Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Волгоградский государственный технический университет»

|  |  |
| --- | --- |
| Факультет | Электроники и вычислительной техники |
| Кафедра | Программное обеспечение автоматизированных систем |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Согласовано | | | | | | | | |  | Утверждаю | | | | | | | | |
|  | | | | | | | | |  | Зав. кафедрой | | | | | | | | |
| (должность гл. специалиста предприятия) | | | | | | | | |  |
|  | | | |  |  | | | |  |  | | | |  | О. А. Сычев | | | |
| (подпись) | | | |  | (инициалы, фамилия) | | | |  | (подпись) | | | |  | (инициалы, фамилия) | | | |
| « |  | » |  | | | 20 |  | г. |  | « |  | » |  | | | 20 |  | г. |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| к | выпускной квалификационной работе бакалавра | | | | | | | | | | | | | | | на тему |
| (наименование вида работы) | | | | | | | | | | | | | | | | |
| Разработка веб-сервиса для организации ссылок на интернет-ресурсы | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | |  | | | | | |  |  | | | | | | | |
| Автор | |  | | | | | |  | Бирюкова Ирина Алексеевна | | | | | | | |
|  | | (подпись и дата подписания) | | | | | |  | (фамилия, имя, отчество) | | | | | | | |
| Обозначение | | | ВКРБ–09.03.04–10.19–03–24 | | | | | | |  | | | | | | |
|  | | | (код документа) | | | | |  | | | | | | | | |
| Группа | | | ПрИн-466 | | | | |  | | | | | | | | |
|  | | | (шифр группы) | | | | |  | | | | | | | | |
| Направление | | | 09.03.04 – Программная инженерия,  Разработка программно-информационных систем | | | | | | | | | | | | | |
|  | | | (код и наименование направления, наименование программы (профиля)) | | | | | | | | | | | | | |
| Руководитель работы | | | | |  | | | | | | | |  | | Гилка В.В. | |
|  | | | | | (подпись и дата подписания) | | | | | | | |  | | (инициалы и фамилия) | |
| Консультанты по разделам: | | | | | | | | | | | | | | | | |
|  | | | | | |  |  | | | | |  | |  | | |
| (краткое наименование раздела) | | | | | |  | (подпись и дата подписания) | | | | |  | | (инициалы и фамилия) | | |
|  | | | | | |  |  | | | | |  | |  | | |
| (краткое наименование раздела) | | | | | |  | (подпись и дата подписания) | | | | |  | | (инициалы и фамилия) | | |
| Нормоконтролер: | | | |  | | | | | | |  | Кузнецова А.С. | | | | |
|  | | | | (подпись и дата подписания) | | | | | | |  | (инициалы и фамилия) | | | | |

Волгоград 2024 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Волгоградский государственный технический университет»

|  |  |
| --- | --- |
| Кафедра | Программное обеспечение автоматизированных систем |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Утверждаю | | | | | Зав. кафедрой | | | |
|  |  | | | |  | О. А. Сычев | | | |
| (подпись) | | | |  | (инициалы, фамилия) | | | |
|  | « |  | » |  | | | 20 |  | г. |

**Задание**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| на | выпускную квалификационную работу бакалавра | | | | | | | | | | | | | | |
|  | (наименование вида работы) | | | | | | | | | | | | | | |
| Студент | | | Бирюкова Ирина Алексеевна | | | | | | | | | | | | |
|  | (фамилия, имя, отчество) | | | | | | | | | | | | | | |
| Код кафедры | | | | 10.19 | Группа | | | | ПрИн-466 | | | |  | | |
| Тема | | Разработка веб-сервиса для организации ссылок на интернет-ресурсы | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
| Утверждена приказом по университету | | | | | | « | 01 | » | | сентября | 20 | 23 | | г. № | 1074-ст |
| Срок представления готовой работы (проекта) | | | | | | | | | |  | | | | | |
|  | | | | | | | | | | (дата, подпись студента) | | | | | |
| Исходные данные для выполнения работы (проекта) | | | | | | | | | | | | | | | |
| Задание, выданное научным руководителем кафедры «ПОАС» | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
| Содержание основной части пояснительной записки | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
| Перечень графического материала | |
| 1) |  |
|  | |
| 2) |  |
|  | |
| 3) |  |
|  | |
| 4) |  |
|  | |
| 5) |  |
|  | |
| 6) |  |
|  | |
| 7) |  |
|  | |
| 8) |  |
|  | |
| 9) |  |
|  | |
| 10) |  |
|  | |
| 11) |  |
|  | |
| 12) |  |
|  | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель работы (проекта) | |  | |  | | Гилка В.В. | |
|  | | (подпись и дата подписания) | |  | | (инициалы и фамилия) | |
| Консультанты по разделам: | |  | | | |  | |
|  |  | |  | |  | |  |
| (краткое наименование раздела) |  | | (подпись и дата подписания) | |  | | (инициалы и фамилия) |
|  |  | |  | |  | |  |

Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Кафедра «Программное обеспечение автоматизированных систем»

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ:  Зав. кафедрой ПОАС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.А. Сычев  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20 г. |

Разработка мобильного приложения для заказа и доставки еды

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ВКРБ–09.03.04–10.19–03–24–81

Листов 65

|  |  |
| --- | --- |
|  | Руководитель работы  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гилка В.В.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_20 г. |
| |  | | --- | | Нормоконтролер  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузнецова А.С.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 г. | | |  | | --- | | Исполнитель  студент группы ПрИн-467  \_\_\_\_\_Бирюкова Ирина Алексеевна «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 20 г. | |

Волгоград, 2024 г.

Аннотация

Настоящий документ является пояснительной запиской к выпускной квалификационной работе бакалавра на тему: «Разработка веб-сервиса для организации ссылок на интернет-ресурсы».

В работе обосновывается актуальность выбранной темы, проводится ….

Документ включает в себя страниц -…, рисунков - …., приложений -...

Ключевые слова: менеджер интернет-ссылок, интернет-ресурсы, отложенное чтение, интеллектуальные карты, систематизация знаний, организация информации, веб-сервис.

Содержание

Введение 7

1. Анализ проблемы и существующих решений 9

1.1 Введение в исследование управления интернет-ресурсами 9

1.2 Определение основных понятий предметной области

1.3 Характеристика предметной области

1.4 Описание существующих способов/процессов решения задачи, их достоинства и недостатки

1.5 Перспективные подходы к решению задачи

1.6 Обзор аналогов

1.6.1 Критерии сравнения

1.6.2 Сравнение аналогов

1.6.2.1 Pocket

1.6.2.2 Papaly

1.6.2.3 Evernote

1.6.2.4 Xmind

2. Предлагаемый способ решения поставленной задачи

2.1 Решение проблемы управления ссылками на интернет-ресурсы

2.2 Подбор компонентов веб-сервиса

2.3 Задачи веб-сервиса

Выводы

3 Проектирование и разработка веб-сервиса

3.1 Требования к функциональным характеристикам

3.2 Требования к нефункциональным характеристикам

3.3 Варианты использования разрабатываемого веб-сервиса

3.4 Проектирование веб-сервиса

3.5 Проектирование базы данных

3.6 Проектирование классов

3.7 Разработка веб-сервиса

3.7.1 Взаимодействие с базой данных

3.7.2 Процедура аутентификации

3.8 Используемые языки разработки и программные средства

Выводы

4 Тестирование веб-сервиса

# 4.1 Ручное тестирование веб-сервиса

# 4.2 Юзабилити-тестирование веб-сервиса

4.3 Пример работы программы

Выводы

# Заключение

# Введение

В современном мире, где информация и ресурсы в интернете растут с невероятной скоростью, возникает острая необходимость в эффективных инструментах для организации и управления ссылками на интернет-ресурсы. Разработка веб-сервиса для организации ссылок становится важным заданием, способным облегчить доступ к нужным данным, улучшить управление знаниями и повысить производительность работы как отдельных пользователей, так и целых организаций. В свете этого, актуальность разработки веб-сервисов для организации ссылок на интернет-ресурсы обусловлена следующим: непрерывный рост объемов информации в интернете приводит к тому, что поиск и управление релевантной информацией становится всё более сложной задачей. Без эффективных инструментов для организации ссылок пользователи могут тратить значительное количество времени на поиски уже ранее найденной информации, что негативно сказывается на их производительности и эффективности. Проблематика разработки таких сервисов заключается в необходимости создания систем, которые были бы достаточно гибкими для удовлетворения различных потребностей пользователей, включая индивидуальные предпочтения в организации и категоризации ссылок.

Разработка подобных сервисов имеет ключевое значение не только для индивидуальных пользователей, но и для предприятий больших размеров. В корпоративной среде эффективная организация ссылок может значительно улучшить совместную работу, облегчить доступ к корпоративным знаниям и ресурсам, а также повысить общую эффективность рабочих процессов. В академической среде такие инструменты могут помочь исследователям и студентам в управлении обширными базами данных литературы, исследований и источников.

Целью работы является организация для пользователей эффективного управления ссылками на интернет-ресурсы, повышение производительности работы и упрощение доступа к необходимой информации.

Задачи:

* произвести анализ предметной области;
* произвести обзор существующих аналогов и выявить их преимущества и недостатки;
* определить требования к разрабатываемому веб-сервису;
* произвести проектирование базы данных исходя из требований;
* разработка веб-сервиса;
* протестировать разработанный веб-сервис и доказать его работоспособности и эффективность.

Объектом исследования в работе является процесс управления информационными ресурсами в сети интернет.

Предметом исследования является разработка веб-сервиса для организации ссылок на интернет-ресурсы.

Методы исследований. Для решения поставленных задач были использованы методы математического моделирования, системного анализа, программной инженерии, объектно-ориентированного программирования, технологии проектирования человеко-машинного взаимодействия.

Практическая ценность работы заключается в том, что разработанный сервис значительно облегчит процесс поиска, хранения и управления информационными ресурсами для пользователей, что позволит экономить время и повысит их производительность. Улучшенные механизмы поиска и категоризации информации позволят пользователям быстрее находить необходимые ресурсы, что способствует более эффективной работе как индивидуальных пользователей, так и организаций.

1. Анализ проблемы и существующих решений

1.1 Введение в исследование управления интернет-ресурсами

В эпоху информационного устройства общества, где доступ к различным знаниям и данным становится неограниченным, управление веб-ссылками приобретает особую значимость. Пользователям необходим практичный инструмент, дающий возможность не потеряться в бесконечном потоке информации.

Cохранение ссылок - это не просто техническая процедура, но и важный элемент личной и профессиональной организации. Нередко люди делятся интересным контентом с друзьями, коллегами или, например, подписчиками в социальных сетях. Так или иначе человек находится в постоянном потоке информации, даже если он не ищет новые материалы целенаправленно.

Помимо управления информацией, хранилище ссылок также становится важным инструментом для создания построения базы знаний. Сохраняя ссылки на интересные статьи, исследования или источники, человек создает собственную коллекцию полезной информации. Этот архив может быть использован в будущем для поддержки аргументов, написания статей или просто для личного образования.

Люди предпочитают сохранять ссылки на интернет-ресурсы по большому количеству причин, и, одной из главных является потребность в доступе к информации не в данный момент, но в будущем. Человеку необходима удобная возможность возвращения к важным ресурсам – статьям, учебным материалам, блогам, видео и т.д. При этом, поиск конкретного сохраненного ресурса должен быть минимален по времени и затраченным усилиям. Особенную важность такой подход приобретает, когда объем интересующей человека информации растет со значительной скоростью. Ведь вместе с сохранением информации приходит и проблема – ориентироваться в ней со временем становится чрезвычайно сложно.

По вышеописанным причинам возникает необходимость разработки новых решений для управления веб-ссылками. Удобный сервис должен не только иметь возможность хранения информации, но и предоставлять сопутствующие этому функции, например, категоризацию, напоминания, визуализацию. Такой подход помогает пользователям не просто сохранять интересующие ресурсы в одном месте, а еще и упрощать их изучение.

Таким образом, создание сервиса, помогающего в организации и управлении своими ресурсами упрощает работу с сохраненными материалами, способствует более комфортному изучению информации, снижает уровень информационной нагрузки на пользователей и риск потери ими информации из-за человеческого фактора.

1.2 Определение основных понятий предметной области

Основные понятия предметной области и их определения:

* веб-ссылка, или URL (Uniform Resource Locator) – это уникальный адрес, который указывает на местоположение ресурса в сети Интернет и используется для получения доступа к нему;
* менеджер ссылок – это программное обеспечение или веб-сервис, чьей функцией является хранение, организация и управление веб-ссылками. Разные менеджеры предоставляют различные инструменты для работы с сохраненными материалами и от целей и потербностей пользователя зависит выбор конкретного сервиса;
* категория — это группа, в которую пользователи могут объединять связанные по смыслу веб-ссылки. С помощью категорий можно легко упорядочить сохраненные материалы, тем самым облегчив навигацию и последующий поиск;
* визуализация ссылок – это представление сохраненных веб-ссылок в графическом, интуитивно понятном виде, например, в формате интеллектуальных карт, таблиц или списков. Наглядность способствует лучшему ориентированию и запоминанию информации, позволяет видеть общую картину в целом. Инструменты визуализации часто используются при мозговых штурмах, в планировании работы или обучения;
* интеллектуальные карты (Mindmaps) — это метод представления информации в виде диаграмм, состоящих из связанных между собой узлов. На карте имеется главный узел, соответствующий теме карты, а от него идут ответвления к другим блокам, содержащим в себе подчиненные темы; могут иметься и отдельные блоки информации, текстовые пояснения;
* система напоминаний — это функция, которая уведомляет пользователей о необходимости вернуться к определенным ссылкам или ресурсам. Она является помощником пользователя и берет на себя работу по контролю прочитанных и отложенных материалов, позволяя не потерять интересные ресурсы. Напоминания могут представлять собой всплывающие уведомления, письма на электронную почту, список в специально отведенном для этого разделе. С такой системой пользователь может быть уверен, что не пропустит важную информацию.

