

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

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра комплексной информационной безопасности электронно-
вычислительных систем (КИБЭВС)

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ИГРЫ В «МАФИЯ»

Пояснительная записка к курсовой работе
по дисциплине «Технологии и методы программирования»

Студент гр.718-1

_____ А.С. Зазуля

«__» _____ 2021

Руководитель

Преподаватель кафедры

КИБЭВС

_____ Койшинов Т.С.

«__» _____ 2021

Томск 2021

Реферат

Пояснительная записка к курсовой работе по дисциплине «Технологии и методы программирования» содержит 24 страниц, 23 рисунков, 6 источников, 1 приложение.

Ключевые слова: ИГРА, ПРОЦЕСС, МАФИЯ, ПЛАТФОРМА, СТРАНИЦА, САЙТ, GIT, АВТОРИЗАЦИЯ, РЕГИСТРАЦИЯ.

Целью работы курсовой работы является разработать приложение, с помощью которого можно будет производить игровой процесс в игре «Мафия»

Результат работы:

- Определены необходимые технологии;
- проработана архитектура приложения;
- разработано приложение;
- реализована автоматизированная развертка приложения на удаленном сервере;
- реализована взаимодействие проекта с Gitlab.

Клиент-серверное приложение разработано в среде разработки Microsoft Visual Studio 2019 с использованием рабочей среды .NET Core 3.1, а также подключена СУБД Microsoft SQL Server.

Пояснительная записка выполнена в текстовом редакторе Microsoft Office Word согласно ОС ТУСУР 01-2013.

Abstract

The explanatory note to the term paper on the discipline "Technologies and programming methods" contains 24 pages, 23 figures, 6 sources, 1 applications.

Keywords: GAME, PROCESS, MAFIA, PLATFORM, PAGE, SITE, GIT, AUTHORIZATION, REGISTRATION.

The purpose of the course work is to develop an application for playing the "Mafia"

This term results:

- The necessary technologies have been identified;
- the application architecture has been developed;
- the application has been developed;
- the automated application deployment on the remote server is implemented;
- project interaction with Gitlab has been implemented.

The client-server application was developed in the Microsoft Visual Studio 2019 development environment using the .NET Core 3.1 working environment, and the Microsoft SQL Server database is also connected.

Explanatory note is made in text editor of Microsoft Office Word according to OS TUSUR 01-2013.

Содержание

1 Введение	5
2 Проектирование системы.....	7
2.1 Проектирование блок-схем	7
2.2 UML-диаграмма.....	9
2.3 Макет приложения	11
3 Функционал системы	14
3.1 Пользовательский интерфейс	14
4 Тестирование	20
4.1 Программное тестирование	20
5 Git	21
5.1 Настройка репозитория	21
6 Заключение	23
Список использованных источников	24
Приложение А	25

1 Введение

Тема проекта: «Разработка приложения для игры в «Мафия»

Целью работы курсовой работы является разработать приложение, с помощью которого можно будет производить игровой процесс в игре «Мафия»

Для достижения поставленной цели были решены следующие задачи:

1. Написать программу, которая будет взаимодействовать с пользователями.
2. Необходимо реализовать страницу регистрации и авторизации.
3. Необходимо реализовать игровую платформу, где будет производиться игровой процесс для пользователей.
4. Необходимо предоставить возможность выхода из веб-приложения авторизованному пользователю.

Поставленные задачи были проанализированы и решены. Решение данных задач было реализовано на языке C# в среде разработки .NET Core 3.1. Для сохранения данных была подключена СУБД Microsoft SQL Server. Были проведены тестирования приложения.

Данное приложение на сегодняшний день актуально. Дело в том, что взаимодействие между игроками происходит в онлайн режиме и никто не знает, что скрывается по ту сторону экрана. Может быть Мафия, Шериф, Дон-Мафия или Мирный житель. На сегодняшний день это очень актуально, потому что нельзя собираться в больших компаниях из-за Covid-19 и единственное решение — это онлайн игры. Это бесплатная онлайн игра, основанная на классической «карточной» мафии. Для того чтобы начать играть, Вам не придется покупать карты для игры в мафию.

Для меня данное приложение интересно тем, что в рамках учебного процесса я могу сама написать такую программу. Играть на онлайн платформе с одноклассниками или друзьями. Я хочу, чтобы общение во время

карантина продолжалось — пусть пока не вживую, а в интернете. Продолжайте играть, не нарушая правил изоляции!

2 Проектирование системы

2.1 Проектирование блок-схем

На рисунке 2.1 приведена блок-схема авторизации и регистрации.

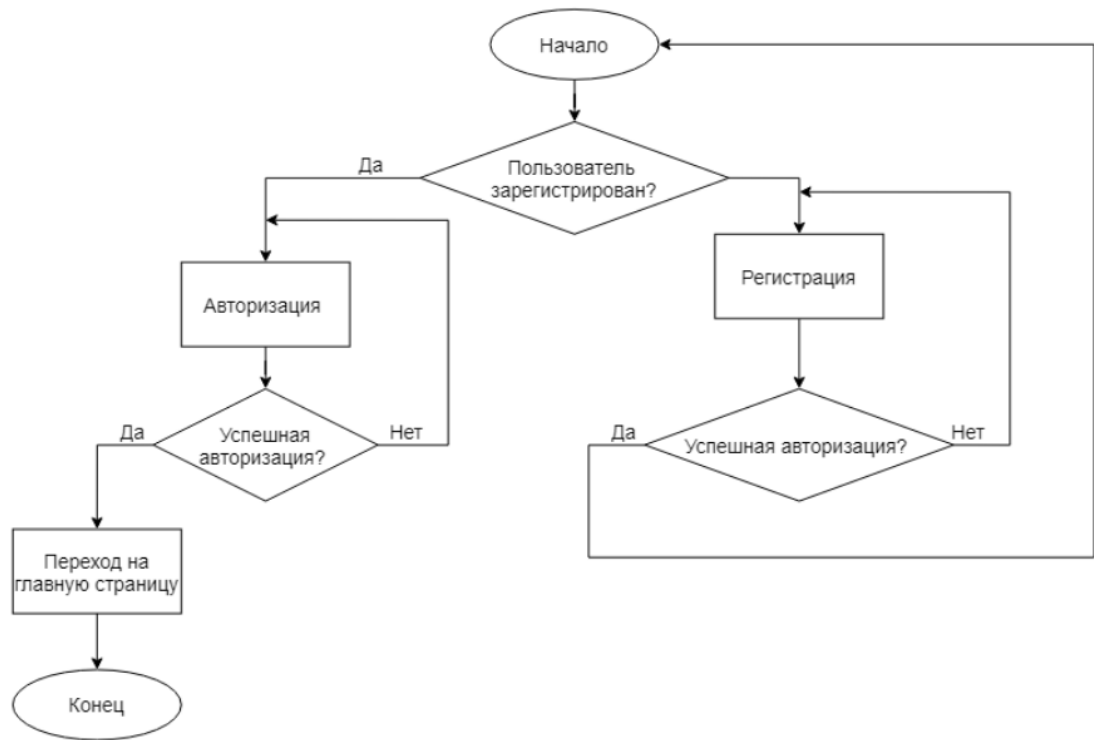


Рисунок 2.1 – Блок-схема авторизации и регистрации

На рисунке 2.2 – 2.3 приведены блок-схемы игровых действий.

