

## Содержание

Введение .....	3
1 Анализ предметной области .....	6
2 Проектирование приложения .....	8
3 Разработка программного обеспечения .....	11
3.1 Описание технологического стека разработки .....	11
3.2 Описание алгоритма работы .....	13
3.3 Описание интерфейса пользователя .....	14
4 Тестирование приложения .....	16
4.1 План тестирования .....	16
4.2 Оценка результатов проведения тестирования .....	17
Заключение .....	19
Список использованных источников .....	21

					ОКЭИ 09.02.07 7024			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.	Эргашева Э.Э.				Отчет УП			
Провер.	Гикасян А. Д.							
Реценз								
Н. Контр.								
Утверд.								
						Лит.	Лист	Листов
							2	31
						Отделение информационных технологий гр. 4661		

## Введение

Веб-приложение – это программа или набор программ, которые выполняются на сервере и доступны пользователям через браузер. Веб-приложения обычно работают с данными, хранящимися на сервере, и могут предоставлять различные функции, такие как обработка информации, управление данными, взаимодействие с пользователем и т.д. Одним из преимуществ веб-приложений является то, что они доступны с любого устройства, имеющего доступ в интернет.

Веб-приложения могут выполнять различные функции, в зависимости от их назначения. Некоторые из наиболее распространенных функций включают:

- обработка информации: веб-приложения могут использоваться для сбора, обработки и анализа данных. например, интернет-магазины используют веб-приложения для обработки информации о заказах, платежах и доставке;
- управление данными: веб-приложения также могут использоваться для управления и хранения данных. например, системы управления контентом (cms) используют веб-приложения для хранения и редактирования контента на сайте;
- взаимодействие с пользователем: веб-приложения могут предоставлять пользователю различные возможности для взаимодействия, такие как формы обратной связи, чаты, форумы и т. д.;
- безопасность: веб-приложения должны обеспечивать безопасность данных пользователей и защиту от возможных угроз, таких как хакерские атаки.

Функции веб-приложения, которые должны быть реализованы в ходе данного проекта:

- регистрация и авторизация пользователей: эта функция позволит пользователям создавать аккаунт в вашем интернет-магазине и входить в него для совершения покупок; регистрация должна быть простой и быстрой, с возможностью использования различных видов учетных записей (например, через социальные сети);
- управление пользователями и ролями: функция управления пользователями и ролями позволит управлять аккаунтами пользователей, изменять их данные и роли в системе;
- отображение подробной информации о мастер-классах в виде модальных окон: для каждого мастер-класса должна отображаться подробная информация, такая как название, описание, время проведения, место проведения, стоимость ;
- запись на мастер-класс: пользователь должен иметь возможность записаться на выбранный мастер-класс, заполнив форму записи, указав свое имя и контактные данные;
- добавление мастер-класса в избранное: пользователи должны иметь возможность добавлять понравившиеся мастер-классы в избранное, чтобы иметь быстрый доступ к информации о них, а также следить за обновлениями и изменениями в расписании;

					ОКЭИ 09.02.07 7024 27 У	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

– добавление нового мастер-класса: администратор веб-приложения должен иметь возможность добавления новых мастер-классов, заполнив форму добавления мастер-класса, указав его название, описание, фотографии (если имеются), время и место проведения, а также стоимость участия.

Актуальность разработки веб-приложения записи на мастер-классы обусловлена рядом преимуществ, которые оно может предоставить как организаторам мастер-классов, так и участникам.

Для организаторов мастер-классов веб-приложение может обеспечить следующие преимущества:

– увеличение посещаемости мастер-классов: веб-приложение позволяет легко и быстро распространять информацию о предстоящих мастер-классах, что может привлечь больше участников;

– простота управления информацией: веб-приложение упрощает процесс управления информацией о мастер-классах и участниках, позволяя организаторам быстро обновлять данные и следить за статистикой.

Преимущества для участников мастер-классов включают:

– возможность записи на мастер-классы без необходимости звонить или отправлять сообщения организатору: участники могут просто заполнить форму записи на веб-приложении, чтобы зарезервировать место на мастер-классе.

Целью данной работы является создание веб-приложения для записи на мастер-классы.

Задачи:

– исследовать предметную область;

– провести анализ предприятия;

– составить техническое задание;

– проанализировать литературные и интернет-источники, посвященные аналогичным продуктам;

– разработать дизайн веб-приложения и его структуру;

– спроектировать веб-приложение;

– провести тестирование веб-приложения;

– реализовать веб-приложение.

Объектом является веб-приложение записи на мастер-классы.

Предметом является процесс создания веб-приложения записи на мастер-классы.

Для реализации проекта потребуются следующие ресурсы:

– фреймворк react для создания пользовательского интерфейса;

– node.js для работы серверного кода;

– postgresql для управления базой данных;

– express.js для создания веб-сервера;

– html и css;

– VS Code.

HTML и CSS являются основными технологиями для создания веб-страниц и управления их внешним видом. HTML используется для разметки и структурирования контента на веб-странице, а CSS - для определения стилей и

внешнего вида этой разметки. Вместе они позволяют создавать красивые и функциональные веб-страницы с текстом, изображениями, ссылками, таблицами и многим другим.

JavaScript позволяет добавлять различные элементы на веб-страницы, такие как анимация, всплывающие окна, формы обратной связи и многое другое. JavaScript также используется для работы с API, такими как геолокация, уведомления и обработка ошибок.

React: открытый JavaScript-фреймворк для создания пользовательских интерфейсов. Он был разработан компанией Facebook и стал одним из самых популярных инструментов для создания веб-приложений и мобильных приложений. React используется для создания интерактивных пользовательских интерфейсов, которые могут быстро обновляться и реагировать на действия пользователя. Он также позволяет создавать более быстрые и эффективные приложения, так как позволяет разбивать интерфейс на отдельные компоненты и управлять ими более эффективно.

PostgreSQL – это система управления базами данных (СУБД), которая используется для хранения и управления данными на сервере. Она поддерживает множество функций, таких как транзакции, индексы, триггеры и многое другое. PostgreSQL также является открытым исходным кодом, что означает, что его можно использовать бесплатно и изменять по своему усмотрению.

Node.js – это платформа для разработки серверных приложений на JavaScript. Она использует модель событий и позволяет обрабатывать множество запросов одновременно. Node.js используется для создания различных веб-приложений, таких как веб-сайты, API, мессенджеры и многое другое.

Express.js – это веб-фреймворк для Node.js, который используется для создания серверной части веб-приложений. Он предоставляет набор инструментов для обработки запросов, маршрутизации, обработки ошибок и многого другого. Express.js является простым в использовании и гибким, что делает его популярным выбором для многих разработчиков.

Visual Studio Code – это редактор кода с открытым исходным кодом, разработанный компанией Microsoft. Он имеет множество функций, которые упрощают процесс разработки, таких как подсветка синтаксиса, IntelliSense, отладка и поддержка множества языков программирования. Visual Studio Code также имеет множество плагинов, которые расширяют его функциональность.

# 1 Анализ предметной области

Веб-приложение – это программное обеспечение, которое запускается в веб-браузере. Веб-приложения позволяют пользователям выполнять различные задачи, такие как просмотр веб-страниц, отправка электронных писем, просмотр видео и многое другое. Веб-приложения являются важным инструментом для многих людей, так как они позволяют получать доступ к информации и сервисам в любое время и из любого места.

Мастер-классы – это обучающие мероприятия, на которых участники могут научиться новым навыкам или улучшить свои существующие навыки. Они могут быть организованы как для профессиональных целей, так и для личного развития.

Существует множество различных типов мастер-классов, таких как кулинарные мастер-классы, мастер-классы по рисованию, мастер-классы по танцам и многие другие.

Мастер-классы обычно проводятся опытными профессионалами или любителями, которые имеют большой опыт и знания в определенной области. Они могут проводить уроки как индивидуально, так и в группах, и часто используют различные методы обучения, такие как демонстрации, объяснения, практические задания и обратную связь.

Целью мастер-классов является обучение людей новым навыкам, развитие их творческих способностей и предоставление возможности для общения и обмена опытом. Они также могут быть хорошим способом для людей найти новые интересы и хобби, а также расширить свой круг общения.

В зависимости от темы мастер-класса, участники могут получить сертификаты или другие доказательства своего участия. Это может быть полезно для тех, кто хочет развиваться в определенной сфере или просто хочет иметь подтверждение своего опыта.

Одним из главных преимуществ мастер-классов является то, что они позволяют участникам получить новые знания и навыки в интересной и увлекательной форме. Кроме того, участие в мастер-классах может быть отличным способом для общения и обмена опытом с другими участниками.

Веб-приложение для записи на мастер-классы должно включать в себя следующие функции:

- регистрация и авторизация пользователей: веб-приложение должно предоставлять пользователям возможность зарегистрироваться с помощью своего email и пароля, а также возможность авторизации с использованием своих учетных данных;
- запись на мастер-класс: пользователи должны иметь возможность записаться на интересующий их мастер-класс, выбрав его из списка доступных мастер-классов и заполнив форму записи;
- добавление мастер-класса в избранное: функция добавления мастер-классов в избранное должна быть доступна пользователям для быстрого доступа к информации о любимых мастер-классах;

– добавление нового мастер-класса: администраторам веб-приложения должна быть предоставлена возможность добавлять новые мастер-классы с указанием всей необходимой информации, такой как название мастер-класса, описание, дата и время проведения, продолжительность, место проведения и стоимость участия.

Целевая аудитория веб-приложения для записи на мастер-классы включает в себя организаторов мастер-классов, участников мастер-классов и потенциальных участников мастер-классов. Организаторы используют приложение для управления информацией о своих мастер-классах и участниках, продажи билетов и отслеживания статистики. Участники используют приложение для поиска и записи на интересующие их мастер-классы, а также для получения уведомлений о новых мероприятиях и обновлениях. Потенциальные участники используют приложение для поиска и выбора мастер-классов на основе своих интересов и предпочтений. Возраст потребителей может варьироваться от 12 до 60 лет, с различным уровнем дохода и социальным статусом.

Конкуренты могут выступать аналогичные организации.

