

Содержание

Введение	3
1 Формирование требований к веб-ресурсу	4
1.1 Отчет об обследовании	4
1.2 Проблемы заказчика	5
1.3 Требования к веб-ресурсу	6
2 Разработка концепции веб-ресурса	8
2.1 Изучение объекта автоматизации	8
2.2 Построение модели предметной области	9
2.3 Варианты концепции веб-ресурса	11
3 Разработка технического проекта	14
3.1 Схема функциональной структуры веб-ресурса	14
3.2 Описание постановки задачи	19
3.3 Логическая модель данных	21
3.4 Обоснование выбора средств разработки	25
3.5 Алгоритм программного приложения	27
4 Разработка рабочего проекта	31
4.1 Разработка программно-информационного компонента	31
4.2 Разработка эксплуатационной документации	39
5 Техника безопасности при работе с веб-ресурсом	45
Заключение	46
Список использованных источников	47
Приложение А Техническое задание. Требования к системе	49
Приложение Б Программный код модулей веб-ресурса	55
Приложение В Тест-кейсы	65

					ККЭП 09.02.07 0087 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Разработка веб-ресурса для бронирования отелей «Holiday Ноте» Пояснительная записка	Лит.	Лист	Листов
Разраб.	Мельникова					ДП	2	67
Провер.	Шостак							
Рецензент	Головко							
Н.контр.	Евтушенко							
Утв.	Головко							
						Гр. 3-Д9-4ИНС		

Введение

Онлайн-площадки для бронирования одинаково полезны как для туристов, так и для гостиниц. Первым они помогают подобрать подходящее жилье, вторым обеспечивают рекламу, увеличивают объемы продаж.

Тренды и тенденции развития IT-рынка данной сферы постоянно меняются, поскольку технологии и пожелания пользователей развиваются и эволюционируют. Но одним из главных условий, которым должен соответствовать информационный продукт неизменны – это безопасность данных и удобство использования.

С уходом зарубежных платформ, которые ранее также и в России пользовались большой популярностью появилась необходимость создания новых технологий, выигрывающих в качестве и удобстве использования.

Сайты бронирования отелей остаются актуальными и популярными в современном мире, поскольку они предлагают ряд преимуществ для пользователей и отрасли гостеприимства.

Основная задача выпускной квалификационной работы включает в себя разработку веб-ресурса «Holiday Home», который позволит упростить процесс поиска и бронирования жилья для отдыха туристам по всей России.

1 Формирование требований к веб-ресурсу

1.1 Отчет об обследовании

Заданием предусмотрена разработка веб-ресурса бронирования и регистрации отелей «Holiday Home».

«Holiday Home» предназначена для автоматизации процессов поиска и бронирования отелей, на основании их компетентности и оценки клиентов, также веб-ресурс должен способствовать развитию новых отелей.

Основными целями создания «Holiday Home» являются:

- сбор, анализ и использование больших объемов данных для принятия обоснованных решений о стратегии продаж, определения трендов и выявления возможностей для роста;
- использование облачных решений для оптимизации операционных процессов, снижения затрат на ИТ-инфраструктуру и упрощения масштабирования бизнеса;
- удобный интерфейс сайта. Помогает максимально быстро найти то, что нужно, даже неопытному пользователю.

Данный веб-ресурс предназначен для выполнения всех следующих задач:

- предоставить пользователям возможность удобного поиска отелей на основании сравнения цен, услуг, наличия номеров и отзывы клиентов в одном месте, что позволяет сэкономить время и усилия;
- предложить гибкие условия отмены и изменения бронирования, что делает процесс более удобным для клиентов, особенно в условиях неопределенности и изменяющихся планов;
- облегчить бронирование отелей в других странах и регионах, предоставляя информацию на разных языках и поддерживая множество валют;
- использовать данные о клиентах для предоставления индивидуальных предложений, рекомендаций и маркетинговых кампаний, что позволяет увеличить

конверсию и удержание клиентов.

При разработке веб-ресурса «Holiday Home» учитываются пожелания заказчика.

1.2 Проблемы заказчика

Заказчики сайтов бронирования отелей также могут столкнуться с рядом проблем и вызовов в процессе разработки, внедрения и управления своими платформами. Вот некоторые из них:

- высокая конкуренция. Для заказчиков необходимо выделиться среди конкурентов и привлечь внимание пользователей;
- технические проблемы и поддержка. Веб-ресурс должен обеспечивать надежность, безопасность и высокую доступность своих платформ, что может потребовать значительных ресурсов на техническую поддержку и обслуживание;
- привлечение и удержание пользователей. Заказчикам необходимо постоянно работать над улучшением пользовательского опыта, предоставлением персонализированных предложений и развитием маркетинговых стратегий для привлечения новых клиентов и удержания существующих;
- соблюдение законодательства и нормативных требований. Это может включать в себя защиту данных, раскрытие информации о ценах и сборах, правила отмены бронирований и многое другое;
- ответственность за проблемы с бронированием.

В первую очередь заказчик разрабатывает веб-ресурс, нацеленный для работы клиентов, как туристов, так и отельеров и должен соответствовать их следующим требованиям:

- предоставляет данные о большом количестве средств размещения на одном сайте;
- структурирует информацию об отелях;

- выдает варианты размещения под конкретные запросы;
- упрощает процесс поиска и бронирования – всё происходит на одной площадке;
- гарантирует относительную безопасность в аренде жилья;
- веб-ресурс для бронирования выступает в качестве рекламной площадки для аудитории несравнимо большей, чем та, что приходит на сайты гостиниц;
- удобно привлекать гостей, когда отель небольшой и у него нет своего сайта;
- сервис играет роль посредника между гостем и отелем в случае недопониманий.

Описанные выше требования формируют полное представление о разрабатываемом веб-ресурсе и включают в себя запросы всех участников, которые в дальнейшем будут её эксплуатировать.

1.3 Требования к веб-ресурсу

Чтобы создать успешный веб-ресурс для бронирования отелей, необходимо учесть ряд ключевых требований и функций:

- интуитивный и привлекательный дизайн. Веб-ресурс должен иметь чистый, привлекательный и интуитивно понятный дизайн, обеспечивающий удобство использования и удовлетворение потребностей пользователей;
- мобильная адаптивность. С учетом растущего использования мобильных устройств для бронирования отелей, веб-ресурс должен быть адаптирован для корректного отображения и работы на различных экранах и платформах;
- поиск и фильтрация. Веб-ресурс должен предоставлять продвинутые функции поиска и фильтрации, позволяющие пользователям легко находить отели по местоположению, цене, категории, наличию определенных удобств и другим параметрам;

– информация об отелях. Веб-ресурс должен предоставлять детальную и актуальную информацию об отелях, включая фотографии, описания, список услуг и удобств, политику отмены и отзывы клиентов;

– прозрачность цен. Веб-ресурс должен четко и наглядно отображать цены, чтобы пользователи могли легко сравнивать предложения и принимать обоснованные решения;

– безопасность и конфиденциальность данных. Веб-ресурс должен обеспечивать высокий уровень безопасности и защиты личных и платежных данных пользователей, в соответствии с требованиями законодательства и стандартов индустрии;

– гибкость и удобство процесса бронирования. Веб-ресурс должен обеспечивать простой и быстрый процесс бронирования, с возможностью изменения или отмены бронирования, если это необходимо;

– поддержка пользователей. Веб-ресурс должен предоставлять качественную поддержку пользователей, доступную через различные каналы (например, электронная почта, телефон, онлайн-чат), чтобы решать возникающие вопросы и проблемы своевременно и эффективно.

Учитывая все вышеуказанные требования, создание успешного веб-ресурса для бронирования отелей может быть сложным, но важным процессом, который поможет обеспечить удобство и удовлетворение потребностей пользователей, а также конкурентоспособность на рынке.

2 Разработка концепции веб-ресурса

2.1 Изучение объекта автоматизации

Заказчиком веб-ресурса является российский бизнесмен директором компании ООО «ОТЕЛЬ Сервис» Смиренко Иваном Михайловичем, который хочет расширить сферу туризма по регионам России, путём разработки удобной оточенной площадки бронирования отелей и их продвижения в сети Интернет на основе собственных пожеланий.

Веб-ресурс «Holiday Home» обязан предоставлять полную и актуальную информацию о жилищных условиях, которые первоначально регистрирует Администратор веб-ресурса.

Жилье, которое может забронировать клиент делится на категории:

- гостиница;
- апарт-отель;
- спа-отель;
- хостел;
- мини-отель;
- мотель;
- другое.

В категорию «другое» включаются те отели, которые не вошли ни в одну из вышеперечисленных.

Отели так же могут быть разного уровня комфортности, который определяется звездами от 1 до 5. Каждый отель предоставляет свои собственные дополнительные услуги, которые делятся в пределах веб-ресурса на: проживание с животными, завтрак, обед, ужин, бесплатные напитки в течение дня. Все оставшиеся уникальные услуги отелей делятся на: трансфер до аэропорта, трансфер до вокзала, билеты на культурные мероприятия, билеты канатные дороги, билеты на аттракционы и другое, которые уточняются по приезду клиента. Номера отелей

делятся по количеству спальных мест (1-х местный, 2-х местный, 3-х местный, 4-х местный, 5-ти и более), включает в себя дополнительную технику (телевизор, чайник, мини-бар, кухня, кулер с водой на этаже, Wi-Fi), также номеров могут иметь свои категории: стандартный, люкс, семейный, экономный.

Для регистрации нового отеля в веб-ресурсе представитель хостеса в любом лице, то есть директор, менеджер, администратор заполняет заявку на его добавление, которая в дальнейшем проверяется администратором веб-ресурса, проходя модификацию и публикуется. В дальнейшем информация об номерах отелей, его ценах и другая часто изменяемая информация формируется или редактируется представителем хостеса, в веб-ресурсе он идентифицируется как менеджер отеля.

Не зарегистрированный ранее в веб-ресурсе клиент принадлежит роли гость и имеет возможность просматривать отели, выбрать несколько понравившихся, добавив во вкладку избранное для дальнейшего сравнения и выбора лучшего для себя варианта. Любой пользователь, которого интересует жилье имеет возможность связаться с менеджером определенного отеля по указанным данным. Каждый гость для дальнейшего бронирования жилья может самостоятельно пройти регистрацию.

Авторизованный пользователь веб-ресурса, который ранее успешно забронировал отель может просматривать историю собственных бронирований, в зависимости от статуса их выполнения. Также пользователь имеет возможность добавлять понравившиеся отели в избранное, тем самым обеспечивая к ним быстрый доступ при дальнейшем бронировании.

2.2 Построение модели предметной области

Веб-ресурс бронирования отелей для отдыхающих требуется следующим заинтересованным лицам:

– гость;

- клиент;
- администратор веб-ресурса;
- хостес.

Соответственно основные прецеденты (варианты использования) для разрабатываемого веб-ресурса следующие.

Основные прецедент для гостя, он же является неавторизованным клиентом веб-ресурса:

- П1 – регистрация в веб-ресурсе;
- П2 – просмотр отелей.

Основные прецедент для клиента:

- П3 – авторизация в веб-ресурсе;
- П4 – добавление отеля в «избранное»;
- П5 – оформление бронирования жилья.

Основные прецедент для администратора веб-ресурса:

- П3 – авторизация в веб-ресурсе;
- П6 – просмотр историй входа;
- П7 – рассмотрение заявок на регистрацию отеля;
- П8 – ведение данных о пользователях веб-ресурса.

Основные прецедент для менеджера отеля:

- П3 – авторизация в веб-ресурсе;
- П9 – оформление заявки на регистрацию отеля;
- П10 – ведение данных о бронировании;
- П11 – ведение данных об отелях;

Представим диаграмму вариантов использования, созданную средствами MS Visio 2019 (профессиональная версия), для проектируемого программного обеспечения на рисунке 1.

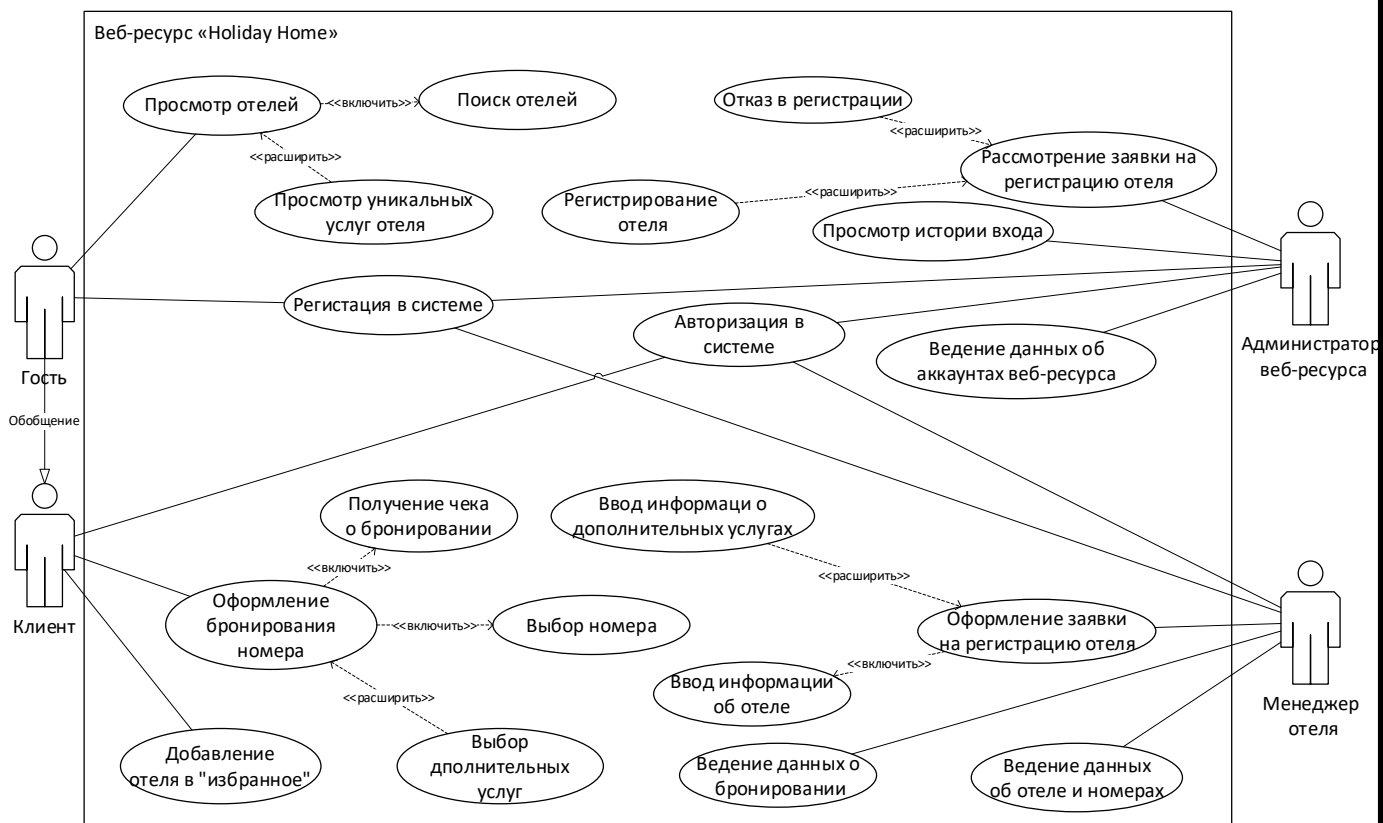


Рисунок 1 – Диаграмма вариантов использования веб-ресурса
«Holiday Home»

Предоставленная диаграмма чётко описывает в себе все прецеденты предметной области для требуемого веб-ресурса.

2.3 Варианты концепции веб-ресурса

На основе глубокого анализа инжиниринга деятельности предприятия, отраженного в разработанных бизнес-моделях предметной области автоматизации, были разработаны концептуальные модели, которые представляют основные сферы работы веб-ресурса.

Опишем варианты концепций для прецедента «Формирование заявки на регистрацию отеля» с помощью диаграммы деятельности, представленных на рисунках 2 и 3.