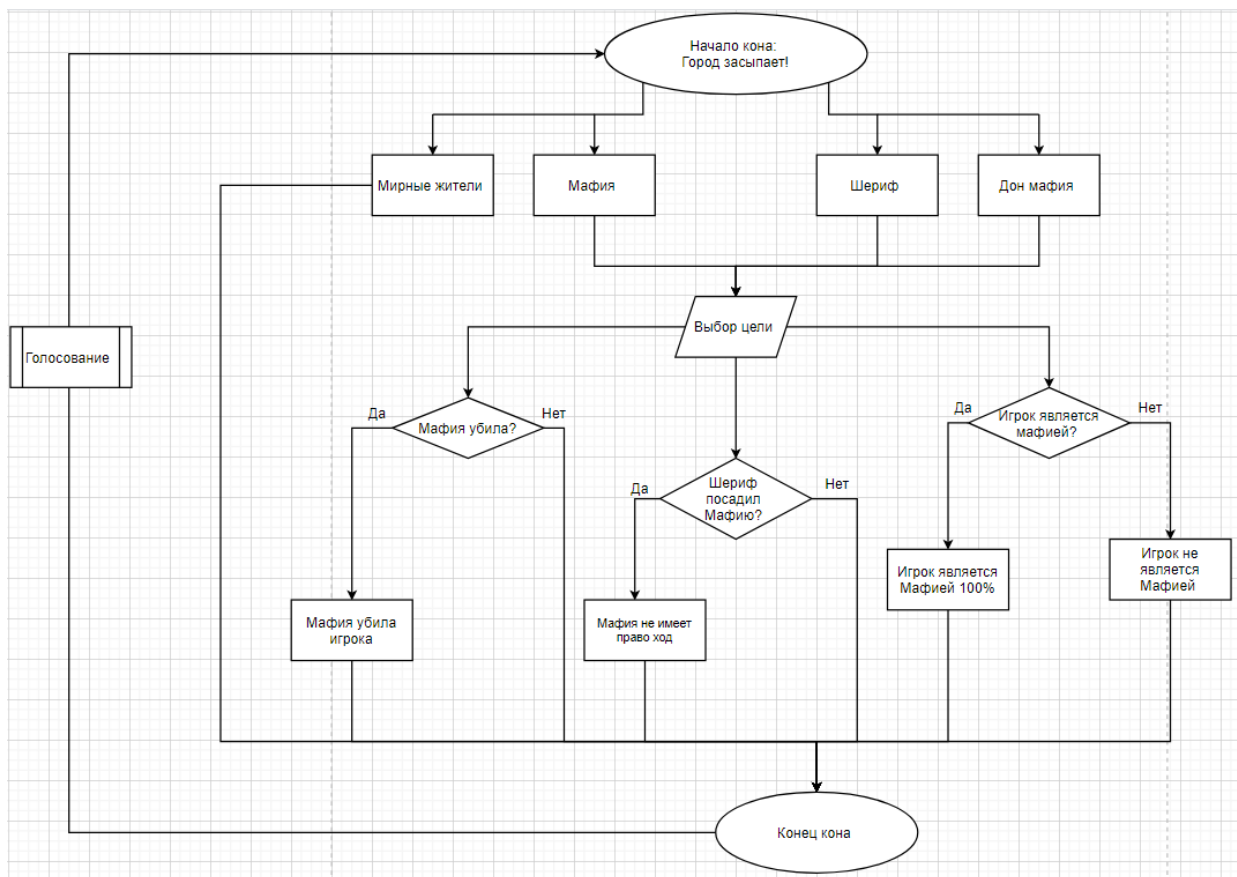


Рисунок 2.2 – Блок-схема игровых действий



Рисунок 2.3 – Блок-схема игровых действий

2.2 UML-диаграмма

В системе имеются 2 действующих лица:

- Авторизованный пользователь
- Неавторизованный пользователь

Каждый вариант использования отражает последовательность действий, которые может осуществлять система в ответ на воздействия пользователей.

Рассмотрим действия, которые доступны неавторизованному пользователю системы:

Авторизация.

1. Вариант использования начинается после того, как пользователь переходит на страницу авторизации.
2. После перехода пользователя на страницу авторизации у него будет вариант зарегистрировать.
3. Вариант использования завершается.

«Авторизация»

Основной поток событий:

1. Вариант использования начинается после того, как неавторизованный пользователь переходит форму авторизации и предлагает пользователю ввести логин и пароль.

2. Пользователь вводит логин и пароль.

3. Приложение проверяет введенные данные и подтверждает их.

Пользователь становится авторизованным пользователем.

4. Авторизованный пользователь имеет право переходить между страницами сайта.

5. Пользователь может участвовать в игровом процессе игры «Мафия».

6. Редактировать свой профиль и просматривать рейтинг.

7. Вариант использования завершается.

На рисунке 2.4 изображена UML-диаграмма, авторизованного пользователя и неавторизованного пользователя.