Из данных понятий состоит основа предметной области управления ссылками на интернет-ресурсы. Знание и понимание данных терминов помогают лучше разобраться в исследуемой сфере, назначении и характеристиках существующих инструментов и сервисов. Оперируя этими терминами можно выстроить полноценный и четкий план по работе над проектом, который связан с управлением ссылками на ресурсы в сети Интернет.

1.3 Характеристика предметной области

Главной целью управления веб-ссылками является упрощение доступа к важной информации, что повышает продуктивность работы пользователя, улучшает организацию его знаний.

На данный момент современными инструментами предлагается широкий выбор функциональных возможностей для управления веб-ссылками. Пользователи могут сохранять систематизировать веб-ссылки путем разделения по категориям или добавления тегов, представлять их в визуальном виде, например, списками или таблицами. Доступна возможность быстрого поиска записей, в том числе при помощи фильтров. Некоторые сервисы предоставляют возможность работы с различных устройств, а также коллективной работы в одном архиве записей. Существуют также и встроенные напоминания, мотивирующие пользователей вернуться к ранее сохраненным или прочитанным записям, что предотвращает забывание и потерю важных материалов.

Потребность в управлении веб-ссылками возникает у различных категорий людей. Студентам это помогает организовать учебные материалы и онлайн-курсы. Создателям контента или фрилансерам – источники и инструменты. Любители и энтузиасты могут собирать материалы, связанные с личными интересами.

Несмотря на очевидные преимущества, в данной сфере имеется ряд проблем, требующих внимания. Существующие сервисы, хоть и предоставляют множество различных функций, но все же имеют такие недостатки как однотипность или полное отсутствие визуальных представлений записей, отсутствие системы напоминаний о прочитанных и отложенных записях. Таким образом, большинство сервисов ориентируется лишь на хранение, но не на активную работу с изучением сохраненных ресурсов. Также, те сервисы, что все же предоставляют подобные функции, часто оказываются перегруженными и сложными для освоения сервисами.

Таким образом, управление веб-ссылками является важной сферой в современной жизни людей, желающих упростить работу с изучением информации. Инструменты и технологии могут значительно повысить производительность работы пользователей и помочь им управлять знаниями. Именно поэтому важно иметь такой сервис, который бы специализировался не только на хранении информации, но и на взаимодействии с ней. И, так как большинство предлагаемых продуктов не соответствуют этому требованию, разработка нового веб-сервиса является актуальной задачей.

Разработка инновационных решений в области управления веб-ссылками становится актуальной задачей, которая может принести значительные выгоды и для индивидуальных пользователей, и для общества в целом. Создание удобных, интуитивно понятных и функциональных инструментов поможет повысить эффективность работы с информацией и стимулировать интеллектуальное развитие.

1.4 Описание существующих способов/процессов решения задачи, их достоинства и недостатки

Существует несколько различных способов организации ссылок на интернет-ресурсы, в их число входят такие как: использование закладок браузера, управление ссылками в рамках приложений для заметок и планирования, а также использование специально предназначенных для организации и управления ссылками веб-сервисов. Некоторым людям требуется упор на визуальную составляющую, поэтому такие пользователи используют современные средства для хранения информации, например, интеллектуальной карты. Все перечисленные способы управления данными имеют как достоинства, так и определенные недостатки.

Далее приведены достоинства и недостатки каждого из способов организации ссылок на интернет-ресурсы:

* закладки в браузере;
  + 1. достоинства;
    2. просты в использовании;
    3. доступны на разных платформах;
    4. имеют возможность создания папок и тегирования сохраненных записей.
    5. недостатки.
    6. возможность категоризации ссылок сильно ограничена;
    7. отсутствует комментирование сохраненных записей;
    8. не имеется визуализация, не предусмотрена система напоминаний.
* приложения для заметок и планирования;
  + 1. достоинства;
    2. удобное добавление заметок с возможностью категоризации;
    3. часто предоставляется синхронизация на разных устройствах;
    4. обеспечивается поддержка совместной работы нескольких пользователей.
    5. недостатки.
    6. отсутствуют специализированные функции для работы с записями в виде ссылок;
    7. возможности визуализации ссылок весьма ограничены либо отсутствуют;
    8. отсутствует система напоминаний.
* специализированные веб-сервисы для организации ссылок;
  + 1. достоинства;
    2. к веб-сервисам легко получить доступ с любого устройства;
    3. имеют расширенные возможности визуализации, при этом она может быть организована оригинальным, новаторским образом;
    4. имеется категоризация и тегирование;
    5. некоторые сервисы предоставляют аналитику использования;
    6. может иметься настраеваемая система напоминаний;
    7. наличие разделения записей по отложенному и прочитанному.
    8. недостатки.
    9. от пользователей перед работой требуется изучение нового сервиса и его индивидуальных особенностей.
* сервисы интеллектуальных карт.
  + 1. достоинства;
    2. широкие возможности визуализации;
    3. зачастую сервисы интеллектуальных карт имеют простой и понятный интерфейс.
    4. недостатки.

1. отсутствуют специализированные функции для работы с записями в виде ссылок;
2. отсутствует категоризация;
3. отсутствует система напоминаний;
4. отсутствует разделение записей на отложенные и прочитанные.

При выборе метода организации ссылок важно учитывать специфику потребностей и предпочтений целевой аудитории. Каждая категория пользователей может иметь запрос на определенную направленность сервиса, а также нуждаться в определенных функциях или особенностях. Тем не менее, учитывая рассмотренные выше варианты, можно сделать вывод о том, что специализированные веб-сервисы являются хорошим выбором для разработки, так как обладают наименьшим числом недостатков и предоставляют потенциально широкий спектр возможностей. Это делает их перспективным вариантом сервиса для управления ссылками на интернет-ресурсы для многих пользователей.

1.5 Перспективные подходы к решению задачи

Для создания веб-сервиса для организации ссылок можно рассмотреть несколько персперктивных и современных подходов к разработке, описанных далее:

* разработка пользовательского интерфейса: создание приятного и удобного пользовательского интерфейса при помощи современных технологий веб-дизайна и разработки фронтенда, таких как HTML, CSS, JavaScript;
* управление базой данных: разработка эффективной структуры базы данных для хранения всей необходимой информации. В это входит выбор подходящей системы управления базами данных, такой как, например, MySQL или PostgreSQ, а также оптимизация запросов для быстрого доступа к данным;
* разработка бэкенда: создание части приложения, которая будет обрабатывать запросы пользователя, управлять данными и обеспечивать безопасность работы. Это подразумевает выбор языка программирования и необходимых для обеспечения функционала веб-сервиса библиотек;
* масштабируемость и производительность: при разработке веб-сервиса необходимо учесть возможность масштабирования и обеспечить высокую производительность, даже при высокой степени нагрузки. Это может быть достигнуто путем оптимизации запросов к базе данных, использованием кеширования, асинхронных операций и т.д.;
* обратная связь: добавление обратной связи для пользователей, например, таких ее вариаций, как формы обратной связи или отзывов и комментариев;
* безопасность: обеспечение безопасной работы с пользовательскими данными, их защита от угроз – взлома, атак на уязвимости, утечек информации. Первостепенной задачей сервиса должено быть обеспечение конфиденциальноси данных пользователя надежным образом.

1.6 Обзор аналогов

Анализ конкурентов веб-сервиса организации ссылок на интернет-ресурсы играет важную роль в определении конкурентного преимущества и разработке уникального предложения для целевой аудитории. Понимание сильных и слабых сторон существующих сервисов поможет выявить пробелы в предлагаемых решениях и определить потенциальные возможности для улучшения качества работы и добавления функций в новый веб-сервис.

Обзор аналогов включает изучение основных характеристик существующих решений, чтобы затем сделать вывод о причинах успеха и популярности определенных сервисов. Также необходимо выявить их слабые места и использовать это как базу для определения плана работы над веб-сервисом, который будет выгодно выделяться на фоне своих конкурентов в данной сфере.

Анализ конкурентов помогает не только понять текущее состояние рынка и ожидания пользователей, но и определить стратегию развития нового веб-сервиса в будущем.

1.6.1 Критерии сравнения

Сравнение сервисов-конкурентов будет проводиться по представленным ниже критериям:

* простота интерфейса (минималистичность, понятность, неперегруженность): пользователи ценят простоту и понятность интерфейса, так как это облегчает использование сервиса и ускоряет его освоение. Минималистичный дизайн помогает избежать отвлекающих элементов и позволяет пользователю быстро находить нужные функции;
* группировка по категориям: эффективная организация информации необходима для быстрого доступа к сохранённым ресурсам. Возможность группировки по категориям или тегам позволяет пользователям легко систематизировать свои записи по собственным предпочтениям и ориентироваться в них;
* разделение по отложенному и прочитанному: разделение помогает пользователю различать только добавленные и уже прочитанные материалы, что особенно важно при управлении большими объемами информации;
* наличие напоминаний: Напоминания полезны для пользователей, которым нужно вернуться к определённым материалам или закрепить полученные некоторое время назад знания;
* визуальное представление записей: Визуальная привлекательность и наглядность информации делают работу с сервисом приятной и эффективной. Хорошо структурированная и визуально понятная информация помогает пользователю быстрее ориентироваться в своих материалах.

Выбор данных критериев обусловлен задачей охватить основные аспекты, которые наиболее важны для пользователей при работе с менеджерами ссылок и записями. Они обеспечивают баланс между функциональностью, удобством использования и визуальной привлекательностью. Именно эти составляющие обычно ожидает увидеть пользователь, когда только начинает знакомиться с веб-сервисом.

Данные критерии являются ключевыми при определении особенностей хорошего веб-сервиса для организации записей и удовлетворяют потребности большинства пользователей.

1.6.2 Сравнение аналогов

1.6.2.1 Pocket

Pocket – это менеджер ссылок для сохранения интернет-ресурсов, таких как статьи, видео и веб-страницы, для последующего прочтения.

Разбор особенностей сервиса по определенным ранее критериям сравнения:

* простота интерфейса: интуитивно понятный, простой, не перегруженный интерфейс;
* группировка по категориям: возможность организации ссылок с помощью тегов;
* разделение по отложенному и прочитанному: чёткое разделение между сохранёнными для последующего чтения и уже прочитанными материалами;
* наличие напоминаний: нет встроенной функции напоминаний об отложенном и прочитаннном;
* визуальное представление: специализированное визуальное представление отсутствует.

1.6.2.2 Papaly

Papaly – это визуальный менеджер закладок, который позволяет организовывать и управлять ссылками в виде досок.

Разбор особенностей сервиса по определенным ранее критериям сравнения:

* простота интерфейса: интуитивно понятный, простой, не перегруженный интерфейс;
* группировка по категориям: Возможность создания досок и категорий для группировки ссылок;
* разделение по отложенному и прочитанному: нет встроенного разделения, можно организовать вручную;
* наличие напоминаний: нет встроенной функции напоминаний об отложенном и прочитаннном;
* визуальное представление: визуально привлекательные доски и категории для организации ссылок.

1.6.2.3 Evernote

Evernote – это мощный менеджер записей, который позволяет сохранять заметки и веб-страницы.

Разбор особенностей сервиса по определенным ранее критериям сравнения:

* простота интерфейса: сложный интерфейс, хорошо структурированный для профессионального использования, но может казаться перегруженным;
* группировка по категориям: поддержка организации записей в блокноты и использование тегов;
* разделение по отложенному и прочитанному: нет встроенного разделения;
* наличие напоминаний: имеется календарь, в который пользователь может добавить напоминания о заметках, но специализированных напоминаний непосредственно о прочитанных и отложенных записях нет;
* визуальное представление: специализированное визуальное представление отсутствует.

1.6.2.4 Xmind

Xmind – это инструмент для создания интеллектуальных карт, который может быть использован для визуальной организации идей и информации.

Разбор особенностей сервиса по определенным ранее критериям сравнения:

* простота интерфейса: простой интерфейс, ориентированный на визуальное представление информации;
* группировка по категориям: нет специализированного разделения по категориям;
* разделение по отложенному и прочитанному: нет встроенного разделения;
* наличие напоминаний: нет встроенной функции напоминаний об отложенном и прочитаннном;
* визуальное представление: наглядное представление информации в виде интеллектуальных карт.

