

Практическая работа № 1

СОЗДАНИЕ БАЗЫ ДАННЫХ

Цель работы: создать БД на сервере, разработать необходимые запросы и хранимые процедуры.

Краткие теоретические сведения

На рисунке 1 представлены структурные элементы, которые должна включать в себя готовая система:

- база данных;
- модуль для аналитика (экономиста);
- модуль для администратора БД;
- модуль для сотрудников;
- модуль для клиентов (web-сайт).

Проектирование БД – важный этап создания информационной системы. На этом этапе происходит рассмотрение предметной области, а также выделение всех сущностей БД и их описание.

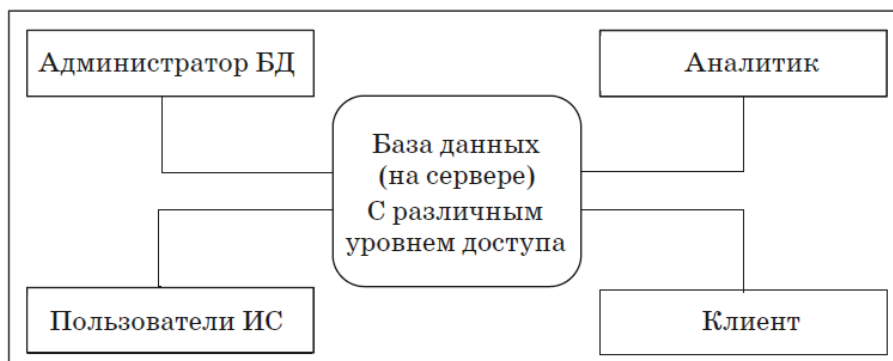


Рис. 1. Структурные элементы ИС

Как результат работы создается предварительная схема данных, между таблицами устанавливаются связи, производится нормализация отношений и приведение их к 3 НФ.

Следующий этап – это создание запросов, хранимых процедур на сервере и триггеров.

Хранимые процедуры – это набор SQL-команд, которые хранятся в базе данных под уникальным именем и в совокупности выполняют некую функцию.

При помощи хранимых процедур можно расширить синтаксис SQL так, что он будет похож на обычный язык программирования.

Триггер – это специальная хранимая процедура, привязанная к событию изменения содержимого таблицы. Существуют три события изменения таблицы, к которым можно привязать триггер: это изменение содержимого при помощи операторов insert, delete и update.

Исходные данные: вся необходимая информация содержится в описании предметной области (Приложение 1).

Порядок выполнения работы

1. Провести анализ предметной области, описать назначение системы, структуру системы.
2. Создать диаграмму вариантов использования.
3. Определить входные и выходные документы.
4. Выделить необходимые сущности, провести нормализацию.
5. Разработать БД на сервере MySQL, установить связи между таблицами, заполнить таблицы необходимыми данными.
6. Разработать необходимые запросы.
7. Разработать хранимые процедуры.
8. Разработать триггеры.

Содержание отчета

1. Цель работы.
2. Описание предметной области, структуры системы и ее назначения.
3. Диаграмма вариантов использования.
4. Определение входных и выходных документов.
5. Схема БД, описание связей между таблицами.
6. Заполненные таблицы.

Следующие пункты следует выполнять, понимая какие запросы потребуются для отображения результатов их выполнения на интерфейсах посетителя, администратора, аналитика и других пользователей информационной системы.

также эти запросы можно реализовать и при выполнении практической работы № 4.

7. Запросы: описание запроса, SQL-код, результат выполнения запроса.

8. Хранимые процедуры: описание хранимой процедуры, скрипт, результат выполнения.

9. Триггеры: описание триггеров, скрипт, результат выполнения.

10. Вывод.

11. Листинги с комментариями.

12. Список использованной литературы.

Вопросы для самопроверки:

1. Для чего нужны хранимые процедуры?

2. Что такое диаграмма вариантов использования?

3. Какие этапы включает в себя проектирование БД?

4. Что такое триггер и зачем он нужен?

ПРИМЕР ВЫПОЛНЕНИЯ № 1

1 Постановка задачи

1.1 Анализ предметной области

Анализ предметной области является начальным этапом разработки любой ИС. На этом этапе необходимо выявить пользователей системы и их потребности по функциональности. Предметной областью данной работы деятельность гостиницы.

Гостиница – это коллективное средство размещения постояльцев, имеющее единое руководство и предоставляющее набор услуг, связанных с проживанием и отдыхом [2]. Как правило, любая гостиница имеет номера различных категорий и вместимости, которые отличаются удобствами и ценой за сутки. В ИС гостиницы сохраняются сведения о сотрудниках, информация о количестве отработанных сотрудником дней и выплатах зарплаты за каждый месяц. Клиенты гостиницы снимают номера на определенный срок, причем оплата производится сразу и в полном размере, также они могут сначала забронировать номер на сайте гостиницы или по телефону. Гостиница предоставляет и дополнительные услуги, которыми клиент может воспользоваться.

От правильной организации работы информационной системы во многом зависит её эффективность. Поэтому необходимо определить основные отделы гостиницы, в которых данная система будет использоваться. К ним можно отнести отдел размещения и бронирования, отдел обслуживания номеров, руководство отеля и аналитический отдел.

Пользователями информационной системы являются:

- администратор гостиницы, регистрирующий и выселяющий постояльцев,
- аналитик-экономист, анализирующий эффективность работы гостиницы,
- сотрудники других подразделений и руководство гостиницы.

1.2 Функциональность информационной системы

Так как ИС характеризуется возможностью хранения данных, в её структуру должна быть включена база данных (БД). В базе должна храниться информация о комнатах, сотрудниках и клиентах, а так же история посещений гостиницы и учет дополнительных услуг.

Помимо базы данных в ИС необходимо реализовать удобный пользовательский интерфейс как для клиента, так и для сотрудников. Необходимость в разработке клиентского сайта вытекает из желания клиентов получать актуальную информацию о гостинице дистанционно и самостоятельно.

Функционирование ИС без доступа к ней сотрудников невозможно, поэтому требуется разработать интерфейс для администратора, экономиста, руководства и других сотрудников гостиницы. Так как различные группы пользователей имеют разные возможности доступа к БД, необходимы средства защиты от несанкционированного доступа.

Для клиента на сайте необходимо реализовать следующие возможности:

- возможность регистрации и авторизации на сайте;
- просмотр информации о гостинице;
- просмотр новостей гостиницы;
- просмотр отзывов и возможность добавления своего отзыва;
- возможность просмотра информации о номерах;
- возможность бронирования номера на определенную дату;
- просмотр бронирований и оставленных отзывов.

При этом возможность бронирования имеется только у зарегистрированных пользователей. При регистрации клиенту необходимо указать персональные данные, а так же логин и пароль для доступа к сайту.

Для администратора гостиницы должны быть предусмотрены возможности, среди которых:

- авторизация в ИС во избежание несанкционированного доступа;
- просмотр личного кабинета;
- учет заселения клиентов;

- возможность бронирования номера для клиента;
- возможность внесения информации о клиенте;
- учет дополнительных услуг;
- просмотр свободных номеров на определенную дату.

Для пользователя экономист-аналитик необходимы следующие возможности:

- авторизация в ИС во избежание несанкционированного доступа;
- просмотр личного кабинета;
- просмотр журнала учета посещений;
- просмотр журнала учета дополнительных услуг;
- просмотр журнала учета выплат сотрудникам;
- возможность визуализации данных (построения графиков) с помощью приложения.

1.3 Обзор существующих аналогов

В настоящее время успешно реализуют свою деятельность отечественные фирмы по автоматизации деятельности гостиниц. По данным маркетингового агентства, специализирующегося на исследовании в области туристического бизнеса, TourBC ведущими системами автоматизации гостиниц являются такие системы как Bnovo PMS, Frontdesk24 и HMA Eсvi [3].

Популярной системой управления гостиницей является система Bnovo PMS от российской компании Bnovo. Данная ИС позволяет управлять онлайн-продажами гостиничных номеров. В её состав входит модуль онлайн-бронирования на сайте отеля, так же менеджер каналов, позволяющий подключить множество онлайн-каналов продаж.

Однако минусом системы является отсутствие возможности быстрого самостоятельного знакомства с системой. Для демонстрации работы необходимо сотрудника компании Bnovo [3]. Интерфейс администратора гостиницы представлен на рисунке 1.

