

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

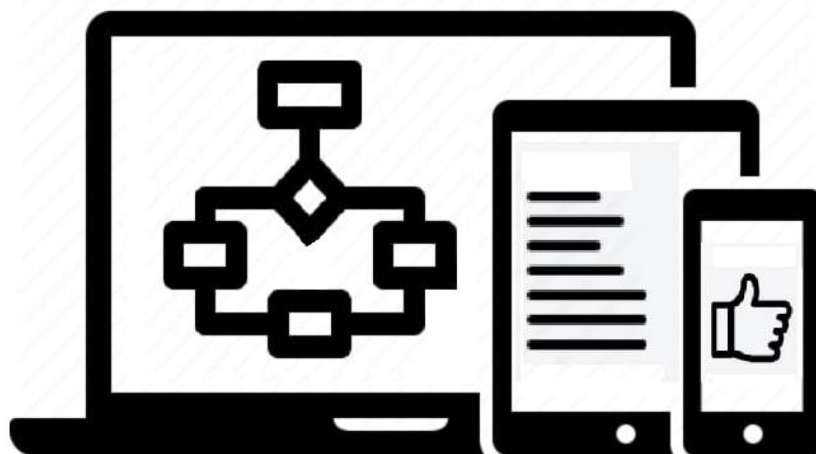
СРЕДСТВА ВЗАИМОДЕЙСТВИЯ ЧЕЛОВЕКА С ВЫЧИСЛИТЕЛЬНЫМИ СИСТЕМАМИ

*Методические рекомендации к курсовому проектированию
для студентов направления подготовки*

19.03.01 Информатика и вычислительная техника

09.03.04 Программная инженерия

очной формы обучения



Могилев 2024

УДК 621.01
ББК 36.4

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» 04 марта 2023 г., протокол № 9

Составитель: канд. тех. наук Ю. В. Вайнилович

Рецензент

Методические указания содержат требования к курсовому проектированию по дисциплине «Средства взаимодействия человека с вычислительными системами».

Учебно-методическое издание

СРЕДСТВА ВЗАИМОДЕЙСТВИЯ ЧЕЛОВЕКА С ВЫЧИСЛИТЕЛЬНЫМИ СИСТЕМАМИ

Ответственный за выпуск
Технический редактор
Компьютерная верстка

В. В. Кутузов
Т.А. Рыжикова

Подписано в печать
Таймс.

. Формат 60х84/16. Бумага офсетная. Гарнитура

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, 2024

Содержание

Введение	4
1 Стандартные этапы разработки web-приложения	5
2 Порядок выполнения и защиты курсовой работы	8
3 Используемые инструменты и технологии	11
4 Минимальные требования к курсовой работе.....	12
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 используемых источников	23
Приложение А (справочное) Примерная тематика курсовой работы.....	24
Приложение Б (справочное) Соглашение о коммитах	25
Приложение В (справочное) Пример аналитического отчета	26
Приложение Г (справочное) Примеры текстовых прототипов web- приложения.....	30
Приложение Д (справочное) Пример описания структуры страницы web- приложения.....	34
Приложение Е (справочное) Пример описания тестового примера.....	35
Приложение Ж (справочное) Пример оформления титульного листа	37

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 должен продемонстрировать следующие умения:

- выполнять предпроектное исследование предметной области, в рамках которой ведется разработка web-приложения;
- осуществлять прототипирование и проектирование архитектуры web-приложения;
- осуществлять разработку дизайна web-приложения;
- разрабатывать web-приложение;
- 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 web-приложения

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

Процесс создания web-приложения, в общем случае, включает восемь основных этапов, представленных на рисунке 1.1. 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.

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 приступить 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 странице. 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 web-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 окажется разрабатывать долго, в итоге затянутся сроки реализации web-приложения.

Далее составляется визуальный прототип, который показывает внешний вид будущего web-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 web-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 использоваться.

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 web-дизайнеры и арт-директор.

Сначала прорабатывается дизайн-концепция будущего web-приложения. Ее задача – передать визуальную идею будущего web-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 web-приложения, детально продумывают внешний вид элементов и эффекты при взаимодействии пользователя с интерфейсом. Для каждого экрана разрабатывают адаптивные макеты, чтобы web-приложение хорошо выглядело и работало на разных устройствах с разными разрешениями экранов. Web-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 frontend-разработчиков, чтобы заранее продумать техническую реализацию.

Далее наступает этап фронтенда. Результат frontend-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 web-приложения является бекенд-разработка. Web-приложение без бэкенда — это не более, чем красивая оболочка. Чтобы web-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.

Бекенд-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 админпанели;

- 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 web-приложение готово, происходит его наполнение информацией. Этим занимается контент-менеджер. Он пишет статьи, добавляет новости, услуги, описывает товары.

Далее идет этап продвижения web-приложения. Этим может заниматься рекламщик, деректолог, SEOшник, таргетолог.



Рисунок 1.1 – Стандартные этапы разработки web-приложения

2 Порядок выполнения и защиты курсовой работы

В соответствии с основными этапами разработки web-приложений планируется работа над проектом (таблица 2.1). В процессе выполнения курсовой работы последовательно проходятся первые шесть этапов, уделяя основное внимание этапам дизайна, фронтенд и бекенд разработки.

Таблица 2.1 – Этапы выполнения курсовой работы

Планируемая работа	Срок, неделя семестра
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 Разработка брифа	
7 Сбор и подготовка контента	3
8 Проектирование структуры web-приложения	
Этап проектирования дизайна web-страниц	
9 Разработка дизайн-макетов	4
10 Верстка макетов	5-6
Этап разработки web-приложения	
11 Проектирование структуры базы данных	7
12 Программирование: бекенд	8-11
13 Программирование: фронтенд	12-16
Заключительный этап	
14 Оформление пояснительной записки	16-17
15 Подготовка материалов к защите	
16 Защита курсовой работы	

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 web-страниц в файле README.md ветки main следует разместить ссылку на fig. файл с прототипами, дизайн-концепцией, дизайн-макетами;

На этапе разработки web-приложения требуется делать не менее трех коммитов в неделю в репозиторий проекта.

После окончания работы над web-приложением добавить в репозиторий GitHub backup базы данных. База данных должна содержать суммарно не менее 200 записей для качественного тестирования web-приложения.

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 секунд, демонстрирующий основные приемы работы с web-приложением.

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

- пояснительная записка в печатном виде в папке-сборнике 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;
- видеоролик.

