

СОДЕРЖАНИЕ

| | |
|--|----|
| СОДЕРЖАНИЕ..... | 3 |
| ВВЕДЕНИЕ..... | 4 |
| 1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ. РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ..... | 5 |
| 1.1 Описание предметной области..... | 5 |
| 1.2 Постановка задачи проектирования..... | 8 |
| 1.3 Сравнительный анализ аналогов и прототипов..... | 9 |
| 1.4 Анализ требований к проекту..... | 10 |
| 2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ..... | 11 |
| 2.1 Обоснование методов решения задач..... | 11 |
| 2.2 Выбор и обоснование средств разработки..... | 14 |
| 2.3 Разработка функциональной модели..... | 16 |
| 2.4 Разработка функциональной модели..... | 18 |
| 2.5 Проектирование интерфейса..... | 20 |
| ЗАКЛЮЧЕНИЕ..... | 22 |

| | | | | | | | | |
|------------------|-------------|-----------------------|----------------|-------------|--|---|-------------|---------------|
| | | | | | <i>Отчет по преддипломной практике</i> | | | |
| | | | | | | | | |
| <i>Изм.</i> | <i>Лист</i> | <i>№ докум.</i> | <i>Подпись</i> | <i>Дата</i> | <i>Разработка web-сайта Библиотечной Системы</i> | <i>для</i> | <i>Лист</i> | <i>Листов</i> |
| <i>Разраб.</i> | | <i>Филимонов К.В.</i> | | | | | <i>3</i> | <i>22</i> |
| <i>Провер.</i> | | <i>Пастухов Д.Ф.</i> | | | | <i>Учреждение образования «Полоцкий государственный университет» гр.19-ИТзс</i> | | |
| <i>Реценз.</i> | | | | | | | | |
| <i>Н. Контр.</i> | | | | | | | | |
| <i>Утверд.</i> | | | | | | | | |

ВВЕДЕНИЕ

Современную библиотеку трудно представить без компьютера, специализированного программного обеспечения и web-сайта, позволяющего упростить работу самой библиотеки.

Интернет развивается стремительно и сегодня большинство компаний уже имеют свои сайты. Впрочем, не только компании, но и частные лица, оказывающие те или иные услуги. Иметь интернет-ресурс – это, скорее, необходимость. Но если вы еще задаетесь вопросом, зачем нужен сайт – я подготовил для вас новую статью. В ней подробно расскажу, какие задачи он поможет решать. Статистика утверждает, что в среднем каждый человек ежедневно проводит в интернете не менее 3-х часов. Это если не говорить о тех, кто связан с ним по работе. Учитывая такие данные, компании стали реагировать соответствующим образом и вести бизнес через интернет.

Люди оценили удобство сети. Ведь она позволяет вести бизнес, узнавать информацию, бронировать билеты и гостиницы, просто развлекаться. Возможности, предоставляемые такими ресурсами, невероятные.

В нынешнее время библиотечная система, представляющая из себя довольно сложный "организм" из нескольких областей, особенно нуждается в качественном web-ресурсе, который предоставлял бы достойный уровень оказания услуг и осуществления маркетинговой деятельности, в доступном формате предоставлял бы нужную информацию, а также помогал бы вести блог проектов, ресурсов, новостей и событий в рамках самой организации.

В конце концов, наличие собственного сайта существенно повышает имидж организации в глазах потенциальных клиентов.

Актуальность данной темы особенно подчёркивается и тем, что не всегда в обычных библиотеках бывает достаточное количество экземпляров книг. Использование web-ресурсов библиотеки позволяет получить возможность доступа к фондам филиалов и всей библиотечной системы в целом.

Для решения всех упомянутых ранее задач как нельзя лучше подойдёт web-сайт, который бы предоставлял доступ к фондам, освещал информационно-событийную деятельность и т.д. Профессионально созданный сайт обеспечивает легкость его нахождения по запросам в поисковых системах, поскольку целевую аудиторию составляют пользователи, которые ищут конкретную информацию.

Главной задачей ресурса является обеспечение эффективности, информативности и полезности как новым читателям библиотеки, так и частым посетителям.

1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ. РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ

1.1 Описание предметной области

Библиотека – это учреждение культуры, организующее сбор, хранение и общественное пользование произведениями печати и другими документами. Библиотеки систематически занимаются сбором, хранением, пропагандой и выдачей читателям произведений печати, а также информационно-библиографической работой, являются общедоступным источником знаний и основной базой для самообразования.

Основными направлениями работы любой библиотеки являются: комплектование и организация книжного фонда; обслуживание читателей.