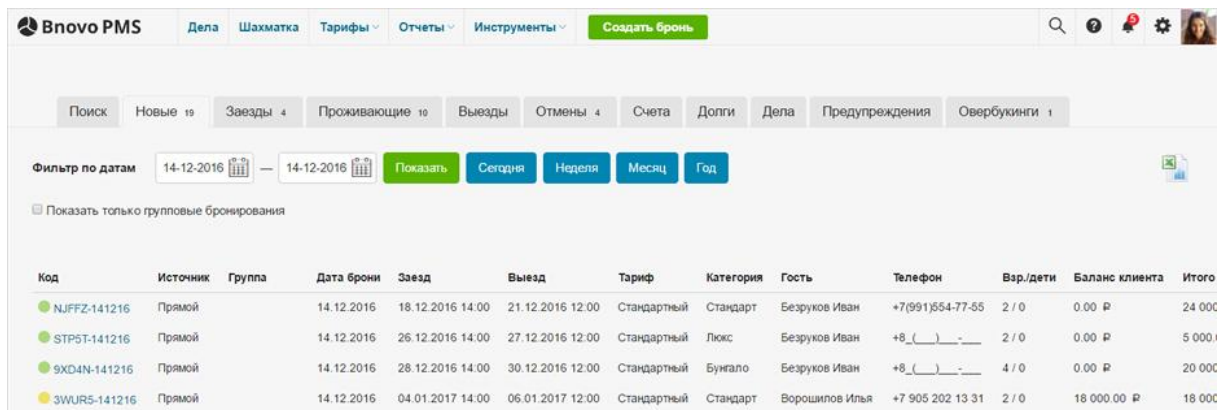


Рисунок 1 – Интерфейс ИС Bnovo PMS

Система Frontdesk24 присутствует на рынке гостиничных систем с 2012 года. Программа используется для автоматизации отелей, хостелов или апартаментов. В системе имеется модуль онлайн-бронирования для сайта, так же менеджер каналов, осуществляющим продажу номеров, используя площадки онлайн-бронирования [3]. Интерфейс представлен на рисунке 2.

Сохранить | Рассчитать | Печатные формы | Действия | Цена рассчитана | История изменений

№ 10 от 15.05.2017 14:02

Заявка: * 15.05.2017 Пн в 12:00 в 12:00

Выезд: * 25.05.2017 Чт в 12:00 в 12:00

10 ночей

Категория: * Стандарт

Номер: 01

Статус: Оплата

Тариф: Весна

Контрагент: не указан

Скидка: 0,00%

Всего к оплате: 0,00

Договор: не указан

Цена: 25 000,00

Оплачено: 0,00

Не оплачено: 0,00

Показать расчет

Бланк:

Ответственный:

Расчет цены:	15 май	16 май	17 май	18 май	19 май	20 май	21 май	22 май	23 май	24 май	25 май	Итого
Курочкина В.	2 500	2 500	2 500	2 500	2 500	2 500	2 500	2 500	2 500	2 500	0	25 000
	0	0	0	0	0	0	0	0	0	0	0	0

Комментарий:

Добавить | Удалить

Гостей на осн. месте: 2

Гостей на доп. месте: 0

Тип питания:

Гости							
#	Фамилия	Имя	Отчество	Возраст	Тип питания	Размещение	#
Редактировать	Курочкина	Вероника				Основное	Данные в ФМС

Добавить

Поручения					
#	Дата и время	Исполнитель	Статус	Текст	Выполнено
Добавить Добавить доп. услугу Удалить					

Начисления			
#	Дата	Сумма	Комментарий
0,00			

Платежи				
#	Дата	Сумма	Тип	Комментарий
0,00				

Рисунок 2 – Интерфейс ИС Frontdesk24

HMA Eсvi (Hotel Manager Assistant Eсvi) система с набором многих необходимых инструментов для управления гостиницей. Программа интегрируется веб-сайтом гостиницы и каналами продаж.

К существенному недостатку системы можно отнести низкую скорость загрузки некоторых функциональных модулей [3]. Интерфейс системы представлен на рисунке 3.

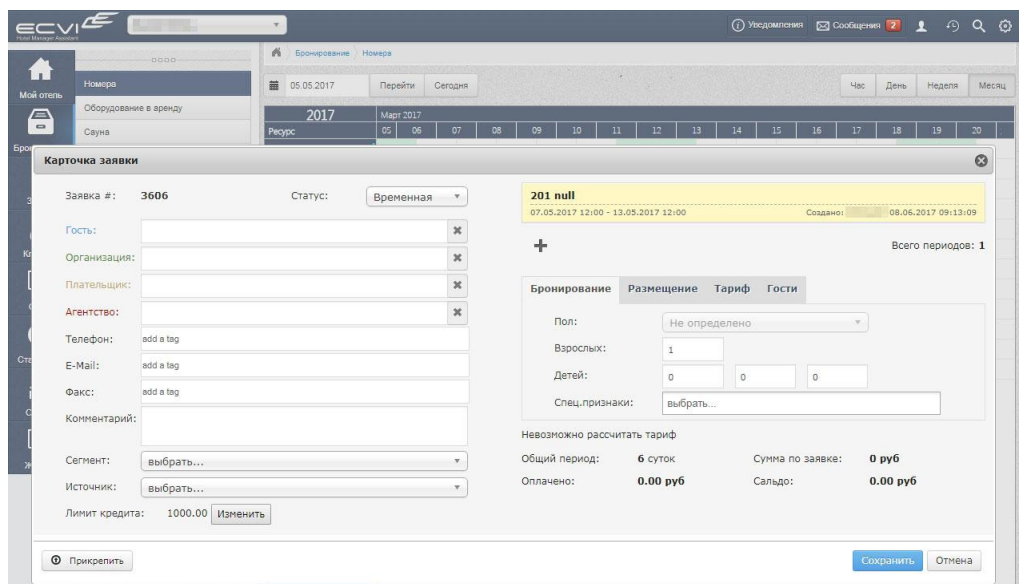


Рисунок 3 – Интерфейс ИС НМА Eсvi

Для оценки функциональности обозреваемых систем были выделены такие критерии сравнения как: функции портье, управление номерным фондом, статистика и финансовая отчетность, онлайн-бронирование, учет гостей, поддержка продукта, удобство интерфейса, возможность самостоятельного внедрения, возможность работы на мобильных устройствах, скорость работы. Результат сравнения представлен в таблице 1.

Таблица 1 – Сравнение систем управления гостиницей

Критерии сравнения	Bnovo PMS	Frontdesk24	НМА Eсvi
Функции портье	+	+	+
Управление номерным фондом	+	+	+
Онлайн-бронирование	+	+	+
Ведение базы данных гостей	+	+	+
Статистика и финансовая отчетность	+	+	
Поддержка продукта	+	+	+
Самостоятельное внедрение продукта	—	+	+
Использование на мобильных устройствах	+	—	+
Высокая скорость загрузки системы	+	+	—
Визуализация статистических данных	+	—	—

Следует отметить, что стоимость таких систем довольно высока. В качестве примера, стоимость Vnovo PMS позволяет выбрать количество подключаемых услуг, а так же период доступа, вне зависимости от количества номеров в отеле. Так, базовый тариф, не включающий в себя онлайн-каналы продаж и модуль онлайн-бронирования обойдется в 840 руб. в месяц, а премиум, не имеющий ограничений на количество подключаемых онлайн-каналов продаж – 9450 руб. в месяц [3].

Важным недостатком перечисленных систем является высокая стоимость продукции и внедрения, а также обязательная необходимость в обслуживании специалистами. Другим недостатком является большое количество функций, не все из которых востребованы. Также достаточно сложно обучить персонал для работы в этих системах.

Данная работа посвящена разработке системы, простой и удобной в использовании, содержащей в себе основной и необходимый функционал:

- возможность онлайн-бронирования,
- управление бронированием и размещением клиента (функции портье),
- управление номерным фондом,
- ведение базы данных клиентов,
- учет сотрудников,
- управление отделом обслуживания номеров,
- визуализация статистических данных.

Помимо перечисленного, преимуществом системы является её адаптивность – возможность использования на мобильных устройствах, а также дружелюбный интерфейс, позволяющий минимизировать затраты на внедрение продукта и обучение пользователей. Важной особенностью системы является защита данных от несанкционированного доступа.

1.4 Варианты использования системы

Диаграммы вариантов использования описывают взаимодействие между вариантами использования системы и действующими лицами (актеры), а так же отражает требования к системе с точки зрения пользователя. Варианты использования являются функциями системы, а действующие лица – это лица,

имеющие отношение к системе и использующие её функции. Диаграммы показывают какие актеры инициируют варианты использования [4].

На основании описания предметной области можно выделить следующих основных актеров, взаимодействующих с системой:

- администратор гостиницы;
- клиент;
- аналитик.