Веб-приложение для записи на мастер-классы имеет ряд преимуществ, таких как:

– удобство использования: веб-приложение обеспечивает простой и удобный интерфейс для поиска, записи и управления информацией о мастер-классах;

– широкий выбор мастер-классов: веб-приложение предлагает большой выбор мастер-классов различных тематик и уровней сложности;

– гибкость: веб-приложение позволяет администраторам обновлять информацию о мастер-классах;

– интеграция с другими сервисами: веб-приложение может быть интегрировано с другими сервисами, такими как платежные системы, системы управления обучением и т.д., что позволяет расширить его функциональность и удобство использования.

Недостатками такой системы являются:

– зависимость от интернет-подключения: веб-приложение требует постоянного подключения к интернету для своей работы, что может создавать проблемы для пользователей, особенно если они не имеют стабильного интернет-соединения;

– технические проблемы: веб-приложения иногда могут сталкиваться с техническими проблемами, такими как ошибки в работе приложения, проблемы с загрузкой данных или проблемы с безопасностью. Это может привести к неудобствам для пользователей и потере доверия к приложению.

В целом, веб-приложение для записи на мастер-классы является перспективным бизнесом, который может привлечь широкую аудиторию потребителей.

## 2 Проектирование приложения

Проектирование веб-приложения записи на мастер-классы можно разделить на следующие этапы:

- разработка дизайна веб-приложения;
- проектирование структуры веб-приложения (разделы, страницы);
- разработка функционала веб-приложения.

Для начала рассмотрим дизайн нашего веб-приложения (см. Приложение Г).

На главной странице сайта представлены следующие элементы:

- логотип;
- кнопка (иконка) перехода на страницу «Авторизация»;
- заголовок;
- карточка мастер-класса;
- модальное окно с подробной информацией о мастер-классе (при нажатии на картинку мастер-класса);
- модально окно с записью на мастер-класс (при нажатии на кнопку «Записаться»).

В модальном окне с подробной информацией о мастер-классе представлены следующие элементы:

- место проведения;
- адрес;
- цена;
- образовательная тематика;
- возрастное ограничение.

В модальном окне с записью на мастер-класс представлены следующие элементы:

- блок с формой ввода данных;
- кнопка «Записаться».

На главной странице сайта в разделе «footer» представлены следующие элементы:

- логотип.

На странице «Личный кабинет» представлены следующие элементы:

- header;
- блок с личной информацией пользователя;
- footer;
- кнопка «Выйти».

На странице «Избранное» представлены следующие элементы:

- header;
- заголовок;
- карточка мастер-класса, добавленная в «Избранное»;
- footer.

На странице «Мои мастер-классы» представлены следующие элементы:

					ОКЭИ 09.02.07 7024 27 У	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

- header;
- заголовок;
- карточка мастер-класса, на который записались;
- footer.

На странице сайта «Авторизация» представлены следующие элементы:

- блок с формой ввода данных;
- кнопка «Войти»;
- ссылка на страницу «Регистрация», если пользователь еще не зарегистрирован.

На странице сайта «Регистрация» представлены следующие элементы:

- блок с формой ввода данных;
- кнопка «Зарегистрироваться»;
- ссылка на страницу «Авторизация», если пользователь уже зарегистрирован.

Теперь перейдем к рассмотрению структуры веб-приложения.

Создание шаблонов страниц веб-приложения необходимо для:

- повышения производительности: шаблоны страниц веб-приложения позволяют быстрее создавать новые страницы, так как они могут использовать уже готовые блоки и элементы дизайна, что снижает затраты времени на разработку и улучшает общую производительность;
- повторное использование кода: шаблоны предоставляют возможность повторного использования кода, что уменьшает количество ошибок и упрощает процесс обновления веб-приложения;
- поддержка масштабируемости: с помощью шаблонов можно легко добавлять новые страницы и разделы в веб-приложение, не нарушая его структуру и дизайн, это обеспечивает большую масштабируемость и гибкость веб-приложения;
- удобный поиск по странице: за счет своей структурированности шаблоны упрощают поиск определенного элемента сайта;
- легкость обновления: если нужно внести изменения в дизайн или функционал веб-приложения, достаточно обновить определенный элемент, и все страницы с этим элементом будут обновлены.

Теперь рассмотрим общий шаблон нашего веб-приложения (см. рисунок 13).

```
<Header />
<div className="wrapper"> ...
</div>
<Footer />
</>
```

Рисунок 1 – Общая структура

Также в этом пункте следует рассмотреть такие общие элементы сайта, как шапка сайта (header) (см. Рисунок 14) и подвал сайта (footer) (см. Рисунок 15), которые используются для всех страниц сайта.



```
<div className="header">
  <Link to={'/afisha'} className='link_logo'><h1 className='logo'>master<span>Class</span></h1></Link>
  <Link to={'/reg'}><img className='profile_icon' src={icon_profile} alt="" /></Link>
</div>
```

Рисунок 2 – Шапка сайта (header)

```
<div className="footer">
<Link to={'/' } className='link_logo'><h1 className='logo'>master<span>Class</span></h1></Link>
</div>
```

Рисунок 3 – Подвал сайта (footer)

Теперь рассмотрим функциональные возможности нашего веб-приложения. На сайте представлены следующие функциональные возможности:

- регистрация и авторизация пользователей: эта функция позволит пользователям создавать аккаунт и входить в него для записи на мастер-классы. Регистрация должна быть простой и быстрой, с возможностью использования различных видов учетных записей (например, через социальные сети);

- управление пользователями и ролями: функция управления пользователями и ролями позволит управлять аккаунтами пользователей, изменять их данные и роли в системе;

- отображение подробной информации о мастер-классах в виде модальных окон: для каждого мастер-класса должна отображаться подробная информация, такая как название, описание, время проведения, место проведения, стоимость ;

- запись на мастер-класс: пользователь должен иметь возможность записаться на выбранный мастер-класс, заполнив форму записи, указав свое имя и контактные данные;

- добавление мастер-класса в избранное: пользователи должны иметь возможность добавлять понравившиеся мастер-классы в избранное, чтобы иметь быстрый доступ к информации о них, а также следить за обновлениями и изменениями в расписании;

- добавление нового мастер-класса: администратор веб-приложения должен иметь возможность добавления новых мастер-классов, заполнив форму добавления мастер-класса, указав его название, описание, фотографии (если имеются), время и место проведения, а также стоимость участия.

Для наглядности работы данных функций в приложении А на рисунке А.1 представлена «Диаграмма прецедентов».

## 3 Разработка программного обеспечения

### 3.1 Описание технологического стека разработки

В качестве технологического стека разработки веб-приложения был выбран Pern stack. Pern stack состоит из PostgreSQL, Express, React и Node. Помогает в создании полнофункционального веб-приложения.

React – это популярная JavaScript-библиотека для создания пользовательских интерфейсов (UI), разработанная компанией Facebook. Она позволяет создавать интерактивные и реактивные интерфейсы, которые могут обновляться в реальном времени при изменении данных или состоянии приложения.

Выбор React в качестве основного стека для разработки веб-приложения продиктован несколькими причинами:

- реактивность: react обеспечивает реактивный подход к созданию пользовательских интерфейсов, что позволяет приложениям обновлять свои элементы автоматически при изменении состояния приложения. Это делает процесс разработки более эффективным и удобным;

- производительность: react оптимизирован для производительности, что делает его идеальным выбором для больших и сложных приложений. Он также поддерживает асинхронное обновление состояния, что снижает нагрузку на браузер;

- компонентный подход: react использует концепцию компонентов, что позволяет разработчикам разбивать приложения на небольшие, независимые и повторно используемые блоки. Это ускоряет разработку и улучшает масштабируемость приложений;

- эффективное использование ресурсов: react позволяет управлять зависимостями между компонентами, что предотвращает перерисовку ненужных элементов и оптимизирует использование ресурсов;

- поддержка сообщества: react имеет активное и большое сообщество разработчиков, что обеспечивает доступ к большому количеству ресурсов, таких как библиотеки, инструменты и примеры кода.

Node.js - это среда выполнения JavaScript, которая позволяет выполнять JavaScript-код на сервере. Node.js использует модель событий, которая позволяет разрабатывать масштабируемые и высокопроизводительные приложения.

Преимущества Node.js:

- высокая производительность: node.js использует асинхронное выполнение кода, что позволяет снизить нагрузку на процессор и увеличить производительность приложения.

- масштабируемость: node.js позволяет легко масштабировать приложения, так как они могут быть распределены на множество серверов без потери производительности.

					ОКЭИ 09.02.07 7024 27 У	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

- веб-сокеты: node.js поддерживает веб-сокеты, что позволяет создавать приложения с постоянной связью между клиентом и сервером.
- большое сообщество: node.js имеет большое и активное сообщество разработчиков, которое помогает новым разработчикам быстрее освоиться в разработке на node.js.
- использование javascript: node.js позволяет использовать тот же язык программирования (javascript) на сервере, что и на клиенте, что упрощает обучение и разработку.

PostgreSQL - это объектно-реляционная система управления базами данных с открытым исходным кодом.

#### Преимущества PostgreSQL:

- открытость и доступность исходного кода: postgresql имеет открытый исходный код, что позволяет разработчикам создавать свои приложения и инструменты, а также изменять код для улучшения производительности или добавления новых функций;
- высокая производительность и масштабируемость: postgresql может обрабатывать большие объемы данных и масштабироваться для работы с еще большими объемами;
- гибкость в работе с данными: postgresql поддерживает множество типов данных и позволяет создавать сложные запросы с использованием языка запросов sql;
- поддержка транзакций: postgresql обеспечивает безопасную работу с данными и гарантирует их целостность благодаря поддержке транзакций;
- совместимость с другими субд: postgresql совместим с другими системами управления базами данных, что позволяет легко интегрировать его с существующими системами;
- поддержка различных платформ: postgresql работает на многих операционных системах, включая linux, macos и windows, что делает его доступным для широкого круга пользователей;
- безопасность: postgresql имеет встроенные механизмы безопасности, которые помогают защитить данные от несанкционированного доступа и других угроз.

Express – это минималистичный и гибкий веб-фреймворк для приложений Node.js, предоставляющий обширный набор функций для мобильных и веб-приложений.

Выбор Express, как фреймворка React, обусловлен следующими преимуществами:

- минимализм: express.js имеет минимальный набор функций, что упрощает ее использование и понимание;
- производительность: express.js оптимизирована для высокой производительности, она быстро обрабатывает запросы и отвечает на них;
- модульность: express.js построена на модульной архитектуре, что позволяет разработчикам выбирать только те функции, которые им нужны;

- поддержка сообщества: `express.js` имеет активное сообщество разработчиков и пользователей, которые предоставляют поддержку и советы по использованию фреймворка;
- гибкость: `express.js` предоставляет гибкость в настройке и расширении, что позволяет разработчикам адаптировать фреймворк под свои нужды.