Комплектование фондов библиотеки состоит из систематического выявления (путем просмотра библиографических источников и литературы) нужных для данной библиотеки изданий и приобретения их. От своевременности и полноты комплектования библиотеки в значительной мере зависит уровень обслуживания читателей.

Организация книжного фонда включает вопросы учета, расстановки, хранения литературы и доставки ее читателю.

Учет библиотечного фонда – это прием поступивших в библиотеку изданий, запись прибывших и исключение по тем или иным причинам непригодных для читателей изданий в документах библиотеки, которые дают точные сведения о наличном составе фонда и помогают обеспечить его сохранность.

Правильная организация фонда облегчает читателю пользование литературой, библиотекарю - быстрое выполнение читательских требований, а также обеспечивает сохранность фондов как общественной собственности.

Обслуживание читателей библиотеки осуществляется различными путём выдачи литературы и помощи отдельным читателям в подборе необходимой им литературы.

Читатели, приходящие в библиотеку, обязаны иметь при себе читательский билет. При выбытии из вуза (отчисление, окончание обучения, увольнение) читатели обязаны вернуть числящиеся за ними издания и сдать читательские билеты.

За нарушение правил пользования библиотекой читатели лишаются права пользования всеми пунктами обслуживания библиотеки на установленные администрацией сроки (1 месяц).

Срок пользования литературой для различных категорий читателей и количество выдаваемых изданий на каждом абонементе определяется администрацией, исходя из вида литературы и категории читателя. Число книг, выдаваемых в читальных залах, не ограничивается.

Библиотечная деятельность связана с учетом большого количества операций, множество книг и читателей серьезно замедляют работу библиотекарей. Сложность поиска нужной книги в каталоге занимает длительное время, и целиком опирается на компетентность работников библиотеки.

Для ведения библиотечных каталогов, организации поиска требуемых изданий и библиотечной статистики в базе должны храниться сведения, большая часть которых размещаются в аннотированных каталожных карточках

В библиотеку приходит много книг из различных издательств. Каждая книга, хранящаяся в библиотеке, имеет следующие параметры:

- название книги;
- уникальный шифр (ISBN);
- библиотечно-библиографическая классификация (ББК);
- автор;
- издательство;
- место издания (город);
- год издания;
- количество страниц.
- Каждая книга может присутствовать в нескольких экземплярах.
- Все книги различаются по своему уникальному шифру - ISBN.

Библиотечно-библиографическая классификация (ББК) распределяет издания по отраслям знания в соответствии с их содержанием. В ней используется цифробуквенные индексы ступенчатой структуры (например, ББК 32.973 Электронные вычислительные машины и устройства). Шифр ББК используется при выделении хранимым изданиям определенных комнат, стеллажей и полок, а также для составления каталогов и статистических отчетов. В библиотеке имеется база данных (БД). Она представляет собой совокупность данных обо всех происходящих процессах в библиотеке (книгах, читателях, проведенных мероприятиях и др.), построенную в хронологической последовательности, в виде таблиц и списков данных. Основное ее назначение - хранение данных, для их последующего целесообразного использования в будущем. БД находится на специальном компьютере в электронном виде, постоянно редактируется и изменяется работниками библиотеки. С базой данных библиотеки могут работать как руководители (администрация) библиотеки, так и работники библиотеки, заинтересованные в получении необходимой информации. При работе с системой библиотекарь должен иметь возможность решать следующие задачи:

- принимать новые книги и регистрировать их в библиотеке;
- относить книги к одной или к нескольким областям знаний;
- проводить каталогизацию книг, то есть назначение новых инвентарных номеров вновь принятым книгам;
- вести учет выданных книг читателям, при этом предполагается два режима работы: выдача книг читателю и прием от него возвращаемых им книг

обратно в библиотеку. При выдаче книг фиксируется, когда и какая книга была выдана данному читателю и на какой срок выдается данная книга. При приеме книги, возвращаемой читателем, проверяется соответствие возвращаемого инвентарного номера книги выданному инвентарному номеру, название книги, и она ставится на свое старое место в библиотеке. Администрация библиотеки должна иметь возможность получать сведения о должниках - читателях библиотеки, которые не вернули вовремя взятые книги.

В библиотеке ведется картотека читателей. О каждом читателе заносятся следующие сведения:

- ФИО;
- номер паспорта;
- адрес;
- контактный телефон.

Каждому читателю присваивается номер читательского билета. Кроме того, учитываются сведения о наличии у читателя штрафов.

В случае выдачи экземпляра книги в библиотеке остается вкладыш, в котором указана дата выдачи, дата предполагаемого возврата и номер читательского билета.

