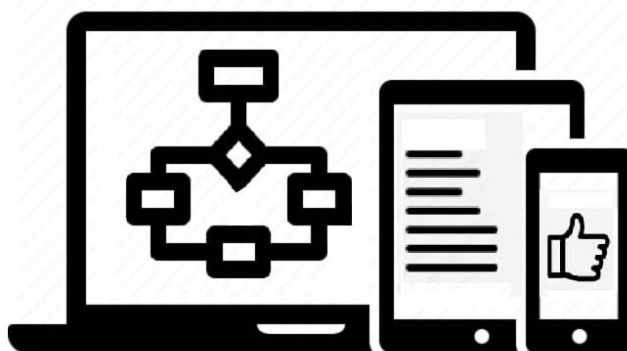


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

СРЕДСТВА ВЗАИМОДЕЙСТВИЯ ЧЕЛОВЕКА С ВЫЧИСЛИТЕЛЬНЫМИ СИСТЕМАМИ

*Методические рекомендации к курсовому проектированию
для студентов направлений подготовки
09.03.01 «Информатика и вычислительная техника»
и 09.03.04 «Программная инженерия»
очной формы обучения*



Могилев 2022

УДК 004
ББК 32.81
С75

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «4» марта 2022 г., протокол № 9

Составитель канд. техн. наук Ю. В. Вайнилович

Рецензент канд. техн. наук В. М. Ковальчук

Методические рекомендации содержат требования к курсовому проектированию по дисциплине «Средства взаимодействия человека с вычислительными системами» для студентов направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» очной формы обучения.

Учебно-методическое издание

СРЕДСТВА ВЗАИМОДЕЙСТВИЯ ЧЕЛОВЕКА С ВЫЧИСЛИТЕЛЬНЫМИ СИСТЕМАМИ

Ответственный за выпуск	В. В. Кутузов
Корректор	А. А. Подошевка
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2022

Содержание

Введение.....	4
1 Стандартные этапы разработки веб-приложения	5
2 Порядок выполнения и защиты курсовой работы	8
3 Используемые инструменты и технологии	11
4 Минимальные требования к курсовой работе.....	11
5 Требования к структуре курсовой работы.....	12
6 Методические рекомендации к содержанию введения, основной части и заключения курсовой работы	13
7 Требования к оформлению пояснительной записки	20
Список литературы	21
Приложение А. Примерная тематика курсовой работы	23
Приложение Б.Соглашение о коммитах	24
Приложение В. Пример аналитических отчетов	25
Приложение Г. Примеры текстовых прототипов веб-приложения	29
Приложение Д. Пример описания структуры страницы веб-приложения	33
Приложение Е. Пример описания тестового примера	34
Приложение Ж. Пример оформления титульного листа	36

Введение

Курсовая работа является одной из форм учебной деятельности, которая выполняется студентом самостоятельно под руководством преподавателя.

Курсовая работа представляет собой учебно-исследовательскую деятельность, требующую от студентов освоения элементов научного исследования. Выполнение курсовой работы направлено на формирование у студентов способности самостоятельно мыслить, анализировать и сопоставлять факты, обобщать и логически излагать материал.

В результате выполнения курсовой работы студент продемонстрирует следующие умения:

- выполнять предпроектное исследование предметной области, в рамках которой ведется разработка веб-приложения;
- осуществлять прототипирование и проектирование архитектуры веб-приложения;
- осуществлять разработку дизайна веб-приложения;
- создавать дизайн веб-приложения;
- выполнять отладку и тестирование проекта;
- оформлять документацию на разрабатываемый проект;
- использовать различные инструментальные средства проектирования, разработки, тестирования, отладки, документирования и дизайна.

Методические рекомендации по выполнению курсовой работы по дисциплине «Средства взаимодействия человека с вычислительными системами» содержат общие положения, порядок выполнения курсовой работы, структуру курсовой работы, требования к содержанию и оформлению курсовой работы, а также приложения, содержащие примеры оформления структурных элементов курсовой работы.

Методические рекомендации предназначены для оказания помощи студентам в организации работы по курсовому проектированию.

1 Стандартные этапы разработки веб-приложения

Разработка веб-приложения (веб-разработка) – это комплекс мер и действий по планированию и созданию сайта в сети Интернет в зависимости от поставленных целей и задач.

Процесс создания веб-приложения, в общем случае, включает восемь основных этапов, представленных на рисунке 1. Каждый из этапов в достаточной степени независим, что позволяет выбирать схему работы и исполнителя для каждого из них в случае работы нескольких человек над одним проектом. Однако все эти этапы может реализовать и один человек.



Рисунок 1 – Стандартные этапы разработки веб-приложения

Рассмотрим каждый этап.

Главная цель этапа исследования – определить целевую аудиторию, исследовать потребности и поведение пользователей, сформировать представление о бизнес-нише, изучить продукты-аналоги и конкурентов.

Для выполнения этого этапа можно использовать данные систем аналитики (например, сервисы jagajam.com, Google Trends и др.), данные открытых источников, перечень вопросов для интервью с заказчиком «117 вопросов упаковки бизнеса». Часть вопросов из опросника являются общими и могут быть

актуальными не для всех предметных областей, поэтому следует адаптировать вопросы для предметной области, выбранной в качестве темы курсовой работы.

На данном этапе задействованы маркетологи и менеджеры проектов.

После того, как получены аналитические данные о предметной области можно приступать к следующему этапу – составлению прототипа. Редактор и копирайтеры собирают информацию и пишут текст для разделов сайта. Фотограф и иллюстратор создают визуальный контент. При необходимости подключается видеопродакшн.

Сначала составляется текстовый прототип. Например, при разработке сайта для продажи квартир в жилом комплексе, копирайтер расписывает, что должно быть на главной странице сайта, что должно быть на внутренних страницах, какие тексты будут на этих страницах, чтобы убедить пользователя оставить заявку. Какие разделы нужны на сайте, какой контент понадобится для каждой страницы. Какую информацию нужно дать пользователям должно быть понятно из анализа предметной области.

В результате разработки тестового прототипа должно быть ясно, под какие материалы следует заложить блоки на каждой странице веб-приложения. Если пропустить данный этап или выполнить его некачественно, могут возникнуть следующие риски: после отрисовки макета может оказаться, что не предусмотрено место под важную информацию, придется переделывать дизайн и верстку или «подгонять» контент под дизайн; дизайнер может спроектировать блоки, под которые контент будет разрабатываться долго, в итоге затянутся сроки реализации веб-приложения.

Далее составляется визуальный прототип, который показывает внешний вид будущего веб-приложения с учетом схематичного расположения всех функциональных элементов вроде кнопок, форм, текстовых блоков, изображений, а также их иерархию. При этом все тексты находятся на своих местах.

На этапе проектирования архитектуры продумывается внутренняя логика работы веб-приложения с технической точки зрения: как веб-продукт работает, какие процессы запускаются в ответ на действия пользователей, какие планируются интеграции и какие API нужны для этого. Также прописывается, какие технологии и как будут использовать.

Проектированием архитектуры занимается команда из менеджера проектов, разработчиков и технического писателя. Для проектов со сложными интерфейсами этот этап делается параллельно с проектированием интерфейса. Тогда к работе подключаются дизайнеры.

На основе визуального прототипа на этапе дизайна разрабатывается дизайн-макет. На данном этапе задействованы веб-дизайнеры и арт-директор.

Сначала прорабатывается дизайн-концепция будущего веб-приложения. Ее задача – передать визуальную идею будущего веб-приложения. Концепция нужна, чтобы согласовать видение дизайна у заказчика и команды разработчиков до того, как будут отрисованы все макеты.

Дизайнеры составляют мудборд и подбирают визуальные образы под бизнес-задачи проекта. На мудборде отражается стиль будущего веб-продукта через цвета, шрифты, формы плашек, кнопок, инпутов и прочих элементов

взаимодействия. Также здесь закладывается анимация и интерактивные решения: как отрабатывают нажатия кнопок, как ведет себя страница при скролле, как отображается переход между экранами и т. д.

Далее дизайнеры, опираясь на текстовый и медиа-контент, визуальный прототип, мудборд разрабатывают макеты всех страниц веб-приложения, детально продумывают внешний вид элементов и эффекты при взаимодействии пользователя с интерфейсом. Для каждого экрана разрабатывают адаптивные макеты, чтобы веб-приложение хорошо выглядело и работало на разных устройствах с разными разрешениями экранов. Веб-приложение должен быть одинаково удобно и на компьютере, и на ноутбуке, и на смартфоне.

Арт-директор контролирует чистоту макетов и единство стилистики.

Если в проекте используется сложная анимация и визуальные эффекты, на этапе разработки дизайн-макетов можно подключить фронтенд-разработчиков, чтобы заранее продумать техническую реализацию.