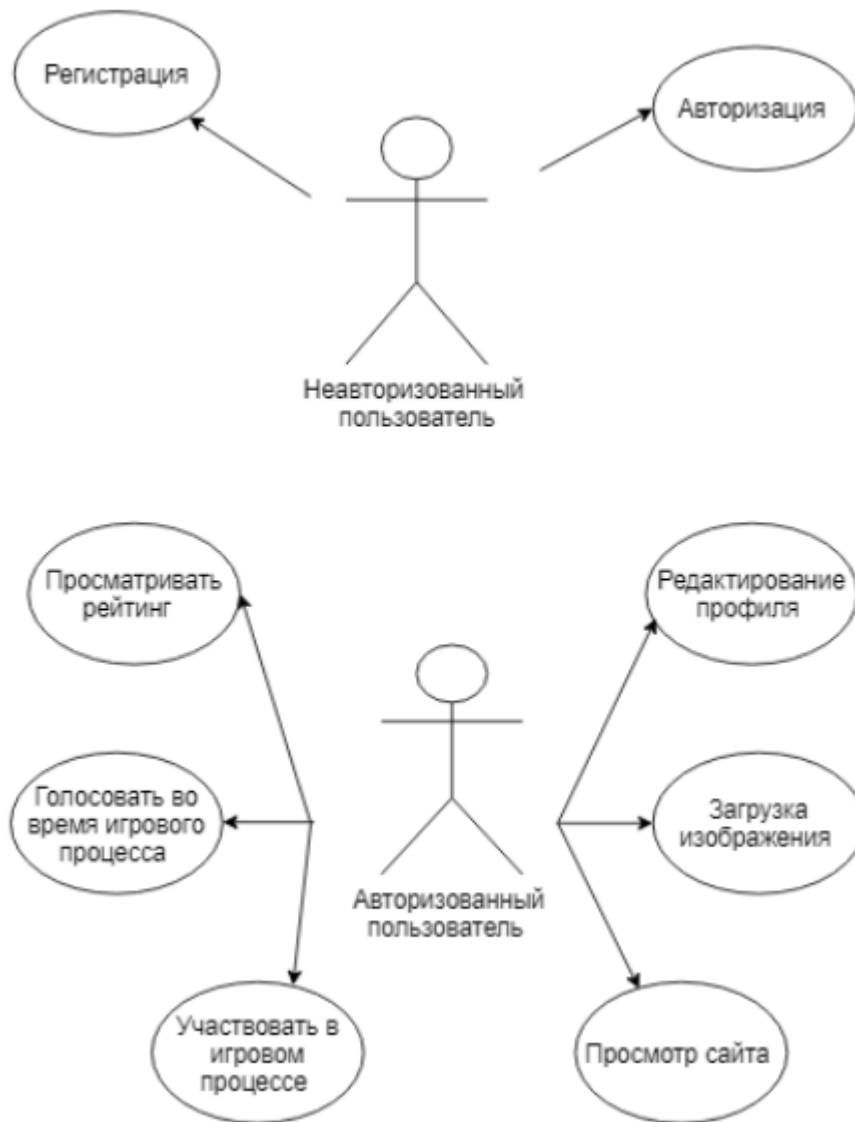
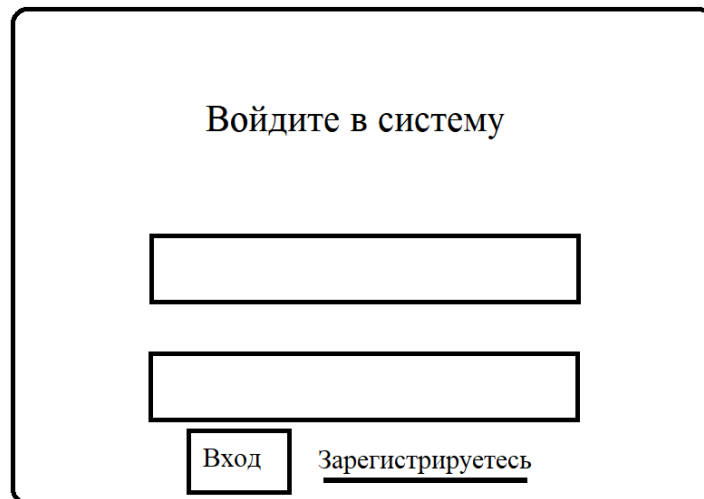


Рисунок 2.4 – UML-диаграмма

2.3 Макет приложения

В самом начале работы был проработан примерный макет сайта. На рисунке 2.5 – 2.6 изображён макет авторизации и регистрации.

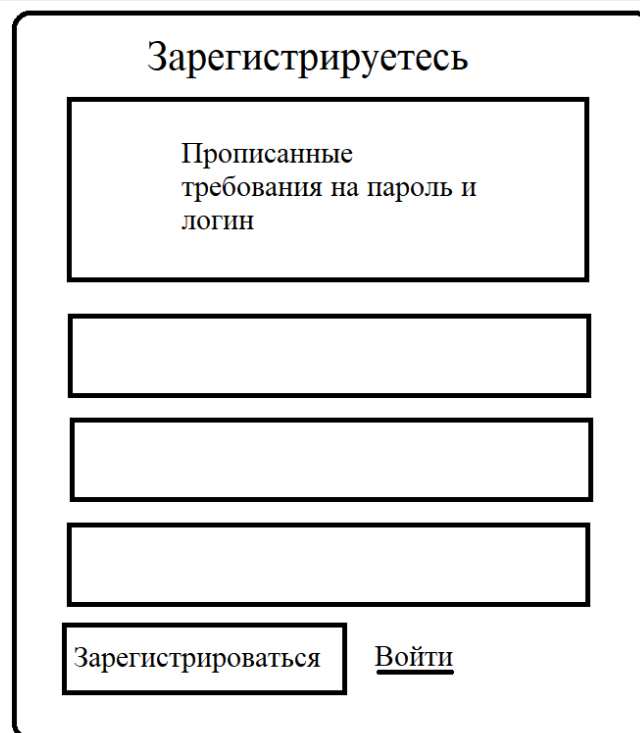


Войдите в систему

Two empty rectangular input fields for username and password.

Вход Зарегистрируетесь

Рисунок 2.5 – Форма авторизации



Зарегистрируетесь

Прописанные
требования на пароль и
логин

Three empty rectangular input fields for registration details.

Зарегистрироваться Войти

Рисунок 2.6 – Форма регистрации

На рисунке 2.7 предоставлена примерная схема главной и последующих страниц.

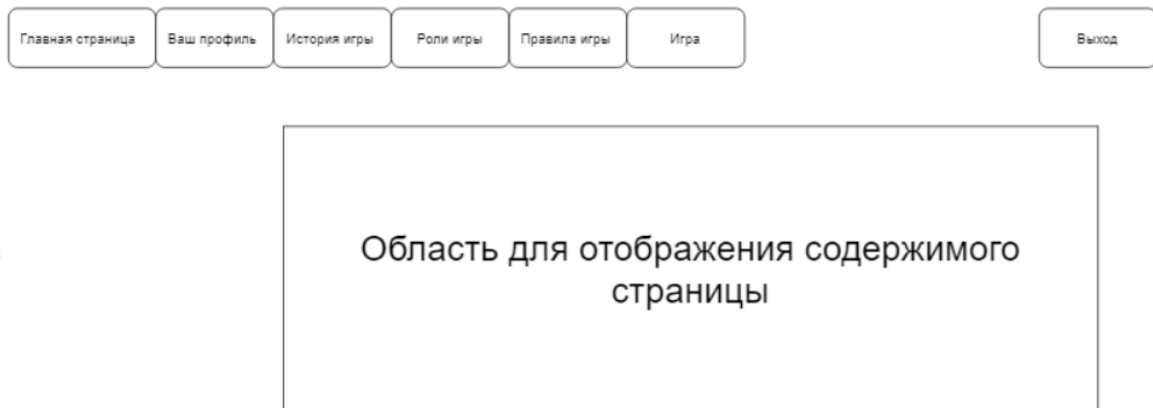


Рисунок 2.7 – Схема макета сайта

На рисунке 2.8 предоставлен макет игровой платформы.