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 минут). Затем демонстрирует web-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);
- web-приложение содержит функционал в соответствии с установленным или 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 п. 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 web-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 п. 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 web-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, **отсутствуют заимствования из методических указаний и элементы чужой работы**, имеется скопированный и не переработанный текст из интернета. Материал изложен логически связно, последовательно. Часть пояснений дана формально или отсутствует. Принятые решения при разработке программного продукта допустимы, но не обоснованы, не рациональны. Пояснительная записка оформлена аккуратно, но имеются грамматические ошибки, несущественные отклонения от требований п. 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.

Внимание! Обращаю внимание на то, что даже на оценку «удовлетворительно» в пояснительной записке не должно быть заимствований из методических указаний и элементов чужой работы. Следует помнить о том, что после заимствований необходимо указывать номер источника (из Списка использованных источников) в квадратных скобках. В противном случае пояснительная записка будет возвращена на доработку с пометкой «Плагат».

3 Используемые инструменты и технологии

Вайрфреймы, дизайн-макеты страниц web-приложения, мудборд, карта маршрутов пользователя разрабатываются в Figma или Pixso.

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, библиотека Redux (Redux Toolkit) или MobX, сборщик модулей Webpack.

Можно использовать библиотеки bootstrap, materialUI, chakraUI, css-фреймворки, html и css препроцессоры, любые react-пакеты.

База данных – PostgreSQL.

ORM (Object Relational Mapping): sequelize, prisma.

Базовые технологии бэкенд-разработки: кросс-функциональная платформа Node.js, фреймворк express.

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/>.

Для проверки оптимизации под мобильные устройства используется сервис https://search.google.com/test/mobile-friendly?utm_source=mft&utm_medium=redirect&utm_campaign=mft-redirect&hl=ru.

4 Минимальные требования к курсовой работе

Минимальные функциональные требования:

- наличие **не менее двух пользовательских ролей** с принципиально различными функциями приложения;
- наличие страниц авторизации и регистрации;
- наличие **не менее 7 страниц**, кроме страниц регистрации и авторизации. Наполнение страниц зависит от тематики курсовой работы;
- реализация **не менее 15 функций** приложения (фильтрация по критерию, сортировка по критерию, напоминка, построение графика, формирование отчета и т. д.), кроме функций авторизации и регистрации;
- формирование **не менее 2 отчетов**, которые можно скачать в формате .docs, .pdf или получить на e-mail, либо это могут быть google docs документы;
- использование **не менее 20** различных компонентов (карточки, слайдеры, снэкбары, таблицы, кнопки и т. д.). **Внимание!** Если у вас есть красная кнопка, зеленая кнопка и круглая кнопка, то это не значит, что у вас есть три различных компонента. Это значит, что у вас есть один компонент «кнопка» стилизованный по-разному в разных местах;
- вёрстка адаптирована для десктопа, планшета и мобильного устройства. Корректность отображения приложения и отсутствие горизонтальной полосы прокрутки проверяется при ширине страницы от 1920 (или 1440)px до 320px;
- наличие интерактивности 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.

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

- наличие **не менее 8** связанных таблиц базы данных, приведенных к третьей нормальной форме;

- способ взаимодействия web-приложения с сервером – RESTful API с базовой CRUD структурой.

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 должен составлять не менее 30-35 страниц формата А4 без учета приложений.

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 Бриф на разработку web-приложения
- 2 Проектирование дизайна web-страниц;
 - 2.1 Структура web-приложения;
 - 2.2 Текстовый прототип;
 - 2.3 Вайрфреймы web-страниц;
 - 2.4 Дизайн-макеты web-страниц;

- 3 Реализация web-приложения;
- 3.1 Выбор технологий и инструментов;
- 3.2 Разработка слоя доступа к данным;
- 3.3 Разработка серверной части;
- 3.4 Разработка клиентской части;
- 4 Руководство системного программиста;
- 5 Описание тестового примера.

Графическая часть курсовой работы состоит из не менее трех листов формата A1. В графическую часть выносятся прототипы пользовательского интерфейса, дизайн-макеты пользовательского интерфейса, карта маршрутов пользователя, любые другие схемы и рисунки на усмотрение студента. Если рисунок имеет мелкий шрифт и не читаем в пояснительной записке, его следует вынести в графическую часть.

На все листы графической части должна быть ссылка в пояснительной записке.

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 web-приложение.

Следует выяснять, есть ли похожие web-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 web-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 web-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. Обратить 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 web-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 источника [3].

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

6.4 Бриф на разработку web-приложения

Бриф (brief – резюме, инструкция, сводка) – это документ, заполняемый заказчиком на начальном этапе сотрудничества. В нём прописывают контекст, в котором существует компания, примерные задачи проекта, разбирают возможные технические нюансы. Иногда брифом называют опросник, который готовит потенциальный исполнитель, а иногда – уже заполненный заказчиком документ.

Не путайте бриф и техническое задание. Бриф и техническое задание – в корне разные документы. Основные отличия:

- разная цель. Бриф помогает познакомиться исполнителю с желаниями заказчика. ТЗ фиксирует чёткие задачи, которые должен выполнить исполнитель.

– на каком этапе составляется документ. Бриф составляется в самом начале обращения заказчика. С этого начинается обсуждение желаемого результата. ТЗ составляется прямо перед началом выполнения работ. Это конечный этап обсуждения требований.

– юридическая сила. Если ТЗ было правильно оформлено и прикреплено к договору, оно имеет юридическую силу. То есть, если исполнитель не выполнил какие-либо обязательства, заказчик вправе обратиться в суд. Бриф не может иметь юридической силы, так как в нём нет никакой конкретики по требованиям к исполнителю.

– детальность проработки. Чаще всего бриф состоит из 1-2 страниц. ТЗ – объёмный документ, он может состоять из нескольких десятков страниц.

– кто готовит. Обычно ТЗ готовит исполнитель, а заказчик согласовывает. Заказчик не всегда знает нюансы работы, особенно если это касается технических задач. Следовательно, не может создать правильное ТЗ. Бриф всегда заполняется заказчиком.

Единого шаблона для создания брифа не существует. Но есть несколько основных разделов, которые обязательно должны в нём присутствовать независимо от сферы деятельности заказчика и типа выполняемой услуги.

– информация о компании. Здесь обычно указывают те данные об организации, которые имеют значение для успешной реализации проекта и влияют на объем работы.

– информация о проекте. Здесь клиент перечисляет все свои запросы, желаемые результаты, указывает бюджет.

– целевая аудитория. Указывается конкретно, кто является потребителем готового продукта. Порой это помогает исполнителю увидеть какие-то дополнительные нюансы, которые клиент упустил.

– цели и задачи. Заказчик обязан прописать, чего он именно хочет добиться, какие результаты хочет видеть после завершения всех процедур.

