Содержание

[Введение 3](#_Toc133011820)

[1 Постановка цели и формулировка задач 4](#_Toc133011821)

[2 Аналитический обзор литературы и постановка задачи 5](#_Toc133011822)

[2.1 Обзор прототипов 5](#_Toc133011823)

[2.2 Приложение «Betera» 5](#_Toc133011824)

[2.3 Приложение «Fonbet» 6](#_Toc133011825)

[2.4 Приложение «MaxLine» 7](#_Toc133011826)

[2.5 Вывод по разделу 8](#_Toc133011827)

[3 Проектирование веб-приложения 9](#_Toc133011828)

[3.1 Проектирование структурной схемы веб-приложения 9](#_Toc133011829)

[3.2 Диаграмма вариантов использования 10](#_Toc133011830)

[3.3 Выбор средств реализации 12](#_Toc133011831)

[3.3.1 Основные языки программирования 12](#_Toc133011832)

[3.3.2 Фреймворки 12](#_Toc133011833)

[3.3.3 Система управления базами данных 13](#_Toc133011834)

[3.3.4 Логическая схема базы данных 13](#_Toc133011835)

[3.4 Вывод по разделу 19](#_Toc133011836)

[Заключение 20](#_Toc133011837)

[Список используемых источников 21](#_Toc133011838)

Введение

Предприятие ООО «Палессит» зарегистрировано Минским городским исполнительным комитетом 2 апреля 2018 г. Компания не имеет в своем составе филиалов и представительств.

Технологической специализацией Компании является разработка web-ориентированного программного обеспечения, веб-сайтов. Сфера применения разработок компании включает такие отрасли, как логистика, онлайн-обучение, развлечения и др. Компания разрабатывает на аутсорсинге высоконагруженные веб-сервисы и приложения для компаний и стартапов из Германии, Англии, США.

Штат компании на данный момент насчитывает до 50 человек, но в планах – расширение штата и привлечение в команду молодых, высококвалифицированных и амбициозных специалистов.

Целью дипломной практики является разработка веб-приложение для букмекерских ставок на футбольные события.

1. Постановка цели и формулировка задач

Нашей задачей стоит разработка веб-приложения для ставок на футбольное событие. Это значит нам надо разработать приложение, которое представляется пользователя как набор страниц, просматриваемых с помощью браузера, между которыми присутствует возможность перехода. Веб-приложение должно взаимодействовать с базой данных, которая будет хранить всю информацию о пользователях и матчах. Приложение будет иметь клиент-серверную архитектуру, связь между ними будет осуществляется с помощью протокола HTTP.

Целью программного средства является реализация возможность игроку пополнять личный счет, ставить на различные футбольные события. Также приложение должно позволять букмекерам выставлять исходы на футбольные события, а также принимать денежные ставки игроков. В приложение еще будет присутствовать роль модератора, которая нужна для верификации пользователя.

Проект должен обладать следующими задачами:

* создание футбольных матчей;
* авторизация и регистрация пользователей;
* верификация пользователей
* добавление исходов футбольных событий;
* прием ставок от игроков;
* просмотр истории ставок.

Так же проект должен обладать бизнес-целями. Проект создается для коммерческой продажи, и одной из целей дипломной работы является расчет и получение прибыли с продажи проекта.

1. Аналитический обзор литературы и постановка задачи

Главная задача приложения состоит в том, чтобы обеспечить пациентам ускоренный вариант записи на приём и вызова врача на дом, вместо долгих очередей и длительных звонков в клинику, а также в предоставлении врачам и регистратору возможности быстрого взаимодействия друг с другом, что позволяет значительно оптимизировать рабочий процесс. Приложение такого характера не является чем-то уникальным в просторах Интернета. Наличие аналогов в данной сфере предоставляет пациентам право выбора: в какую клинику обратиться и как это сделать.

* 1. Обзор прототипов

На сегодняшний день существует достаточно много медицинских веб-приложений. Рассмотрим некоторые из них и выявим сильнее и слабые стороны. Для обзора схожих по тематике ресурсов были выбраны следующие веб-приложения:

* *Betera* [1];
* *Fonbet* [2];
* *MaxLine* [3].
  1. Приложение «Betera»

Первым рассматриваемым проектом среди конкурентов является приложение «Betera». Пример главной страницы продемонстрировано на рисунке 1.1 ниже.

