов.
- Amount (double) - процент скидки с плавающей запятой.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Первичный Ещё
2	title	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё
3	amount	double			Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё

Рис. 2.4 Готовая структура данных таблицы «Скидка»

Данная сущность (таблица) имеет сформированный вид данных с заранее заполненными полями на рисунке 2.5. Изменение данных в таблице возможно только администратором системы (сотрудником имеющим доступ к внутренней структуре, полям базы данных).

	id	title	amount
<input type="checkbox"/> Изменить Копировать Удалить	1	5 процентов	5
<input type="checkbox"/> Изменить Копировать Удалить	2	10 процентов	10
<input type="checkbox"/> Изменить Копировать Удалить	3	15 процентов	15

Рис. 2.5 Примерное представление таблицы «Скидка»

Следующая таблица называется «house - таблица информации сайта на главной странице», на рисунке 2.6. представлено создание таблицы.

```

CREATE TABLE IF NOT EXISTS `house` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(255) DEFAULT NULL,
  `description` longtext,
  `address` varchar(255) DEFAULT NULL,
  `phone` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AVG_ROW_LENGTH=16384 AUTO_INCREMENT=2 ;

--
-- Дамп данных таблицы `house`
--

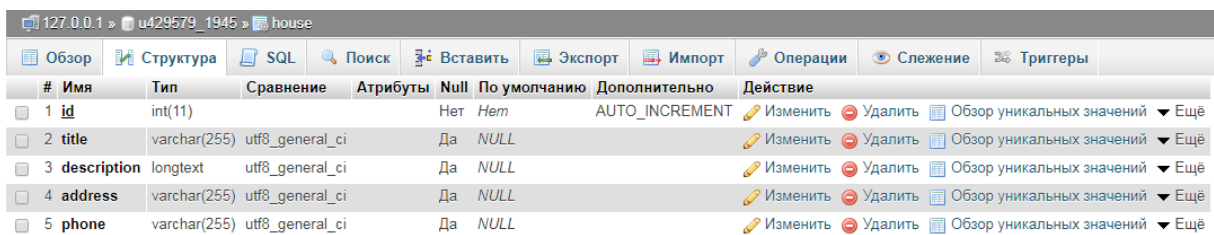
INSERT INTO `house` (`id`, `title`, `description`, `address`, `phone`) VALUES
(1, 'Гостиничный комплекс', 'Отель, открытый в 2013 г., расположен на второй бере

```

Рис. 2.6 Создание таблицы «house»

В таблице «Информация сайта» на рисунке 2.7 (house) представим всю необходимую информацию про гостиничный комплекс:

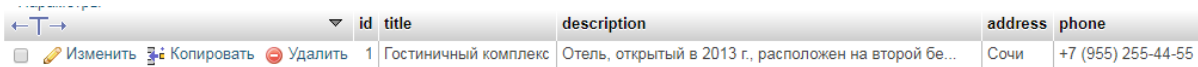
- id (int(11)) - генерация уникального идентификатора, для нашего сайта подойдет и одна строка, но сделаем таблицу более гибкой;
- title (varchar(255)) - наименование гостиничного комплекса, предполагает не более 255 символов;
- description (longtext) - содержит всю информацию о гостиничном комплексе, ограничим данные 65 536 символами;
- address (varchar(255)) - одержит адрес гостиничного комплекса, предполагает не более 255 символов;
- phone (varchar(255)) - телефонный номер гостиничного комплекса.



#	Имя	Тип	Атрибуты	Null	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	id	int(11)		Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	title	varchar(255)	utf8_general_ci	Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	description	longtext	utf8_general_ci	Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	address	varchar(255)	utf8_general_ci	Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	phone	varchar(255)	utf8_general_ci	Да	NULL		Изменить Удалить Обзор уникальных значений Ещё

Рис. 2.7 Структура данных таблицы «Информация сайта»

Данная сущность (таблица) имеет сформированный вид данных с заранее заполненными полями на рисунке 2.8. Изменение данных в таблице возможно только администратором системы (сотрудником имеющим доступ к внутренней структуре, полям базы данных).



id	title	description	address	phone
1	Гостиничный комплекс	Отель, открытый в 2013 г., расположен на второй бе...	Сочи	+7 (955) 255-44-55

Рис. 2.8 Примерное представление таблицы «Информация ос сайта»

Таблица «Изображение» (house_image) на рисунке 2.9 содержит в себе:

- id (int(11)) - идентификатор;
- id_house (int(11)) - идентификатор корпуса, целочисленное значение;
- number (int(11)) - идентификатор номера, целочисленное значение не превышающее 11 символов;
- title ос (varchar(255)) - наименование номера, предполагает не более 255 символов;
- description ос (longtext) - содержит всю информацию о гостиничном номере, ограничим данные 65 536 символами;
- image_file ос (varchar(255)) - будем хранить наименования изображения;
- main_image ос (bit(1)) - данный флаг будем использовать для размещения.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Ещё
2	id_house	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
3	number	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
4	title	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
5	description	longtext	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
6	image_file	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
7	main_image	bit(1)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё

Рис. 2.9 Структура данных таблицы «Изображение»

Данная сущность (таблица) имеет сформированный вид данных с заранее заполненными полями на рисунке 2.10. Изменение данных в таблице возможно только администратором системы (сотрудником имеющим доступ к внутренней структуре, полям базы данных).

	id	id_house	number	title	description	image_file	main_image
Изменить Копировать Удалить	1	1	1	Главный вид отеля		3eab8ab4e9fd3207817720ac9b2699e4.jpg	1
Изменить Копировать Удалить	2	1	2	Главный вид отеля		55.jpg	0
Изменить Копировать Удалить	3	1	3	Главный вид отеля		05.jpg	0
Изменить Копировать Удалить	4	1	4	Главный вид отеля		ada4c5feeabc48328790057896fc7508.jpg	0

Рис. 2.10 Примерное представление таблицы «Изображение»

Следующая таблица называется «request - регистрация пользователя», на рисунке 2.11 представлено создание таблицы.

```

CREATE TABLE IF NOT EXISTS `request` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `registration_date` datetime DEFAULT NULL,
  `id_user` int(11) DEFAULT NULL,
  `id_room` int(11) DEFAULT NULL,
  `begin_date` date DEFAULT NULL,
  `end_date` date DEFAULT NULL,
  `cost` double DEFAULT NULL,
  `people_count` int(11) DEFAULT NULL,
  `id_request_state` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FK_request_request_state_id` (`id_request_state`),
  KEY `FK_request_room_id` (`id_room`),
  KEY `FK_request_user_id` (`id_user`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AVG_ROW_LENGTH=5461 AUTO_INCREMENT=7 ;

```

Рис. 2.11 Создание таблицы «request»

Таблица «Гости» (request) представлена на рисунке 2.12 и включает в себя:

- id (int(11)) - идентификатор;
- registration_date (datetime) - дата регистрации;
- id_user int(11) - идентификатор пользователя;
- id_room int(11) - идентификатор номера;
- begin_date date - дата заезда;
- end_date date - дата отъезда;
- cost double - стоимость номера;
- people_count int(11) - количество человек;
- id_request_state int(11) - идентификатор регистрации.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	1 id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	2 registration_date	datetime			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	3 id_user	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	4 id_room	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	5 begin_date	date			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	6 end_date	date			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	7 cost	double			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	8 people_count	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
<input type="checkbox"/>	9 id_request_state	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё

Рис. 2.12 Структура данных таблицы «Гости»

Таблица «Парковка» представлена на рисунке 2.13 и содержит:

- id int(11) - идентификатор;
- id_request int(11) - идентификатор гостя;
- id_service int(11) - идентификатор услуги (парковка);
- cost double - стоимость парковочного места;
- amount int(11) - количество;
- total double - итоговая сумма.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	1 id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Первичный Ещё
<input type="checkbox"/>	2 id_request	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё
<input type="checkbox"/>	3 id_service	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё
<input type="checkbox"/>	4 cost	double			Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё
<input type="checkbox"/>	5 amount	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё
<input type="checkbox"/>	6 total	double			Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё

Рис. 2.13 Структура данных таблицы «Парковка»

Следующая таблица называется «request_state» - данные заказов, на рисунке 2.14 представлено создание таблицы.