Сравнение аналогов разрабатываемого веб-сервиса по выделенным критериям приведено далее в таблице 1.

Таблица 1 – Сравнение аналогов по критериям

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерий | Pocket | Papaly | Evernote | Xmind |
| Простота интерфейса | + | + | - | + |
| Группировка по категориям | + | + | + | - |
| Отложенное и прочитанное | + | - | - | - |
| Наличие напоминаний | - | - | + | - |
| Визуальное представление | - | + | - | + |

Итак, по результатам анализа существующих в данной области сервисов видно, что большинство аналогов не отвечают всем критериям, что является мотивацией для разработки собственного веб-сервиса. При ней будут учтены все недочеты, свойственные изученным конкурентам, что сделает разрабатываемый веб-сервис потенциальным выбором пользователей среди всего многообразия вариантов.

Существующие сервисы организации ссылок часто ориентированы на хранение ссылок без возможности представления их в удобном наглядном виде, например в виде интеллектуальной карты. В то время как одни сервисы полезны для сохранения ссылок, другие - для визуализации, третьи - для возвращения к прочитанной информации с целью запоминания, их недостаточно для пользователя, который хочет создать собственную систему знаний с помощью сервиса, совмещающего подобные функции.

Таким образом, создание веб-сервиса с интеллектуальной картой и удобной персональной настройкой хранилища ссылок позволит пользователям визуализировать связи между ссылками в удобной форме и организовать их в соответствии с их потребностями для обучения или работы. Это открывает новые перспективы для пользователей в организации и управлении информацией. Такой сервис обеспечил бы пользователей инструментами для создания собственных наглядных систем ссылок, адаптированных под их конкретные цели и способы обучения, что выгодно отличает его от существующих сервисов организации ссылок.

2. Предлагаемый способ решения поставленной задачи

2.1 Решение проблемы управления ссылками на интернет-ресурсы

Большинство имеющихся на рынке сервисов хранения и управления ссылками на интернет-ресурсы имеют определенный ряд проблем. Сервисы хоть и отлично подходят для создания архива записей ссылок, все же не лишены недостатков.

Первая из проблем заключается в отсутствии удобной визуализации сохраненных ресурсов. Многие сервисы предоставляют однотипную и негибкую визуализацию, что затрудняет работу для тех пользователей, которым необходима наглядность. Списки ссылок представляют информацию в однообразном виде, что существенно затрудняет запоминание информации и ориентирование в ней, поэтому пользователи тратят на это значительно больше времени, вследствие чего эффективность их работы снижается.

Такое положение дел вынуждает пользователей искать отдельные сервисы для выполнения задач визуализации, отдельно изучая особенности каждого рассматриваемого варианта. Затем им приходится вручную переносить сохраненные данные, что отнимает значительное количество времени.

Следующая проблема – отсутствие напоминаний о прочитанных и отложенных записях. Работая в Интернете люди сталкиваются с постоянным и мощным потоком данных, обработка которых занимает большое количество времени и не может быть выполнена как только человек найдет интересующий его ресурс. В данном случае очевидна полезность системы напоминаний о сохраненных записях, помогающая человеку не терять полезную информацию среди множества других сохраненных ссылок.

Нередка ситуация, когда человек за небольшой промежуток времени находит сразу множество интересующих его статей, видео или иных ресурсов, и, разумеется, не может сразу изучить их все. Пользователь сохраняет их для будущего просмотра, однако, если такая ситуация повторяется часто, возникает высокий риск забыть о давно отложенных записях.

Также, если человек желает изучить какой-то материал, то эффективным подходом будет вернуться к прочитанному через некоторое время, чтобы повторить пройденный материал. Это помогает качественнее запоминать информацию на долгий срок.

Третья проблема заключается в перегруженности функционалом тех сервисов, где первые две проблемы так или иначе могут быть решены. Такие сервисы имеют множество возможностей и, несмотря на выгодность данного подхода, он также может иметь и негативный эффект – некоторых пользователей может отпугнуть избыточность и сложность приложения. Множество функций могут остаться неиспользованными и попросту мешать человеку.

Не всем пользователям необходимы сложные сервисы, решающие сразу множество задач. Если для одних это плюс, то другие предпочитают иметь под рукой более узкоспециализированный инструмент, в котором им не придется долго разбираться. К тому же, подобное устройство сервисов может не только создавать путаницу, но и нагружать устройство пользователя, требуя слишком много ресурсов. В конце концов, чем больше имеется фунций в сервисе, тем медленнее может оказаться его работа.

Создание нового веб-сервиса решает вышеописанные проблемы следующим образом: веб-сервис объединяет в себе архив записей ссылок и возможность их визуализации при помощи одной из наиболее популярных систем – интеллектуальных карт. В таком варианте визуализации пользователь сможет создать гибкую систему под собственные нужды, разделяя карты по разным областям знаний, выстраивая собственную логику взаимосвязей между найденными в Интернете ресурсами.

Интеллектуальные карты показывают общую картину собранных по определенной теме ресурсов и помогают пользователю ориентироваться. Человек получает возможность переходить от одного ресурса к другому последовательно, плавно изучая тему и выстраивая в голове сеть ассоциаций. В итоге человек имеет не отрывочные знания, а логически связанную между собой информацию. Отдельным плюсом является возможность быстро заметить, где имеются пробелы и какие новые материалы пользователю стоит отдельно поискать.

Кроме того, данный веб-сервис будет предоставлять возможность получения напоминаний о прочитанных и отложенных записях, что гарантирует снижение риска потери пользователем сохраненной информации в архиве. Пользователи смогут отслеживать, какие из материалов они уже обработали и какие нет, понимать, к каким записям стоит вернуться для закрепления изученной информации. Данная функция повысит эффективность управления сохраненными ссылками.

Система напоминаний будет устроена таким образом, чтобы выдавать пользователям уведомления о необходимости вернуться к ранее сохраненным или прочитанным ссылкам через определенные периоды времени. Это поможет как в организации рабочего процесса, так и в управлении личным временем, что особенно полезно в условиях жизни в большом потоке информации.

В связи ориентированием веб-сервиса на более узкую сферу деятельности его отличительной особенностью также будет простота и минималистичность. Это способствует быстрому освоению сервиса и позволяет не потерять потенциальных пользователей.

Так, создаваемый веб-сервис является актуальным и востребованным решением. Он позволит пользователям лучше организовывать свои информационные ресурсы, повышать продуктивность и упрощать работу с данными.

2.2 Подбор компонентов веб-сервиса

Разработка веб-сервиса для организации ссылок на интернет-ресурсы включает множество компонентов, от базовой регистрации и входа в систему до сложного взаимодействия с интеллектуальными картами. Внимательное проектирование каждой из частей обеспечит удобство использования и функциональность, необходимые для привлечения и удержания пользователей.

При создании веб-сервиса важно учитывать такие аспекты пользовательского взаимодействия как понятность интерфейса и привычки пользователей. При начале работы человек должен быстро сориентироваться в устройстве веб-сервиса, поэтому последний должен быть устроен так, как устроено большинство веб-сайтов в сети Интернет.

Создаваемый веб-сервис должен иметь следующие страницы: главная, профиль, архив записей, архив интеллектуальных карт, страница с отдельной интеллектуальной картой, страницы входа и регистрации.

Пользователям также важна удобная навигация, поэтому на каждой странице должны отображаться меню и шапка. Меню содержит ссылки на главную страницу, архив записей и архив интеллектуальных карт. В шапке находится логотип сервиса, название и ссылка на страницу входа (если пользователь не в системе) или страницу профиля (если пользователь вошел в систему).

Содержание каждой страницы веб-сервиса:

* главная страница: имеет простой интерфейс, состоящий из приветственного сообщения, объясняющего суть веб-сервиса, и кнопки для начала работы, которая перемещает пользователя на страницу входа (если он не в системе) или на архив записей (если пользователь вошел в систему). Главная страница должна быть простой, чтобы плавно ввести пользователя в принцип работы веб-сервиса и направить его дальнейшее взаимодействие с ним;
* страница профиля: содержит информацию пользователя, настройки(напоминаний, визуального стиля), раздел уведомлений о прочитанных и непрочитанных записях, а также кнопку выхода из системы. На данной странице собрано то, что относится непосредственно к самому пользователю, особенностям и характеристикам его работы;
* страница архива записей: отображает списки записей и категорий, а также кнопки для взаимодействия с ними. Позволяет открыть запись и посмотреть все ее характеристики. Запись содержит такую информацию как название, url-адрес ресурса, категорию, описание, отметку прочитано или не прочитано. Категории отображаются сбоку от списка записей, есть 3 стандартных пункта категорий – "все", "прочитанные" и "непрочитанные";
* страница архива интеллектуальных карт: отображает список интеллектуальных карт и кнопки для взаимодействия с ними. Здесь пользователь управляет всеми своими интеллектуальными картами в целом;
* страница интеллектуальной карты: страница открывается при выборе карты в архиве интеллектуальных карт. Содержит блок с интеллектуальной картой, а также кнопки для взаимодействия с ней. Отображаемые на карте записи рядом с названием сразу имеют ссылку представляемый ими интернет-ресурс, поэтому пользователь, при желании, может быстро получить доступ к соответствующим материалам;
* страница входа: имеет поля для ввода электронной почты и пароля, а также кнопку, подтверждающую вход. Под формой входа находится ссылка на страницу регистрации на случай, если у пользователя еще нет аккаунта. Также, при нескольких неудачных попытках входа, пользователь может восстановить пароль, перейдя по появляющейся ссылке;
* страница регистрации: имеет поля для ввода имени, email-почты, пароля и повторения пароля, а также кнопку, подтверждающую регистрацию. Под формой регистрации находится ссылка на страницу входа на случай, если пользователь уже зарегистрирован.

Для обеспечения работы веб-сервиса необходимо разработать серверное приложение, которое будет обрабатывать запросы пользователей и управлять всеми взаимодействиями. Оно должно отвечать требованиям надежности и безопасности. Также требуется создать и настроить базу данных, которая будет хранить данные о пользователях, включая их учетные записи, ссылки, категории и интеллектуальные карты. База данных должна быть оптимизирована под быстрый доступ и обработку большого объема данных.

2.3 Задачи веб-сервиса

Данный веб-сервис должен обеспечивать корректное выполнение следующих задач:

* предоставление возможностей регистрации и авторизации в системе, при необходимости – восстановления пароля. Ползователям необходима возможность легкого и быстрого создания нового аккаунта или входа в уже существующий. Веб-сервис должен обеспечивать безопасность и защиту данных пользователя, поэтому для регистрации и авторизации необходимо использовать современные методы защиты данных и аутентификации;
* информирование пользователя о некорректности вводимых данных. Веб-сервис должен проводить валидацию вводимых пользователем данных, например, на страницах регистрации и входа, а также при создании и редактировании записей, категорий, интеллектуальных карт. Уведомления должны подробно описывать возникшую проблему и доступно объяснять способ ее решения так, чтобы любому пользователю он был понятен;
* предоставление возможностей создания, редактирования и удаления записей, категорий, интеллектуальных карт. Так, пользователь должен иметь полный контроль и возможность управления своими данными в любой момент времени;
* обеспечение возможности сортировки записей при нажатии на соответствующую категорию. Пользователь сможет найти как прочитанные и непрочитанные записи, так и те записи, что он сам определил в конкретную персональную категорию. Такая функция значительно облегчает работу с архивом записей, состоящим из большого числа элементов;
* предоставление возможностей работы с интеллектуальной картой – добавление на нее новых записей, удаление имеющихся, установление связей между записями. На интеллектуальной карте рядом с каждой записью должны показываться ссылки на интернет-ресурсы (если они есть) для обеспечения быстрого перехода на нужный веб-сайт;
* создание напоминаний о прочитанных и отложенных записях. Напоминание о прочитанной записи появляется, если пользователь отметил запись как прочитанную некоторое время назад. Напоминание об отложенной записи появляется, если пользователь добавил запись некоторое время назад и до нынешнего времени не прочел ее. Пользователь может настроить временные периоды, через которые появляются напоминания каждого типа. Система напоминаний – одна из главных функций разрабатываемого веб-сервиса для организации ссылок на интернет-ресурсы.

Выводы

В данной главе описаны основные элементы создаваемого веб-сервиса, объяснена их важность и причины выбора на базе сравнения с имеющимися сервисами. Описано содержание всех страниц веб-сервиса, а также его функционал. Так, благодаря проведенному анализу были выявлены основные задачи разработки и наиболее важные ее аспекты.

Благодаря подробному описанию всех функций и компонентов разрабатываемого веб-сервиса можно выявить более точные требования к нему, предположить ожидания пользователей. Это является одним из самых важных аспектов работы над качественным продуктом, который будет востребованным и конкурентноспособным. Его фукнционал не должен уступать существующим продуктам, а кроме этого, должен иметь ряд преимуществ и решать существующие в данной области проблемы, тем самым привлекая пользователей.

Особенно важно уделить внимание и техническим аспектам – сохранению данных в безопасности, практичности дизайна, доступности и удобности, чтобы пользователи могли использовать веб-сервис в любых условиях и быть уверенными в его стабильной, безопасной и качественной работе.