При возврате книги во вкладыше отмечается срок возврата. При просрочке возврата книги, читатель получает предупреждение. Когда предупреждений у читателя накапливается больше определенного предела, он лишается правом пользования библиотекой на определенный срок.

При потере книги читатель лишается правом пользования библиотекой на определенный срок, независимо от количества предупреждений.

В библиотеке есть ряд ограничений: нельзя выдавать книги больше установленного периода, нельзя выдавать больше определенного количества книг.

1.2 Постановка задачи проектирования

Целью данной работы является разработка фронтенд web-сайта Библиотечной Системы, а также удобной и доступной панели администрирования.

Сайт предназначен для повышения эффективности работы обслуживающего персонала библиотеки и быстрого обслуживания читателей за счёт услуги автоматизированного поиска книги, сокращения времени поиска и оформления выдачи необходимого издания читателю, а также осуществления маркетинговой деятельности и возможности вести блог проектов, ресурсов, новостей и событий в рамках самой организации.

Web-сайт Библиотечной Системы работает в трёх режимах, в зависимости от типов пользователей: в режиме посетителя, сотрудника библиотеки или администратора.

Автоматизация технологических процессов в библиотеке, предоставляемая сайтом, призвана обеспечить следующие возможности:

Администратор поддерживает сайт, обновляет вовремя информацию. Сотрудник библиотеки вносит и изменяет информацию, загружает документы, обновляет список мероприятий, продвигает сайт и рекламирует его, улучшает и модернизирует. Посетители сайта могут просматривать информацию, читать книги в электронном формате, информироваться о прошедших/близжащих мероприятиях и точном местонахождении библиотеки.

Для достижения указанной цели диплома ставятся и решаются следующие задачи:

- 1) анализ предметной области;
- 2) проектирование и разработка web-сайта Библиотечной Системы;
- 3) тестирование системы.

Разработка должна быть основана на результатах проектирования системы и включать обоснование выбора средств программной реализации и описание структур данных, классов, функций программного комплекса.

Тестирование и анализ результатов необходимы для выявления ошибок в структуре алгоритма или его программной реализации, структуре входных и внутренних данных, а также для ликвидации конфликтных ситуаций с другим ПО.

1.3 Сравнительный анализ аналогов и прототипов

При создании приложения был произведен поиск аналогов программного продукта в всемирной сети Интернет.

Самым прямым аналогом является нынешний сайт ГУК ЦБС г. Новополоцк. По функционалу он похож на разрабатываемый проект, но не представляет возможности удобного ознакомления со всей имеющейся на нём информации, а также не имеет доступного и простого способа управления контентом, что вынуждает специалистов из отдела автоматизации во всём сопровождать действия сотрудников отдела маркетинга, краеведов и т.д. Его дизайн также вызывает вопросы в плане вёрстки и цветовых решений, напрямую влияющих не только на восприятие пользователя, но и имидж.

На фоне нынешнего решения разработка в рамках диплома является по своей сути преемником идеи и прямым улучшением предоставляемых им возможностей. Разрабатываемый сайт также призван популяризировать библиотечную деятельность посредством дизайнерских решений.

| | | | | | | |
|------|------|----------|---------|------|---------------------------------|------|
| | | | | | Отчет по преддипломной практике | Лист |
| | | | | | | 9 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

1.4 Анализ требований к проекту

Первым этапом создания любого сайта, в том числе и сайта библиотеки, является проведение детального анализа информационных потребностей его потенциальных посетителей.

Следует сразу принять за правило, что сайт создается именно для читателей, а не для самих библиотекарей, поэтому пользовательские интересы должны быть в первую очередь учтены при выработке общей структуры и стратегии наполнения виртуального ресурса.

Многолетняя практика создания сайтов позволила сформулировать ряд общих требований, которым должны отвечать виртуальные представительства библиотек.

Все они базируются на общих критериях для web-серверов. К ключевым показателям качества web-серверов в настоящее время относятся:

- Глубина содержания;
- Простота навигации;
- Стабильность информационных ресурсов;
- Оперативность обновления информации;
- Доступность для пользователей;
- Единство дизайна всех разделов.

Глубина содержания определяется объемом имеющейся информации, степенью ее детализации и, конечно же, ценностью. Естественно, все эти показатели должны стремиться к максимуму. Информативность библиотечных сайтов может быть достигнута путем размещения материалов, представляющих наибольший интерес для виртуальных посетителей, которыми потенциально становятся все пользователи Интернет.