Таблица выводит информацию о том, на каком этапе находится заявка. Статус заявки может быть: активным, принятым, отклоненным, выполненным, может находиться в процессе выполнения или редактирования.

```

CREATE TABLE IF NOT EXISTS `request_state` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AVG_ROW_LENGTH=2730 AUTO_INCREMENT=7 ;

--
-- Дамп данных таблицы `request_state`
--

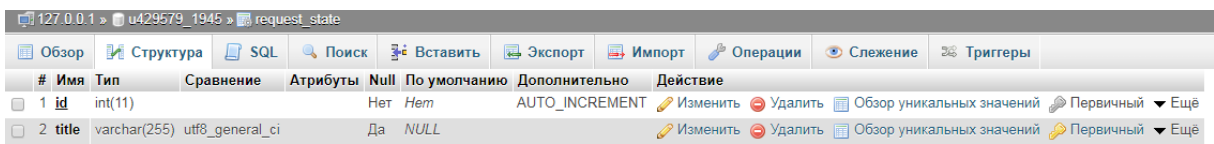
INSERT INTO `request_state` (`id`, `title`) VALUES
(1, 'Редактируется'),
(2, 'Активна'),
(3, 'Принята'),
(4, 'Выполняется'),
(5, 'Выполнена'),
(6, 'Отклонена');

```

Рис. 2.14 Создание таблицы «request_state»

Таблица «Список заказов» представлена на рисунке 2.15 и включает в себя:

- id int(11) - идентификатор;
- title varchar(255) - состояние заказа.

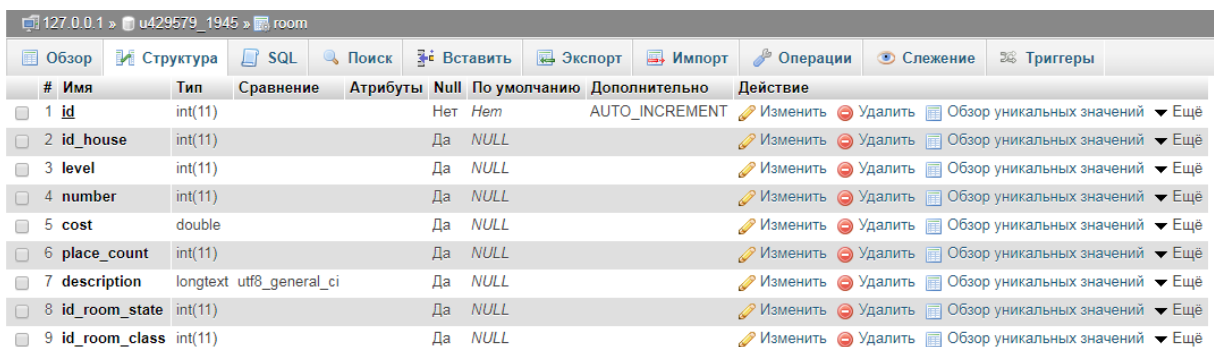


#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Первичный Ещё
2	title	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё

Рис. 2.15 Структура данных таблицы «Список заказов»

Таблица «Номер» представлена рисунке 2.16 и содержит:

- id int(11) -идентификатор;
- id_house int(11) - идентификатор ос корпуса;
- level int(11) - этаж;
- number int(11) - номер;
- cost double - стоимость.;
- place_count int(11) - количество ос мест;
- description longtext - описание ос номера;
- id_room_state int(11) - идентификатор ос состояния ос комнаты;
- id_room_class int(11) - идентификатор ос класса ос номера.



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Ещё
2	id_house	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
3	level	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
4	number	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
5	cost	double			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
6	place_count	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
7	description	longtext	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
8	id_room_state	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
9	id_room_class	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё

Рис. 2.16 Структура данных таблицы «Номер»

Таблица «Виды номеров» продемонстрирована на рисунке 2.17 и включает в себя:

- id int(11) - идентификатор;
- title varchar(255) - наименования видов комнат.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Первичный Ещё
2	title	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё

Рис. 2.17 Структура данных таблицы «Виды номеров»

Таблица «Изображение номеров» продемонстрирована на рисунке 2.18 и включает в себя:

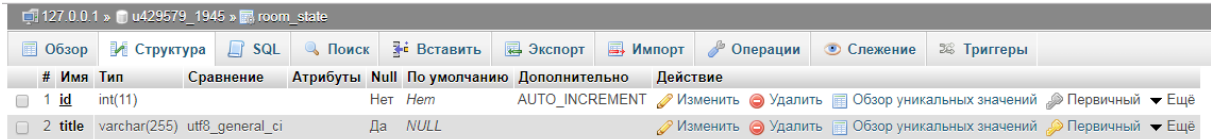
- id int(11) - идентификатор;
- id_room int(11) - идентификатор ос номера;
- number int(11) - номер ос комнаты;
- title varchar(255) - наименование ос комнаты;
- description longtext - описание ос номера;
- image_file varchar(255) - имя ос файла ос изображения ос номера;
- main_image bit(1) - флаг ос видимости.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Ещё
2	id_room	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
3	number	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
4	title	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
5	description	longtext	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
6	image_file	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
7	main_image	bit(1)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё

Рис. 2.18 Структура данных таблицы «Изображение номеров»

Таблица «Статус номеров» продемонстрирована на рисунке 2.19 и включает в себя:

- id int(11) - идентификатор;
- title varchar(255) - статус номера.

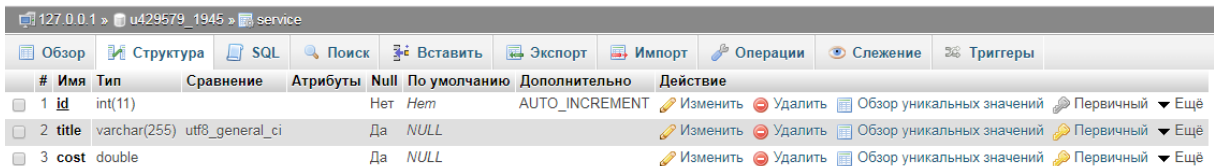


#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Первичный Ещё
2	title	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё

Рис. 2.19 Структура данных таблицы «Статус номеров»

Таблица «Услуги» продемонстрирована на рисунке 2.20 и включает в себя:

- id int(11) - идентификатор;
- title varchar(255) - с наименованием услуги;
- cost double - стоимость услуги.



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Первичный Ещё
2	title	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё
3	cost	double			Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё

Рис. 2.20 Структура данных таблицы «Услуги»

Таблица «Пользователи» продемонстрирована на рисунке 2.21 и включает в себя:

- id int(11) - идентификатор;
- username varchar(255) - пользователь;
- password_str varchar(255) - пароль;
- password_hash varchar(255) - пароль;
- auth_key varchar(32) - пароль;
- password_reset_token varchar(255);
- activation_token varchar(255) - код активации;

- email varchar(255) - электронная почта;
- status oc smallint(6) - статус пользователя;
- created_at int(11) - код редактирования;
- updated_at int(11) - код удаления;
- nickname varchar(255) - логин;
- id_user_type int(11) - тип пользователя;
- last_name varchar(255) - имя;
- first_name varchar(255) - фамилия;
- middle_name varchar(255) - отчество;
- phone varchar(255) - телефон;
- id_discount int(11) - идентификатор скидки;
- passport_series varchar(255) - серия паспорта;
- passport_number varchar(255) - номер паспорта;
- passport_issued_date datetime - дата выдачи паспорта;
- passport_issued_by varchar(255) - кем выдан паспорт.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Ещё
2	username	varchar(255)	utf8_unicode_ci		Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
3	password_str	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
4	password_hash	varchar(255)	utf8_unicode_ci		Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
5	auth_key	varchar(32)	utf8_unicode_ci		Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
6	password_reset_token	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
7	activation_token	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
8	email	varchar(255)	utf8_unicode_ci		Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
9	status	smallint(6)			Нет	10		Изменить Удалить Обзор уникальных значений Ещё
10	created_at	int(11)			Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
11	updated_at	int(11)			Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
12	nickname	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
13	id_user_type	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
14	last_name	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
15	first_name	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
16	middle_name	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
17	phone	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
18	id_discount	int(11)			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
19	passport_series	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
20	passport_number	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
21	passport_issued_date	datetime			Да	NULL		Изменить Удалить Обзор уникальных значений Ещё
22	passport_issued_by	varchar(255)	utf8_unicode_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Ещё

Рис. 2.21 Структура данных таблицы «Пользователи»

Таблица «Должность» продемонстрирована на рисунке 2.22 и включает в себя:

- id int(11) - идентификатор;
- title varchar(255) - должность.
-

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Первичный Ещё
2	title	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё

Рис. 2.22 Структура данных таблицы «Должность»

Таблица «Пользователь» продемонстрирована на рисунке 2.23 и включает в себя:

- id int(11)- идентификатор;
- title varchar(255) - пользователь.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Первичный Ещё
2	title	varchar(255)	utf8_general_ci		Да	NULL		Изменить Удалить Обзор уникальных значений Первичный Ещё

Рис. 2.23 Структура данных таблицы «Пользователь»

В результате анализа информационного обеспечения выявлены и сформулированы ограничения и правила поддержания целостности данных, которые должны быть учтены при дальнейшем проектировании. Таким образом, было создано 15 таблиц, 86 атрибутов.

2.3 Используемые классификаторы и система кодирования

В составе исследуемого комплекса задач особенное пространство определено классификаторам финансовых данных: обеспечивать сжатие прозрачной части (ID) характеристик, а еще уменьшить размер сохраненных данных в ЭВМ и время на нахождение данных, подходящих для выполнения задач, облегчить обработку данных может помочь их классификация и кодирование.