Далее наступает этап фронтенд. Результат фронтенд-разработки – всё, с чем взаимодействует пользователь на сайте или в приложении. Главная цель этапа – сделать сайт удобным для пользователей и эффективным с точки зрения бизнеса.

На данном этапе задействованы верстальщики и фронтенд-разработчики. Они программируют визуальные решения, пользовательские сценарии и функциональность, заложенные дизайнерами.

Задачи данного этапа:

- написать скрипты, при помощи которых контент динамически подгружается на страницы при первом обращении к сайту. В результате страницы в браузере пользователя загружаются быстрее, а переход между ними выглядит более плавным;

- сделать элементы интерактивными, чтобы кнопки, инпуты, формы ввода и другие элементы логично реагировали на действия пользователя и вели к нужным конверсиям;

- реализовать сложные анимации и визуальные эффекты;

- написать API для интеграции с бэкендом или CMS-системой. То есть не просто создать клиентскую часть, но и связать все действия пользователя с бизнес-логикой веб-продукта;

- написать юнит-тесты для проверки после разработки каждой функции. Эти тесты нужны для проверки, насколько корректно работает та или иная функциональность.

Следующим этапом разработки веб-приложения является бэкенд-разработка. Веб-приложение без бэкенда – это не более, чем красивая оболочка. Чтобы веб-приложение функционировало, нужно запрограммировать логику хранения данных и правила обмена информацией между сервером и клиентской частью.

Чаще всего этот этап идет параллельно фронтенд-разработке, а разработчики бэкенда и фронтенда работают в тесной связке.

Бэкенд-разработчики программируют внутреннюю логику работы сайта или веб-продукта:

- программируют серверную часть сайта или приложения;
 - проектируют базы данных, в которых хранится вся информация о содержимом всех страниц сайта. например, информацию о товарах и категориях на сайте интернет-магазина;
 - настраивают пользовательские роли и управление сайтом в админ-панели;
 - для проектов на готовой CMS на этом этапе настраивают админ-панель для управления контентом на сайте;
 - разрабатывают API для интеграции со сторонними сервисами: почтой, программами складского и товарного учета вроде 1С, CRM, эквайрингом и т. д.
- После того, как веб-приложение готово, происходит его наполнение информацией. Этим занимается контент-менеджер. Он пишет статьи, добавляет новости, услуги, описывает товары.
- Далее идет этап продвижения веб-приложения. Этим может заниматься рекламщик, директолог, SEOшник, таргетолог.

2 Порядок выполнения и защиты курсовой работы

В соответствии с основными этапами разработки веб-приложений планируется работа над проектом (таблица 1). В процессе выполнения курсовой работы последовательно проходятся первые шесть этапов, уделяя основное внимание этапам дизайна, фронтенд- и бэкенд-разработки.

Таблица 1 – Этапы выполнения курсовой работы

Планируемая работа	Неделя семестра
1	2
Подготовительный этап	
1 Согласовать тему курсовой работы с преподавателем (примерная тематика курсовой работы представлена в приложении А)	1
2 Получить задание на курсовое проектирование	
3 Зарегистрироваться на GitHub, создать репозиторий и сообщить ссылку преподавателю	
Этап аналитики и проектирования	
4 Анализ предметной области	2
5 Анализ конкурентов и продуктов-аналогов	
6 Сбор и подготовка контента	3
7 Проектирование архитектуры проекта	
Этап проектирования дизайна веб-страниц	
8 Разработка дизайн-макетов	4
9 Верстка макетов	5–6

Окончание таблицы 1

1	2
Этап разработки веб-приложения	
10 Программирование: фронтенд	7–12
11 Программирование: бекенд	13–16
Заключительный этап	
12 Оформление пояснительной записки	16–17
13 Подготовка материалов к защите	
14 Защита курсовой работы	

Вся работа над проектом ведется в репозитории GitHub.

В своём GitHub-аккаунте создать публичный репозиторий. Название дать по названию проекта.

В конце этапа аналитики и проектирования в файле README.md ветки main следует разместить ссылку на файл либо прикрепить файл с результатами предпроектного исследования.

В начале этапа проектирования дизайна веб-страниц в файле README.md ветки main следует разместить ссылку на fig. файл с прототипами, дизайн-концепцией, дизайн-макетами.

На этапе разработки веб-приложения требуется делать не менее двух коммитов в неделю в репозиторий проекта.

На основе аналитики, прототипов, дизайн-концепции и дизайн-макетов студент готовит публикацию на конференцию, соответствующую тематике курсовой работы.

Для защиты курсовой работы готовится презентация, доклад и видеоролик продолжительностью не более 45 с, демонстрирующий основные приемы работы с веб-приложением.

Не позже чем за неделю до защиты курсовой работы на проверку руководителю сдается:

- пояснительная записка в печатном виде в папке-сборщике с подписями студента;
- пояснительная записка в электронном виде в формате .doc;
- UI-kit – готовый набор графических элементов в формате исходного файла (файл .fig, мудборд, иконки, картинки, шрифты и т. д);
- презентация.

Защита курсовой работы проводится в присутствии руководителя и преподавателей кафедры. На защите студент делает краткое сообщение о теме работы, целях и задачах работы, разработанном дизайн-решении (до 10 мин). Затем демонстрирует веб-приложение и отвечает на вопросы.

Работа студента оценивается в соответствии со следующими критериями.

Оценка «отлично» ставится, если:

- работа сдана вовремя;
- выполнены требования к коммитам в репозиторий проекта;

- имеется публикация (тезис доклада, статья) на научной конференции по тематике курсовой работы (или имеется информация, что материал принят к опубликованию);

- веб-приложение содержит функционал сверх установленного минимума;

- верстка соответствует дизайн-макетам;

- верстка корректно отображается на всех типах экранов;

- пояснительная записка содержит все разделы в соответствии с заданием, выполнена в полном объеме, грамотно, отсутствуют заимствования из методических указаний и элементы чужой работы, скопированный и не переработанный текст из интернета. Материал изложен логически связно, последовательно, полно. Используются дополнительные источники информации. Пояснительная записка оформлена аккуратно, в соответствии с требованиями раздел 7 методических рекомендаций;

- изложение доклада краткое, последовательное, логическое, язык грамотный, выразительный. Выводы аргументированы, доказательны, выделены отличительные черты проекта.

Оценка «хорошо» ставится, если:

- работа сдана с опозданием;

- коммиты в репозиторий проекта делались реже двух раз в неделю, но не реже одного раза в неделю. Названия коммитов не соответствуют гайдлайну;

- отсутствует публикация на научной конференции по тематике курсовой работы;

- реализованы минимальные требования к веб-приложению;

- имеются незначительные отклонения верстки от дизайн-макетов;

- верстка корректно отображается только на одном типе устройства;

- пояснительная записка содержит все разделы в соответствии с заданием, грамотно, отсутствуют заимствования из методических указаний и элементы чужой работы, скопированный и не переработанный текст из интернета. Материал изложен логически связно, последовательно. Часть пояснений изложена кратко, формально. Пояснительная записка оформлена аккуратно, но имеются грамматические ошибки, несущественные отклонения от требований раздел 7 методических рекомендаций;

- содержание и основная цель проекта раскрыты. Доклад изложен достаточно последовательно, грамотно, с выделением главных моментов. Отдельные принятые решения обоснованы недостаточно убедительно. В отдельных случаях имеется неправильное использование терминологии.

Оценка «удовлетворительно» ставится, если:

- работа сдана с опозданием;

- коммиты в репозиторий проекта делались реже одного раза в неделю. Названия коммитов не соответствуют гайдлайну;

- отсутствует публикация на научной конференции по тематике курсовой работы;

- реализованы минимальные требования к веб-приложению;

- имеются существенные отклонения верстки от дизайн-макетов;
- верстка корректно отображается только на одном типе устройства;
- пояснительная записка содержит все разделы в соответствии с заданием, имеются заимствования из методических указаний, элементы чужой работы, скопированный и не переработанный текст из интернета. Материал изложен логически связно, последовательно. Часть пояснений дана формально. Принятые решения при разработке программного продукта допустимы, но не обоснованы, не рациональны. Пояснительная записка оформлена аккуратно, но имеются грамматические ошибки, несущественные отклонения от требований раздел 7 методических рекомендаций;

- содержание и основная цель проекта раскрыты частично. Есть нарушения и в логике и в последовательности изложения доклада. Прослеживаются затруднения в определении главного и второстепенного при обобщении материала, в аргументации принятых в проекте решений. Допускается множественное неправильное использование терминологии.

3 Используемые инструменты и технологии

Визуальный прототип, дизайн-концепцию и дизайн-макеты разрабатывают в Figma, для создания иллюстраций, иконок визуальных эффектов можно использовать Photoshop и Illustrator, можно использовать отдельный софт для 3D- и моушен-графики.