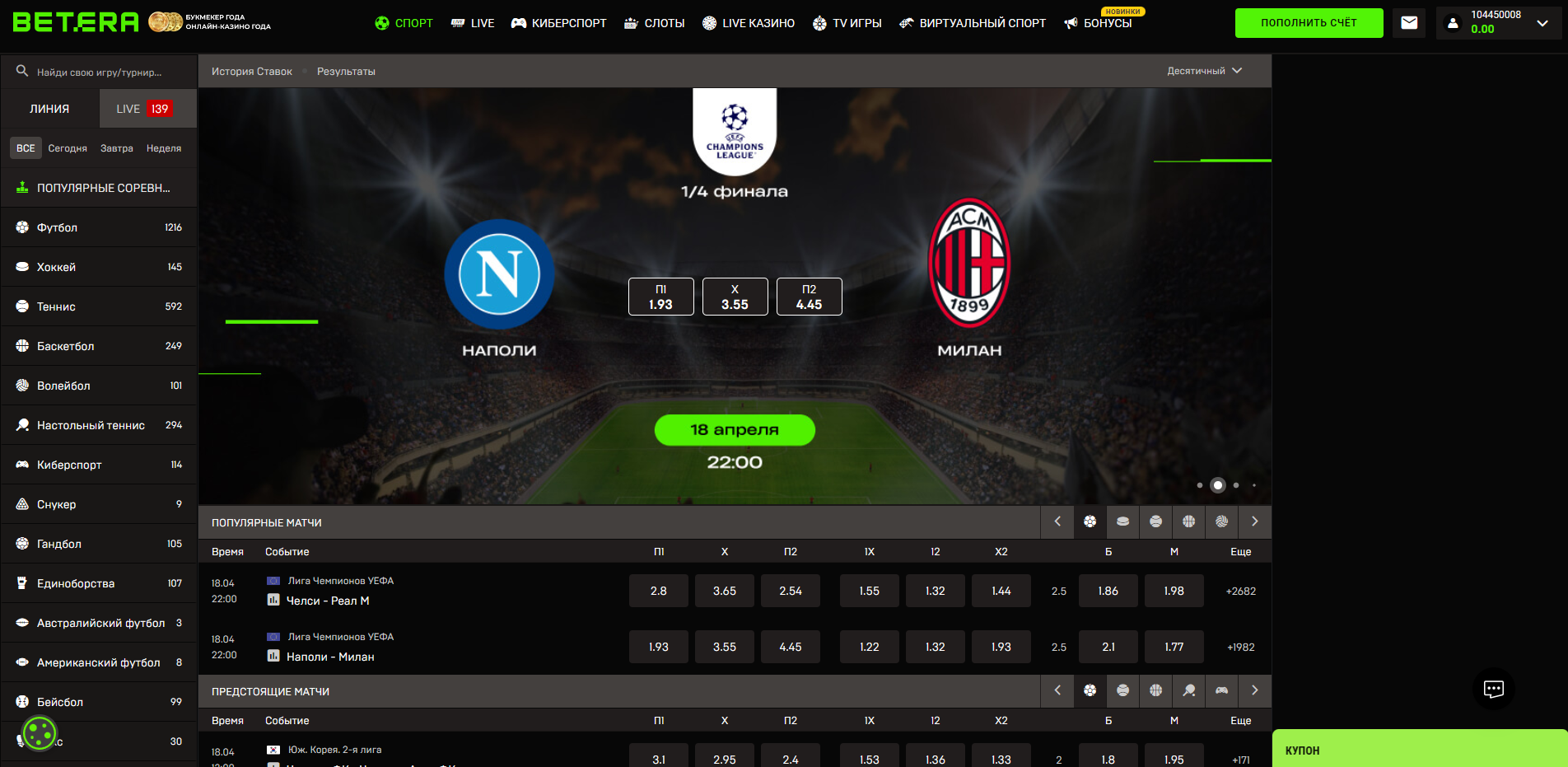


Рисунок 1.1 – Приложение «Betera»

«Betera» является одним из самых популярных приложений. В этом приложении большой выбор ставок для игроков, а также удобный интерфейс. На главной странице мы видим сразу же главные матчи дня. Сразу указываются основные исходы. С левой стороны находится навигация по странам и чемпионатам. Все матчи сегодняшнего дня разделены на популярные матчи и остальные. Также мы можем на главной странице ввести купон для получения бонусов в приложении. В правом верхнем углу мы можем пополнить наш счет, а также увидеть наш баланс. Как мы видим во всех матчах сразу доступна основная линия.

В приложение страница представлена история ставок. На ней доступны все ставки, которые игрок делал ранее. Мы также можем отсортировать по дате, выиграл игрок или поиграл, а также когда ставка была сделана. Так же представлена вся информация о ставке, когда она была сделана, на какое событие, коэффициент события, возможный выигрыш.

На странице матча мы можем увидеть все представленные исходы. Представлены исходы предыдущих встреч этих команд. На странице мы можем сделать ставку на любой исход. В режиме реального времени будут обновляться коэффициенты, а также вся статистика встречи.

Исходя из описания можно сделать краткие выводы: это удобное приложение для игроков, которые дает огромный выбор для ставок. Игрок может ознакомится со своей историей ставок, а также легко пополнить свой баланс. Из минусов можно отметить темный дизайн, который может понравится не всем пользователям.

* 1. Приложение «Fonbet»
* Следующее приложение, которое мы рассмотрим является приложение «Fonbet» [2]. Пример главной страницы представлено на рисунке 1.2 ниже.