Классификатор возможно именовать документом, который может помочь воплотить в жизнь формализованное описание финансовых данных в ИС, включающей названия объектов, названия

классификационных группировок и их конкретные обозначения. В рамках использования они имеют все шансы делиться на 3 на подобии:

- общегосударственные классификаторы, используемые во множества секторах экономики и на множества уровнях управления для схожего и повсеместного обозначения объектов;
- отраслевые (ведомственные) классификаторы, используемые в границах отдельной отрасли(ведомства);
- местные, используемые в границах фирмы или же группы фирм.

Для абсолютной формализации финансовых данных невозможно их элементарно систематизировать, в следствие этого нередко реализуется процедура кодировки.

Кодирование считается ходом наделения относительными обозначениями объектов и классификационных групп по кое-какой системе кодировки.

Система шифровки считается сочетанием правил определения объектов и группировок с использованием кодов.

Кодом возможно именовать кое-какое обозначение объектов или же группировок в облике символа или же совокупы символов в рамках правил

принятой системы. Все системы кодировки группируют на 2 подмножества: классификационные и регистрационные системы кодировки.

Характеристики, которые надо удовлетворять разрабатываемым классификаторам, таковы:

- общность охвата объектов и симптомов систематизации хоть каким классификатором;
- общность симптомов разделения множеств объектов с способами обработки финансовых данных;
- однозначность имен объектов и их кодовых названий;
- доступность кодировки и вероятность автоматизации кодировки и классификации;
- доступность увязки с другими классификаторами и способами обозначения;
- целесообразность использования классификатора в процессе обработки данных.

В этой ВКР станут применены надлежащие локальные классификаторы.

1) Классификатор служащих (пользователей ИС).

Структурная схема классификатора служащих представлена на рисунке 2.24.

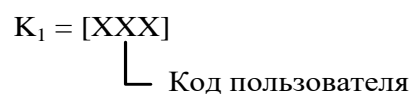


Рис. 2.24 Структурная схема классификатора сотрудников

2) Классификатор номеров

Структурная схема классификатора представлена на рисунке 2.25.

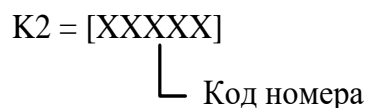


Рис. 2.25 Структурная схема классификатора документов

3) Классификатор брони

Структурная схема классификатора представлена на рисунке 2.26



Рис. 2.26 Структурная схема классификатора брони

Сводное описание всех вышеописанных классификаторов представлено в таблице 2.2.

Таблица 2.2

Сводная таблица используемых классификаторов и систем кодирования

№ п/п	Наименование кодируемого множества	Значимость кода	Система кодирования	Мощность кода	Система классификации	Вид классификатора
1	Пользователи	3	порядковая	999	отсутствует	локальный
2	Номера	5	разрядная	99999	отсутствует	локальный
3	Бронь	8	разрядная	99999999 9	отсутствует	локальный

Техническое обеспечение - это индивидуальный компьютер, оргтехника, части связи, оснащение сеток. Картина информационной технологии, зависящий от технической оснащенности (ручной,

автоматический, удаленный) воздействует на сбор, обработку и передачу инфы.

Комплекс технических средств оформляют:

- компьютеры;
- устройства сбора, скопления, обработки, передачи и вывода инфы - строгие диски, прибора сбережения данных, сканеры, принтеры, факсимильные аппараты;
- устройства передачи данных и рядов связи - модемы;
- эксплуатационные материалы - бумага, CD (DVD)- диски и т.п.

При выборе компа нужно управляться вблизи рядом данных. К этим чертам относятся надежность, стоимостные издержки, производительность, простота применения и др. От смысла обозначенных характеристик находится в зависимости вероятность работы с требуемыми программными способами, а значит, и триумф сотворения системы.

Для всякого из составляющих предоставленной схемы отличается список критериев, более весомых при претворении в жизнь выбора технического обеспечения. Эти аспекты таковы:

- тактовая частота процессора;
- разрешение монитора;
- объем оперативной памяти.

Для заключения установленной задачи нужно применить индивидуальный компьютер с уровнем вычислительной мощности AMD 2000 Мгц, или Intel 2000 Мгц, с размером оперативной памяти от 2 Гб. Система оптимизирована для работы в экранном разрешении 1024x768 на мониторе диагональю 17 дюймов. Все эти технические способы владеют необходимой для заключения задачи конфигурацией. В таблице 2.3 приведены свойства оптимальных трудящихся станций.

Таблица 2.3

Характеристики рабочих станций

Модель процессора	Intel Core i3-370M 2.4GHz	Intel Atom Single Core N270 - 1.66GHz
Оперативная память	3072Mb	2048Mb
Жесткий диск	320Gb	160 Gb
Оптический привод	Нет	DVD-RW
Видеоадаптер	Интегрирован в чипсет	Интегрирован в чипсет

Для печати, сканирования и копирования документов надлежит существовать отвечающее оснащение. Для обеспечения сохранности данных при аварийном выключении электропитания индивидуальный компьютер обязан быть оборудован блоком бесперебойного питания.

В предоставленном облике информационная система станет готова к внедрению автоматической системы отеля.

2. 4 Требования программного обеспечения

Программное обеспечение системы должно быть достаточным для выполнения реализуемых функций системы, а еще владеть средством организации требуемых процессов обработки данных, разрешающих вовремя выполнять необходимые функции во всех режимах функционирования системы [18].

Для хорошего функционирования системы необходимо наличие операционной системы Microsoft Vista/7/8.

Для выбора языка программирования нужно определить требования к среде программирования. В процессе анализа требований и систематизации самой проектируемой системы были определены следующие требования к среде проектирования:

- кроссплатформенность, необходимая для интеграции с другими подсистемами;
- гибкость, позволяющая без дополнительных затрат изменять, дорабатывать информационную систему;
- безопасность;
- язык программирования должен предоставлять широкие возможности по работе с современными СУБД [48];
- открытость исходного кода, которая позволит динамическое совершенствования ИС и исправление ошибок [26].

В реальное время более всераспространены следующие скриптовые языки веб-программирования [27].

- Visual Basic Scripting Edition;
- JScript;
- Python;
- Ruby;
- PHP;
- Perl.

Проведем сравнительный анализ (табл. 2.4) перечисленных языков программирования по 6 аспектам, который актуальный на сегодняшний день средств разработки.

Анализ начнется с имеющихся на этот момент языков с технологий создания веб-проектов, численность которых не очень большое, но любой язык или технология в кое-чем лучше другого и разработан для более узкой цели, чем другой. Приведем существующие языки программирования и технологии и опишем их.

Таблица 2.4

Сравнительный анализ языков программирования

Критерий	Perl	PHP	ASP.NET	JSP	Ruby
Кроссплатформенность	+	+	+	+	+

Гибкость	+	+	+	+	-
Безопасность	-	+	+	+	+
Поддержка современных СУБД	+	+	+	+	+
Универсальность	-	+	-	-	-
Открытость исходного кода	+	+	-	-	-

PHP возможно расшифровать как гипертекстовый препроцессор. В случае если готворить обычным языком, то это язык программирования, который применяется для генерации страничек вебсайта, разбирающихся на гипертекстовую разметку. Сейчас PHP считается одним из более распространённым язвком программирования для массовой сети: на нем сотворено уже больше 10 милл. русских веб-сайтов. Почти все обладатели хостингов для собственных юзеров по умолчанию включают интерпретатор. Он считается достаточно обычным языком, владеющим высочайшей скоростью обработки и разрешает включать иные модули. Применяя PHP в связке с другими модулями возможно без задач трудиться с криптографической обороной, графикой, базами данных, документацией и т.д. Интерпретаторы - аппаратное или же программное средство, выполняющее тест, обработку и выполнение начального кода или же программки, для языка PHP сейчас есть для всех типов ОС.

ASP - разработка, представляющая собой кое-какие странички на сервере, которые при вызове делают кое-какой программный код. Как поведет себя юзер, например и отреагирует странички. Предоставленная разработка разрешает делать странички вебсайтов в облике маленьких приложений, довольно аналогичными на обыденные программки. Как правило целый код в ASP разделяют на серверную и клиентскую доля. Последняя сотворена для формирования зрительного представления вебсайта, а вот серверная доля отвечает за выполнение всех поступков -

аутентификацию юзера на сервере, работу с базами ос данных, передачу и замен информацией, сбережение данных. Чисто на техническом уровне это всего только текстовые файлы с кусочками HTML-кода, скриптами на Java и Basic.