– сроки выполнения работ.

– контактная информация. Обязательно стоит перечислить номера телефонов, указать социальные сети или электронную почту на случай, если вдруг у исполнителя возникнут дополнительные вопросы.

Часто такой документ содержит и технические характеристики желаемого продукта. Например, при написании текстовых материалов заказчик может задать стиль оформления документа, его формат (Excel, Word, PDF и так далее), стиль написания и так далее. Или при разработке сайта можно перечислить основные страницы и разделы, из которых он будет состоять.

Так как заказчик для реализации учебного проекта отсутствует, то предлагается форма, представленная в таблице 6.1.

Пример брифа на разработку web-приложения

Таблица 6.1 – Пример брифа на разработку мобильного приложения автосервиса

Тема проекта	Мобильное приложение автосервиса
Категория системы	Мобильное приложение (предпочтительно облачное решение)
Настоящая ситуация	
Автоматизируемая деятельность <i>(должно быть согласовано с п. 1.1 пояснительной записки)</i>	1. Ведение журнала состояния и обслуживания автомобиля
Заинтересованные стороны <i>(должно быть согласовано с п. 1.1 пояснительной записки)</i>	1. Автовладельцы 2. Владельцы автосервисов 3. Автомастера сервисов
Текущее решение <i>(должно быть согласовано с п. 1.1 пояснительной записки)</i>	Для автомобилиста и про-пользователя: Различные самостоятельные попытки вести данные по комплектующим, пробегу и срокам ТО для нескольких автомобилей: на бумажных носителях, в Excel-е и т.п.
Решаемые проблемы <i>(описать проблемы, а не просто факты)</i> <i>(должно быть согласовано с п. 1.1 пояснительной записки)</i>	Для автомобилиста: – потери до 15 000 руб. в год от некачественных услуг на автосервисах – затраты до 40 часов в год при поиске качественного сервиса. – Для автосервиса: – потери до 10% дохода от продолжительной диагностики при отсутствии истории обслуживания автомобиля (при новом клиенте)
Целевая ситуация	
Цель для заказчика <i>(что получит/заработает бизнес после реализации проекта?)</i> <i>(должно быть согласовано с п. 1.1 пояснительной записки)</i>	– заработать 50 000 000,00 рублей за 3 года; – привлечь порядка 10 млн. автомобилистов в РБ (примерно 25%).
Назначение <i>(цель для пользователя)</i> <i>(Что даст пользователю хорошего?)</i> <i>(должно быть согласовано с п. 1.1 пояснительной записки)</i>	Для автомобилиста: 1. Снизить количество поездок на сервис и в магазин за комплектующими в 1,5 раза. 2. Получить скидку от сервиса за использование приложения в 5% (Возможность отправить в сервис ссылку на журнал ТО конкретного авто, где мастер мог бы записать перечень проведенных работ.). 3. Снизить риск покупки неподходящих комплектующих до 0,01%. 4. Сократить время поиска комплектующих в 10 раз. Для автосервиса: 1. Сокращение времени на диагностику неполадок, на 25 %.

	2. Сокращение затрат на привлечение нового клиента на 50%.
Концепция решения	
Количество видов пользователей (ролей) (не менее двух ролей, см. п. 4 Минимальные требования)	<ol style="list-style-type: none"> 1. Автомобилисты - владельцы автомобилей 2. Платные (про-) пользователи 3. Автомастера автосервисов 4. Администратор системы
Ключевые свойства и возможности (не менее 15 функций, которые будут реализованы, см. п. 4 Минимальные требования)	<p>Для автомобилиста:</p> <ol style="list-style-type: none"> 1. Возможность ведения автомобилистом нескольких автомобилей. 2. Напоминания графика ТО, замены жидкостей и т.п.. 3. Возможность видеть историю сервисов, ТО, замен комплектующих. 4. Возможность получить отчет по своим автомобилям. 5. Веб-страница для автомашинистов. <p>Для платных (про-) пользователей:</p> <p>...</p> <p>Для автомашинистов автосервисов:</p> <p>...</p> <p>Для администратора системы:</p> <p>...</p>
Состав надсистемы	<p>В рамках проекта:</p> <ul style="list-style-type: none"> – мобильное приложение. <p>За рамками проекта:</p> <ul style="list-style-type: none"> – бэкофисное веб-приложение.
Смежные системы (интеграции) (если есть или предполагается сделать в будущем)	<ul style="list-style-type: none"> – система, предоставляющая по VIN номеру данные по комплектующим автомобиля и автомобилю (марка, г.в.); – система, предоставляющая по VIN номеру данные о пробеге, сроках ТО, возможно проведенных сервисах.
Количество страниц (см. п. 4 Минимальные требования)	<p>Мобильное приложение:</p> <ol style="list-style-type: none"> 1. Авторизация/регистрация 2. Домашняя страница 3. Страница добавления авто 4. Карточка автомобиля 5. Отчеты 6. Биллинг для владельца 7. Оплаты пользователя
Количество отчетов (см. п. 4 Минимальные требования)	<p>Для автомобилистов - владельцев авто:</p> <ol style="list-style-type: none"> 1. текущее состояние: <ol style="list-style-type: none"> a. наименование узла + артикул; b. срок обслуживания (ближайшая дата). 2. журнал (история) обслуживания (расходы); 3. ближайшее ТО: состав работ. <p>Платные (про-) пользователи</p> <ol style="list-style-type: none"> 1. текущая подписка (окончание периода оплаты);

	2. история оплат (списания); 3. те же формы, что у автомобилистов. Администратор 1. история оплат 2. регистрации в системе
Системы-аналоги (должно быть согласовано с п. 1.2 пояснительной записки)	exist.ru

Подробное описание реализуемых функций и сопоставление web-страниц реализуемым функциям представить в виде таблицы 6.2.

Обратите внимание! Для примера, желтым цветом в таблице 6.1, в разделе «Ключевые свойства и возможности» выделена функция, а в разделе «Количество страниц» выделена страница. В таблице 6.2 показано, что данная функция будет реализована на данной странице, а также приведено подробное описание содержания функции.

Такие возможности разрабатываемого приложения как создание, редактирование, удаление и просмотр списка какой-либо сущности (например, заказа, услуг, товаров и т. д.) ни в брифе на разработку ни в описании в таблице 6.2 не приводятся, поскольку и так понятно, что все сущности предметной области должны как-то попадать в систему и находиться в ней в актуальном состоянии.

