

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» (УрФУ)

Институт радиоэлектроники и информационных технологий – РТФ

ОТЧЕТ

о проектной работе

по теме: WEB-Разработка Digital Portfolio 2024

по дисциплине: Проектный практикум 1A

Команда: Переводники

Лавров Никита РИ-130944

Гушшамов Кирилл РИ-130933

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc169903425)

[1. Целевая аудитория 5](#_Toc169903426)

[2. Определение проблемы 7](#_Toc169903427)

[3. Подходы к решению проблемы 9](#_Toc169903428)

[4. Анализ аналогов 12](#_Toc169903429)

[5. Календарный план проекта 15](#_Toc169903430)

[6. Сценарии использования 17](#_Toc169903431)

[7. Требования к продукту и к MVP 20](#_Toc169903432)

[8. Стек для разработки 24](#_Toc169903433)

[9. Прототипирование 32](#_Toc169903434)

[10. Проектирование и разработка системы 34](#_Toc169903435)

[ЗАКЛЮЧЕНИЕ 38](#_Toc169903436)

[СПИСОК ЛИТЕРАТУРЫ 39](#_Toc169903437)

ВВЕДЕНИЕ

Актуальность разработки веб-приложения для создания и обмена маршрутами по городу Екатеринбург обусловлена современными потребностями горожан и туристов города в удобных и интерактивных инструментах для навигации и знакомства с городом. В условиях растущей урбанизации и увеличения как числа туристов, так и населения города в общем, традиционные способы ориентирования и исследования городских пространств становятся менее эффективными. В то время как различные существующие карты и путеводители предоставляют базовую информацию, они часто не учитывают индивидуальные интересы и предпочтения пользователей. Введение возможности создания пользовательских маршрутов позволяет обогатить опыт исследования города, делая его более персонализированным и интерактивным.

На сегодняшний день подобные веб-приложения получают все большее распространение в крупных городах по всему миру, однако в основном они являются платными. Существующие ресурсы часто предлагают статическую информацию, не позволяя пользователям активно участвовать в создании контента и делиться своими уникальными маршрутами. Это создает необходимость в разработке нового подхода, который бы учитывал современные требования пользователей к интерактивности и персонализации информации.

Цель данного проекта – разработка веб-приложения, позволяющего пользователям создавать и просматривать маршруты по городу Екатеринбург, делиться своими впечатлениями и рекомендациями с другими пользователями.

Для реализации решения потребуется разработка серверной и клиентской части и последующее сопровождение на хостинге. Полноценный бэкенд позволит хранить множество различных маршрутов, а проработанный фронтенд доступно представит данные пользователю. Основываясь на требованиях, поставленных в техническом задании, и проведя их анализ, можно будет выявить требования к стеку технологий и подобрать подходящие инструменты.

Для достижения этой цели необходимо решить следующие задачи:

1. Изучить существующие решения и методы поиска пользовательских маршрутов в веб-приложениях.
2. Описать архитектуру и функциональные требования к веб-приложению, обеспечивающему создание и обмен маршрутами.
3. Разработать и реализовать интерфейс пользователя, обеспечивающий удобный ввод и просмотр маршрутов.
4. Установить механизм интеграции картографических сервисов для визуализации маршрутов на карте города.
5. Провести тестирование веб-приложения с целью выявления и устранения возможных ошибок и недочетов.

Выполнение данных задач позволит создать удобный инструмент для улучшения пользовательского опыта при исследовании города Екатеринбург, а также способствует популяризации интерактивных технологий в сфере туризма и городского планирования.

1. Целевая аудитория

Для определения ЦА и предпочтений потребителей было проведено маркетинговое исследование в форме онлайн-опроса. Метод опроса используется приблизительно в 90% исследований. К достоинствам данного метода можно отнести: возможность выявления потребностей и предпочтений; гибкость формы проведения; возможность статистической обработки информации, минимизация затрат. Выборка составила 92 человека. При составлении анкеты был сделан акцент на методику 5W Марка Шеррингтона. С помощью данной методики можно составить психологический портрет из существующей базы и целевой аудитории. В основе лежат легкие пять вопросов. Психологические портреты, которые получились с помощью методики 5W, помогут «прочувствовать» боли аудитории и точнее попасть в её потребности. Это эффективнее, чем просто разделить её по полу и возрасту. По результатам исследования были определены потенциальные пользователи и потребители нашего сервиса.

В итоге Методика Шеррингтона позволила выделить две целевые группы:

|  |  |  |
| --- | --- | --- |
| **Вопрос** | **ЦА 1** | **ЦА 2** |
| What?  (Что?) | Необычные места города | Популярные маршруты, достопримечательности, места города |
| Who?  (Кто?) | Жители Екатеринбурга | Туристы в Екатеринбурге |
| Why?  (Почему?) | Более глубокое погружение в культуру собственного города, приятно провести вечер/день, открыть новые необычные места | Желание увидеть основные достопримечательности, окунуться в культуру города |
| When?  (Когда?) | Круглый год, преимущественно вечером | Преимущественно летом, дневное время |

Также мы выделили средний возраст нашей ЦА, для более удобного восприятия создали диаграммы на основе всех данных:



На основе проведенного маркетингового исследования, выполненного с использованием онлайн-опроса и методики 5W Марка Шеррингтона, были выявлены две основные целевые группы пользователей для нашего сервиса. Эти группы включают жителей Екатеринбурга, заинтересованных в открытии новых и необычных мест города, а также туристов, стремящихся увидеть основные достопримечательности и погрузиться в культуру города.

Анализ показал, что жители Екатеринбурга предпочитают исследовать город круглый год, преимущественно в вечернее время, в то время как туристы более активны летом и днем. Оба сегмента проявляют интерес к маршрутам, охватывающим различные точки города, что подчеркивает необходимость гибкости и разнообразия в предлагаемом контенте.