Базовые технологии фронтенд-разработки – HTML5, CSS, JavaScript, TypeScript, библиотека React, Redux (Redux Toolkit) и др. Можно использовать bootstrap, material design, css-фреймворки, html и css препроцессоры.

Для бэкенд-разработки можно использовать системы управления базами данных MySQL, PostgreSQL, IndexedDB, MongoDB, языки программирования – PHP, Python, Node.js.

Используемая система контроля версий – GitHub.

В качестве инструмента для проверки соответствия вёрстки макету используется расширение PerfectPixel.

Для проверки валидности вёрстки используется сервис <https://validator.w3.org/>.

4 Минимальные требования к курсовой работе

Разрабатываемое веб-приложение должно удовлетворять следующим минимальным функциональным требованиям:

- содержать не менее 5 страниц. Наполнение страниц зависит от тематики курсовой работы;

- вёрстка адаптирована для десктопа, планшета и мобильного устройства. Корректность отображения приложения и отсутствие горизонтальной полосы

прокрутки проверяется при ширине страницы от 1920px до 320px;

- должна присутствовать интерактивность элементов, с которыми пользователи могут взаимодействовать, изменение внешнего вида самого элемента и состояния курсора при наведении, использование разных стилей для активного и неактивного состояния элемента, плавные анимации;

- должны быть реализованы сортировка, фильтрация и поиск, если это позволяет разрабатываемый продукт;

- выбранные пользователем параметры (фильтры, порядок сортировки, добавленные в избранное товары и т. д.) сохраняются при перезагрузке страницы (используется local storage). Есть кнопка сброса настроек, которая очищает local storage;

- наличие backend части.

Технические требования:

- верстка валидная;

- верстка семантическая;

- обязательно использование библиотеки React;

- работа приложения проверяется в браузере Google Chrome последней версии.

Требования к коммитам:

- история коммитов должна отображать процесс разработки приложения;

- названия коммитам даны согласно гайдлайну [1] (приложение Б).

5 Требования к структуре курсовой работы

Объем пояснительной записки курсовой работы – 30–35 страниц формата А4.

Пояснительная записка имеет следующую структуру:

- титульный лист;

- задание на курсовую работу;

- содержание;

- введение;

- основная часть;

- заключение;

- список использованных источников;

- приложения.

Основная часть пояснительной записки имеет следующую структуру.

1 Предпроектное исследование.

1.1 Обоснование и цели проекта.

1.2 Конкурентная аналитика.

1.3 Текстовый прототип.

1.4 Визуальный прототип.

1.5 Архитектура проекта.

2 Проектирование дизайна веб-страниц.

- 3 Разработка веб-приложения.
- 3.1 Выбор технологий и инструментов.
- 3.2 Разработка клиентской части.
- 3.3 Разработка серверной части.
- 4 Руководство системного программиста.
- 5 Описание тестового примера.

6 Методические рекомендации к содержанию введения, основной части и заключения курсовой работы

6.1 Введение

В общем случае во введении следует:

- охарактеризовать проблему, к которой относится тема работы;
- указать цель выполнения работы;
- изложить задачи, которые необходимо решить в процессе выполнения работы;
- обозначить объект и предмет исследования;
- изложить ожидаемые результаты.

6.2 Обоснование и цели проекта

В данном разделе формулируются бизнес-цели создаваемого проекта, исследуется пользовательская аудитория и формулируются проблемы, которые будет решать разрабатываемое веб-приложение.

Следует выяснять, есть ли похожие веб-приложения и существует ли вообще рынок. Если похожие веб-приложения есть, значит рынок есть. В этом случае следует проанализировать, есть ли простор для развития и есть ли потребность именно в том приложении, которое будет реализовано в рамках курсовой работы. Например, можно посмотреть, сколько пользователей у конкурентов, есть ли вариации приложений.

Если таких веб-приложений нет, значит ниша либо не занята либо кто-то уже пытался делать подобное приложение и провалился.

Для сбора и анализа маркетинговой информации можно воспользоваться аналитическим сервисом Google Trends [2].

Если приложение разрабатывается для конкретной компании (фирмы, бренда), то следует изучить информацию о компании (фирме, бренде), проанализировать существующие решения компании (фирмы, бренда), изучить недостатки существующих решений компании (фирмы, бренда), сформулировать цель разработки нового веб-приложения и сформировать перечень новых задач.

6.3 Конкурентная аналитика

Основная задача данного раздела – изучить аналоги конкурентов. Следует провести анализ сильных и слабых сторон конкурентов и продуктов-аналогов, собрать информацию о технологиях у продуктов-аналогов.

Рекомендуется найти и проанализировать не менее пяти продуктов-аналогов и оценить, что понравилось или не понравилось в этих сайтах. Аргументировать свою позицию. Например, если понравился дизайн, то описать, чем конкретно он хорош. Обратит внимание на структуру сайтов (разделы и подразделы), функционал (есть ли там какие-то интерактивные сервисы как: комментарии, регистрация, поиск по параметрам и т. д.), на качество подачи информации, насколько детально описаны товары и продукция, есть ли фотографии, статьи, примеры по теме и т. д. Анализ следует проводить с точки зрения, что хорошо, а что плохо именно для вашего веб-приложения. Следует отобрать хорошие качества дизайна, структуры и функционала, которые хотелось бы реализовать в Вашем проекте по четырем направлениям: функционал, структура, дизайн, информация.

Перед выполнением данного раздела рекомендуется ознакомиться со статьей из [3].

Примеры анализа аналогов разрабатываемого веб-приложения представлены в приложении В.

6.4 Текстовый прототип

Прототип сайта или каркас (англ. Wireframe (Website wireframe)) – это базовый макет сайта, который визуализирует расположение всех элементов и функций.

Разработка прототипа помогает решить следующие проблемы и задачи:

- визуализация идеи, представление о внешнем виде будущего сайта на ранних сроках. Ситуация, когда на начальном этапе работы нет четкого видения будущего сайта, далеко не редкость. Случаются и противоречия внутри команды, когда у каждого из участников есть свое видение. Прототип помогает визуализировать все идеи и прийти к компромиссу без необходимости переделывать уже готовый дизайн;

- возможность внести изменения и финализировать видение ценой минимальных расходов;

- возможности быстрого внесения изменений в функциональность продукта без кода;

- объяснение логики взаимодействия пользователя с продуктом;

- презентация идеи;

- возможность более точной оценки сроков и бюджета полного объема работ;

- понимание перспектив использования и развития сайта, раннее обсуждение гипотез.

В разработке лендинга, прототип, и небольшие текстовые пояснения по технической части – это есть готовое техническое задание для работы веб-дизайнера.

Если в работе многостраничный сайт, то редко когда нужно прототипировать все страницы, достаточно спланировать ключевые, например, главную страницу, каталог, карточку товара, страницу чек-аута.

Текстовый прототип – самый простой и быстрый вариант разработки. Как правило, он включает в себя информацию о разделах, заголовках и представляет собой список с необходимыми компонентами. Иногда такой список разделен на блоки, что помогает лучше структурировать информацию, в результате чего она легче воспринимается.

Пример текстового прототипа представлен в приложении Г.

6.5 Визуальный прототип

Визуальный прототип создается с целью заложить структуру уникальных страниц и не упустить важные контентные блоки. Он помогает схематически визуализировать основные элементы и функции веб-приложения. В визуальном прототипе наглядно изображаются структура, элементы интерфейса, расположение иллюстраций и кнопок, меню и иконок, но не оформление.

Визуальный прототип – это изображение результата, который должен создать дизайнер на этапе разработки дизайн-макета.

Пример визуального прототипа представлен на рисунке 2.