Пример

Таблица 6.2 – Описание реализуемых функций

Функция	Описание функции	Web-страница
Возможность видеть историю сервисов, ТО, замен комплектующих	Содержит регистрационную информацию автомобиля, историю обслуживания автомобиля: дата, тип обслуживания, выполненные работы, замена комплектующих, мастер, выполнявший обслуживание.	Карточка автомобиля
Управление списком комплектующих	Функция предоставляет возможность добавлять новые комплектующие, изменять количество комплектующих в наличии.	Отчеты
...

6.5 Структура web-приложения

Структура web-приложения — 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 структура 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 структурой – 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 6.1).

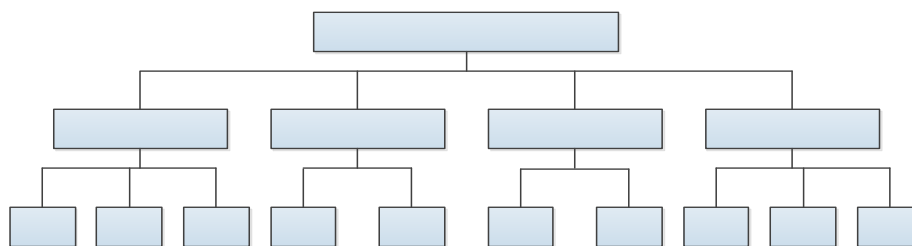


Рисунок 6.1 – Плоская структура web-приложения

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 6.2 обозначены пунктирными линиями).

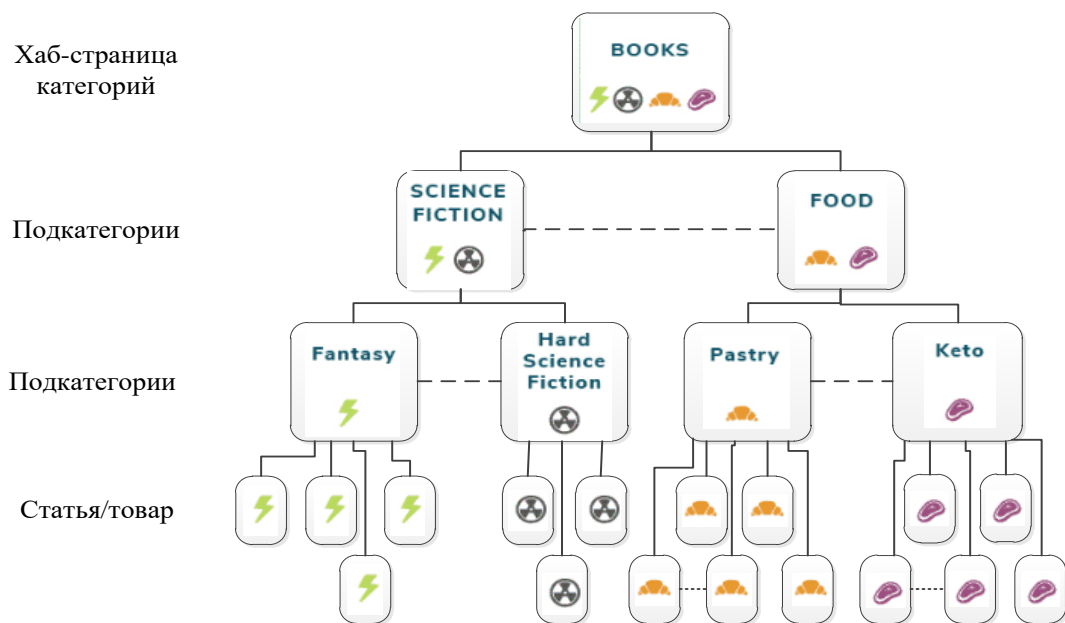


Рисунок 6.2 – SILO-структура контента web-приложения

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

Каждая страница должна иметь ссылочный номер и метку (рисунок 6.3).

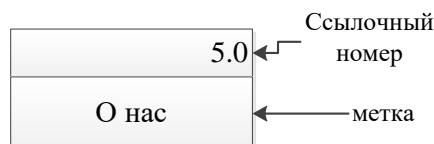


Рисунок 6.3 – Изображение страницы

На верхнем уровне следует разместить домашнюю страницу, основные страницы в первом ряду, второстепенные страницы в следующем и т.д.

Пример

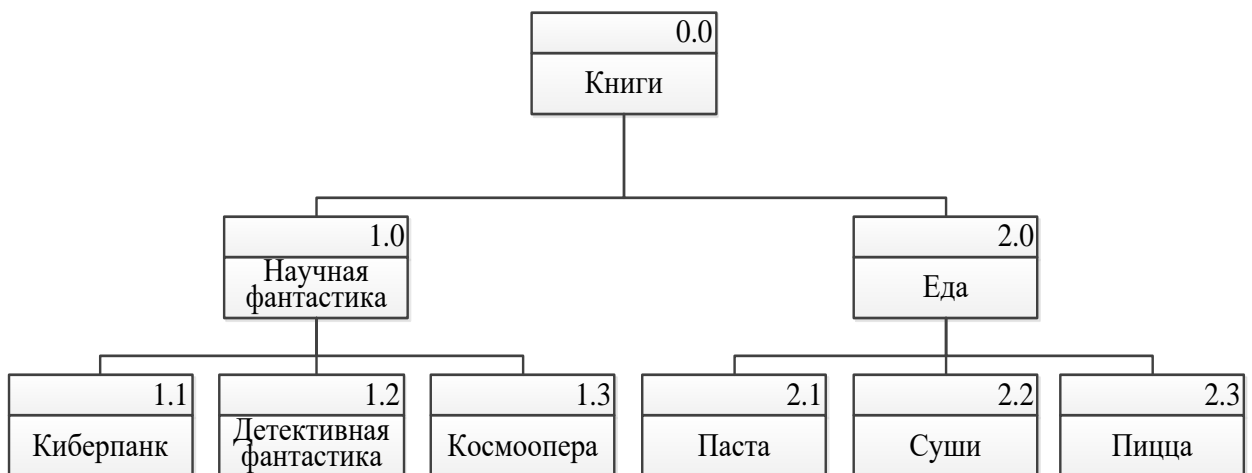


Рисунок 6.4 – Пример структуры web-приложения

Перед выполнением данного раздела рекомендуется ознакомиться со статьями из источников [4-7].

Внимание! Структура web-приложения должна быть согласована с брифом на разработку в части «Количество экранов».

6.6 Текстовый прототип

Прототип сайта или каркас – 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.

Внимание! Тестовый прототип должен содержать ВСЕ страницы, составляющие структуру web-приложения. Обязательно указывайте в скобках ссылочный номер.