Диаграмма вариантов использования для администратора гостиницы представлена на рисунке 4. Для администратора возможны следующие варианты использования системы:

- авторизация в системе;
- просмотр информации о номерах;
- занесение информации о клиентах;
- занесение информации о бронировании номера, заселении и выселении постояльца;
- внесение информации об оказанных услугах.



Рисунок 4 – Диаграмма вариантов использования для администратора гостиницы

Диаграмма вариантов использования для аналитика гостиницы представлена на рисунке 5. Для аналитика выделены следующие варианты использования:

- сбор данных;
- анализ данных;
- просмотр визуализированных данных (графиков и диаграмм).



Рисунок 5 – Диаграмма вариантов использования для аналитика

Диаграмма вариантов использования для клиента представлена на рисунке 6. При этом для него выделены следующие варианты использования:

- регистрация;
- авторизация;
- просмотр информации о номерах;
- просмотр новостей;
- бронирование номера;
- просмотр отзывов;
- добавление отзыва.

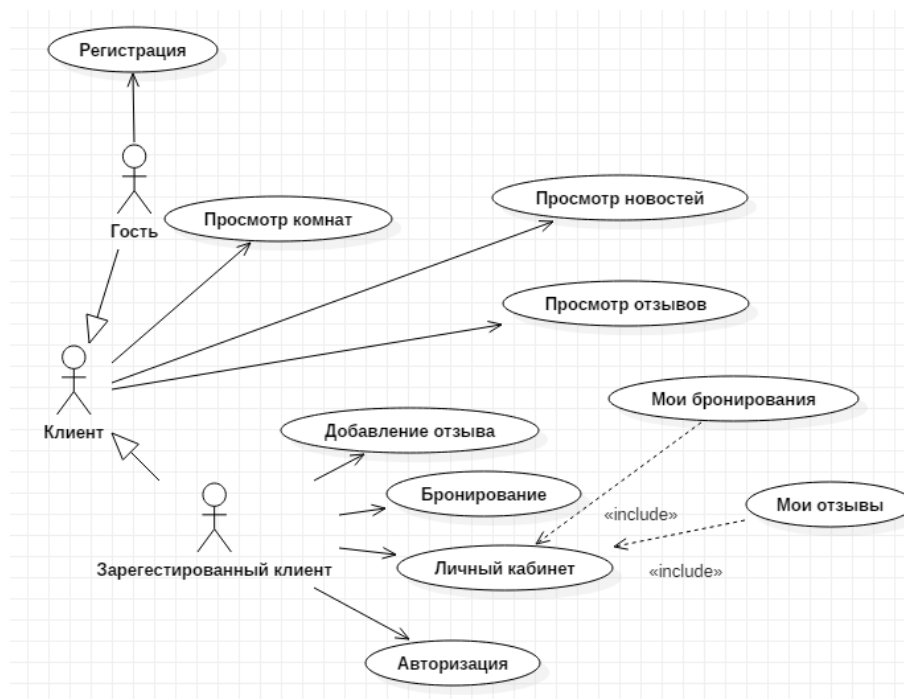


Рисунок 6 – Диаграмма вариантов использования для клиента гостиницы

1.5 Выбор средств разработки

База данных является основой любой информационной системы. Для проектирования базы данных существует множество средств. Одним из наиболее широко известных средств является ERWin Data Modeler. Именно это средство чаще всего используется профессиональными разработчиками при создании модели данных.

ERwin предлагает два уровня представления данных: логический и физический. Логический отражает сущности, атрибуты и связи между ними, а для физической модели можно выбрать конкретную СУБД и настроить типы атрибутов. В ERWin осуществлена возможность создания спроектированной структуры непосредственно в СУБД, что значительно упрощает и ускоряет создание БД.

Для работы с базами данных используются СУБД – комплекс языковых и программных средств, предназначенных для создания, ведения и совместного использования БД несколькими пользователями [5]. Выбор СУБД является важным этапом при создании ИС, так как именно СУБД позволяет создавать базу, редактировать её, добавлять данные и иметь к ним доступ.

При выборе СУБД необходимо учитывать следующие факторы:

- поддержка доступа для нескольких пользователей,
- возможность коммерческого использования,
- поддержка хранимых процедур и триггеров,
- наличие среды для настройки и администрирования БД.

Для создания БД в данной работе была выбрана СУБД MySQL. MySQL – СУБД, которая поддерживает структурированный язык запросов (SQL) и может применяться в качестве SQL-сервера. Для администрирования используется среда MySQL Workbench.

К преимуществам СУБД MySQL можно отнести:

- её простоту и расширяемость,
- поддержку большого количества функций SQL,
- наличие продвинутых функций для защиты данных,
- масштабируемость и легкость работы с большими объемами данных,
- скорость и производительность за счет упрощения некоторых стандартов[6].

Для разработки пользовательского интерфейса был выбран объектно-ориентированный язык программирования C# и среда MS Visual Studio 2017, включающая интегрированную среду разработки и множество полезных инструментов. Главными преимуществами Visual Studio 2017 являются производительность и удобство использования.

К другим преимуществам среды Visual Studio можно отнести:

- возможность интеграции с MySQL;
- поддержка технологии .NET Framework;
- поддержка нескольких языков и различных типов проектов.

Специально для платформы .NET разрабатывался язык C#, обладающий рядом преимуществ, таких как хорошая организация и строгость языка, логичность конструкций. Благодаря мощной библиотеке классов платформы .NET имеется возможность решать сложные задачи, игнорируя массу рутинных операций и используя готовые конструкции. C# имеет полный хорошо определенный набор основных типов данных [7].

2 Проектирование и реализация информационной системы

2.1 Структурная схема информационной системы

По сравнению с классическими информационными системами веб-системы обладают следующими особенностями.

- Кроссплатформенностью. Позволяют работать с вычислительными мощностями и программным обеспечением, которые находятся в распоряжении конечного пользователя. Для работы с веб-системой используют любую операционную систему, например, Microsoft Windows, MAC, Linux, и браузер, в частности, Google Chrome, Internet Explorer, Opera, FireFox.

- Отсутствием фазы развертывания. Установка приложения происходит только на стороне веб-сервера. Подключение клиента происходит после набора URL – фактического адреса веб-приложения в браузере.

- Отсутствием версионности. Внедрение модифицированной веб-системы, доступ к ее функциональным возможностям и данным происходят сразу после загрузки программного продукта на сервер вместо его предыдущей версии.

- Централизованным хранением и защитой информации. Безопасность данных осуществляется средствами специализированных технических средств и программного обеспечения на стороне сервера веб-приложения. При этом веб-система имеет единую точку входа, информационную безопасность которой настраивают централизованно.

- Отсутствием хранилища или базы данных на вычислительном устройстве пользователя. Для адекватного отображения информации, с которой работает потребитель, достаточно иметь любой тип портативного или локального устройства, подключенного по сети к интернет-ресурсу, вычислительные мощности которого позволяют работать с полученными данными.

- Масштабируемостью. Увеличение количества подключенных абонентов происходит без переписывания программного кода приложения, как правило, настройкой аппаратных средств поддержки веб-ресурса. Добавление функциональных и нефункциональных возможностей веб-системы происходит за счет подключения новых модулей.

- Низкими эксплуатационными расходами связаны с отсутствием необходимости поддерживать дорогостоящее обслуживание и постоянную модернизацию вычислительной техники персонала предприятия для корректной работы программного обеспечения.

- Доступом из любой точки мира. Для получения информации и управлением бизнес-процессами требуется только подключения рабочего места к веб-серверу.

- Наличием большого количества пользователей, имеющие практические навыки эффективной работы в Интернете с веб-ресурсами.

Информационная система состоит из двух веб-приложений, одно из которых ориентировано на сотрудников гостиницы, а второе – на клиентов. На

рисунке 7 представлена структурная схема информационной системы гостиницы.