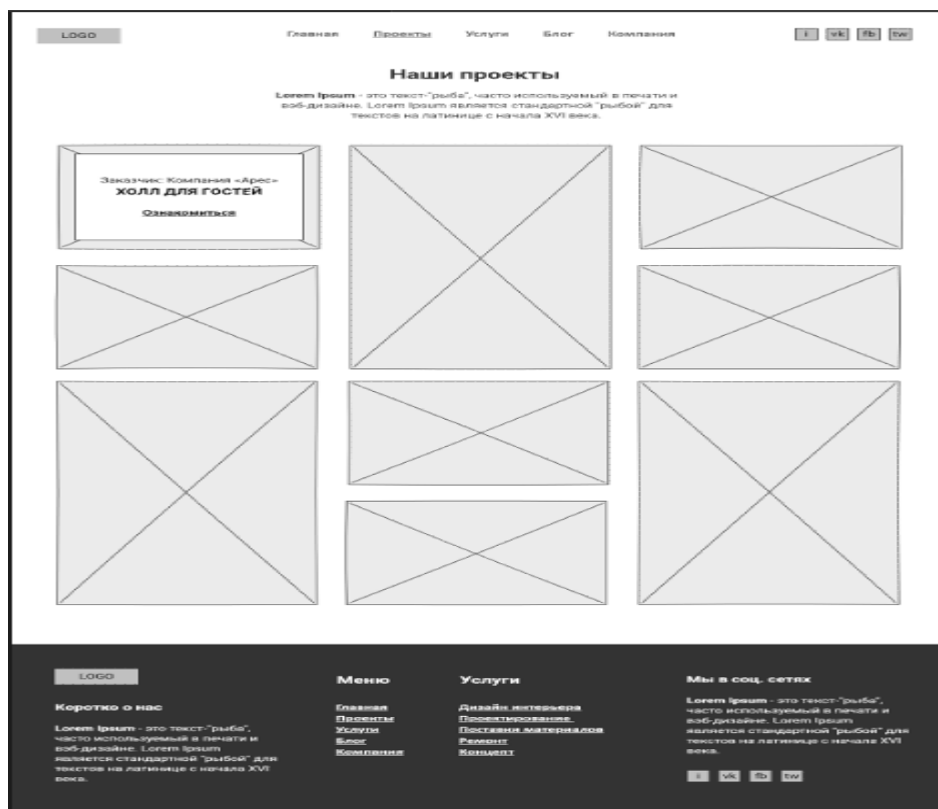


Рисунок 2 – Пример визуального прототипа дизайн-студии

В данном разделе курсовой работы необходимо разработать визуальный прототип веб-приложения. Должны быть учтены все требования, описанные в текстовом прототипе.

Так как в курсовой работе разрабатывается адаптивный (или отзывчивый) веб-дизайн, то для каждого окна необходимо разработать три варианта прототипа – для браузера, планшета и мобильного устройства.

6.6 Архитектура проекта

Архитектура сайта – это способ организации страниц, доступа к ним и навигации. В нее входит навигация по страницам, сеть ссылок, «хлебные крошки», страницы категорий, файлы карты сайта, контент и другие элементы, из чего сайт состоит.

Правильная архитектура помогает пользователям и поисковым системам находить то, что они ищут. Кроме того, она говорит системе о значимости и релевантности контента. Она направляет пользователя и поисковых ботов на самые важные страницы, рассказывает, что собой представляет контент.

Работа над архитектурой – работа для того, чтобы сайт был простым и понятным, удобным и приятным.

Оптимизаторы и дизайнеры придерживаются правила «трех кликов». Оно говорит, что любая важная страница сайта должна быть на расстоянии не более трех кликов от домашней страницы сайта (или, возможно, другой страницы с высоким авторитетом).

Следует иметь в виду, что это скорее это скорее принцип, чем правило. Есть много ситуаций, когда архитектура с тремя кликами не имеет смысла. Главное правило – располагать ключевые страницы как можно ближе к домашней.

Поместить важные страницы ближе к главной помогает так называемая плоская структура сайта, где для перемещения с главной на любую внутреннюю страницу нужно как можно меньше кликов, при этом все страницы связаны (рисунок 3).

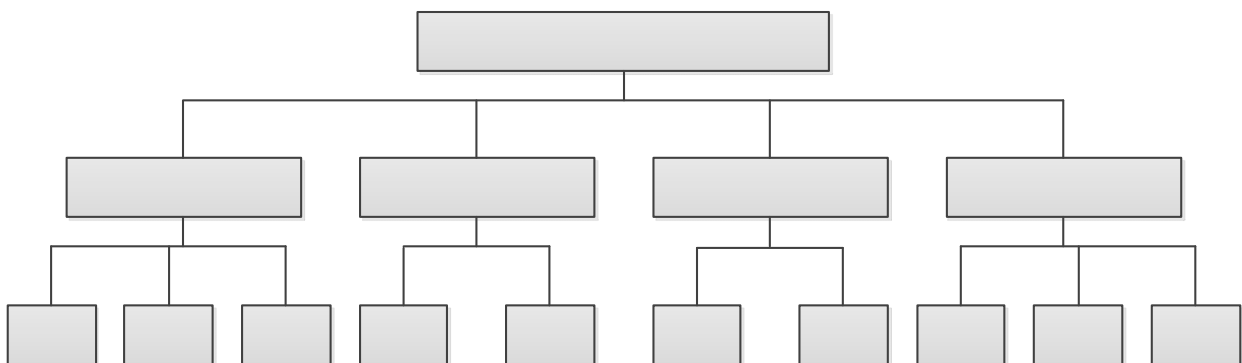


Рисунок 3 – Плоская структура сайта

Такую организованную структуру можно сделать по модели SILO. SILO-структура («бункер», «закрытая система») – это плоская структура ресурса,

основанная на семантике контента, подразумевающая распределение тем по иерархии.

Страница-хаб (или страница-концентратор) объединяет контент в общую ветку, а структура SILO отвечает за распределение внутри этой ветки (рисунок 4). Каждая ступень иерархии линкуется со ступенями выше и ниже себя (на рисунке 4 обозначены сплошными линиями). Страницы-хабы находятся вверху иерархии и обычно содержат навигацию (в том числе «хлебные крошки»), контекстные ссылки, структуру URL.

Кроме ссылок в иерархии, страницы одного уровня могут быть также перелинкованы между собой (на рисунке 4 обозначены пунктирными линиями).

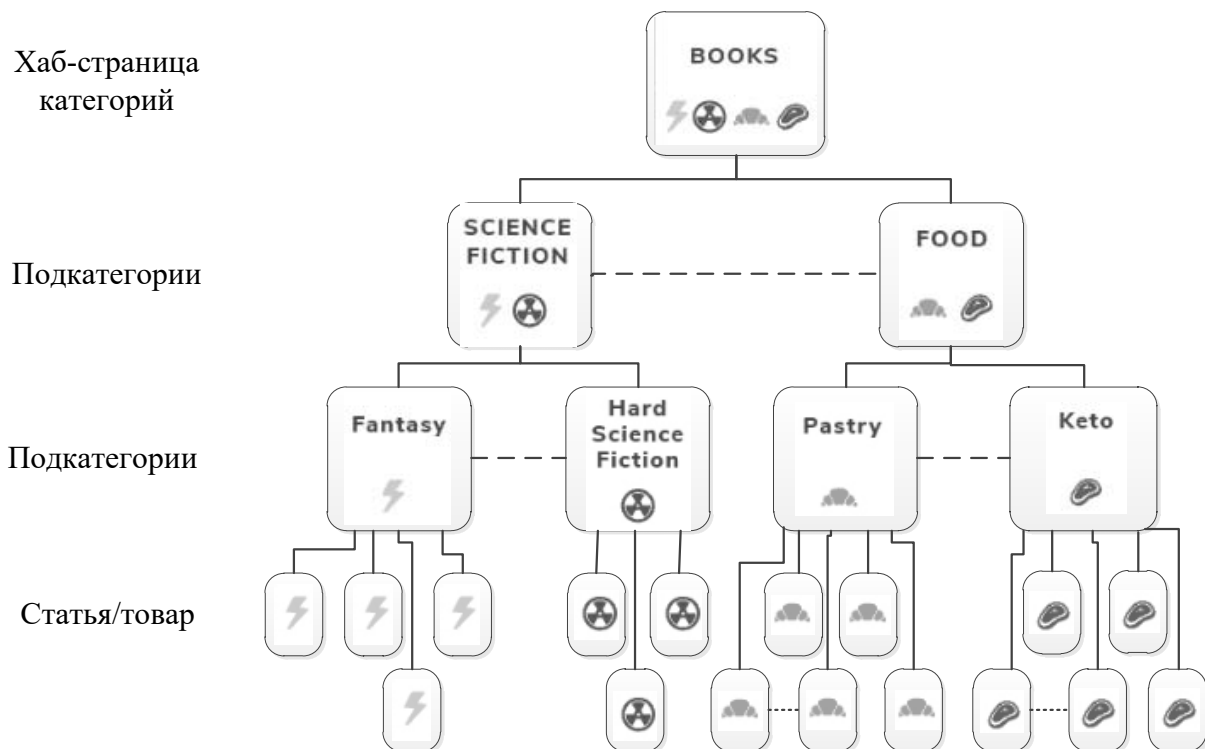


Рисунок 4 – SILO-структура контента веб-приложения

В данном разделе курсовой работы необходимо разработать структуру сайта, описать сеть ссылок, «хлебные крошки», страницы категорий, логику размещения контента, обосновать наличие связи между страницами.

Перед выполнением данного раздела рекомендуется ознакомиться со статьями из [4–6].

