**УО «Белорусский государственный технологический университет»**

Факультет **информационных технологий**

Кафедра **программной инженерии**

Специальность **1-40 05 01-03 «Информационные системы и технологии (издательско-полиграфический комплекс)»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОЙ РАБОТЫ**

**по дисциплине «**Объектно-ориентированное программирование»

**тема «**Программное средство “Хоккейная лига”»

|  |  |  |
| --- | --- | --- |
| **Исполнитель** |  | |
| Студент 2 курса группы3 | подпись, дата | Д. С. Шкабров  инициалы и фамилия |
|  |  |  |
| **Руководитель** |  |  |
| Старший преподаватель | подпись, дата | И. Г. Сухорукова  инициалы и фамилия |

|  |  |
| --- | --- |
| **Курсовая работа защищена с оценкой** | **\_\_\_\_\_\_\_\_\_\_** |
| **Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  (подпись) | И. Г. Сухорукова  инициалы и фамилия |

**Минск 2022**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПЕУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра программной инженерии

Утверждаю

Заведующий кафедрой ПИ

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_ Н.В. Пацей

подпись инициалы и фамилия

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022г.

**ЗАДАНИЕ**

**к курсовому проектированию**

**по дисциплине** "Объектно-ориентированное программирование"

|  |  |
| --- | --- |
| Специальность: 1-40 05 01-03 «Информационные системы и технологии»  Студент: Шкабров Д.С. | Группа:\_\_3\_\_ |
| **Тема: Программное средство «Хоккейная лига»** | |

**1. Срок сдачи студентом законченной работы**: "20 мая 2022 г."

**2. Исходные данные к проекту:**

**2.1**. Функционально ПС поддерживает:

* + Поддерживать работу c базой данных (Команды, даты проведения матчей, тренера и т. д.);
  + Выполнять регистрацию и авторизацию;
  + Заполнять форму заказа билета по заданным формам;
  + Показывать изображения клубов;
  + Оставлять отзывы.

**2.2.** При выполнении курсового проекта необходимо использовать принципы проектирования ООП. Приложение разрабатывается под ОС Windows и представляет собой настольное приложение. Отображение, бизнес логика должны быть максимально независимы друг от друга для возможности расширения. Диаграммы вариантов использования, классов реализации задачи, взаимодействия разработать на основе UML. Язык разработки проекта – C#. Управление программой должно быть интуитивно понятным и удобным. При разработке использовать несколько наиболее подходящих шаблонов проектирования ПО.

**3. Содержание расчетно-пояснительной записки**

(перечень вопросов подлежащих разработке)

* Введение
* Постановка задачи и обзор литературы (алгоритмы решения, обзор прототипов, актуальность задачи)
* Проектирование архитектуры проекта (структура модулей, классов).
* Разработка функциональной модели и модели данных программного средства (выполняемые функции)
* Тестирование
* Заключение
* Список используемых источников
* Приложения

**4. Форма представления выполненной курсовой работы:**

* + Теоретическая часть курсового проекта должны быть представлены в формате docx. Оформление записки должно быть согласно выданным правилам.
  + Листинги программы представляются частично в приложении.
  + Пояснительную записку, листинги, проект (инсталляцию проекта) необходимо загрузить диск, указанный преподавателем.

#### Календарный план

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта | Примечание |
| 1 | Введение | 19.02.2022 |  |
| 2 | Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования | 12.03.2022 |  |
| 3 | Анализ и проектирование архитектуры приложения (построение диаграмм, проектирование бизнес-слоя, представления и данных) | 26.03.2022 |  |
| 4 | Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса | 02.04.2022 |  |
| 5 | Кодирование программного средства | 23.04.2022 |  |
| 6 | Тестирования и отладка программного средства | 30.04.2022 |  |
| 7 | Оформление пояснительной записки | 07.05.2022 |  |
| 9 | Сдача проекта | 20.05.2022 |  |

**5. Дата выдачи задания \_\_\_\_**14.02.2022**\_\_\_\_**

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *И.Г. Сухорукова*

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Д.С. Шкабров*

(дата и подпись студента)

Оглавление

[**Титульник и задание** 2](#_Toc104766932)

[**Введение** 4](#_Toc104766933)

[**1. Аналитический обзор литературы и постановка задачи** 5](#_Toc104766934)

[**1.1Анализ прототипов** 5](#_Toc104766935)

[**1.2 Постановка задачи** 9](#_Toc104766936)