ASP.NET считается продолжением ASP-технологии. Впоследствии обновления перед почти всеми программерами стал некоторый программный продукт, похожий с Microsoft ASP, но поддерживающий важно более функций. В принципе предоставленная среда стала одной из частей плана по развитию Онлайн - платформы NET Framework - программном компоненте, выпущенно фирмой Microsoft. Почвой для предоставленного ПО стала исполняющая среда Common Language Runtime (CLR), которая поддерживает пуск как нормальных программ, например и серверных приложений NET Framework поддерживает разработку программ, написанных на различных языках программирования. А все веб-проекты, реализованные по свежей технологии, выделяются собственной быстротой выполнения и компиляции. Для больше высококачественного процесса программирования еще сделали свежую среду разработки и отладчик, коим имел возможность без задач обнаружить почти все сокрытые и очевидные промахи ещё на рубеже написания кода. Еще перетерпела конфигурации компиляция и осуществление кода меж try и catch операторами. Добавились свежие составляющие управления, собственно что важно упрощало разработку самого интерфейса для юзера.

Perl прежде позиционировался для сотворения докладов на базе собранных из хранилища данных, в следствие этого применялся он в ведущем для работы с гигантскими базами данных, из коих было надо изымать отдельные материалы и представлять в больше читабельном облике. Эмблемой языка стал верблюд, который сам с облику ужасен и неказист, но имеет возможность длительное время исполнять 1 и ту же изнурительную работу. Perl еще деятельно применялся при разработке вебсайтов, потому что умеет очень хорошо трудиться с базами данных и

поддерживает большую численность инструментов для работы с словом. Сейчас он уже применяется не например нередко, но до сих времен кое-какие серверные утилиты для обработки слова и энциклопедические веб-сайты возводятся как раз на нем.

JSP - разработка, представляющая собой странички, написанные на Java с производимые на сервере, собственно что более всего находится в зависимости от компилятора. При применении Jasper сначала формируется высоко оптимизированный байтовый код, который вслед за тем передается Java-машине для выполнения. А при применении Tomcat сервлет-контейнера -программы, представляющей из себя сервер, поддерживающий обычную нормальную работу сервлетов на протяжении всего из актуального цикла в зависимости от спецификации. Имеет возможность применяться и как автономный сервер, и быть поставщиком страничек для иного веб-сервера, а еще встраиваться в java EE сервер приложений. JSP поддерживает замен данными меж сервлетом и покупателями, делает нередко эти функции, как создание программной среды для обычной работы сервлета, авторизацию и идентификацию юзеров, возведение сессии для них. Считается достаточно известным заключение при разработке интернет-каталогов или же аукционов. Тайна языка в том, собственно что отлаженные код имеет возможность производиться как на сервере, например и в пользовательском браузере, еще наличествует вероятность полновесного программирования с классами, делением прав доступа и иными удобствами, предоставляемыми настоящей Java.

HTML можно смело именовать основой всего онлайн. Создателю возможно применить всевозможные фреймы и строчить на всяком языке программирования, но лишь только при поддержке HyperText Markup Language вполне вероятно графическое представление данных в окошке браузера. Готовая страничка как правило произведено из комплекта тегов, между коих располагается сам слово и производится включение иных посторонних компонент. Почаще всего теги видятся парные - с раз

раскрывает и имеет в для себя все опции слова, другие записи, а вслед за тем запирается. Случаются и одинарные теги. Спасибо HTML разработчик программного обеспечения, верстальщик и в том числе и обыватель юзер имеют все шансы трудиться с графикой, дополнениями браузера, таблицами, документами, шрифтами, фреймами, другими технологиями программирования и т.д.

XHTML считается адептом гипертекстовой разметки, но реализован не на тегах, а на иерархиях XML. С технической точки зрения эта разметка не достаточно выделяется от оригинала, но вот в проекте читабельности важно выигрывает, потому что в XML все надлежит находится строго и по порядку. Структурированная иерархия освобождает ядро браузера от безграничного сбережения шаблонов, а еще сокращает время на обработку и отражение странички, собственно что выделяет вероятность применить это техническое заключение на мобильных аксессуарах.

CSS применяется программерами для разработки наборов стилей на базе объектно-ориентированного программирования. Сущность языка заключается в способности описания классов на страничке и всех входящие объектов. Т.е. возможно показать краску, шрифт, метод отражения, величина и иные качества для всякого объекта и применить готовый манера необходимое численность один без вспомогательного описания. То есть, внедрение каскадных таблиц стилей важно уменьшает величина HTML-кода, а еще, разрешает настраивать отражение вебсайта для различных браузеров и приборов. При это возможно включить иной комплект стилей, и документ воспримет свежий картина без конфигурации кода. РазработкаТехнология каскадности разрешает группировать объекты различных категорий в раз контейнер и задавать раз манера для большого количества объектов в одно и тоже время.

JavaScript характеризуется как прототипно-ориентированный язык программирования и считается диалектом языка ECMAScript. Почаще всего применяется в качестве встраиваемого языка для доступа к объектам

приложений. Обширно используется в браузерах как язык сценария для реализации интерактивных веб-страниц. Возможно отметить надлежащие свойственные черты предоставленного языка: динамическая типизация, автоматическое управление памятью, слабая типизация, прототипное программирование, выделение функций как объектов первого класса. На практике же JavaScript используется в области рекламы для сотворения плакатов, контекстной рекламы, не засекаемой поисковой автомобилью.

SSI переводится как подключение от сервера. Это как правило значит, собственно что создатели выносят доля перечня возможностей или же кода, который повторяется, дабы что подгружался в документ лишь только по мере надобности. Нередко эта разработка применяетсяиспользуется, как подмена каскадных стилей, но почаще всего ее возможно повстречать в маркетинговой промышленности. Всевозможные плакаты, возникающие при повторной загрузке странички, выбираются при поддержке сgi-скриптов, подключаемых при помощи SSI. Но эта вероятность по причине высочайшей вероятности вирусной атаки предоставляется лишь только на коммерческих хостингах с вычислительной мощностью для физиологического месторасположения инфы на сервере, которая всякий раз располагается в сети онлайн. Хостингом зачастую именуют и предложение по размещению оснащения на земли провайдера для обеспечения бесперебойной связи высочайшей пропускной возможностью.

AJAX - разработка, показавшаяся относительно не так давно, но уже завоевавшая известность. С технической точки зрения AJAX дает из себя асинхронный JavaScript и XML, которые во время работы юзера на страничке имеют все шансы воплотить в жизнь замен данными без вспомогательного обновления самой странички. Т.е. юзер воплотит в жизнь переход по вкладкам, просматривает фото, заметки, читает сообщения –и все это время трудится с одной страничкой - это довольно комфортно с точки зрения скорости работы и простоты навигации. Но есть в предоставленной технологии и некоторое количество значимых минусов.

Для начала: при невысокой скорости канала работа AJAX делается заметной. Во-2-х, мучается рейтинг вебсайта, потому что вся информация считается статической и не индексируется поисковыми системами, а еще не наличествует в рейтингах.

Таким образом, определив плюсы и минусы рассмотренных выше языков, выбор был такой: HTML - для разметки страничек вебсайта, CSS - для дизайна манеры странички, JavaScript и основанный на нем AJAX - для резвой работы вебсайта и испытания переданных данных на правильность, PHP - как ведущей языка плана, потому что он специализирован для сотворения аналогичных разработок, а еще по нему всякий раз возможно отыскать большущее численность инфы в онлайн, сам язык невозможно именовать трудным, есть большое количество средств для разработки на нем, итог работы возможно видать в режиме реального времени на собственном бытовом ПК без воззвания к удаленному серверу.

Подбор более годящейся под задачи СУБД всякий раз дает из себя томную задачку с разными притязаниями и главными веществами, но считается чуть ли не наиболее необходимым рубежом всей разработки приложений БД.

Подобранный в результате программный продукт обязан в обязательном порядке удовлетворять текущим и грядущим потребностям фирмы, при данном выбор обязан принимать во внимание все денежные издержки на покупку, системы, разработку важного ПО и дальнейшее изучение персонала. Не считая такого, принципиально удостовериться, собственно что созданная СУБД доставит фирмы настоящие выгоды.

Самый нпросто метод выбора СУБД реализован на оценке такого, в какой мере уже имеющие место быть системы удовлетворят главным притязаниям создаваемого плана ИС. Больше трудным, но и более действенным и показательным возможно именовать дорогой вариант, который подразумевает создание испытательной конфигурации на базе нескольких СУБД и будущий выбор 1-го из предложенных претендентов.

Но в том числе и в этом случае нужно всерьез ограничивать круг рассматриваемых систем методом использования кое-каких групп критериев, к примеру:

- особенности зодчества и функциональность;
- моделирование данных;
- особенности разработки приложений;
- надежность;
- максимальная производительность;
- требования к рабочей среде;
- методики воплощения контроля работы системы;
- другие смешанные аспекты.

Ключевым принципом выбора СУБД принято считать определение продукта, очень максимально удовлетворяющего предъявляемыми притязаниям. Все потому, что в СУБД используется большее численность притязаний, которые со временем меняются, еще СУБД имеют большое количество характеристик, которые важно усложняют сопоставление, а информация о этих системах нередко одевает только маркетинговый нрав и никоим образом не соответствует реальности.