### 3.2 Описание алгоритма работы

Веб-приложение – это программа, которая работает на компьютере пользователя через интернет. Она может быть написана на любом языке программирования и использовать различные технологии.

Алгоритм работы веб-приложения – это последовательность действий, которые выполняются при запуске приложения и при взаимодействии с пользователем. Алгоритм может включать в себя обработку ввода пользователя, работу с базой данных, отправку запросов на сервер и получение ответов, а также обработку ошибок и сбоев.

Принцип функционирования веб-приложения заключается в следующем:

- пользователь открывает веб-браузер и вводит адрес веб-сайта в адресной строке;
- веб-браузер отправляет запрос на сервер, где находится веб-приложение;
- сервер обрабатывает запрос и отправляет ответ обратно пользователю;
- веб-приложение отправляет `html`-код и другие ресурсы (например, `css`, `javascript`, изображения) на устройство пользователя;
- веб-браузер отображает полученную веб-страницу на экране пользователя;
- пользователь может взаимодействовать с веб-приложением, нажимая на кнопки, вводя информацию в формы и т.д.;
- веб-приложение обрабатывает полученные данные и отправляет их на сервер для дальнейшей обработки;
- сервер выполняет необходимые операции с полученными данными и отправляет ответ обратно веб-приложению;
- веб-приложение отображает результат на экране пользователя.

Теперь рассмотрим принцип функционирования нашего веб-приложения.

Веб-приложение для записи на мастер-классы использует базу данных PostgreSQL для хранения данных. PostgreSQL – это объектно-реляционная система управления базами данных с открытым исходным кодом. Она поддерживает множество языков программирования, включая JavaScript (через библиотеку `pg`). Ознакомиться с моделью базы данных можно в Приложении В.

При регистрации пользователя данные, такие как имя, адрес электронной почты и пароль, сохраняются в базе данных. При записи на мастер-класс вводимая информация также записывается в базу данных.

					ОКЭИ 09.02.07 7024 27 У	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

В приложении используются события, такие как клики на кнопки, выбор мастер-класса или изменение избранного. Эти события запускают различные функции JavaScript, которые обрабатывают данные действия и обновляют состояние приложения. Например, при клике на кнопку «Добавить в избранное» выполняется функция, которая добавляет выбранный товар в избранное и обновляет информацию об избранном в БД.

Для наглядного представления функционирования веб-приложения в Приложении Б на рисунке Б.1 представлена «Диаграмма деятельности».

### 3.3 Описание интерфейса пользователя

Интерфейс веб-приложения - это внешний вид и функциональность сайта или приложения, который пользователи видят и используют для взаимодействия с ним. Он включает в себя все элементы сайта, такие как кнопки, формы, изображения, тексты и другие компоненты, которые помогают пользователям выполнять задачи и достигать своих целей.

Принципы разработки интерфейса веб-приложения:

- простота использования: интерфейс должен быть простым и интуитивным, чтобы пользователь мог легко найти нужную информацию и выполнить необходимые действия;
- гибкость: интерфейс должен адаптироваться под различные устройства и разрешения экрана, чтобы пользователи чувствовали себя комфортно на любом устройстве;
- эффективность: интерфейс должен быть эффективным и быстрым, чтобы пользователи не тратили много времени на поиск нужной информации;
- безопасность: интерфейс должен обеспечивать защиту данных пользователя и безопасность его транзакций;
- доступность: интерфейс должен быть доступен для всех пользователей, включая людей с ограниченными возможностями;
- эстетика: интерфейс должен выглядеть привлекательно и соответствовать корпоративному стилю компании.

На рисунке 1 изображена схема взаимодействия пользователя с сайтом.

При входе на сайт пользователь попадает на главную страницу сайта, где может ознакомиться с представленными мастер-классами.

При нажатии на мастер-класс пользователю высвечивается модальное окно с подробной информацией о мастер-классе.

При нажатии на кнопку «Записаться» пользователю высвечивается модальное окно с записью на мастер-класс, где он должен ввести свои личные данные для записи.

При нажатии на кнопку (иконку) «Избранное» мастер-класс добавляется в «Избранное» (избранные мастер-классы представлены в личном кабинете пользователя в разделе «Избранное»). Мастер-классы, добавленные в «Избранное» записываются в базу данных.

При нажатии на кнопку (иконку) «Личный кабинет» пользователь переходит на страницу регистрации, в случае если пользователь уже зарегистрирован, то он может перейти на страницу авторизации по кнопке, в ином случае пользователь регистрируется на сайте.

После регистрации пользователь должен войти в систему, введя свои данные в форму авторизации.

После успешной авторизации в профиле пользователю выведутся его личные данные и кнопки для перехода на страницы «Избранное», «Мои мастер-классы». Если авторизовался администратор, то в профиле выведутся его личные данные и кнопка для добавления нового мастер-класса.

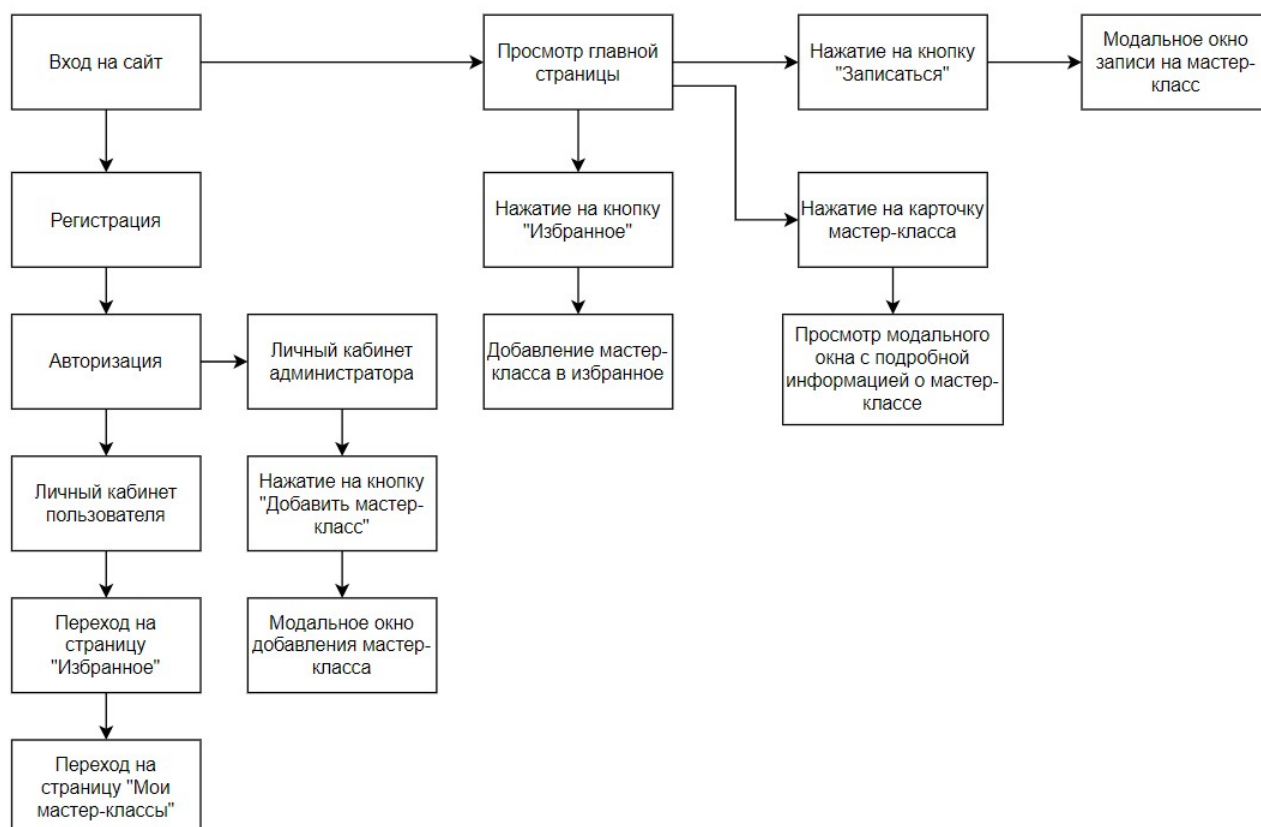


Рисунок 4 – Схема интерфейса

## 4 Тестирование приложения

### 4.1 План тестирования

Интеграционное тестирование - это процесс тестирования, который проверяет насколько хорошо различные компоненты системы работают вместе.

Цель интеграционного тестирования - убедиться, что компоненты системы взаимодействуют друг с другом корректно и не вызывают ошибок или сбоев в работе всей системы.

Некоторые задачи интеграционного тестирования:

- проверка взаимодействия модулей;
- обеспечение совместимости;
- раннее обнаружение проблем;
- повышение общей надёжности системы;
- повышение качества ПО за счёт выявления и устранения ошибок до того, как это станет более сложным и дорогостоящим процессом.

Преимущества интеграционного тестирования:

- позволяет выявить проблемы на более раннем этапе разработки, что снижает вероятность возникновения серьёзных ошибок в будущем;
- улучшает качество кода, так как позволяет обнаружить и исправить ошибки, которые могли бы остаться незамеченными при модульном тестировании;
- уменьшает количество регрессионных тестов, так как интеграционное тестирование проверяет взаимодействие между компонентами, которое может измениться при внесении изменений в один из компонентов.

Недостатки интеграционного тестирования:

- может занимать больше времени, чем модульное тестирование, так как требует тестирования всех комбинаций компонентов;
- может быть сложнее в организации и управлении, так как включает в себя больше людей и процессов;
- может требовать больше ресурсов, таких как время программистов и тестовых специалистов.

План проведения интеграционного тестирования:

Определение целей и задач интеграционного тестирования:

- проверить корректность работы всех функций и возможностей веб-приложения;
- определить эффективность взаимодействия между различными компонентами веб-приложения и его окружением;
- выявить возможные ошибки и недочёты в процессе интеграции отдельных частей приложения.

Выбор тестовых сценариев и методов:

- разработать тестовые сценарии, которые будут проверять все основные функции веб-приложения, а также его интеграцию с другими системами;

					ОКЭИ 09.02.07 7024 27 У	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

– выбрать подходящие методы тестирования, такие как функциональное тестирование, интеграционное тестирование, регрессионное тестирование и т. д.