В качестве базовых можно предложить следующие три критерия оценки сайтов: информативность, открытость, интерактивность.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Обоснование методов решения задач

Фронтенд – презентационная часть информационной или программной системы, её пользовательский интерфейс и связанные с ним компоненты; применяется в соотношении с базисной частью системы, её внутренней реализацией, называемой в этом случае бэкендом.

Разделение программных систем на фронтенд и бэкенд – одно из стандартных решений для архитектуры программного обеспечения, связанное в программной инженерии с принципом разделения ответственности между внешним представлением и внутренней реализацией. Как правило, бэкенд реализует API, используемые фронтендом, и таким образом фронтенд-разработчику не нужно знать особенности реализации внутренней части, а бэкенд-разработчику – интерфейсные решения. Кроме того, такое разделение позволяет использовать различные инструменты для реализации внутренней и внешней части системы, более эффективные для соответствующих задач. Например, в веб-разработке к технологиям фронтенда относятся HTML, CSS, JavaScript; а бэкенд может быть реализован на PHP, Java, Python, JavaScript (Node.js).

Иногда под бэкендом в контексте информационных систем понимается административная часть системы (включая интерфейс), не отображаемая конечному пользователю, в частности, такая терминология используется в системах управления содержанием (в этом случае разделение ближе к используемому в сфере организационного управления, где бэк-офис отвечает за административные функции, а фронт-офис – за непосредственную работу с заказчиками).

Специфическое разделение применяется в инфраструктуре компиляции LLVM: под «фронтендами» в нём понимаются трансляторы исходных кодов программ различных языков программирования в универсальное промежуточное представление, тогда как единый бэкенд, работающий уже только с промежуточным представлением, непосредственно занимается оптимизацией, интерпретацией, компиляцией, компоновкой.

Таким образом, фронтенд-разработка – это создание пользовательского интерфейса на клиентской стороне веб-сайта или приложения. Это всё, что видит пользователь, когда открывает веб-страницу, и с чем он взаимодействует: кнопки, баннеры и анимация. Фронтенд связан с бизнес-логикой продукта (клиентская часть постоянно «общается» с серверной), но её разработкой занимаются бэкенд-программисты.

Объектно-ориентированное программирование – способ создания контента, базирующийся на разнообразных элементах. Их можно визуализировать, что значительно упрощает программирование. Логика и «голый код» здесь не выступают в качестве основополагающих.

| | | | | | | |
|------|------|----------|---------|------|---------------------------------|------|
| | | | | | Отчет по преддипломной практике | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 11 |

JavaScript – это скриптовый язык программирования. Используется для разнообразных целей – от написания аналитических утилит до игр. Основное его применение – это веб-утилиты и браузерный контент. Но элементы ООП в нем все равно присутствуют.

Объектно-ориентированное программирование – способ коддинга, который позволяет создавать разнообразные объекты посредством другого объекта. В процессе проектирования задействован так называемый конструктор. Его принцип работы основан на объектах, их создании и взаимодействии.

Общий объект – это план. Может также носить название проекта или схемы. Создаваемые посредством одного элемента – экземпляры.

Стоит обратить внимание на то, что за счет ООП в JS удастся достаточно быстро писать программы, обладающие сложной структурой. Рассматриваемая концепция предусматривает несколько ключевых аспектов:

- 1) Каждый экземпляр класса обладает свойствами, которые наследуются от родителей. Также есть собственные параметры.
- 2) Структурированный код предусматривает возможность размещения нескольких уровней в проекте. Процедура носит название наследования или классификации. Во втором случае целесообразно говорить о создании подклассов.
- 3) Инкапсуляция помогает скрывать детали реализации кодификации от сторонних глаз. Это приводит к тому, что функции и переменные, а также иные важные объекты приложения становятся не доступными извне. Таковая концепция шаблонов проектирования «Фасад» и «Модуль».

Объектно-ориентированное программирование принесло в разработку ПО новые подходы к проектированию приложений. В частности, ООП позволило программистам комбинировать сущности, объединённые некоей общей целью или функционалом, в отдельных классах, рассчитанных на решение самостоятельных задач и независимых от других частей приложения. Однако само по себе применение ООП не означает, что разработчик застрахован от возможности создания непонятного, запутанного кода, который тяжело поддерживать. Роберт Мартин, для того, чтобы помочь всем желающим разрабатывать качественные ООП-приложения, разработал пять принципов объектно-ориентированного программирования и проектирования, говоря о которых, с подачи Майкла Фэзерса, используют акроним SOLID.

Класс должен быть ответственен лишь за что-то одно. Если класс отвечает за решение нескольких задач, его подсистемы, реализующие решение этих задач, оказываются связанными друг с другом. Изменения в одной такой подсистеме ведут к изменениям в другой.