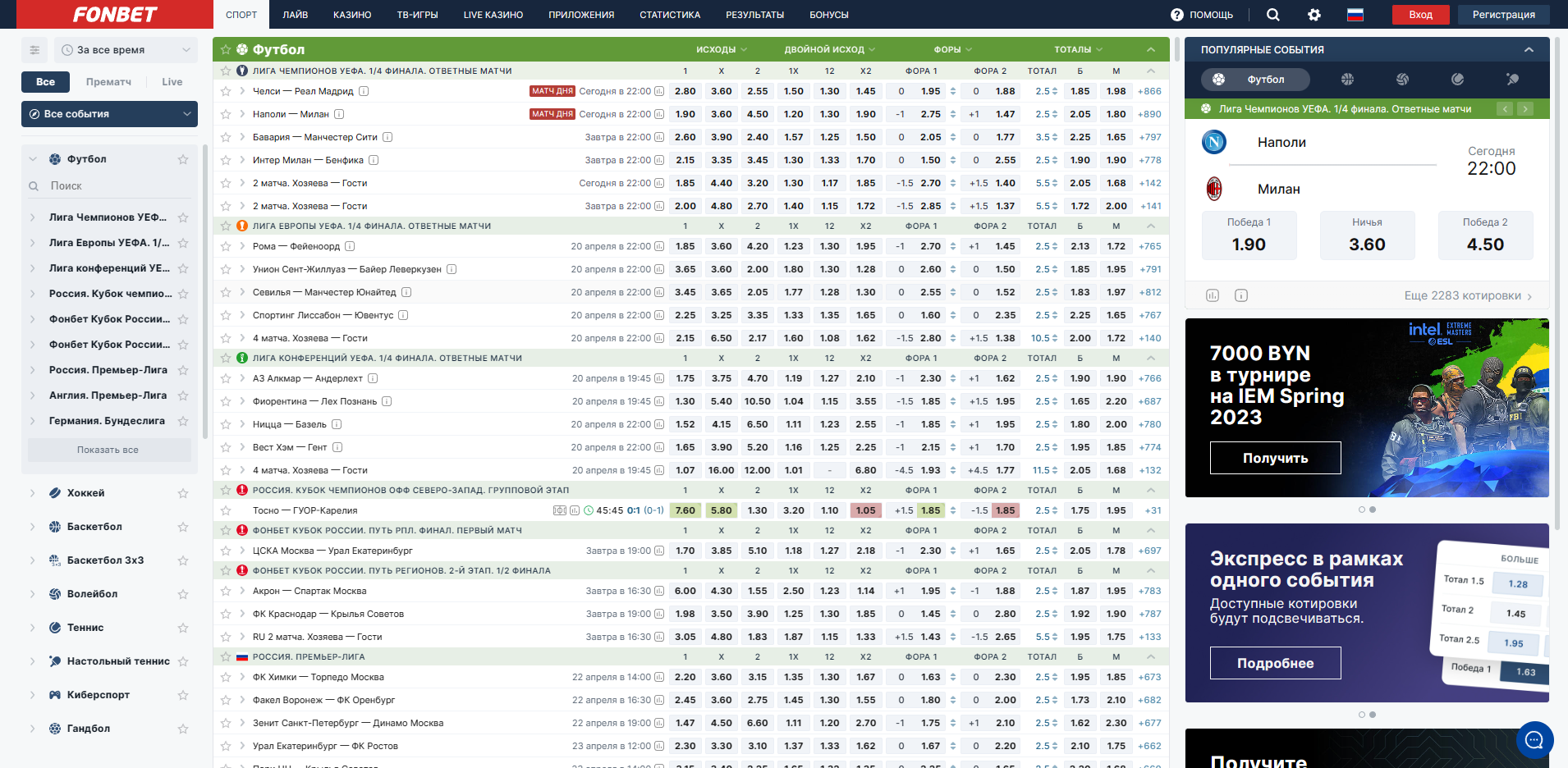


Рисунок 1.2 – Приложение «Fonbet»

Также, как и в предыдущем приложении на главной странице мы видим все матчи. Первое отличие, которое мы видим это то что показаны не только матчи на сегодня, а также на ближайшую неделю. Возле них сразу представлены основные исходы и можно сделать ставку, не переходя на страницу матча. Справа мы видим также основные матчи сегодняшнего дня и основные коэффициенты. В отличие от первого приложения у нас матчи не сортированы как на популярные и не популярные, а идут все подряд. Как и в первом приложении мы можем выбрать турнир на панели слева.

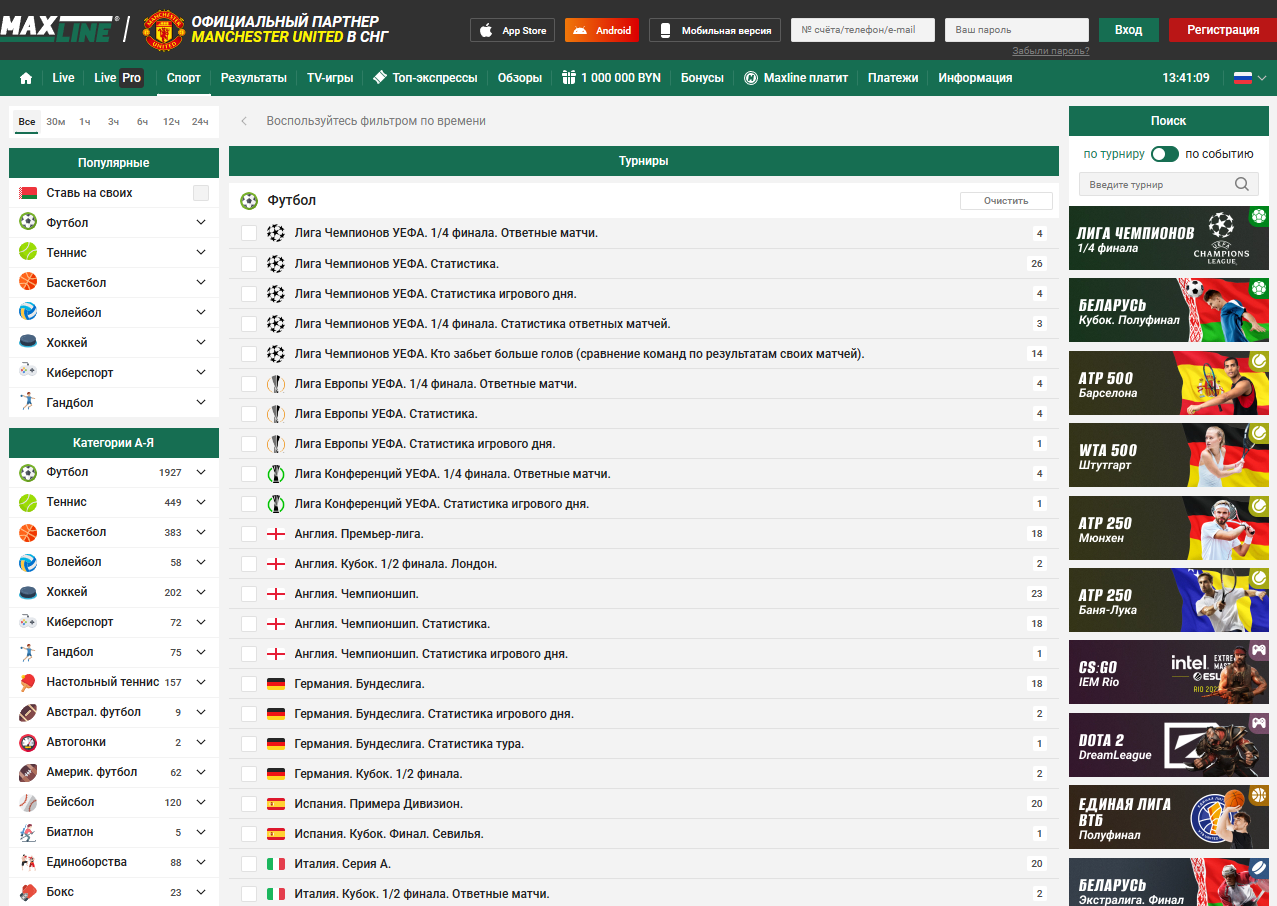
Здесь присутствует история ставок, на который мы так же можем посмотреть все ставки, сделанные игроком, их коэффициент и возможный выигрыш. Так же мы можем посмотреть история пополнения баланса, будем указана внесенная сумма и дата поступления.