Подготовка тестовой среды:

– создать тестовую среду, которая будет имитировать реальную среду функционирования веб-приложения (например, использовать виртуальные машины);

– настроить тестовую среду для проведения интеграционного тестирования, включая настройку серверов, баз данных, сетей и т.п.

Выполнение тестирования:

– провести интеграционное тестирование веб-приложения в соответствии с разработанными тестовыми сценариями;

– зафиксировать результаты тестирования, включая обнаруженные ошибки, проблемы и несоответствия.

Анализ результатов тестирования:

– проанализировать полученные результаты тестирования и определить, соответствуют ли они требованиям и ожиданиям;

– подготовить отчёт о проведённом интеграционном тестировании, указав в нем все обнаруженные проблемы и рекомендации по их устранению.

Устранение выявленных проблем:

– осуществить корректировку и исправление обнаруженных проблем и ошибок в веб-приложении;

– повторно провести интеграционное тестирование после исправления ошибок для подтверждения их устранения.

Завершение тестирования:

– завершить процесс интеграционного тестирования и оформить все необходимые документы, подтверждающие успешное завершение тестирования;

– передать веб-приложение в эксплуатацию после успешного завершения тестирования.

## 4.2 Оценка результатов проведения тестирования

После составления плана тестирования было проведено само интеграционное тестирование следующих функций:

– регистрация;

– авторизация;

– добавление мастер-классов в избранное;

– запись на мастер-класс;

– добавление нового мастер-класса (со стороны администратора).

Регистрация

При тестировании регистрации были подтверждены функции:

– регистрация нового пользователя происходит корректно;



- подтверждение регистрации создаёт новую учетную запись;
- если данные введены некорректно, система выдает ошибку и просит повторить ввод данных.

#### Авторизация

Тестирование авторизации прошло успешно. Были использованы различные учетные записи для входа в систему. В результате тестирования было подтверждено, что:

- учетные записи создаются при регистрации пользователей;
- пользователи могут успешно войти в систему, используя правильные учетные данные;
- некорректный ввод учетных данных вызывает сообщение об ошибке;
- после нескольких неудачных попыток входа учетная запись блокируется.

#### Добавление товаров в избранное:

- тестирование прошло успешно, товары были добавлены в избранное с использованием разных учетных записей;
- количество мастер-классов в избранном соответствовало количеству добавленных мастер-классов;
- мастер-классы были успешно удалены из избранного, и количество мастер-классов соответствовало новому состоянию избранного.

#### Запись на мастер-класс:

- данные успешно вводились в форму;
- данные были корректно отправлены на сервер;
- мастер-класс был успешно удален из списка моих мастер-классов.

#### Добавление нового мастер-класса (со стороны администратора):

- данные успешно вводились в форму;
- данные были корректно отправлены на сервер;
- новый мастер-класс успешно появился в списке общих курсов.

Интеграционное тестирование веб-приложения записи на мастер-классы было успешно проведено. Все функции, включая регистрацию и авторизацию пользователей, добавление товаров в избранное, запись на мастер-класс и добавление нового мастер-класса (со стороны администратора), работают корректно. Система правильно обрабатывает различные сценарии, такие как ввод некорректных учетных данных или добавление товаров в избранное. Все ошибки и исключения обрабатываются должным образом, и система предоставляет пользователю понятные сообщения об ошибках. Также было подтверждено, что корректно работает запись на мастер-класс и добавление нового мастер-класса (со стороны администратора). В целом, веб-приложение записи на мастер-классы функционирует должным образом и готово к запуску и использованию пользователями.

## Заключение

В заключении проекта по созданию веб-приложения для записи на мастер-классы, можно сделать вывод, что данный проект является актуальным и перспективным. Она предоставляет пользователям удобный интерфейс для поиска и записи на интересующие мастер-классы, а также позволяет администраторам управлять списком мастер-классов и участниками.

Разработанное приложение отличается высокой функциональностью, гибкостью настроек и использованием современных технологий.

В ходе выполнения данного проекта был спроектирован, реализован и внедрен программный продукт «Веб-приложение с интерактивной афишей для мастер-классов».

В процессе достижения вышеуказанной цели была исследована предметная область, а также проведен анализ целевой аудитории и конкурентов. Также были проанализированы литературные и интернет-источники, посвященные программным продуктам, аналогичным разрабатываемому, и сами аналогичные программные продукты, вследствие чего были сформулированы требования к разрабатываемому веб-приложению. Определены характеристики, которые делают веб-приложение понятным, удобным и функциональным:

- простой и понятный интерфейс: веб-приложение должно иметь простой и интуитивно понятный интерфейс, который позволяет пользователям легко находить нужные товары и оформлять заказы;
- быстрая загрузка страниц: веб-приложение должно быстро загружаться, чтобы пользователи не теряли терпение и не уходили на другие сайты;
- наличие поиска: веб-приложение должно иметь функцию поиска, чтобы пользователи могли быстро найти нужный товар;
- удобное меню навигации: веб-приложение должно иметь удобное меню навигации, которое позволяет пользователям быстро переходить на нужные страницы;
- наличие фильтров и сортировки товаров: веб-приложение должно предоставлять возможность фильтрации и сортировки товаров по различным параметрам, таким как цена, бренд, категория и т.д.;
- возможность добавления товаров в избранное или в список сравнения: пользователи должны иметь возможность добавлять товары в избранное или сравнивать их между собой;
- поддержка разных способов оплаты и доставки: веб-приложение должно поддерживать различные способы оплаты и доставки товаров, чтобы удовлетворить потребности разных пользователей;
- наличие отзывов и рейтингов товаров: пользователи часто обращают внимание на отзывы и рейтинги товаров перед покупкой, поэтому веб-приложение должно предоставлять эту информацию.

Функции, которые были реализованы в ходе данного проекта:

- авторизация;
- регистрация;
- запись на мастер-классы;
- добавление мастер-классов в избранное;
- добавление нового мастер-класса (со стороны администратора).

Сложности, с которыми мы столкнулись в ходе данного проекта:

- разработка адаптивного дизайна для мобильных устройств;
- интеграция с системами оплаты;
- функция поиска;
- организация системы отзывов и рейтингов товаров;
- обеспечение безопасности личных данных пользователей и защита от мошенничества.

Возможности для развития веб-приложения:

- улучшение пользовательского интерфейса: добавление новых функций, таких как возможность просмотра расписания мастер-классов или отзывов о них, а также улучшение дизайна и юзабилити;
- расширение списка мастер-классов: привлечение новых организаторов и увеличение количества предлагаемых занятий;
- внедрение системы лояльности: предоставление скидок или бонусов для постоянных пользователей, а также возможность накопления баллов за участие в мастер-классах;
- интеграция с социальными сетями: возможность регистрации и записи на мастер-классы через социальные сети, а также обмен отзывами и впечатлениями с другими пользователями;
- партнерства с другими организациями: сотрудничество с другими компаниями, предлагающими услуги в области образования или развлечений, для расширения аудитории приложения;
- анализ данных и обратная связь: сбор и анализ обратной связи от пользователей для улучшения качества работы приложения и предлагаемых мастер-классов;
- разработка мобильного приложения для удобства пользователей;
- оптимизация работы приложения для повышения скорости загрузки и улучшения пользовательского опыта.

После исследования предметной области и анализа информационных источников, были разработаны структура и дизайн сайта.

После проверки на соответствие требованиям и тестирования, сайт был внедрен для автоматизации бизнес-процессов.

Таким образом, следует считать, что задачи курсовой работы полностью выполнены и цель достигнута.

					ОКЭИ 09.02.07 7024 27 У	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

## Список использованных источников

- 1 Building a RESTful API with Express and PostgreSQL. - [Электронный ресурс]. - Режим доступа: <https://scotch.io/tutorials/getting-started-with-node-express-and-postgres-using-sequelize> (дата обращения: 15.12.2023)
- 2 React.js Tutorial - Full Course for Beginners. - [Электронный ресурс]. - Режим доступа: <https://www.freecodecamp.org/news/react-js-tutorial-for-beginners/> (дата обращения: 14.12.2023)
- 3 Node.js Tutorial - Learn Node.js in 1 Hour. - [Электронный ресурс]. - Режим доступа: [https://www.youtube.com/watch?v=TlB\\_eWDSMt4](https://www.youtube.com/watch?v=TlB_eWDSMt4) (дата обращения: 20.12.2023)
- 4 Единая система документации. - [Электронный ресурс]. - Режим доступа: <https://docs.cntd.ru/document/1200007627?ysclid=l4b3gsfkta201134956> (дата обращения: 22.12.2023)
- 5 Пользовательские сценарии. - [Электронный ресурс]. - Режим доступа: [https://pcnews.ru/top/articles/mat-drawer/polzovatelskie\\_scenarii\\_eto\\_eto\\_takoe\\_kak\\_i\\_dla\\_cego\\_ih\\_nuzno\\_stroit-753459.html?ysclid=l4b3qqrz5g980601158](https://pcnews.ru/top/articles/mat-drawer/polzovatelskie_scenarii_eto_eto_takoe_kak_i_dla_cego_ih_nuzno_stroit-753459.html?ysclid=l4b3qqrz5g980601158) (дата обращения: 9.12.2023)
- 6 Пользовательские сценарии. - [Электронный ресурс]. - Режим доступа: <https://netology.ru/blog/users-scenarios> (дата обращения: 10.12.2023)
- 7 Резервное копирование. - [Электронный ресурс]. - Режим доступа: <https://backupsolution.ru/backup-plan/?ysclid=l4b3itm4ha196679905> (дата обращения: 11.12.2023)
- 8 Информационная система. - [Электронный ресурс]. - Режим доступа: <https://docs.cntd.ru/document/0900006974?ysclid=l4b3lgvmgj187665433> (дата обращения: 16.12.2023)
- 9 Резервная копия сайта. - [Электронный ресурс]. - Режим доступа: <https://1zaicev.ru/kak-sozdat-rezervnuyu-kopiyu-sajta/?ysclid=l4igv9jj2h33094655> (дата обращения: 13.12.2023)
- 10 Средства автоматизации. - [Электронный ресурс]. - Режим доступа: <https://docs.cntd.ru/document/0900083083?ysclid=l4igwrig24723096681> (дата обращения: 20.12.2023)
- 11 Этапы и виды тестирования сайтов. - [Электронный ресурс]. - Режим доступа: <https://polyarix.com/blog/testirovanie-sajta/?ysclid=l4ih2vzzrt25295302> (дата обращения: 17.12.2023)
- 12 Руководство программиста. - [Электронный ресурс]. - Режим доступа: [https://studbooks.net/2236932/informatika/rukovodstvo\\_programmista?ysclid=l4ih6t76gp393362569](https://studbooks.net/2236932/informatika/rukovodstvo_programmista?ysclid=l4ih6t76gp393362569) (дата обращения: 1.12.2023)
- 13 Обратная связь на сайте. - [Электронный ресурс]. - Режим доступа: <https://www.imagecms.net/blog/obzory/zachem-nuzhna-obratnaia-sviaz-na-saite> (дата обращения: 10.12.2023)
- 14 Документация React (официальный сайт). - [Электронный ресурс]. - Режим доступа: <https://ru.reactjs.org/> (дата обращения: 11.12.2023)

15 Документация Node.js (официальный сайт). - [Электронный ресурс]. - Режим доступа: <https://nodejs.org/ru/docs/> (дата обращения: 16.12.2023)

16 Документация PostgreSQL (официальный сайт). - [Электронный ресурс]. - Режим доступа: <https://www.postgresql.org/docs/> (дата обращения: 13.12.2023)

17 Джойс Д. Безупречный JavaScript: идеи, стили и лучшие практики. - Санкт-Петербург: Питер, 2015.