6.7 Проектирование дизайна интерфейса

На данном этапе необходимо разработать дизайн-концепцию веб-приложения и воплотить ее в виде дизайн-макетов веб-страниц.

Результат этого этапа – набор всех страниц сайта или экранов приложения под все необходимые разрешения и устройства, правила использования всех элементов на них.

На этапе концепции следует продумать стилистику веб-приложения, подобрать визуальные образы под бизнес-задачи проекта, отразить стиль будущего веб-продукта через цвета, шрифты, формы плашек, кнопок, инпутов и прочих элементов взаимодействия.

Дизайн-концепция презентуется в форме мудборда – подборки изображений, характеризующих будущий дизайн-проект.

Она олицетворяет результат, который хочет получить заказчик, и может состоять из нескольких различных элементов:

- цветовых схем;
- шрифтов и начертаний;
- слоганов;
- паттернов;
- иллюстраций;
- фотографий.

Все перечисленные графические материалы указывают на настроение и дух проекта и размещаются в форме коллажа. Сделать его можно в фоторедакторе или одной из предназначенных для этого программ.

Перед выполнением данного раздела рекомендуется ознакомиться с материалом из [7].

На этапе разработки дизайн-макетов развивается стилистика для всех страниц сайта, детально продумывается внешний вид элементов и эффекты при взаимодействии пользователя с интерфейсом, закладывается анимация и интерактивные решения: как отрабатывают нажатия кнопок, как ведет себя страница при скролле, как отображается переход между экранами и т. д.

Планом для оформления всех экранов являются текстовый и визуальный прототипы.

Для каждого экрана разрабатываются адаптивные макеты для устройств с разными разрешениями экранов: десктопная версия, для планшета, мобильного устройства.

В пояснительной записке для каждой основной страницы веб-приложения приводят дизайн-макеты для трех разрешений и словесное описание происходящих на них процессах и явлениях. Пример описания структуры веб-страницы приведен в приложении Д.

6.8 Выбор технологий и инструментов

Данный подраздел содержит описание используемых программных средства для разработки дизайн-концепции и дизайн-макетов страниц веб-приложения, описание языков программирования, фреймворков, плагинов, CMS, баз данных, используемых для программной реализации веб-приложения с обоснованием выбора.

6.9 Разработка клиентской части

В данном подразделе необходимо описать разработанные компоненты для каждой веб-страницы. Описание можно сопроводить UML-диаграммой компонентов или любой другой диаграммой, наглядно демонстрирующей структуру приложения.

Описать API для интеграции с бэкендом приложения.

6.10 Разработка серверной части

В данном подразделе проектируется и описывается база данных. Структура базы данных проектируется в программе Enterprise Architect. Для каждой таблицы необходимо дать словесное пояснение о содержащейся в ней информации, обосновать установленные связи «один-ко-многим».

Далее следует описать все запросы к базе данных. Тексты запросов можно привести в приложении к пояснительной записке курсовой работы.

Если проект реализуется, на готовой CMS описываются настройки админ-панели для управления контентом на сайте.

Описываются API для интеграции со сторонними сервисами и структура JSON-файлов, в случае их использования.

6.11 Руководство системного программиста

Описывается порядок установки программного продукта для выполнения тестового примера.

Пример:

на компьютере должна быть установлена платформа node.js 16.16.0. Официальные установочные пакеты для всех основных платформ можно скачать с сайта <https://nodejs.org> и т. д.

6.12 Описание тестового примера

Тестовый пример предназначен для демонстрации возможностей веб-приложения. Тестовый пример должен охватывать все основные функции веб-приложения. Пример приведен в приложении Е.

6.13 Заключение

В заключении должно быть указано:

- все ли цели курсового проектирования были достигнуты;
- все ли задачи курсовой работы были выполнены;
- выводы о достоинствах и недостатках разработанного в рамках курсовой работы веб-приложения;

- предложения по дальнейшему развитию и усовершенствованию;
- трудности, возникшие в рамках курсового проектирования и пути их преодоления;
- результаты самостоятельного изучения инструментальных средств и технологий.

7 Требования к оформлению пояснительной записки

Титульный лист пояснительной записки оформляется в соответствии с требованиями ГОСТ Р 2.105–2019 и содержит информацию: об учебном заведении, специальности, теме курсовой работы, разработчике, руководителе работы и другие данные. Образец титульного листа представлен в приложении Ж.

Задание на курсовую работу выдается руководителем работы и оформляется на бланке установленного образца.

Пояснительная записка оформляется в соответствии с ГОСТ Р 2.105–2019 *Единая система конструкторской документации. Общие требования к текстовым документам с применением ЭВМ.*

Шрифт – Times New Roman, размер шрифта – 14 пт, межстрочный интервал – 1,15 см, отступ «красной строки» – 1,25 см.

Расстояние от внутренней рамки формата до границ текста:

- слева и справа – 5 мм;
- сверху и снизу – 10 мм.

Пояснительную записку нумеруют сквозной нумерацией, отсчет ведут с титульного листа. Задание не нумеруется.

Список использованных источников оформляется в соответствии с ГОСТ Р 7.0.100–2018 *Библиографическая запись. Библиографическое описание. Общие требования и правила составления.*

Список должен содержать не менее 15 источников. На все источники должны быть даны ссылки в тексте пояснительной записки.

Материал, дополняющий основной текст пояснительной записки, допускается помещать в **приложениях**. Приложения должны быть перечислены в «Содержании» курсовой работы с указанием их заголовков и страниц. В тексте работы на все приложения должны быть даны ссылки. Приложения располагают в порядке ссылок на них в тексте курсовой работы.

Основные надписи выполняют в соответствии с ГОСТ 2.104–2006 *Единая система конструкторской документации. Основные надписи.*

Содержание оформляют на листе с основной подписью формы 2 (40 мм, ГОСТ 2.104–2006, рисунок 5), а последующие листы – с основной надписью формы 2а (15 мм, ГОСТ 2.104–2006, рисунок 6).

Приложения оформляются на листах без рамки, включаются в сквозную нумерацию, номер страницы проставляется внизу, посередине листа.

					КР XX.XX.XX.XXXXXXXXXX.XX ПЗ					
Изм.	Лист	№ докум.	Подп.	Дата						
Разраб.	ФИО студента				Тема курсовой работы			Лит.	Лист	Листов
Пров.	ФИО преподав.								2	
Н.контр.					Курсовая работа			БРУ гр. № группы		
Утв.										

Рисунок 5 – Основная надпись для заглавного листа

					КР XX.XX.XX.XXXXXXXXXX.XX ПЗ				Лист
Изм.	Лист	№ Докум.	Подп.	Дата					4

Рисунок 6 – Основная надпись для последующих листов

Расшифровка шифра в основной надписи и титульном листе приведена на рисунке 7.

КР XX.XX.XX.XXXXXXXXXX.XX ПЗ

Рисунок 7 – Расшифровка шифра

Список литературы

1 Соглашение о коммитах [Электронный ресурс]. – Режим доступа: <https://www.conventionalcommits.org/en/v1.0.0/>. – Дата доступа: 01.12.2022.

2 Как пользоваться Google Trends: полное руководство. [Электронный ресурс]. – Режим доступа: <https://timeвеб.com/ru/community/articles/kak-polzovatsya-google-trends-polnoe-rukovodstvo>. – Дата доступа: 01.12.2022.

3 Как делать конкурентный анализ для эффективного продвижения приложения? [Электронный ресурс]. – Режим доступа: <https://vc.ru/marketing/224293-kak-delat-konkurentnyy-analiz-dlya-effektivnogo-prodvizheniya-prilozheniya>. – Дата доступа: 01.12.2022.

4 Полное руководство по планированию архитектуры сайта: 15 советов для максимального SEO [Электронный ресурс]. – Режим доступа: <https://semantica.in/blog/polnoe-rukovodstvo-po-planirovaniyu-arkhitektury-sajta-15-sovetov-dlya-maksimalnogo-seo.html>. – Дата доступа: 01.12.2022.

5 15 советов по архитектуре сайта: как составить SEO-структуру, навигацию, разметку и перелинковку [Электронный ресурс]. – Режим доступа: <https://pr-cy.ru/news/p/7111-15-sovetov-po-seo-arkhitecture-sayta>. – Дата доступа: 01.12.2022.

6 Как сделать правильную структуру сайта [Электронный ресурс]. – Режим доступа: <https://protraffic.com/baza-znaniy/kak-sdelat-pravilnuyu-strukturu-veb-saita-13726.html>. – Дата доступа: 01.12.2022.

7 Что такое мудборд и зачем дизайнеры используют его [Электронный ресурс]. – Режим доступа: <https://timeвеб.com/ru/community/articles/chto-takoe-mudbord>. – Дата доступа: 01.12.2022.