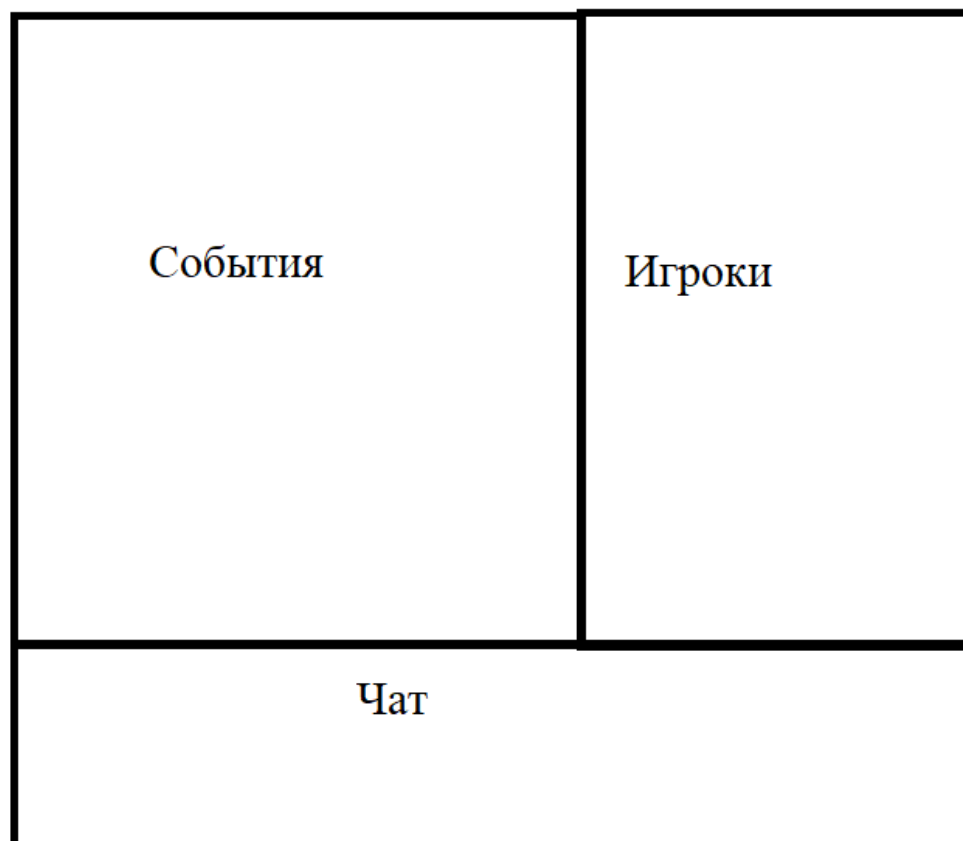


Рисунок 2.8 – Схема игровой платформы.

Сайт должен позволять пользователям:

- зарегистрироваться;
- авторизоваться;
- осуществлять навигацию по сайту (переход между страницами);
- изменять профиль;
- участвовать в игровом процессе;
- голосовать во время игрового процесса;
- выход со своего аккаунта.

Необходимо создать следующие страницы сайта:

- регистрация/авторизация;
- главная страница;
- профиль пользователя;
- история игры;
- роли игры;
- правила игры;
- игра.

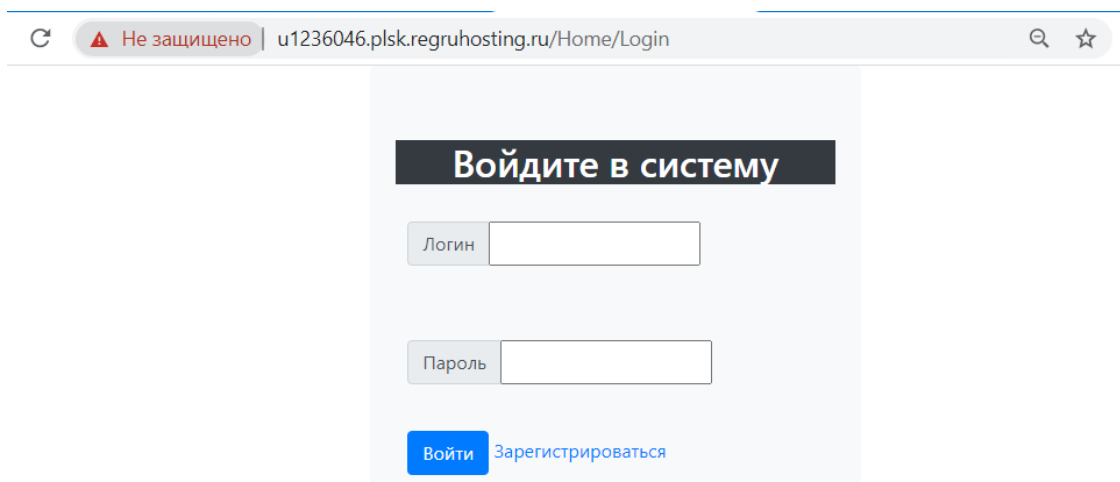
Необходимо создать меню навигации по сайту на рисунке 2.7
предоставлен примерное меню по сайту. Меню должно состоять из
следующих элементов:

- главная страница;
- профиль пользователя;
- история игры;
- роли игры;
- правила игры;
- игра.
- контакты.

3 Функционал системы

3.1 Пользовательский интерфейс

Первое, что доступно пользователю – это авторизация. Так как пользователь первый раз находится на сайте нужно перейти на страницу регистрации. (Рисунок 3.1)



Скриншот веб-браузера, отображающий страницу авторизации. В адресной строке виден URL: u1236046.plsk.regruhosting.ru/Home/Login. Над формой входа находится заголовок «Войдите в систему». Форма содержит два поля ввода: «Логин» и «Пароль». Внизу формы расположены две кнопки: «Войти» (синяя) и «Зарегистрироваться» (синий текст).

Рисунок 3.1 – Страница авторизации

Перейдя на страницу регистрации, пользователь должен придумать себе логин и пароль, которые будет проходить по требованиям, которые описаны на странице. (Рисунок 3.2)

Не защищено | u1236046.plsk.reg.ru/Account/Register

Зарегистрируйтесь

- Пароль и логин должен содержать хотя бы один цифровой символ.
- Пароли и логин должны содержать хотя бы одну строчную букву («a» - «z»).
- Пароль и логин должен быть больше 5 символов.

Логин

Пароль

Пароль

[Зарегистрироваться](#) [Войти](#)

Рисунок 3.2 – Страница регистрации

После успешной регистрации и авторизации пользователь попадает на главную страницу. (Рисунок 3.3)

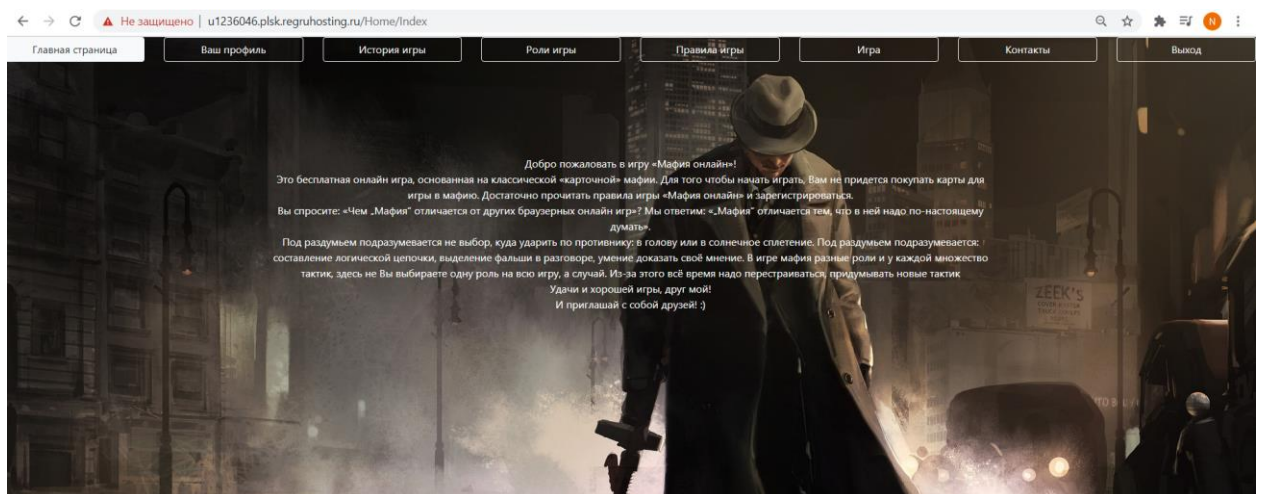


Рисунок 3.3 – Главная страница сайта

Пользователь имеет право переходить между страницами. Допустим на страницу «Ваш профиль». На странице можно увидеть свой рейтинг и изменить фотографию профиля. (Рисунок 3.4 - 3.5)