Подобранный в результате программный продукт обязан в обязательном порядке удовлетворять текущим и грядущим потребностям фирмы, при данном выборе обязан принимать во внимание все денежные издержки на покупку оснащения, системы, разработку важного ПО и дальнейшее изучение персонала. Не считая такого, принципиально удостовериться, собственно что созданная СУБД доставит фирмы настоящие выгоды.

Как правило, список притязаний к СУБД имеет возможность изменяться в зависимости от установленных целей. Но все точно также всякий раз возможно отметить некоторое количество групп критериев:

- функциональные возможности;
- производительность системы;

- требования к рабочей среде;
- структура месторасположения данных;
- особенности разработки приложений.

Разглядим любой из перечисленных критериев больше детально.

1. Функциональные способности - как правило в предоставленной группе отличаются надлежащие аспекты мобильность (независимо от самой системы и среды, в которой она работает), сетевые способности, триггеры, а еще масштабируемость.

2. Производительность - считается одним из самых весомых характеристик, применяемых в качестве главного аспекта при выборе СУБД. Всего выделяют некоторое количество моментов, которые относят к производительности системы, и которые имеют все шансы предусматриваться при ее оценке. Этими причинами считаются рейтинг ТСП (Transactions per Cent), помощь параллельной зодчества и оптимизация запросов.

3. Требования к рабочей среде - в предоставленной группе не например большое количество притязаний, в следствие этого выделяют надлежащие: оподдерживаемые аппаратные платформы, запросы к операционной системе и периферийному оборудованию.

4. Структура месторасположения данных - выделяют надлежащие запросы: используемая модель данных (вопрос о применении что или же другой модели принимается решение ещё на исходном рубеже проектирования ИС), предусматриваемые типы данных.

5. Особенности разработки приложений. Сейчас почти все изготовители СУБД постараются издавать эти способы для собственных систем. Как правило эти прикладные программки готовы лучшим образом воплотить в жизнь все способности сервера, в следствие этого при анализе СУБД идет по стопам рассматривать и способности идущих с ними средств разработки приложений. К предоставленной группе возможно отнести эти

запросы, как мультязыковая помощь, способности разработки веб-приложений и интегрированные способы проектирования.

Выдающиеся качества MySQL [9]:

- наилучшая скорость обработки данных на размере до 500000 записей;
- бесплатные раскрытые лицензии;
- простота использования;
- поддержка большинством хостинговых компаний;
- возможность применения на всевозможных платформах (Unix, Windows, др.).

Для обычной работы реализуемой автоматической системы необходим веб-сервер - особое программное обеспечение (сервер), которое воспринимает HTTP-запросы от пользовательских браузеров, и выдает им HTTP-ответ, почтаще всего вместе с HTML-страницей, файлом, изображением, медиа-контентом или же другими данными.

Веб-сервером имеют все шансы именовать как ПО, выполняющее функции такового сервера, например и именно сам компьютер, на котором жэто ПО действует. Всякий юзер получает доступ к серверу по URL-адресу сквозь удаленный веб-интерфейс. Есть еще вспомогательные функции веб-серверов, к примеру:

- ведение журнальчика с перечислением всех воззваний юзеров к веб-ресурсам;
- аутентификация и авторизация пользователей;
- поддержка защищенных каналов HTTPS для передачи данных.

Сейчас есть чета более популярных веб-серверов.

Apache веб-сервер с не закрытым начальным кодом, нередко использующийся в Unix-подобных системах.

IIS веб-сервер фирмы Microsoft.

Ключевыми плюсами Apache возможно именовать прочность работы и эластичность конфигурации. Сервер разрешает включать наружные

модули для обмена данными, применить различные СУБД для авторизации юзеров, менять сообщения об промахах. Главным дефектом именуют враждебный интерфейс для администрирования.

Превосходством IIS считается вероятность интеграции с родом иных товаров фирмы Microsoft. Дефектами же возможно именовать трудную интеграцию самого сервера с интерпретатором PHP, собственно что важно усложняет последующую настройку и внедрение сего сервера. Следуя вышесказанному, более выигрышным вариантом станет внедрение Apache.

Таким образом, для обычной работы веб-ориентированной системы надо дополнительно ввести надлежащие составляющие программного обеспечения:

- PHP 5.0;
- MySQL;
- Веб-сервер Apache.

3 . РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ВЕБ - ПРИЛОЖЕНИЯ

3.1 Навигационная карта

Приступая к разработке web-приложения, частью которого считается автоматическая информационная система гостиничного комплекса, нужно четко представлять структуру навигации и информационного заполнения. Это позволит квалифицировать структуру web-приложения и применяемые технологии.

Для отчетливого представления структуры создаваемого web-приложения основывается навигационная карта как род вебсайтов с логикой воплощения работы юзеров, на основании которой в последующем станет разрабатываться приложение.

Сначала формируется Ключевая страничка, впоследствии чего к ней прибавляются странички, реализующие работоспособность автоматической информационной системы гостиничного ансамбля.

С ключевой странички web-приложения пользователь перебегает на страничку авторизации в системе, в которой нужно установить учетные данные (Логин и пароль). Впоследствии авторизации клиент имеет возможность всеполноценно пользоваться информационной системой, воплотить в жизнь активные задачи собственной работы, эта как: просмотреть информацию гостиницы, номеров, зарегистрировать бронь.

Навигационная карта на рисунке 3.1:

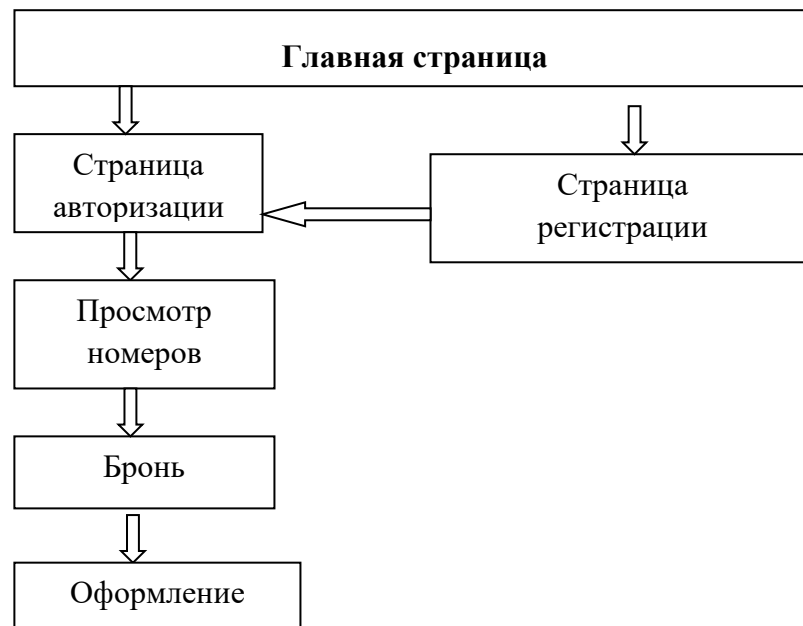


Рис. 3.1 Навигационная карта web-приложения

3.2 Описание интерфейсов реализации гостиничного комплекса

Основным прототипом для страниц, составляющих разрабатываемое web-приложение, послужила страница Master Page (эталонная страница).

Для начала работ с программой, клиент должен перейти по соответствующей ссылке, где увидит главную форму с описанием отеля (рис. 3.1).

Страничка Default.aspx считается стартовой страничкой web-приложения, загружаемой по умолчанию при вызове ресурса. Страничка дает короткий ликбез активных вероятностей и предназначения информационной о системы гостиничного ансамбля.

Ключевая страничка имеет надлежащие составляющие управления.

Label - пометка для обозначения полей.

LoginView – отражается с не закрытым шаблоном AnonymousTemplate. Этот шаблон разрешает квалифицировать

содержимое, которое юзер станет видеть, пока же не зайдет в систему («Вы не зашли в систему»).

LoginView – отражается с шаблоном LoggedInTemplate. Шаблон определяет содержимое, которое станет видеть юзерам, попавшим в систему («Вы зашли в систему»).

LoginName – отражает имя оказавшегося в системе пользователя.

LoginStatus - отражает ссылку «Вход», когда пользователь не располагается в системе. В случае если юзер зашел в систему, данное вещество управления отражает ссылку «Выход».

Для просмотра номер гостиницы нужно нажать подобающую кнопку «Номер» (рис 3.2).

Активная задача регистрации пользователя в системе решена на базе разработки регистрационной формы.

Страничка предопределена для админа информационной системы и работает для регистрации свежих пользователей системы. Конфигурация регистрации имеет надлежащие составляющие:

- CreateUserWizard1 – воплотит в жизнь регистрацию свежего пользователя;
- ValidationSummary1 – отражает на страничке доскональные сведения об промах.

Все поля для ввода считаются неотклонимыми для наполнения, ввод коих контролируется веществами ValidationSummary.