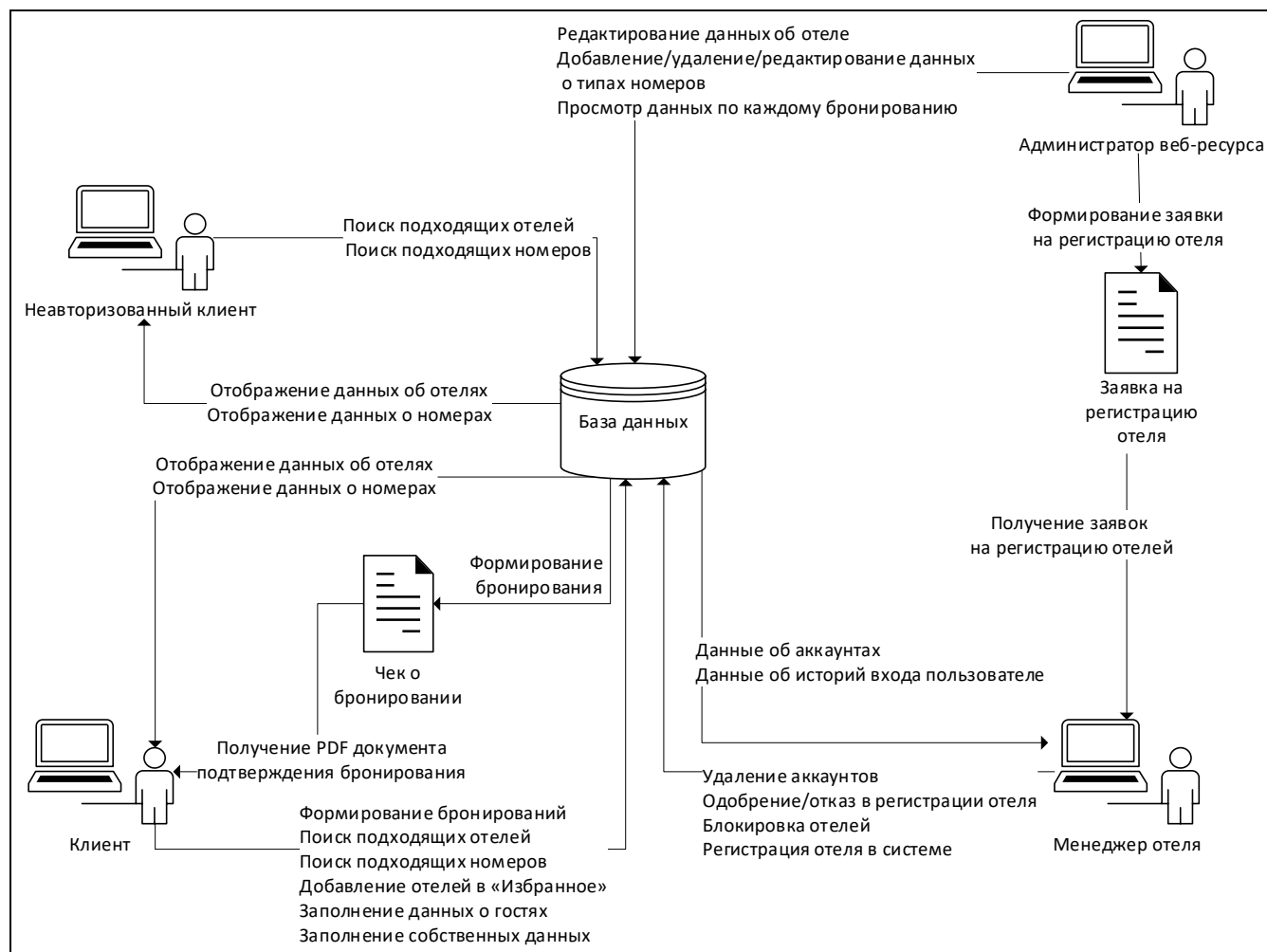


Рисунок 2 – Вторая концептуальная модель веб-ресурса «Holiday Home»

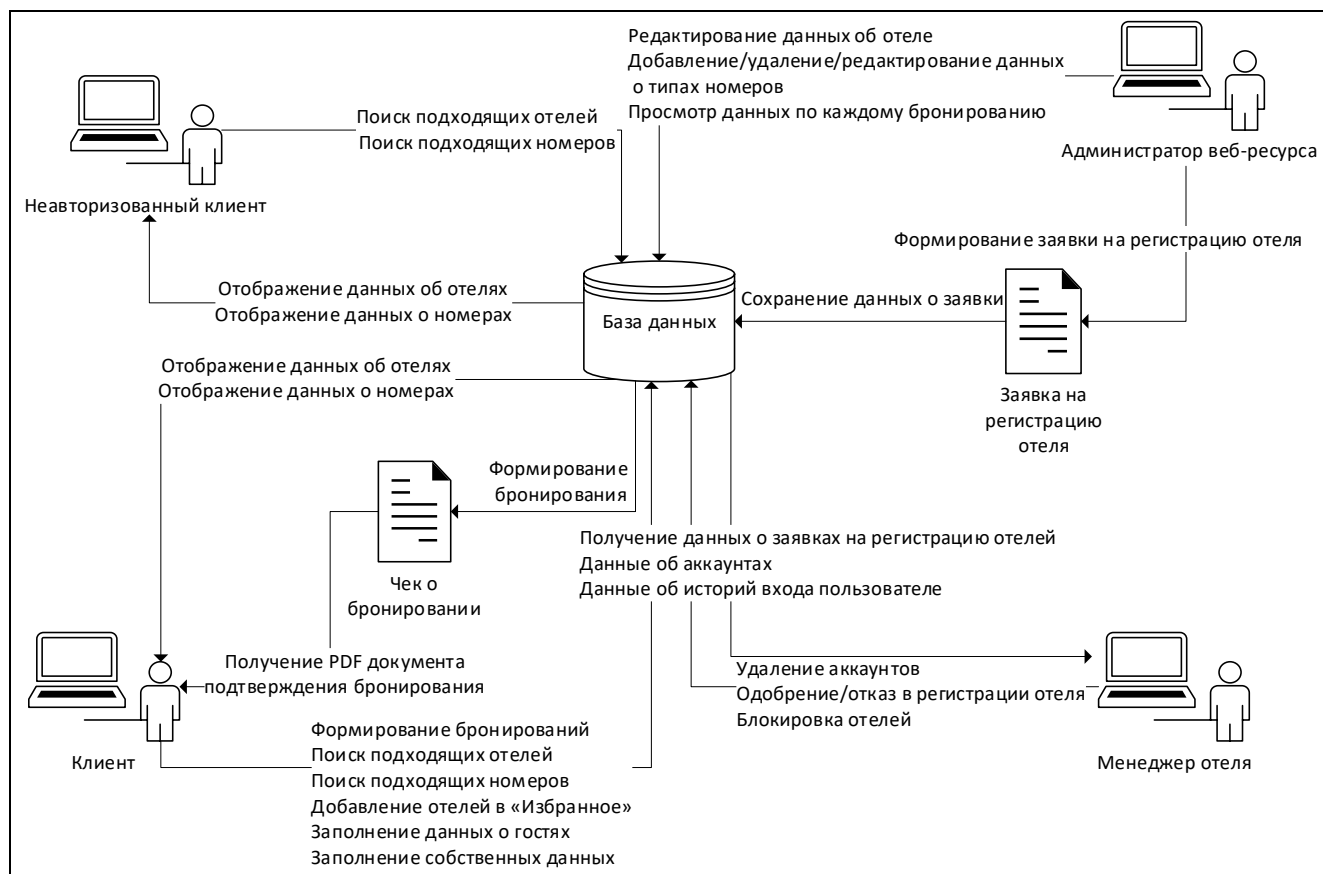


Рисунок 3 – Первая концептуальная модель веб-ресурса «Holiday Home»

Основным различием двух концептуальных моделей является хранение и тип формирования заявки регистрации отеля. Во второй модели сформированная заявка менеджером отеля сразу сохраняется в базу данных, к которой администратор веб-ресурса имеет быстрый доступ. Первая непосредственно отправляется администратору системы, который заносит данные с заявки в базу данных.

Из выше указанных особенностей моделей было принято решение использовать вторую модель, так как она позволяет быстрее регистрировать отели в системе и тем самым увеличивает производительность администраторов проверки заявок, исключая ошибок.

При выборе концепций для веб-ресурса учитывались технические возможности и возможность выполнения «Holiday Home» предоставленным требованиям.

3 Разработка технического проекта

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0087 ПЗ

Лист

13

3.1 Схема функциональной структуры веб-ресурса

Для разработки веб-ресурса «Holiday Home» приведенная схему функциональной структуры согласно требованиям РД50-34.698-90 (Автоматизированные системы. Требования к содержанию документов), в которой отображены элементы функционала, информационные связи между элементами и с внешней средой с кратким указанием содержания сообщений. Схема функциональной структуры веб ресурса разработана по нотации IDEF0, представленная на рисунке 4.

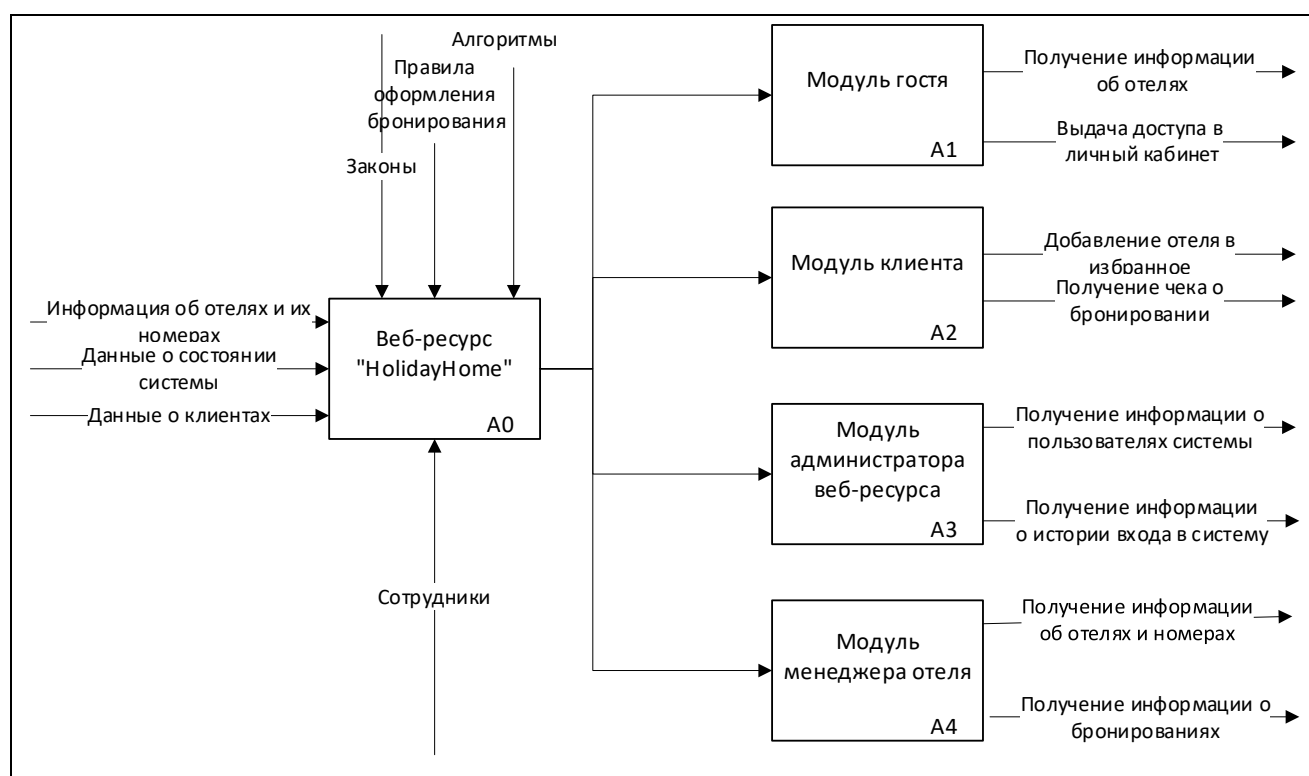


Рисунок 4 – Схема функциональной структуры веб-ресурса «Holiday Home»

Более подробная детализация элементов каждого модуля веб-ресурса «Holiday Home» представлены на рисунках 5 – 8.