Средний возраст пользователей варьируется до 40 лет, с наибольшей концентрацией интереса среди молодых и активных людей до 25 лет. Это указывает на важность создания удобного и интуитивного интерфейса, который будет привлекательным и легким в использовании для этой возрастной группы.

Таким образом, наше исследование подтвердило значимость и потенциальную востребованность сервиса среди целевых групп, предоставив четкое понимание их предпочтений и поведения. Это позволит нам сфокусироваться на удовлетворении конкретных потребностей пользователей, что будет способствовать успешному запуску и дальнейшему развитию нашего продукта.

1. Определение проблемы

Порой люди не знают, куда можно пойти во время своего досуга, из-за этого выбирают вместо активного отдыха сидеть дома. Это особенно актуально для жителей крупных городов, таких как Екатеринбург, где разнообразие мест для посещения может вызывать растерянность и неопределенность в выборе. Также существует проблема для гостей города, которые не знакомы с местными достопримечательностями и интересными маршрутами, и, как следствие, проводят время пассивно, вместо того чтобы исследовать город.

Основные проблемы, которые мы выделили:

* **Отсутствие ресурсов на решение проблемы**: недостаток времени и информации для планирования активного отдыха.
* **Неудачные попытки решить проблему**: попытки найти интересные маршруты и мероприятия самостоятельно часто заканчиваются неудачей из-за разрозненности информации.
* **Недовольство текущим состоянием**: неудовлетворенность пассивным отдыхом дома и желание найти более интересные и активные способы проведения досуга.

Если отнести проблему пользователей к одной из данных групп, то картина становится ясной, что требуется целевой аудитории, и проще составить предложение.

Выделяют также три уровня "боли":

1. **Технический**. Неудобства, вызванные отсутствием единой платформы для планирования маршрутов и досуга в Екатеринбурге.
2. **Финансовый**. Потеря потенциальных возможностей для активного отдыха, что приводит к неудовлетворенности и неэффективному использованию времени.
3. **Личный интерес**. Личная мотивация пользователей связана с желанием получить новые впечатления, исследовать город и провести время с пользой.

Определение проблем пользователей на каждом из этих уровней позволило нам сформировать четкое понимание их потребностей и продумать веб-приложение, которое удовлетворит их ожидания и решит существующие проблемы.

1. Подходы к решению проблемы

Для формирования решения проблемы, мы проанализировали основные варианты пользователей для нахождения маршрутов и достопримечательностей в городе Екатеринбург, а именно:

1. **Поиск вручную.** Использование сервисов, предоставляющих карту города, при помощи которых пользователь может искать пути. Отсюда можем выделить минусы метода: долго и неинформативно.
2. **Обсуждения с другими людьми.** Дискуссия между группой лиц, в ходе которой определяется тот или иной маршрут. Данный метод решения проблемы также имеет ряд минусов: отсутствие заинтересованных людей и отсутствие удобных способов визуально отобразить маршрут.
3. **Использование специализированных сервисов.** Данный метод является наиболее удобным, но зачастую подобные сервисы являются платными и с гидами, ограничивая тем самым свободное передвижение.

На основе преимуществ всех данных методов, мы определили наиболее удобный и быстрый вариант – создание собственного веб-сервиса, который позволит пользователям создавать, обсуждать и сохранять различные маршруты в городе. Наше веб-приложение объединяет возможности поиска, обсуждения и использования специализированных сервисов, устраняя их недостатки и предлагая ряд преимуществ:

* **Создание маршрутов.** Пользователи могут самостоятельно создавать маршруты, отмечая на карте города точки интереса, добавляя описания и фотографии. Это позволяет каждому пользователю делиться своим уникальным опытом и рекомендациями.
* **Просмотр и использование маршрутов.** Все созданные маршруты доступны для просмотра другим пользователям. Они могут выбирать понравившиеся маршруты, следовать им и оставлять свои отзывы и комментарии, делая процесс исследования города более интерактивным и социально значимым.
* **Обсуждение маршрутов.** Встроенные функции комментариев и форумов позволяют пользователям обсуждать маршруты, делиться своими впечатлениями и советами. Это создает сообщество активных исследователей города, которые могут помогать друг другу находить интересные места.
* **Интеграция с картографическими сервисами.** Наше приложение интегрировано с популярными картографическими сервисами, что обеспечивает точное отображение маршрутов и возможность навигации в реальном времени.
* **Бесплатный доступ.** В отличие от многих специализированных сервисов, наше приложение предоставляет бесплатный доступ ко всем основным функциям, что делает его доступным для широкого круга пользователей.

Следует отметить основные преимущества поиска маршрута, используя наше приложение:

* **Удобство использования.** Интуитивно понятный интерфейс позволяет пользователям быстро создавать и находить маршруты, а также обсуждать их с другими.
* **Социальное взаимодействие.** Возможность обсуждения маршрутов и обмена впечатлениями способствует созданию активного сообщества пользователей.
* **Доступность.** Бесплатный доступ к основным функциям делает наше приложение доступным для всех жителей и гостей Екатеринбурга, независимо от их финансовых возможностей.
* **Интерактивность.** Интерактивные карты и возможность добавления мультимедийного контента делают процесс создания и использования маршрутов более увлекательным.

Создание веб-приложения для формирования и обмена маршрутами по городу Екатеринбург решает проблему недостатка удобных и доступных инструментов для планирования активного досуга. Наша платформа объединяет лучшие черты существующих методов и устраняет их недостатки, предоставляя пользователям возможность легко и быстро находить интересные маршруты, делиться своими открытиями и взаимодействовать с другими любителями активного отдыха.

1. Анализ аналогов

Прямые конкуренты – это аналогичное программное обеспечение на аналогичном рынке, работающее с целевой аудиторией, что и разрабатываемый продукт. Косвенные конкуренты – это похожее программное обеспечение с другими характеристиками или абсолютно другой продукт, но работающий с целевой аудиторией, что и разрабатываемый продукт.