18 Официальная документация Express.js. - [Электронный ресурс]. - Режим доступа: <https://expressjs.com/> (дата обращения: 17.12.2023)

19 Stack Overflow. - [Электронный ресурс]. - Режим доступа: <https://stackoverflow.com/> (дата обращения: 1.12.2023)

20 Tutorial v6.21.0 | React Router. - [Электронный ресурс]. - Режим доступа: <https://reactrouter.com/en/main/start/tutorial> (дата обращения: 10.12.2023)

					ОКЭИ 09.02.07 7024 27 У	Лист
						22
Изм.	Лист	№ докум.	Подпись	Дата		

## Приложение А (обязательное)

### Диаграмма прецедентов (use case)

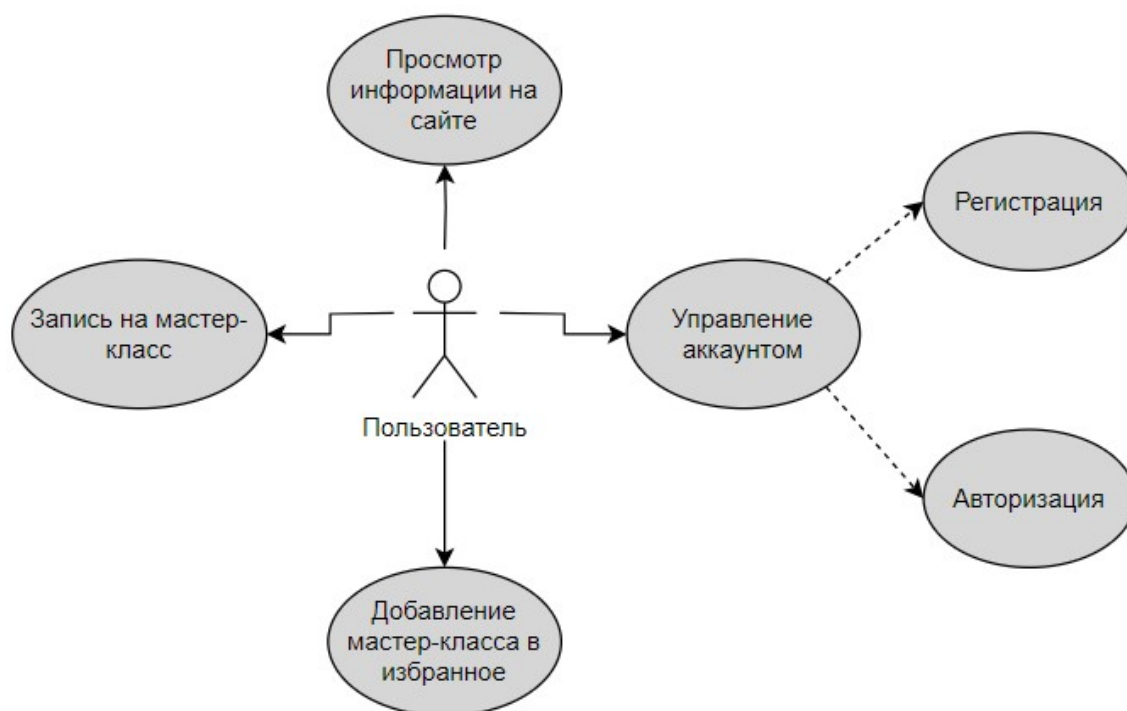


Рисунок А.1 – Диаграмма прецедентов

# Приложение Б (обязательное)

## Диаграмма деятельности

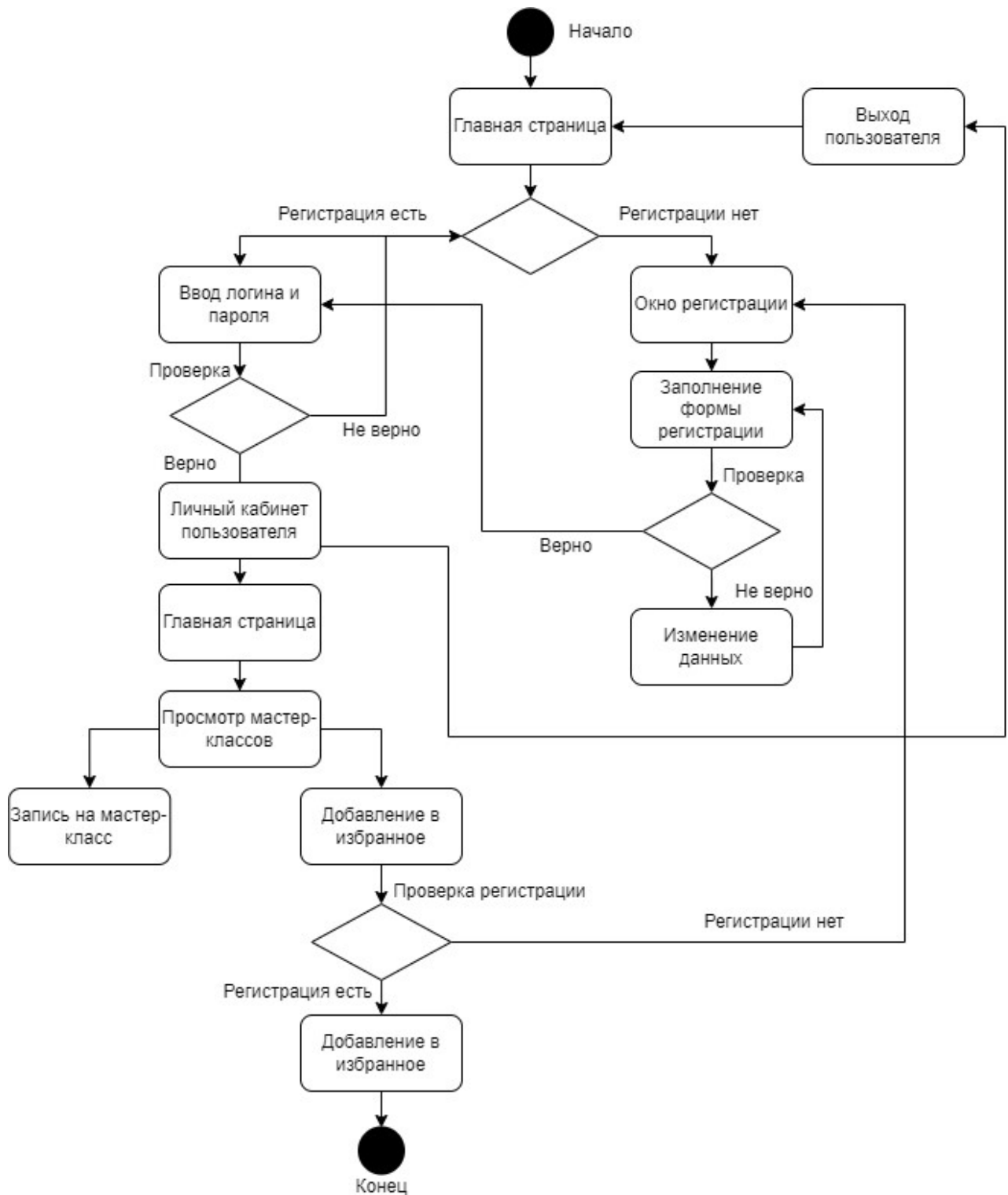


Рисунок Б.1 – Диаграмма деятельности

Приложение В  
(обязательное)