Программные сущности (классы, модули, функции) должны быть открыты для расширения, но не для модификации.

Необходимо, чтобы подклассы могли бы служить заменой для своих суперклассов.

Необходимо создание узкоспециализированных интерфейсов, предназначенных для конкретного клиента. Клиенты не должны зависеть от интерфейсов, которые они не используют.

Объектом зависимости должна быть абстракция, а не что-то конкретное.

Таким образом, принципы SOLID помогают писать расширяемый и поддерживаемый код. Поскольку в процессе разработки данного проекта преследуются модульность, гибкость, адаптивность и масштабируемость, придерживаться данных принципов будет уместно

Разрабатываемый web-сайт представлен динамически формирующим страницу конструктором на JavaScript. Основу приложения составит каркас из классов и методов, динамически собирающий страницу на уровне фронтенда. Данный каркас формирует модель – конфигурацию страницы из визуальных и функциональных HTML-компонентов.

2.2 Выбор и обоснование средств разработки

Существуют различные способы создания web-ресурсов:

При помощи программирования.

Этот способ – наиболее длительный и трудный, хотя для создания сайта он наиболее предпочтительный. Необходимо хорошо знать хотя бы один из распространенных языков веб-программирования, например, HTML, PHP, CSS, а также работу с базами данных, например, MySQL.

На основе популярных программ.

Например, MS FrontPage, Macromedia Dreamweaver.

Создание сайта при помощи системы управления сайтом (CMS)

Создание сайта при помощи конструктора.

В данной курсовой работе для создания web-сайта используется способ программирования. Конкретно написание web-сайта будет производиться в IntelliJ IDEA и Visual Studio Code.

Для создания web-сайта Библиотечной Системы использовались следующие инструментальные языки разработки:

1. HTML – это язык гипертекстовой разметки. Гипертекстом можно назвать текст в любом формате, некоторые части текста которого, могут ссылаться на другие части этого текста, что позволяет быстро переходить от одной части текста к другой.

На сегодняшний день язык HTML существенно усовершенствован. Его прекрасно дополняют технологии JavaScript, DHTML, CSS, PHP и др., что позволяет использовать новые возможности, но основой всего этого остаются все те же стандартные теги, которые и задают внешний вид web-страницы. Браузер, считывая информацию со страниц, размещенных в Интернете, отображает ее на экране монитора в том виде, в котором четко указывают ему эти самые теги. Все существующие теги заключены в угловые скобки (< тег >)*, это и указывает браузеру на то, что перед ним тег, который ориентирует на определенное отображение страницы, причем сами теги на экране не отображаются.

HTML-документ состоит из трех основных частей:

- строка декларации типа документа;
- заголовок документа, заключенный в тег <head>...</head>;
- тело документа, заключенное в тег <body>...</body> или <frameset>...</ frameset>.

Заголовок и тело документа заключаются в объединяющий их тег <html >...< / html >.

Строка декларации является служебной и указывает на стандарт разработки документа. Для документа, строго соответствующего стандарту HTML 4.0, она представляется унарным тегом вида

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">.

Заголовок в теге <head> содержит информацию об общих свойствах документа. Содержимое тега <head> непосредственно не отображается в окне браузера, однако используется браузером при работе с файлом. Заголовок может включать в себя следующие теги:

- титул окна документа (тег <title>);
- метатеги документа (теги <meta>);
- базовый адрес для ссылок (тег <base>);
- связь с другими документами (теги <link>);
- встроенные таблицы стилей для оформления (тег <style>);
- встроенные программы-сценарии клиента (теги <script>).

CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

2.3 Разработка функциональной модели

Для начала, следует выделить ключевые функциональные возможности, которыми должно обладать разрабатываемое средство. На этапе проектирования выделение функциональных возможностей происходит по методологии UML, они отображаются на диаграмме вариантов использования.

Как правило, исходной моделью, с которой начинается процесс моделирования на языке UML является диаграмма вариантов использования. Она позволяет описать функциональное назначение системы в самом общем виде с точки зрения всех её пользователей и заинтересованных лиц.