8 **Немцова, Т. И.** Компьютерная графика и веб-дизайн: учебное пособие / Т. И. Немцова, Т. В. Казанкова, А. В. Шнякин; под ред. Л. Г. Гагариной. – Москва: ФОРУМ; ИНФРА-М, 2020. – 400 с.

Приложение А (справочное)

Примерная тематика курсовой работы

- 1 Проектирование и разработка веб-приложения по покупке авиабилетов.
- 2 Проектирование и разработка онлайн магазина.
- 3 Проектирование и разработка интерфейса веб-приложения «Автосалон».
- 4 Проектирование и разработка веб-приложения «Салон красоты».
- 5 Проектирование и разработка веб-приложения «Ресторан».
- 6 Проектирование и разработка мобильного приложения банка.
- 7 Проектирование и разработка веб-приложения продуктового магазина.
- 8 Проектирование и разработка веб-приложения по покупке парфюмерии, уходовой и декоративной косметики.
- 9 Проектирование и разработка веб-приложения фитнес-центра.
- 10 Проектирование и разработка интерфейса веб-приложения магазина одежды.
- 11 Проектирование и разработка веб-приложения учебного заведения.
- 12 Проектирование и разработка веб-приложения туристического агентства.
- 13 Проектирование и разработка веб-приложения отеля.
- 14 Проектирование и разработка веб-приложения по изучению иностранного языка.
- 15 Проектирование и разработка веб-приложения кинотеатра.
- 16 Проектирование и разработка веб-приложения страховой компании.
- 17 Проектирование и разработка веб-приложения по обмену книгами.
- 18 Проектирование и разработка веб-приложения «Я помогаю детям».
- 19 Проектирование и разработка веб-приложения по интересам.

Приложение Б (справочное)

Соглашение о коммитах

Требования к именам коммитов:

- названия коммитов должны быть согласно гайдлайну [1];
- должен использоваться present tense ("add feature" not "added feature");
- должен использоваться imperative mood ("move cursor to..." not "moves cursor to...").

Примеры имен коммитов:

- `init`: используется для начала проекта/задачи. Примеры:

```
init: start youtube-task
init: start mentor-dashboard task
```

- `feat`: это реализованная новая функциональность из технического задания (добавил поддержку зумирования, добавил footer, добавил карточку продукта). Примеры:

```
feat: add basic page layout
feat: implement search box
feat: implement request to youtube API
feat: implement swipe for horizontal list
feat: add additional navigation button
feat: add banner
feat: add social links
feat: add physical security section
feat: add real social icons
```

- `fix`: исправил ошибку в ранее реализованной функциональности. Примеры:

```
fix: implement correct loading data from youtube
fix: change layout for video items to fix bugs
fix: relayout header for firefox
fix: adjust social links for mobile
```

- `refactor`: новой функциональности не добавлял / поведения не менял. Файлы в другие места положил, удалил, добавил. Изменил форматирование кода (white-space, formatting, missing semi-colons, etc). Улучшил алгоритм, без изменения функциональности. Примеры:

```
refactor: change structure of the project
refactor: rename vars for better readability
refactor: apply eslint
refactor: apply prettier
```

- `docs`: используется при работе с документацией/readme проекта. Примеры:

```
docs: update readme with additional information
docs: update description of run() method
```


Приложение В (справочное)

Примеры аналитических отчетов

Веб-приложения по продаже кондитерских изделий

Для анализа будет рассмотрено три веб-приложения, посвященных продаже кондитерских изделий, печенья и конфет:

- 1) магазин кондитерских изделий «Сказка», <http://skazkatd.ru/> (рисунок XX);
- 2) фабрика печенья «Авангард», <http://www.avangard-pf.ru/> (рисунок XX);
- 3) оптовая компания ООО «Паллада», <http://www.pallada-mo.ru/> (рисунок XX).

В рассматриваемых веб-приложения будут проанализированы следующие характеристики: эргономичный дизайн, наличие электронного каталога с названием, описанием, фотографией, ценой; возможность оформить заказ онлайн; возможность безналичной оплаты онлайн; возможность отправить заявку менеджеру.

Для оценки характеристик будет использоваться 5-балльная шкала, где 1 – функция отсутствует, а 5 – функция есть. Сравнение характеристик рассматриваемых веб-приложений представлено в таблице В.1.

Таблица В.1 – Сравнение характеристик рассматриваемых веб-сайтов

Характеристики	Веб-приложения		
	Магазин кондитерских изделий Сказка, http://skazkatd.ru/	Фабрика печенья «Авангард», http://www.avangard-pf.ru/	Оптовая компания ООО «Паллада», http://www.pallada-mo.ru/
1 Эргономичность дизайна	3 (дизайн сайта устарел, не отслеживается структура сайта)	5	4 (устаревшая структура сайта, неудобное расположение элементов)
2 Наличие электронного каталога с названием, описанием, фотографией, ценой	2 (отсутствует описание, фотография и цена)	4 (отсутствует цена)	5
3 Возможность оформить заказ онлайн	1	1	1
4 Возможность безналичной оплаты онлайн	1	1	1
5 Возможность отправить заявку менеджеру	1	3 (быстро заявку по рассматриваемому товару отправить нельзя, есть форма обратной связи на странице Контакты)	4 (реализовано, но не как функция «быстрого заказа»)
Итого	8	14	16

По анализу приложений сделаны следующие выводы.

Оптимальным сайтом является сайт оптовой компании ООО «Паллада», <http://www.pallada-mo.ru/> – данный сайт имеет функционал для просмотра кондитерских изделий, печенья и конфет с возможностью оставить заявку на приобретение интересующей продукции. Недостаток сайта заключается в устаревшем дизайне, из-за чего конверсия продаж не растет.

Оптимальный дизайн представлен сайтом фабрики печенья «Авангард», <http://www.avangard-pf.ru/>.

Приложения для демонстрации рекламы

Для сравнения были выбраны клиентские приложения от компаний Scala, SignBox, Bright Sign.

Система отображения рекламы Scala, схема которой изображена на рисунке В.1, включает в себя дизайнер как способ размещения контента и системы контроля устройств, менеджер контента в качестве хранилища и плеер в качестве клиентского приложения. Клиентское приложение содержит в себе набор функций, таких как воспроизведение видео, отображение изображений и веб-сайтов, загрузку контента на устройство для воспроизведения, т. е. предоставляет аналогичные возможности клиентскому приложению в режиме циклического воспроизведения. В качестве недостатков данного решения можно отметить то, что оно не предоставляет статистику по показам, и не предоставляет интерфейса для дистанционного вызова функций.

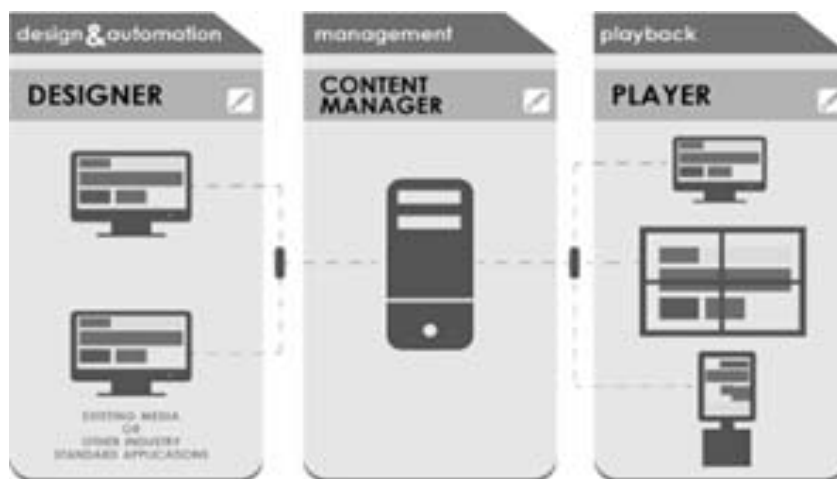


Рисунок В.1 – Составные компоненты системы Scala

Компания BrightSign предоставляет готовые устройства для демонстрации рекламного контента. Данные устройства специально разработаны для показа рекламы и не предназначены для иных целей. С поставляемым устройством в комплекте идет приложение настройки устройства и создания списка контента. Рассмотрим его подробнее.

На рисунке В.2 изображен внешний вид настройки приложения. Он включает в себя похожие параметры настроек, как и клиентское приложение: имя устройства, описание, имя пользователя и пароль. Из данного рисунка, можно заметить, что устройство также включает возможность ведения журнала воспроизведения, событий, состояния и диагностики, что позволяет осуществлять дальнейшую поддержку устройства и сбор статистики об осуществленных показах. Следующая возможность приложения – возможность создавать презентации и осуществлять загрузку на устройство через файловую систему или интернет.