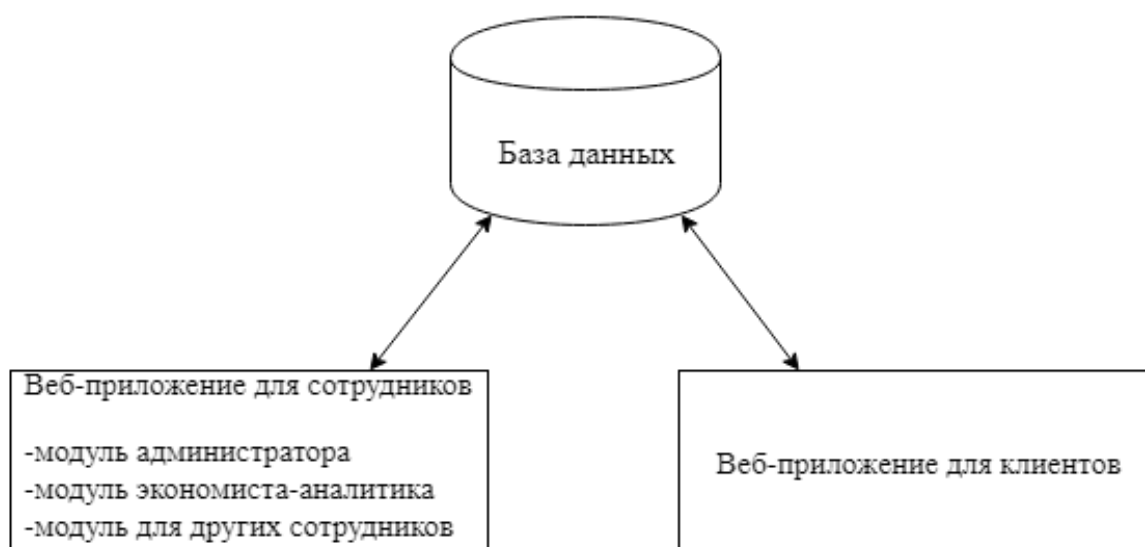


Рисунок 7 – Структурная схема ИС

Архитектура веб-приложений строится на основе трехуровневой модели взаимодействия «клиент-сервер». Такой вид архитектуры подразумевает, что пользовательским интерфейсом является браузер, который установлен на персональном компьютере пользователя (рис. 8). Веб-сервер выполняет роль сервера приложений. Он является клиентом для сервера базы данных и обеспечивает доступ к информационным ресурсам SQL-сервера – серверной системы управления базами данных (СУБД), поддерживающей структурированный язык запросов SQL (Structured Query Language).



Рисунок 8 – Архитектура ИС в web-среде

2.2. Проектирование базы данных

База данных – это важнейший компонент любой ИС, так как содержит структурированную информацию о деятельности предприятия и позволяет не только использовать ИС, а также модифицировать ее и развивать в дальнейшем. БД включает в себя набор постоянно хранимых данных, используемых для того, чтобы рационализировать ведение отчетности на предприятии.

База данных информационной системы, проектируемой в данной работе, строится на основе ER-модели. К основным понятиям данной модели можно отнести понятия сущности, связи и атрибута. Сущность – это некоторый объект, информация о котором должна храниться в БД. В процессе анализа предметной области были выделены следующие сущности:

- заказы (orders);
- сотрудники (employees);
- должности (positions);
- должности сотрудников (empl_pos);

- номера (rooms);
- типы номеров (room_type);
- клиенты (clients);
- дополнительные услуги (suppl_services);
- услуги, оказанные клиентам (client_services);
- отзывы клиентов (reviews);
- статусы заказа (states);
- способы оплаты (pay_methods);
- новости (news);
- выплаты зарплаты (payroll);
- хозяйственные задания (housekeeping_jobs);
- состояния заданий (jobs_states).

Атрибут является свойством сущности. Опишем для каждой сущности атрибуты и зададим ключевые поля.

Сущность «orders» - заказы, оформляемые при заселении клиента или при бронировании клиентом номера. О заказах необходимо сохранять следующую информацию: код клиента, код номера, код сотрудника, дата заселения, дата выселения, состояние заказа, дата бронирования, метод оплаты и общая стоимость заказа. Атрибуты данной сущности представлены в таблице 2.

Таблица 2 – Атрибуты сущности «orders»

Название	Тип данных
id_order	счетчик (int)
id_client	числовой (int)
id_room	числовой (int)
id_employee	числовой (int)
date_in	дата (date)
date_out	дата (date)
id_state	числовой (int)
booking_date	дата (date)
id_method	числовой (int)

total_cost	числовой (float)
------------	------------------

Сущность «employees» создана для описания каждого сотрудника гостиницы. К атрибутам данной сущности можно отнести: фамилию, имя, отчество сотрудника, дату рождения, номер телефона, серию и номер паспорта, email, логин и пароль для авторизации в ИС, дата приема на работу. Атрибуты сущности «employees» представлены в таблице 3.

Таблица 3 – Атрибуты сущности «employees»

Название	Тип данных
id_employee	счетчик (int)
surname	символьный (char)
first_name	символьный (char)
patronymic	символьный (char)
dob	дата (date)
phone_num	символьный (char)
passport	символьный (char)
email	символьный (char)
login	символьный (char)
passw	символьный (char)
hire_date	дата (date)

Сущность «positions» необходима для описания должностей, существующих в данной гостинице. О должностях в базе данных должна сохраняться следующая информация: название должности и зарплата за рабочий день (таблица 4).

Таблица 4 – Атрибуты сущности «positions»

Название	Тип данных
id_position	счетчик (int)
position_name	символьный (char)
salary	числовой (float)

Каждый сотрудник может иметь одну или несколько должностей. Для сохранения этой информации в БД была создана сущность «empl_pos» – должности сотрудников (таблица 5).

Таблица 5 – Атрибуты сущности «empl_pos»

Название	Тип данных
idep	счетчик (int)
id_employee	числовой (int)
id_position	числовой (int)

Гостиница имеет номерной фонд, комнаты в котором имеют номера и различаются по типу. Для описания номера гостиницы была создана сущность «rooms». Атрибуты представлены в таблице 6.

Таблица 6 – Атрибуты сущности «rooms».

Название	Тип данных
id_room	счетчик (int)
r_num	символьный (char)
id_type	числовой (int)

Номера в гостинице имеют разные категории. Для описания категории номера в БД имеется сущность «room_type», свойствами которой являются: название типа, количество мест, стоимость в сутки (таблица 7).

Таблица 7 – Атрибуты сущности «room_type»

Название	Тип данных
id_type	счетчик (int)
r_type	символьный (char)
beds_num	числовой (int)
day_cost	числовой (float)

О клиентах в базе данных сохраняется следующая информация: фамилия, имя и отчество клиента, дата рождения, серия и номер паспорта, номер телефона, email, логин и пароль для авторизации на сайте (таблица 8).

Таблица 8 – Атрибуты сущности «clients»

Название	Тип данных
----------	------------

id_client	счетчик (int)
surname	символьный (char)
first_name	символьный (char)
patronymic	символьный (char)
dob	дата (date)
phone_num	символьный (char)
passport	символьный (char)
email	символьный (char)
login	символьный (char)
passw	символьный (char)

В гостинице клиентам оказываются дополнительные услуги. О них в БД фиксируется название и стоимость за единицу. Информация об услугах описывается сущностью «suppl_services». Атрибуты представлены в таблице 9.

Таблица 9 – Атрибуты сущности «suppl_services»

Название	Тип данных
id_service	счетчик (int)
service_name	символьный (char)
cost	числовой (float)

Услуги, оказанные определенному клиенту, так же сохраняются в базе. Об оказанных услугах сохраняется следующая информация: клиент, которому услуга была оказана, количество услуг, дата оказания услуги и общая стоимость. Для этого создана сущность «client_services» (таблица 10).

Таблица 10 – Атрибуты сущности «client_services»

Название	Тип данных
idcs	счетчик (int)
id_client	числовой (int)
id_service	числовой (int)
amount	числовой (int)
id_employee	числовой (int)
service_date	дата (date)

total	числовой (float)
-------	------------------

В информационной системе предусмотрена возможность добавления отзыва о гостинице. В таблице новости необходимо сохранять информацию о клиенте, добавившем отзыв, дате добавления отзыва, его заголовок, текст отзыва, рейтинг, выбранный клиентом. Сущность «reviews» имеет атрибуты, представленные в таблице 11.

Таблица 11 – Атрибуты сущности «reviews»

Название	Тип данных
id_review	счетчик (int)
review_title	символьный (char)
id_client	числовой (int)
review	символьный (char)
review_date	дата (date)
rating	числовой (int)

Клиент может оплачивать проживание в гостинице наличным и безналичным способом. Сущность «pay_methods» имеет атрибуты, представленные в таблице 12.

Таблица 12 – Атрибуты сущности «pay_methods»

Название	Тип данных
id_method	счетчик (int)
method_name	символьный (char)

На сайте отеля размещаются новости. В базе данных сохраняется заголовок и текст новости, дата её размещения, а так же информация о сотруднике, разместившем её. Атрибуты сущности «news» показаны в таблице 13.

Таблица 13 – Атрибуты сущности «news»

Название	Тип данных
id_news	счетчик (int)
news_headers	символьный (char)

news	символьный (char)
n_date	дата (date)
empl	числовой (int)

В базе данных хранится информация о ежемесячных выплатах заработной платы сотрудникам: дата платежа, количество отработанных дней, количество дней оплачиваемого отпуска, сумма премии и общая сумма выплаты. Атрибуты сущности «payroll» представлены в таблице 14.