Диаграмма классов

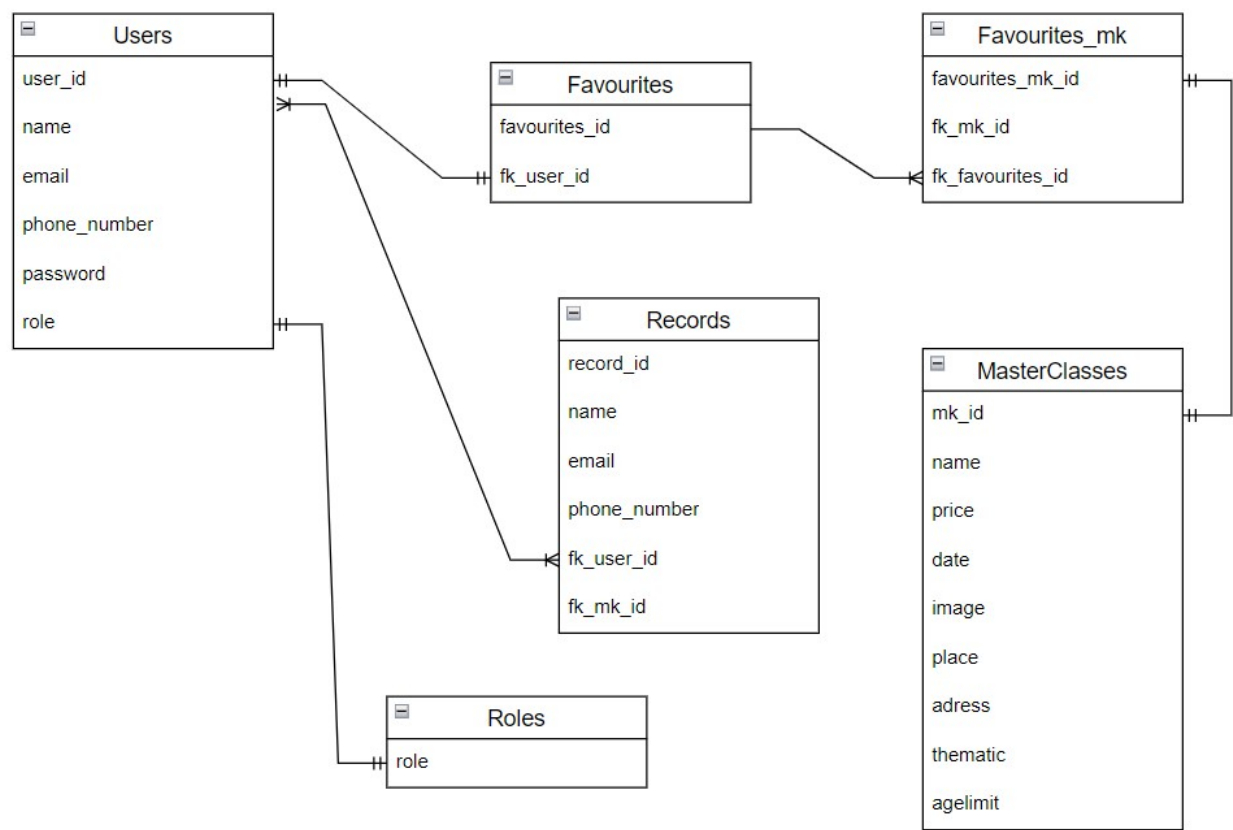


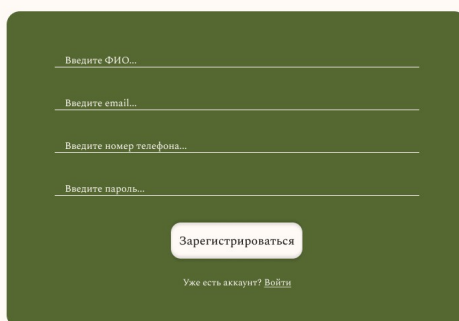
Рисунок В.1 – Диаграмма классов



# Приложение Г (обязательное)

## Дизайн веб-приложения

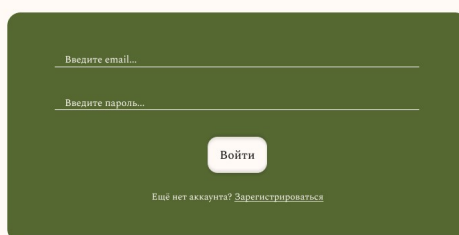
### Регистрация



Registration form design: A dark green rounded rectangle containing four input fields with placeholder text: 'Введите ФИО...', 'Введите email...', 'Введите номер телефона...', and 'Введите пароль...'. Below the fields is a white button with the text 'Зарегистрироваться'. At the bottom, there is a link: 'Уже есть аккаунт? Войти'.

Рисунок Г.1 – Страница «Регистрация»

### Авторизация



Authorization form design: A dark green rounded rectangle containing two input fields with placeholder text: 'Введите email...' and 'Введите пароль...'. Below the fields is a white button with the text 'Войти'. At the bottom, there is a link: 'Еще нет аккаунта? Зарегистрироваться'.

Рисунок Г.2 – Страница «Авторизация»

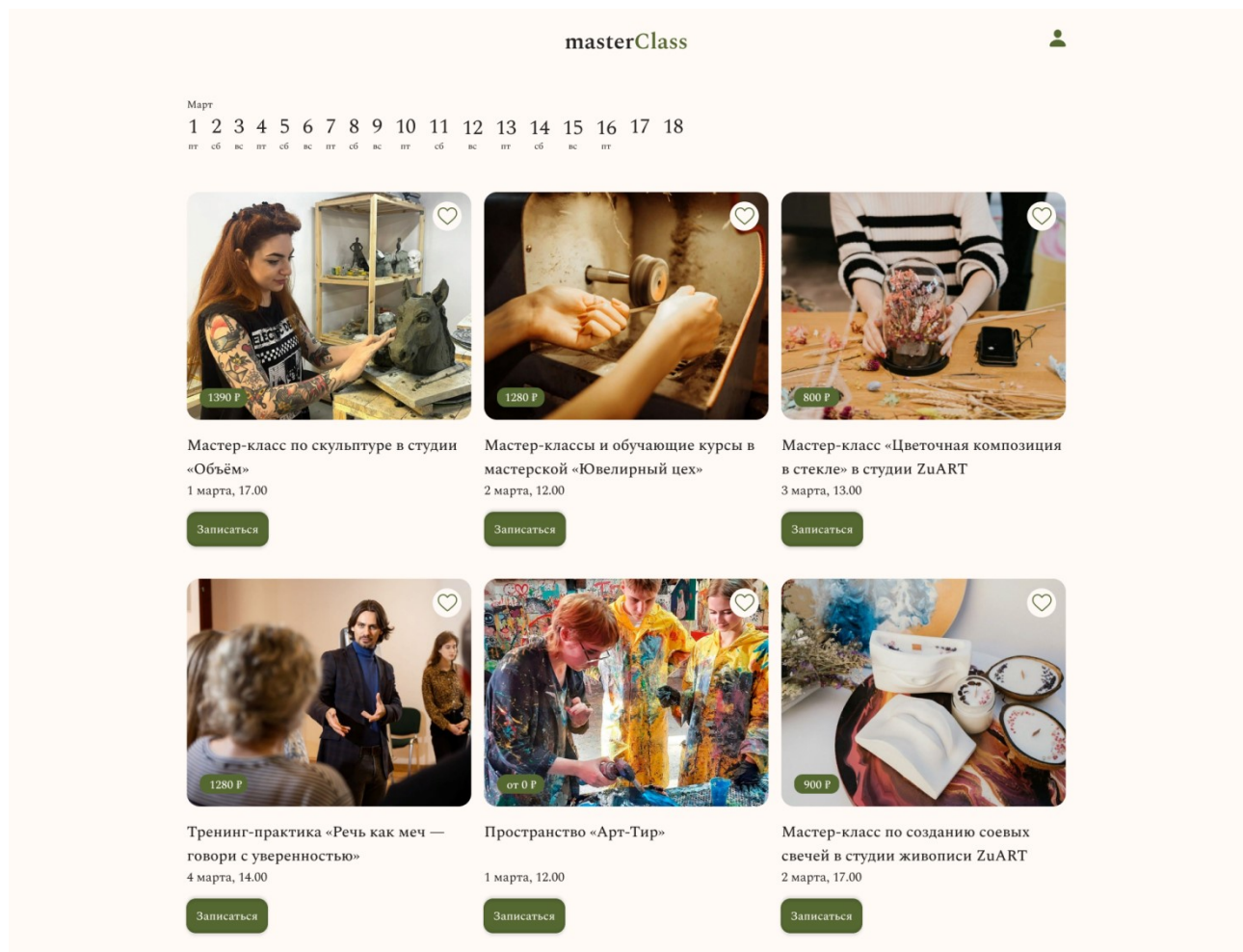


Рисунок Г.3 – Главная страница

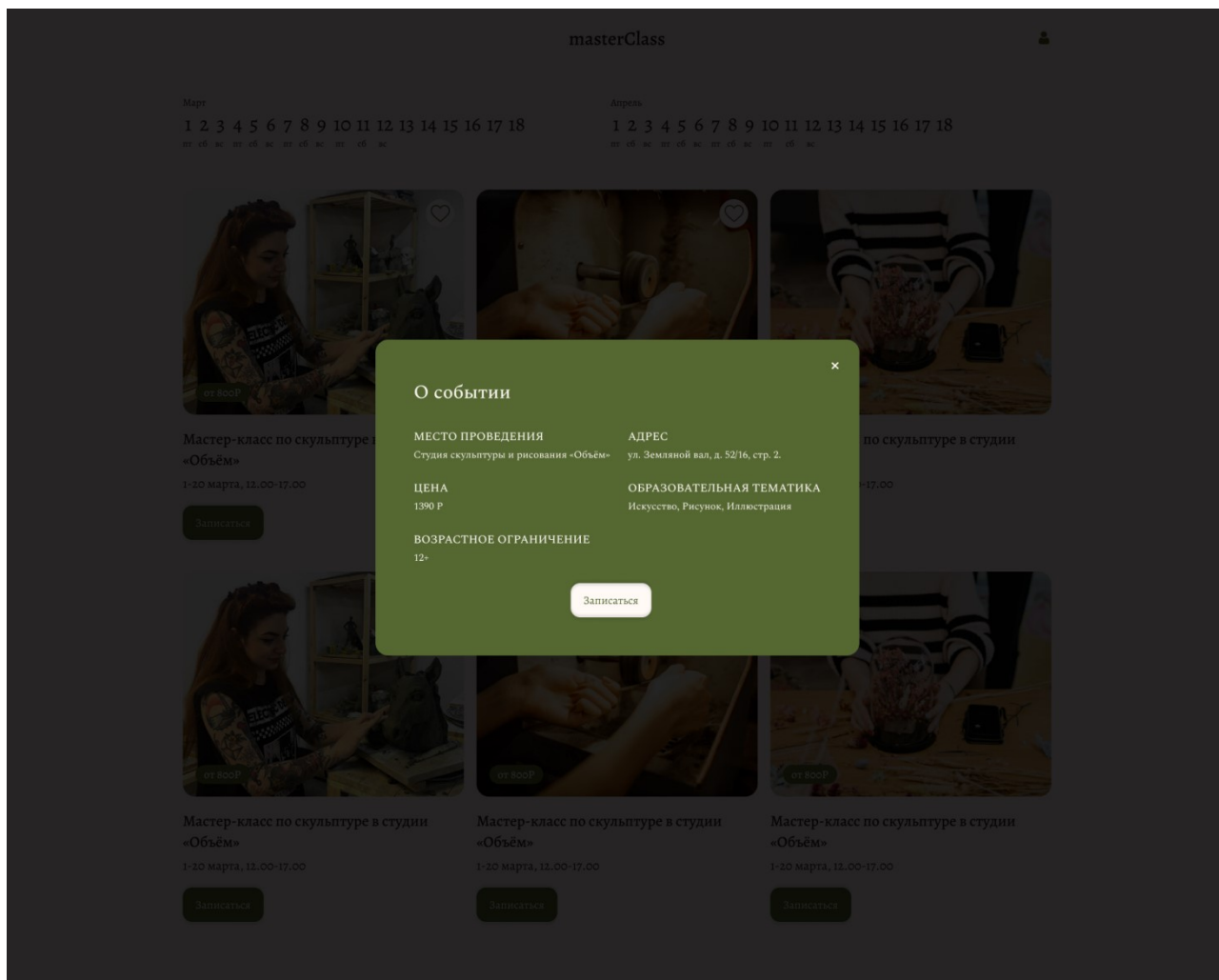


Рисунок Г.4 – Модальное окно «О событии»