Прямые конкуренты нашего веб-приложения – это сервисы, предоставляющие возможности для создания, поиска и обсуждения маршрутов в городе. Мы определили следующие основные конкуренты:

**Туристер**. Сервис, предлагающий пользователям разнообразные маршруты по различным городам, включая Екатеринбург. Пользователи могут просматривать маршруты, оставлять отзывы и делиться своими впечатлениями.

**Трипстер**. Платформа, позволяющая искать и бронировать экскурсии и маршруты с гидами по различным городам. Пользователи могут выбирать из множества предложений, ориентируясь на отзывы и рейтинги.

Критериями для анализа конкурентов будут следующие:

1. Удобство использования
2. Функциональные возможности (создание маршрутов, отзывы, комментарии)
3. Наличие мобильного приложения
4. Стоимость услуг
5. Наличие социальных функций (обсуждения, форумы)
6. Доступность и охват (языковая поддержка, географический охват)

Для анализа конкурентов составим таблицу, в которой будем сравнивать их по указанным критериям.

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **Туристер** | **Трипстер** | **Наш продукт** |
| |  | | --- | | Удобство использования |  |  | | --- | |  | | |  | | --- | | Среднее |  |  | | --- | |  | | |  | | --- | | Среднее |  |  | | --- | |  | | |  | | --- | | Интуитивно понятный интерфейс |  |  | | --- | |  | |
| |  | | --- | | Функциональные возможности |  |  | | --- | |  | | |  | | --- | | Просмотр и создание маршрутов, отзывы |  |  | | --- | |  | | |  | | --- | | Бронирование экскурсий, отзывы |  |  | | --- | |  | | |  | | --- | | Создание маршрутов, отзывы, обсуждения |  |  | | --- | |  | |
| |  | | --- | | Наличие мобильного приложения |  |  | | --- | |  | | Да | Да | Планируется |
| |  | | --- | | Стоимость услуг |  |  | | --- | |  | | |  | | --- | | Бесплатный доступ |  |  | | --- | |  | | |  | | --- | | Платные экскурсии |  |  | | --- | |  | | |  | | --- | | Бесплатный доступ |  |  | | --- | |  | |
| |  | | --- | | Наличие социальных функций |  |  | | --- | |  | | |  | | --- | | Обсуждения |  |  | | --- | |  | | Отсутствуют | Обсуждения |
| |  | | --- | | Доступность и охват |  |  | | --- | |  | | Широкий географический охват, много языков | Ограниченный географический охват, несколько языков | Охват по Екатеринбургу, планируется расширение |
| Наличие блога | Да | Да | Планируется |
| Широкая фильтрация | Да | Нет | Планируется |
| Карта | Нет | Нет | Да |

Сильные стороны нашего продукта:

* Бесплатный доступ ко всем функциям
* Наличие социальных функций (обсуждения, форумы), что способствует созданию активного сообщества
* Интуитивно понятный интерфейс

Слабые стороны:

* Отсутствие мобильного приложения на текущий момент
* Ограниченный географический охват, что пока ограничивает аудиторию только Екатеринбургом

На основе данного анализа мы определили сильные стороны наших конкурентов и выбрали функции и особенности, которые могли бы заимствовать в нашем приложении.

1. Календарный план проекта

Название проекта: Movo

Руководитель проекта: Шадрин Денис Борисович

Таблица 1 – Календарный план

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Ответственный** | **Длительность** | **Дата начала** | **Временные рамки проекта** | | | |
| 1 нед | 2 нед | 3 нед | 4 нед |
| ***Анализ*** | | | | | | | | |
| *1.1* | *Определение проблемы* | Гушшамов Кирилл | 48 часов | 03.04 | + |  |  |  |
| *1.2* | *Выявление целевой аудитории* | Никита Лавров | 32 часа | 04.04 | + |  |  |  |
| *1.3* | *Конкретизация проблемы* | Гушшамов Кирилл | 24 часа | 04.04 | + |  |  |  |
| *1.4* | *Подходы к решению проблемы* | Никита Лавров | 16 часов | 06.04 |  |  |  |  |
| *1.5* | *Анализ аналогов* | Никита Лавров | 38 часов | 10.04 |  | + |  |  |
| *1.6* | *Определение платформы и стека для продукта* | Гушшамов Кирилл | 16 часов | 07.04 | + |  |  |  |
| *1.7* | *Формулирование требований к MVP продукта* | Никита Лавров | 16 часов | 08.04 | + |  |  |  |
| *1.8* | *Определение платформы и стека для MVP* | Никита Лавров | 4 часа | 09.04 | + |  |  |  |
| *1.9* | *Формулировка цели* | Гушшамов Кирилл | 24 часа | 18.04 |  |  | + |  |
| *1.10* | *Формулирование требований к продукту* | Гушшамов Кирилл | 16 часов | 22.04 |  |  | + |  |
| *1.11* | *Определение задач* | Никита Лавров | 16 часов | 25.04 |  |  | + |  |
| ***Проектирование*** | | | | | | | | |
| *2.1* | *Архитектура системы (компоненты, модули системы)* | Никита Лавров | 24 часа | 05.05 |  |  |  | + |
| *2.2* | *Разработка сценариев использования системы* | Гушшамов Кирилл | 16 часов | 10.05 |  |  |  | + |
| *2.3* | *Прототипы интерфейсов* | Никита Лавров | 24 часа | 12.05 |  |  |  | + |
| *2.4* | *Дизайн-макеты* | Никита Лавров | 16 часов | 15.05 |  |  |  | + |
| *2.5* | *Архитектура системы (компоненты, модули системы)* | Гушшамов Кирилл | 24 часа | 16.05 |  |  |  | + |
| ***Разработка*** | | | | | | | | |
| *3.1* | *Написание кода* | Никита Лавров | 48 часов | 20.05 |  |  |  | + |
| *3.2* | *Тестирование приложения* | Гушшамов Кирилл | 16 часов | 29.05 |  |  |  | + |
| ***Внедрение*** | | | | | | | | |
| *4.1* | *Оформление MVP* | Никита Лавров | 8 часов | 02.06 |  |  |  | + |
| *4.2* | *Внедрение MVP* | Гушшамов Кирилл | 8 часов | 03.06 |  |  |  | + |
| *4.3* | *Написание отчета* | Никита Лавров | 8 часов | 04.06 |  |  |  | + |
| *4.4* | *Оформление презентации* | Гушшамов Кирилл | 8 часов | 05.06 |  |  |  | + |
|  | *Защита проекта* |  |  | 07.06 - 15.06 |  |  |  |  |

1. Сценарии использования

Основной концепцией веб-приложения является возможность делиться с другими пользователями маршрутами. Для решения данной проблемы, была продумана UX составляющая проекта.