Таблица 14 – Атрибуты сущности «payroll»

Название	Тип данных
id_payroll	счетчик (int)
idep	числовой (int)
dateofpay	дата (date)
num_workdays	числовой (int)
num_paidleaves	числовой (int)
bonus	числовой (float)
result	числовой (float)

В базе данных содержится информация о хозяйственных заданиях для сотрудников – сущность «housekeeping_jobs» (таблица 15) и состоянии выполнения этих заданий – сущность «jobs_states» (таблица 16). О заданиях сохраняется содержание задания, исполнитель, состояние выполнения, дата размещения и номер комнаты.

Таблица 15 – Атрибуты сущности «housekeeping_jobs»

Название	Тип данных
id_job	счетчик (int)
job_note	символьный (char)
id_doer	числовой (int)
id_state	числовой (int)
job_date	дата (date)
id_room	числовой (int)

Таблица 16 – Атрибуты сущности «jobs_states»

ПРИМЕР ВЫПОЛНЕНИЯ № 2

Архитектура и среда реализации проектируемой информационной web-системы

2.1 Выбор архитектуры проектируемой web-системы

Web-сервер должен обладать возможностью многопользовательского доступа. Чтобы добиться этого используют архитектуру вида «Клиент - Сервер». Схема представлена на рисунке 5.

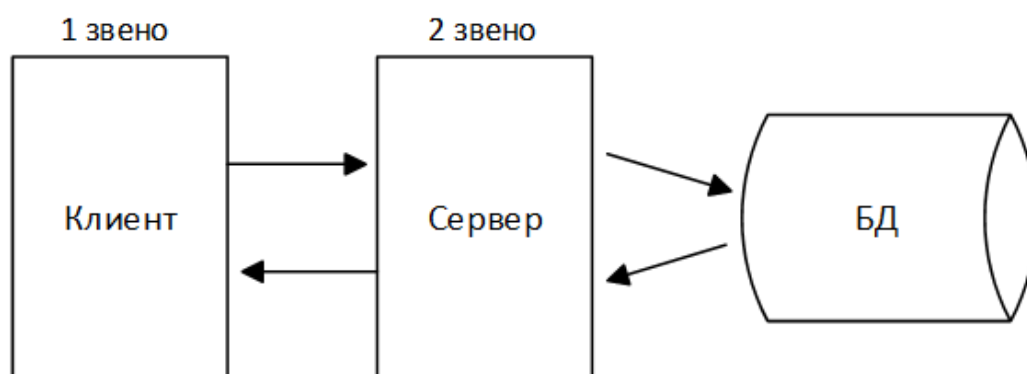


Рисунок 5 – Схема архитектуры «клиент-сервер»

Клиентская часть – это то, что видит пользователь в своем браузере. Формируется запрос для серверной части и отправляется по протоколам из сети Интернет. Серверная часть, в свою очередь состоит из Web-сервера и СУБД. Web-сервер принимает запросы обрабатывает их, создает ответы и отправляет пользователю[9]. Ответы отображаются пользователю в соответствии с его запросами, а так же в соответствии с созданным интерфейсом. При использовании данной архитектуры «клиент – сервер» основным достоинством является то, что на стороне пользователей системы не требуется никакого дополнительного программного обеспечения, кроме Интернет-браузера [4]. Код создается на стороне сервера. В СУБД используется и создается БД. В базе данных храниться информация о досуговых учреждениях, их составляющих и о событиях.

2.2 Диаграмма вариантов использования

Сейчас эффективно визуально проектировать ИС при помощи UML (Unified Modeling Language). Диаграмма – это графическое представление

множества объектов, которое подчиняется нотации UML и изображается в соответствии с ней [6]. Действующие лица, поведение системы все это легко отобразить здесь. Для описание функциональной модели строим диаграмму вариантов (use case diagram), на которую заносятся отношения между действующими лицами(актерами). Всё это делается для того, чтобы выявить общие требования к поведению ИС и разработать модель системы для ее последующей детализации. Выделим следующих основных актеров, взаимодействующих с системой:

- администратор БД;
- пользователь;
- контент-менеджер.

На рисунке 6 мы видим варианты использования для администратора БД. Он может:

- присваивать права пользователям;
- редактировать содержимое БД;
- управлять ИС;
- обновлять и добавлять данные в БД.

На рисунке 7 представлена диаграмма для контент-менеджера. Он в свою очередь может:

- добавлять новые записи в список мест;
- обрабатывать существующие записи в БД.

На рисунке 8 представлена диаграмма для клиентов.

Пользователь может:

- просматривать информацию на сайте;
- просматривать личный кабинет;
- оставлять комментарии.

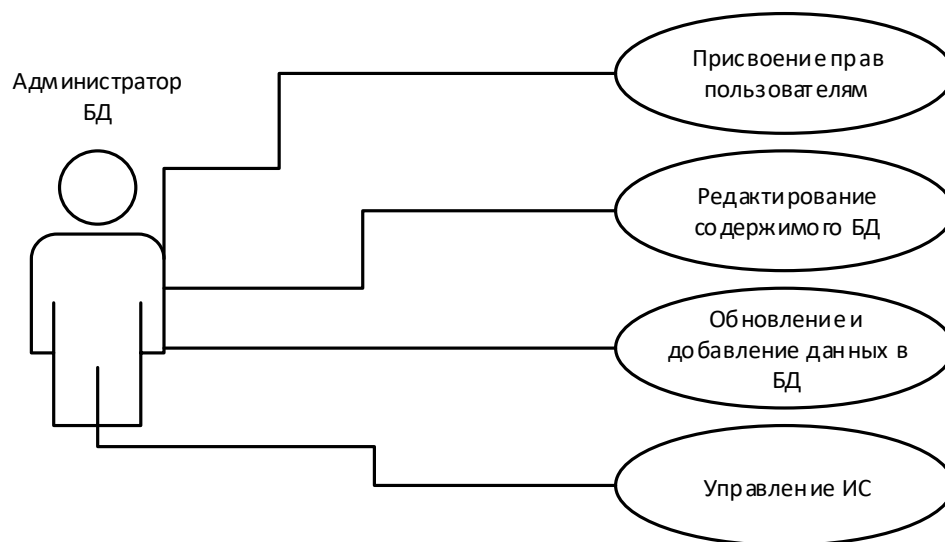


Рисунок 6 – Диаграмма для администратора БД

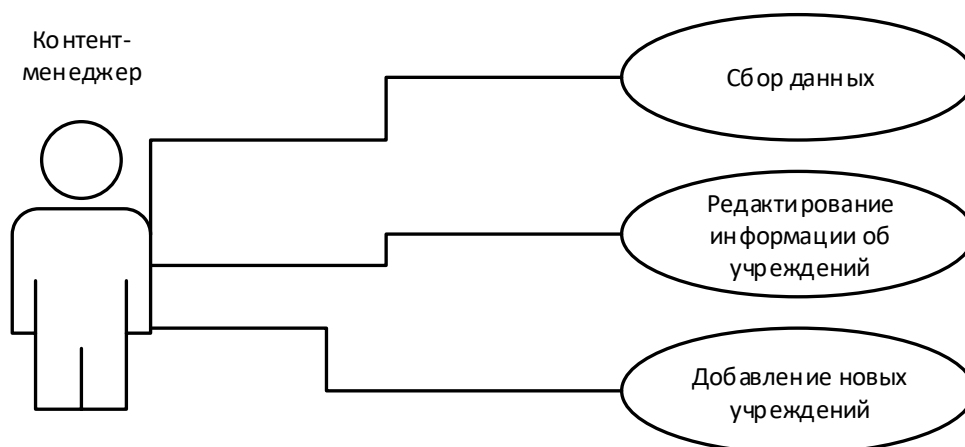


Рисунок 7 – Диаграмма для менеджера

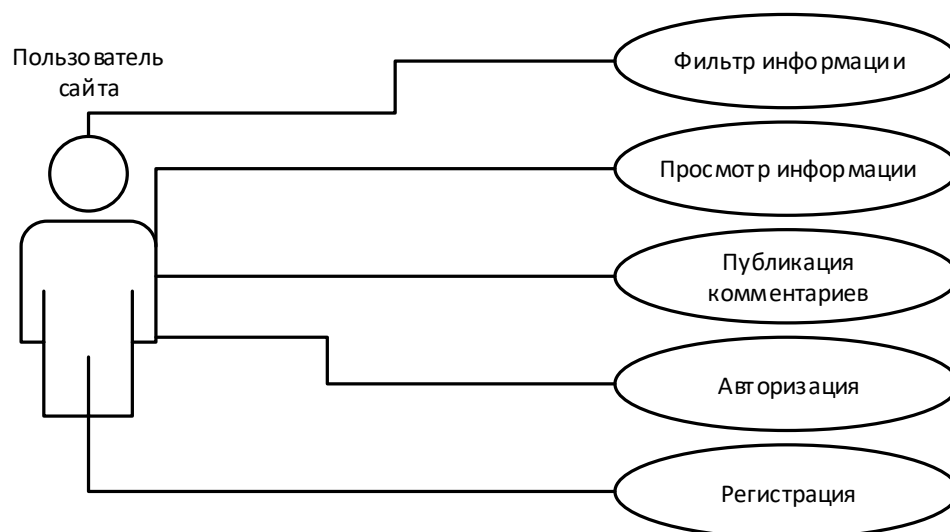


Рисунок 8 – Диаграмм для пользователя

2.3. Обоснование выбора программных средств реализации

web - приложения

Рынок информационных технологий насыщен программным обеспечением от различных производителей:

- **Apache** — является самым распространенным сервером в сети. Считается более гибким и надежными по отношению его аналогам. Он крайне нетребователен к ресурсам и способен обслуживать несколько сайтов. Приложение обслуживает большинство известных операционных систем. Сейчас именно Apache используют 71 % пользователей;

- **Microsoft IIS** — Сервер от такой кампании, как Microsoft. Его же использует примерно 14% пользователей. Он менее гибкий и поддерживает всего два языка программирования (*VBScript и JScript*);

- **NGINX** Самый популярный сервер именно в Российском интернете. Он самый простой в познании и не имеет лишних функций. **NGINX** входит в тройку самых популярных веб серверов в мире. Его использование составляет около 6,5 %.

Для реализации поставленной задачи был выбран web-сервере apache. Он является самым популярным и распространённым в сети интернет (ссылка на исследование популярности апаш). Данный web - сервер гибкий в настройках и имеет возможность подключения новых модулей. На нём можно разрабатывать сайты на таких языках программирования как: PHP, Python, Ruby, Perl, ASP и Tcl.

В настоящее время программисты используют интегрированные среды разработки, в которые входит, кроме локального сервера Apache, множество других сред разработки: компилятор PHP, компоненты для работы с базами данных и др. Одной из таких интегрированных сред разработки является **XAMP**, под операционную систему Windows. Он распространяется бесплатно и поддерживает работу в большинстве современных операционных систем. Обладает интуитивно-понятным пользовательским интерфейсом.

Среди систем управления баз данных (СУБД), на сегодняшний день, самыми популярными являются PostgreSQL, MySQL, MS SQL Server.

PostgreSQL соответствует SQL-стандартам ANSI/ISO (ссылку, откуда). К преимуществам PostgreSQL можно отнести:

- База данных не ограничена в размере;
- Полная совместимость SQL;
- Поддерживаемые драйверы: JDBC, ODBC, .NET, PL/pgSQL, PHP, Perl, Python, Ruby, C/C++, Java.

– Поддерживаемые типы данных:

- Числовые;
- Даты/времени;
- Символьный тип;
- Двоичные данные;
- Данные XML.

MS SQL Server – разработана корпорацией Microsoft. Используется для работы с персональными данными, а так же с данными больших промышленных предприятий. Из минусов можно выделить работу исключительно на операционной системе Windows. Преимущества использования MS SQL:

- Размер базы ограничен исключительно его хранилищем;
- 1024 столбца максимально допустимое количество хранимых столбцов;
- Поддерживаемые драйверы: JDBC, ODBC, .Net, PHP, Node.js, Python, Ruby.

Поддерживаемые типы данных:

- Числовые;
- Даты/времени;
- Символьный тип;
- Двоичные данные;
- Данные XML;
- Пространственные.

MySQL одна из самых популярных СУБД. Очень простая в изучении, а также имеет хорошую информационный справочный материал. Приложения общаются с базой данных через процесс-демон [1].

Преимущества MySQL:

- Простота в установке;
- Имеется визуализированный интерфейс;
- Поддержка функционала SQL;
- Хорошая защита информации;
- Может работать с большими объёмами данных;
- Хорошая производительность;
- Поддерживаемые драйверы: ODBC, JDBC, .NET, Python, Perl, PHP, Ruby, C/C++.

Поддерживаемые типы данных:

- Числовые типы;
- Дата/время;
- Символьные типы;
- Булев тип;
- Двоичные данные.

Моим выбором стала СУБД MySQL, как самая быстрая и лёгкая в эксплуатации. Большинство web-приложений разработаны на основе MySQL, так как она славится своей проверенной производительностью, надёжностью и простотой использования. Также её часто выбирают в качестве встроенной базы данных, распространяемой тысячами поставщиков программного обеспечения и производителей оборудования (<https://oracle.com/ru/mysql/>).

Визуальное проектирование разрабатываемой базы данных выполним в среде MySQL Workbench. Для работы администраторов базы данных используем Microsoft Access 2013, потому что он входит в состав офисного программного обеспечения Microsoft Office. Импортировать таблицу из MySQL можно с помощью протокола Open Database Connectivity.

3 Описание основной функциональности web-сайта

3.1 Выбор языка web-программирования

Самыми популярными языками для написания Web-сайта сейчас выступают [2]: HTML, CSS, Java Script и PHP. Базовым языком разметки создания сайтов, является HTML. В пятой версии HTML улучшен уровень

поддержки мультимедиа-технологий, разработанный код представлен в виде удобном для чтения. HTML5 – теговый язык разметки документов именно он был выбран языком разметки, так как считается более удобной и функциональной версией HTML. Но если сайт создается исключительно на HTML, то он будет статичен, информация занесенная на сайт путем вставки в код будет читаться, но чтобы ее исправить, нужно каждый раз переделывать код. Поэтому при разработке сайта предполагается использование CSS.

CSS – язык описывающий внешний вид Web-сайта, созданный с использованием HTML. Он используется при описании положения отдельных блоков Web-ресурса на странице, а также через него можно задавать цвета фона или текста и других аспектов представления внешнего вида Web-страниц. При хорошем владении CSS, можно представить одну и ту же страницу в различных стилях или методах вывода страницы на экран, например, экранное представление или печатное представление, а также возможно представить страницу в шрифте Брайля, имея специальное устройство вывода [3]. На данный момент, актуальной является третья версия CSS. Преимуществом последней версии является, сокращение времени разработки, за счет упрощения элементов кодировки. Поэтому вместе с использованием HTML5 было принято решение использовать и CSS3.

Основным же языком для нашего сайта будет выступать PHP. Изначально он разрабатывался как язык программирования Интернет-приложений. К основным достоинствам можно отнести:

- Код легко встраивается в страницу без необходимости подключения сторонних библиотек;
- Быстрое взаимодействие с уже выбранной СУБД MySQL;
- PHP генерирует конечные html-страницы, которые отправляются в браузер клиента.

Все перечисленные положительные качества языка PHP свидетельствуют о том, что данный язык оптимально подходит для реализации поставленной задачи. Поэтому наилучшим, на мой взгляд, языком программирования для разрабатываемой системы является именно язык PHP.

3.2 Итоговый набор языков программирования web-системы

В этом разделе приведён итоговый набор программных средств, необходимый для реализации унифицированного Web-сайта.

Для создания клиентского приложения в ВКР будем использовать:

- 1) Язык программирования: PHP ;
- 2) Языки вёрстки: HTML5 и CSS3;
- 3) Язык программирования: Java Script и его библиотека JQuery.

Для реализации серверной части выбираем следующие программные среды:

- 1) Web-сервер Apache;
- 2) СУБД MySQL.

Программная среда для написания и отладки программных кодов:

Редактор кода: Sublime Text.

4.1 Проектирование базы данных

В ходе выполнения данной работы была разработана база данных, предназначенная для извлечения данных на наш информационный сайт культурно-развлекательных учреждений. Создание таблиц осуществлялось в СУБД MySQL с оболочкой MySQL Workbench. Название и описание созданных таблиц представлены в таблице 1. Схема базы данных – рисунок 9.

Таблица 1 – Состав таблиц базы данных разработанного приложения

Название	Сведения о содержимом таблиц
category	Содержит информацию о категориях культурно-развлекательных учреждений.
mesto	Содержит информацию о культурно-развлекательных учреждениях города Санкт-Петербурга.
rayon	Содержит наименования районов города.
metro	Содержит наименование станций метро в городе, которые привязаны к конкретным районам.
afisha	Содержит обновляемую информацию о ближайших событиях в городе.
usertbl	Содержит информацию о зарегистрированных пользователях, хранится информация о логинах и паролях.