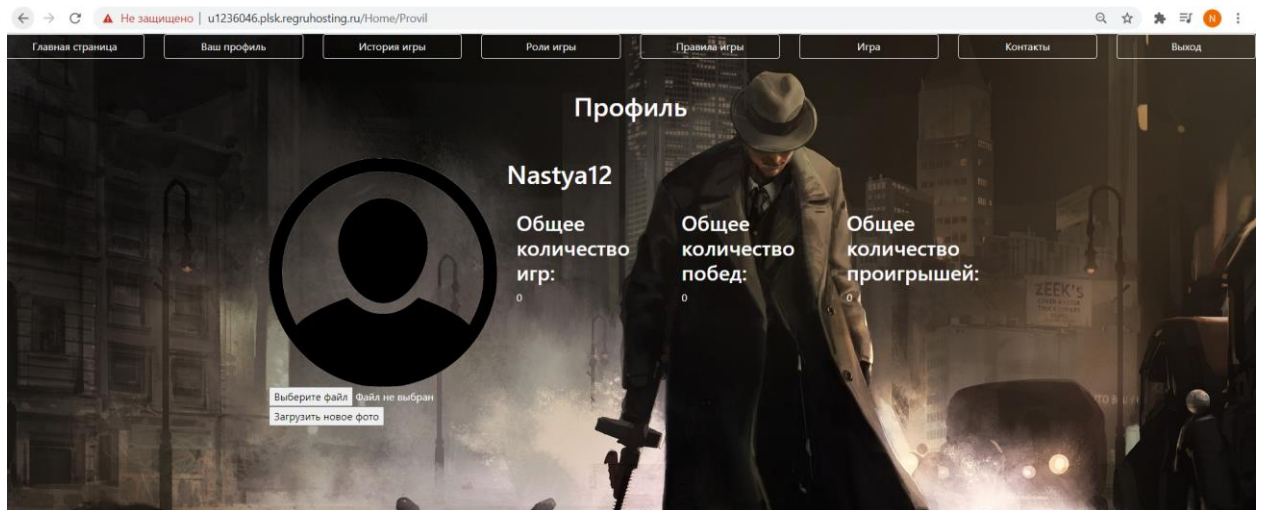


Рисунок 3.4 – Страница профиля

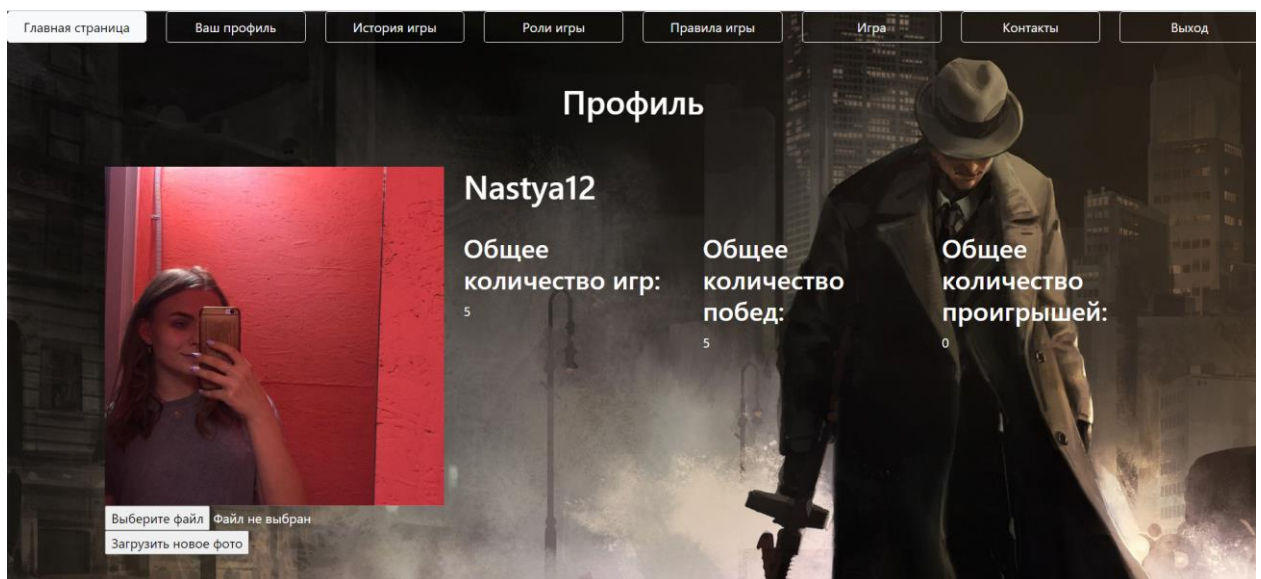


Рисунок 3.5 - Страница профиля

Для пользователя было созданы такие разделы как «История игры», «Роли игры», «Правила игры», «Контакты» для лучше ознакомления с игрой. (Рисунок 3.5-3.8)