6.7 Вайрфреймы web-страниц

Вайрфреймы создаются с целью заложить структуру уникальных страниц и не упустить важные контентные блоки [9]. Он помогает схематически визуализировать основные элементы и функции web-приложения. **В вайрфрейме наглядно изображаются структура, элементы интерфейса, расположение иллюстраций и кнопок, меню и иконок, но не оформление**, потому что цель вайрфрейма — показать, КАК и ГДЕ будут располагаться блоки с информацией на экранах и КАК эти экраны будут сменять друг друга. То есть вайрфрейм показывает логику приложения.

Вайрфреймы — это изображение результата, который должен создать дизайнер на этапе разработки дизайн-макета.

Обозначение элементов на вайрфрейме может варьироваться в зависимости от того, как вы создаете ваши вайрфреймы и какой инструмент используете для этого. Однако, в целом, существуют общие соглашения по обозначению элементов интерфейса на вайрфреймах:

- изображения обозначаются простыми прямоугольниками с крестиком, пиктограммой фотоаппарата внутри или с надписью «Image». Также, если изображение является кликабельным, внутри прямоугольника может быть указана стрелка.

- кнопки обозначаются прямоугольниками с надписью на них, например «Submit», «Save», «Cancel», и т.д. Часто используются также стандартные пиктограммы, например для кнопки «Search» используется изображение лупы.

- текстовые поля обозначаются прямоугольниками с надписью «Input» или «Text». Если поле является обязательным для заполнения, то иногда внутри прямоугольника может быть указана звездочка.

– текст обозначается простыми прямоугольниками с надписью на них. Если текст является заголовком, то его обычно выделяют большим шрифтом и жирным начертанием.

Пример вайрфрейма представлен на рисунке 6.5.

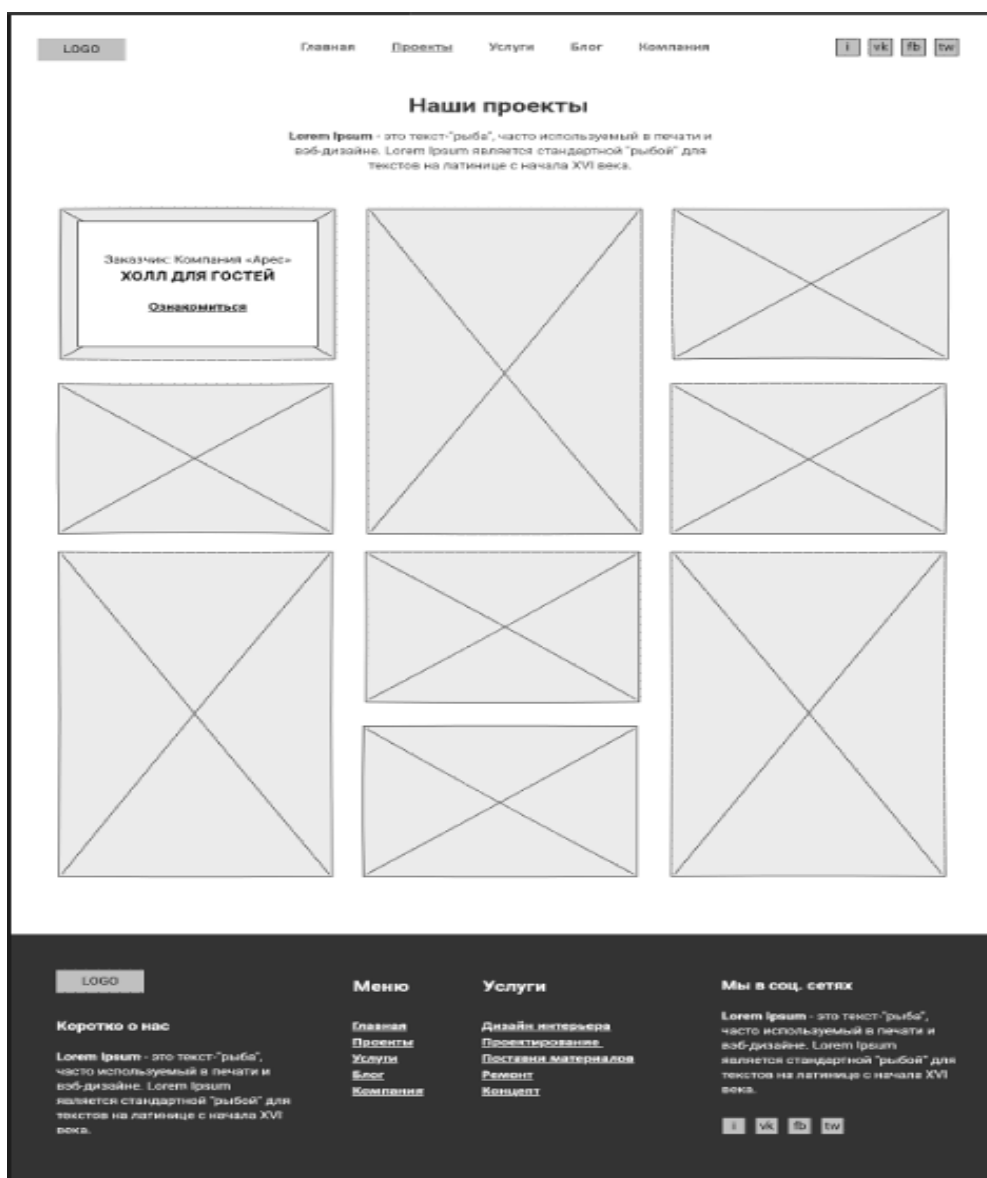


Рисунок 6.5 – Пример вайрфрейма страницы

В данном разделе курсовой работы необходимо разработать вайрфреймы ВСЕХ страниц web-приложения, представленных в архитектуре web-приложения и имеющих текстовый прототип. Должны быть учтены все требования, описанные в текстовом прототипе.

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

Внимание! При описании вайрфреймов обязательно указывать ссылочный номер страницы в структуре web-приложения (рисунок 6.4).

После разработки вайрфреймов страниц сайта строится карта маршрутов пользователя (customer journey map). Customer journey map (CJM) – таблица или инфографика, которая показывает все точки контакта пользователя с web-приложением с момента первого соприкосновения.

Главная задача customer journey map – оптимизация взаимодействия с пользователем. Визуальный путь пользователя, подкрепленный данными, помогает определить точки, где необходима проработка опыта или сервиса.

Пример карты маршрута пользователя



Рисунок 6.6 – Карта маршрутов пользователя

6.8 Дизайн-макеты web-страниц

На данном этапе необходимо разработать дизайн-концепцию web-приложения и воплотить ее в виде дизайн-макетов web-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 web-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 источника [8].

