МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Факультет компьютерных наук

Кафедра информационных систем в телекоммуникациях

Система онлайн платежей

**Курсовой проект**

09.03.02 Информационные системы и технологии  
Информационные системы в телекоммуникациях

Руководитель В.С. Тарасов, ст. преподаватель\_\_.\_\_20

Обучающийся Д.А. Загреба, 3 курс, д/о Обучающийся Н.А. Ягодинцев, 3 курс, д/о Обучающийся К.Р. Ходжаев, 3 курс, д/о Обучающийся Я. Сулиман, 3 курс, д/о

Воронеж 2023

Содержание

[Содержание 2](#_Toc144669594)

[Введение 3](#_Toc144669595)

[1 Постановка задач 4](#_Toc144669596)

[1.1 Требования к разрабатываемой системе 4](#_Toc144669597)

[1.1.1 Функциональные требования 4](#_Toc144669598)

[1.1.1.1 Для клиентов 4](#_Toc144669599)

[1.1.1.2 Для модераторов 4](#_Toc144669600)

[1.1.1.3 Для администраторов 4](#_Toc144669601)

[1.1.2 Технические требования 5](#_Toc144669602)

[1.2 Требования к интерфейсу 5](#_Toc144669603)

[1.3 Задачи, решаемые в процессе разработки 5](#_Toc144669604)

[2 Анализ предметной области 7](#_Toc144669605)

[2.1 Терминология (глоссарий) предметной области 7](#_Toc144669606)

[2.2 Обзор аналогов 9](#_Toc144669607)

[2.3 Моделирование системы 10](#_Toc144669608)

[2.3.1 Диаграмма в стиле методологии IDEF0 10](#_Toc144669609)

[2.4 Аналитика веб-приложения 10](#_Toc144669610)

[2.5 Границы проекта 10](#_Toc144669611)

[3 Реализация 11](#_Toc144669612)

[3.1 Средства реализации 11](#_Toc144669613)

[3.2 Реализация серверной части веб-приложения 12](#_Toc144669614)

[3.3 Реализация клиентской части веб-приложения 12](#_Toc144669615)

[4 Тестирование 14](#_Toc144669616)

[4.1 Ручное UI–тестирование 14](#_Toc144669617)

[4.2 Юзабилити–тестирование 14](#_Toc144669618)

[Заключение 15](#_Toc144669619)

[Список используемых источников 16](#_Toc144669620)

Введение

В настоящее время Интернет-технологии активно используются в различных сферах, в том числе и в системах онлайн платежей.

В наше время, когда электронная коммерция с каждым днём занимает всё более прочную позицию в мировой экономике, необратимо встаёт вопрос о том, как принимать платежи и осуществлять денежные переводы онлайн. В этой области отличным помощником становятся платежные системы.

В данной курсовой работе рассмотрена разработка веб–приложения онлайн-кассы с детальной проработкой одного из способов оплаты.

Основная цель проекта заключается в создании удобной и функциональной системы, которая позволит пользователям легко и быстро начать принимать платежи на своем сайте.

В работе будет рассмотрен процесс проектирования и разработки веб-приложения, включая выбор подходящих технологий и инструментов, создание базы данных, разработку интерфейса и реализацию основных функций приложения. Кроме того, будут рассмотрены вопросы безопасности и защиты данных пользователей.

1. Постановка задач

Целью данного проекта является создание веб-приложения онлайн кассы.

Основными задачами проекта являются реализации услуг …….

Для достижения поставленных целей необходимо, во–первых, иметь представление о разрабатываемой системе, представленное необходимыми UML–диаграммами и разработанным дизайном веб-приложения, как в целом, так и в отдельных сценариях.

Во–вторых, необходимо реализовать базы данных, которые будут хранить информацию о магазинах клиентов и об осуществленных покупках.

Также, необходимо провести тестирование системы и ее аналитику.