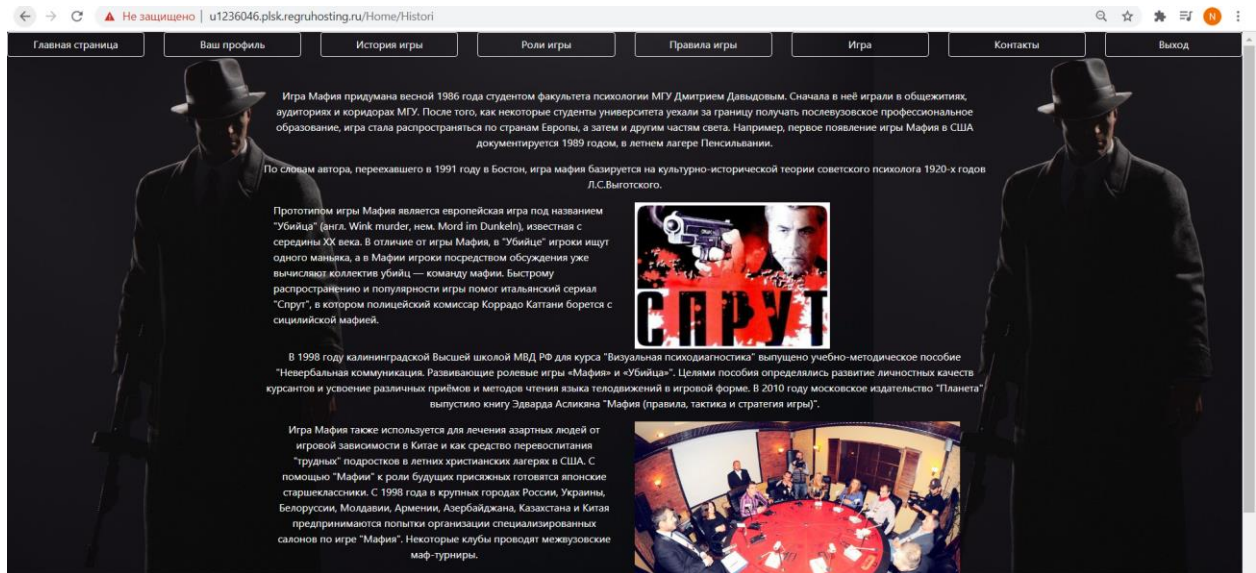


Рисунок 3.5 – Страница истории игры

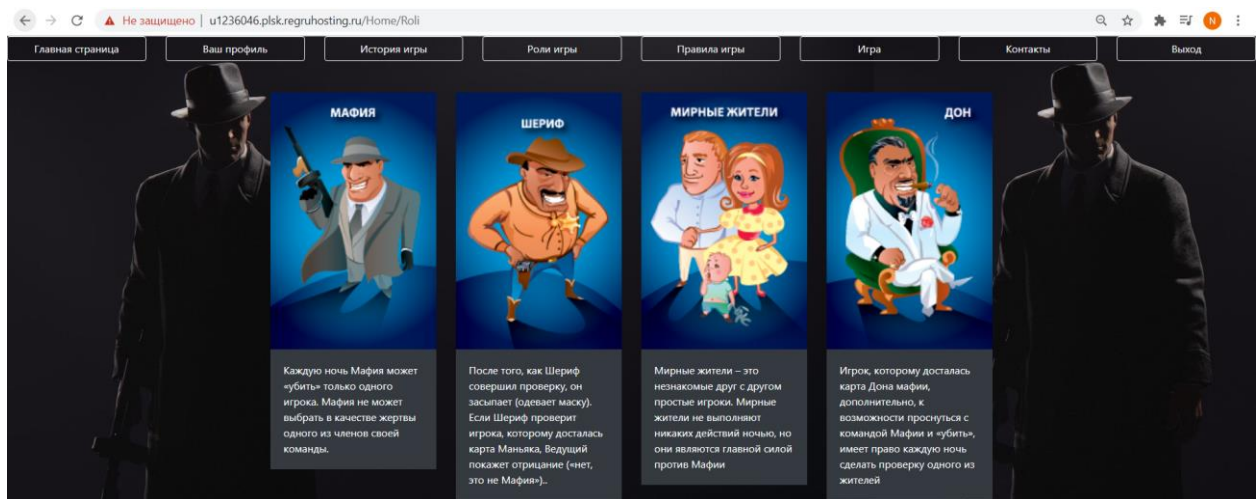


Рисунок 3.6 – Страница ролей в игре

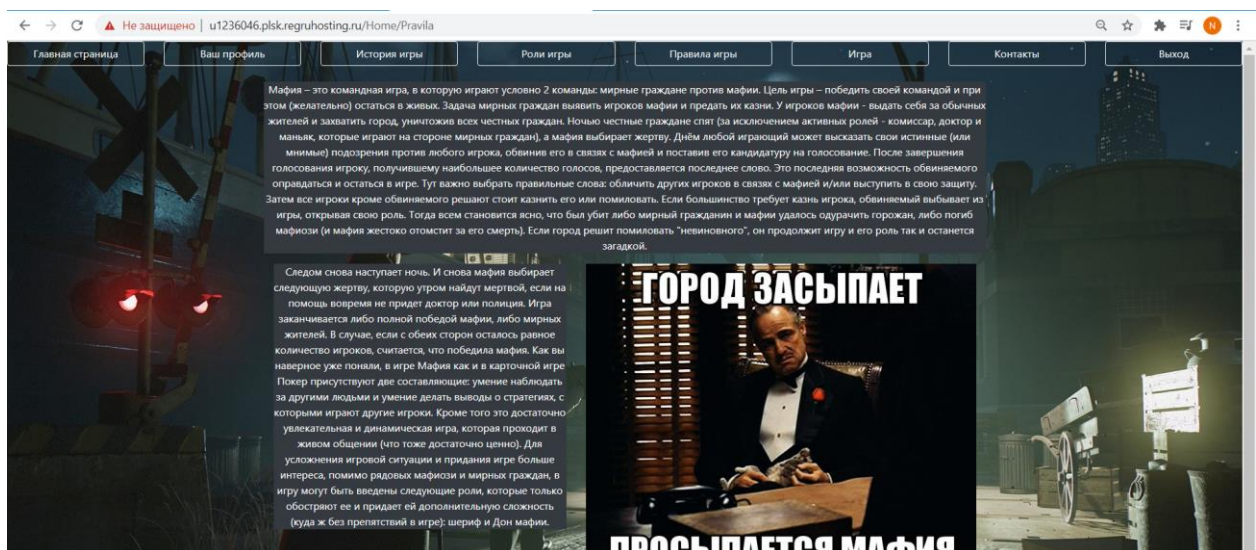


Рисунок 3.7 – Страница правил игры

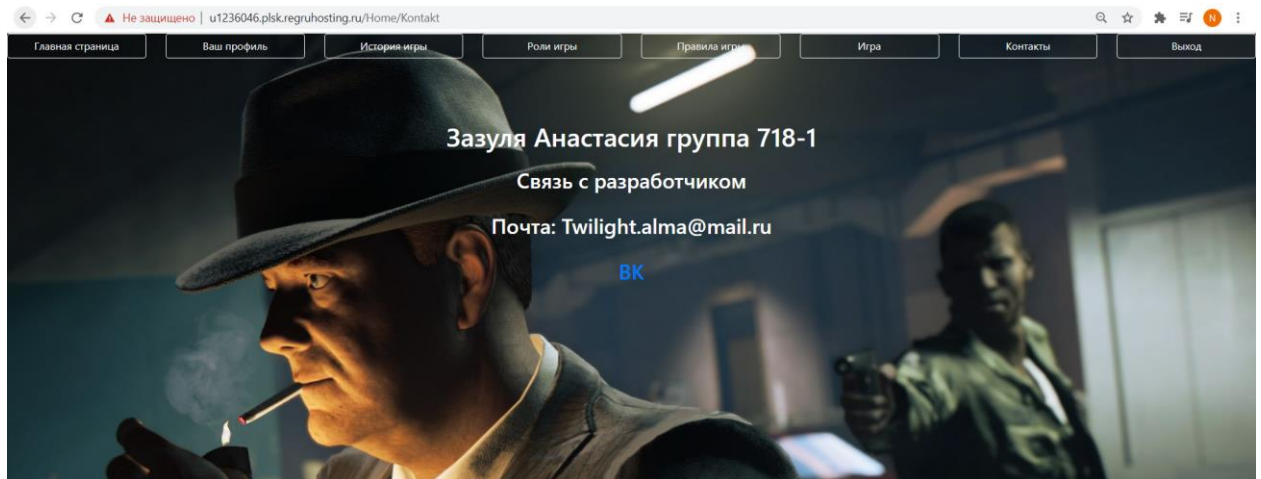


Рисунок 3.8 – Страница связи с разработчиком

Так же авторизированный пользователь может войти на игровую платформу нажав на кнопку «Игра». (Рисунок 3.9)