Сценарий использования описывает процесс создания маршрута зарегистрированным пользователем на веб-платформе для обмена маршрутами. Платформа позволяет пользователям создавать и делиться своими маршрутами с другими участниками, а также сохранять их в избранное и комментировать. Основной задачей данного сценария является демонстрация взаимодействия пользователя с системой при создании нового маршрута, включая ввод информации, отметку точек на карте и публикацию маршрута. Сценарий помогает выявить пользовательские и функциональные требования к системе, обеспечивая достаточный уровень детализации для понимания необходимого набора свойств новой системы.

**Актор:** Зарегистрированный пользователь.

**Цель:** Создание нового маршрута и публикация его на сайте.

**Основной поток событий:**

1. Пользователь, находясь на главной странице, нажимает кнопку "Создать маршрут".
2. Система отображает форму создания маршрута с полями для ввода информации.
3. Пользователь вводит название маршрута, заполняет описание и загружает фотографии.
4. Пользователь отмечает на интерактивной карте начальную, промежуточные и конечную точки маршрута, указывая название и описание каждой точки.
5. Система сохраняет маршрут и публикует его, делая доступным для других пользователей.
6. Система перенаправляет пользователя на страницу созданного маршрута для просмотра и дальнейшего редактирования.

**Результат:**

* Пользователь успешно создает и публикует новый маршрут, который становится доступным для других пользователей на главной странице и в поиске.
* В случае сохранения в черновики, маршрут доступен пользователю для дальнейшего редактирования и публикации в будущем.

Также для полного взаимодействия с другими пользователями, необходимо настроить профиль, то есть задать понятный, читаемый логин и добавить аватар. Если логин можно добавить при регистрации, то аватар можно добавить по желанию пользователя. Рассмотрим сценарий редактирования профиля.