* 1. Требования к разрабатываемой системе
     1. Функциональные требования

К разрабатываемому приложению выдвинуты следующие требования:

* разделение пользователей на: клиентов, мерчантов, администраторов и модераторов;
* клиент должен иметь возможность выбора способа оплаты которым он планирует осуществлять оплату;
* мерчант должен иметь возможность создавать новые магазины, следить за операциями проходящими через него, отслеживать суммы и статусы каждой операции;
* модератор должен иметь возможность отслеживать все операции и обрабатывать их в случае их нарушения, так же он должен иметь возможность пропускать или отменять операции в случае подозрительных действий и блокировать магазин мерчанта в случае нарушения правил площадки;
* администратор должен иметь возможность следить за всеми проходящими внутри процессами и управлять ими.
  + - 1. Для клиентов
    - Обеспечение возможности подключать свой магазин к платформе;
    - Обеспечение возможности отслеживания переводов денежных средств;
      1. Для модераторов
    - Обеспечение возможности просматривать подробную информацию о конкретной заявке;
    - Обеспечение возможности вывода денежных средств из системы;
      1. Для администраторов
    - Обеспечение возможности просматривать подробную информацию ( все транзакции по каждому «магазину»;
    - Обеспечение возможности просматривать успешные переводы на каждый номер телефона.
    1. Технические требования

Программный продукт должен обеспечить:

* авторизацию пользователей с использованием почты и пароля;
* шифрование пароля при записи в БД;
* блокировка карт на время перевода средств одним из клиентов;
* балансировка среднего баланса на каждой карте;
* использование отдельного микросервиса для обработки запросов на Go;
* панель управления на Laravel;
* интерфейс пользователя на React;
* хранение данных в БД.
  1. Требования к интерфейсу

Интерфейс должен быть выполнен в единой для всех экранов цветовой гамме, едином стиле. Все надписи должны быть легко читаемы, все элементы управления должны быть выполнены в едином стиле, размере, должны выделяться на фоне содержимого экранов.

Интерфейс должен содержать только необходимую для пользователя информацию. Информация должна находиться в тех местах приложения, где она будет актуальна. Основные элементы управления должны быть заметны для пользователя.

* 1. Задачи, решаемые в процессе разработки

Перед проектом были поставлены следующие задачи:

* анализ предметной области;
* анализ аналогов;
* написание технического задания;
* описание разрабатываемой системы UML диаграммами;
* разработка БД;
* реализация ролей;
* реализация функциональных возможностей ролей;
* разработка функциональных возможностей сайта;
* создание макета дизайна и его реализация;
* реализация интерфейса;
* проведение тестирования;
* описание процесса разработки и результата.

1. Анализ предметной области
   1. Терминология (глоссарий) предметной области

Проект, система – разрабатываемое веб-приложение.

Интернет-эквайринг – это прием, который позволяет совершить оплату картой через интернет с использованием веб-интерфейса, предоставляемого банком или платежной системой.

Front–end – клиентская часть приложения. Отвечает за получение информации с программно–аппаратной части и отображение ее на устройстве пользователя.

Клиентская сторона – компьютер, использующий ресурсы сервера и предоставляющий пользователю возможность взаимодействия с системой.

Back–end – программно–аппаратная часть приложения. Отвечает за функционирование внутренней части приложения.

Сервер, серверная часть – компьютер, обслуживающий другие компьютеры (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач.

GitHub – веб–сервис для хостинга IT–проектов и их совместной разработки.

Фреймворк– программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

REST API (REST) – стиль архитектуры программного обеспечения для построения масштабируемых веб–приложений.

CSS – формальный язык описания внешнего вида веб–страницы, написанного с использованием языка разметки (HTML).

HTML – стандартизированный язык разметки для просмотра веб-страниц в браузере.

Юзабилити – тестирование – это метод оценки интерфейса со стороны удобства и эффективности его использования.

React – JavaScript-библиотека для создания пользовательских интерфейсов.

Go – компилируемый многопоточный язык программирования, разработанный внутри компании Google.

Laravel –  это бесплатный PHP-фреймворк с открытым исходным кодом, специально разработанный для создания сложных веб-приложений.

* 1. Обзор аналогов
     1. Тинькофф

Банк-эквайер Тинькофф позиционирует себя как инновационный и открытый банк для любого бизнеса, в том числе и малого.

У банка собственный процессинговый центр, прозрачная комиссия, которая зависит от оборота компании, а также открытые тарифы на услуги.

К недостаткам пользователи относят процесс валютного контроля, который зачастую может занять много времени.

* + 1. Robokassa

**Robokassa** - платежный сервис, который был запущен в 2008 году и на данный момент по популярности уступает только Яндекс.Кассе.