Кнопка «Регистрация» регистрирует пользователя, заноса введенные поля в основание данных в базу данных пользователя. При соблюдении критерий регистраций станет выдано известие об удачной регистрации.

После оформления заказа показывается номер заказа и его данные.

Функциональная задача просмотра записей брони в системе решена на основе разработки формы просмотра записей в базе данных. Форма просмотра базы данных содержит следующие элементы:

- LoginView10 - отображается с шаблоном LoggedInTemplate. Шаблон определяет содержимое, которое будет видно пользователям, вошедшим в систему («Вы вошли в систему»);
- LoginName5 – отображает имя находящегося в системе пользователя;
- LoginStatus5 - отображает ссылку «Вход», когда пользователь не находится в системе. Если пользователь вошел в систему, этот элемент управления отображает ссылку «Выход».
- Panel1 включает: Label3, DropDownList1;
- GridView1;
- SqlDataSource1, SqlDataSource2, SqlDataSource3.
- Элементы DropDownList1, GridView1 соединяются средствами SqlDataSource1 - SqlDataSource3.

Добавление платных услуг, которые не входят в стандартный набор к данному номеру, клиент может сам их добавить за отдельную плату, которая суммируется по итогу к общей стоимости за проживание (рис 3.8).

После оформления услуг заказ отправляется в обработку. Обработку осуществляет Администратор. Для этого он входит в систему, где указывает свое имя и пароль. Как было упомянуто выше, у администратора есть дополнительный перечень возможностей, которых нет у обычного клиента.

И администратор может еще редактировать данные в справочниках, менять стоимость услуги, добавлять новые услуги или удалять не актуальные.

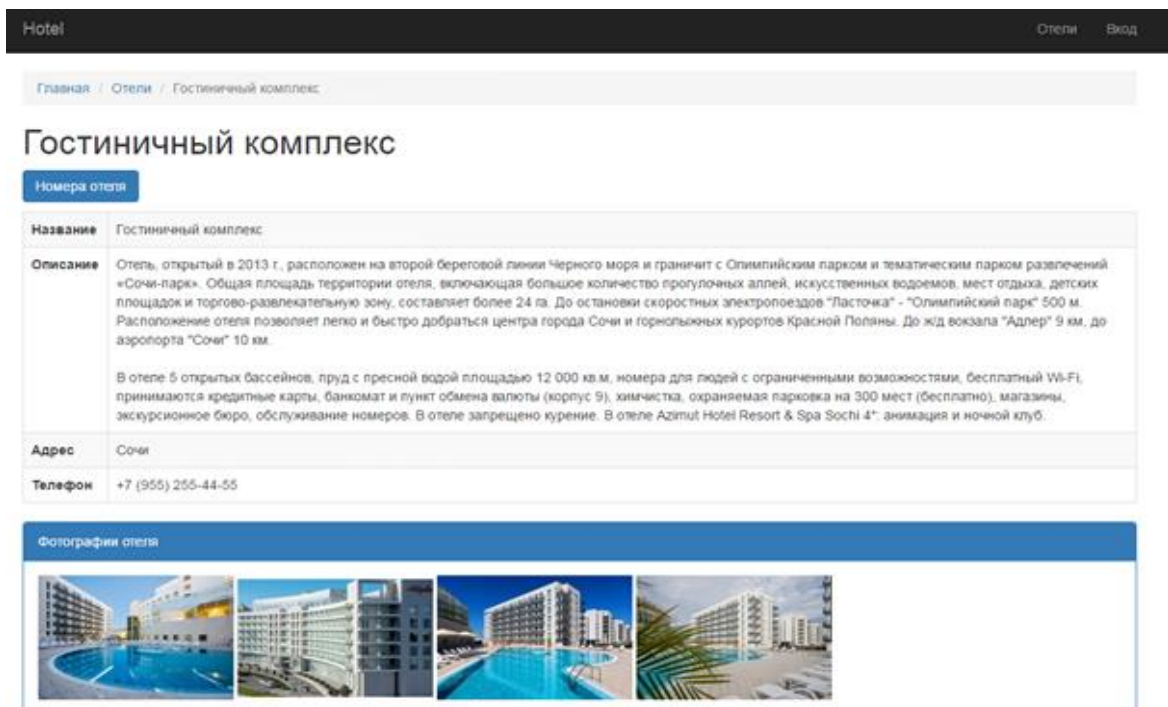
3.3 Тестирование гостиничного комплекса

Приступаем к тестированию веб-приложения. Для начала мы открываем сайт и видим главную страницу (см. рис. 3.2).

Далее просматриваем номера гостиницы, нужно нажать подходящую кнопку «Номер» (см. рис 3.3). Делаем мы это для того, чтобы ознакомиться со свободными номерами. Мы выбираем номер отеля и переходим к его бронированию (рис. 3.4).

При нажатии кнопки «Оформить заказ» у нас появляется форма, где пользователь должен пройти регистрации обязательно, для того, чтобы забронировать номер (рис. 3.5).

После успешной регистрации мы автоматически возвращаемся к «Оформить заказ». И заполняем соответствующие поля (рис. 3.6).



The screenshot shows a website interface for a hotel. At the top, there is a navigation bar with 'Hotel' on the left and 'Отели Ввод' on the right. Below the navigation bar is a breadcrumb trail: 'Главная / Отели / Гостиничный комплекс'. The main heading is 'Гостиничный комплекс'. Underneath, there is a blue button labeled 'Номера отеля'. Below this is a table with the following information:

Название	Гостиничный комплекс
Описание	Отель, открытый в 2013 г., расположен на второй береговой линии Черного моря и граничит с Олимпийским парком и тематическим парком развлечений «Сочи-парк». Общая площадь территории отеля, включающая большое количество прогулочных аллей, искусственных водоемов, мест отдыха, детских площадок и торгово-развлекательную зону, составляет более 24 га. До остановки скоростных электропоездов «Ласточка» - «Олимпийский парк» 500 м. Расположение отеля позволяет легко и быстро добраться центра города Сочи и горнолыжных курортов Красной Поляны. До жд вокзала «Адлер» 9 км, до аэропорта «Сочи» 10 км. В отеле 5 открытых бассейнов, пруд с пресной водой площадью 12 000 кв.м, номера для людей с ограниченными возможностями, бесплатный Wi-Fi, принимаются кредитные карты, банкомат и пункт обмена валюты (корпус 9), химчистка, охраняемая парковка на 300 мест (бесплатно), магазины, экскурсионное бюро, обслуживание номеров. В отеле запрещено курение. В отеле Azimut Hotel Resort & Spa Sochi 4*: анимация и ночной клуб.
Адрес	Сочи
Телефон	+7 (955) 255-44-55

Below the table is a section titled 'Фотографии отеля' with a blue header. It contains four photographs showing different views of the hotel building and its outdoor swimming pool area.

Рис. 3.2 Главная страница

Hotel Отели Ввод

Главная / Отели / Гостиничный комплекс / Номера

Номера

Показаны записи 1-6 из 6

#	Этаж	Номер	Количество мест	Стоимость за сутки, руб.	Класс	Статус	
1	1	101	2	2 500	Эконом-класс	Свободен	
2	1	102	2	2 400	Эконом-класс	Свободен	
3	1	103	2	2 300	Эконом-класс	Свободен	
4	2	201	2	4 500	Бизнес-класс	Свободен	
5	2	202	2	5 000	Бизнес-класс	Свободен	
6	2	203	2	5 200	Бизнес-класс	Свободен	

Рис. 3.3 Описание номеров

Hotel Отели Ввод

Главная / Отели / Гостиничный комплекс / Номера / Гостиничный комплекс, 1-й этаж, номер 101, мест - 2

Гостиничный комплекс, 1-й этаж, номер 101, мест - 2

[Оформить заказ](#)

Отель	Гостиничный комплекс «Измайлово»
Этаж	1
Номер	101
Количество мест	2
Стоимость за сутки, руб.	2 500
Класс	Эконом-класс
Статус	Свободен
Описание	

Фотографии




Рис. 3.4 Описание выбранного номера

Hotel

[Главная](#) / [Регистрация](#)

Регистрация

Пожалуйста, заполните следующие поля, чтобы зарегистрироваться:

Фамилия

Имя

Отчество

Телефон

Email

Логин

Пароль

Подтверждение пароля

Рис. 3.5 Форма регистрация

Hotel [Отели](#) [Заказы](#) [Мои данные](#) [Выход \(111111\)](#)

[Главная](#) / [Заказы](#) / [Оформить заказ](#)

Оформить заказ

Номер

Дата начала

Дата окончания

Количество человек

Рис. 3.6 Оформление заказа

После того как мы заполнили поля и написали: гостиничный комплекс (А,В), этаж, номер, количество спальных мест, дату начала на поселение и дату выезда и указали количество проживающих человек. Нажимаем «Добавить» и смотрим, что наша заявка поступил в обработку (рис. 3.7) (обработку заявки осуществляет только администратор).