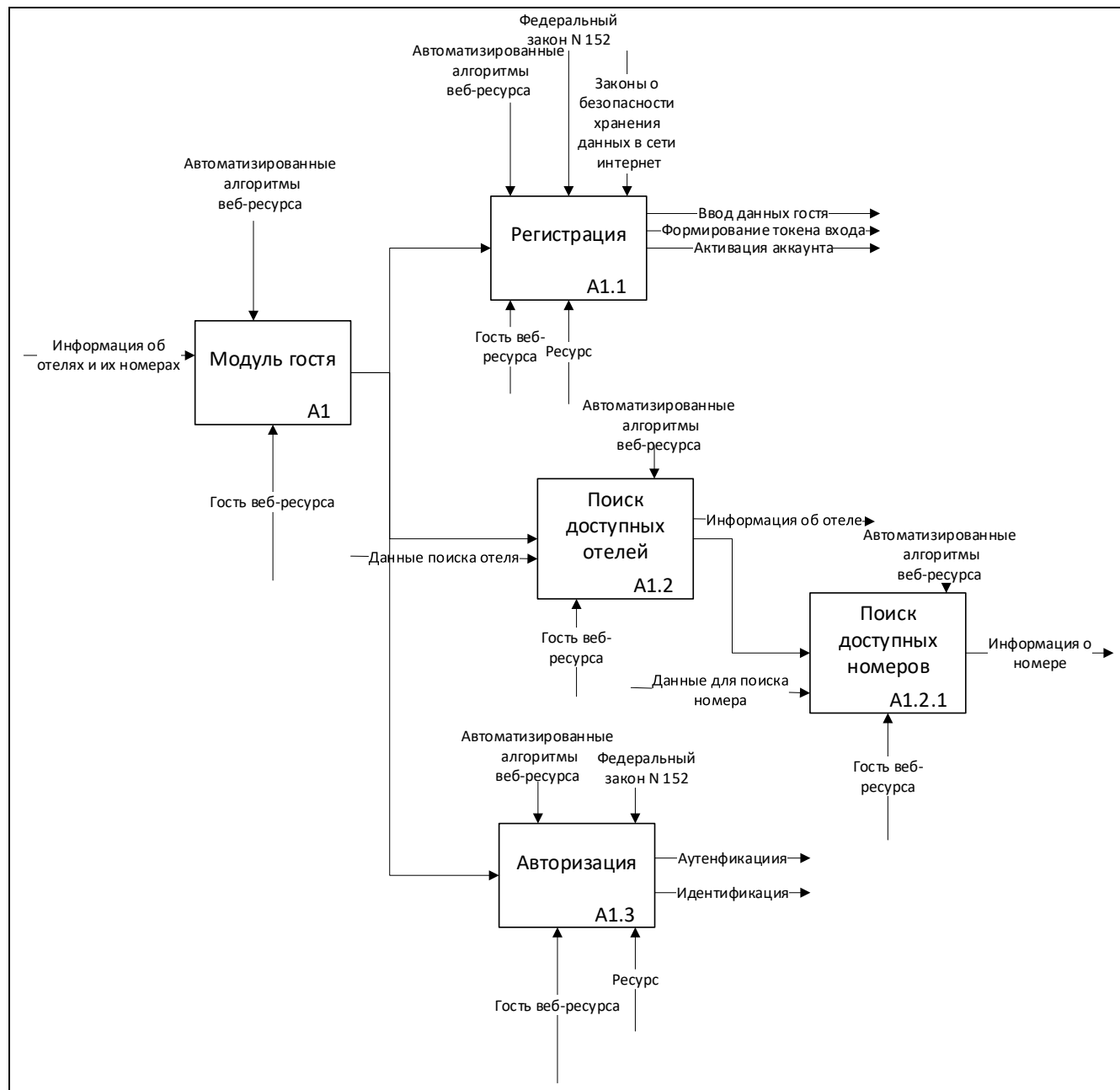


Рисунок 5 – Декомпозиция модуля гостя

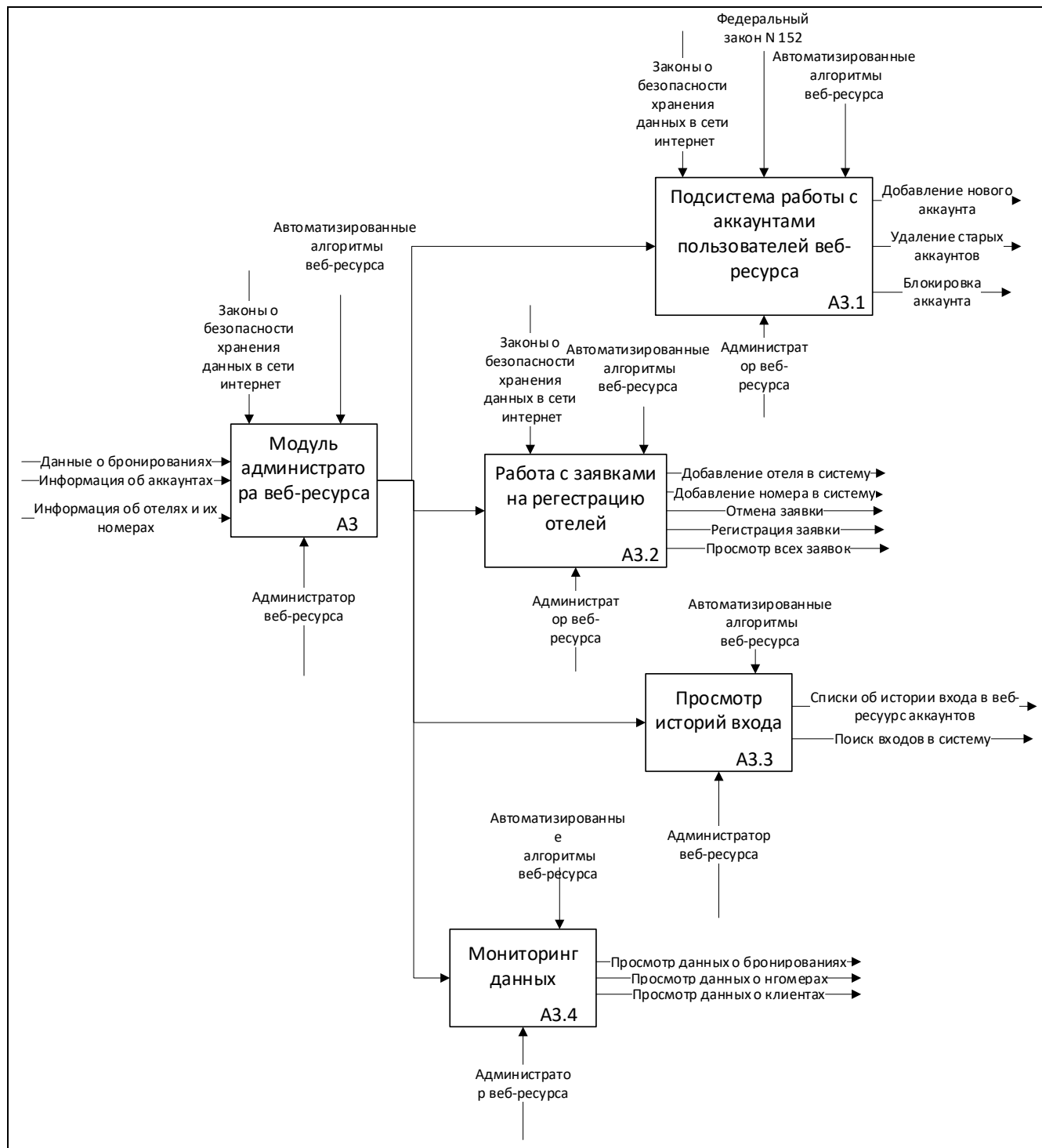


Рисунок 7 – Декомпозиция модуля администратора веб-ресурса

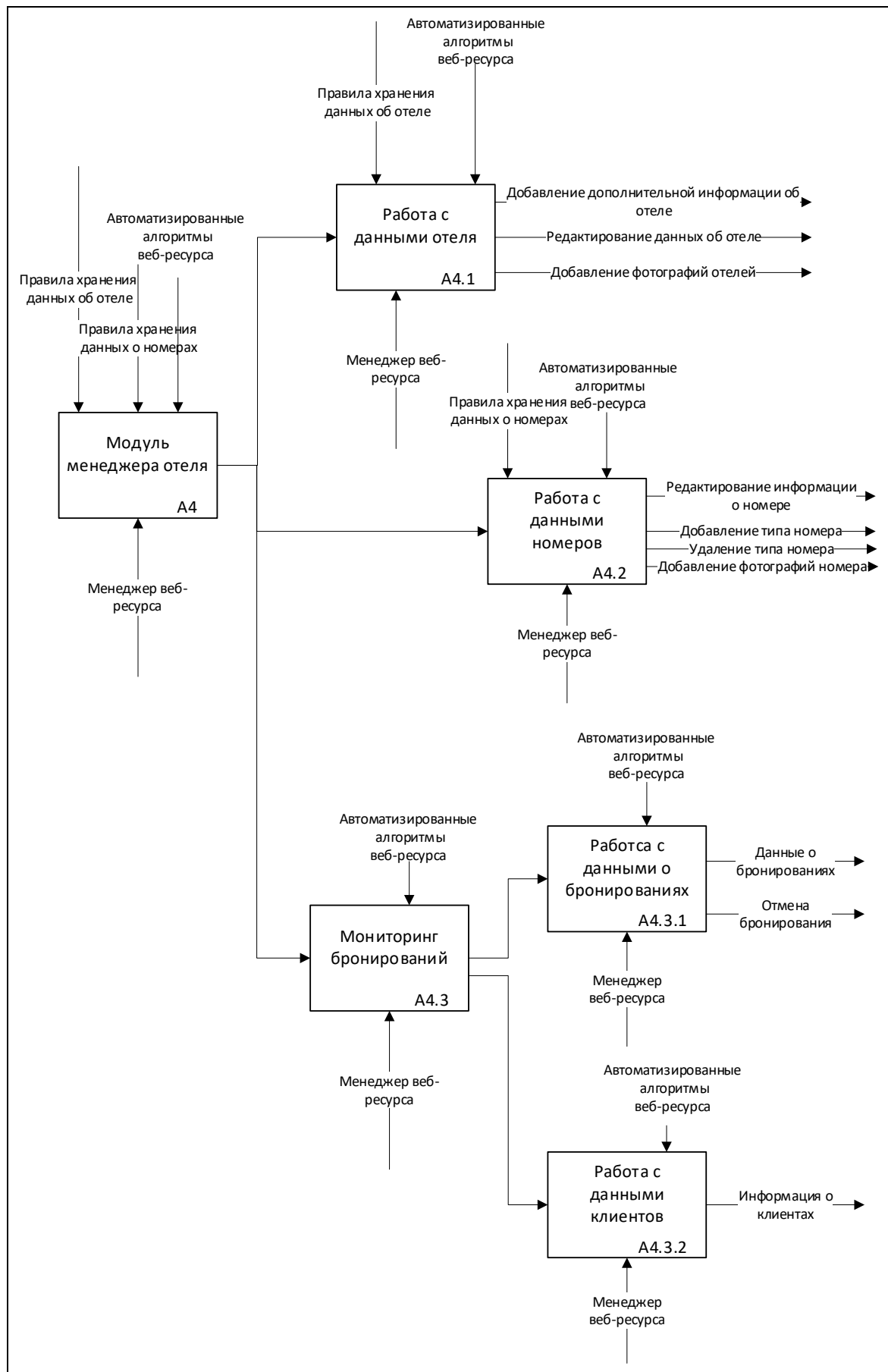


Рисунок 8 – Декомпозиция модуля менеджера отеля

Благодаря декомпозиции можно более детально рассмотреть функции разрабатываемого веб-ресурса. На основе представленных схем, можно определить, что функционал менеджеров отелей и администратора самые объёмные и требуют больше всего функционала, нацеленного на работу с данными, а именно добавление, удаление и редактирование.

Функционал модулей гостя и клиента, заключается в поиске подходящего номера поэтапным методом путём добавления необходимых данных. Клиентский модуль имеет так же личный кабинет и может проводить бронирования и отслеживать их.

3.2 Описание постановки задачи

Для решения поставленных задач необходимо разработать веб-ресурс, выполняющий регулярный мониторинг данных об отелях и бронированиях, тем самым обеспечивающий пользователям веб-ресурса возможность быстрого выбора и анализа на основе представления актуальной информации и корректных данных.

Веб-ресурс будет включать в себя 4 функциональных модуля:

- модуль гостя;
- модуль клиента;
- модуль администратора веб-ресурса;
- модуль менеджера отеля.

Каждый из модулей содержит основные страницы:

- страница «Главная»;
- страница «Авторизация»;
- страница «Восстановление пароля».

Модуль гостя содержит страницы

- страница «Просмотр отелей» – содержит краткую информацию о доступных отелях;

– страница «Просмотр номеров» – содержит полную информацию об отеле и номерах.

Данные страницы содержат поиск бронирований по общим характеристикам (дата приезда и уезда, город, количество гостей) и также более расширенные фильтры по сортировке, поиску по конкретным характеристикам.

Модуль клиента содержит в себе все вышеперечисленные страницы клиента и общие, но также имеет доступ:

– страница «Избранное» – содержит отели, которые пользователь хранит для дальнейшего сравнения;

– страница «Ваши бронирования» – содержит информацию об истории бронирований пользователя;

– страница «Бронирование» – оформление бронирования.

После оформления бронирования пользователь может перейти на страницу «Ваш билет», к которой в дальнейшем будет иметь доступ на странице «Ваши бронирования».

Модуль администратора веб-ресурса содержит информацию для отслеживания данных на страницах:

– страница «Агенты» – содержит полную информацию о менеджерах отелей и их параметрах с возможностью заблокировать аккаунт или создать новый;

– страница «Клиенты» – содержит полную информацию о клиентах их параметрах с возможностью заблокировать аккаунт или создать новый;

– страница «Бронирования» – содержит информацию о всех бронированиях только для просмотра;

– страница «Заявки на регистрацию отеля» – отображает приходящие заявки на регистрацию новых отелей, имеет возможность дать ответ на заявку.

Страницы администратора выполняют роль основного мониторинга большего количества данных.

Последний модуль менеджера отеля включает в себя страницы для мониторинга номеров зарегистрированного отеля:

- страница «Отель» – содержит основную информацию об отеле с возможностью её редактирования;
- страница «Номера» – содержит полную информацию о доступных номерах, созданию новых, удалению старых и обновлению информации;
- страница «Бронирования» – содержит информацию по текущим бронированиям, для каждого конкретного номера.

Для реализации поставленных задач требуется определить входную и выходную информацию для данного веб-ресурса.

Входные данные:

- заявка на регистрацию нового отеля – номер заявки, полное название, короткое название, фото логотипа и здания, адрес, звёздность, данные о менеджере отеля, также его контактные данные, информация о номерах, их количество, время въезда и выезда, дата отправки заявки;
- данные о номерах отеля – категория, количество спальных мест, цена за сутки, дополнительная техника, дополнительные услуги;
- сведения о пользователях веб-ресурса – фамилия, имя, отчество, дата рождения, номер телефона, электронных адрес почты, роль в веб-ресурсе.

Выходные данные:

- бронь отеля – дата оформления бронирования, дата въезда и выезда, номер отеля, цена бронирования, дополнительные услуги;
- ваучер о бронировании – цена, данные о брони, клиенте.

Входная и выходная информация должны храниться представляться перед пользователем веб-ресурса в полном виде и структурно храниться в баз данных системы.

3.3 Логическая модель данных

Логическая модель представляет собой базу данных, которая не привязана к

конкретной СУБД. В ней выделяют основные объекты БД и определяют связи между этими объектами. В таблице 1 представлены выделены основные сущности и их атрибуты, составляющие будущую базу данных.

Таблица 1 – Атрибуты сущностей

Сущность	Назначение сущности	Ключ	Характеристики связей
Аккаунт	Содержит данные об аккаунтах пользователей: – логин аккаунта; – пароль аккаунта; – описание аккаунта.	ID_Аккаунта (PK).	Аккаунт пользователя – Пользователь (1..n). Аккаунт пользователя – Роль (n..1). Аккаунт пользователя – История входа (n..1).
Роль	Содержит данные о ролях аккаунтов в веб-ресурсе: – наименование роли; – описание роли.	ID_Роли (PK).	Роль – Аккаунт пользователя (1..n).
История входа	Содержит данные о истории входов сотрудников в веб-ресурс: – дата и время входа аккаунта в веб-ресурс.	ID_Входа (PK).	История входа – Аккаунт пользователя (n..1).
Администратор	Содержит данные об администраторе: – имя; – фамилия; – отчество; – дата рождения имя.	ID_Администратора (PK).	Администратор – Заявка на регистрацию отеля (1..n).
Заявка на регистрацию отеля	Содержит данные о заявке: – дата и время формирования заявки; – информация об отеле; информация о номерах отеля.	ID_Заявки (PK).	Заявка на регистрацию отеля – Отель (1..1). Заявка на регистрацию отеля – Номер (1..n). Заявка на регистрацию отеля – Статус (n..1).

Продолжение таблицы 1

Сущность	Назначение сущности	Ключ	Характеристики связей
Авторизованный пользователь	Содержит данные об авторизованных пользователей: – имя; – фамилия; – отчество; – пол; – дата рождения; – номер телефона; – электронный адрес почты.	ID_Авторизованный пользователь (PK).	Пользователь – Отзыв об отеле (1..n). Пользователь – Заявка на регистрацию отеля (1..n). Пользователь – Бронь (n..1). Пользователь – Аккаунт пользователя (1..1). Пользователь – Паспортные данные (1..1).
Отель	Содержит данные об отелях: – полное наименование; – сокращенное наименование; – город; – точный адрес; – звёздность; – количество номеров.	ID_Отеля (PK).	Отель – Лист услуг (1..n) Отель – Категория отеля (n..1). Отель – Фото лист (1..n). Отель – Номер (1..n).
Номер	Содержит данные о номерах конкретного отеля: – наименование; – количество спальных мест; – количество комнат; – площадь в квадратных метрах; – цена за ночь.	ID_Номера отеля (PK).	Номер – Лист теники (1..n). Номер – Категория номера (n..1). Отель – Фото лист (1..n). Номер – Отель (n..1).

Продолжение таблицы 1

Сущность	Назначение сущности	Ключ	Характеристики связей
Бронирование	Содержит данные о бронировании: – дата и время формирования бронирования; – дата и время заселения; – дата и время выезда; – количество ночей; – цена.	ID_Бронирования (PK).	Бронирование – Статус бронирования (п..1). Бронирование – Номер (п..1). Бронирование – Гость (1..n).

После определения основных атрибутов, можно приступить к составлению логической модели данных. Для создания данной схемы необходимо нормализовать связи между атрибутами, чтобы избежать дублирования, связей между неключевыми полями и отношения «многие-ко-многим», а также уточнить типы данных выявленных атрибутов. Исходя из этого, была составлена логическая модель данных, представленная на рисунке 9.