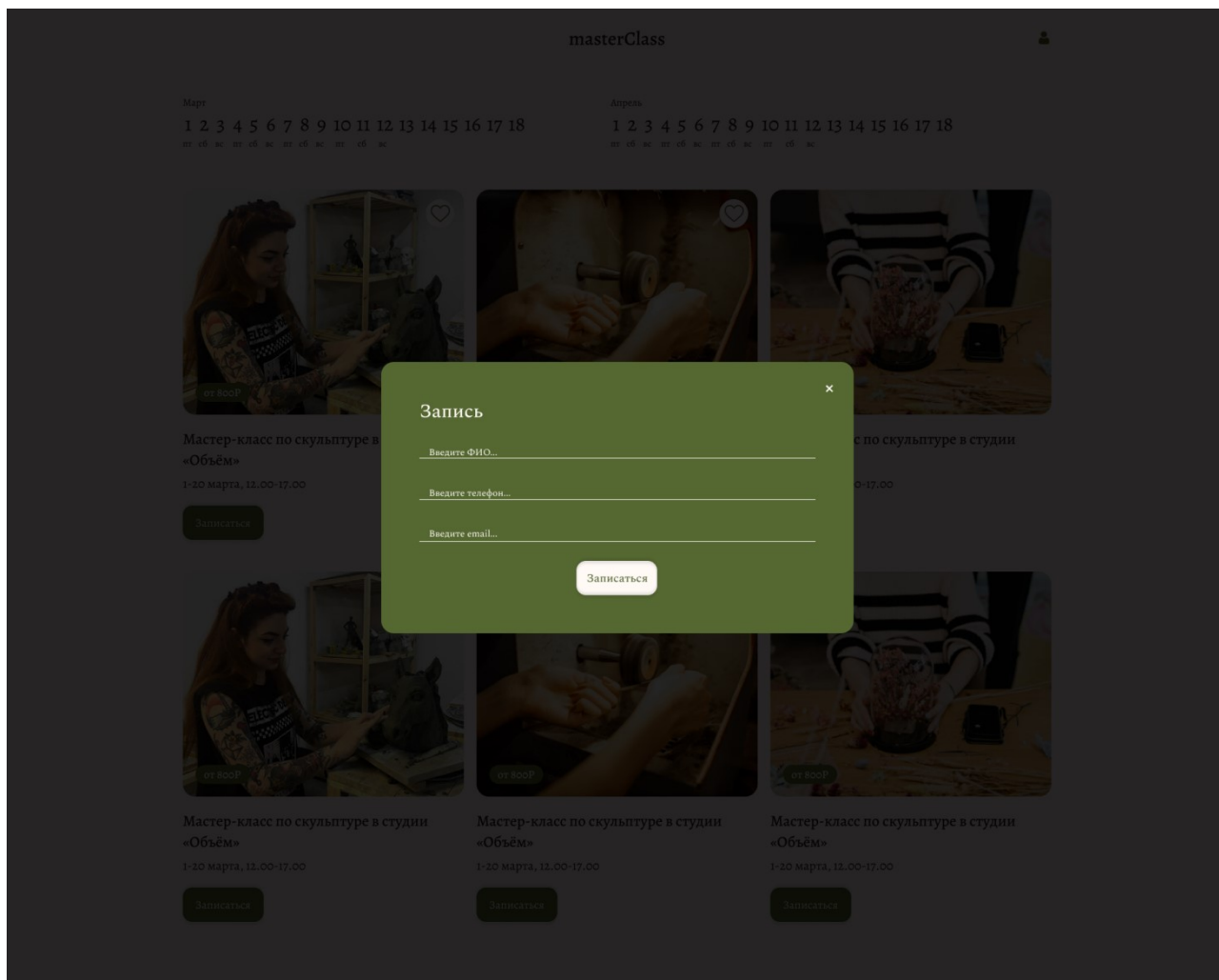


Рисунок Г.5 – Модальное окно «Запись»

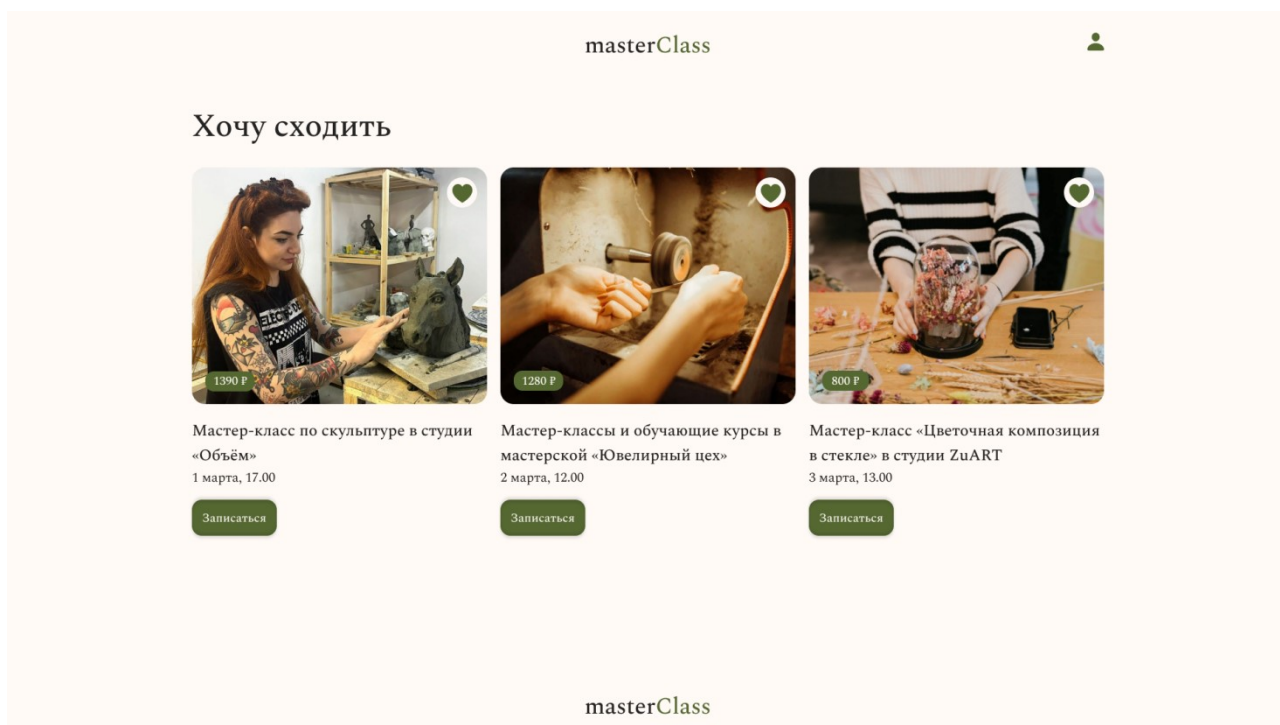


Рисунок Г.6 – Страница «Избранное»

					ОКЭИ 09.02.07 7024 27 У	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

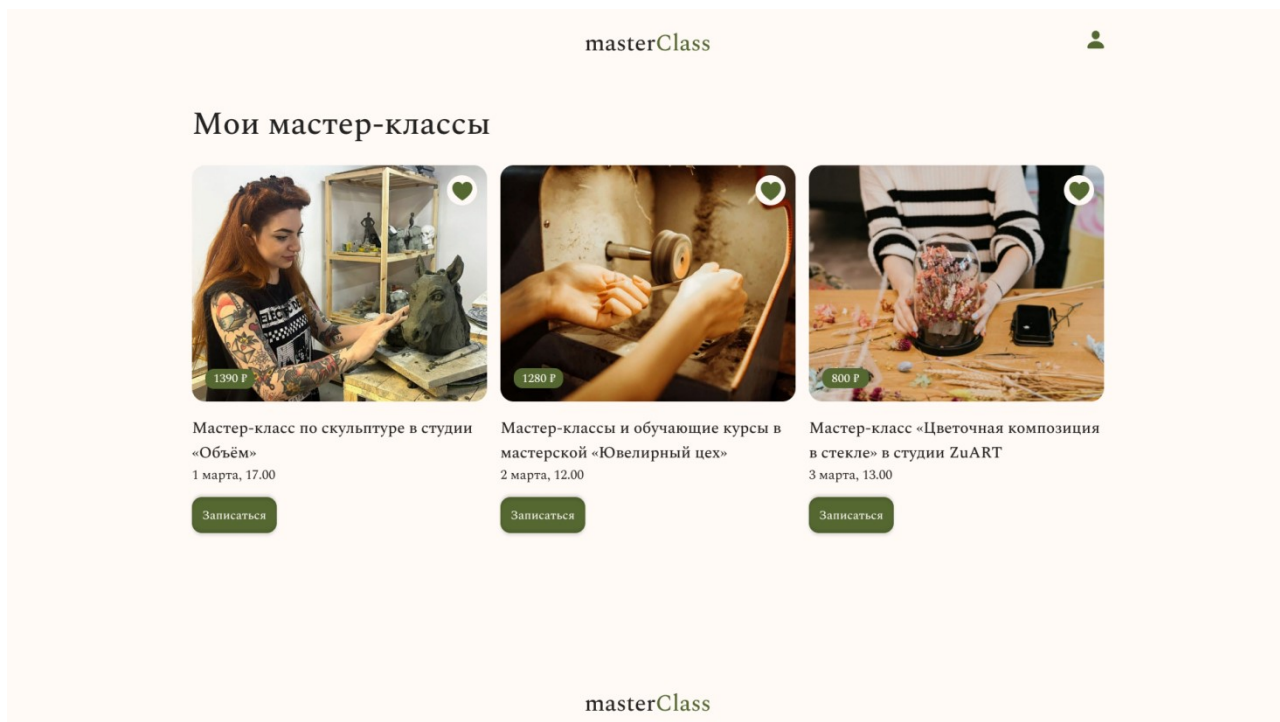


Рисунок Г.7 – Страница «Мои мастер-классы»