**Актор:** Зарегистрированный пользователь.

**Цель:** Изменить логин и добавить аватар профиля.

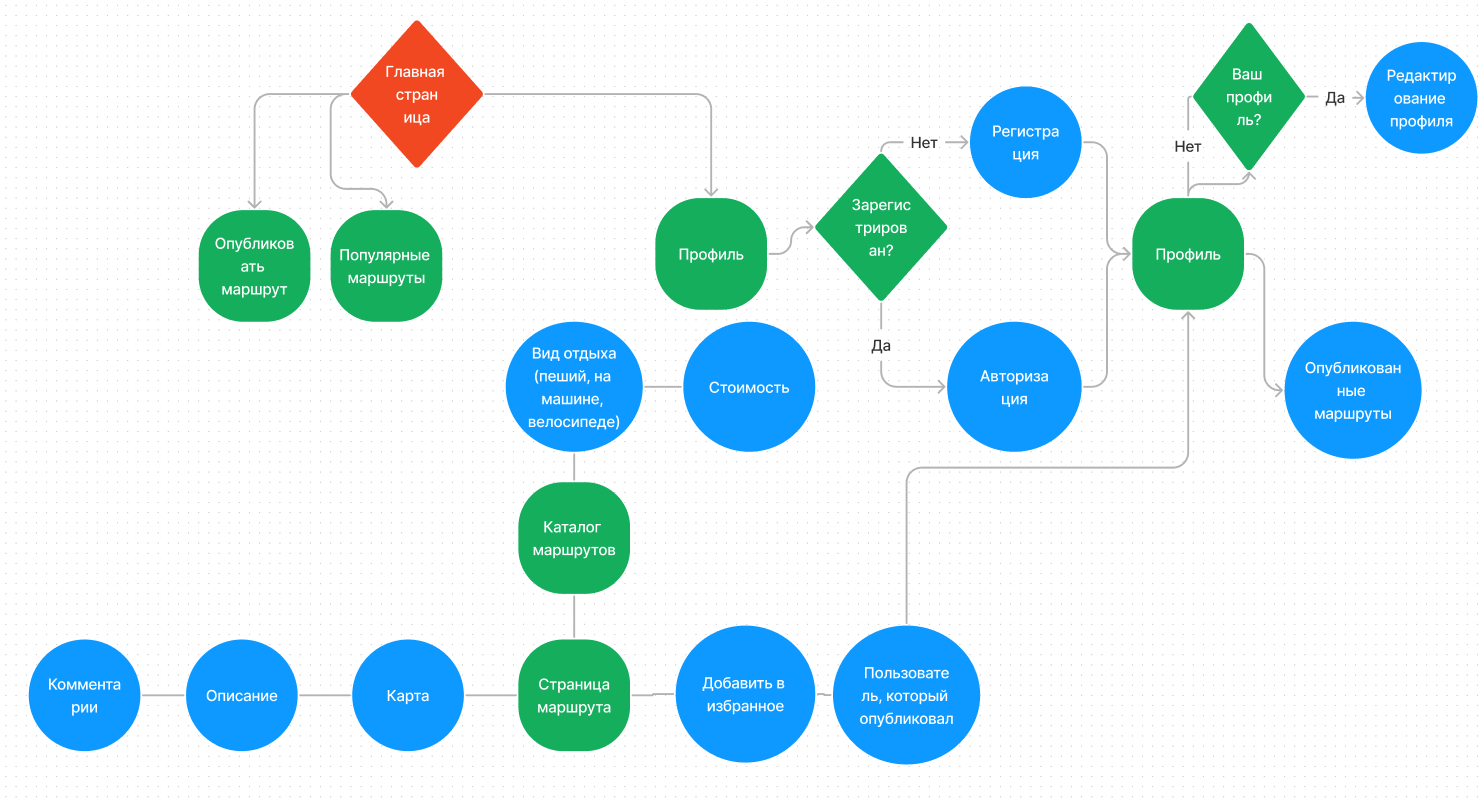
**Основной поток событий:**

1. Пользователь, находясь на главной странице, переходит в “Профиль” с помощью Header’a (Нажимает на аватар).
2. Система перенаправляет пользователя на страницу профиля.
3. Пользователь нажимает на специальную кнопку редактирования.
4. Система перенаправляет на форму редактирования.
5. Пользователь заполняет необходимые поля и добавляет аватар.
6. Система сохраняет измененные данные и перенаправляет на страницу профиля.

**Результат:**

* Пользователь успешно редактирует и сохраняет изменения профиля.

Все пути, которые может проделать пользователь можно увидеть на схеме.



1. Требования к продукту и к MVP

Требования к продукту и к MVP

**Пользовательские требования:**

* Пользователь должен иметь возможность создать новый маршрут, указывая его на карте.
* Пользователь должен иметь возможность публиковать маршрут для общего доступа.
* Пользователь должен иметь возможность загружать фотографии к маршруту.
* Пользователь должен иметь возможность видеть созданный маршрут в своем личном кабинете.
* Пользователь должен иметь возможность редактировать созданные маршруты.

Функциональные требования

**Основные функции:**

* Система должна предоставлять форму для создания маршрута, включающую поля для названия, описания и загрузки фотографий.
* Система должна предоставлять интерактивную карту для отметки начальной, промежуточных и конечной точек маршрута.
* Система должна публиковать маршрут, делая его доступным другим пользователям на главной странице и в поиске.
* Система должна отображать опубликованный маршрут на отдельной странице с полным описанием, картой и фотографиями.
* Система должна предоставлять пользователю возможность редактировать опубликованные маршруты.

Нефункциональные требования

**Удобство использования:**

Интерфейс создания и редактирования маршрутов должен быть интуитивно понятным и доступным для пользователей с базовыми навыками работы с компьютером.

**Надежность:**

Система должна обеспечивать сохранность данных при любом сбое, предоставляя возможность

**Расширяемость:**

Система должна быть спроектирована таким образом, чтобы в будущем можно было легко добавить новые функции, такие как оценка маршрутов, социальный обмен и интеграция с другими картографическими сервисами.

**Удобство сопровождения:**

Код системы должен быть документирован и структурирован для облегчения поддержки и обновления.

Производные требования

* Возможность редактирования маршрутов подразумевает наличие функционала для хранения версии маршрута до редактирования, что обеспечит возврат к предыдущей версии в случае необходимости.
* Публикация маршрута должна автоматически обновлять главный интерфейс и поиск, обеспечивая отображение нового маршрута для всех пользователей.
* Возможность добавления фотографий к маршруту требует реализации механизма загрузки и хранения изображений в базе данных.

Пример:

**Бизнес-требование:** Пользователям необходим инструмент для создания и публикации маршрутов, чтобы делиться ими с другими пользователями.

**Пользовательское требование:** Пользователь должен иметь возможность прикрепить фотографии к создаваемому маршруту.

**Функциональное требование:** Система должна иметь функционал прикрепления фотографий к маршруту, обеспечивая их загрузку и отображение.

**Нефункциональные требования:**

Фотографии должны быть ограничены по размеру до 10 Мб, чтобы уменьшить нагрузку на сервер и сократить затраты на хранение данных.

MVP (Минимально жизнеспособный продукт)

**Основные функции:**

* Регистрация и авторизация пользователей.
* Возможность создания маршрута с указанием точек на интерактивной карте.
* Форма для ввода названия и описания маршрута.
* Возможность загрузки и хранения фотографий.
* Публикация маршрута с отображением на главной странице.
* Просмотр и редактирование созданных маршрутов пользователем.

**Нефункциональные требования для MVP:**

* Ограничение размера загружаемых фотографий до 10 Мб.
* Интуитивно понятный интерфейс для создания и редактирования маршрутов.

1. Стек для разработки

**Анализ технического задания**

В первую очередь необходимо проанализировать написанное техническое задание. Исходя из списка требований к сервису можно выявить требования к стеку технологий и составить список задач в рамках разработки приложения. Только после этого можно будет приступать к выбору инструментов разработки и непосредственно к самому написанию платформы.

Помимо интерактивной карты платформа должна содержать функционал с возможностью оставить комментарии и добавить маршруты в избранное. Настройка координат точек на карте должны производиться на этапе создания маршрута.

Платформа должна иметь систему аккаунтов и личный кабинет пользователя, в первую очередь, для возможности полного взаимодействия с приложением.

Каталог маршрутов объединяет потребность в гибком управлении контентом, то есть должна быть возможность заполнять и редактировать материалы без вмешательства разработчика. Со стороны разработки для реализации потребуется возможность представления этих материалов по различным шаблонам, основанным на предоставленных дизайн-макетах. Также необходимо реализовать систему разделения прав, чтобы ограничить возможность редактирования для обычных пользователей. Указанные разделы должны содержать инструменты сбора обратной связи. Основным таким инструментом является функционал написания комментариев или отзывов под материалами. В ходе анализа технического задания были выявлены и скорректированы основные задачи на разработку, определен формат разрабатываемого приложения. Далее можно приступать к выбору технологического стека и инструментов разработки.

**Анализ возможностей инструментов веб-разработки**

Исходя из анализа технического задания можно выявить требования к инструментам разработки серверной и клиентской частей приложения. Выбранный стек технологий должен содержать:

* возможность быстрого заполнения сервиса контентом;
* расширяемость сервиса во время разработки;
* гибкую работу с формами и шаблонами;
* механизмы аутентификации пользователей;
* встроенные инструменты безопасности.