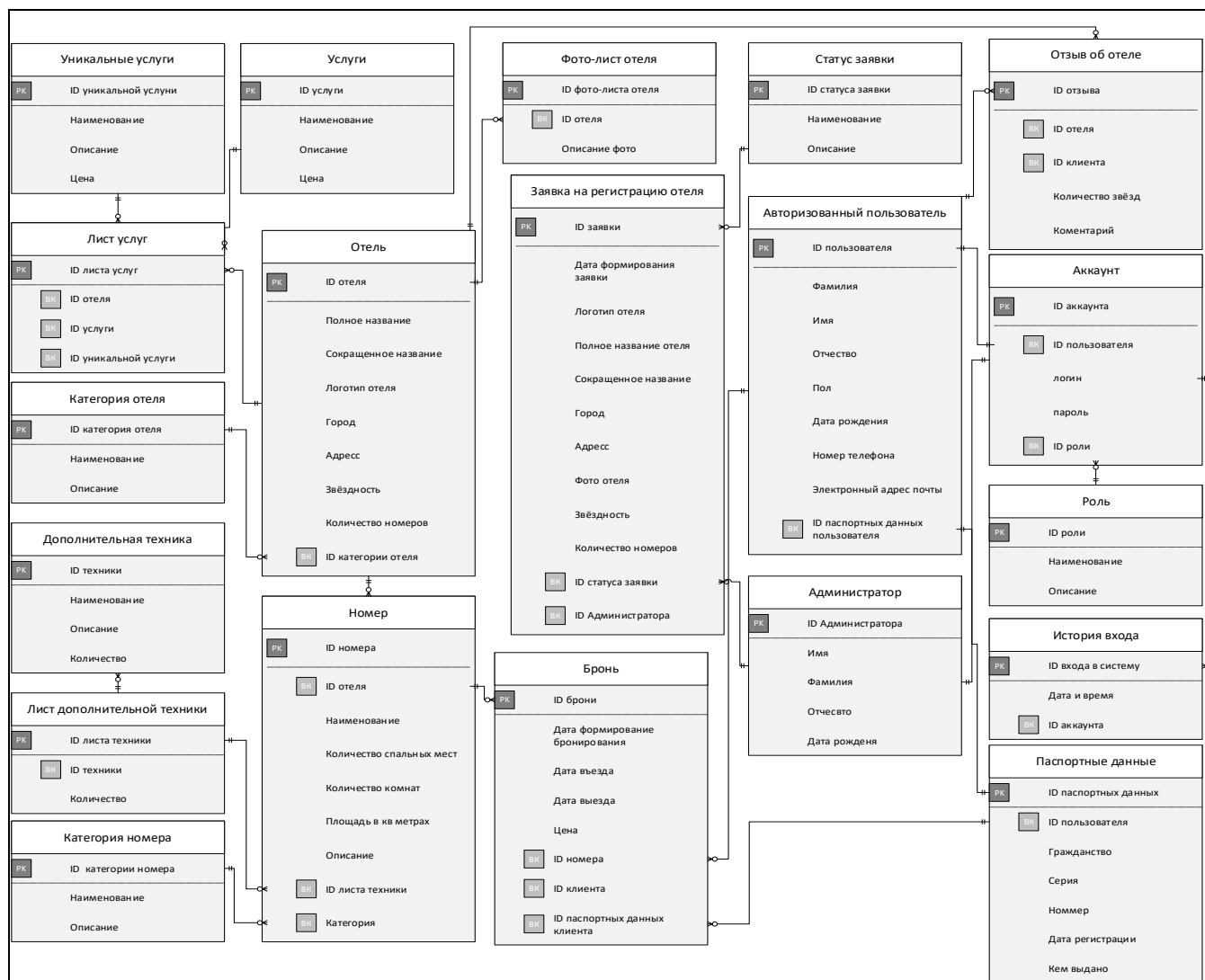


Рисунок 9 – ER-диаграмма веб-ресурса «Holiday Home»

ER-диаграмма, позволяет четко описать требования к представлению логической структуры данных, на основе которой в последующем будет разработана физическая структура данных для хранения во внешней памяти и программной обработки.

3.4 Обоснование выбора средств разработки

Веб-разработка – это непрерывный жизненный цикл, включающий множество этапов. Начиная с выбора языка программирования, подходящего для

работы фреймворка и заканчивая поиском надежного веб-сервера для размещения приложения – все это требует большой работы. Инструменты веб-разработки предназначены для упрощения и оптимизации жизненного цикла веб-разработчиков без ущерба для производительности.

При разработке веб-ресурса «Holiday Home» использовались следующие инструменты разработки: фреймворки Laravel и Bootstrap, такие языки программирования такие PHP, JavaScript.

Выбор средства разработки сайта на Laravel может быть обоснован рядом причин:

- стабильность и надежность. Laravel – это фреймворк с открытым исходным кодом, который имеет большое сообщество разработчиков и множество документации. Это делает Laravel стабильным и надежным средством разработки;

- продуктивность. Laravel предоставляет широкий спектр инструментов и функциональности, которые позволяют разработчикам быстро создавать веб-приложения высокого уровня сложности;

- гибкость. Laravel предоставляет множество готовых компонентов, которые можно использовать для создания приложений любой сложности, а также позволяет легко настраивать и расширять существующий функционал;

- безопасность. Laravel предоставляет встроенные механизмы защиты от основных видов атак на веб-приложения, что делает его одним из самых безопасных фреймворков для web-разработки;

- удобство работы с базой данных. Laravel предоставляет ORM (Object Relational Mapping) - мощный механизм работы с базами данных, который упрощает и ускоряет работу с данными.

Выбор Laravel в качестве средства разработки выбран на основе вышеперечисленных достоинств.

В целом, выбор PHP в качестве средства разработки сайта может быть обоснован как доступностью и гибкостью языка, так и наличием большого

сообщества разработчиков и богатым выбором инструментов и фреймворков для веб-разработки.

Bootstrap – это бесплатный фреймворк для создания адаптивных веб-сайтов. Он предоставляет программистам мощные инструменты для быстрого и простого создания красивых, современных и адаптивных сайтов.

Использование Bootstrap для создания сайта является оптимальным выбором, который обеспечивает адаптивность, эффективность, гибкость, кроссбраузерность и поддержку сообщества разработчиков.

MySQL является одной из самых популярных систем управления реляционными базами данных. Эта система зарекомендовала себя как надежное и эффективное решение для создания веб-сайтов всех масштабов, от небольших блогов до крупных электронных коммерческих платформ. MySQL является для создания базы данных на сайте оптимальным решением, которое обеспечивает быстроедействие, надежность, гибкость, простоту использования и экономичность.

Выбранные инструменты представлены в полном виде, имеют достаточно привилегий для разработки качественного и технически оснащённого веб-ресурса.

3.5 Алгоритм программного приложения

Проанализировав технологическую цепочку обработки информации для задач и модель базы данных, были разработаны алгоритмы программных модулей в соответствии с техническим заданием и описанием постановки задач. Алгоритм обработки информации представлены на рисунках 10 – 13.

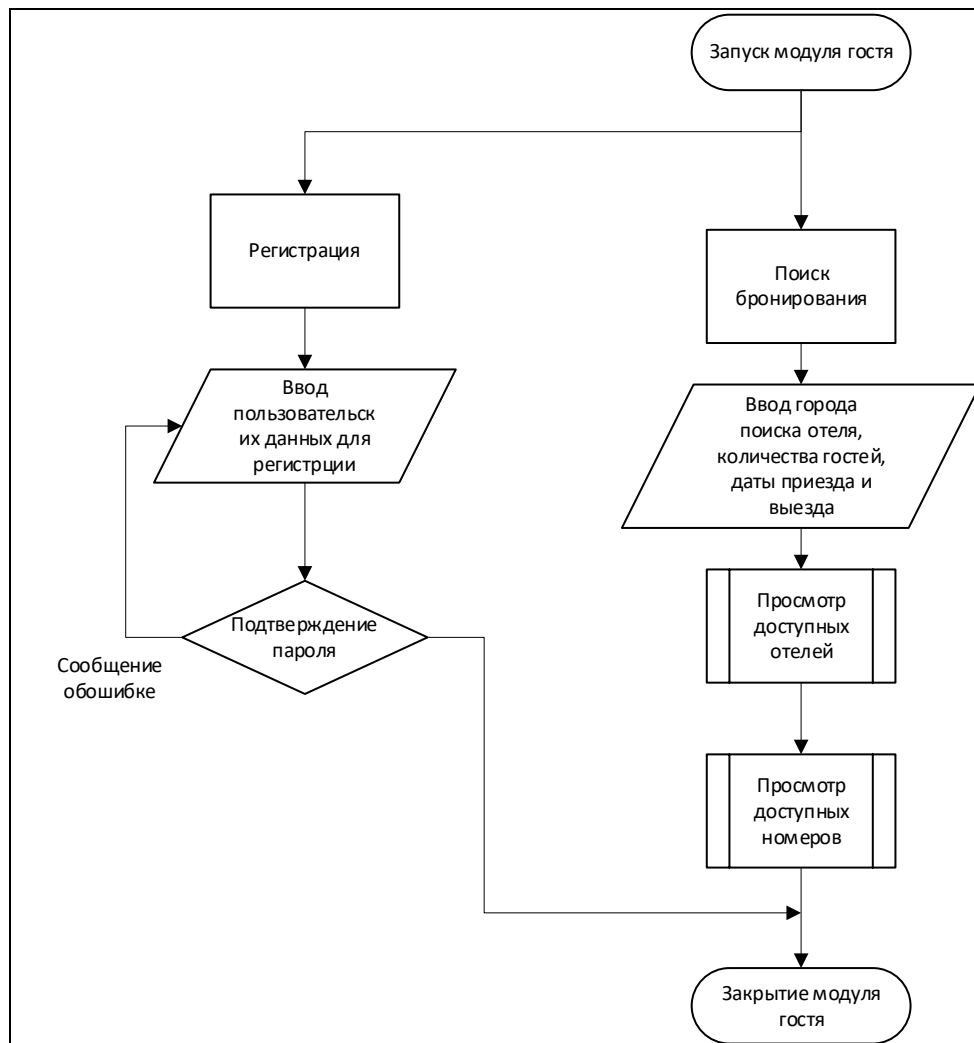


Рисунок 10 – Алгоритм работы модуля гостя

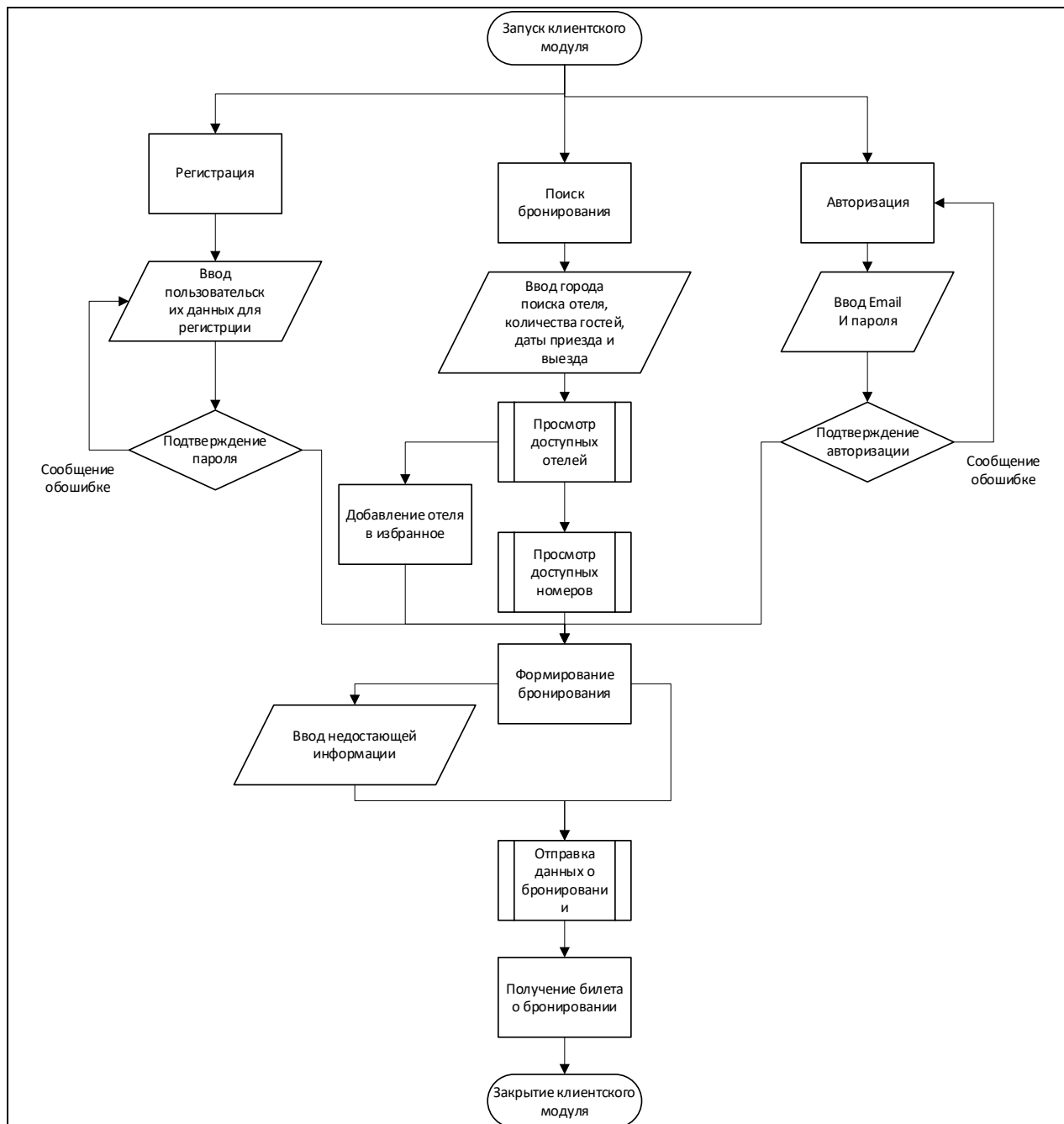


Рисунок 11 – Алгоритм работы модуля клиента

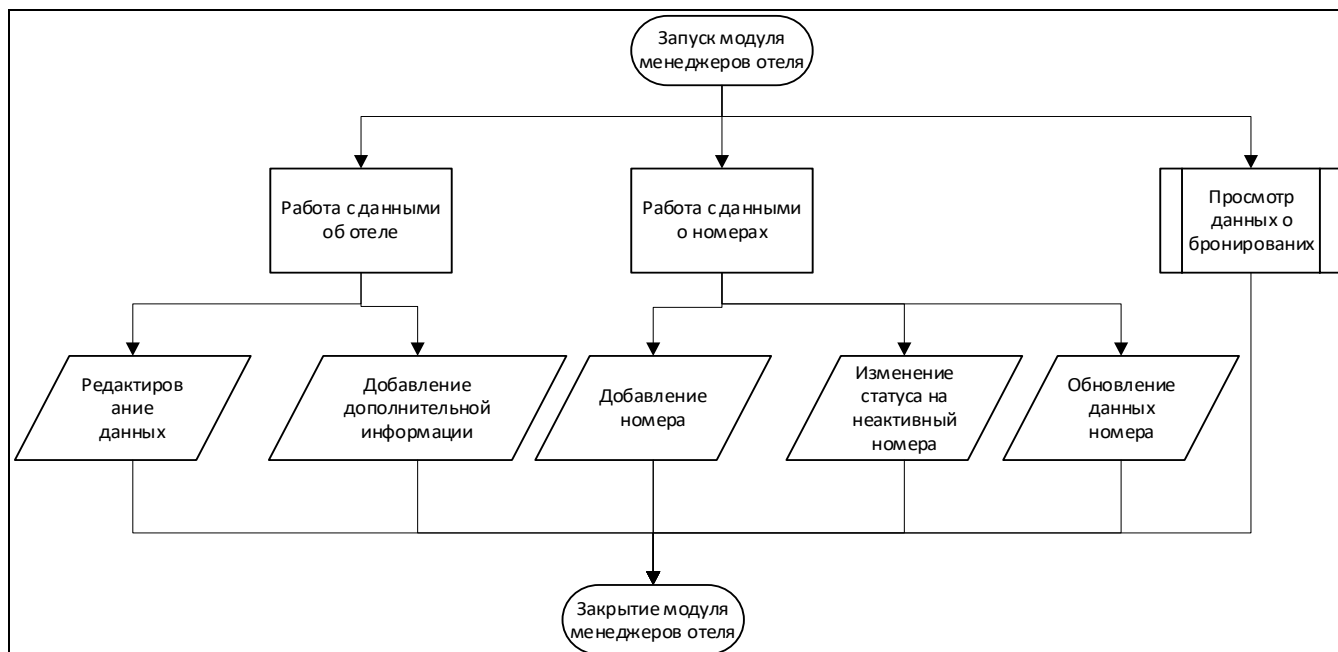


Рисунок 12 – Алгоритм работы модуля менеджера отеля

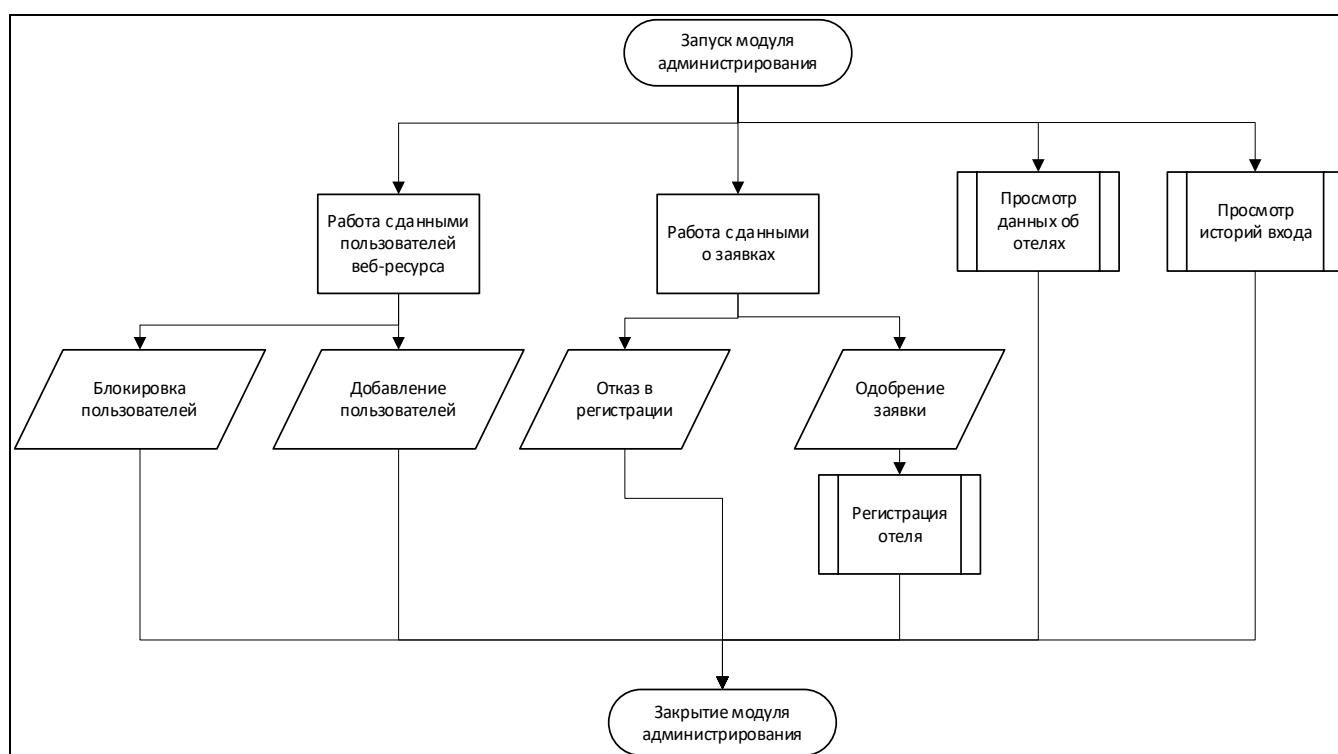


Рисунок 13 – Алгоритм работы модуля администратора системы

Благодаря данным алгоритмам можно более детально рассмотреть последовательность, в которой выполняются действия в разных модулях.