Таким образом, создаваемый веб-сервис будет актуальным, востребованным решением. Он позволит пользователям лучше организовывать свои информационные ресурсы, повышать продуктивность и упрощать работу с данными. В конечном итоге это приведёт к улучшению их опыта и удовлетворённости от использования сервиса, что обеспечит стабильную и активную аудиторию.

3 Проектирование и разработка веб-сервиса

3.1 Требования к функциональным характеристикам

Веб-сервис должен предоставлять пользователю следующие функции:

* возможность создания нового аккаунта с использованием имени, электронной почты и пароля. Должен иметься повторный ввод пароля для подтверждения. На почту должно отправляться письмо для подтверждения регистрации;
* возможность входа в систему с использованием электронной почты и пароля. Если у пользователя возникают проблемы с паролем, он должен иметь возможность воспользоваться функцией восстановления пароля. В таком случае он получает письмо на электронную почту с дальнейшими инструкциями к действиям;
* возможность выхода из системы. Таким образом обеспечивается безопасное завершение сеанса работы с веб-сервисом и предоствращение потери личных данных пользователем;
* показ архива со списком всех имеющихся записей с возможностями их просмотра, создания, редактирования и удаления; возможность задания записи таких характеристик как название, ссылка, описание, категория и статус прочтения(прочитано или не прочитано); наличие в архиве списка категорий, обеспечивающих сортировку записей по 3 стандартным (все, прочитанные, непрочитанные) или пользовательским категориям;
* показ архива со списком всех интеллектуальных карт с возможностями их просмотра, создания, редактирования и удаления;
* отображение страницы отдельной интеллектуальной карты, где показываются все добавленные на нее записи с установленными между ними связями, присутствуют возможности добавления на карту новых или удаления имеющихся на ней записей, редактирования связей между записями;
* возможность редактирования личных данных пользователя на странице профиля;
* возможность настраивать веб-сервис по личным предпочтениям – выбор внешнего вида интерфейса веб-сервиса, указание периода времени, через которое должны появляться напоминания каждого типа;
* вывод напоминаний на странице профиля: о том, что запись была добавлена некоторое время назад и все еще не прочитана, и о том, что запись прочитана некоторое время назад и для закрепления информации ее следует перечитать.

3.2 Требования к нефункциональным характеристикам

* веб-сервис должен хранить данные пользователей в безопасности. Необходимо постоянное обеспечение надежного хранения данных пользователей, в чем могут помочь современные технологии шифрования информации, использование безопасных протоколов передачи данных. Также, положительный эффект окажет проведение регулярных проверок сервиса на уязвимости и улучшение работы в соответствии с передовым опытом защиты информации;
* веб-сервис должен корректно выполнять заявленные функции. Все функции должны работать так, как от них ожидается, исключено неожиданное поведение компонентов веб-сервиса. Качественная работа всех функций – залог положительного опыта работы у пользователей;
* веб-сервис должен оптимально использовать ресурсы устройства пользователя. Он должен иметь минимальную нагрузку на устройство прользователя. Программный код должен быть оптимизирован, передаваться и загружаться должны только самые необходимые данные;
* в работе веб-сервиса не должно наблюдаться ошибок и сбоев. Сервис должен быть хорошо протестирован. Также, необходим постоянный мониторинг за текущим состоянием веб-сервиса в процессе работы. Это позволит быстро отреагировать на неполадки и исправить их;
* веб-сервис должен быть доступен для пользователей в любой момент времени. Ожидается, что доступ к веб-сервису будут иметь все пользователи в любое время суток, независимо от местоположения;
* пользователь должен иметь возможность получить доступ к веб-сервису при помощи веб-браузера, например, одного из представленных ниже.
  1. Яндекс Браузер;
  2. Google Chrome;
  3. Microsoft Edge;
  4. Mozilla Firefox;
  5. Safari.

3.3 Варианты использования разрабатываемого веб-сервиса

Далее, на рисунке номер, представлена диаграмма вариантов использования веб-сервиса.

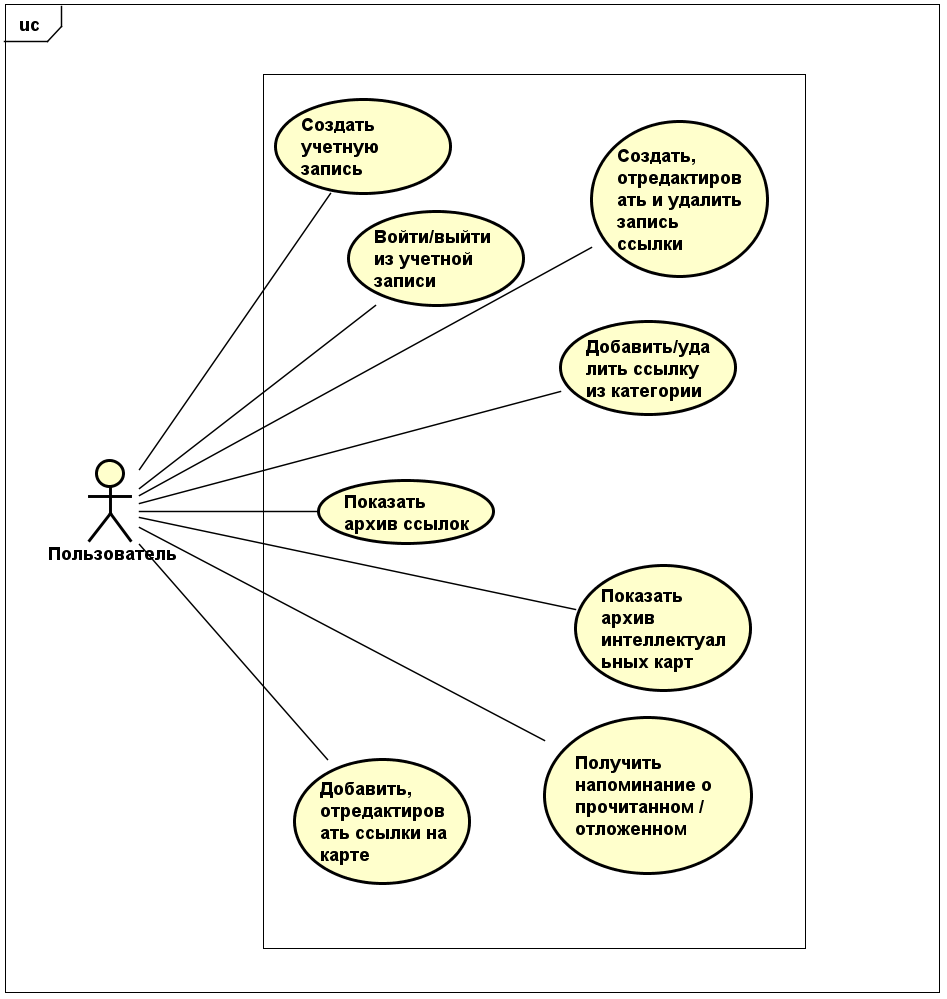


Рисунок номер – Диаграмма вариантов использования веб-сервиса

Далее приведены сценарии использования веб-сервиса.

Сценарий процесса регистрации пользователя:

* пользователь переходит на страницу регистрации;
* пользователь заполняет необходимые поля своими данными;
* пользователь подтверждает регистрацию;
* пользователь попадает на страницу профиля.

Сценарий процесса входа пользователя:

* пользователь переходит на страницу входа;
* пользователь заполняет необходимые поля своими данными;
* пользователь подтверждает вход;
* пользователь попадает на страницу профиля.

Сценарий добавления новой записи в архив:

* пользователь переходит на страницу архива записей;
* пользователь нажимает кнопку добавления записи;
* пользователь заполняет необходимые поля данными записи;
* пользователь подтверждает добавление записи.

Сценарий редактирования данных записи:

* пользователь переходит на страницу архива записей;
* пользователь нажимает кнопку редактирования соответствующей записи;
* пользователь корректно изменяет данные записи;
* пользователь подтверждает изменения.

Сценарий удаления записи из архива:

* пользователь переходит на страницу архива записей;
* пользователь нажимает кнопку удаления на соответствующей записи.

Сценарий добавления категории:

* пользователь переходит на страницу архива записей;
* пользователь нажимает кнопку добавления категории;
* пользователь вводит название категории;
* пользователь подтверждает добавление категории.

Сценарий добавления записи на интеллектуальную карту:

* пользователь переходит на страницу архива интеллектуальных карт;
* пользователь выбирает нужную карту и заходит на ее страницу;
* пользователь выбирает опцию добавления записи;
* пользователь выбирает запись из появившегося окна архива записей и она добавляется на карту.

3.4 Проектирование веб-сервиса

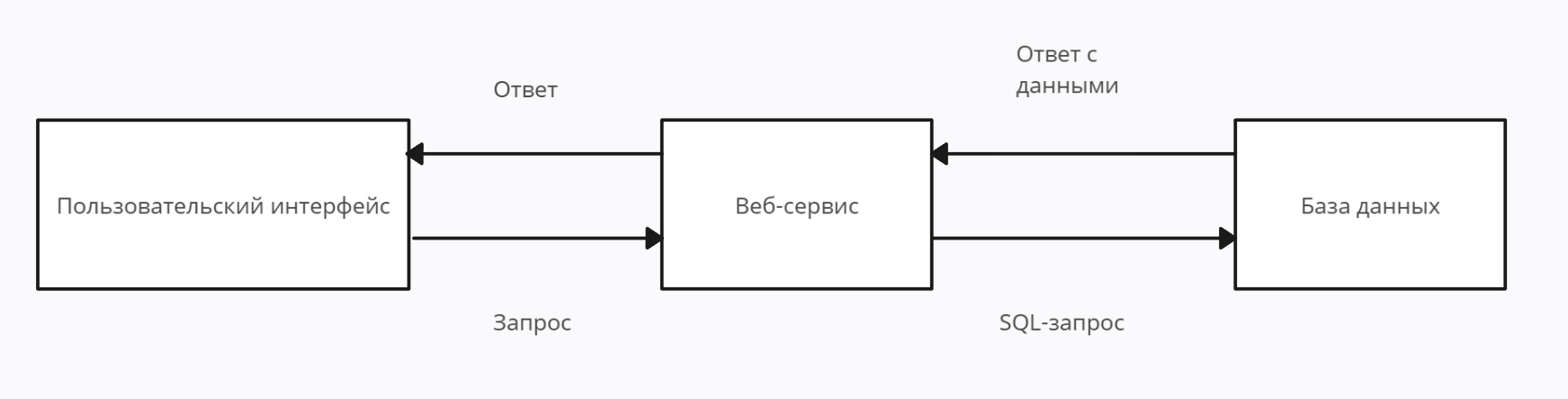


Рисунок номер – Устройство веб-сервиса

Пользовательский интерфейс отображает данные, хранимые в базе данных и обеспечивает удобный доступ к ним. С этой частью непосредственно взаимодействует пользователь.

Формирование данных для отображения и подготовка запросов для базы данных обеспечивается веб-сервисом. Он выполняет работу по получению и обработке запросов пользователя, запрашивает и получает информацию из базы данных, обрабатывает и возвращает ее для вывода на пользовательский интерфейс.

При начале работы с веб-сервисом пользователь должен авторизоваться или зарегистрироваться и, при успешном прохождении этого этапа, веб-сервис получает уникальный токен сессии, использующийся для взаимодействия с сервером.

Пользователь взаимодействует с веб-сервисом через интерфейс – управляет профилем, работает с записями и интеллектуальными картами. При отправке пользователем запроса на изменение данных веб-сервис проверяет корректность данных. В случае, если были обнаружены ошибки, он выдает сообщение о них. Если с данными все в порядке, он посылает серверу один из следующих типов запросов:

* PUT-запрос для редактирования;
* POST-запрос для создания;
* GET-запрос для получения;
* DELETE-запрос для удаления данных.

После этого, если запрос корректен, в базу данных отправляется SQL-запрос. Если работа с данными прошла успешно, база данных возвращает ответ формата HTTP, сообщая серверу о результате операции над данными.

3.5 Проектирование базы данных

Далее, на рисунке номер, приведена схема базы данных с таблицами и связями между ними.