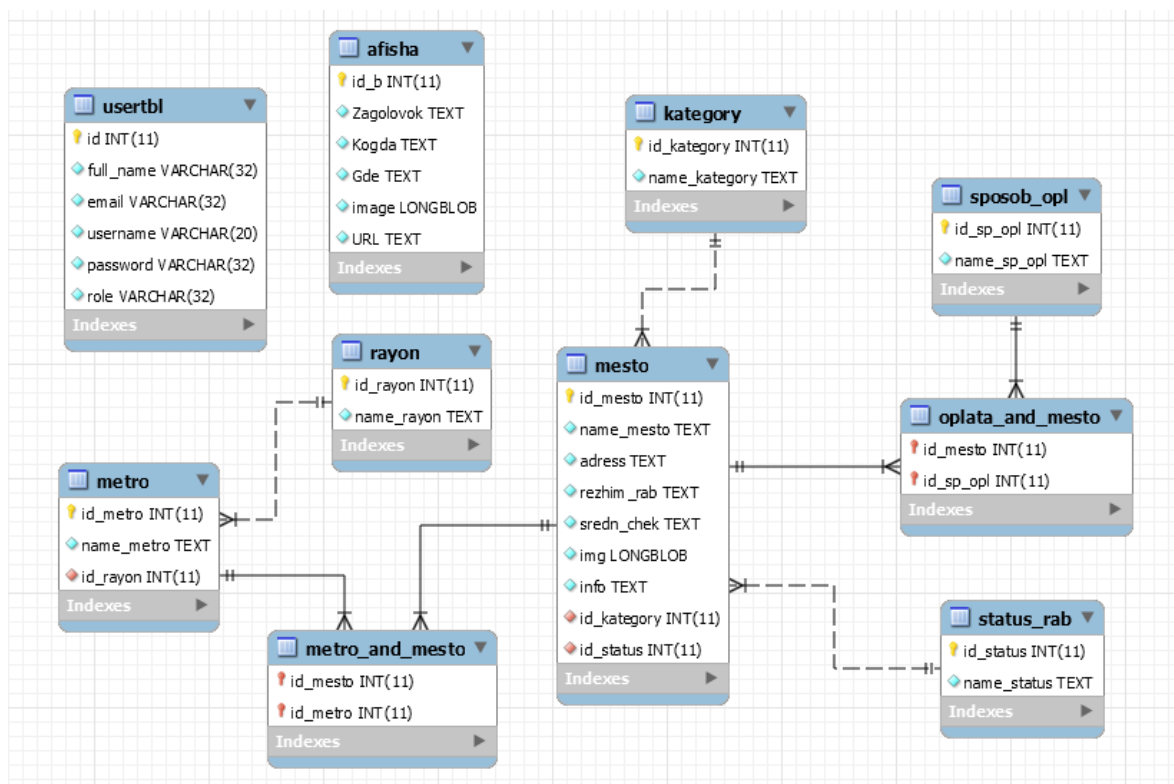


Рисунок 9 – Схема базы данных

4.2 Структура таблиц базы данных

Создание таблиц осуществлялось в СУБД MySQL через пользовательское приложение WorkBench. Приложение Microsoft Office Access осуществляет управление созданной базой данных[7].

Далее представлено описание списка полей базы данных с подробными комментариями к ним.

1. Таблица базы данных kategory (содержит информацию о категориях).

id_kategory	name_kategory
1	Кофейни
2	Кино
3	Парки
4	Театры

Рисунок 10 – Содержание таблицы базы данных kategory

Таблица 2 – Описание полей таблицы kategory:

Наименование поля	Тип данных и параметры поля	Описание
id_kategory	int NOT NULL auto_increment PRIMARY KEY	Идентификатор категории

name_category	text NOT NULL	Наименование категории
---------------	---------------	---------------------------

2. Таблица базы данных mesto (Содержит информацию о культурно-развлекательных учреждениях города).

id_mesto	name_mesto	adress	rezhim_rab	sredn_chek	img	info
1	Tokio City	Индустриальный пр., 26/24	пн-вс	500	BLOB	пвсто
2	Baodins Coffee	ш. Революции, 8А	пн-вс	300	BLOB	Место, которое може
3	Coffee Like	пр. Большевиков, 9 корпус 1	пн-вс	350	BLOB	пвсто
4	Cub n Cup	Пискаревский пр., 16	пн-вс	400	BLOB	пвсто
5	Starbucks	Европейский просп., 21	пн-вс	550	BLOB	пвсто
6	Формула Кино	ул. Народная, 2	пн-вс	800	BLOB	пвсто
7	Morv Cinema	Невский пр., 90	пн-вс	1100	BLOB	пвсто
8	Люксор	ул. Чайковского, 18	пн-вс	400	BLOB	пвсто
9	Художественный	Невский пр., 60	пн-вс	350	BLOB	пвсто
10	Парк "Малиновка"	Индустриальный пр., 26/24	пн-вс	0	BLOB	пвсто
11	Парк 300-летия	ш. Революции, 8А	пн-вс	1500	BLOB	пвсто
12	Парк Победы	пр. Большевиков, 9 корпус 1	пн-вс	0	BLOB	пвсто
13	Охта парк	Пискаревский пр., 16	пн-вс	0	BLOB	пвсто
14	Диво-остров	Европейский просп., 21	пн-вс	350	BLOB	пвсто

Рисунок 11 – Содержание таблицы базы данных mesto

Таблица 3 – Описание полей таблицы mesto:

Наименование поля	Тип данных и параметры поля	Описание
id_mesto	int NOT NULL auto_increment PRIMARY KEY	Идентификатор учреждения
name_mesto	text NOT NULL	Наименование учреждения
adress	text NOT NULL	Адрес учреждения
rezhim_rab	text NOT NULL	Режим работы
sredn_chek	text NOT NULL	Средний чек
img	LONGBLOB NOT NULL	Изображение
info	text NOT NULL	Описание учреждения
id_kategory	int NOT NULL	Идентификатор категории
id_status	int NOT NULL	Идентификатор статуса

3. Таблица базы данных rayon (Содержит наименования районов города).

id_rayon	name_rayon
1	Красногвардейский
2	Невский
3	Адмиралтейский

Рисунок 12 – Содержание таблицы базы данных rayon

Таблица 4 – Описание полей таблицы rayon:

Наименование поля	Тип данных и параметры поля	Описание
id_rayon	int NOT NULL auto_increment PRIMARY KEY	Идентификатор района
name_rayon	text NOT NULL	Наименование района

4. Таблица базы данных metro (Содержит наименование станций метро в городе, которые привязаны к конкретным районам).

id_metro	name_metro	id_rayon
1	м. Ладужская	1
2	м. Новочеркасская	1
3	м. Адмиралтейская	3
4	м. Сенная площадь	3
5	м. Садовая	3
6	м. Балтиская	3
7	м. вл. Дыбенко	2
8	м. Пр-т Большевиков	2
9	м. Елизаровская	2

Рисунок 13 – Содержание таблицы базы данных metro

Таблица 5 – Описание полей таблицы metro:

Наименование поля	Тип данных и параметры поля	Описание
id_metro	int NOT NULL auto_increment PRIMARY KEY	Идентификатор ст. метро
name_metro	text NOT NULL	Наименование ст. метро
id_rayon	int NOT NULL	Идентификатор района

5. Таблица базы данных afisha (Содержит обновляемую информацию о ближайших событиях в городе).

id_b	Zagolovok	Kogda	Gde	ii
1	Мюзикл «Бал вампиров»	11 - 30 Июня	Театр музыкальной комедии	В
2	Тимур Шаов «Летний концерт»	6 Июня, 19:30	Концертный зал в Финляндского	В
3	Концерт «Сон в летнюю ночь»	7 Июня, 19:00	Академическая Капелла Санкт-Петербурга	В
4	Музыкальный спектакль «Бармалей»	9 Июня - 14 Июля	Театр Эстрады им. Аркадия Райкина	В
5	Концерт Дэви Премал и Митена	30 Сентября, 19:00	ДК им. Ленсовета	В
6	Фестиваль «Ночь света в Гатчине»	16 - 17 Августа	Музей-заповедник «Гатчина»	В
7	Театральная карта мира	21 Мая - 28 Июня		В
8	Театральная олимпиада 2019	16 Июня - 15 Июля		В
9	Спектакль «Спаси рядового Гамлета»	16 Октября, 19:00	ДК Выборгский	В
10	Спектакль «Аквитанская львица»	17 - 18 Июня	ДК Выборгский	В
11	Концерт Карлоса Д'Онофрио	13 Июня, 20:00	Арт-пространство «Freedom»	В
12	Фестиваль акустической музыки «А...	25 - 28 Июля		В
13	Концерт Квартета Евгения Пономарева	4 Июля, 19:30	Александринский театр	В
14	Концерт «Земфира. Трибют»	6 Июля, 20:00	Чаплин-Холл	В