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 скроле, 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 web-приложения приводят дизайн-макеты для трех разрешений и словесное описание происходящих на них процессах и явлениях. Пример описания структуры web-страницы приведен в приложении Д.

6.9 Выбор технологий и инструментов

Данный подраздел содержит описание используемых программных средства для разработки дизайн-концепции и дизайн-макетов страниц web-приложения, описание языков программирования, фреймворков, плагинов, CMS, ORM, библиотек, пакетов, баз данных используемых для программной реализации web-приложения с обоснованием выбора. Четко разграничить, какие технологии использовались для разработки клиентской части приложения, серверной части приложения и слоя доступа к данным.

6.10 Разработка слоя доступа к данным

В данном подразделе проектируется и описывается база данных. Структура базы данных проектируется в программе Enterprise Architect (можно скопировать из используемой СУБД в случае ее качественного отображения и читабельности). Для каждой таблицы необходимо дать словесное пояснение о содержащейся в ней информации (рисунок 6.3).

Пример

Таблица 6.3 – Структура таблицы user

Наименование атрибута	Хранимая информация	Тип данных	NOT NULL	Тип
id	идентификатор пользователя	uuid	true	primary_key
name	имя пользователя	text	true	
password	пароль	text	true	
email	email	text	true	unique_constraint
role	роль пользователя (администратор, пользователь)	character varying	true	foreign_key

6.11 Разработка серверной части

В данном подразделе необходимо описать структуру и логику работы серверной части проекта. Описание следует сопровождать UML-диаграммой или любой другой диаграммой, наглядно демонстрирующей структуру серверной части приложения (например, смотрите рисунок 6.7 – Фрагмент структуры клиентской части проекта).

Описать API для интеграции со слоем доступа к данным приложения, API для интеграции со сторонними сервисами и структуру JSON-файлов, в случае их использования.

Пример

```
GET /imports/$import_id/citizens
```

Обработчик возвращает всех жителей с указанным import_id. Если указанная выгрузка не существует, возвращается HTTP-ответ 404: Not Found.

6.12 Разработка клиентской части

В данном подразделе необходимо описать структуру и логику работы клиентской части приложения: указать **ВСЕ** разработанные компоненты, описать логику работы **КАЖДОЙ** страницы web-приложения с описанием интерактивности и указанием функций, в которых реализовано то или иное

событие (смотри пример). Описание необходимо сопроводить UML-диаграммой или любой другой диаграммой, наглядно демонстрирующей структуру клиентской части приложения.

На рисунке 6.7 представлена UML-диаграмма компонентов клиентской части web-приложения.

Внимание! Скрин структуры приложения из среды разработки вставлять в записку запрещается.

Далее необходимо описать все, использованные для разработки клиентской части, компоненты (см. п. 4 Минимальные требования к курсовой работе).

Пример

Структура клиентской части приложения представлена на рисунке 6.7.

Внутри проекта идёт разделение на следующие пакеты:

- `api` – пакет, который содержит в себе бизнес-логику для осуществления запросов в backend;
- `components` – компоненты написанные на JSX, которые будут использоваться на страницах;
- `constants` – константы, которые хранят в себе конфигурацию текущего окружения;
- `pages/auth` содержит ...;
- `pages/conferences` содержит ...;
- ... и т. д.

В соответствии с брифом на разработку web-приложения разработано XXX web-страниц.

Логика работы страницы «Авторизация» (рисунок XXX (*указывается ссылка на соответствующий дизайн-макет*)):

- пользователь вводит свои данные в соответствующие поля форм;
- нажимает на кнопку «Зарегистрироваться»;
- из клиентского кода веб-приложения уходит HTTP-запрос в backend (**функция `pages/auth/click()`**);
- backend возвращает JWT-токен, который далее используется для аутентификации запросов получения контента на остальных страницах;
- JWT-токен сохраняется в локальное хранилище браузера (**функция, в которой реализован данный функционал**);
- в случае возникновения ошибки регистрации – логин или почта пользователя уже существуют в системе – пользователь увидит страницу регистрации с уведомлением об ошибке, которая представлена на рисунке XXX (*указывается ссылка на соответствующий дизайн-макет*);

– в случае успешной регистрации, пользователь переадресуется на страницу со списком конференций (указывается ссылка на соответствующий дизайн-макет).

... и т. д.

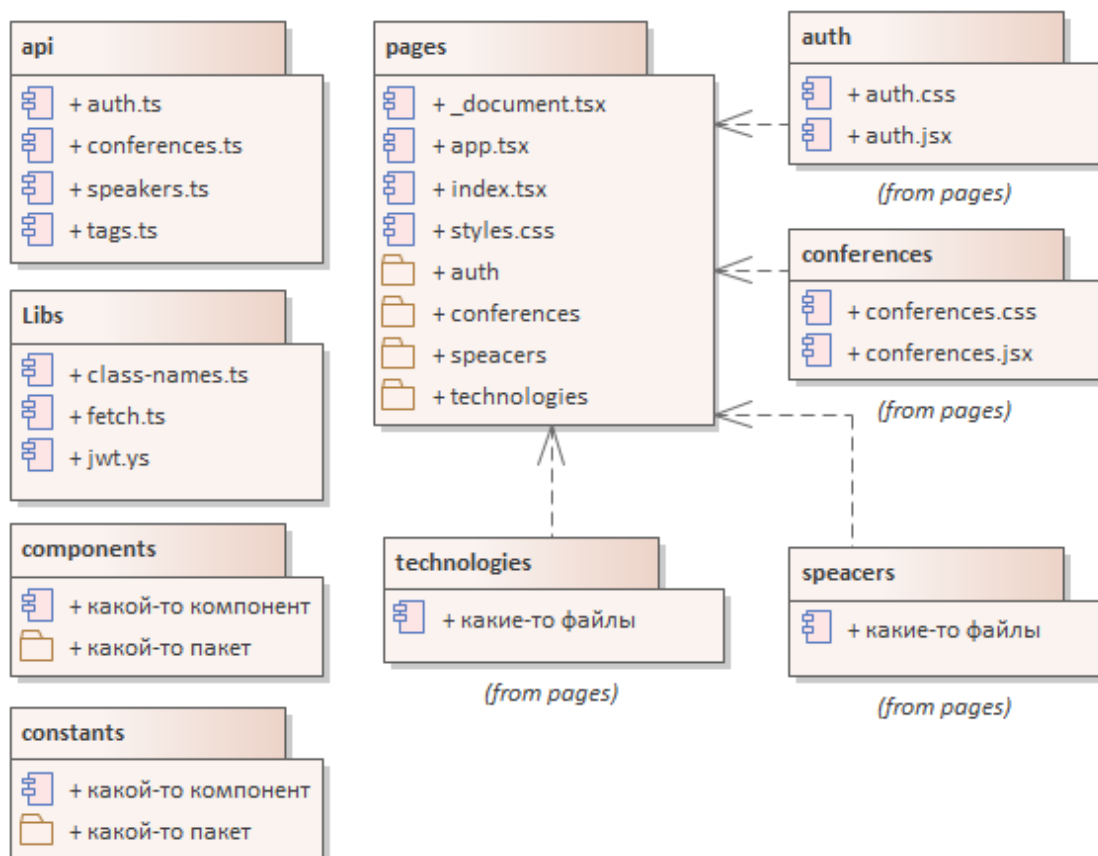


Рисунок 6.7 – Фрагмент структура клиентской части проекта

Для разработки клиентской части приложения использовался фреймворк React material UI. Используемые компоненты представлены в таблице 6.4.