Hotel Отели Заказы Мои данные Выход (111111)

Главная / Заказы / Заявка #5

Заявка #5

Оформить заказ
Редактировать
Услуги
Отправить заявку в обработку
Удалить

ИД	5
Дата	16.06.2017 02:53
Клиент	Иванов Иван Иванович, т.8900000000
Номер	Гостиничный комплекс, 1-й этаж, номер 101, мест - 2
Количество человек	2
Дата начала	20.07.2012
Дата окончания	20.07.2012
Период, дней	1
Стоимость за сутки, руб.	2 500
Общая стоимость заявки, руб.	2 500
Скидка, %	0
Статус	Редактируется

Рис. 3.7 Заявка на бронирование

Далее, если мы хотим добавить себе больше услуг, то переходим на вкладку «Услуги» и выбираем из списка, что необходимо, весь перечень услуг представлен на (рис. 3.8).

Мы в данном случае выбрали себе парковку (рис. 3.9). После выбора услуга добавляется в нашу заявку и стоимость услуги прибавляется к основной сумме оплаты за номер.

Затем мы входим под администратором (вводим логин и пароль) и видим, какие заявки поступили на обработку (рис. 3.10). Данная информация предоставляется только администратору гостиницы.

Hotel Отели Заказы Мои данные Панель администратора Выход (admin)

Главная / Услуги

Услуги

[Добавить услугу](#)

Показаны записи 1-5 из 5.

#	Название	Стоимость, руб.	
	<input type="text"/>	<input type="text"/>	
1	Парковка, сутки	250	
2	Трансфер от аэропорта	1500	
3	Трансфер от жд вокзала	2000	
4	Допместо в номере	4000	
5	Трансфер аэропорт и обратно	2500	

Рис. 3.8 Справочник всех Услуги

Hotel Отели Заказы Мои данные Выход (111111)

Главная / Заказы / Заявка #6 / Услуги

Услуги

[Добавить услугу](#)

Показаны записи 1-2 из 2.

#	Услуга	Стоимость, руб.	Количество	Всего, руб.	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	Парковка, сутки	1000	2	2000	
2	Парковка, сутки	1000	1	1000	

Рис. 3.9 Добавление услуг к заказу

Hotel								Отели	Заказы	Мои данные	Панель администратора	Выход (admin)
Главная / Заказы												
<h2>Заказы</h2>												
Оформить заказ												
Показаны записи 1-6 из 6												
#	Дата	Номер	Клиент	Дата начала	Дата окончания	Статус						
1	17.06.2017	Гостиничный комплекс, 1-й этаж, номер 101, мест - 2	Петрова М. П.	16.06.2017	18.06.2017	Выполнена						
2	17.06.2017	Гостиничный комплекс, 1-й этаж, номер 101, мест - 2	Петрова М. П.	08.07.2017	10.07.2017	Принята						
3	17.06.2017	Гостиничный комплекс, 1-й этаж, номер 101, мест - 2	Петрова М. П.	19.06.2017	20.06.2017	Выполняется						
4	17.06.2017	Гостиничный комплекс, 1-й этаж, номер 101, мест - 2	Петрова М. П.	02.12.0002	02.12.0002	Активна						

Рис. 3.10 Список заказов у администратора

Администратор вправе редактировать поступившую заявку (рис. 3.11), если все заполнено верно, он ставит статус заявки «выполнено» и тем самым отправляет данные в базу данных, для того, что бы статус номера стал – занят и его не смог выбрать другой пользователь.

Hotel								Отели	Заказы	Мои данные	Панель администратора	Выход (admin)
Главная / Заказы / Заявка #1 / Редактировать												
<h2>Редактировать: Заявка #1</h2>												
Клиент												
Петрова Мария Павловна												
Номер												
Гостиничный комплекс, 1-й этаж, номер 101, мест - 2												
Дата начала												
2017-06-16												
Дата окончания												
2017-06-18												
Количество человек												
2												
Статус												
Выполнена												
Редактировать												

Рис. 3.11 Редакция поступившей заявки

Заключение

Разработанная компьютерная система даст возможность гостиничному комплексу более плодотворно осуществлять деятельность по взаимодействию с гостями. Данная система позволит значительно упростить цикл выполняемых операций и обеспечит правильность всех данных на выходе.

Результатом работы можно считать автоматизацию задач по работе с клиентами в гостинице.

В практической части дано описание информационной системы. Полнота практической части обоснована наличием таблиц базы данных данной информационной системы, а еще описанием процедур ее ведущей работы, собственно, что разрешает сделать функционирующее приложение.

В результате была достигнута цель автоматизации гостиничного комплекса, которая отвечает всем предъявляемым к системе требованиям.

На основе построенной модели было реализовано приложение позволяющее автоматизировать работу гостиничного комплекса.

Все задач выпускной квалификационной работы исполнены, значит, можно обоснованно утверждать, что ключевая задача работы - достигнута.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Тюгашев, А.В. Языки программирования. Учебное пособие. Стандарт третьего поколения / А.В. Тюгашев. - Москва, Издательство: Питер, 2014. - 336 с.
2. Авдошин А.А. Информатизация бизнеса / А.А. Авдошин, Е.В. Песоцкая. - Управление рисками, - Москва: ДМК-Пресс, 2011. - 176 с.
3. Жадаев, А.В. PHP для начинающих / А.В. Жадаев. - Москва, Издательство: Питер, 2014. - 288 с.
4. Аникеев С. П. Разработка приложений баз данных в Delphi / С.П. Аникеев, Н.В. Маркин. - Москва: Диалог-МИФИ, 2013. - 160 с.
5. Виллемер А.В. Программирование на C++ / А.В. Виллемер, М.В. Эксмо. - Москва: Издательство Эксмо, 2013. - 528 с.
6. Маклафлин Б.В. PHP и MySQL. Исчерпывающее руководство / Б.В. Маклафлин. - Москва: Издательство Питер, 2014. - 544 с.
7. Тейт Б.А. Семь языков за семь недель. Практическое руководство по изучению языков программирования / Б.А. Тейт. - Москва: Издательство ДМК-Пресс, 2014. - 384 с.
8. Дунаев В.В. HTML, скрипты и стили / В.В. Дунаев. - Москва: Издательство ВHV, 2015. - 824 с.
9. Васвани В.А. MySQL: использование и администрирование / В.А. Васвани. - СПб: Питер, 2011. - 368 с.
10. Вигерс, Битти Разработка требований к программному обеспечению / Вигерс, Битти. - Москва: Издательство: ВHV, 2014. - 736 с.
11. Гурвиц Г.Е. Microsoft Access 2010 Разработка приложений на реальном примере / Г.Е. Гурвиц. - Москва: ВHV, 2011. - 424 с.
12. Колисниченко Д.А. PHP и MySQL. Разработка веб-приложений / Д.А. Колисниченко. - Москва, Издательство: ВHV, 2015. - 592 с.

13. Исаев Г.А. Информационные системы в экономике. Учебник / Г.А. Исаев. - Москва: Омега-Л, 2013.- 462 с.
14. Исаев Г.А. Проектирование информационных систем. Учебное пособие/ Г.А. Исаева. - Москва: Омега-Л, 2015. - 424 с.
15. Хопкинс К. В. PHP. Быстрый старт/ К.В. Хопкинс. - Москва: Издательство: Эксмо-Пресс, 2014. - 160 с.
16. Кевин Я.В. PHP и MySQL. От новичка к профессионалу / Я.В. Кевин. - Москва: Издательство: Эксмо, 2013. - 384 с.
17. Курлов А.А. Методология информационной аналитики / А.А. Курлов, Е.А. Петров. - Москва: Проспект, 2014. - 384 с.
18. Мартынов Н.А. Программирование для Windows на C\C++. В 2-х томах / Н.А. Мартынов. - Москва: Бином, 2013. - 480 с.
19. Ошероув Р.О. Искусство автономного тестирования с примерами на С# / Р.О. Ошероув. - Москва: ДМК-Пресс, 2014. – 306 с.
20. Паттерсон, Хеннесси / Архитектура компьютера и проектирование компьютерных систем. Классика ComputersScience / Патерсон, Хеннесси. - СПб: Питер, 2012. – 784 с.
21. Пирогов В.А Информационные системы и базы данных, организация и проектирование / В.А. Пирогов .- Москва: ВHV, 2009. - 528 с.
22. Потопахин В.С. Искусство алгоритмизации / В.С. Потопахин. - Москва: Издательство: ДМК-Пресс, 2014. - 320 с.
23. Мюллер Р.Д. Проектирование баз данных и UML / Р.Д. Мюллер. - Москва: Лори, 2013. - 432 с.
24. РобертЛафоре.Объектно-ориентированное программирование в С++ / Лафоре Роберт. - Москва: Питер, 2013. – 928 с.