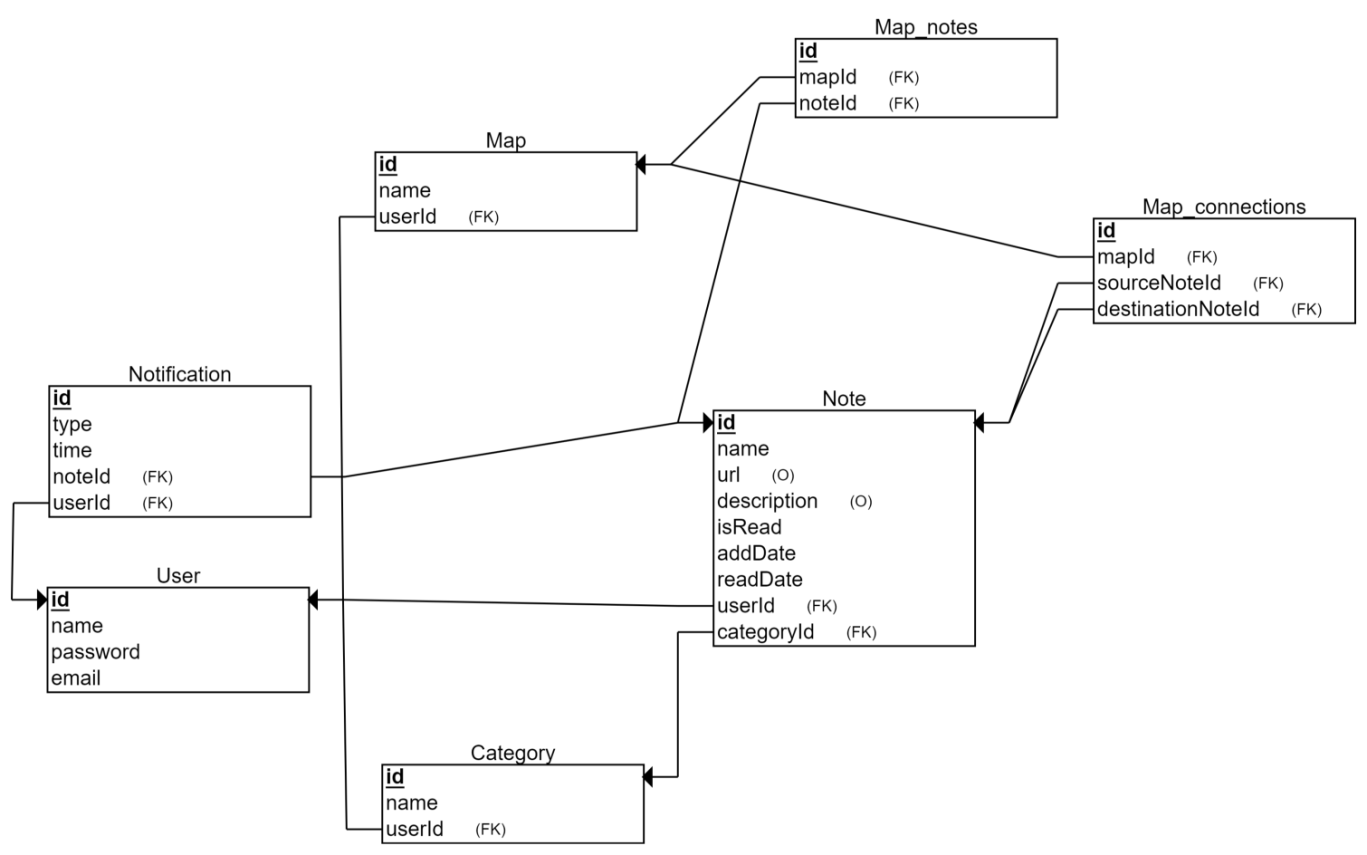


Рисунок номер – схема базы данных

На схеме базы данных представлены различные таблицы, хранящие следующую информацию:

* пользователь: хранит информацию об имени, электронной почте и пароле пользователя. Таблица необходима для идентификации и аутентификации пользователей в системе;
* запись: хранит информацию о названии, ссылке на интернет-ресурс, описании, категории, статусе прочитано или отложено. Также указывается дата добавления записи и дата ее прочтения (если запись отмечена как прочитанная, иначе – пустое значение) для создания напоминаний. Запись хранит id пользователя, которому принадлежит;
* категория: хранит информацию о своем названии категории, а также id пользователя, которому принадлежит;
* интеллектуальная карта: хранит информацию о своем названии, а также id пользователя, которому принадлежит;
* напоминание: хранит значение своего типа – либо это напоминание о давно добавленной и непрочитанной записи, либо напоминание о давно прочитанной записи, чтение которой стоит повторить. Напоминание хранит id пользователя, которому принадлежит.
* для размещения записей и связей между ними на интеллектуальной карте имеются вспомогательные таблицы.
  1. Карта-записи: хранит информацию о том, какие записи на каких интеллектуальных картах расположены.
  2. Записи-связи: хранит информацию о том, между какими записями существует связь на интеллектуальной карте.

3.6 Проектирование классов

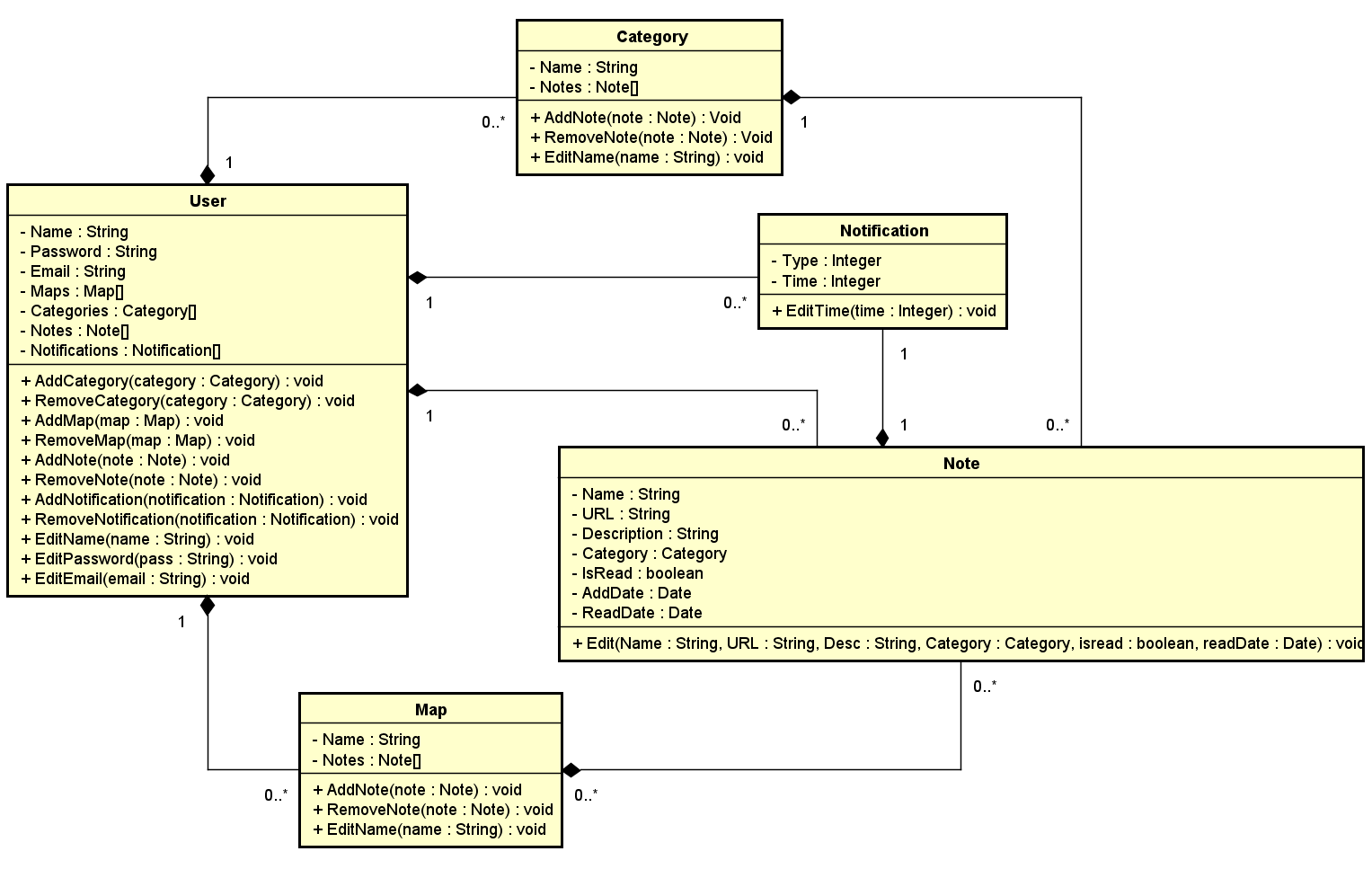


Рисунок номер – Диаграмма классов веб-сервиса

Веб-сервис имеет следующие классы:

* пользователь: содержит информацию об имени, электронной почте, пароле; имеет методы добавления и удаления записей, интеллектуальных карт, категорий, напоминаний, методы изменения имени, почты и пароля;
* категория: содержит информацию о названии и соответствующих записях; имеет методы изменения названия, добавления и удаления записей;
* напоминание: содержит информацию о своем типе и времени, через которое оно должно появиться; имеет метод изменения времени;
* запись: содержит информацию о названии, url-адресе, описании, категории, дате добавления и прочтения, статусе (прочитано или отложено); имеет метод изменения вышеперечисленных характеристик за исключением даты добавления;
* интеллектуальная карта: содержит информацию о названии; имеет методы добавления и удаления записей, изменения названия.

3.7 Разработка веб-сервиса

Архитектурой веб-сервиса была выбрана REST API, поскольку она обеспечивает удобное разделение на клиентскую и серверную часть, делая взаимодействие между ними удобным и гибким.

Данная архитектура позволяет обеспечить высокую производительность работы и облегчить разработку сервиса. Это достигается за счет использования стандартных методов HTTP и наличия множества удобных инструментов благодаря широкому распространению и популярности данного подхода.

При разработке был использован фреймворк Node.js, который обладает широкой базой библиотек и инструментов для разработки серверных приложений. Он также включает в себя популярный веб-сервер Express.js, который был использован при разработке данного проекта. Использование Node.js позволило создать эффективный и быстродействующий веб-сервис с возможностью масштабирования под высокие нагрузки.

Фреймворк Node.js и входящий в него веб-сервер Express.js были выбраны по причине их популярности и наличия исчерпывающей документации. Это делает написание и отладку кода значительно проще, поскольку необходимая информация по решению проблем и вопросов легко доступна.

Подобный подход к разработке веб-сервиса обеспечивает ему не только высокую производительность и гибкость взаимодействия с пользовательской частью, но и также упрощает поддержку и развитие проекта в долгосрочной перспективе.

3.7.1 Взаимодействие с базой данных

При разработке веб-сервисов с использованием Node.js разработчикам часто требуется инструмент для эфеективного, удобного и быстрого взаимодействия с базой данных. При этом средство должно в равной степени обеспечивать производительность и надежность программы. Популярным выбором являются системы ORM (Object Relationship Management), упрощающие работу с базами данных.

В данном случае была выбрана Prisma ORM. СУБД, выбранной для работы, стала MySQL. Prisma ORM прекрасно работает с базами данных MySQL, обеспечивая гибкость и безопасность при написании запросов. Данный инструмент позволяет создать все необходимые модели с помощью кода и не обращаться к СУБД для внесения изменений в структуру базы данных.

3.7.2 Процедура аутентификации

Аутентификация веб-сервиса реализована при помощи JWT-токенов. Это объясняется тем, что данная технология предоставляет криптографические алгоритмы. Они позволяют держать данные пользователей в безопасности, предотвращая несанкционированный доступ и кражу личной информации.

С поддержкой таких токенов появляется возможность упростить разрабатываемую систему, так как нет необходимости хранить информацию о сессии пользователя на стороне сервера. JWT-токены являются гибким инструментом, дающим возможность хранения дополнительных данных и создания собственных функций.

3.8 Используемые языки разработки и программные средства

Работа по созданию веб-сервиса была проведена в среде разработки Visual Studio Code. Для работы с базой данных использовалась среда разработки MySQL Workbench 8.0 CE.

Основным языком разработки был выбран TypeScript, что облегчает поиск и предотвращение ошибок, упрощает чтение и работу с кодом. Также в разработке использован React, предоставляющий мощные инструменты для быстрого и гибкого создания пользовательских интерфейсов. Вместе TypeScript и React позволяют создать надёжное и удобное в поддержке приложение.

Для разработки была также использована библиотека React Mindmap для создания интеллектуальных карт в виде реакт-компонента.

Взаимодействие с базой данных обеспечивается при помощи Prisma ORM, которая предоставляет собой удобный и простой способ манипулирования данными благодаря декларативному подходу и автоматической генерации кода.

За серверную часть отвечает веб-фреймворк для Node.js – Express.js. Это эффективный инструмент, предоставляющий такие возможности как управление маршрутизацией, middleware, шаблонизацией и т.д.

Для контроля версий разрабатываемого веб-сервиса использовалась система распределенного контроля версий Git и соответствующий ей веб-сервис GitHub.

Выводы

Данная глава играет ключевую роль в работе над веб-сервисом по управлению ссылками на интернет-ресурсы.

В первую очередь были описаны функциональные и нефункциональные требования. Структурируя план работы таким образом можно учесть все необходимые аспекты проекта, четко обозначить границы разработки и выявить наиболее важные составляющие технической части веб-сервиса.

Описывается также и структура веб-сервиса, для этого используются наглядные схемы. Это важно для лучшего понимания архитектуры системы разработчиками, что уменьшает число ошибок или случаев, когда необходимо переделать уже готовую часть проекта. Были визуализированы общая структура разрабатываемого продукта в виде схемы взаимодействия между компонентами, а также схема базы данных.