На странице матча указаны все исходы, которые для удобства разделены на разные вкладки. Не указаны последние встречи этих команд между собой, но указаны формы команд с которой они подходят к этой встрече. В режиме реального времени будут обновляться коэффициенты, а также вся статистика встречи. В отдельном окне мы можем посмотреть более глубокую статистику команд, состав, и тренеров.

Подводя итоге этого приложения, мы можем сказать, что оно более информативно для пользователя, так как мы можем посмотреть более обширную статистику по командам. Из-за того, что матчи представлены более компактно, мы можем увидеть больше информации на одной странице. Из минусов можно отметить не своевременное обновление информации о матче, а также долгий вывод средств.

* 1. Приложение «MaxLine»

Третьим и последним сравнение аналогом станет приложение «MaxLine». Главная страница указана на рисунке 1.3 ниже.

* 
* Рисунок 1.3 – Приложение «MaxLine»

Приложение «MaxLine» уже более 10 лет находится на рынке. На главной странице в отличии от двух предыдущих аналогов на показываются все действующие турниры. На главной странице так же не указывается главные матчи сегодняшнего дня, что менее удобно чем в других аналогов. При выборе турнира мы видим матчи, которые состоятся в ближайшее время, пример на рисунке 1.4.

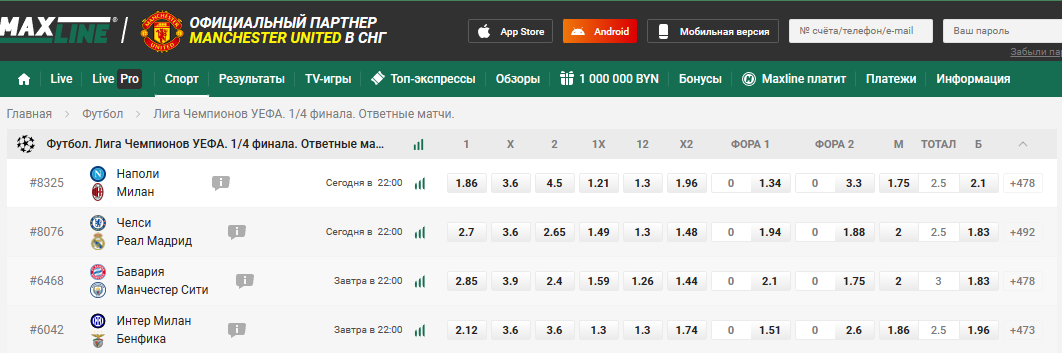


Рисунок 1.4 – Список матчей

Как и в остальных приложениях сразу указывается основные исходы матчей. При выборе матча на раскроется вкладка со всеми исходами. Из информации мы можем узнать только последние несколько игр команд.

В приложении мы можем посмотреть нашу историю ставок. Ее можно отсортировать по времени. Можно пополнить счет личного кабинета и посмотреть историю его пополнения.

Если говорить о приложении, то можно выделить более понятный и удобный интерфейс, который понравился большинству пользователей. Из минусов можно выделить то оно содержит мало информации, а самом матче, а также отсутствие отдельной странице, для сортировки исходов.

* 1. Вывод по разделу

В данном разделе были поставлены задачи, описаны обзоры аналогичных решений на рынке, был проведен патентный поиск по теме дипломного разрабатываемого программного модуля

В ходе обзора трех аналогов разрабатываемого приложения были отмечены достоинства и недостатки каждого из проектов, выбраны положительные стороны для дальнейшей разработки и учтены недостатки. Дополнительно, на основании существующих дизайнов проектов конкурентов, определены точные критерии для создания дизайна приложения, позволяющего осуществлять наиболее простое и интуитивное использование приложения.