Так как сервис должен содержать типовой функционал (наполнение контентом, система аккаунтов), то нет необходимости создавать приложение полностью с нуля на чистых языках программирования. В данной ситуации можно рассмотреть разработку на фреймворках или CMS системах.

CMS отлично подойдет, если требуется разработать типовое решение и быстро поднять сайт, созданный по шаблонам. Данное решение покроет ряд базовых сценариев: система аккаунтов, механизмы аутентификации, создание и модерирование материалов на сайте, то есть управление контентом. Но для разработки некоторых разделов сервиса потребуется множество изменений, чтобы реализовать его функционал в полной мере. Также решение на CMS может содержать в себе лишние компоненты, которые затормозят работу платформы.

Разработка на фреймворке займет больше времени: множество механизмов придется разрабатывать с нуля или дорабатывать «из коробки» фреймворка, но при этом сохранится возможность реализации любого своего функционала, выбор всего стека технологий будет гораздо шире, в то время как при работе с CMS инструменты заранее предопределены и очень ограничены.

В результате рассмотрения плюсов и минусов данных вариантов можно сделать вывод, что CMS имеют ряд ограничений, из-за чего теряется гибкость разработки и может пострадать расширяемость сервиса, а на старте разработки это является важной необходимостью. Поэтому стоит остановиться на рассмотрении различных фреймворков для написания web-приложений.

Наиболее популярными фреймворками на данный момент являются Laravel, Django, Flask и Node.js

Проанализируем эти инструменты по основным критериям:

* расширяемость сервиса во время разработки;
* инструменты по работе с шаблонами или представлениями;
* встроенные инструменты безопасности;
* встроенные механизмы аутентификации пользователей;
* инструменты по созданию Rest API;
* наличие административной панели.

**Анализ возможностей фреймворка Laravel**

Laravel – это фреймворк на PHP для разработки бэкенда. Используется для решения сложных задач и написания нестандартных проектов.

Из преимуществ Laravel можно выделить следующее:

* множество готовых инструментов: пакеты аутентификации, маршрутизация, рендеринг HTML, поддержка API.
* наличие ORM;
* базовые механизмы защиты данных;
* гибкость.

Отдельным плюсом и отличием данного фреймворка является гибкость: под него можно найти множество инструментов под самые разные нужды. Также стоит отметить интуитивно понятный код, что упрощает разработку для новичков и производительность.

Из недостатков Laravel можно отметить отсутствие административной панели, но вместо нее существуют готовые консоли, как платные, так и бесплатные. Потребуется дополнительная установка такой консоли для своего приложения или же разработка своей, что займет дополнительное время.

При выборе данного фреймворка покроются почти все требования к инструментам разработки, выявленные на предыдущем этапе.

**Анализ возможностей фреймворка Django**

Django – это веб-фреймворк, написанный на Python. Используется для разработки многофункциональных приложений. Основная цель фреймворка – позволить разработчикам создавать приложения за максимально короткие сроки.

Основными особенностями фреймворка являются:

* высокий уровень безопасности;
* наличие готовых инструментов: приложение аутентификации, маршрутизация, шаблонизаторы;
* встроенная панель администратора;
* возможность постоянного развития приложения;
* наличие ORM.

Можно отметить масштабируемость и производительность фреймворка. Django может обрабатывать высокий трафик, оптимизирует работу с изображениями, а также обгоняет многие другие фреймворки по скорости ответа.

Django имеет удобные инструменты по работе с формами. Это упрощает разработку, так как данные форм можно хранить в виде объектов, а также здесь задействованы механизмы безопасности, например, CSRF защита.

Уникальной особенностью Django является наличие дополнительных библиотек предоставляющим SEO-оптимизацию, что облегчит продвижение проекта в поисковых системах после внедрения и запуска на хостинге.

Из недостатков можно выделить достаточно сложный код, но при этом хорошо структурированный. Также фреймворк не имеет встроенных инструментов для создания собственного Rest API, но существуют отдельные библиотеки для устранения этой проблемы. Время на их подключение достаточно незначительное.

Проанализировав основной инструментарий фреймворка, можно сделать вывод, что он полностью покрывает выявленные требования и дает ряд дополнительных возможностей на перспективу.

**Анализ возможностей фреймворка Flask**

Flask является микрофреймворком, написанным на Python. Во многом похож на Django, предоставляет аналогичный функционал в плане обработки запросов, поддержке документов, но имеет меньший масштаб.

Так как Flask очень похож на Django, то стоит перечислить только их отличия:

* не поддерживает формы по умолчанию;
* отсутствует ORM, но есть инструменты, такие как SQLAlchemy;
* нет приложения аутентификации, но есть безопасные cookies;
* отсутствует административный интерфейс, но есть расширение Flask-Admin.

Учитывая все перечисленные отличия, можно сделать вывод, что Flask по умолчанию имеет гораздо более скромный функционал. Это упрощает разработку для новичков и повышает гибкость для продвинутых разработчиков – можно добавлять только те компоненты, которые действительно нужны, или же разрабатывать их самим. Но так как основная часть перечисленных инструментов потребуется для создания сервиса визуальных новелл, то разработка на Flask займет больше времени, чем разработка на Django.

**Анализ возможностей фреймворка ExpressJS**

ExpressJS – фреймворк для создания веб-приложений на базе Node.js, серверной платформы JavaScript. Является очень гибким, имеет небольшой функционал, но может восполнять недостающие компоненты благодаря внешним модулям.

Преимуществами фреймворка являются:

* гибкость;
* поддержка популярных шаблонизаторов;
* высокая производительность за счет асинхронности;
* поддержка API.

Хорошим преимуществом также является то, что при использовании данного фреймворка проще синхронизировать серверную и клиентскую часть, так как разработка осуществляется на одном языке программирования.