Выделим основные плюсы Робокассы:

* широкий спектр возможностей по приему платежей: через онлайн банки, платежные терминалы, в некоторых розничных сетях, может списать средства со счета мобильного телефона или принять электронную валюту и т.д.;
* круглосуточная поддержка покупателей;
* система обслуживает как юридических, так и физических лиц;
* удобный и несложный процесс подключения.

К недочетам можно отнести:

* достаточно высокая комиссия - 5%;
* в истории сервиса были случаи временного прекращения приема платежей ИП и физическим лицам.

* 1. Моделирование системы
     1. Диаграмма в стиле методологии IDEF0
  2. Аналитика веб-приложения

В качестве системы для сбора данных об использовании приложения пользователями используется сервис «Яндекс метрика».

1. Реализация
   1. Средства реализации

Веб-приложение имеет архитектуру, соответствующую шаблону клиент-серверного приложения и разделяется на back–end и front–end посредством REST API. При этом на back-end используется микросервисная архитектура в стадии разработки.

Данная архитектура веб–приложения соотносится с основными требованиями к проекту, а именно:

* разделение пользователей на: клиентов, мерчантов, администраторов и модераторов;
* авторизация пользователя.

Для реализации программно–аппаратной части были выбраны следующие технологии:

* laravel для панели управления ролями и интерфейсом для работы с заказами и клиентами;
* Go для обработки запросов на API системы;
* MySQL для работы с базой данных/

Для реализации клиентской части были выбраны следующие технологии:

* React для взаимодействия с пользователем и улучшения пользовательского опыта.
  1. Реализация серверной части веб-приложения

Для осуществления основных сценариев веб–приложения: ……… необходимы регистрация, авторизация и аутентификация.

* 1. Реализация клиентской части веб-приложения

Для реализации основных сценариев веб-приложения, клиентская часть разработки делится постранично. Каждая страница описывается языком программирования JavaScript, языком разметки HTML и благодаря использованию фреймворка React. За реализацию заранее осуществленного и утвержденного командой разработчиков дизайна используется язык стилей CSS.

Архитектура разработки была организована согласно бизнес-логике проекта на основании модульного подхода, по которому все компоненты и логика находятся рядом друг с другом.

Все страницы веб-приложения были реализованы и представлены командой разработчиков в соответствии с заявленным дизайном и соответствующими правками, внесенными в процессе разработки системы.

1. Тестирование
   1. Ручное UI–тестирование
   2. Юзабилити–тестирование

Юзабилити–тестирование[[11]](#_Юзабилити-тестирование_[Электронный) является исследованием, выполняемым с целью определения, удобен ли некоторый искусственный объект для его предполагаемого применения.

Заключение

В ходе выполнения курсового проекта командой было разработано веб-приложение ……., соответствующее поставленным перед проектом задачам.

В начале разработки был проведен анализ предметной области, определены основные требования к разрабатываемой системе, определены основные сценарии веб-приложения и пользовательские истории.

По результатам разработки проводился ряд тестов с целью проверки работоспособности системы.

В процессе работы были реализованы следующие задачи проекта:

* разделение пользователей на: …….;
* обеспечение возможности ……..;
* ………..

Список используемых источников

# Python 3.9.16 documentation [Электронный ресурс]. – Режим доступа: https://docs.python.org/3.9/

# Django: The web framework for perfectionists with deadlines [Электронный ресурс]. – Режим доступа: https://docs.djangoproject.com/en/4.2/.

# PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.

# Документация JavaScript-библиотеки React.js [Электронный ресурс]. Режим доступа: https://ru.react.js.org/docs/getting-started.html

# Документация Sass - метаязыка на основе CSS [Электронный ресурс]. Режим доступа: https://sass-lang.com/documentation/

# Основы HTML [Электронный ресурс] Режим доступа: https://html5book.ru/osnovy-html/

# Ручное UI-тестирование [Электронный ресурс]. Режим доступа: https://ux-journal.ru/kak-provodit-ui-testirovanie-sravnenie-instrumentov.html

# Юзабилити-тестирование [Электронный ресурс]. Режим доступа: https://lpgenerator.ru/blog/2016/07/28/yuzabiliti-testirovanie-poshagovaya-instrukciya-na-primere-yelp/

# ГОСТ 7.32-2001 [Электронный ресурс]. Режим доступа:https://kpfu.ru/portal/docs/F1867381138/gost7\_32\_2001.pdf