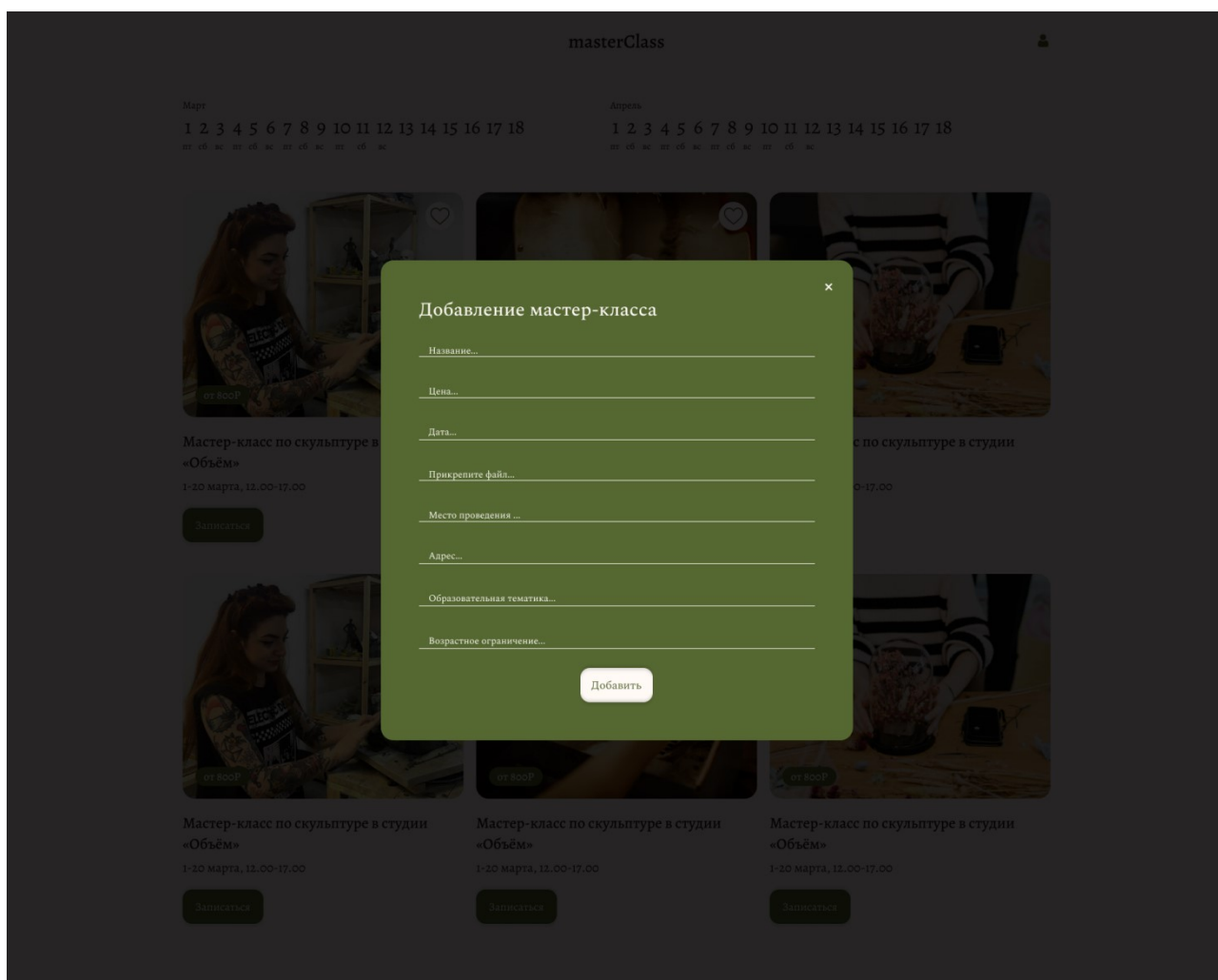


Рисунок Г.8 – Модальное окно «Добавление мастер-класса»

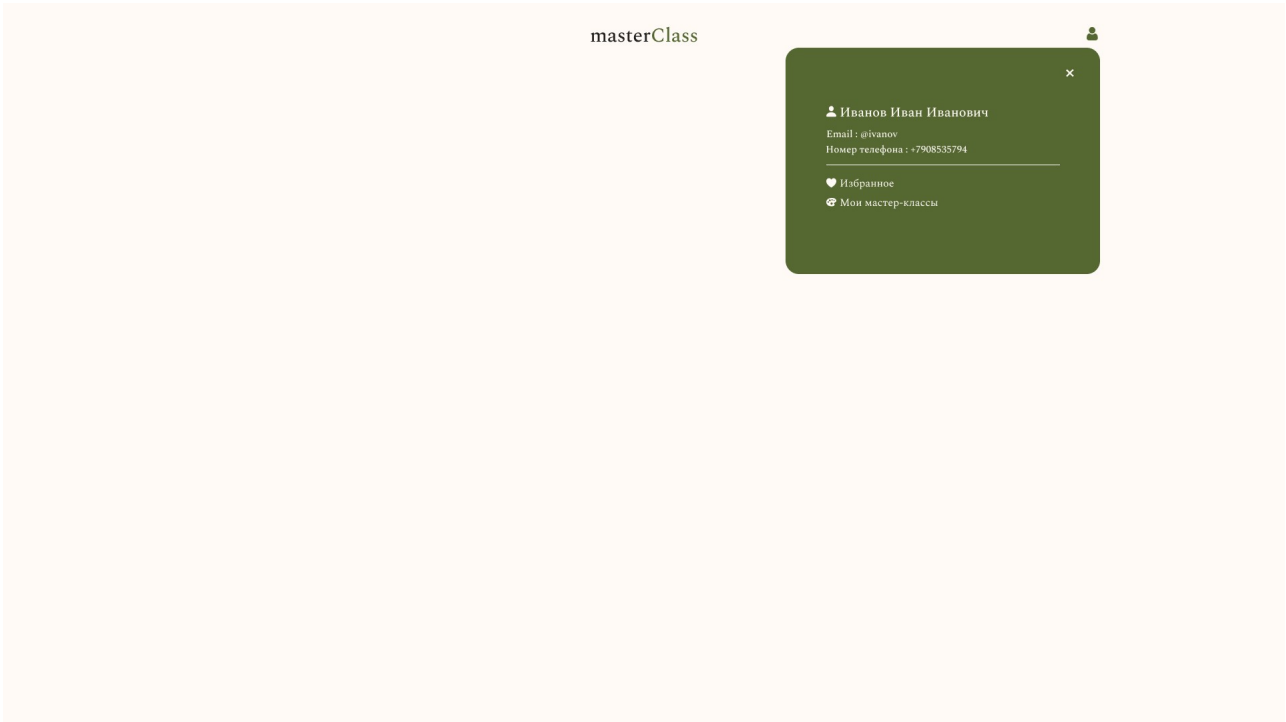


Рисунок Г.9 – Модальное окно «Личный кабинет пользователя»

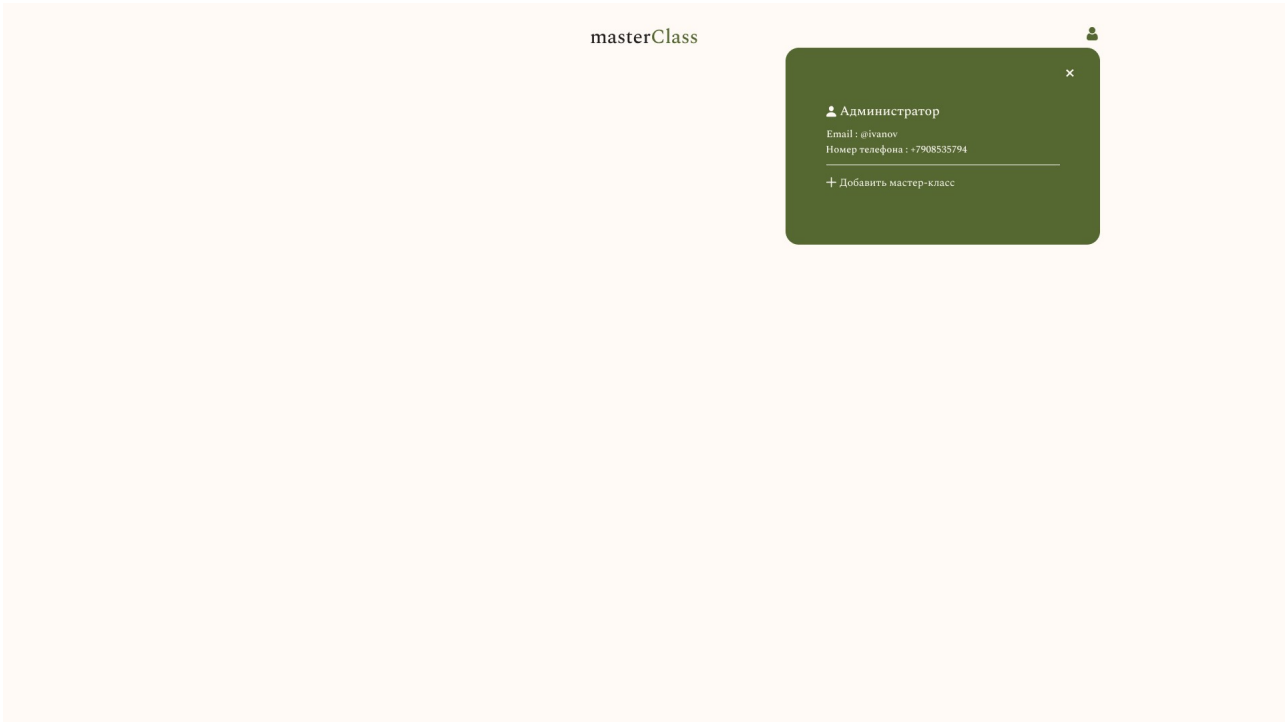


Рисунок Г.10 – Модальное окно «Личный кабинет администратора»