Рисунок 14 – Содержание таблицы базы данных afisha

Таблица 6 – Описание полей таблицы afisha:

Наименование поля	Тип данных и параметры поля	Описание
id_b	int NOT NULL auto_increment PRIMARY KEY	Идентификатор поста афиши
Zagolovok	text NOT NULL	Заголовок поста афиши
Kogda	text NOT NULL	Дата и время события
Gde	text NOT NULL	Адрес события
image	LOB NOT NULL	Изображение
URL	text NOT NULL	Ссылка на источник

6. Таблица базы данных usertbl (Содержит информацию о зарегистрированных пользователях).

id	full_name	email	username	password	role
1	Oloa	Ol@gmail.com	admin	12345	Администратор
2	Arseniv	A@mail.ru	manager	123	Менеджер
3	Sasha	S@mail.ru	user1	111	Пользователь
4	Viktoriva	V@mail.com	user2	222	Пользователь

Рисунок 15 – Содержание таблицы базы данных usertbl

Таблица 7 - Описание полей таблицы usertbl:

Наименование поля	Тип данных и параметры поля	Описание
id	int(11) NOT NULL auto_increment	Идентификатор пользователя
full_name	varchar(32) collate	Имя пользователя

	utf8_unicode_ci NOT NULL	
email	varchar(32) collate utf8_unicode_ci NOT NULL	Адрес электронной почты пользователя
username	varchar(20) collate utf8_unicode_ci NOT NULL	Логин пользователя
password	varchar(32) collate utf8_unicode_ci NOT NULL	Пароль пользователя
role	varchar(32) collate utf8_unicode_ci NOT NULL	Роль пользователя

4.3 Хранимые процедуры

Для упрощения работы с базой данных в MySQL Workbench были разработаны и реализованы хранимые процедуры.

Далее представлено описание хранимых процедур с подробными комментариями к ним.

1. Хранимая процедура proc1, при вводе названия категории выводятся все входящие в эту категорию позиции учреждений.

Листинг 1 – Фрагмент программного кода процедуры proc1

```
delimiter //
CREATE procedure proc1 (IN Input_category text) begin
SELECT mesto.name_mesto
FROM kategory JOIN mesto ON kategory.id_kategory = mesto.id_kategory
WHERE (((kategory.name_kategory)= Input_category));
end//
delimiter;
Call proc1 ('Парки');
```

name_mesto
Парк "Малиновка"
Парк 300-летия
Парк Победы
Охта парк
Диво-остров
Парк хороший
Парк плохой
Парк средний
Парк "Корова"
Парк "Канноха"
Парк "Вондерфул лайф"

Рисунок 16 – Результат хранимой процедуры proc1

2. Хранимая процедура proc2, при вводе наименования района выводятся все станции метро, принадлежащие этому району.

Листинг 2 – Фрагмент программного кода процедуры proc2

```
delimiter //
CREATE procedure proc2 (IN Input_rayon text) begin
SELECT metro.name_metro
FROM rayon JOIN metro ON rayon.id_rayon = metro.id_rayon
WHERE (((rayon.name_rayon)= Input_rayon));
end//
delimiter;
Call proc2 ('Невский');
```

name_metro
м. вл. Дыбенко
м. Пр-т Большевиков
м. Елизаровская

Рисунок 17 – Результат хранимой процедуры proc2

3. Хранимая процедура, при вводе названия станции метро, выводится информацию об учреждениях, которые находятся в зоне доступности этой станции метро.

Листинг 3 – Фрагмент программного кода процедуры proc3

```
delimiter //
CREATE procedure proc3 (IN Input_st_metro text) begin
SELECT mesto.name_mesto, adress, sposob_opl, status_rab
FROM (metro JOIN metro_and_mesto USING (id_metro))
JOIN mesto USING (id_mesto)
WHERE metro.name_metro = Input_st_metro;
end//
delimiter;
```

Call proc3 ('м. Адмиралтейская');

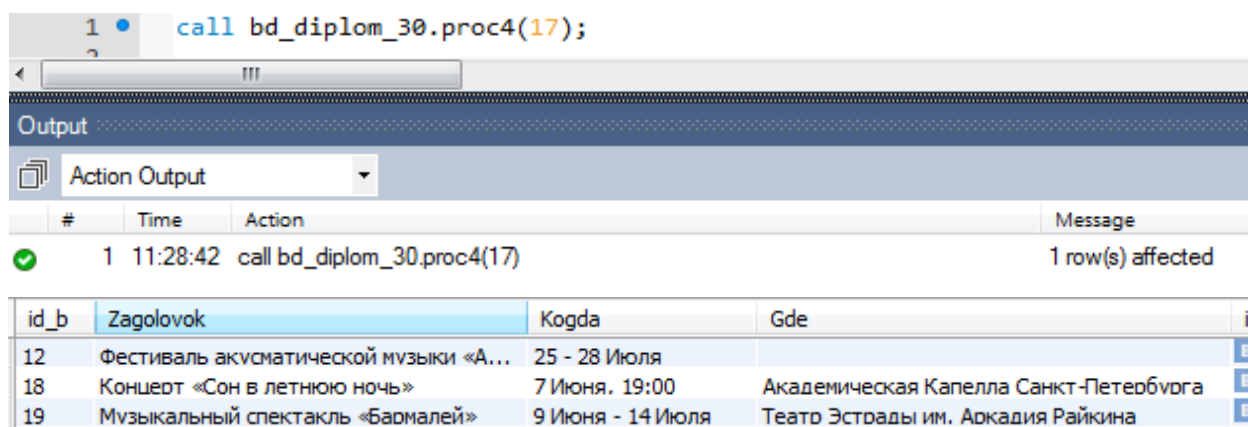
name_mesto	adress
Bacchins Coffee	ш. Революции. 8А
Coffee Like	пр. Большевиков. 9 корпус 1
Starbucks	Европейский просп.. 21
Формула Кино	вл. Народная. 2
Художественный	Невский пр.. 60
Парк "Малиновка"	Индустриальный пр.. 26/24

Рисунок 18 – Результат хранимой процедуры proc3

4. Хранимая процедура proc4 – удаляет записи из таблицы Afasha по номеру ID, на сайте они так же удаляются.

Листинг 4 – Фрагмент программного кода процедуры proc4:

```
delimiter //  
create procedure proc4 (IN Del_post int) begin  
DELETE FROM afisha  
WHERE afisha.id_b = Del_post;  
end//  
delimiter;  
Call proc4 ('17');
```



#	Time	Action	Message
1	11:28:42	call bd_diplom_30.proc4(17)	1 row(s) affected

id_b	Zagolovok	Kogda	Gde	ir
12	Фестиваль акустической музыки «А...	25 - 28 Июля		
18	Концерт «Сон в летнюю ночь»	7 Июня. 19:00	Академическая Капелла Санкт-Петербурга	
19	Музыкальный спектакль «Бормалей»	9 Июня - 14 Июля	Театр Эстрады им. Аркадия Райкина	

Рисунок 19 – Результат хранимой процедуры proc4

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1 Определение и классификация информационных систем, URL: http://tsput.ru/res/informat/sist_seti_fmo/lekcii/lekciiy-1.html (дата обращения: 02.05.2019)

2 Сенин, В.С. Организация международного туризма: Учебное пособие / М.: Финансы и статистика, 2014. 399 с.

3 Лучшие системы управления отелем из России, URL: <https://www.tourbc.ru/tehnologii/obzory/486-luchshie-sistemy-upravleniya-otelim-iz-rossii.html#ecvi> (дата обращения: 03.05.2019)

4 Камаев, В.А. Технологии программирования / Камаев В.А., Костерин В.В. М.: Высш. шк., 2006. 454 с.

5 Дейт, К. Введение в системы баз данных / К.Дж. Дейт; пер.с англ.К.А. Птицына. М.: Издательский дом «Вильямс», 2005. 1328 с.

6 Обзор систем управления баз данными для систем контроля и управления доступом, URL: <https://drawgrow.ru/obzor-sistem-upravleniya-bazami-dannyh-subd-dlya-sistem-kontrolya-i.html> (дата обращения: 10.05.2019)