В главе объясняется процесс реализации приложения, включая взаимодействие с базой данных и серверной частью. Это помогает разобраться как веб-сервис будет функционировать на практике и впоследствии выбрать технологии и методы организации связи между данными компонентами.

В завершение, были описаны инструментальные и программные средства, с помощью которых проводилась работа над реализацией веб-сервиса. Сюда включено программное обеспечение, языки разработки и вспомогательные ресурсы.

Таким образом, данная глава помогает обеспечить полноту исследования процесса разработки и спланировать всю работу над технической составляющей веб-сервиса.

# 4 Тестирование веб-сервиса

# 4.1 Ручное тестирование веб-сервиса

Для тестирования функциональности сервиса были созданы тестовые сценарии, охватывающие все возможности взаимодействия пользователя с веб-сервисом. А именно: взаимодействие с архивом записей и непосредственно записями, архивом интеллектуальных карт и непосредственно картами, профилем. В сценариях описываются действия тестировщиков и результат, который должен быть получен при корректной работе веб-сервиса.

Далее приводятся тест-кейсы, согласно которым должно быть проведено ручное тестирование веб-сервиса.

Тест-кейс №1. Пользователь корректно регистрируется в системе веб-сервиса.

Действия тестировщика:

1) Нажать на кнопку Login

2) Перейти по ссылке для регистрации

3) Ввести корректные имя, почту и пароль

4) Нажать на кнопку для регистрации

Ожидаемый результат: Перед пользователем появляется страница профиля.

Тест-кейс №2. Пользователь некорректно регистрируется в системе веб-сервиса, заполнив не все поля.

Действия тестировщика:

1) Нажать на кнопку Login

2) Перейти по ссылке для регистрации

3) Заполнить не все поля данных

4) Нажать на кнопку для регистрации

Ожидаемый результат: Перед пользователем появляется уведомление о том, что все поля необходимо заполнить.

Тест-кейс №3. Пользователь некорректно регистрируется в системе веб-сервиса, введя имя длиной более 50 символов или менее 3 символов.

Действия тестировщика:

1) Нажать на кнопку Login

2) Перейти по ссылке для регистрации

3) Заполнить все поля данных, введя имя длиной менее 3 символов или более 50 символов

4) Нажать на кнопку для регистрации

Ожидаемый результат: Перед пользователем появляется уведомление о том, что нужно ввести корректное имя.

Тест-кейс №4. Пользователь некорректно регистрируется в системе веб-сервиса, введя некорректный email.

Действия тестировщика:

1) Нажать на кнопку Login

2) Перейти по ссылке для регистрации

3) Заполнить все поля данных, введя некорректный email

4) Нажать на кнопку для регистрации

Ожидаемый результат: Перед пользователем появляется уведомление о том, что нужно ввести корректный email.

Тест-кейс №5. Пользователь некорректно регистрируется в системе веб-сервиса, введя некорректный пароль.

Действия тестировщика:

1) Нажать на кнопку Login

2) Перейти по ссылке для регистрации

3) Заполнить все поля данных, введя некорректный пароль

4) Нажать на кнопку для регистрации

Ожидаемый результат: Перед пользователем появляется уведомление о том, что нужно ввести корректный пароль.

Тест-кейс №6. Пользователь корректно входит в систему веб-сервиса.

Действия тестировщика:

1) Нажать на кнопку Login

2) Заполнить все поля корректными данными

3) Нажать на кнопку для входа

Ожидаемый результат: Перед пользователем появляется страница его профиля.

Тест-кейс №7. Пользователь некорректно входит в систему веб-сервиса, заполнив не все поля.

Действия тестировщика:

1) Нажать на кнопку Login

2) Заполнить не все поля данных

3) Нажать на кнопку для входа

Ожидаемый результат: Перед пользователем появляется уведомление о том, что все поля необходимо заполнить.

Тест-кейс №8. Пользователь пытается войти в систему веб-сервиса незарегистрировавшись.

Действия тестировщика:

1) Нажать на кнопку Login

2) Заполнить все поля, введя незарегистрированный email

3) Нажать на кнопку для входа

Ожидаемый результат: Перед пользователем появляется уведомление о том, что такого пользователя не существует.

Тест-кейс №9. Пользователь пытается войти в систему веб-сервиса с неправильным паролем.

Действия тестировщика:

1) Нажать на кнопку Login

2) Заполнить все поля, введя зарегистрированный email и несоответствующий ему пароль

3) Нажать на кнопку для входа

Ожидаемый результат: Перед пользователем появляется уведомление о том, что пароль не корректный.

Тест-кейс №10. Пользователь выходит из системы веб-сервиса.

Действия тестировщика:

1) Нажать на кнопку Profile

2) Нажать на кнопку Log out

Ожидаемый результат: Перед пользователем появляется главная страница сайта и происходит выход из аккаунта

Тест-кейс №11. Пользователь пытается переключиться на страницы веб-сервиса, кроме главной, не войдя в систему.

Действия тестировщика:

1) Нажать на кнопку Archive или Maps в меню сайта, не войдя в систему

Ожидаемый результат: Перед пользователем появляется страница с надписью "Please login."

Тест-кейс №12. Пользователь корректно изменяет имя профиля.

Действия тестировщика:

1) Нажать на кнопку Profile

2) Нажать на кнопку редактирования рядом с именем пользователя

3) Ввести новое корректное имя

4) Нажать на кнопку подтверждения

Ожидаемый результат: Имя пользователя изменено.

Тест-кейс №13. Пользователь некорректно изменяет имя профиля.

Действия тестировщика:

1) Нажать на кнопку Profile

2) Нажать на кнопку редактирования рядом с именем пользователя

3) Ввести новое некорректное имя (длиной менее 3 символов или более 50 символов)

4) Нажать на кнопку подтверждения

Ожидаемый результат: Имя пользователя не изменено.

Тест-кейс №14. Пользователь корректно добавляет новую запись в архив

Действия тестировщика:

1) Нажать кнопку Archive в меню

2) Нажать на кнопку добавления рядом с названием блока "Notes"

3) Ввести корректные данные – непустое имя длиной не более 60 символов, url длиной до 2048 символов и описание длиной до 2000 символов

4) Нажать на кнопку Save

Ожидаемый результат: Пользователь возвращается к списку записей. Новая запись добавлена в архив.

Тест-кейс №15. Пользователь отменяет добавление новой записи в архив

Действия тестировщика:

1) Нажать кнопку Archive в меню

2) Нажать на кнопку добавления рядом с названием блока - "Notes"

3) Ввести корректные данные – непустое имя длиной не более 60 символов, url длиной до 2048 символов и описание длиной до 2000 символов

4) Нажать на кнопку Back

Ожидаемый результат: Пользователь возвращается к списку записей. Новая запись не добавлена в архив.

Тест-кейс №16. Пользователь некорректно добавляет новую запись в архив

Действия тестировщика:

1) Нажать кнопку Archive в меню

2) Нажать на кнопку добавления рядом с названием блока "Notes"

3) Ввести некорректные данные – пустое имя или имя длиной более 60 символов, url длиной более 2048 символов или описание длиной более 2000 символов

4) Нажать на кнопку Save

Ожидаемый результат: Пользователь получает уведомление о том, что длина текста в соответствующем поле выходит за допустимые границы. Запись не добавлена.

Тест-кейс №17. Пользователь удаляет запись из архива.

Действия тестировщика:

1) Нажать кнопку Archive в меню

2) Нажать на кнопку удаления на соответствующей записи

Ожидаемый результат: Запись удалена из архива.

Тест-кейс №18. Пользователь корректно редактирует запись из архива.

Действия тестировщика:

1) Нажать кнопку Archive в меню

2) Нажать на кнопку редактирования на соответствующей записи

3) Ввести корректные данные – непустое имя длиной не более 60 символов, url длиной до 2048 символов и описание длиной до 2000 символов

4) Нажать на кнопку Save

Ожидаемый результат: Пользователь возвращается к списку записей. Запись изменена.

Тест-кейс №19. Пользователь некорректно редактирует запись из архива.

Действия тестировщика:

1) Нажать кнопку Archive в меню

2) Нажать на кнопку редактирования на соответствующей записи

3) Ввести некорректные данные – пустое имя или имя длиной более 60 символов, url длиной более 2048 символов или описание длиной более 2000 символов

4) Нажать на кнопку Save

Ожидаемый результат: Пользователь получает уведомление о том, что длина текста в соответствующем поле выходит за допустимые границы. Запись не изменена.

Тест-кейс №20. Пользователь корректно добавляет новую категорию.

Действия тестировщика:

1) Нажать кнопку Archive в меню

2) Нажать на кнопку добавления рядом с названием блока "Categories"

3) Ввести корректное название категории – не пустое и длиной не более 60 символов

4) Нажать на кнопку подтверждения

Ожидаемый результат: Новая категория добавлена.

Тест-кейс №21. Пользователь пытается добавить новую категорию без названия.

Действия тестировщика:

1) Нажать кнопку Archive в меню

2) Нажать на кнопку добавления рядом с названием блока "Categories"

3) Оставить название категории пустым

4) Нажать на кнопку подтверждения

Ожидаемый результат: Новая категория не добавлена.

Тест-кейс №22. Пользователь пытается добавить новую категорию со слишком длинным названием.

Действия тестировщика:

1) Нажать кнопку Archive в меню

2) Нажать на кнопку добавления рядом с названием блока "Categories"

3) Ввести название длиной более 60 символов

4) Нажать на кнопку подтверждения

Ожидаемый результат: Пользователь получает уведомление о том, что длина названия категории выходит за допустимые границы. Новая категория не добавлена.

Тест-кейс №23. Пользователь корректно добавляет новую интеллектуальную карту.

Действия тестировщика:

1) Нажать кнопку Maps в меню

2) Нажать на кнопку добавления рядом с названием блока "Maps"

3) Ввести непустое название длиной не более 60 символов

4) Нажать на кнопку подтверждения

Ожидаемый результат: Пользователь возвращается к списку карт. Новая карта добавлена в архив.

Тест-кейс №24. Пользователь некорректно добавляет новую интеллектуальную карту.

Действия тестировщика:

1) Нажать кнопку Maps в меню

2) Нажать на кнопку добавления рядом с названием блока "Maps"

3) Ввести пустое название или название длиной более 60 символов

4) Нажать на кнопку подтверждения

Ожидаемый результат: Пользователь возвращается к списку карт. Новая карта не добавлена в архив.

Тест-кейс №25. Пользователь корректно редактирует данные интеллектуальной карты.

Действия тестировщика:

1) Нажать кнопку Maps в меню

2) Нажать на кнопку редактирования на соответствующей карте

3) Ввести корректные данные - непустое название длиной не более 60 символов

4) Нажать на кнопку подтверждения

Ожидаемый результат: Пользователь возвращается к списку записей. Данные карты изменены.

Тест-кейс №26. Пользователь некорректно редактирует данные интеллектуальной карты.

Действия тестировщика:

1) Нажать кнопку Maps в меню

2) Нажать на кнопку редактирования на соответствующей карте

3) Ввести некорректные данные - пустое название или название длиной более 60 символов

4) Нажать на кнопку подтверждения

Ожидаемый результат: Пользователь получает уведомление о том, что длина названия карты выходит за допустимые границы. Данные карты не изменены.

Тест-кейс №27. Пользователь отменяет добавление новой интеллектуальной карты.

Действия тестировщика:

1) Нажать кнопку Maps в меню

2) Нажать на кнопку добавления рядом с названием блока "Maps"

3) Ввести непустое название длиной не более 60 символов

4) Нажать на кнопку Back

Ожидаемый результат: Пользователь возвращается к списку записей. Новая карта не добавлена в архив.

Тест-кейс №28. Пользователь добавляет на интеллектуальную карту новую запись.

Действия тестировщика:

1) Нажать кнопку Maps в меню

2) Открыть существующую карту или создать новую и открыть ее

3) Нажать на кнопку добавления записи на карту

4) Выбрать запись из архива

Ожидаемый результат: Выбранная запись добавлена на карту.

Тест-кейс №29. Пользователь добавляет связь между записями на интеллектуальной карте.

Действия тестировщика:

1) Нажать кнопку Maps в меню

2) Открыть существующую карту или создать новую и открыть ее

3) Нажать на кнопку добавления связи между записями

4) Выбрать 2 связываемые записи

Ожидаемый результат: Связь между выбранными записями добавлена.

# 4.2 Юзабилити-тестирование веб-сервиса

Юзабилити-тестирование — это метод оценки веб-сервиса или приложения, в ходе которого реальным пользователям предлагается выполнить определённые задачи. Целью такого тестирования является выявление проблем с удобством использования интерфейса, а также сбор данных о впечатлениях пользователей. Это помогает разработчикам понять, насколько легко и эффективно пользователи могут взаимодействовать с продуктом, и внести необходимые улучшения.

Список вопросов к пользователям веб-сервиса:

* были ли у вас какие-либо затруднения при регистрации или входе?
* быстро ли вы разобрались в устройстве веб-сервиса?
* были ли случаи, когда вы не могли найти нужную информацию?
* легко ли было добавлять и редактировать записи ссылок?
* были ли у вас затруднения с сортировкой ссылок по категориям?
* было ли удобно работать с интеллектуальными картами?
* были ли полезны напоминания о прочитанных и отложенных записях?
* удобно ли вам было пользоваться сервисом в целом?
* есть ли у вас пожелания для улучшения сервиса?