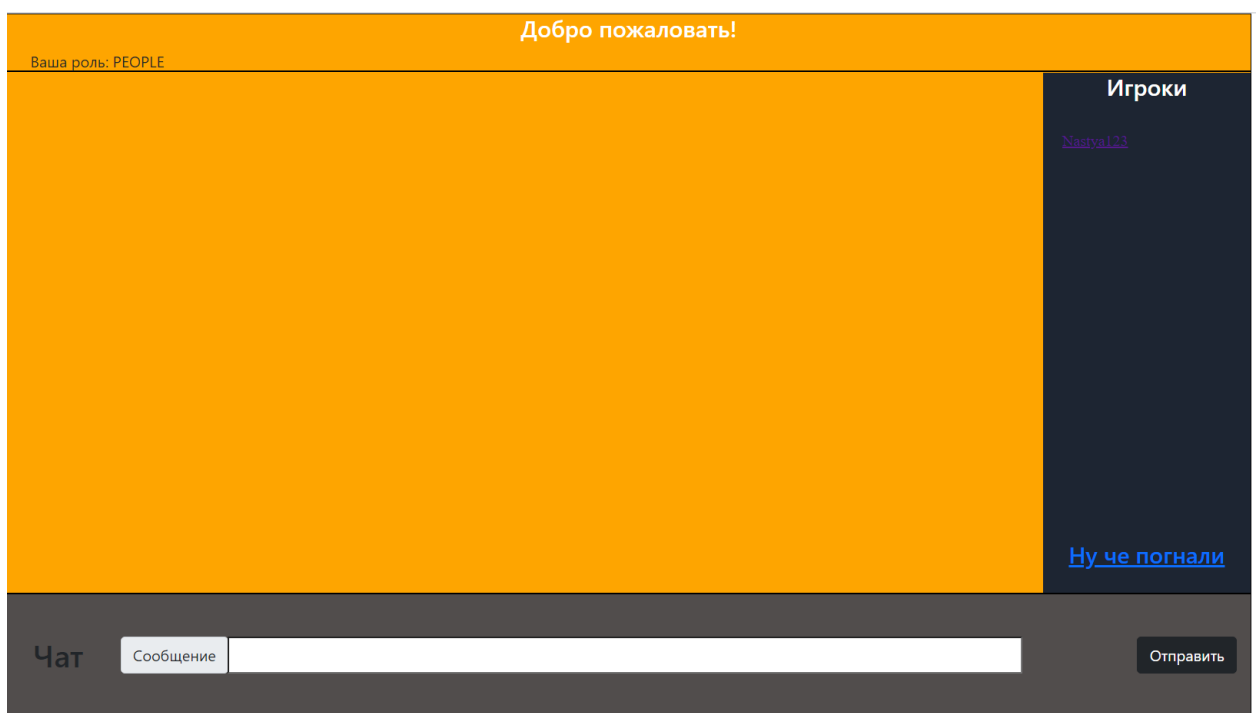


Рисунок 3.9 – Игровая платформа

На игровой платформе можно увидеть кнопку «Ну че погнали» нажав на нее начнется игровой процесс.

Так же на сайте имеется кнопка «Выход» - нажав на кнопку пользователь выходит из системы без возврата обратно и прощается на старницу авторизации. (Рисунок 3.10-3.11)

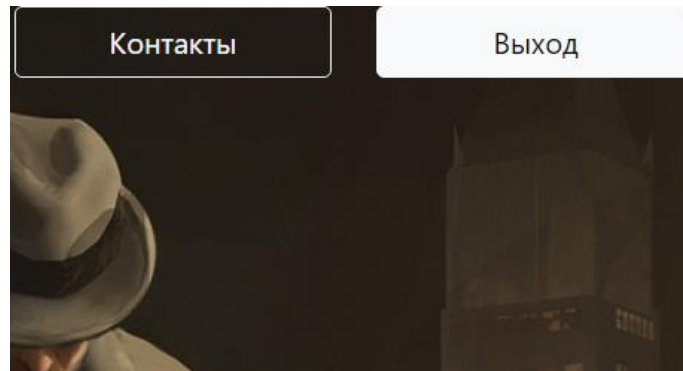


Рисунок 3.10 – Кнопка выхода

A screenshot of a login page. At the top, there is a dark banner with the text 'Войдите в систему' (Log in) in white. Below the banner, there are two input fields: 'Логин' (Login) and 'Пароль' (Password). Below the input fields, there are two buttons: 'Войти' (Log in) and 'Зарегистрироваться' (Register).

Рисунок 3.11 – Страница авторизации

Дизайн сайта должен быть выполнен с использованием языка HTML и CSS(вид и структура), при необходимости для создания отдельных графических элементов допустимо использование технологии Bootstrap.

4 Тестирование

4.1 Программное тестирование

Для тестирования проекта, было решение написать модульное тестирование или unit test. Модульное тестирование позволяет проверить на корректность отдельные модули исходного кода программы. Это позволяет достаточно быстро проверить. Не привело ли очередное изменение кода к появлению ошибок. Пример одного из модульного тестирования предоставлен на рисунке 4.1.

```
[SetUp]
Ссылка: 0
public void Setup()
{
    ds = new Mock<PlayersDataSource>();
    playerMock = new PlayerMock(ds.Object);

    players.Add(new Player("Nastya", Roles.PEOPLE, true, 1243, 12, false));
    players.Add(new Player("Nastya1", Roles.Looser, true, 1623, 12, false));
    players.Add(new Player("Nastya2", Roles.PEOPLE, true, 1123, 12, false));
}

[Test]
Ссылка: 0
public void Test1() //Проверка, что ожидаемое значение верно
{
    Player actual = new Player("Max", Roles.SHERIF, true, 1221313, 12, false);
    players.Add(actual);

    ds.Setup(p => p.getPlayers()).Returns(players);

    Player expected = playerMock.RolPlayer(actual.Role);

    Assert.AreEqual(expected, actual);
}
```

Рисунок 4.1 – Пример модульного тестирования

На рисунке 4.2 результат тестирования.