4 Разработка программно-информационного компонента веб-ресурса

4.1 Разработка программно-информационного компонента веб-ресурса

После проектирования была начата работа над программным компонентом веб-ресурса. Были разработаны четыре модуля веб-ресурса, каждый из которых включает в себя определенный функционал, а именно:

- модуль гостя;
- модуль клиента;
- модуль администратора веб-ресурса;
- модуль менеджера отеля.

Модуль гостя просмотр данных об отелях и номерах, на основе поиска нужны данных, что представлено на рисунках 14 – 17.

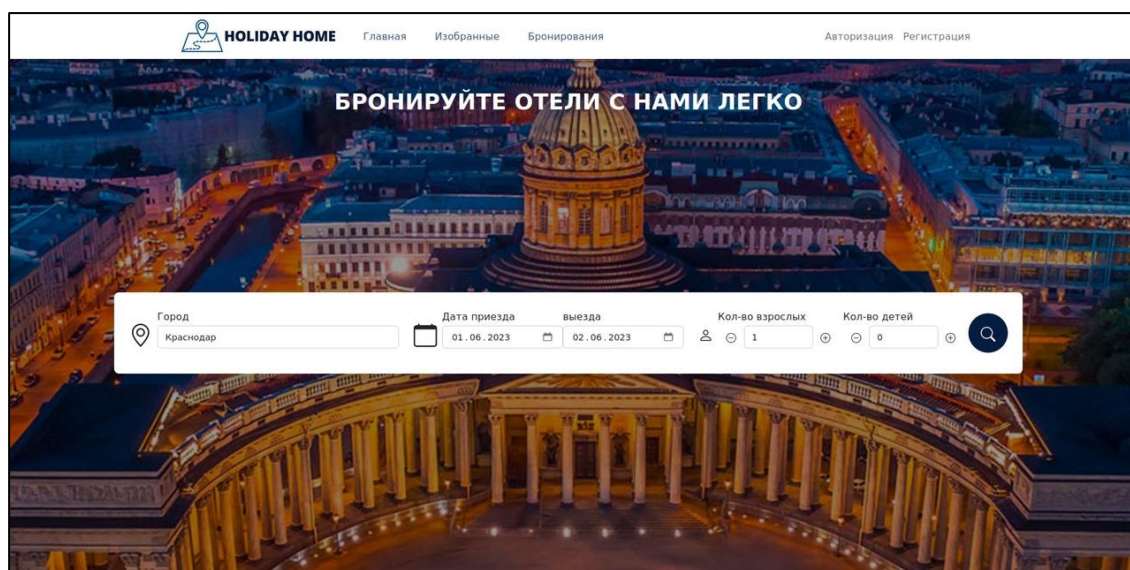


Рисунок 14 – Главная страница поиска отелей

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0087 ПЗ

Лист

31

ДОСТУПНЫЕ ОТЕЛИ

Город

Краснодар

Дата приезда

01.06.2023

выезда

02.06.2023

Кол-во взрослых

1

Кол-во детей

0

🔍

🔍 Фильтр

Название отеля

Рейтинг звёзд

1★

2★

3★

4★

5★


Тип отеля

Отель

Сброс

Применить

Рисунок 15 – Страница поиска отелей с дополнительными фильтрами




★★★★★

Гостиница Интурист-Краснодар в Краснодаре

• завтрак • обед • ужин

5000 руб /день

Детали →




★★★★★

Гостиница Платан южный в Краснодаре

1600 руб /день

Детали →




★★★★★

Бизнес-отель "Форум" в Краснодаре

5300 руб /день

Детали →



Лучший сервис для бронирования отелей

Главная

Личный кабинет

Клиентам

Список отелей

Партнёрам

Размещение отеля

Рисунок 16 – Страница с результатами поиска отелей

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0087 ПЗ

Лист

32

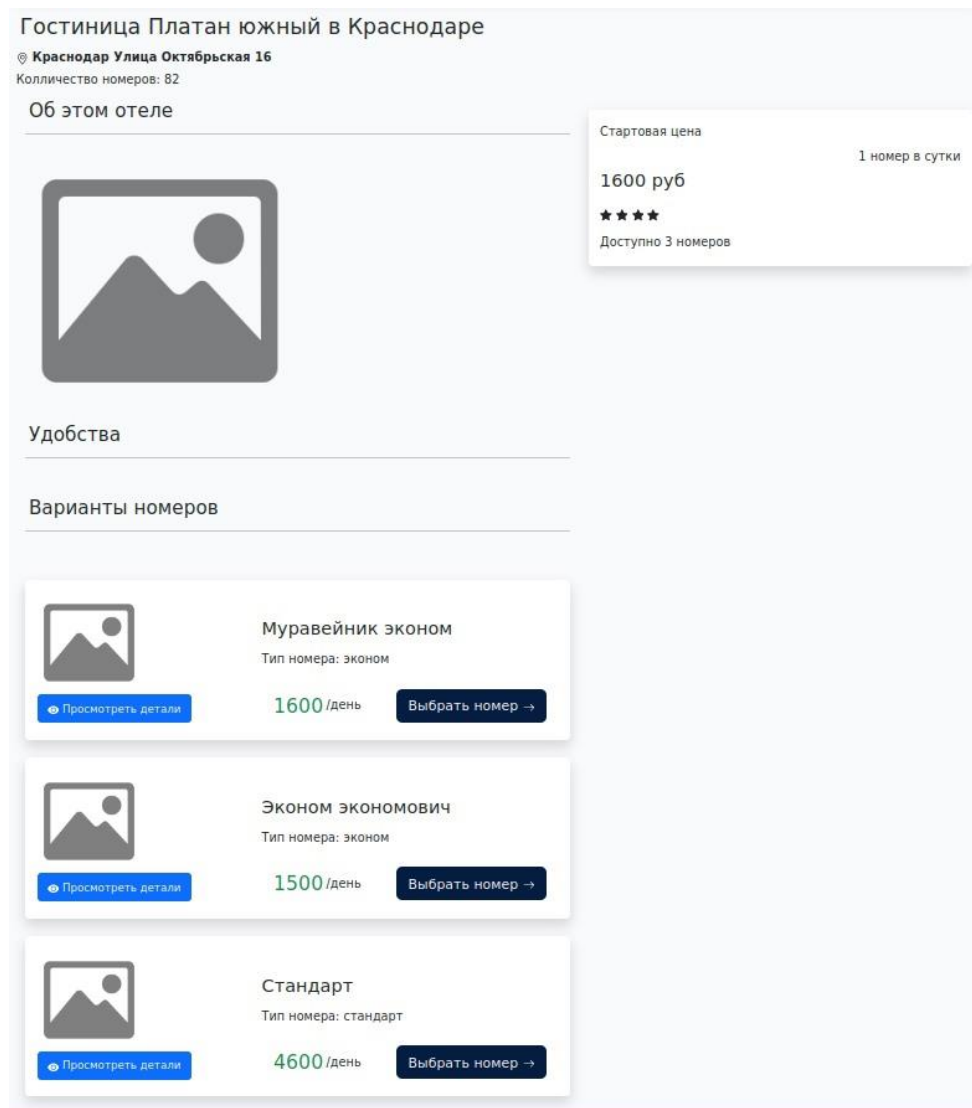


Рисунок 17 – Страница отеля с номерами, удовлетворяющие поиск

Модуль клиента содержит в себе ту же возможность подбора отеля, что и гость, но в его функционал добавляется подтверждение бронирования и получение чека, что представлено на рисунках 18 – 19.

Информация об отеле

Муравейник эконом

Платан южный

Краснодар, Улица Октябрьская 16

★★★★

Прибытие

2023-06-01

🕒 14:00

Уезд

2023-06-02

🕒 12:00

Гости

взрослых

детей

✳️ 2 дня - 1 ночи

Техника в номере

Цена

Цена номера за сутки	1600
Количество суток	1
Общая цена	1600

Подтвердить бронирование

Персональная информация о вас

Фамилия

Имя

Отчество

Фамилия

Имя

Отчество

Email

1524517@gmail.com

Номер телефона

8-000-000-00-00

Дата рождения*

дд . мм . гггг

📅

Выберете пол*

☐ Мужской

☐ Женский

Сохранить изменения

Введите ваши паспортные данные

Гражданство

Российское

Серия

Номер

Дата выдачи*

01 . 06 . 2023 📅

Кем выдан

Наименование организации

Сохранить изменения

Рисунок 18 – Страница подтверждение бронирования

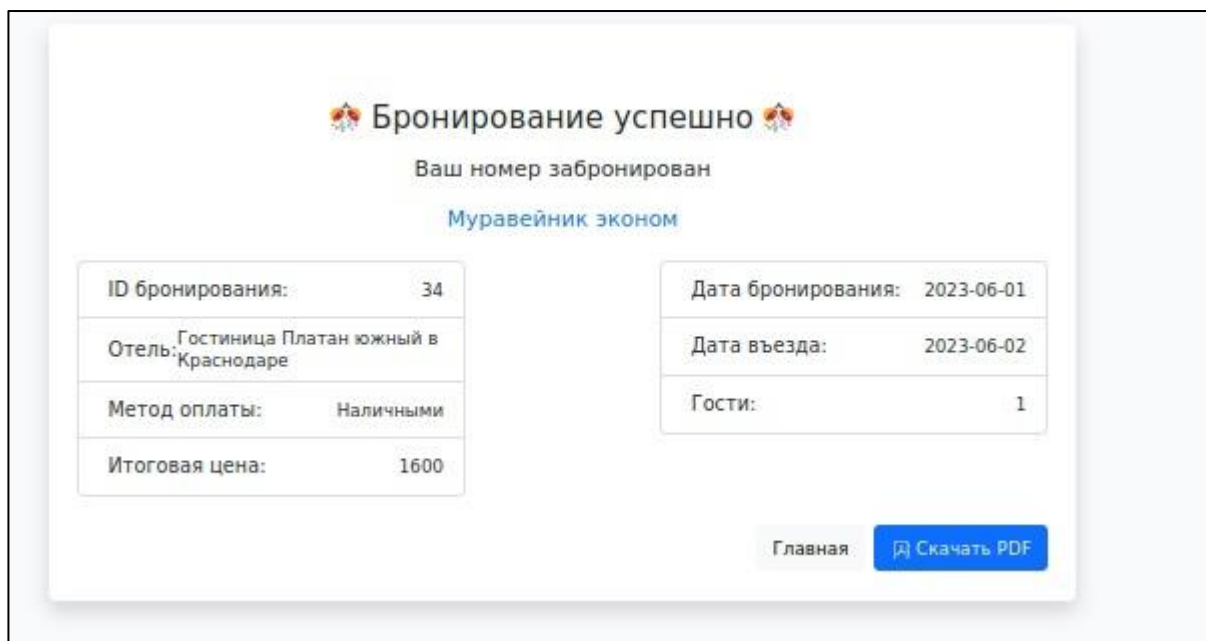


Рисунок 18 – Страница об успешном бронировании

Большая часть функционал администратора заключается в мониторинге данных об аккаунтах пользователей, страницы которых, что представлено на рисунках 19 – 22.

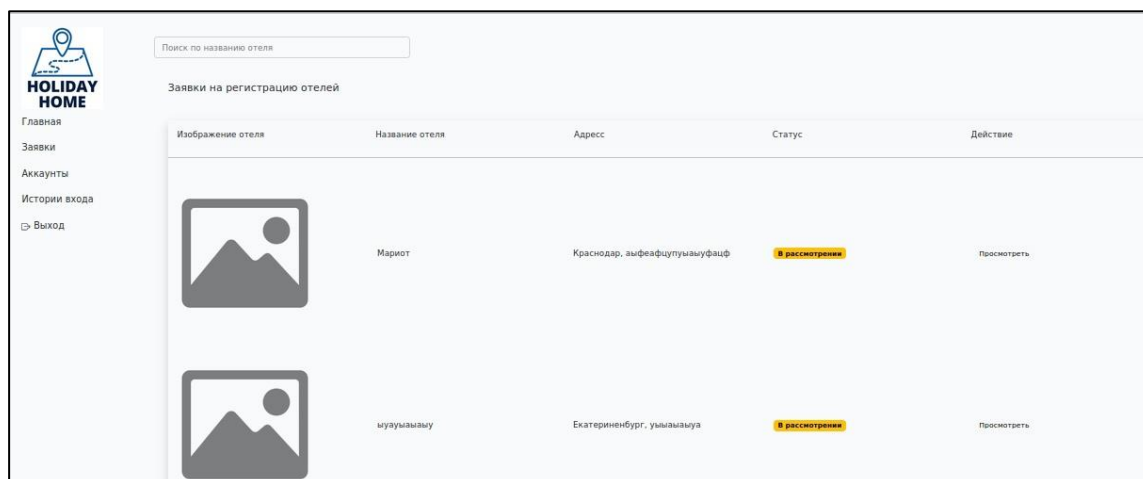


Рисунок 19 – Страница мониторинга заявок на регистрацию отелей

Функционал страниц менеджера отелей разделяется на мониторинг данных о бронированиях работа с данными об отеле и его номерах, представленных на рисунках 23 – 26.