На рисунке номер приведены результаты опроса по юзабили-тестированию.



Рисунок номер – Результаты опроса юзабилити-тестирования, часть 1



Рисунок номер – Результаты опроса юзабилити-тестирования, часть 2

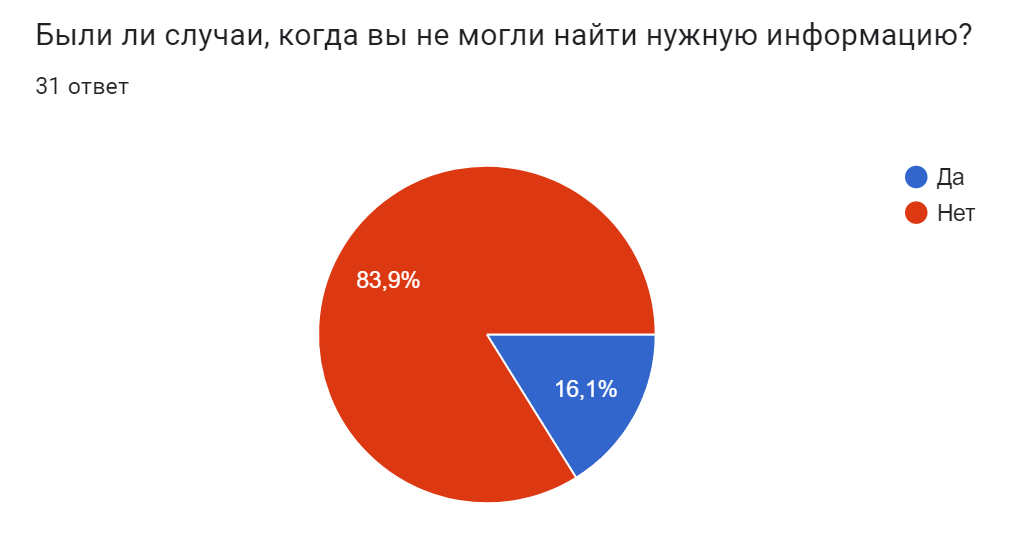


Рисунок номер – Результаты опроса юзабилити-тестирования, часть 3



Рисунок номер – Результаты опроса юзабилити-тестирования, часть 4



Рисунок номер – Результаты опроса юзабилити-тестирования, часть 5



Рисунок номер – Результаты опроса юзабилити-тестирования, часть 6



Рисунок номер – Результаты опроса юзабилити-тестирования, часть 7



Рисунок номер – Результаты опроса юзабилити-тестирования, часть 8

Как видно по диаграммам, большинство пользователей довольны веб-сервисом, не испытывали затруднений и оценивают работу с ним положительно.

Тем не менее, присутствуют некоторые пожелания по улучшению работы, которые пользователи также оставили в соответствующем поле опроса:

* улучшить навигацию по архиву веб-ссылок добавив поиск;
* добавить всплывающие подсказки для пользователей, которые испытывают трудности;
* добавить более гибкую настройку напоминаний о прочитанных или отложенных записях.

Эти отзывы и предложения помогут направить дальнейшие улучшения веб-сервиса и сделать его более удобным и функциональным для пользователей. Реагирование на желания пользователей является одной из важнейших задач при разработке и поддержке веб-сервиса, ведь это помогает ему оставаться практичным, актуальным и конкурентноспособным.

4.3 Пример работы программы

На рисунке номер приведены примеры работы страниц веб-сервиса.

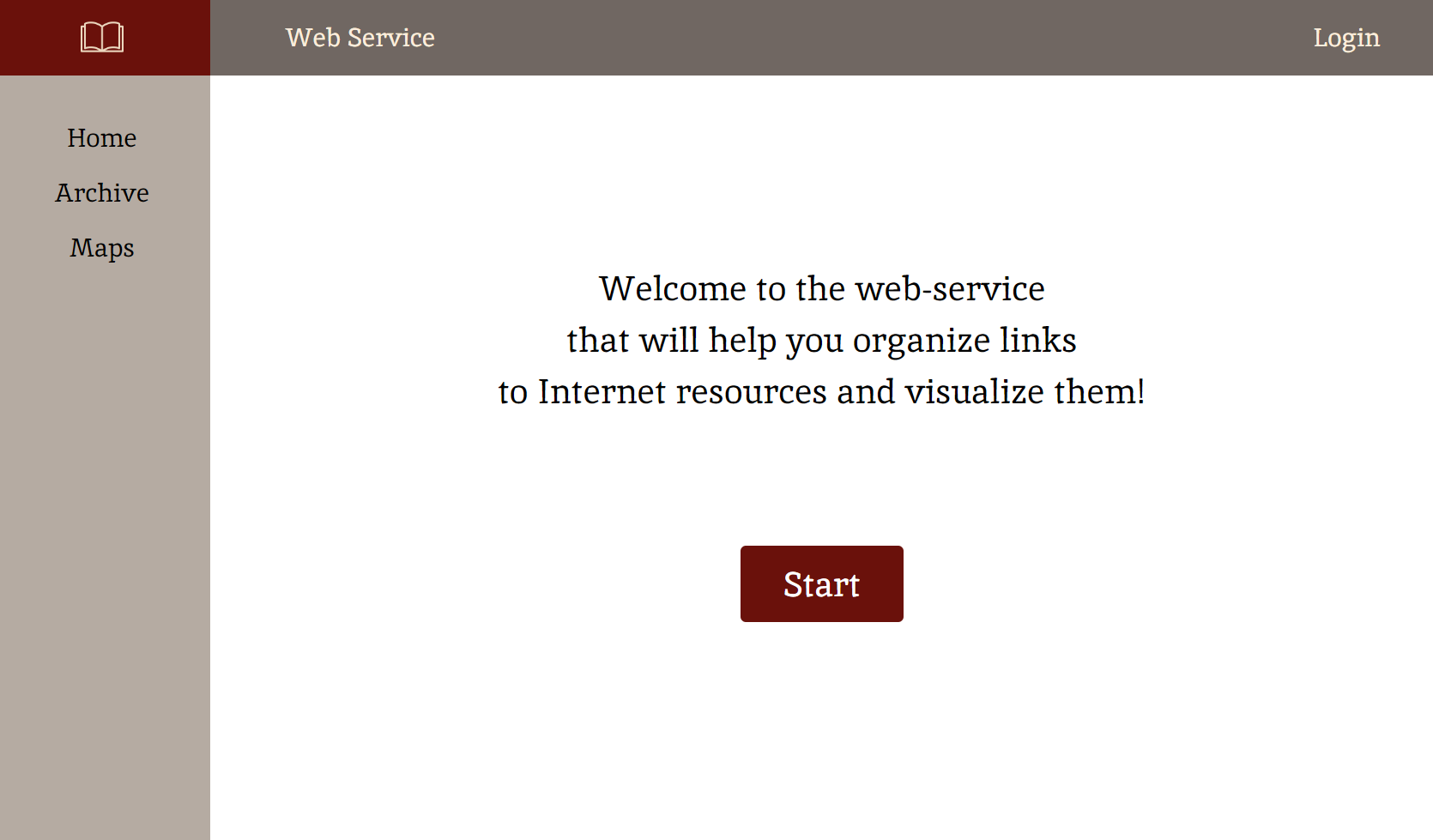


Рисунок номер – Пример работы главной страницы

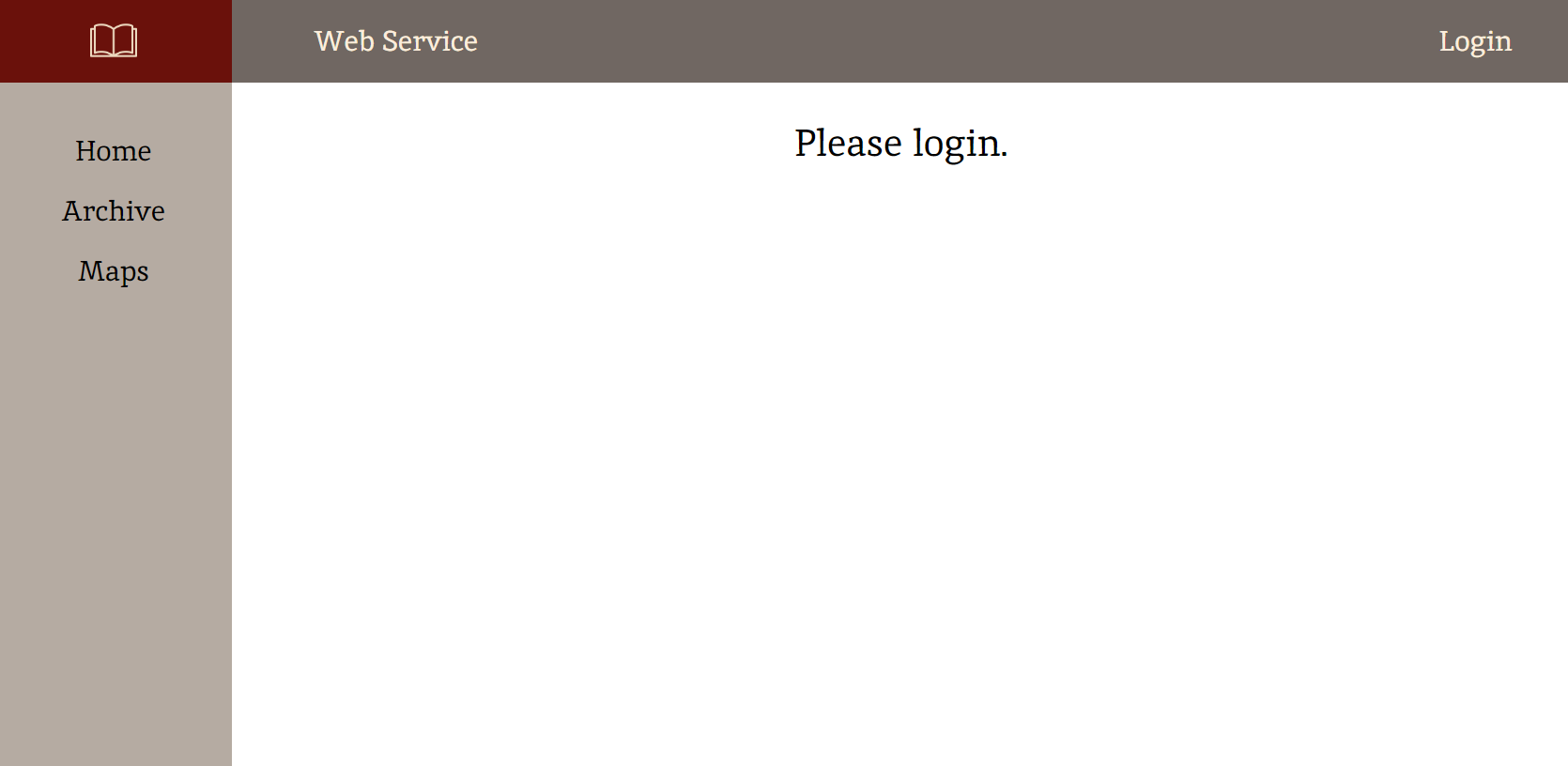


Рисунок номер – Пример работы страницы архива записей, профиля или интеллектуальных карт, если пользователь не авторизовался