1. Проектирование веб-приложения

Проектирование программного средства – важная задача в процессе работы над приложением, потому что в зависимости от нее определяется уровень зависимости между компонентами приложения, и то, насколько легко его можно будет расширить и масштабировать. Хорошо продуманная архитектура веб-приложения может справляться с различными нагрузками и умело адаптироваться к изменяющимся бизнес-требованиям, обеспечивая быстрое взаимодействие с клиентами. Это, разумеется, повышает производительность приложения.

* 1. Проектирование структурной схемы веб-приложения

Структурная схема веб-приложения – это графическое представление архитектуры приложения, показывающее его основные компоненты и связи между ними. Она помогает разработчикам лучше понимать архитектуру приложения и оптимизировать его работу. Она также обеспечивает более эффективное управление проектом и позволяет более эффективно распределять задачи между разработчиками.

Для визуализации элементов и компонентов программы была разработана структурная схема приложения, представленная на рисунке 2.1.

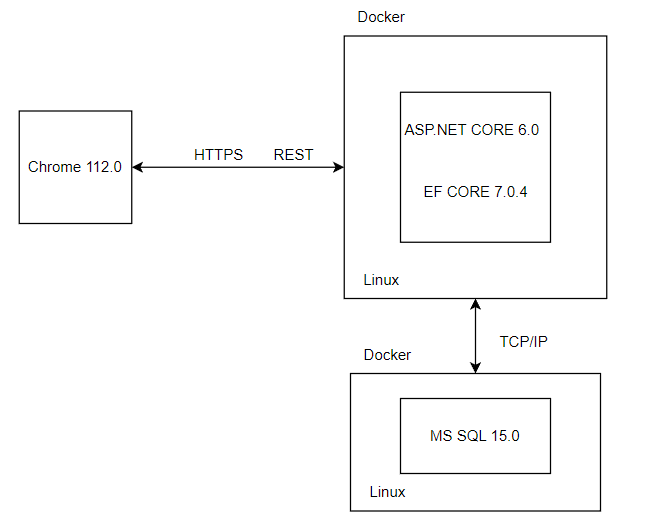


Рисунок 2.1 – Структурная схема веб-приложения

Данная схема показывает все компоненты веб-приложения и показывает взаимодействие между ними.

* 1. Диаграмма вариантов использования

Диаграмма вариантов использования нужна для того, чтобы представить проектируемую систему в виде множество клиентов с определенными ролями взаимодействующие с системой с помощью вариантов использования. Диаграмма вариантов использования для администратора представлена на рисунке 2.2.

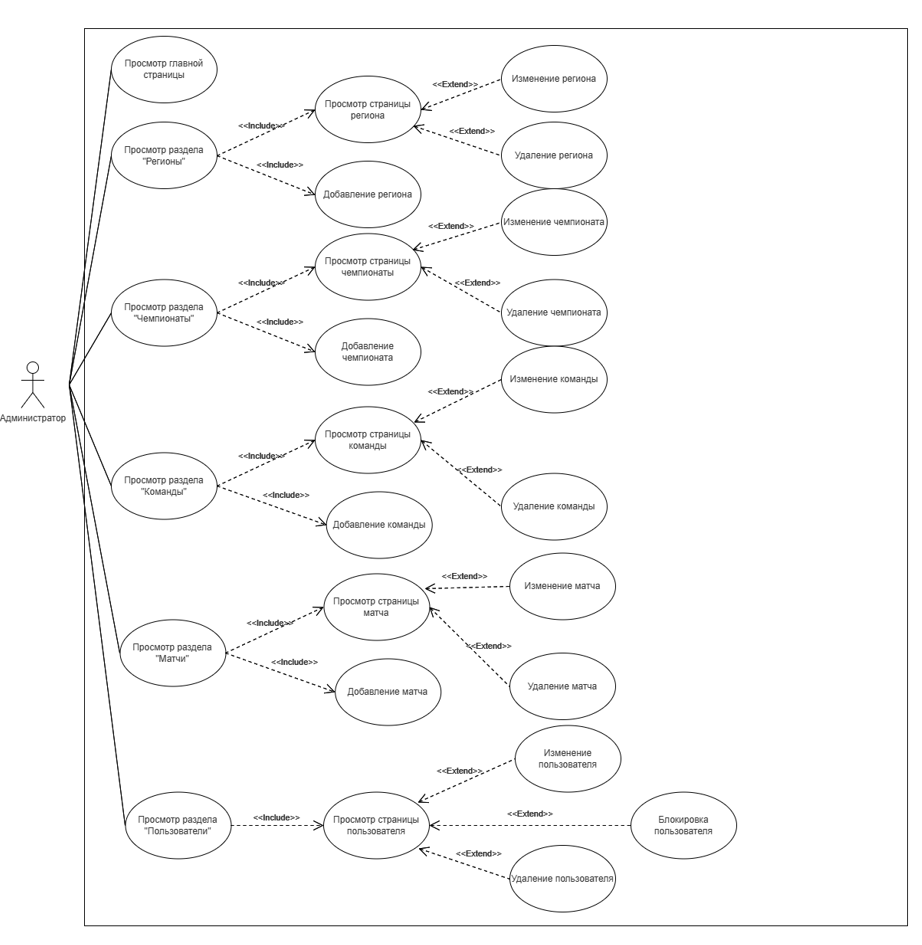


Рисунок 2.2 – Диаграмма вариантов использования

Администратор отвечает за добавление регионов, чемпионатов, команд и матчей. Также он может их изменять или удалять. Всеми зарегистрированными пользователями тоже может управлять администратор, менять им роли, удалять или блокировать. Диаграмма вариантов использования для букмекера представлена на рисунке 2.3.