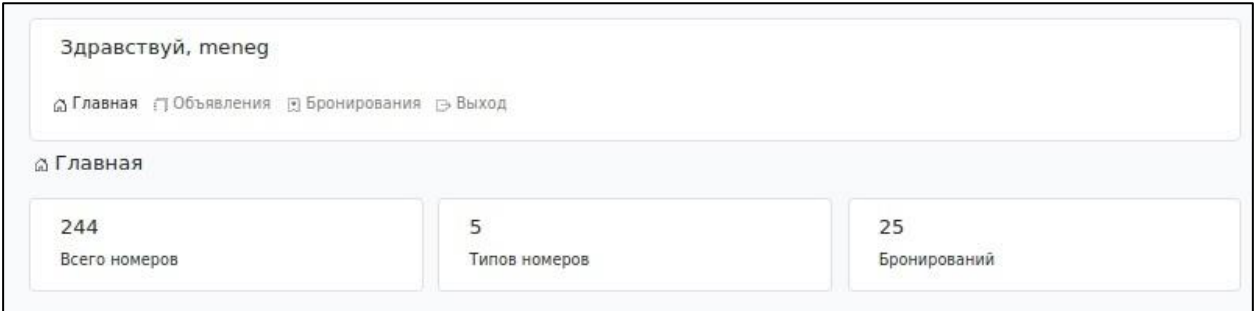


Рисунок 23 – Страница главного меню менеджера отеля

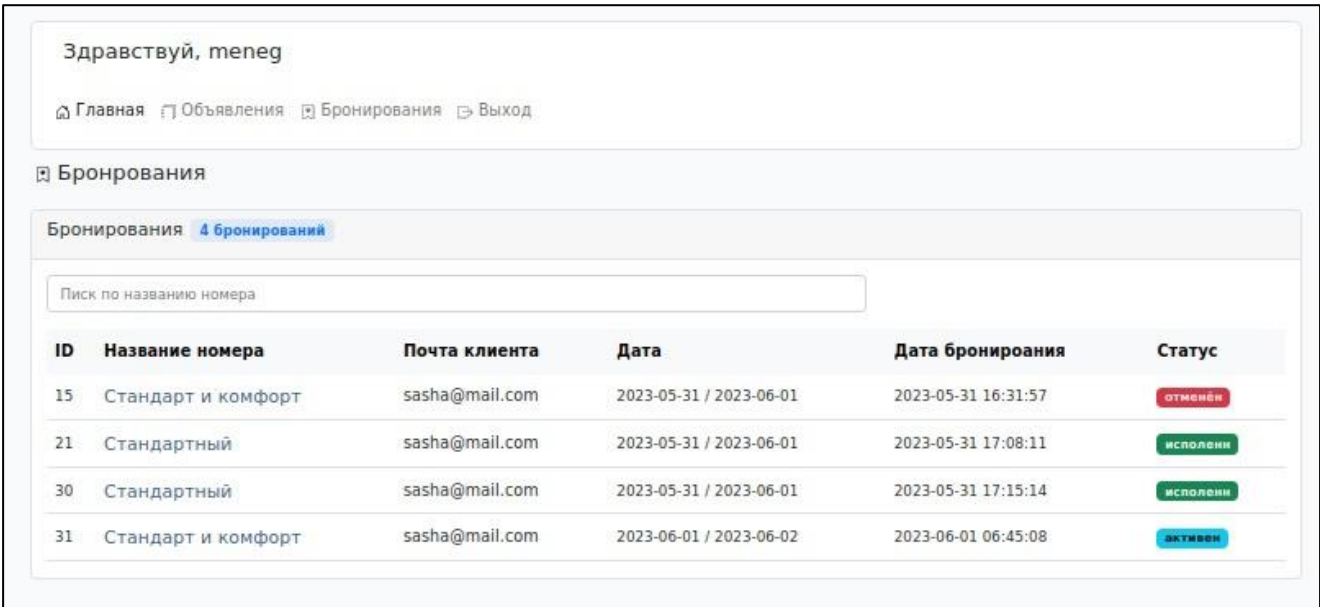


Рисунок 24 – Страница с данными о бронированиях

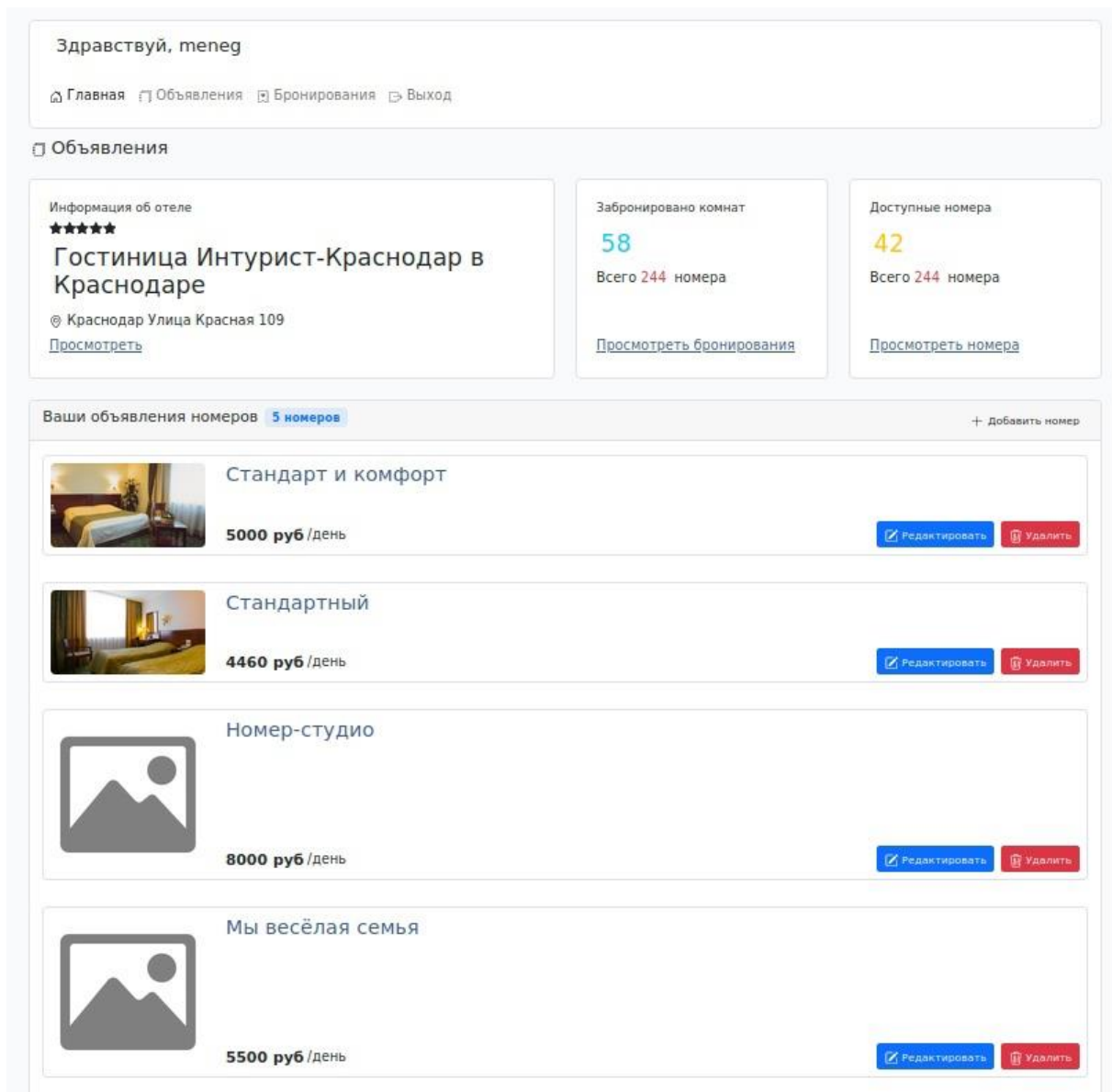


Рисунок 26 – Страница с данными номерах отеля

Представленные выше страницы должны обязательно корректно функционировать в веб-ресурсе и содержаться в полном виде.

4.2 Разработка эксплуатационной документации

Для входа в личный кабинет пользователю необходимо заранее зарегистрироваться в веб-ресурсе, заполнив форму корректными данными, представленную на рисунке 27.

Имя	<input type="text"/>
Email адрес	<input type="text"/>
Пароль	<input type="password"/>
Повторите пароль	<input type="password"/>
<input type="button" value="Зарегистрироваться"/>	

Рисунок 27 – Форма регистрации клиента в системе

В разделе «Ваши бронирования» вы сможете наблюдать историю бронирований, получить чек о бронировании или отменить его, как представлено на рисунках 28 – 30.

Ваши бронирования		
Предстоящие бронирования	Отменённые бронирования	Выполненные бронирования
Предстоящие бронирования		
Гостиница Интурист-Краснодар в Краснодаре Стандарт и комфорт ID бронирования: 31		<input type="button" value="Скачать билет"/> <input type="button" value="Отменить"/>
Дата заселения 2023-06-01	Дата выселения 2023-06-02	Забронировано hdrhdr rhd hdrhdr
Гостиница Платан южный в Краснодаре Муравейник эконом ID бронирования: 34		<input type="button" value="Скачать билет"/> <input type="button" value="Отменить"/>
Дата заселения 2023-06-01	Дата выселения 2023-06-02	Забронировано hdrhdr rhd hdrhdr

Рисунок 28 – Страница с данными действующих бронирований

Ваши бронирования		
Предстоящие бронирования	Отменённые бронирования	Выполненные бронирования
Отменено бронирований		
Гостиница Интурист-Краснодар в Краснодаре Стандарт и комфорт ID бронирования:15		Бронирование отменено
Дата заселения 2023-05-31	Дата выселения 2023-06-01	Отменено hdrhdr rhd hdrhdr
Гостиница Платан южный в Краснодаре Муравейник эконом ID бронирования:16		Бронирование отменено
Дата заселения 2023-05-31	Дата выселения 2023-06-01	Отменено hdrhdr rhd hdrhdr
Гостиница Платан южный в Краснодаре Муравейник эконом ID бронирования:17		Бронирование отменено
Дата заселения 2023-05-31	Дата выселения 2023-06-01	Отменено hdrhdr rhd hdrhdr
Гостиница Платан южный в Краснодаре Муравейник эконом ID бронирования:18		Бронирование отменено
Дата заселения 2023-05-31	Дата выселения 2023-06-01	Отменено hdrhdr rhd hdrhdr
Гостиница Платан южный в Краснодаре Стандарт ID бронирования:20		Бронирование отменено
Дата заселения 2023-05-31	Дата выселения 2023-06-01	Отменено hdrhdr rhd hdrhdr

Рисунок 29 – Страница с данными номерах отменённых бронирований

Ваши бронирования		
Предстоящие бронирования	Отменённые бронирования	Выполненные бронирования
Выполнено бронирований		
Гостиница Платан южный в Краснодаре Стандарт ID бронирования: 19		Просмотрите билет
Дата заселения 2023-05-31	Дата выселения 2023-06-01	Выполнено hdrhdr rhd hdrhdr
Гостиница Интурист-Краснодар в Краснодаре Стандартный ID бронирования: 21		Просмотрите билет
Дата заселения 2023-05-31	Дата выселения 2023-06-01	Выполнено hdrhdr rhd hdrhdr
Гостиница Интурист-Краснодар в Краснодаре Стандартный ID бронирования: 30		Просмотрите билет
Дата заселения 2023-05-31	Дата выселения 2023-06-01	Выполнено hdrhdr rhd hdrhdr

Рисунок 30 – Страница с данными о выполненных бронированиях

В случае если вы хотите зарегистрироваться как менеджер отеля, то вам необходимо в пункте нижнего меню в разделе «Партнёрам» выбрать «Размещение отелей». Далее вам будет предложено добавить отель, при регистрации которого вы введёте, как и свои данные, так и отеля, шаги действий представлены на рисунках 31 – 35.



Рисунок 31 – Страница для дальнейшего перехода регистрации отеля

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0087 ПЗ

Лист

42

Основная информация

Зарегистрируйтесь как менеджер отеля

Имя

Email адрес

Пароль

Повторите пароль

Рисунок 32 – Форма регистрации менеджера отеля

Информация об отеле

Выберите категорию отеля

Выберите тип отеля *

Отель

Отель: Профессиональные предприятия гостиничного бизнеса, которые обычно имеют уникальный стиль или тема, определяющая их бренд и декор

Полное название отеля *

Гостиница Интурист-Краснодар в Краснодаре

Включает в себя тип и название вашего жилья для туризма и город в котором она находится

Короткое название отеля *

Интурист-Краснодар

Маркировочное название отеля

Размер вашего отеля

Общее количество номеров *

Количество звёзд отеля *

Описание отеля *

Enter keywords

Локация

Город *

Выберите город из списка или добавьте новый

Добавить новый город

Адрес *

Введите адрес

Рисунок 33 – Форма с обязательными полями необходимых для заполнения при регистрации отеля

Выберете удобства отеля * ☐ завтрак ☐ обед ☐ ужин ☐ напитки в течении дня ☐ проживание с животными ☐ спортзал ☐ трансфер ☐ бассейн ☐ сауна ☐ кулер с водой ☐ Wi-Fi Максимум 14 ключевых слов. Ключевые слова должны быть написаны строчными буквами и разделены запятыми.

Загрузите фотографии отеля

Загрузите изображения *

Обзор...

Файл не выбран.

Обзор...

Файл не выбран.

Обзор...

Файл не выбран.

Обзор...

Файл не выбран.

Примечание: Только JPG, JPEG и PNG. Наш рекомендуемые размеры 600px * 450px. Изображение большего размера будет обрезано до 4:3, чтобы соответствовать нашим миниатюрам/превью.

Сохранить

Рисунок 34 – Форма с дополнительными полями при регистрации отеля

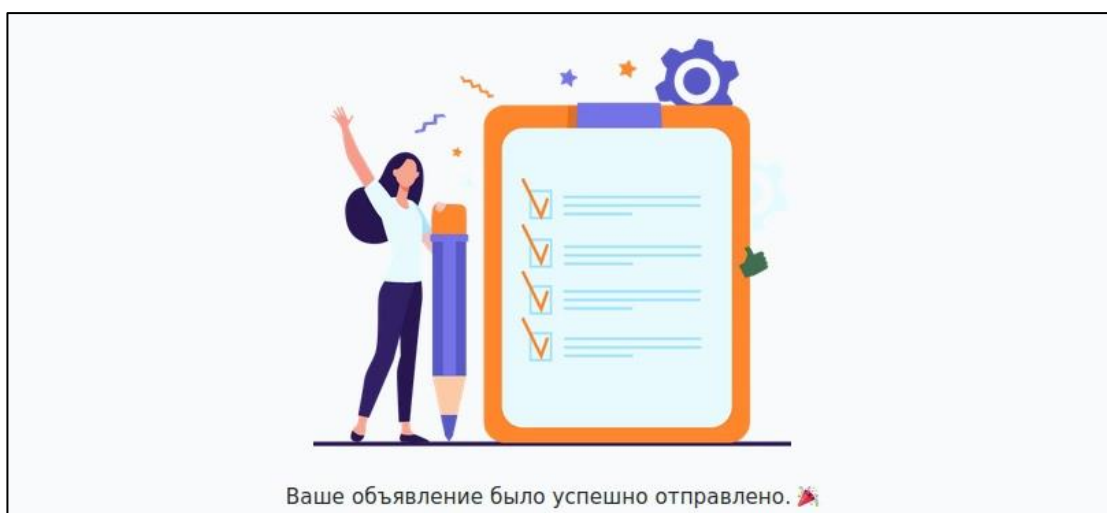


Рисунок 35 – Страница о успешной регистрации заявки отеля

Сразу же вы сможете авторизоваться как менеджер отеля, но данные о нём сразу отображаться не будут, как только администратор ресурса подтвердит данные о вашем отеле, вы сможете его администрировать.

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0087 ПЗ

Лист

44

5 Техника безопасности при работе с веб-ресурсом

При работе с веб-ресурсом существует несколько важных мер безопасности, которые следует соблюдать:

- использование надежных паролей: при регистрации на веб-ресурсе следует использовать сложные пароли, состоящие из различных символов, цифр и букв, чтобы их было сложнее угадать или подобрать методом перебора. Также не следует использовать одинаковый пароль на разных веб-ресурсах;

- осторожность при открытии вложений и ссылок: некоторые веб-ресурсы могут содержать вредоносные файлы или ссылки, которые могут привести к инфицированию компьютера вирусами или шпионским ПО. Поэтому следует быть очень осторожным при открытии вложений и ссылок, особенно если они пришли от незнакомых отправителей;

- обновление программ и браузера: следует регулярно обновлять программы и браузер, чтобы избежать уязвимостей и обеспечить безопасность работы с веб-ресурсом;

- использование антивирусного ПО: необходимо установить и регулярно обновлять антивирусное ПО для защиты компьютера от вирусов и других вредоносных программ;

- проверка сертификата безопасности: перед вводом личной информации на веб-ресурсе, необходимо убедиться, что на сайте установлен сертификат безопасности. Это можно сделать, проверив наличие замка в адресной строке браузера и наличие протокола "https";

Соблюдение этих мер безопасности поможет обезопасить работу с веб-ресурсом и предотвратить возможные угрозы безопасности.