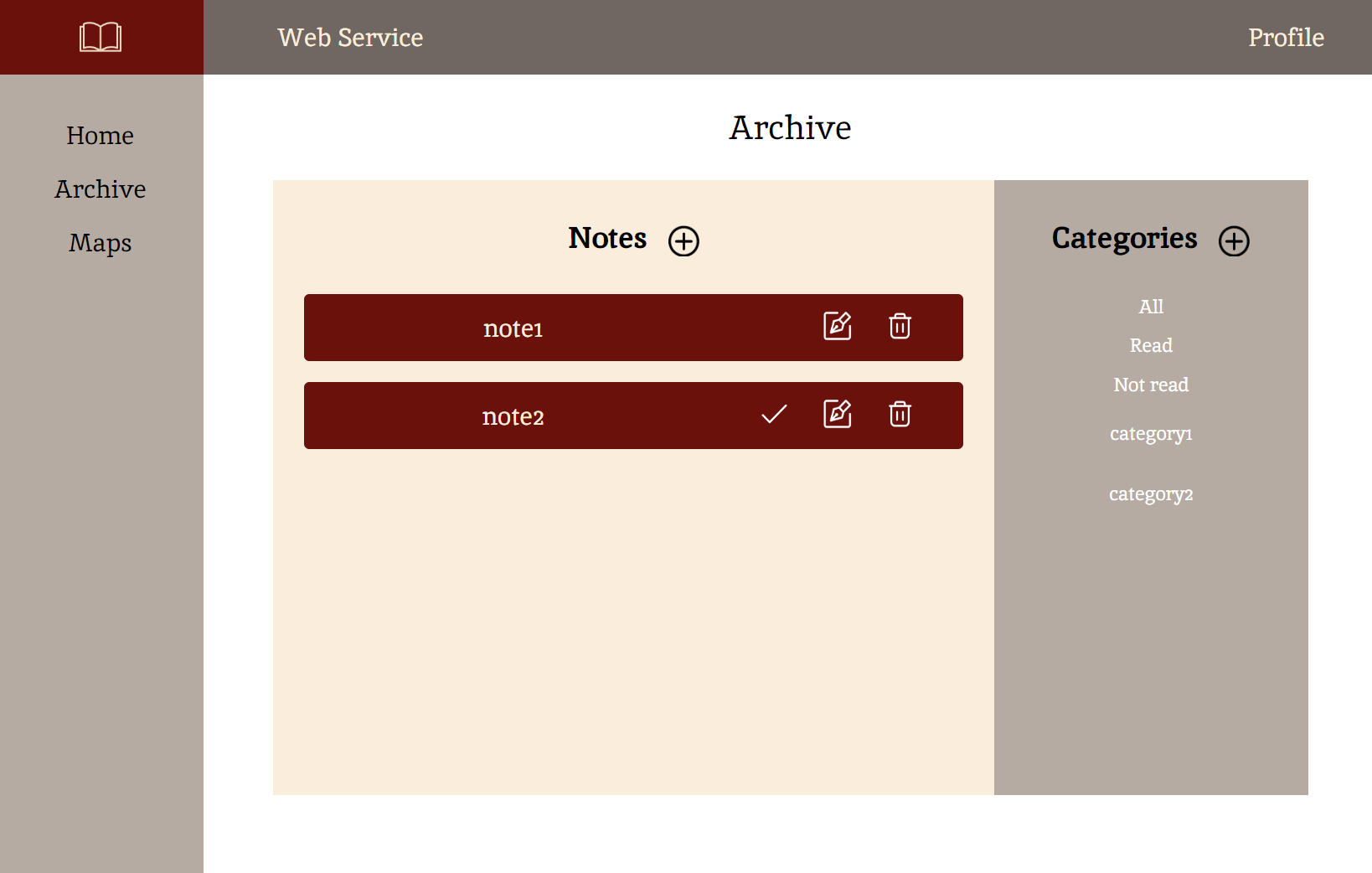


Рисунок номер – Пример работы страницы архива записей

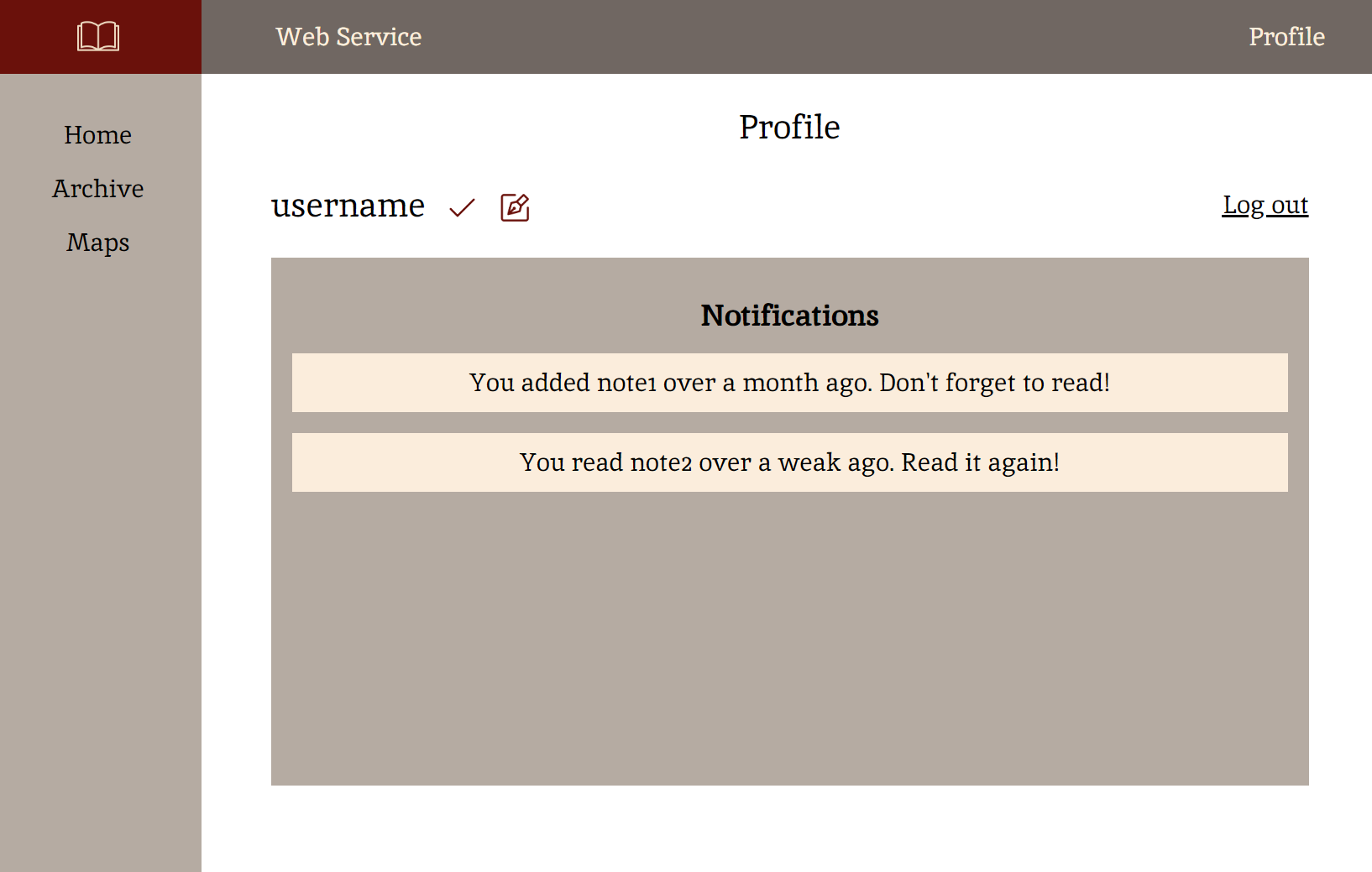


Рисунок номер – Пример работы страницы профиля

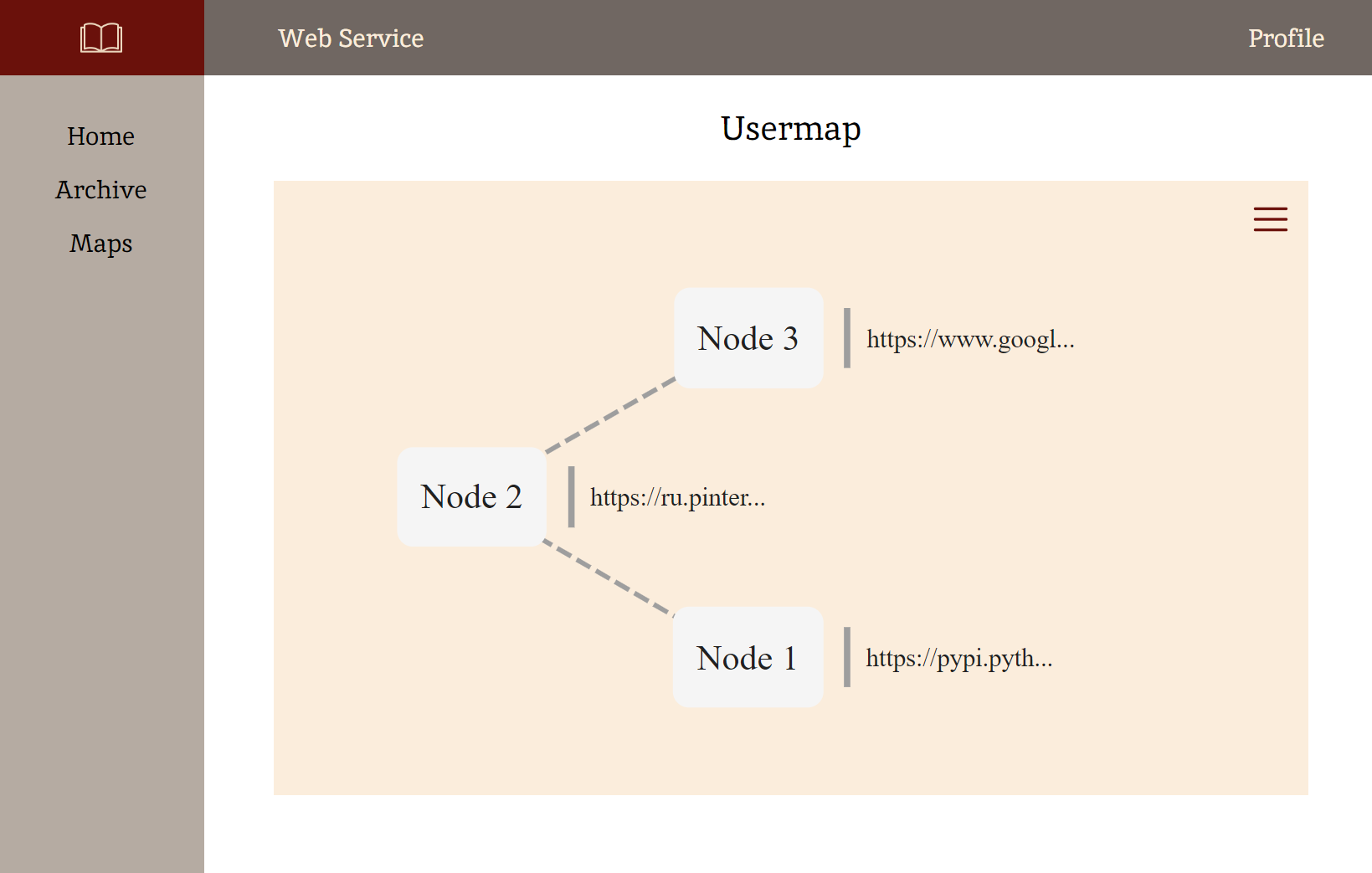


Рисунок номер – Пример работы страницы интеллектуальной карты

Выводы

В данном разделе были разобраны и описаны ручное тестирование и юзабилити-тестирование. Эти этапы проверки качества веб-сервиса являются критически важными для обеспечения его надежности и удобства использования.

Был создан список ручных тестов с подробным описанием действий тестировщика и ожидаемым результатом. Этот подход позволил выявить и устранить возможные ошибки и несоответствия на ранних этапах разработки. В результате ручного тестирования веб-сервис показал корректную и стабильную работу, что подтверждает его техническую состоятельность и готовность к использованию.

Контроль юзабилити-тестирования проводился при помощи опроса пользователей, в котором они должны были ответить на несколько вопросов, оценивающих различные аспекты интерфейса. Также пользователи могли оставить пожелания по улучшению веб-сервиса в свободной форме. Большинство пользователей остались довольны предоставленным интерфейсом, отметив его простоту и интуитивность.

Также в данной главе были приведены примеры работы веб-сервиса в виде изображений. Они демонстрируют дизайн пользовательского интерфейса и корректную работу ключевых функций сервиса.

Результаты ручного тестирования и юзабилити-тестирования показали, что разработанный веб-сервис является надежным и удобным инструментом для управления интернет-ссылками. Созданный веб-сервис нравится пользователям и оставляет положительные впечатления от работы.

Таким образом, итоги тестирования подтверждают правильность и эффективность выбранного подхода к разработке продукта и помогают дальнейшему его совершенствованию в соответствии с пожеланиями опрошенных пользователей.

# Заключение

В рамках выполнения данной работы была успешно достигнута заявленная ранее цель – организация эффективного управления ссылками на интернет-ресурсы. В ходе разработки веб-сервиса решены такие задачи:

* был проведен анализ предметной области, который позволил нам лучше понять потребности пользователей и особенности работы с интернет-ссылками;
* сделан обзор существующих аналогов, выявлены их основные преимущества и недостатки, что позволило учесть имеющийся опыт и избежать типичных ошибок;
* требования к разрабатываемому веб-сервису были определены на основе потребностей пользователей и анализа аналогов, что позволило получить представление о конечном продукте;
* база данных была спроектирована согласно заявленным требованиям, что позволило создать соответствующую структуру хранения данных и обеспечить работу с ними;
* разработан веб-сервис, предоставляющий функционал по организации, хранению и быстрому доступу к Интернет-ссылкам;
* разработанный веб-сервис прошел тестирование, результаты которых подтвердили его работоспособность и результативность.

Таким образом, был создан веб-сервис, отвечающий заявленным требованиям. Данный сервис обладает всеми необходимыми функциями для повышения производительности работы пользователя и упрощения доступа к информации, что подтверждает актуальность его создания и практическую значимость. В дальнейшем на основе полученных данных и пользовательской обратной связи можно будет расширить функционал веб-сервиса, что еще больше повысит его удобство и эффективность.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. М

# Приложение А

# Справка о результатах проверки выпускной квалификационной работы на наличие заимствований

# Приложение Б

# Техническое задание

# Приложение В

# Руководство системного программиста