По умолчанию ExpressJS имеет очень ограниченный функционал и больше подходит для разработки, совместной с другими решениями. Фреймворк совершенно не оснащен механизмами защиты данных, что потребует дополнительного времени на проработку этой проблемы. Возможно, придется искать уже готовые решения и интегрировать их. Также для администрирования контентом придется разрабатывать свои решения, так как изначально такой возможности нет.

На данном фреймворке можно разработать мощное Rest API. Также на этом стеке можно проводить достаточно нагруженные вычисления, но для хранения данных и управления контентом этот стек не подходит.

Исходя из анализа функционала и возможностей ExpressJS можно прийти к выводу о том, что данный стек не подходит для решения поставленной задачи. Разработка займет достаточно много времени, так как потребуется своя реализация очень многих компонентов, что можно сделать проще и быстрее при помощи других фреймворков.

После анализа возможных стеков технологий было принято решение написать веб-приложение на фреймворке Django. Это обусловлено тем, что Django предоставляет множество уже готовых инструментов, которые не придется писать с нуля и тем самым затягивать разработку. Такими инструментами являются административная панель с графическим интерфейсом для работы с данными, механизмы аутентификации пользователя, механизмы защиты данных от распространенных атак, упрощенная работа с формами и наличие шаблонизаторов.

Административная панель поможет быстро заполнить платформу контентом (статьями, лекциями в онлайн-курсе), а также позволит в будущем это делать без участия программиста.

Готовая аутентификация ускорит разработку личного кабинета и предоставит «из коробки» возможности авторизации, регистрации и сброса пароля пользователей.

Формы в Django упростят верстку и написание логики работы онлайн-курса. Почти во всех темах есть тесты на закрепление теории или же задачи на написание кода, так как каждый тест представляет из себя как минимум одну форму.

Учитывая выбранный стек технологий, можно определить требования к программному обеспечению. Для серверной стороны приложение и все его файлы должны быть оптимизированы для использования следующего набора программного обеспечения: Apache/nginx, Python 3, PostgreSQL. С клиентской стороны сервис должен быть доступен для посетителей, использующих браузеры FireFox 3.6 и выше, Opera 9 и выше, Google Chrome 10 и выше.

**Разработка на фреймворке Django**

Фреймворк Django позволит быстро создать MVP сервиса. В первую очередь потребуется проработать структуру базы данных: прописать модели для хранения информации, задать им необходимые свойства. Так как Django обладает поддержкой ORM с богатым функционалом, то при создании приложения не придется напрямую работать с СУБД. Взаимодействие с базой осуществляется полностью через код на Python. Также Django поддерживает инструменты, которые помогают разрабатывать REST API (например, Django REST framework). Сериализация данных и создание API приложения необходимы для связи с клиентской частью.

В качестве СУБД в проекте используется SQLite. Django позволяет реализовать клиентскую часть приложения при помощи обычной верстки на HTML и поддержки шаблонов. Для верстки и добавления интерактивности некоторых разделов веб-сервиса, дополнительно потребуется JavaScript.

В итоге можно перечислить все используемые инструменты для разработки:

1. серверная часть:
   * Python 3;
   * Django;
   * Django REST framework.
2. СУБД:
   * SQLite.
3. клиентская часть:
   * HTML;
   * CSS;
   * JavaScript.
4. Прототипирование

Процесс создания прототипа состоит из четырёх шагов:

* определение начальных требований;
* разработка макета сайта в figma;
* этап изучения сайта потенциальным пользователем. Получение обратной связи о необходимых изменениях и дополнениях;
* переработка макета с учетом полученных замечаний и предложений.

*Качества, которыми должен обладать эффективный прототип*

Этап создания прототипа не должен быть затяжным.

Эффективные прототипы являются одноразовыми. Они предназначены для того чтобы донести идею до заинтересованного лица. После того как идея была донесена, прототип может быть отвергнут.

Эффективные прототипы являются сфокусированными, это означает что следует обращать внимание на сложные части при создании прототипов. Необходимо найти шаблонные взаимодействия, которые давно известны в теории юзабилити.

Необходимо обращать внимание на элементы взаимодействия, которые принесут пользу вашему продукту.

Даже самые опытные специалисты-разработчики допускают ошибки. Это правило особенно очевидно в коллективной работе. По мере выполнения проекта накапливаются мелкие ошибки, допущения, неверные решения. В результате получается плохой продукт при хорошей исходной идеи. Разработка прототипа – средство, позволяющее проанализировать идеи, прежде чем на них будут потрачены время и деньги. Это распространенный в инженерной практике метод. Основная цель, достигаемая при создании прототипа, – это экономия времени и ресурсов. Ценность прототипа заключается в том, что он является внешней оболочкой-моделью отражающей существенные отношения разрабатываемого продукта. По сравнению с реальным продуктом прототипы просты и недороги в разработке. При минимальном вложении средств можно обнаружить ошибки создателей и юзабилити проблемы, и улучшить пользовательский интерфейс до того, как сделаны значительные инвестиции в окончательную разработку и технологии.

Создание эффективного прототипа интерфейса является чрезвычайно важной задачей. Прототип должен хорошо выглядеть, чтобы понравиться заказчику и не вызвать вопросов у субъектов тестирования, он должен быть максимально дёшев, максимально полон и, что немаловажно, должен с лёгкостью обновляться.

Требования к прототипу изменяются со временем. Сначала наиболее актуальными его свойствами являются скорость создания и простота модификации. Эти свойства позволяют быстро разработать и проверить несколько версий интерфейса, при этом ещё и исправить значительную часть ошибок.

Затем на первый план выходят функциональность и эстетичность, простота же модификации уже не столь важна, поскольку с каждой новой исправленной ошибкой снижается вероятность того, что прототип придётся полностью переделывать при обнаружении новой ошибки.