[**2. Анализ и проектирование архитектуры приложения** 10](#_Toc104766937)

[**3.4 Проектирование базы данных** 14](#_Toc104766938)

[4. Создание (реализация) программного средства 17](#_Toc104766939)

[**4.1. Основные классы программного средства** 17](#_Toc104766940)

[**4.2. Описание классов и методов программного средства** 17](#_Toc104766941)

[**4.2.1. Выполнение входа и регистрации** 17](#_Toc104766942)

[**4.2.2. Просмотр Расписания** 18](#_Toc104766943)

[**4.2.3. Заполнение формы бронирования билета** 18](#_Toc104766944)

[**4.2.4. Просмотр Таблицы** 18](#_Toc104766945)

[**4.2.5. Просмотр Результаты** 18](#_Toc104766946)

[5. Тестирование, проверка работоспособности и анализ полученных результатов 18](#_Toc104766947)

[5.1 Мануальное тестирование 19](#_Toc104766948)

[6. Руководство по использованию 20](#_Toc104766949)

[ЗАКЛЮЧЕНИЕ 24](#_Toc104766950)

[**Список литературы** 25](#_Toc104766951)

[**Приложения** 26](#_Toc104766952)

# **Введение**

В наше время очень большое количество приложений могут нам рассказать, показать, научить и т. д. И большинство людей следят за спортом, и возьмем в пример хоккей, фанату невозможно побывать на всех матчах, но для быстрой информативности он может посмотреть результаты матча и заказать билет на матч. Особенно возникает проблема, когда ты следишь за хоккейной лигой, которая проходит на другом континенте и когда у нас ночь, то у них в это время день. Я, как хоккеист, интересуюсь хоккейными матчами в лигах, и болею за любимую команду, поэтому я решил взять тему моего курсового проекта «Хоккейная лига».

В моем приложении будет показ матчей в лиге, заказ билетов на матч, регистрация, авторизация, показ результатов матча, ссылки на видео об интересных моментах в матчах. Пользователям и фанатам хоккея будет интересно вкратце узнать о результатах матча и заказать билет на предшествующий матч. Это приложение будет просто в использовании и информативно расскажет об интересующих вас вещах. Для защиты личных данных в приложении предусмотрена регистрация и авторизация.

Для успешной реализации курсового проекта необходимо:

* провести анализ соответствующей литературы;
* ознакомиться с прототипами программных средств выбранной мной темы;
* определить функциональные требования;
* продумать структуру базы данных;
* продумать структуру проекта;
* реализовать программное средство;
* протестировать программное средство;
* написать руководство пользователя.

# **Аналитический обзор литературы и постановка задачи**

Для определения с установкой, функционалом и примерным дизайном надо окончательно определиться с постановкой задачи моего курсового проекта, поэтому я проанализирую похожие прототипы программных средств выбранной темы.

## **1.1Анализ прототипов**

**Championat** — это распространенная программа, в ней рассказывают итоги и анонсы не только по хоккею, но по другим видам спорта, как футбол, теннис и т.д. Рассказывает новости в спорте и краткие итоги об матчах (счет и кто забросил гол). Когда пользователь только заходит в нее, она может показаться ему очень простой и примитивной, но она имеет богатый функционал.

Плюсы:

* простота в использовании;
* краткие сведения об итогах матчей;
* многофункциональность в использовании (Таблицы, календарь (по месяцам каждый матч), и т.д.);
* использование с телефона, данные синхронизируются между разными устройствами;

Минусы:

* нельзя заказать билеты на матчи;
* присутствие разных лиг в хоккее, вдобавок присутствуют другие виды спорта;