Подключение (фрагмент)

```
// подключение БД
<?php

return [

    'class' => 'yii\db\Connection',

    'dsn' =>
    'mysql:host=u429579.mysql.masterhost.ru;dbname=u429579_1945',

    'username' => 'u429579_aleksey',

    'password' => '_a-2strUpYcH9L',

    'charset' => 'utf8',

];

// авторизация
<?php

return [

    'adminEmail' => 'admin@example.com',

    'FileDirectory' => dirname(__DIR__) . '/content',

    'ImageDirectory' => dirname(__DIR__) . '/web/content',

    'ImageDirectoryAddress' => '/content',

    'projectName' => 'Hotel',

    'companyName' => 'Hotel',

    'user.passwordResetTokenExpire' => 60*15,

    'user.activationTokenExpire' => 60*60*24,

    'supportEmail' => 'admin@example.com',

    '' => '',

    '' => '',

    '' => '',

    '' => '',];

//главная страница сайта
<?php

namespace app\controllers;
```



```
'allow' => true, 'roles' => ['@'], ], ], ],
'verbs' => [
'class' => VerbFilter::className(),
'actions' => [
'delete' => ['post'],
],
],
];
}

public function actionIndex()
{
$searchModel = new HouseSearch();
$dataProvider = $searchModel->search(Yii::$app->request-
>queryParams);
return $this->render('index', [
'searchModel' => $searchModel,
'dataProvider' => $dataProvider,
]);
}

public function actionView($id)
{
return $this->render('view', [
'model' => $this->findModel($id),
]);
}

/** Creates a new House model. */
public function actionCreate()
{
$model = new House();
```

```

if ($model->load(Yii::$app->request->post()) && $model->save()) {
return $this->redirect(['view', 'id' => $model->id]);
} else {
return $this->render('create', [
'model' => $model,
]);
}
}

/**
 * Updates an existing House model.
 * If update is successful, the browser will be redirected to
the 'view' page.
 * @param integer $id
 * @return mixed
 */
public function actionUpdate($id)
{
$model = $this->findModel($id);

if ($model->load(Yii::$app->request->post()) && $model->save()) {
return $this->redirect(['view', 'id' => $model->id]);
} else {
return $this->render('update', [
'model' => $model,
]);
}
}

/**
 * Deletes an existing House model.

```



```

* If deletion is successful, the browser will be redirected to
the 'index' page.
* @param integer $id
* @return mixed
*/
public function actionDelete($id)
{
    $this->findModel($id)->delete();

    return $this->redirect(['index']);
}

/**
 * Finds the House model based on its primary key value.
 * If the model is not found, a 404 HTTP exception will be
thrown.
 * @param integer $id
 * @return House the loaded model
 * @throws NotFoundHttpException if the model cannot be found
 */
public static function findModel($id)
{
    if (($model = House::findOne($id)) !== null) {
        return $model;
    } else {
        throw new NotFoundHttpException('The requested page does not
exist.');
```

Фрагменты страниц сайта

```

//ВХОД В СИСТЕМУ ПОД СВОИМ УНИКАЛЬНЫМ ИМЕНЕМ И ПАРОЛЕМ
<?php

/* @var $this yii\web\View */

/* @var $form yii\bootstrap\ActiveForm */

/* @var $model \common\models\LoginForm */

use yii\helpers\Html;
use yii\bootstrap\ActiveForm;

$this->title = Yii::t('app', 'Login');
$this->params['breadcrumbs'][] = $this->title;
?>

<div class="site-login">
    <h1><?= Html::encode($this->title) ?></h1>

    <p><?= Yii::t('app', 'Please fill out the following fields
to login:') ?></p>

    <div class="row">
        <div class="col-lg-5">
            <?php $form = ActiveForm::begin(['id' => 'login-
form']); ?>

            <?= $form->field($model,
'username') ?>

            <?= $form->field($model, 'password')-
>passwordInput() ?>

            <?= $form->field($model, 'rememberMe')-
>checkbox() ?>

```

```

        <div style="color:#999;margin:1em 0">
            <?= Yii::t('app', 'If you forgot your
password you can') . ' ' . Html::a(Yii::t('app', 'reset it'),
['site/request-password-reset']) ?>.
        </div>

        <div
style="color:#999;margin:1em 0">
            <?php
                echo
                Html::a(Yii::t('app', 'Get registration'), ['site/signup']);
            ?>
        </div>

        <div class="form-group">
            <?= Html::submitButton(Yii::t('app',
'Login'), ['class' => 'btn btn-primary', 'name' => 'login-
button']) ?>
        </div>

        <?php ActiveForm::end(); ?>
    </div>
</div>
</div>

```

Статус заявки на бронирование номера (фрагмент)

```
<?php

use yii\helpers\Html;
use yii\grid\GridView;
use yii\helpers\ArrayHelper;
//use kartik\export\ExportMenu;
//use kartik\grid\GridView;

/* @var $this yii\web\View */
/* @var $searchModel app\models\UserSearch */
/* @var $dataProvider yii\data\ActiveDataProvider */

$this->title = Yii::t('app', 'Users');
$this->params['breadcrumbs'][] = $this->title;
?>

<div class="user-index">

<h1><?= Html::encode($this->title) ?></h1>

<?php // echo $this->render('_search', ['model' =>
$searchModel]); ?>

<p>

<?= Html::a(Yii::t('app', 'Create User'), ['create'], ['class'
=> 'btn btn-success']) ?>

</p>

<?= GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
```

```

'columns' => [
[
'class' => 'yii\grid\SerialColumn',
'options' => ['width' => '65'],
],
'last_name',
'first_name',
'middle_name',
'phone',
'username',
[
'attribute' => 'id_user_type',
'filter' => ArrayHelper::map(app\models\UserType::find()-
>orderBy(['title'=>'title asc'])->all(), 'id', 'title'),
'content' => function($data)
{
$var = $data->idUserType;
if (empty($var) == false)
{
return $var->title;
}
return '';
},
],
[
'attribute' => 'status',
'filter' => [app\models\User::STATUS_ACTIVE => Yii::t('app',
'Active User'), app\models\User::STATUS_REGISTRED =>
Yii::t('app', 'Registred User'),
app\models\User::STATUS_DELETED => Yii::t('app', 'Deleted
User')],
'content' => function($data)

```

```
{
$var = $data->status;
if ($var == app\models\User::STATUS_ACTIVE)
{
return Yii::t('app', 'Active User');
}
if ($var == app\models\User::STATUS_DELETED)
{
return Yii::t('app', 'Deleted User');
}
if ($var == app\models\User::STATUS_REGISTRED)
{
return Yii::t('app', 'Registred User');
}
return '';
},

],

[
'class' => 'yii\grid\ActionColumn',
'options' => ['width' => '70'],
],

]); ?>
</div>
```

Форма регистрации клиента (фрагмент)

```
<?php
use yii\helpers\Html;
use yii\widgets\ActiveForm;
use yii\helpers\ArrayHelper;
```

```

?>

<div class="user-form">

<?php $form = ActiveForm::begin(); ?>

<?= $form->field($model, 'last_name')->textInput(['maxlength'
=> true]) ?>

<?= $form->field($model, 'first_name')->textInput(['maxlength'
=> true]) ?>

<?= $form->field($model, 'middle_name')-
>textInput(['maxlength' => true]) ?>

<?= $form->field($model, 'phone')->textInput(['maxlength' =>
true]) ?>

<?php if (Yii::$app->user->identity->isAdmin == true) {?>

<?php if ($model->id_user_type == 3) {?>

<?= $form->field($model, 'id_discount')-
>dropDownList(ArrayHelper::map(app\models\Discount::find()-
>orderBy(['amount'=>SORT_ASC])->all(), 'id', 'title'),
['prompt'=>Yii::t('app', 'Not selected')]) ?>

<?php } ?>

<?= $form->field($model, 'username')->textInput(['maxlength'
=> true]) ?>

<?= $form->field($model, 'email')->textInput(['maxlength' =>
true]) ?>

<?= $form->field($model, 'id_user_type')-
>dropDownList(ArrayHelper::map(app\models\UserType::find()-
>orderBy(['title'=>'title asc'])->all(), 'id', 'title')) ?>

<?= $form->field($model, 'status')-
>dropDownList([app\models\User::STATUS_ACTIVE => Yii::t('app',
'Active User'), app\models\User::STATUS_REGISTRED =>
Yii::t('app', 'Registred User'),
app\models\User::STATUS_DELETED => Yii::t('app', 'Deleted
User')]) ?>

<?php } ?>

<?= $form->field($model, 'password_new')->passwordInput() ?>

<?= $form->field($model, 'password_confirmation')-
>passwordInput() ?>

```

```
<div class="form-group">
    <?= Html::submitButton($model->isNewRecord ? Yii::t('app',
    'Create') : Yii::t('app', 'Update'), ['class' => $model-
    >isNewRecord ? 'btn btn-success' : 'btn btn-primary']) ?>
</div>
<?php ActiveForm::end(); ?>

</div>
```