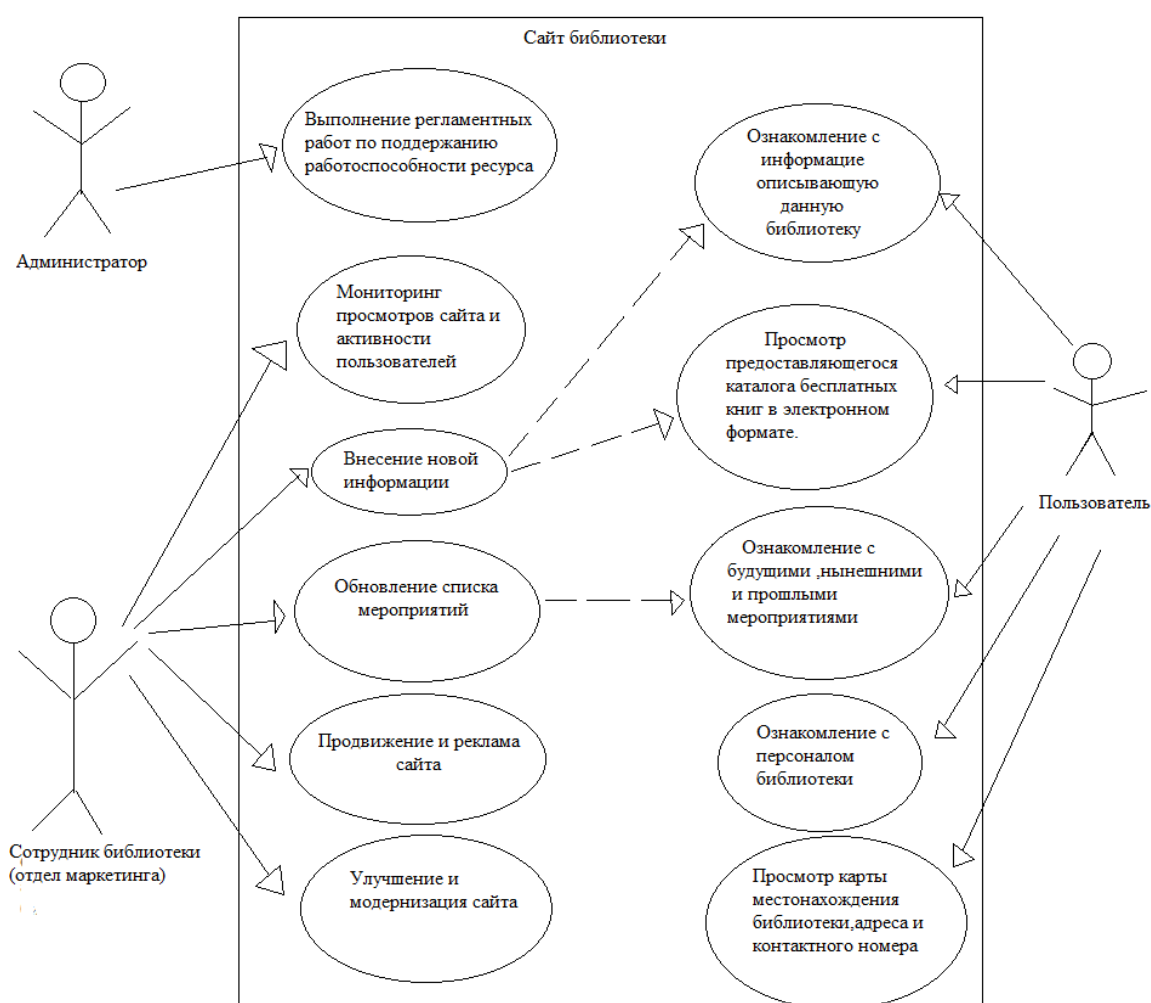


Рисунок 1 – Диаграмма вариантов использования

Варианты использования определяют последовательность действий, которые должны быть выполнены проектируемой системой при взаимодействии ее с соответствующим актером.

Пользователями системы являются: администратор, сотрудник библиотеки (из отдела маркетинга), посетитель сайта.

Администратор поддерживает сайт, обновляет вовремя информацию. Сотрудник библиотеки вносит и изменяет информацию, загружает документы, обновляет список мероприятий, продвигает сайт и рекламирует его, улучшает и модернизирует. Посетители сайта могут просматривать информацию, читать книги в электронном формате, информироваться о мероприятиях и точном местонахождении библиотеки.

Доступ к административной части имеют системный администратор и администратор хостинга.

Доступ к административной части для других пользователей, кроме системного администратора, не осуществляется.

Все данные сайта хранятся на хостинге. Исключения составляют книги в электронном формате, они находятся на отдельном электронном хранилище и открываются при помощи гиперссылки на модуль.

2.4 Разработка функциональной модели

HTML (от англ. HyperText Markup Language – «язык гипертекстовой разметки») – стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

Язык HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986-1991 годах в стенах Европейского Центра ядерных исследований в Женеве (Швейцария). HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки.

С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже.

CSS (англ. Cascading Style Sheets – каскадные таблицы стилей) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML.