Таблица 6.4 – Компоненты Material UI

Компонент	Ссылка на страницу дизайн-макета	Действие пользователя	Реакция	Интерактивность
Card	Рисунок XX	щелчок мышью по карточке	Открывается модальное окно с детальным описанием товара	Изменение формы курсора, подсветка карточки при наведении мыши (<i>ссылка на мудборд, если таковой разрабатывался</i>)
Modal	Рисунок XX	щелчок мышью в любом месте карточки	Карточка закрывается	
Pagination	Рисунок XX	щелчок мышью по кнопке с номером страницы	Переход на соответствующую страницу	Карточки отображаются по 20 штук на странице. При наведении мыши на кнопку пагинации кнопка изменяет цвет и увеличивается в размере.
...

6.13 Руководство системного программиста

Описывается порядок установки и запуска программного продукта для выполнения тестового примера.

Сначала следует описать все необходимое ПО для развертывания web-приложения.

Затем необходимо прописать четкую последовательность команд установки и запуска разработанного приложения.

Пример

... на компьютере должна быть установлена платформа node.js 21.1.0. Официальные установочные пакеты для всех основных платформ можно скачать с сайта <https://nodejs.org>

и т. д.

Алгоритм установки программного обеспечения на локальный компьютер:

- распаковать архив с программным обеспечением;
- в среде разработки Visual Studio Code перейти в папку server;
- установить зависимости командой

```
npm install
```

- выполнить импорт дампа базы данных *myStore.sql* в PgAdmin 4;
- и т. д.

6.14 Описание тестового примера

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

6.15 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 web-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 7.1), а последующие листы – с основной надписью формы 2а (15 мм, ГОСТ 2.104 – 2006, рисунок 7.2).

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

Графическая часть оформляется на листах формата А1. с основной подписью формы 1 (ГОСТ 2.104 – 2006, рисунок 7.3)

					КР XX.XX.XX.XXXXXXXX.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 7.1 – Основная надпись для заглавного листа

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

Рисунок 7.2 – Основная надпись для последующих листов

					КР XX.XX.XX.XXXXXXXX.XX ГЧ						
Изм.	Лист	№ докум	Подп.	Дата							
Разраб	ФИО студента				Название листа				Лит.	Масса	Масштаб
Проверил	ФИО препод.										
Т. контр.					Курсовая работа				Лист	номер	Листов
Н. бюро											
Н. конт.									БРУ, гр. номер группы		
Утв.											

Рисунок 7.3 – Основная надпись для графической части

Расшифровка шифра в основной надписи и титульном листе приведена на рисунке 7.4.

КР XX.XX.XX.XXXXXXXXXX.XX ПЗ

шифр
специальности

номер зачетной
книжки

номер по
списку

Рисунок 7.4 – Расшифровка шифра

Список используемых источников

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

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

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>

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>

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>

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

7 Как создать карту сайта: простое руководство. [Электронный ресурс]. Режим доступа: <https://vnikitinsky.medium.com/как-создать-карту-сайта-простое-руководство-5b5c483dad5c>

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

9 Вайрфреймы, прототипы и мокапы – в чем разница? [Электронный ресурс]. Режим доступа: <https://awdee.ru/wireframes-prototypes-and-mockups/>

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

Приложение А
(справочное)
Примерная тематика курсовой работы

Табу на разработку интернет-магазинов, чат-ботов, социальных сетей, мессенджеров!!!

Примеры тем для индивидуальной разработки

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

Пример темы для командной разработки

Разработка REST-сервера отдела кредитования банка.
Разработка клиента REST-сервера отдела кредитования банка.

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 (whitespace, 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

Web-приложения по продаже кондитерских изделий

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

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

В рассматриваемых web-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-бальная шкала, где 1 – функция отсутствует, а 5 – функция есть. Сравнение характеристик рассматриваемых web-приложений представлено в таблице В.1 приложения В.

Таблица В.1 – Сравнение характеристик рассматриваемых web-сайтов

Характеристики	Web-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 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 Фабрика 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 рис. 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

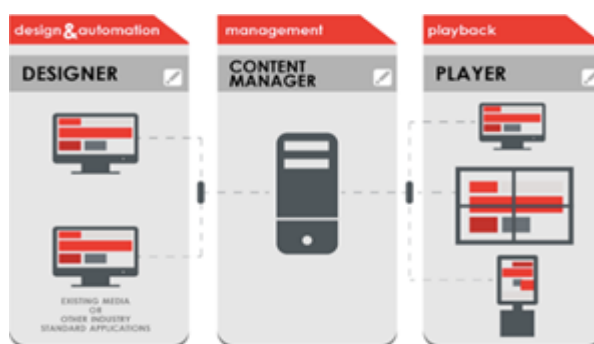


Рисунок В.1 - Составные компоненты системы Scala

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 рис. 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.