Рисунок В.2 – Настройка устройства BrightSign

В соответствии с рисунком В.3, можно увидеть, что клиентское приложение поддерживает несколько зон воспроизведения рекламы.



Рисунок В.3 – Шаблоны клиентского приложения BrightSign

Также данное приложение позволяет создать плейлист и опубликовать его в сети BrightSign. Этот процесс демонстрируется на рисунке В.4.

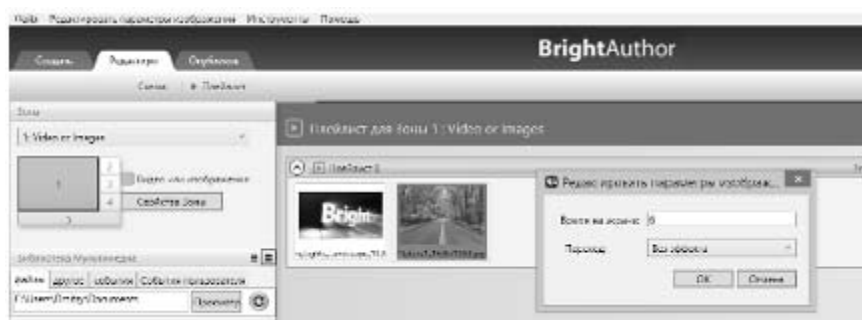


Рисунок В.4 – Процесс создания плейлиста

Процесс создания рекламных плейлистов был продемонстрирован с целью показать важную функцию решения: взаимодействие с устройством для показа рекламы без использования интернет-соединения, что не поддерживается клиентским приложением. В остальном, как было показано на рисунках, клиентское приложение в режиме циклического воспроизведения обладает такими же функциями, как и решение BrightSign. В качестве минуса использования данного решения можно отметить тот факт, что компания не поддерживает

загрузку рекламы на устройства, не принадлежащим рекламодателям, соответственно, сужает объем аудитории для рекламодателя.

Решение компании signBox представляет собой систему, сходную с системой Scala, т. е. веб-сайт для управления устройствами, хранилище данных и клиентское приложение, устанавливаемое на устройства. Клиентское приложение имеет инструмент SignEye для обеспечения выдачи таргетированного контента для определенной аудитории. Данный инструмент анализирует следующие параметры аудитории: возрастная группа, пол, число людей смотрящих на экран, число людей смотрящих на экран не внимательно, время просмотра рекламы пользователем. В совокупности система компании SignBox предоставляет больше возможностей анализа параметров аудитории, чем клиентское приложение для показа городской рекламы, но данное решение не распространено на российском рынке.

Разработанное клиентское приложение содержит большинство функций предоставляемых описанными компаниями. Сильной стороной приложения является демонстрация таргетированной рекламы, используя информацию о поле наблюдателя, которое не распространено на российском рынке, но существует и реализуется иностранными компаниями. Сравнение клиентских приложений по предоставляемым функциям приведено в таблице В.2.

Таблица В.2 – Наличие функций в клиентских приложениях

Наименование функции	Scala	BrightSign	Signbox	Разработанное приложение
Управление и воспроизведение медиа-контента	+	+	+	+
Поддержка нескольких зон воспроизведения	+	+	+	+
Показ таргетированной рекламы	–	–	+	+
Анализ внимания пользователя	–	–	+	–
Показ рекламы при обнаружении движения	–	–	–	+
Взаимодействие с приложением с помощью сенсорного экрана	+	+	+	–
Сбор и отправка статистики о показах	–	+	+	+

Приложение Г (справочное)

Примеры текстовых прототипов веб-приложения

Текстовый прототип для компании, специализирующейся на профессиональной разработке дизайна и реализации уникальных интерьеров

Базовая структура сайта: Главная, Проекты, Услуги, Блог, Информация о компании.

Главная страница.

Желаемая структура главной страницы.

- 1 Шапка в виде слайдера с крутыми работами, сверху – Логотип, Меню, обязательно иконки социальных сетей, где Instagram на первом месте.
- 2 Секция доверия, с нашими достижениями, телефоном и кнопкой для обратной связи.
- 3 Секция с лучшими работами с возможностью «Подгрузить работы» ниже. По сути, здесь должны выводиться лучшие работы из портфолио (меню «Проекты»).
- 4 Секция с краткой информацией о нас, вызывающая доверие, сделать акцент на преимущества нашей компании. Оформить по своему вкусу.
- 5 Обязательно разместить секцию партнеров и поставщиков, с которыми работаем.
- 6 В последней секции можно разместить последние записи блога с возможностью перейти в раздел «Блог».

Проекты.

Страница проектов должна быть лаконична и отображать проекты в виде карточек – от лучших до «худших». В админ-панели должна быть возможность сортировки работ контент-менеджером.

На странице каждого проекта обязательно должна быть карусель изображений сверху, под заголовком с фотографиями объекта, ниже спецификация (основные данные) проекта, а также должен быть указан старший архитектор. Текстовая часть оформляется в свободном стиле с возможностью добавлять фотографии и форматировать текст.

Услуги.

На данной странице должен быть оформлен список услуг по вашему вкусу. Отдельная страница услуги, блога или преимущества (универсальная страница) должна содержать фотографию поста сверху и текст. Внизу страницы должна быть кнопка, при клике на которую пользователь попадает в корневой раздел, например, блог, все услуги или все преимущества.

Список наших услуг (ключевые направления, по важности): **Дизайн интерьера, Проектирование (пространства), Поставки материалов, Ремонт и реставрация, Создание концепций.**

Блог.

На данной странице должен быть оформлен список статей. Обязательно должна быть указана дата публикации (без года), заголовок, фотография, краткий текст и ссылка на полную статью. Все статьи должны выводиться одной лентой, без разделов. Навигацию между старыми и новыми записями блога можно сделать обычной – вперед, назад и, возможно, промежуточные страницы 1, 2, 3 и т. д. (пагинация).

Информация о компании.

1 Компания / О компании.

На данной странице должен быть оформлен текст «О компании». Обязательно разместить фотографию нашей инсталляции в тексте.

2 Компания / Преимущества.

Здесь следует оформить список наших преимуществ. Можно оформить как-то интересно, добавить иконки и т. д., на ваш вкус.

3 Компания / Наши партнеры.

Список наших партнеров с логотипом партнёра и текстом о каждом партнере. Отдельная страница для каждого партнёра пока не требуется.

4 Компания / Контакты.

На данной странице обязательно разместить форму обратной связи в видимой части. Операторы компании отвечают достаточно быстро, поэтому можно указать и эту информацию где-нибудь. Оформить адрес и телефон. Вставить карту с адресом из какого-нибудь сервиса – Yandex, Google и т.д.

Текстовый прототип для школы танцев

1 экран.

8 (987) 654-32-10

Заказать звонок

Танцевальная школа KiDance объявляет набор детей в возрасте от 5 до 15 лет в г. Москва

Первое занятие – бесплатно

Записаться на пробное занятие

Ваше имя:

Ваш телефон:

Записаться

Мы свяжемся с Вами в ближайшее время

2 экран.

6 причин почему вам стоит отдать ребенка на танцы:

Активное физическое и умственное развитие

Во время танца ребенок контролирует мышцы, суставы, мимику, запоминает и выполняет сложные элементы.

Танцы – это залог крепкого здоровья вашего ребенка

Положительно влияют на мышцы, суставы, работу сердца. Предотвращают сколиоз и плоскостопие.

Развитие музыкального слуха

Ребенок учится «считать» музыку, слышать ритм, выражает эмоции во время зажигательного танца.

Танцы – отличный способ самовыражения

Во время танца ребенок растет личностно, раскрывает скрытые таланты и возможности. Реализовывает свой потенциал.

Танцы дарят чувство свободы

Ваш ребенок забудет про комплексы, ведь в танцах он обретет уверенность в себе и будет раскрыт перед миром.

Танцы – отличный способ побороть беспокойство

Вашего ребенка что-то тревожит? Отдайте его на танцы и он забудет о любых тревогах. Станет стрессоустойчивым и сумеет постоять за себя.

Оставить заявку на пробное занятие
Оставить заявку

3 экран.

Кто будет заниматься с вашим ребенком?

Знакомьтесь, основательница и хореограф школы KiDance – Марина Немцова

Выпускница СГАКИ '15 (Кафедры хореографии) МГУКИ '16

7 лет опыт работы детским хореографом (с 2007 года)

Солистка ансамбля, лауреатов всероссийских и международных конкурсов «Волжские узоры»

Успешно прошла курсы для повышения квалификации «Педагогика и Психология»

Оставить заявку на пробное занятие

Оставить заявку

4 экран.

Вот таким чудесным образом идет процесс нашей работы

5 экран.