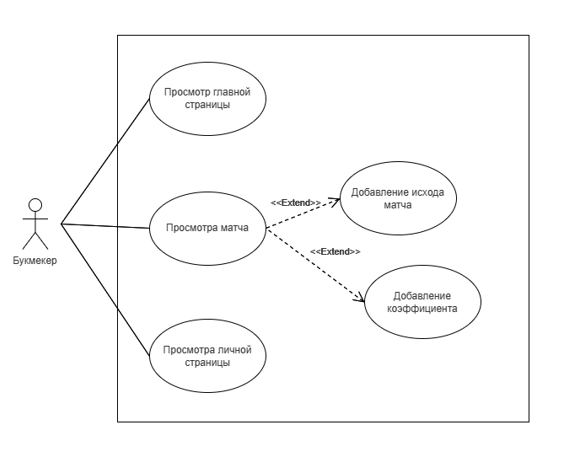


Рисунок 2.3 – Диаграмма использования для букмекера

Букмекеру доступна возможность добавления исходов, а также добавление коэффициентов на матч. Диаграмма вариантов использования для гостя и авторизированного пользователя представлена на рисунке 2.4.

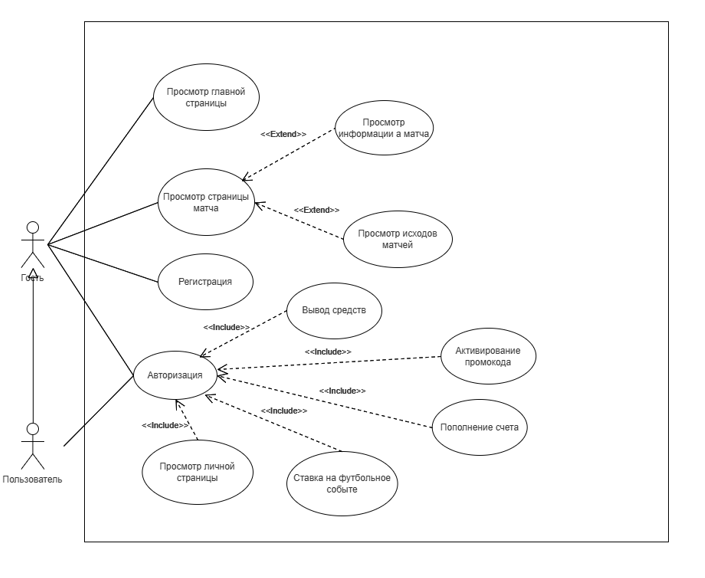


Рисунок 2.4 – Диаграмма использования для пользователя

Для многих действий пользователю необходимо авторизоваться в приложении. Для пополнения счета, активирование промокода, просмотра личной страницы и ставить на футбольное событие.

* 1. Выбор средств реализации

При разработке приложения необходимо подобрать языки программирования и фреймворки, для клиентской и серверной части, которую наилучшим образом соответствуют требуемому функционалу. При выборе правильного языка программирования, фреймворка и системы управления базами данных оказывает значительное влияние на проект, его сроки и сложность реализации.

* + 1. Основные языки программирования

Перед началом разработки приложения необходимо выбрать языки программирования для написания клиентской и серверной части.

C# - современный объектно-ориентированный язык программирования [4]. На сегодняшний момент язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли. В настоящий момент на нем пишутся самые различные приложения: от небольших десктопных программок до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей. Также огромным плюсом является то, что данный язык программирования получает обновления и поддерживается компанией Microsoft на протяжении долгих лет.

После выбора языка программирования для серверной части необходимо выбрать язык программирования для клиентской части.

TypeScript – язык программирования, представленный компанией Microsoft в 2012 году [5]. TypeScript - это расширенная версия JavaScript. То есть он содержит в себе все то же самое, что и JavaScript, но с некоторыми дополнениями. Главная причина использовать TypeSciprt - это возможность добавить статическую типизацию к JavaScript. Тип переменной со статической типизацией не может быть изменен после ее объявления. Это может предотвратить большое количество багов.

* + 1. Фреймворки

Фреймворк – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Как правило на основе фреймворков создаются приложения.

Для разработки серверной части веб-приложения был выбран кроссплатформенный фреймворк ASP.NET CORE [6]. ASP.NET Core является кроссплатформенной, высокопроизводительной средой с [открытым исходным кодом](https://github.com/dotnet/aspnetcore) для создания современных облачных приложений, подключенных к Интернету. Данный фреймворк может работать поверх кроссплатформенной среды .NET Core, которая может быть развернута на основных популярных операционных системах: Windows, Mac OS, Linux. И таким образом, с помощью ASP.NET Core мы можем создавать кроссплатформенные приложения. А для развертывания веб-приложения можно использовать традиционный IIS.

Для реализации клиентской части была выбрана библиотека React [7]. React представляет JavaScript-библиотеку для создания пользовательских интерфейсов, написанный на языке JavaScript и предназначенный для создания веб-приложений клиентского уровня.

* + 1. Система управления базами данных

Для хранения данных я использовал Microsoft SQL Server [8]. Microsoft SQL Server – систему управления реляционными базами данных, разработанная корпорацией Microsoft. СУБД позволяет гибко управлять базами данных (БД). С ее помощью можно создавать, модифицировать или удалять записи, отправлять транзакцию — набор из нескольких последовательных запросов на особом языке запросов SQL. MSSQL базируется на языке SQL и поддерживает многочисленные возможности.