Обозреватель тестов				Сводка по группе	
Тестирование	Длительность	Признаки	Сообщение об ошибке	Tests	
✔ NUnitTestProject1 (2)	616 мс			Тесты в группе: 1	
✔ NUnitTestProject1 (2)	616 мс			⌚ Общая длительность: 610 мс	
✔ Tests (1)	610 мс			Результаты	
✔ Test1	610 мс			✔ 1 Пройден	
✔ Tests1 (1)	6 мс				
✔ Test2	6 мс				

Рисунок 4.2 – Результат тестов

5.1 Настройка репозитория

Git – это одна из самых известных систем контроля версии с открытым исходным кодом, на которую полагаются миллион проектов по всему миру. Git – это полностью бесплатное программное обеспечение, поддерживает множество ОС, такие как Mac, Linux, Windows и Solaris. []

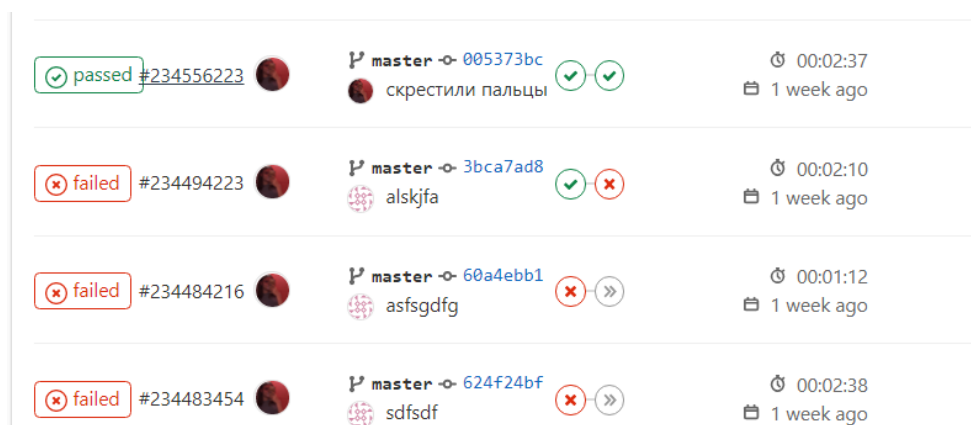
GitLab – это система управления репозиториями программного кода для Git. GitLab предлагает решение для хранения кода и совместной разработки масштабных программных проектов. Репозиторий включает в себя систему контроля версий для размещения различных цепочек разработки и веток. []

Разработка данной курсовой работы, проводилось с использованием репозитория GitLab, сделано это было с целью отслеживания выполняемых действий, а также проведения автоматизированного тестирования

после загрузки данных в репозиторий, GitLab выполнял какие либо действия – нужно создать в папке с проектом файл «.gitlabci.yml» и описать в нем действия, которые произойдут после обновления данных в репозитории.

После обновления проекта на GitLab, начинается pipelines, которые запускаются ранее написанные тесты.

Результат прохождения pipelines. (Рисунок 5.1)















passed #234556223	 master -> 005373bc скрестили пальцы	 	⌚ 00:02:37 📅 1 week ago
failed #234494223	 master -> 3bca7ad8 alskjfa	 	⌚ 00:02:10 📅 1 week ago
failed #234484216	 master -> 60a4ebb1 asfsgdfg	 	⌚ 00:01:12 📅 1 week ago
failed #234483454	 master -> 624f24bf sdfsdg	 	⌚ 00:02:38 📅 1 week ago

Рисунок 5.1 – Результат прохождения pipelines

Более детальная информация пройденного pipelines. (Рисунок 5.2)

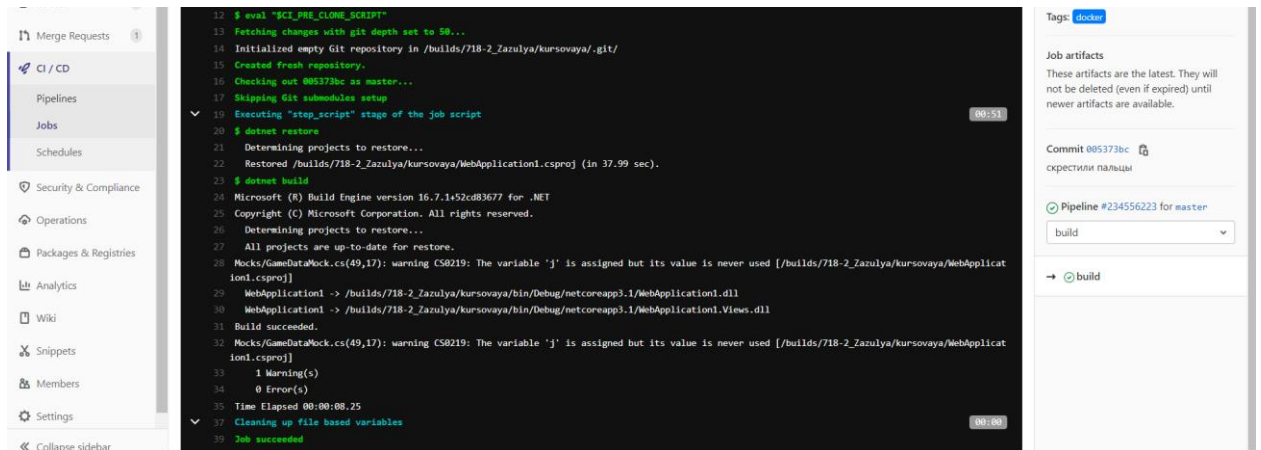


Рисунок 5.2 – Детальная информация пройденного pipelines

В случае неудачного прохождения pipelines, на почту разработчика приходит письмо с небольшим кусом информации, по котором можно догадаться из-за чего произошла ошибка.

6 Заключение

В результате выполнения курсовой работы был разработан сайт для игры в «Мафия».

Для достижения поставленных целей были решены следующие задачи:

1. Написать программу, которая будет взаимодействовать с пользователями.
2. Необходимо реализовать страницу регистрации и авторизации.
3. Необходимо реализовать игровую платформу, где будет производиться игровой процесс для пользователей.
4. Необходимо предоставить возможность выхода из веб-приложения авторизованному пользователю.

Поставленные задачи были проанализированы и решены. Решение данных задач было реализовано на языке C# в среде разработки .NET Core 3.1. Для сохранения данных была подключена СУБД Microsoft SQL Server. Были проведены тестирования приложения.

Благодаря написанию курсовой работы овладела навыками разработки, документирования, освоена работа с git, разработки клиент-серверных приложений, тестирования и отладки программного обеспечения в соответствии с современными технологиями и методами программирования; навыками разработки программной документации; навыками программирования с использованием эффективных реализаций структур данных и алгоритмов.

Список использованных источников

1. ОС ТУСУР [Электронный ресурс] / Режим доступа: <https://regulations.tusur.ru/documents/70> (дата обращения: 28.11.2020) ;
2. SQLite Tutorial [Электронный ресурс] / Режим доступа: <https://www.w3schools.com/sql/> (дата обращения: 28.11.2020) ;
3. Git [Электронный ресурс] / Режим доступа: <https://www.hostinger.ru/rukovodstva/osnovi-git-cto-takoe-git> (дата обращения: 12.12.2020) ;
4. Gitlab [Электронный ресурс] / Режим доступа: <https://itglobal.com/ru-ru/company/glossary/gitlab/> (дата обращения: 12.12.2020);
5. Репозиторий с курсовой работой [Электронный ресурс] / Режим доступа: https://gitlab.com/Срgrach/course_work_timp (дата обращения: 12.12.2020) ;
6. Репозиторий с курсовой работой [Электронный ресурс] / Режим доступа: https://gitlab.com/718-2_Zazulya/kurovaya (дата обращения: 12.12.2020) ;

Приложение А

(Обязательное)

Компакт-диск

Компакт-диск содержит:

- электронную версию пояснительной записки в форматах *.pdf;
- архив с исходным кодом приложения в формате .zip.