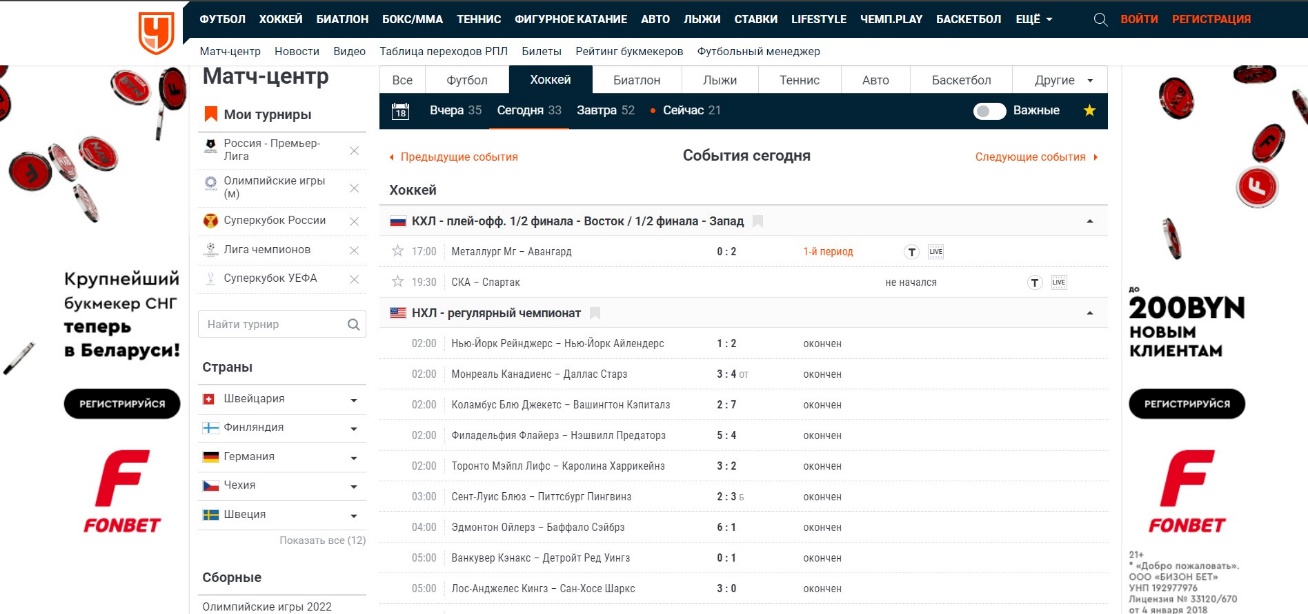


Рисунок 1.1 – Интерфейс приложения «**Championat**»

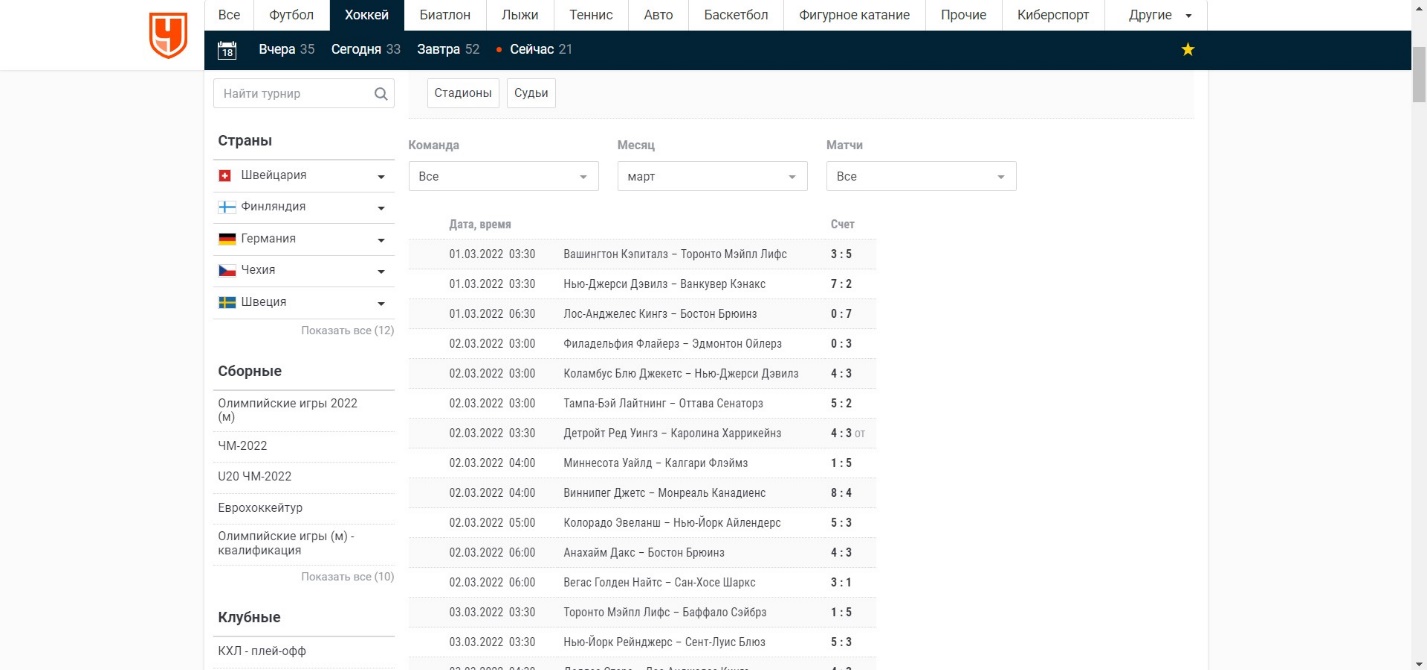


Рисунок 1.2 – Интерфейс приложения «**Championat**»