* + 1. Логическая схема базы данных

Для разработки веб-приложения в рамках дипломного проекта понадобилась база данных с необходимой конфигурацией сущностей для хранения всей необходимой информации. Как было аргументировано ранее, в качестве системы управления базами данным выступает Microsoft SQL Server.

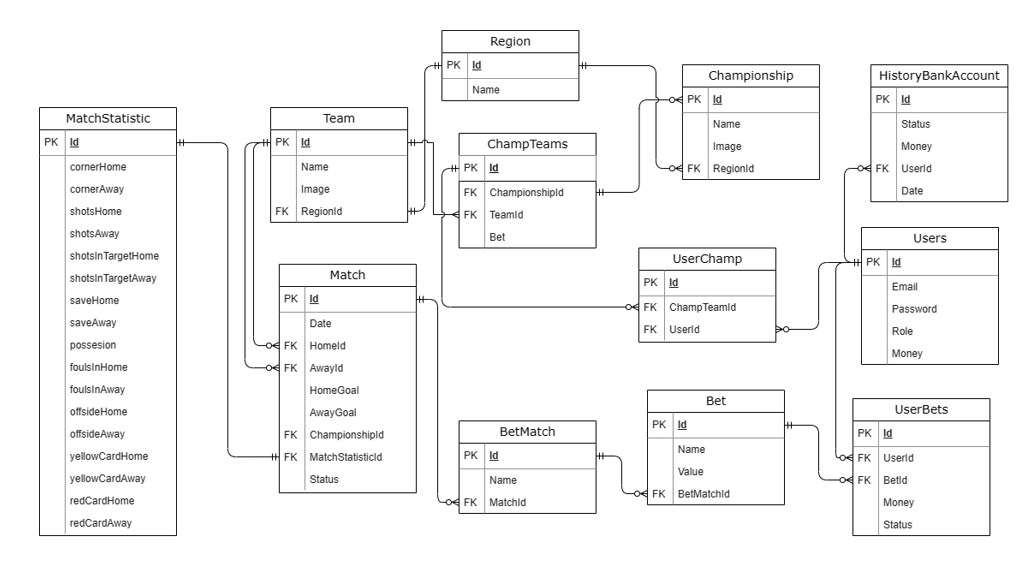


Рисунок 2.5 – Логическая схема базы данных

Далее приведена структура и описание всех таблиц, используемых в веб-приложении.

Таблица «Region» предназначена для хранения регионов. Полное описание данной таблицы представлена в таблице 2.1.

Таблица 2.1 – Структура таблицы «Region»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор региона |
| Name | Nvarchar | Название региона |

Таблица «Team» предназначена для хранения команд. Полное описание данной таблицы представлена в таблице 2.2.

Таблица 2.1 – Структура таблицы «Team»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор команды |
| Name | Nvarchar | Название команды |
| Image | Nvarchar | Ссылка на изображение |
| RegionId | Int | Идентификатор региона, внешний ключ |

Таблица «Championship» предназначена для хранения чемпионатов. Полное описание данной таблицы представлена в таблице 2.3.

Таблица 2.3 – Структура таблицы «Championship»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор чемпионата |
| Name | Nvarchar | Название команды |
| Image | Nvarchar | Ссылка на изображение |
| RegionId | Int | Идентификатор региона, внешний ключ |

Таблица «Users» предназначена для хранения пользователей. Полное описание данной таблицы представлена в таблице 2.4.

Таблица 2.4 – Структура таблицы «Users»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор пользователя |
| Email | Nvarchar | Электронная почта |
| Password | Nvarchar | Пароль пользователя |
| Role | Nvarchar | Роль пользователя |
| Money | Real | Личный счет пользователя |

Таблица «Match» предназначена для хранения матчей. Полное описание данной таблицы представлена в таблице 2.5.

Таблица 2.5 – Структура таблицы «Match»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор матча |
| Date | DateTime2(7) | В |
| HomeGoal | Int | Голы домашней команды |
| AwayGoal | Int | Голы гостевой команды |
| HomeId | Int | Идентификатор домашней команды, внешний ключ |
| AwayId | Int | Идентификатор гостевой команды, внешний ключ |
| ChampionshipId | Int | Идентификатор чемпионата, внешний ключ |
| MatchStatisticId | Int | Идентификатор статистики, внешний ключ |
| Status | Nvarchar | Статус матча |

Таблица «ChampTeams» предназначена для хранения команд, которые участвуют в конкретном чемпионате. Полное описание данной таблицы представлена в таблице 2.6.

Таблица 2.6 – Структура таблицы «ChampTeams»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор чемпионата |
| ChampionshipId | Int | Идентификатор чемпионата, внешний ключ |
| TeamId | Int | Идентификатор команды, внешний ключ |
| Bet | Real | Ставка на команду |

Таблица «UserChamp» предназначена для хранения ставок пользователей, на победу команды в чемпионате. Полное описание данной таблицы представлена в таблице 2.7.

Таблица 2.7 – Структура таблицы «UserChamp»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор записи |
| ChampTeamId | Int | Идентификатор записи, внешний ключ |
| UserId | Int | Идентификатор пользователя, внешний ключ |

Таблица «MatchStatistic» предназначена для хранения статистики матча. Полное описание данной таблицы представлена в таблице 2.8.