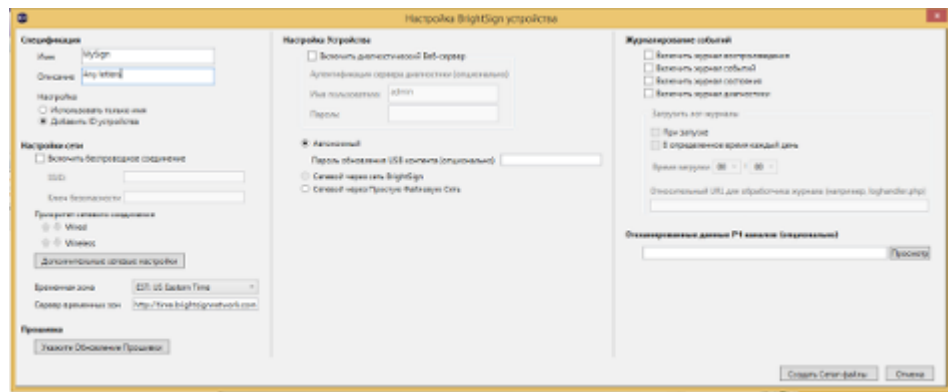


Рисунок В.2 - Настройка устройства BrightSign

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

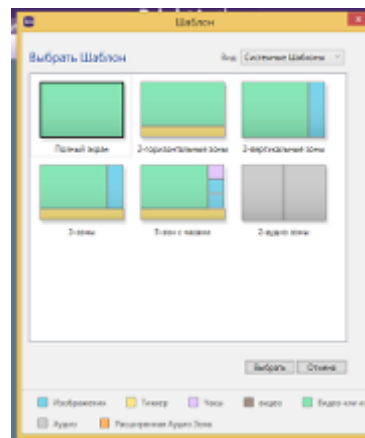


Рисунок В.3 - Шаблоны клиентского приложения BrightSign

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

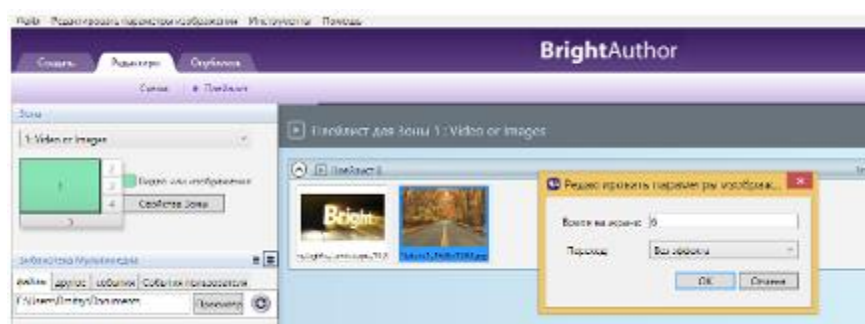


Рисунок В.4 - Процесс создания плейлиста

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 приложения для показа городской рекламы, но данное решение не распространено на российском рынке.

Разработанное клиентское приложение предоставляет 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 web-приложения

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

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

Главная страница (ссылочный номер XXX)

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 (ссылочный номер XXX)

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 (ссылочный номер XXX)

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 (ссылочный номер XXX)

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 (ссылочный номер XXX)

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

2 Компания / Преимущества (ссылочный номер XXX)

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

3 Компания / Наши партнеры (ссылочный номер XXX)

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

4 Компания / Контакты (ссылочный номер XXX)

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.

Тестовый прототип для школы танцев

1 экран (ссылочный номер XXX)

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 (ссылочный номер XXX)

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 (ссылочный номер XXX)

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 (ссылочный номер XXX)

Вот таким чудесным образом идет процесс нашей работы

5 экран (ссылочный номер XXX)

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.

Кликаая 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 хоть 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 web-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 Г1.

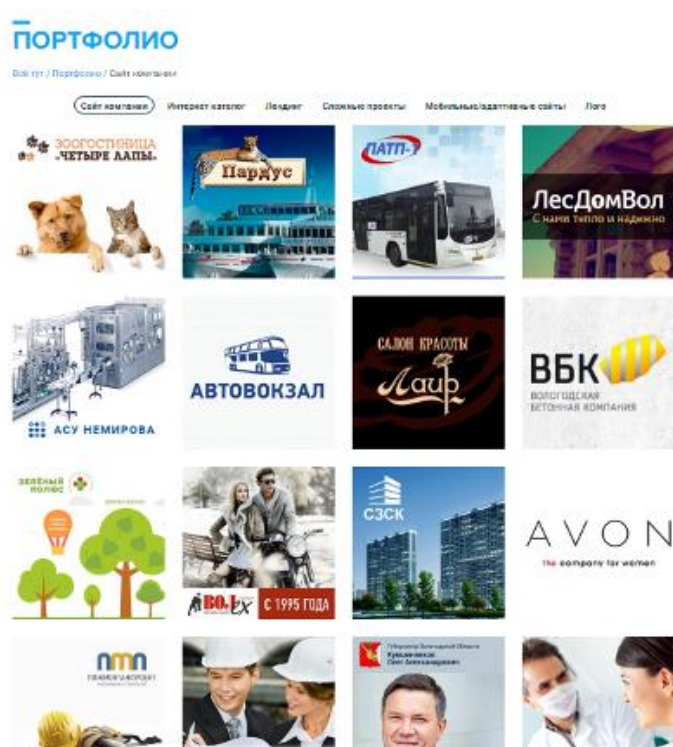


Рисунок Г1 – страница «Портфолио»

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 web-приложения оптово-розничной фирмы по торговле кондитерскими изделиями будет рассмотрен процесс заказа нескольких позиций продукции и кондитерских изделий и оплата заказа с помощью банковской карты.

Чтобы запустить web-приложение необходимо в адресной строке ввести `http://????.???.`

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

Скрин главной страницы

Рисунок Д.1 – Главная страница web-приложения

Для просмотра состава продукции необходимо кликнуть по фотографии или названию интересующей продукции. Например, кликнуть по «Вафли Артек». На рисунке Д.2 Приложения Д 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 Д.2 – Карточка товара

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

Скрин страницы «Корзина покупателя»

Рисунок Д.3 – Корзина web-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 добавление необходимо оформить данный заказ. Для этого на странице «Оформления заказа» заполняются данные заказчика, дополнительная информация и выбирается способ оплаты. На рисунке Д.4 Приложения Д представлена заполненная страница «Оформление б3 заказа».

Скрин страницы «Оформление заказа»

Рисунок Д.4 – Страница оформления заказа web-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 корзины. Если все сохранено успешно, то генерируется номер заказа и

пользователь перенаправляется на страницу «Проверка заказа», представленная на рисунке Д.5 Приложения Д.

Скрин страницы «Проверка заказа»

Рисунок Д.5 – Страница проверки заказа web-приложения

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

Скрин страницы «Оплата»

Рисунок Д.6 – Страница оплаты заказа web-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 Д.7 Приложения Д. Таким образом, был оформлен заказ и продемонстрированы основные функции web-сайта.

Скрин страницы системы управления заказами

Рисунок Д.7 – Страница управления заказами web-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.ХХХХХХХХ.ХХ ПЗ

Разработал студент _____ **ФИО студента**
(подпись)

Группа _____ **номер группы**

Руководитель _____ **ФИО руководителя**
(подпись)

Работа допущена к защите «__» _____ 20**ХХ** г.

Работа защищена с оценкой _____ «__» _____ 20**ХХ** г.

Комиссия _____

Могилев 20**ХХ**