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0087 ПЗ

Лист

45

Заключение

При разработке веб-ресурса бронирования отелей необходимо учитывать множество факторов, которые определяют успешность проекта. В частности, следует обратить внимание на следующие аспекты:

- удобство использования. Пользователи должны легко находить информацию о доступных отелях и свободных номерах, а также быстро и удобно осуществлять бронирование;
- качество предоставляемой информации. Вся информация о гостиницах и номерах должна быть корректной, полной и актуальной.
- безопасность данных. Веб-ресурс бронирования должна гарантировать надежную защиту пользовательских данных;
- маркетинговые возможности. Ресурс должен предоставлять возможности по продвижению гостиниц и привлечению новых клиентов.

Результатами реализации информационно-программного компонента является полностью готовый веб-ресурс «Holiday Home» для бронирования отелей. Программное приложение полностью соответствует требованиям, указанных в техническом задании и спроектировано на основе технического проекта.

В случае дальнейшей доработки веб-ресурса может дополняться внедрением дополнительных модулей по поддержки клиентов путём разработки чата с менеджеров отелей, разработкой подсистемы лояльности клиентов, ведением клиентской базы.

Результаты проведённой работы по созданию программного приложения «Holiday Home» можно использовать в работе бронирования и поиска отелей, ведение и учёт свободных номеров, необходимой документации для работы с клиентами и их активностями.

Список использованных источников

1. ГОСТ 7.0.1-2003. Библиографическая запись. Общие требования и правила составления (утв. постановлением Государственного комитета РФ по стандартизации и метрологии от 25 ноября 2003 г. № 332-ст). – Москва: ИПК Изд-во стандартов, 2004. – 47-48 с. – Текст непосредственный.

2. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания (утв. постановлением Государственного комитета СССР по управлению качеством продукции и стандартам от 29.12.90 N 3469). – Москва: ИПК Изд-во стандартов, 2004. – 43 с. – Текст непосредственный.

3. Гагарина Л.Г., Киселёв Д.В., Федотова Е.Л. Разработка и эксплуатация автоматизированных информационных систем: учеб. Пособие / Под ред. проф. Л.Г.Гагариной. – М.: ИД «Форум»: ИНФРА-М, 2017. – 384с. – Текст непосредственный.

4. Казарин, О. В. Основы информационной безопасности: надежность и безопасность программного обеспечения : учебное пособие для среднего профессионального образования / О. В. Казарин, И. Б. Шубинский. — Москва : Издательство Юрайт, 2023. — 342 с. — (Профессиональное образование). — ISBN 978-5-534-10671-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/518005> (дата обращения: 02.06.2023).

5. Колицына О. Л., Максимов Н. В., Попов И. И. Базы данных: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2014 – 89 с. с ил.

6. Кувшинов, Д. Р. Основы программирования : учебное пособие для среднего профессионального образования / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2022. — 105 с. — (Профессиональное образование). — ISBN 978-5-534-07560-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/493565> (дата обращения: 02.06.2023).

7. Рогозов Ю.И., Стукотий Л.Н., Свиридов А.С. Моделирование систем. Учебное пособие. Таганрог: Изд-во ТРТУ, 2014. – 105 с.

					<i>ККЭП 09.02.07 0087 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		47

8. Берн В.Г. Atmel.com [Электронный ресурс] – Atmel системы:
<http://www.atmel.com/ru/ru/products/microcontrollers/avr/default.aspx> (дата обращения 02.02.2023)

9. Стружкин, Н. П. Базы данных: проектирование : учебник для вузов / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2023. — 477 с. — (Высшее образование). — ISBN 978-5-534-00229-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/511019> (дата обращения: 12.04.2023).

10. Маркин, А. В. Программирование на SQL : учебное пособие для среднего профессионального образования / А. В. Маркин. — Москва : Издательство Юрайт, 2023. — 435 с. — (Профессиональное образование). — ISBN 978-5-534-11093-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/518166> (дата обращения: 15.05.2023).

11. Официальная документация фреймворка Laravel: [сайт]. – Текст. Изображение: электронные. – URL: <https://laravel.com/docs/10.x/installation> (дата обращения: 29.03.2023).

Приложение А
(обязательное)

Техническое задание. Требования к системе

1 Введение

1.1 Наименование программы

Наименование программной подсистемы «Holiday Home»

1.2 Краткая характеристика области применения

Информационный веб-ресурс «Holiday Home» предназначен для автоматизации процесса поиска и бронирования жилья для отдыха, путём анализа заинтересованности клиентов.

Одна из главных обязанностей веб-ресурса – предоставлять полную и актуальную информацию о жилищных условиях, которые первоначально регистрирует Администратор веб-ресурса по заявкам Менеджеров отелей.

Не зарегистрированный ранее в веб-ресурсе клиент принадлежит роли гость и имеет возможность просматривать отели, выбрать несколько понравившихся, добавив во вкладку избранное для дальнейшего сравнения и выбора лучшего для себя варианта. Любой пользователь, которого интересует жилье имеет возможность связаться с менеджером определенного отеля по указанным данным. Каждый гость для дальнейшего бронирования жилья может самостоятельно пройти регистрацию.

					ККЭП 09.02.07 0087 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		49

2 Основания для разработки

Основанием для разработки является Договор 36 от 10.04.2023. Договор утвержден Директором компании ООО «Отель Сервис» Смиренко Иваном Михайловичем, именуемым в дальнейшем Заказчиком, и Мельниковой Александрой Михайловной, именуемая в дальнейшем Исполнителем, 10.07.2023.

Согласно Договору, Исполнитель обязан разработать и разместить программный веб-ресурс «Holiday Home» на выделенном сервере не позднее 10.06.2023, предоставить исходные коды и документацию к разработанному веб-ресурсу не позднее 10.06.2023.

Наименование темы разработки – «Разработка веб-ресурса бронирования отелей «Holiday Home».

					ККЭП 09.02.07 0087 ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

3 Назначение разработки

3.1 Функциональное назначение

Веб-ресурс будет использоваться четырьмя категориями пользователей: гость, клиент, менеджер отеля и администратор веб-ресурса.

Веб-ресурс должен обеспечить возможности: выявление групп доступных номеров отеля на определённые даты, выбранные гостем или клиентом; обеспечивать регулярный мониторинг данных для представления актуальной информации работой с ней; обеспечить удобную систему бронирования и её отмену в случае необходимости.

Гость имеет возможность просмотра данных об отелях и доступных номеров, по входным параметрам поиска.

Клиент в свою очередь также как и гость может просматривать данные о номерах, но и также оформлять бронирование, на основе которого получает чек, также добавлять понравившееся отели в категорию избранных и удалять из этого списка.

Менеджер отеля имеет возможность просматривать, вносить и редактировать данные об отелях и номерах, управлять данными.

Администратор может регистрировать новых пользователей веб-ресурса и удалять старых, отслеживать историю входа.

3.2 Эксплуатационное назначение

Веб-ресурс будет использоваться на рабочих местах пользователей, представляющих собой персональные компьютеры, имеющие доступ к сети Интернет и будет располагаться на выделенном хостинге.

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0087 ПЗ

Лист

51

4 Требования к программе

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

В разрабатываемом веб-ресурсе должна быть предоставлена возможность гибкого поиска отелей, на основе параметров. Так же удобное добавление отеля и его проверка перед публикацией в веб-ресурсе.

Веб-ресурс из базы данных должна брать необходимые данные об отелях и клиента, необходимые ей для функционирования, осуществления бронирования, отслеживания лучших вариантов и рекомендации их клиентам.

4.1.2 Требования к организации входных и выходных данных

Требования к входным и выходным данным описаны на основе данных собеседования с заказчиком.

Входные данные:

- заявка на регистрацию нового отеля – номер заявки, полное название, короткое название, фото логотипа и здания, адрес, звёздность, данные о менеджере отеля, также его контактные данные, информация о номерах, их количество, время въезда и выезда, дата отправки заявки;
- данные о номерах отеля – категория, количество спальных мест, цена за сутки, дополнительная техника, дополнительные услуги;
- сведения о пользователях системы – фамилия, имя, отчество, дата рождения, номер телефона, электронный адрес почты, роль в системе.

Выходные данные:

- бронь отеля – дата оформления бронирования, дата въезда и выезда, номер отеля, цена бронирования, дополнительные услуги;

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0087 ПЗ

Лист

52

- ваучер бронировании – цена, данные о брони, клиенте;
- отзыв – количество звёзд об оценки обслуживания, комментарий, данные о комментаторе.

4.2 Требования к надежности и безопасности

Веб-ресурс должен функционировать в многопользовательском режиме, поэтому каждый пользователь должен иметь свой пароль доступа в систему и уникальный адрес электронной почты (email) в пределе системы. Необходимо разграничить пользовательские права, т.е. не дать гостю возможности бронирования жилья. Одновременно в системе может находиться несколько пользователей. Веб-ресурс должен обеспечить одновременный доступ к одним и тем же данным в базе данных, для чего необходимо применение архитектуры клиент–сервер. Кроме того, в веб-ресурсе должно быть предусмотрено резервное копирование и восстановление данных, а также самовосстановление веб-ресурса после сбоев в операционной системе или отключения электропитания.

4.3 Требования к составу и параметрам технических средств

Состав технических средств предусматривает наличие компьютеров-рабочих станций, мобильных телефонов – рабочих станций и компьютеров-серверов.

Компьютер-рабочая станция включает:

- процессор Intel Pentium 4 / Athlon 64 или более поздней версией с поддержкой SSE2;
- оперативную память объемом, не менее 4 Гб;
- свободное место на диске 350 Мб;
- видеокарту, монитор, мышь, клавиатура.

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0087 ПЗ

Лист

53

Мобильный телефон-рабочая станция включает:

- оперативную память объемом, не менее 8 Гб;
- версия телефона не позднее Android 7.0 Nougat.

Два компьютера-сервера (для функционирования СУБД) (основной и резервный), включают в себя:

- процессор x86 с тактовой частотой, не менее 1 ГГц;
- оперативную память объемом, не менее 1 Гб;
- видеокарту, монитор, мышь.

4.4 Требования к информационной и программной совместимости

На рабочих станциях должна быть установлена любая из перечисленных поисковых систем Google, FireFox, Opera, Яндекс, также интернет-соединение со скоростью не ниже 3G.

На компьютерах-рабочих станциях должна быть установлена операционная система ОС Ubuntu. Все формируемые отчеты должны иметь возможность экспортирования в редактор электронных таблиц в текстовом формате.

Приложение Б

(обязательное)

Программный код модулей веб-ресурса

Программный код представлен в контексте текстового редактора JetBrains PhpStorm Версия 2022.2.1, на рисунках Б.1 – Б.13.

```
33 Route::get('/hotel-accommodation', [NewHotelController::class, 'index'])->name('hotel-accommodation.index');
34 Route::get('/hotel-accommodation/agent-information', [NewHotelController::class, 'viewAgentInformation'])->name('agent.information.add');
35 Route::get('/hotel-accommodation/hotel-information', [NewHotelController::class, 'viewAddHotel'])->name('hotel.information.add');
36 Route::get('/hotel-accommodation/end', [NewHotelController::class, 'viewEndAccommodationHotel'])->name('hotel-accommodation.end');
37 Route::post('/hotel-accommodation/agent-information/store', [NewHotelController::class, 'storeAgentInformation'])->name('agent.information.store');
38 Route::post('/hotel-accommodation/hotel-information/store', [NewHotelController::class, 'storeAddHotel'])->name('hotel.information.store');
39
40 Route::resource('hotels', HotelsController::class);
41 Route::resource('rooms', RoomController::class);
42 Route::resource('booking', BookingController::class);
43 Route::post('/hotels-search', [HotelsController::class, 'searchHotels'])->name('hotels.search');
44 Route::get('/hotels-search/{id}', [HotelsController::class, 'searchHotelsForCity'])->name('hotels.city.search');
45 Route::post('/hotels-search-support', [HotelsController::class, 'searchSupportHotels'])->name('hotels.search.support');
46 Route::get('/hotels-search-support-disable', [HotelsController::class, 'searchSupportDisableHotels'])->name('hotels.search.support.disable');
47 Route::get('/room-for-search-hotel/{id}', [HotelsController::class, 'showHotel'])->name('hotels.list.search');
48 Route::middleware('auth')->group(function () {
49     Route::get('/booking-review/{id}', [BookingController::class, 'viewBookingReview'])->name('booking.review');
50     Route::get('/favorites-hotels/save/{id}', [FavoritesHotelController::class, 'save'])->name('favorites.hotels.save');
51     Route::resource('favorites-hotels', FavoritesHotelController::class);
52     Route::get('/favorites-hotels/delete/{id}', [FavoritesHotelController::class, 'delete'])->name('favorites.hotels.delete');
53     Route::get('/profile', [UserProfileController::class, 'show'])->name('profile.show');
54     Route::post('/save-user-data', [UserProfileController::class, 'saveUserData'])->name('user.data.save');
55     Route::post('/save-user-password-data', [UserProfileController::class, 'saveUserPassportData'])->name('user.passport.save');
56     Route::get('/check-booking/{id}', [BookingController::class, 'checkBooking'])->name('check.booking');
57     Route::get('/booking', [BookingController::class, 'index'])->name('index.booking');
58     Route::get('/booking/ticket/{id}', [BookingController::class, 'ticket'])->name('booking.ticket');
59     Route::get('/booking/disable/{id}', [BookingController::class, 'bookingDisable'])->name('booking.disable');
60     Route::get('/admin-panel', [AdminController::class, 'showPanel'])->name('admin.panel');
61     Route::get('/admin/login-history', [LoginHistoryController::class, 'historyAdmin'])->name('admin.history');
62     Route::get('/admin/login-history/guests', [LoginHistoryController::class, 'historyGuests'])->name('admin.history.guests');
63     Route::get('/admin/login-history/agents', [LoginHistoryController::class, 'historyAgents'])->name('admin.history.agents');
64     Route::get('/admin/hotels', [AdminHotelController::class, 'index'])->name('admin.hotels');
65     Route::get('/admin/hotels/new', [AdminHotelController::class, 'newHotels'])->name('admin.hotels.new');
66     Route::get('/admin/hotel/{id}', [AdminHotelController::class, 'show'])->name('admin.hotel.info');
67     Route::post('/admin/hotel/save/status/{id}', [AdminHotelController::class, 'statusSave'])->name('status.hotel.save');
68     Route::get('/admin/delete/{id}', [AdminController::class, 'delete'])->name('admin.delete');
69     Route::get('/guests/delete/{id}', [GuestController::class, 'delete'])->name('guests.delete');
70     Route::get('/agents/delete/{id}', [AgentController::class, 'delete'])->name('agents.delete');
71     Route::resource('admin', AdminController::class);
72     Route::resource('guests', GuestController::class);
73     Route::resource('agents', AgentController::class);
74     Route::resource('admin-booking', BookingAdminController::class);
75     Route::resource('lists', ListController::class);
76     Route::get('/agent/settings', [SettingsAccountController::class, 'index'])->name('agent.settings');
77     Route::get('/agent/dashboard', [AgentController::class, 'showPanel'])->name('agent.panel');
78     Route::get('/agent/booking', [AgentController::class, 'showBooking'])->name('agent.booking');
79     Route::get('/agent/hotel/settings', [SettingsAccountController::class, 'hotel'])->name('agent.settings.hotel');
80     Route::post('/agent/hotel/settings/update/{id}', [SettingsAccountController::class, 'updateHotel'])->name('hotel.information.update');
81 });
82 Route::middleware('auth', 'timeLogin', 'admin', 'agent')->group(function () {
83     Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
84 });
85 Auth::routes();
```

Рисунок Б.1 – Программный код роутера веб-ресурса

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0087 ПЗ

Лист

55

```

37     public function viewBookingReview(string $roomId)
38     {
39
40         $booking['count_adults'] = session('count_adults');
41         $booking['count_children'] = session('count_children');
42
43         $room = Room::find((int)$roomId);
44         $roomEquipmentLists = RoomEquipmentList::where('room_id', (int)$roomId)->get();
45         $hotel = Hotel::find($room->hotel_id);
46         $booking = $this->creatBookingDefault($roomId, $room->hotel_id);
47
48         $timeArrival = strtotime($booking['arrival_date']);
49         $timeDeparture = strtotime($booking['date_departure']);
50         $countNight = ($timeDeparture - $timeArrival) / (60 * 60 * 24);
51
52         $userId = Auth::user()->id;
53         $user = User::find($userId);
54         $userData = DataUsers::where('user_id', $userId)->get();
55         $userPassportData = PassportDataUsers::where('data_user_id', $userId)->get();
56
57         $sumPrice = $room->price * $countNight;
58         return view('booking/review-booking', [
59             'room' => $room,
60             'roomEquipmentLists' => $roomEquipmentLists,
61             'hotel' => $hotel,
62             'dataBooking' => $booking,
63             'countNight' => $countNight,
64             'sumPrice' => $sumPrice,
65             'userData' => $userData,
66             'userPassportData' => $userPassportData,
67             'user' => $user
68         ]
69     );
70 }