Таблица 2.8 – Структура таблицы «MatchStatistic»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор статистики |
| cornerHome | Int | Угловые домашней команды |
| cornerAway | Int | Угловые гостевой команды |
| shotsHome | Int | Удары домашней команды |
| shotsAway | Int | Удары гостевой команды |
| shotsInTargetHome | Int | Удары в створ домашней команды |
| shotsInTargetAway | Int | Удары в створ гостевой команды |
| saveHome | Int | Спасение домашней команды |
| saveAway | Int | Спасение гостевой команды |
| possession | Int | Владение мячом |
| foulsInHome | Int | Фолы домашней команды |
| foulsInAway | Int | Фолы гостевой команды |
| offsideHome | Int | Офсайды домашней команды |
| offsideAway | Int | Офсайды гостевой команды |
| yellowCardHome | Int | Желтые карточки домашней команды |
| yellowCardAway | Int | Желтые карточки гостевой команды |
| redCardHome | Int | Красные карточки домашней команды |
| redCardAway | Int | Красные карточки гостевой команды |

Таблица «BetMatch» предназначена для хранения всех ставок на матч. Полное описание данной таблицы представлена в таблице 2.9.

Таблица 2.9 – Структура таблицы «BetMatch»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор чемпионата |
| Name | Nvarchar | Название ставки |
| MatchId | Int | Идентификатор матча, внешний ключ |

Таблица «Bet» предназначена для хранения коэффициентов на матч. Полное описание данной таблицы представлена в таблице 2.10.

Таблица 2.10 – Структура таблицы «Bet»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор чемпионата |
| Name | Nvarchar | Название исхода |
| Value | Real | Коэффициент |
| BetMatchId | Int | Идентификатор матча, внешний ключ |

Таблица «UserBets» предназначена для хранения коэффициентов на матч. Полное описание данной таблицы представлена в таблице 2.11.

Таблица 2.11 – Структура таблицы «UserBets»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор чемпионата |
| UserId | Int | Идентификатор пользователя, внешний ключ |
| BetId | Int | Идентификатор ставки, внешний ключ |
| Money | Real | Ставка |
| Status | Nvarchar | Статус ставки |

Таблица «HistoryBankAccount» предназначена для хранения истории счета пользователя. Полное описание данной таблицы представлена в таблице 2.12.

Таблица 2.12 – Структура таблицы «HistoryBankAccount»

|  |  |  |
| --- | --- | --- |
| Написание | Тип | Описание |
| Id | Int | Идентификатор чемпионата |
| Status | Nvarchar | Информация о счете |
| Money | Real | Величина перевода |
| Date | DateTime2(7) | Дата |
| UserId | Int | Идентификатор пользователя, внешний ключ |

* 1. Вывод по разделу

В данном разделе были обоснованы выбраны основные языки программирования, фреймворки и СУБД для клиентской и серверной части веб-приложения. Были продемонстрированы диаграммы использования приложения, которые показывают возможные сценарии использования. Кроме того, была предоставлена логическая схема базы данных с описанием таблиц и всех полей.

Заключение

В ходе преддипломной практики было создано веб-приложение букмекерских ставок на футбольное события, была проанализирована предметная область, рассмотрены основные аналоги – их достоинства и недостатки, изучены возможности рассматриваемых программных средств. Были проанализированы и выбраны основные технологии и средства для разработки курсового проекта. На основе выбранных технологий и рассмотренных аналогов была разработана архитектура веб-приложения, а также выбраны инструменты разработки.

В качестве инструмента разработки серверной части была использована платформа *ASP.NET CORE*. Для реализации клиентского приложения использовалась библиотека *React JS*, а также технологии *HTML, CSS, Axios.*

Для хранения, использования и манипулирования данными была спроектирована и реализована база данных *MS SQL*. Для удобной работы с данными была выбрана *ORM*-библиотека *ENTITY FRAMEWORK CORE.* В результате проектирования было разработано 12 коллекций: *MatchStatistic, Team, Championship, Users, Region, Bet, Match, BetMatch,UsersBets, HistoryBankAccount, ChampTeams, UserChamp.*

В результате выполнения преддипломной практики была достигнута поставленная цель: создание простого и удобного в использовании приложения, позволяющего пользователя делать ставки и отслеживать, состояние ставки, букмекер с легкостью и быстрой скоростью выставлять коэффициента, а администраторам создать, изменять и удалять матчи.

Созданная по итогам преддипломной практики программа соответствует основным требованиям и выполняет требуемые функции. Проект является завершенным, присутствует возможность дальнейшего расширения веб-приложения и увеличение его функционала.

Список используемых источников

1. Betera» [Электронный ресурс]. – Режим доступа: <https://pm.by/>. – Дата доступа: 18.04.2023
2. «Fonbet» [Электронный ресурс]. – Режим доступа: https://www.fonbet.by/. – Дата доступа: 18.04.2023
3. «MaxLine» [Электронный ресурс]. – Режим доступа: https://maxline.by. – Дата доступа: 18.04.2023
4. Документация по C# [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/csharp/ – Дата доступа: 20.04.2023
5. TypeScript [Электронный ресурс]. – Режим доступа: https://www.typescriptlang.org – Дата доступа: 20.04.2023
6. Документация по ASP.NET [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-6.0 – Дата доступа: 20.04.2023
7. React JS [Электронный ресурс]. – Режим доступа: <https://ru.reactjs.org> – Дата доступа: 20.04.2023
8. Microsoft SQL Server [Электронный ресурс]. – Режим доступа: <https://www.microsoft.com/en-us/sql-server> – Дата доступа: 20.04.2023