1. Проектирование и разработка системы

На этапе проектирования и разработки системы формируется общая структура программного комплекса. В соответствии с технологией нисходящего структурного программирования система разбивается на программные модули, для каждого из которых формулируются требования по реализуемым функциям и разрабатываются алгоритмы.

Описание программных модулей

**Модуль регистрации и авторизации**

**Функции:** Регистрация новых пользователей, авторизация существующих пользователей, управление сессиями.

**Алгоритм:**

1. Пользователь вводит данные (логин, пароль).
2. Система проверяет данные на корректность и существование пользователя в базе данных.
3. В случае успешной проверки пользователь получает доступ к системе.
4. При регистрации система создает новую учетную запись и сохраняет данные в базе данных.

**Модуль создания маршрута**

**Функции:** Создание нового маршрута, сохранение маршрута как черновика, публикация маршрута.

**Алгоритм:**

1. Пользователь переходит на страницу создания маршрута.
2. Пользователь вводит название и описание маршрута, загружает фотографии.
3. Пользователь отмечает точки маршрута на интерактивной карте.
4. Пользователь выбирает действие: сохранить как черновик или опубликовать.
5. Система сохраняет маршрут в базе данных и перенаправляет пользователя на страницу созданного маршрута.

**Модуль управления маршрутами**

**Функции:** Просмотр, редактирование, удаление маршрутов.

**Алгоритм:**

1. Пользователь переходит в личный кабинет.
2. Система отображает список созданных пользователем маршрутов.
3. Пользователь выбирает действие (редактировать, удалить).
4. Система выполняет соответствующее действие и обновляет данные в базе данных.

**Модуль просмотра маршрутов**

**Функции:** Просмотр опубликованных маршрутов, фильтрация и поиск маршрутов.

**Алгоритм:**

1. Пользователь переходит на главную страницу.
2. Система отображает список опубликованных маршрутов.
3. Пользователь может воспользоваться фильтрами и поиском для нахождения интересующих маршрутов.
4. Пользователь выбирает маршрут для просмотра, и система отображает детальную информацию о маршруте.

**Модуль комментариев и избранного**

**Функции:** Добавление комментариев к маршрутам, добавление маршрутов в избранное.

**Алгоритм:**

1. Пользователь открывает страницу маршрута.
2. Пользователь вводит текст комментария и нажимает кнопку для его добавления.
3. Система сохраняет комментарий в базе данных и обновляет страницу.
4. Пользователь может добавить маршрут в избранное, и система сохраняет эту информацию в базе данных.

**Алгоритм работы приложения**

1. Пользователь заходит на сайт и авторизуется.
2. Пользователь попадает на главную страницу, где видит список доступных маршрутов.
3. Пользователь может просматривать маршруты, использовать поиск и фильтры.
4. Для создания нового маршрута пользователь нажимает кнопку "Создать маршрут".
5. Пользователь заполняет форму создания маршрута, указывает точки на карте и загружает фотографии.
6. Пользователь сохраняет маршрут как черновик или публикует его.
7. Пользователь может редактировать и управлять своими маршрутами через личный кабинет.
8. Другие пользователи могут просматривать опубликованные маршруты, добавлять их в избранное и оставлять комментарии.

ЗАКЛЮЧЕНИЕ

Проектирование и разработка веб-приложения для создания и управления маршрутами на карте является комплексной задачей, требующей детального анализа целевой аудитории, тщательной разработки структуры системы и применения современных методов программирования и тестирования.

На основе проведенного маркетингового исследования с использованием методики 5W Марка Шеррингтона были определены две основные целевые группы пользователей: жители Екатеринбурга и туристы, посещающие город. Анализ их потребностей и предпочтений позволил сформировать требования к функционалу и интерфейсу веб-приложения. Процесс проектирования включал разбиение системы на модули, каждый из которых отвечает за определенный функционал, такой как регистрация и авторизация, создание маршрутов, управление маршрутами, просмотр маршрутов, комментарии и избранное. Алгоритмы работы каждого модуля разработаны с учетом удобства и интуитивности взаимодействия для пользователей. Разработка системы велась по методологии Agile, что обеспечивало гибкость и возможность оперативно вносить изменения в зависимости от новых требований и результатов тестирования. Взаимодействие между модулями, а также хранение и обработка данных организованы таким образом, чтобы обеспечить надежность и масштабируемость приложения. Таким образом, созданное веб-приложение отвечает требованиям целевой аудитории, предоставляет удобный интерфейс для создания и управления маршрутами, а также обеспечивает стабильную работу за счет продуманной архитектуры и использования передовых методов разработки. Реализованные функции и возможности приложения позволяют пользователям получать максимум пользы и удовольствия от использования сервиса, что является залогом его успешного функционирования и развития в будущем.

СПИСОК ЛИТЕРАТУРЫ

1. CMS или фреймворк: что выбрать для разработки сайта : сайт. – URL: https://codestudio.org/ru/blog/cms\_vs\_framework/
2. Most Popular Backend : сайт. – URL: https://statisticsanddata.org/data/most-popular-backend-frameworks-2012-2022/
3. Сравнение Django, Laravel и Express. : сайт. – URL: https://ru.itprofit.dev/blog/sravnenie-django-laravel-i-express-js/
4. Python: Django ORM Основные концепции : сайт. – URL: https://ru.hexlet.io/courses/python-django-orm/lessons/concepts/theory\_unit
5. Django. Разработка веб-приложений на Python / Джефф Форсье, Пол Биссекс, Уэсли Чан ; [пер. с англ. А. Киселева] .— Санкт-Петербург ; Москва : Символ-Плюс, 2009 .— 456 с. : ил. ; 24 см .— (High tech) .— Алф. указ.: с. 430-451
6. Поведение потребителей : Блэкуэлл Р., Миниард П., Энджел Дж. 10-е изд. / Пер. с англ. — СПб.: Питер, 2007. — 944 с:ил. — (Серия «Классический зарубежный учебник»)
7. Django-документация : сайт – URL: https://django.fun/docs/django/5.0/