```

Рисунок Б.2 – Программный код функции вывода информации о подтверждении бронирования

```

72     public function checkBooking(int $bookingId)
73     {
74         $booking = Booking::find($bookingId);
75         $guests = Guest::where('booking_id', $bookingId->get());
76         $children = Children::where('booking_id', $bookingId->get());
77         $userData = DataUsers::where('user_id', Auth::user()->id)->first();
78
79         if (!isset($userData) || count($guests) + 1 != (int)session('count_adults') || count($children) != (int)session('count_children')) {
80             return back()->exceptInput();
81         }
82
83         $booking->status_id = 2;
84         $booking->save();
85
86         $timeArrival = strtotime($booking['arrival_date']);
87         $timeDeparture = strtotime($booking['date_departure']);
88         $countNight = ($timeDeparture - $timeArrival) / (60 * 60 * 24);
89         return view('booking/ticket-booking', ['booking' => $booking, 'countNight' => $countNight]);
90     }
91
92     private function creatBookingDefault(int $roomId, int $hotelId)
93     {
94         return Booking::create([
95             'user_id' => Auth::user()->id,
96             'arrival_date' => session('arrival_date'),
97             'date_departure' => session('date_departure'),
98             'count_night' => session('count_adults'),
99             'room_id' => $roomId,
100             'status_id' => 1,
101             'hotel_id' => $hotelId
102         ]);
103     }

```

Рисунок Б.3 – Программный код функции проверки бронирования и его создания

```

120     public function ticket(int $bookingId) {
121         $booking = Booking::find($bookingId);
122         $hotel = Hotel::find($booking->hotel_id);
123         $userData = DataUsers::where('user_id', $booking->user->id)->first();
124
125         $pdf = new \FPDF('P', 'mm', 'A4');
126         $pdf->AddPage();
127         $pdf->AddFont('DejaVu', '', 'DejaVuSansCondensed.ttf', true);
128         $pdf->SetFont('DejaVu', '', 14);
129
130         $text = "Билет о бронировании отеля";
131         $pdf->Cell(40, 10, $text);
132
133         $text = "ФИО: " . $userData->surname . " " . $userData->name . " " . $userData->patronymic;
134         $pdf->Cell(10, 2, $text);
135         $text = $hotel->full_title . "номер: " . $booking->room->title;
136         $pdf->Cell(10, 40, $text);
137         $text = "Дата приезда: " . $booking->arrival_date;
138         $pdf->Cell(10, 80, $text);
139         $text = "Дата выезда: " . $booking->date_departure;
140         $pdf->Cell(10, 90, $text);
141
142         $pdf->Output('D', 'certificate.pdf');
143     }
144 }

```

Рисунок Б.4 – Программный код функции вывод билета в PDF файл


```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\FavoritesHotels;
6  use App\Models\Hotel\HotelServicesList;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Auth;
9
10 class FavoritesHotelController extends Controller
11 {
12     /**
13      * Display a listing of the resource.
14      */
15     public function index()
16     {
17         $favorites = FavoritesHotels::where('user_id', (int)Auth::user()->id)->get();
18         $servicesList = HotelServicesList::all();
19         return view('hotel/favorites-hotels', ['favorites' => $favorites, 'servicesList' => $servicesList]);
20     }
21     /**
22      * Remove the specified resource from storage.
23      */
24     public function delete(string $idHotel)
25     {
26         $favorite = FavoritesHotels::where('user_id', Auth::user()->id)
27             ->where('hotel_id', $idHotel)
28             ->delete();
29
30         return redirect()->route('favorites-hotels.index');
31     }
32
33     public function save(string $idHotel)
34     {
35         FavoritesHotels::create([
36             'user_id' => Auth::user()->id,
37             'hotel_id' => $idHotel,
38         ]);
39
40         return redirect()->route('hotels.index');
41     }
42 }
43

```

Рисунок Б.5 – Программный код контроллера, обеспечивающий добавление и вывод избранных пользователем отелей

```

78 ✓ public function searchHotels(SearchHotelRequest $request)
79 {
80     $validate = $request->validated();
81
82     if ($validate['arrival_date'] >= $validate['date_departure']) {
83         return back()->withInput();
84     }
85
86     $countGuest = $validate['count_children'] + $validate['count_adults'];
87     $city = City::where('title', $validate['city'])->get();
88
89     $hotelsWithRooms = DB::table('hotels')
90         ->join('rooms', 'rooms.hotel_id', '=', 'hotels.id')
91         ->leftJoin('bookings', 'bookings.room_id', '=', 'rooms.id')
92         ->where([
93             ['number_beds', '>=', $countGuest],
94             ['city_id', '=', (int)$city[0]->id],
95             ['arrival_date', '=', $validate['arrival_date']],
96             ['date_departure', '=', $validate['date_departure']],
97             ['bookings.status_id', '=', 2]
98         ])
99         ->get();
100
101     $hotels = $hotelsWithRooms->unique("hotel_id");
102
103     $categories = HotelCategories::all();
104     $services = HotelServices::all();
105     $servicesList = HotelServicesList::all();
106
107     session(['city' => $validate['city']]);
108     session(['arrival_date' => $validate['arrival_date']]);
109     session(['date_departure' => $validate['date_departure']]);
110     session(['count_adults' => $validate['count_adults']]);
111     session(['count_children' => $validate['count_children']]);
112
113     return view('hotel/hotel-list', ['hotels' => $hotels, 'categories' => $categories, 'services' => $services, 'servicesList' =>
114     ]);
115

```

Рисунок Б.6 – Программный код функции, которая выводит отели, удовлетворяющие первоначальный поиск

```

20      class HotelsController extends Controller
141 ✓    public function searchSupportHotels(Request $request)
142    {
143        $validate = [
144            'city' => session('city'),
145            'arrival_date' => session('arrival_date'),
146            'date_departure' => session('date_departure'),
147            'count_adults' => session('count_adults'),
148            'count_children' => session('count_children')
149        ];
150
151        $countGuest = $validate['count_children'] + $validate['count_adults'];
152        $city = City::where('title', $validate['city'])->get();
153
154        $hotelsWithRooms = DB::table('hotels')
155            ->rightJoin('rooms', 'rooms.hotel_id', '=', 'hotels.id')
156            ->where([
157                ['number_beds', '>=', $countGuest],
158                ['city_id', '=', (int)$city[0]->id],
159            ])
160            ->get();
161        $hotels = $hotelsWithRooms->unique("hotel_id");
162
163        foreach ($hotels as $key => $hotel) {
164            if ($hotel->short_title != $request->title && !is_null($request->title)) {
165                unset($hotels[$key]);
166            }
167            if ($hotel->category_id != $request->category) {
168                unset($hotels[$key]);
169            }
170
171            if ($hotel->star != (float)$request->star && isset($request->star)) {
172                unset($hotels[$key]);
173            }
174        }
175
176        $categories = HotelCategories::all();
177        $services = HotelServices::all();
178        $servicesList = HotelServicesList::all();
179
180
181        return view('hotel/hotel-list', ['hotels' => $hotels, 'categories' => $categories, 'services' => $services, 'servicesList' => $
182    }

```

Рисунок Б.7 – Программный функции, которая выводит отели, удовлетворяющие
дополнительный поиск

```

22 class NewHotelController extends Controller
34 ✓ public function viewAddHotel()
35 {
36     $cities = City::all();
37     $typeHotels = HotelCategories::all();
38     $services = HotelServices::all();
39     return view('hotel-accommodation/basic-information', ['cities' => $cities, 'typeHotels' => $typeHotels, 'services' =>
40     ]);
41 }
42 public function viewEndAccommodationHotel()
43 {
44     return view('hotel-accommodation/end-add-listing');
45 }
46 }
47 ✓ public function storeAgentInformation($request)
48 {
49     $validate = $request->validated();
50 }
51     $user = User::create([
52         'name' => $validate['name'],
53         'email' => $validate['email'],
54         'password' => Hash::make($validate['password']),
55     ]);
56 }
57     Agents::create([
58         'user_id' => $user['id'],
59     ]);
60 }
61     session(['agentId' => $user['id']]);
62 }
63     return redirect()->route('hotel.information.add');
64 }

```

Рисунок Б.8 – Программный код вывода форм для добавления нового отеля

```

66 ✓ public function storeAddHotel(AddBaseInformationHotelRequest $request)
67     {
68         $data = $request->all();
69         $validate = $request->validated();
70         $cityId = $validate['city_id'];
71
72         if ($validate['city_id'] == 0 && !is_null($validate['city_add'])) {
73             $city = City::create([
74                 'type' => 'город',
75                 'title' => $validate['city_add'],
76             ]);
77
78             $cityId = $city->id;
79         }
80
81         $hotel = Hotel::create([
82             'agent_id' => session('agentId'),
83             'full_title' => $validate['full_title'],
84             'short_title' => $validate['short_title'],
85             'city_id' => $cityId,
86             'address' => $validate['address'],
87             'number_rooms' => $validate['number_rooms'],
88             'description' => $validate['description'],
89             'star' => $validate['star'],
90             'category_id' => $validate['category_id'],
91             'status_id' => 1
92         ]);
93
94         $hotelId = $hotel->id;
95         $services = HotelServices::all();
96         for ($i = 0; $i < $services[count($services) - 1]->id; $i++) {
97             if (isset($data[$i]) && $data[$i] == "on") {
98                 HotelServicesList::create([
99                     'hotel_id' => $hotelId,
100                     'service_id' => $i
101                 ]);
102             }
103         }
104
105         return redirect()->route('hotel-accommodation.end');
106     }

```

Рисунок Б.9 – Программный код сохранения данных о новом отеле

```

113 ✓      public function storeAddRoomEquipmentForHotel(Request $request)
114      {
115          $roomId = session('roomIdForNumber' . $request['room_number']);
116
117          $equipment = RoomEquipment::all();
118
119          $data = $request->all();
120          for ($i = 0; $i < $equipment[count($equipment) - 1]->id; $i++) {
121              if (isset($data[$i]) && $data[$i] == "on") {
122                  RoomEquipmentList::create([
123                      'room_id' => $roomId,
124                      'equipment_id' => $i
125                  ]);
126              }
127          }
128
129          return back()->withInput();
130      }

```

Рисунок Б.10 – Программный код сохранения техники номера

```

25 ✓      public function saveUserData(UserDataRequest $request)
26      {
27          $valid = $request->validated();
28          $usersData = DataUsers::where('user_id', Auth::user()->id)->get();
29          $id = (int)Auth::user()->id;
30
31          if (!isset($usersData->id) && count($usersData) == 0) {
32              DataUsers::create([
33                  'user_id' => $id,
34                  'surname' => $valid["surname"],
35                  'name' => $valid["name"],
36                  'gender' => $valid["gender"],
37                  'patronymic' => $valid["patronymic"],
38                  'phone_number' => $valid["phone_number"],
39                  'date_birth' => $valid["date_birth"],
40                  'passport_data_id' => 0
41              ]);
42          } else {
43              $usersData[0]->surname = $valid["surname"];
44              $usersData[0]->name = $valid["name"];
45              $usersData[0]->gender = $valid["gender"];
46              $usersData[0]->patronymic = $valid["patronymic"];
47              $usersData[0]->phone_number = $valid["phone_number"];
48              $usersData[0]->date_birth = $valid["date_birth"];
49              $usersData[0]->save();
50          }
51
52          return back()->withInput();
53      }
54

```

Рисунок Б.11 – Программный код сохранения и обновления данных пользователя

```

15 class AdminHotelController extends Controller
16 {
17     public function index() {
18         $hotels = Hotel::all();
19         $statuses = Status::all();
20         return view('admin/hotels/hotels-list', ['hotels' => $hotels, 'statuses' => $statuses]);
21     }
22
23     public function show(int $hotelId) {
24         $hotel = Hotel::find($hotelId);
25         $servicesList = HotelServicesList::where('hotel_id', $hotelId)->get();
26         $rooms = Room::where('hotel_id', $hotelId)->get();
27         $roomEquipmentLists = RoomEquipmentList::all();
28         $statuses = Status::all();
29         $agentData = Agents::where('user_id', $hotel->agent_id)->get();
30         $userData = User::find($agentData[0]->user_id);
31
32         return view('admin/hotels/hotel-list-info', ['hotel' => $hotel, 'rooms' => $rooms, 'servicesList' => $servicesList,
33     ]
34
35     public function statusSave(Request $request, int $hotelId){
36         $hotel = Hotel::find($hotelId);
37         $hotel->status_id = (int)$request["status_id"];
38         $hotel->save();
39         return redirect()->route('admin.hotels');
40     }
41
42     public function newHotels() {
43         $hotels = Hotel::where('status_id', 1)->get();
44         $statuses = Status::all();
45         return view('admin/hotels/hotels-list', ['hotels' => $hotels, 'statuses' => $statuses]);
46     }
47 }

```

Рисунок Б.12 – Программный код контроллера панели администратора

```

1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Admin;
7  use Illuminate\Support\Facades\DB;
8
9  class LoginHistoryController extends Controller
10 {
11     public function historyAdmin()
12     {
13
14         $historyUsers = DB::table('admins')
15             ->join('login_history', 'login_history.user_id', '=', 'admins.user_id')
16             ->join('users', 'login_history.user_id', '=', 'users.id')
17             ->get();
18
19         return view('admin/history/history_main', ['historyUsers' => $historyUsers, 'title' => 'Администратора']);
20     }
21
22     public function historyGuests()
23     {
24         $historyUsers = DB::table('users')
25             ->join('login_history', 'login_history.user_id', '=', 'admins.user_id')
26             ->get();
27         return view('admin/history/history_main', ['historyUsers' => $historyUsers, 'title' => 'Клиентов']);
28     }
29
30     public function historyAgents()
31     {
32         $historyUsers = DB::table('agents')
33             ->join('login_history', 'login_history.user_id', '=', 'agents.user_id')
34             ->join('users', 'login_history.user_id', '=', 'users.id')
35             ->get();
36         return view('admin/history/history_main', ['historyUsers' => $historyUsers, 'title' => 'Менеджеров отелей']);
37     }
38 }

```

Рисунок Б.13 – Программный код контролера историй входа пользователей

Приложение В
(обязательное)

Тест-кейсы

Таблица В.1 – Общая информация о тестировании

Название проекта	Holiday Home
Номер версии	holiday_home_01
Имя тестера	Мельникова Александра Михайловна
Даты тестирования	12.06.2023

Таблица В.2 – Test case #1

Test Case #	TC_UI_1
Приоритет теста	Высокий.
Название тестирования/Имя	Авторизация не зарегистрированного пользователя в системе.
Резюме испытания	Безуспешная авторизация.
Шаги тестирования	Ввод логина, ввод пароля, нажать кнопку «Авторизация».
Данные тестирования	Логин: любые данные. Пароль: любые данные. Логин и пароль не должны совпадать с уже имеющимися в бд .
Ожидаемый результат	Система не даёт доступ к работе с бизнес-процессами системы.
Фактический результат	Система не авторизовала пользователя.
Предпосылки	У пользователя нет аккаунта в системе, и он не зарегистрирован.
Постусловия	Вывод окна с ошибкой авторизации.
Статус (Pass/Fail)	Pass
Комментарии	

Таблица В.3 – Test case #2

Test Case #	ТС_UI_2
Приоритет теста	Высокий.
Название тестирования/Имя	Авторизация зарегистрированного пользователя в системе.
Резюме испытания	Успешная авторизация.
Шаги тестирования	Ввод логина, ввод пароля, нажать кнопку «Авторизация».
Данные тестирования	Логин: meneg. Пароль: 1234.
Ожидаемый результат	Система успешно авторизует пользователя и даёт доступ к работе.
Фактический результат	Система авторизует пользователя и даёт доступ к работе в роли менеджера по продажам.
Предпосылки	Пользователь зарегистрирован в системе и имеет активный аккаунт, данные о котором хранятся в бд.
Постусловия	Предоставляет доступ к работе с системой.
Статус (Pass/Fail)	Pass
Комментарии	