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы от описания внешнего вида этой веб-страницы. Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

JavaScript – прототипно-ориентированный сценарный язык программирования.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам

Bootstrap – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML и CSS шаблоны оформления для типографики, веб форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript расширения.

Bootstrap использует современные наработки в области CSS и HTML, поэтому необходимо быть внимательным при поддержке старых браузеров.

Bootstrap – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML и CSS шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейсов, включая JavaScript расширения.

Bootstrap использует самые современные наработки в области CSS и HTML, поэтому необходимо быть внимательным при поддержке старых браузеров, для них придется писать дополнительный код.

Основные преимущества Bootstrap:

Экономия времени – вы можете сэкономить много времени и усилий, используя шаблоны дизайнера и классы Bootstrap и сконцентрироваться на других разработках;

Высокая скорость – с использованием Bootstrap вы можете легко создавать динамичные макеты. Макеты Bootstrap легко масштабируются на разные устройства и разрешения экрана без каких-либо изменений в разметке;

Гармоничный дизайн – все компоненты платформы Bootstrap используют единый стиль и шаблоны с помощью центральной библиотеки. В итоге дизайн и макеты веб-страниц согласуются друг с другом;

Простота в использовании – платформа очень проста в использовании. Поэтому любой пользователь с базовыми знаниями HTML и CSS может начать разработку с Bootstrap;

Совместимость с браузерами – Bootstrap совместим со всеми современными браузерами, такими как Mozilla Firefox, Google Chrome, Safari, Internet Explorer и Opera;

Открытое программное обеспечение – самая лучшая особенность Bootstrap, которая предполагает удобство использования, посредством открытости исходных кодов и бесплатной загрузки.

Основные инструменты Bootstrap:

Сетки – заранее заданные размеры колонок, которые можно сразу же использовать, например ширина колонки 140px относится к классу .span2, который можно использовать в CSS описании документа.

Типографика – Описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т. п.

Медиа – Представляет некоторое управление изображениями и Видео.

Таблицы – Средства оформления таблиц, вплоть до добавления функциональности сортировки.

2.5 Проектирование интерфейса

Интерфейс – это средство взаимодействия пользователя и веб-продукта (сайт, сервис, приложение) через браузер, монитор, смартфон, простыми словами это картинка, которую мы видим на экране монитора.

Часто интерфейс упоминают в значении «графический пользовательский интерфейс», то есть сюда относят все элементы, представленные визуально (кнопки, пиктограммы и т. д.). Но нас в первую очередь интересуют те элементы, с которыми будут взаимодействовать пользователи вашего сайта или приложения: кнопки, формы захвата, фильтры. Ведь если посетитель будет понимать, куда ему нажимать для достижения цели, вероятность добиться от него этого действия многократно возрастает.

UI-дизайн интерфейса представляет собой разновидность скинов, где в роли одного участника процесса выступает человек, а вторым является машина. Такой интерфейс сайта является совокупностью средств и методов, за счет которых пользователь работает с различными, обычно достаточно сложными, устройствами.

В пользовательский скин входят оборудование, то есть физический компонент, и программное обеспечение, или логический компонент. Сегодня разработаны интерфейсы для различных систем, поэтому UI позволяют осуществлять два типа взаимодействия.

UX-дизайн интерфейса, то есть опыт взаимодействия, включает в себе поведение, отношение, эмоции человека, которые он испытывает во время использования продукта, системы, услуги. Также к UX относится восприятие системы как полезной, простой в использовании, эффективной. Данный показатель глубоко субъективен, поскольку предполагает индивидуальное восприятие и оценку.

UX является непостоянным понятием, так как его параметры меняются вместе с системами, обстоятельствами, технологиями и проч.

Прежде чем приступить к разработке UX, важно выбрать основную аудиторию и сформировать образ целевого пользователя, набор его качеств. На базе имеющихся сведений можно вводить специальные требования к проекту. А именно: необходимо подготовить информационную архитектуру интерфейса сайта и иерархию содержимого, что позволяет упростить работу над прототипом.

Приложение должно легко восприниматься визуально, для этого ее следует выполнить в одной цветовой гамме. Интерфейс программы необходимо построить так, чтобы пользователю не составляло большого труда разобраться в функционале приложения.

Главное окно разрабатываемой системы будет содержать панель для навигации по программе. Страница работ будет содержать набор компонентов, обеспечивающих удобный доступ к контенту и данным.

В рамках темы проектирования дипломного проекта был сформирован макет конфигурации компонентов главной страницы сайта Библиотечной Системы. Он представлен на рисунке 2.