Готовы подарить ребенку эмоции, которые не заменит никакой iPad?

Внесите больше жизни и радости в дни вашего ребенка. Для этого:

Запишитесь на пробное занятие по телефону +7 (962) 944-47-70

Выберите танцевальное направление

Приобретите месячный абонемент

или разовые занятия по выгодным вам условиям

Записаться на пробное занятие

Ваше имя:

Ваш телефон:

Записаться

Мы свяжемся с Вами в ближайшее время

Текстовый прототип для интернет-магазина

1 Главная страница содержит все товары магазина, а также фильтры, строку поиска, поле для сортировки.

2 Карточка товара содержит изображение товара, название, количество данного товара на складе, год выхода на рынок, цвет, производитель и т. д., находится ли товар в корзине.

Кликавая по карточке с товаром или по кнопке на нем, товар можно добавлять в корзину или удалять. Карточки добавленных в корзину товаров внешне отличаются от остальных.

На странице отображается количество добавленных в корзину товаров. При попытке добавить в корзину больше 20 товаров, выводится всплывающее уведомление с текстом "Извините, все слоты заполнены".

3 Сортировка. Сортируются только те товары, которые в данный момент отображаются на странице:

- сортировка товаров по названию в возрастающем и убывающем порядке;
- сортировка товаров по году их выхода на рынок в возрастающем и убывающем

порядке;

4 Фильтры в указанном диапазоне от и до:

- фильтры по количеству;
- фильтры по году выпуска на рынок.

Для фильтрации в указанном диапазоне используется range slider с двумя ползунками. При перемещении ползунков отображается их текущее значение, разный цвет слайдера до и после ползунка.

5 Фильтры по значению:

- фильтры по производителю;
- фильтры по цвету;
- фильтры по размеру;
- можно отобразить только популярные товары;
- можно отфильтровать товары по нескольким фильтрам одного типа.

Для нескольких фильтров одного типа отображаются товары, которые соответствуют хотя бы одному выбранному фильтру.

6 Можно отфильтровать товары по нескольким фильтрам разного типа. Для нескольких фильтров разного типа отображаются только те товары, которые соответствуют всем выбранным фильтрам.

7 Если товаров, соответствующих всем выбранным фильтрам, нет, на странице выводится уведомление «Извините, совпадений не обнаружено».

8 Выбранные фильтры выделяются стилем.

9 Сброс фильтров. Есть кнопка reset для сброса фильтров. Кнопка reset сбрасывает только фильтры, не влияя на порядок сортировки или товары, добавленные в избранное. После использования кнопки reset фильтры остаются работоспособными. При сбросе фильтров кнопкой reset ползунки range slider сдвигаются к краям, значения ползунков возвращаются к первоначальным, range slider закрашивается одним цветом.

10 Сохранение настроек. Выбранные пользователем фильтры, порядок сортировки, добавленные в избранное товары сохраняются при перезагрузке страницы. Есть кнопка сброса настроек.

11 Поиск:

- при открытии приложения курсор находится в поле поиска;
- автозаполнение поля поиска отключено (нет выпадающего списка с предыдущими запросами);
- есть placeholder;
- в поле поиска есть крестик, позволяющий очистить поле поиска;
- если нет совпадения последовательности букв в поисковом запросе с названием товара, выводится уведомление «Извините, совпадений не обнаружено»;
- при вводе поискового запроса на странице остаются только те товары, в которых есть указанные в поиске буквы в указанном порядке. При этом не обязательно, чтобы буквы были в начале слова. Регистр символов при поиске не учитывается. Поиск ведётся только среди товаров, которые в данный момент отображаются на странице;
- если очистить поле поиска, на странице отображаются товары, соответствующие всем выбранным фильтрам и настройкам сортировки.

Приложение Д (справочное)

Пример описания структуры страницы веб-приложения

Структура страницы «Портфолио».

Страница «Портфолио» содержит в себе список всех работ, которые сделала компания «ГИК». Работы представлены в виде блоков с картинкой при наведении на которые появляется краткое описание во всплывающем окне, а при нажатии осуществляется переход на внутреннюю страницу с работой.

Вверху страницы расположен заголовок страницы и «хлебные крошки» для быстрого перехода. Рядом с заголовком расположены ссылки на внутренние страницы: сайт компании, интернет-каталог, лендинг, сложные проекты, мобильные (адаптивные) сайты и лого.

Дизайн-макет страницы «Портфолио» представлен на рисунке Д.1.

Дизайн-макет страницы «Портфолио»

Рисунок Д.1 – Страница «Портфолио»

Структура внутренних страниц «Портфолио»:

- сайт компании. Большие и объемные сайты для компаний;
- интернет-каталог. Сайты, содержащие продукцию, позволяющие просматривать товар и заказывать его;
- лендинг. Одностраничные сайты, промо-сайты, которые настраивают посетителя на совершение определенного действия;
- сложные проекты. Сложные сервисы, направленные на достижение и облегчение разных сторон бизнеса;
- мобильные (адаптивные) сайты. Сайты, разработанные не только под компьютер, но и под мобильные устройства (Iphone, Android и т. д.);
- лого. Логотипы и фирменный стиль проектов.

Приложение Е (справочное)

Пример описания тестового примера

Для демонстрации работы веб-приложения оптово-розничной фирмы по торговле кондитерскими изделиями будет рассмотрен процесс заказа нескольких позиций продукции и кондитерских изделий, а также оплата заказа с помощью банковской карты.

Чтобы запустить веб-приложение необходимо в адресной строке ввести [http://????.???.](http://????.???)

Откроется главная страница веб-приложения, где расположены элементы главной навигации, фильтры, блок с отзывами о работе и представлены популярные товары (продукция). Результат представлен на рисунке Е.1.

Скрин главной страницы

Рисунок Е.1 – Главная страница веб-приложения

Для просмотра состава продукции необходимо кликнуть по фотографии или названию интересующей продукции. Например, кликнуть по «Вафли Артек». На рисунке Е.2 изображена карточка данного товара. На карточке товара представлена полная информация о товаре, а именно: название товара, цена, фотография товара, состав, пищевая ценность, энергетическая ценность, условия хранения и срок годности.

Скрин страницы «Карточка товара»

Рисунок Е.2 – Карточка товара

Для добавления продукции в корзину необходимо нажать кнопку «Заказать». Данный товар попадет в корзину, которая визуально расположена внизу-справа (рисунок Е.3).

Скрин страницы «Корзина покупателя»

Рисунок Е.3 – Корзина веб-приложения

Дополнительно поместим в корзину:

- «Тульский Пряник с фруктовой начинкой 140 г» в количестве 3 шт.;
- «Коломенское Вафли сливочные» в количестве 5 шт.;
- «Вафли Артек-супер 240 г» в количестве 2 шт.

После добавления необходимо оформить данный заказ. Для этого на странице «Оформление заказа» заполняются данные заказчика, дополнительная информация и выбирается способ оплаты. На рисунке Е.4 представлена заполненная страница «Оформление 63 заказа».

Скрин страницы «Оформление заказа»

Рисунок Е.4 – Страница оформления заказа веб-приложения

После заполнения всех полей необходимо нажать кнопку «Проверить». Все данные сохраняются в базе данных, а именно: сохраняются данные о заказчике, данные о заказе,

данные о корзине. Если все сохранено успешно, то генерируется номер заказа и пользователь перенаправляется на страницу «Проверка заказа», представленная на рисунке Е.5.

Скрин страницы «Проверка заказа»

Рисунок Е.5 – Страница проверки заказа веб-приложения

Оплаченный заказ необходимо оплатить с помощью банковской карты через платежную систему Яндекс.Деньги (рисунок Е.6).

Скрин страницы «Оплата»

Рисунок Е.6 – Страница оплаты заказа веб-приложения

В системе управления заказами можно просмотреть данный заказ, отредактировать или изменить статус. Например, поставить статус «Заказ оплачен. Передан на сборку».

Данная система представлена на рисунке Е.7. Таким образом, был оформлен заказ и продемонстрированы основные функции веб-сайта.

Скрин страницы системы управления заказами

Рисунок Е.7 – Страница управления заказами веб-приложения

**Приложение Ж
(справочное)**

Пример оформления титульного листа

**МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Программное обеспечение информационных технологий»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по дисциплине «Средства взаимодействия человека с вычислительными
системами»

Тема курсовой работы

(тема курсовой работы)

КР ХХ.ХХ.ХХ.ХХХХХХХ.ХХ ПЗ

Разработал студент

(подпись)

ФИО студента

Группа

Руководитель

(подпись)

ФИО руководителя

Работа допущена к защите

« ___ » _____ 20XX г.

Работа защищена с оценкой _____

« ___ » _____ 20XX г.

Комиссия

Могилев 20XX