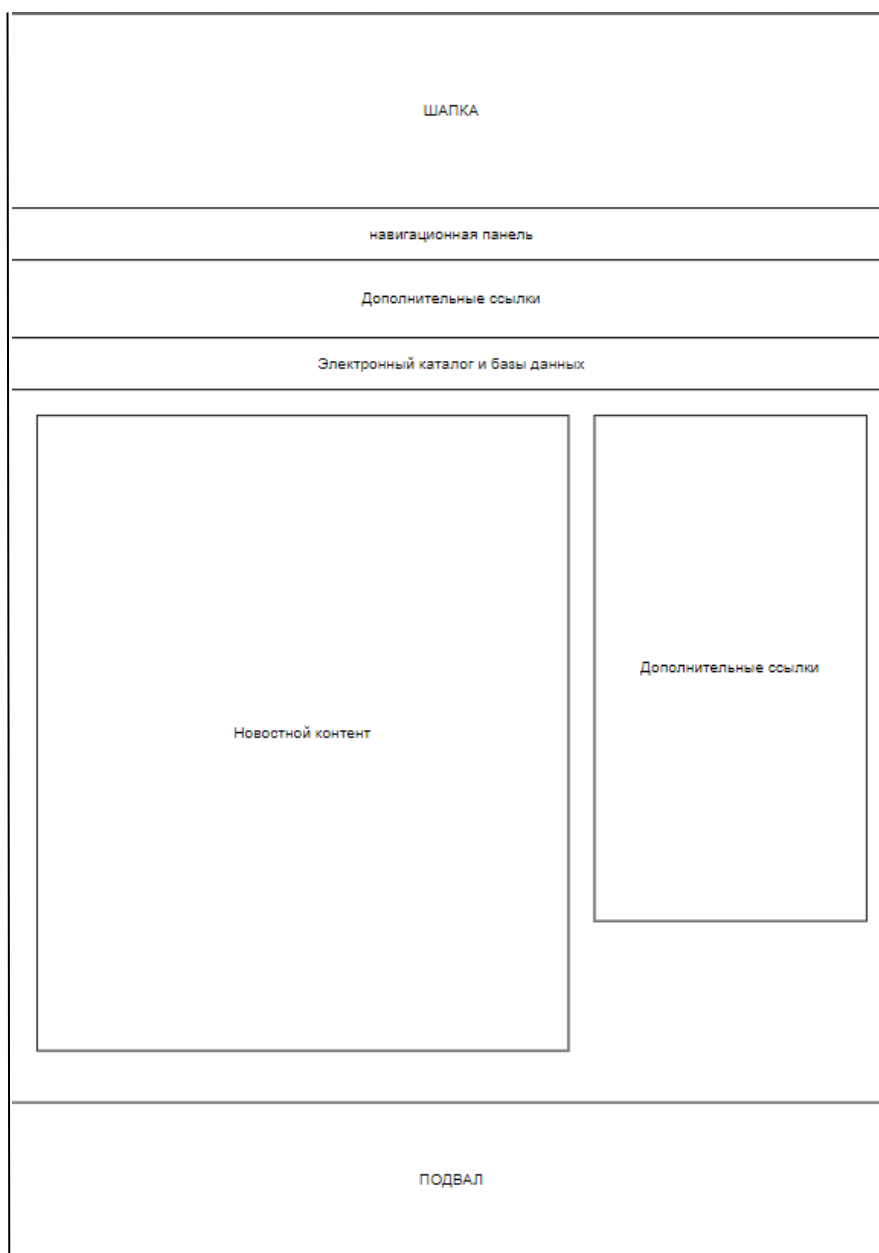


Рисунок 1 – Диаграмма вариантов использования

ЗАКЛЮЧЕНИЕ

В ходе выполнения данного дипломного проекта был спроектирован и разработан web-сайт Библиотечной Системы.

Также были изучены:

- история предприятия;
- структурная организация предприятия;
- мероприятия по охране труда и технике безопасности;
- методы проектирования и разработки программного обеспечения;
- методы оценки качества программного обеспечения и его сертификации;
- принципы формирования экономического эффекта от использования современных информационных технологий, формирования себестоимости программного продукта, рентабельности и прибыли.

Исходя из вышеперечисленных выводов о прохождении преддипломной практики, можно заключить, что она прошла качественно и эффективно.

Результаты выполнения дипломного проектирования представлены в виде программы, эксплуатационной программной документации и рабочей документации информационного обеспечения.

| | | | | | | |
|------|------|----------|---------|------|---------------------------------|------|
| | | | | | Отчет по преддипломной практике | Лист |
| | | | | | | 22 |
| Изм. | Лист | № докум. | Подпись | Дата | | |