**NHL** – Официальное приложения для НХЛ (Национальная хоккейная лига), где расписано все о лиге, итогах и заказ билетов на матч. Представляет собой программы для Windows, Mac, Android, iPhone.

Плюсы:

* удобный и простой интерфейс;
* возможность заказа билета;
* обзор прошедшего матча;
* календарь всех матчей;
* анонс предшествующего матча;
* новости в лиге;
* присутствие магазина;

Минусы:

* Пустая главное меню приложения;

На рисунке 1.3 представлены личные задачи, а на 1.4 бизнес задачи.

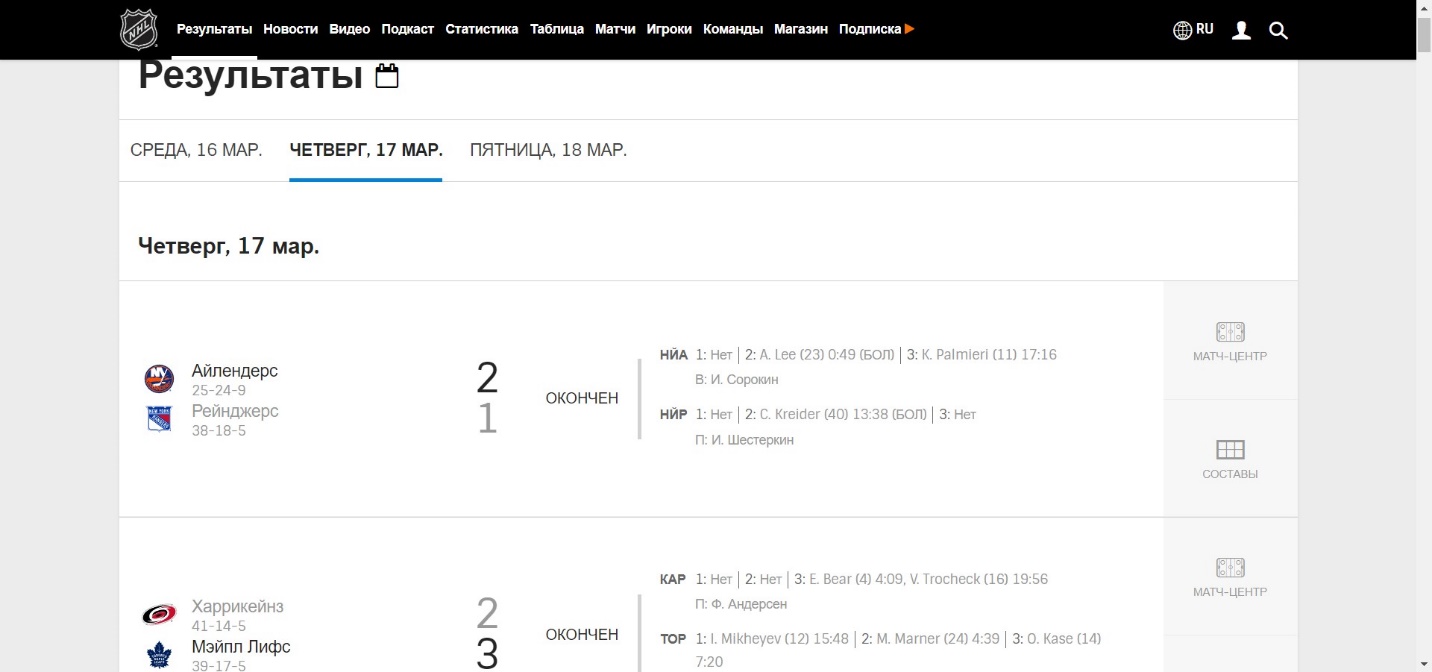


Рисунок 1.3 – Интерфейс приложения «**NHL**»

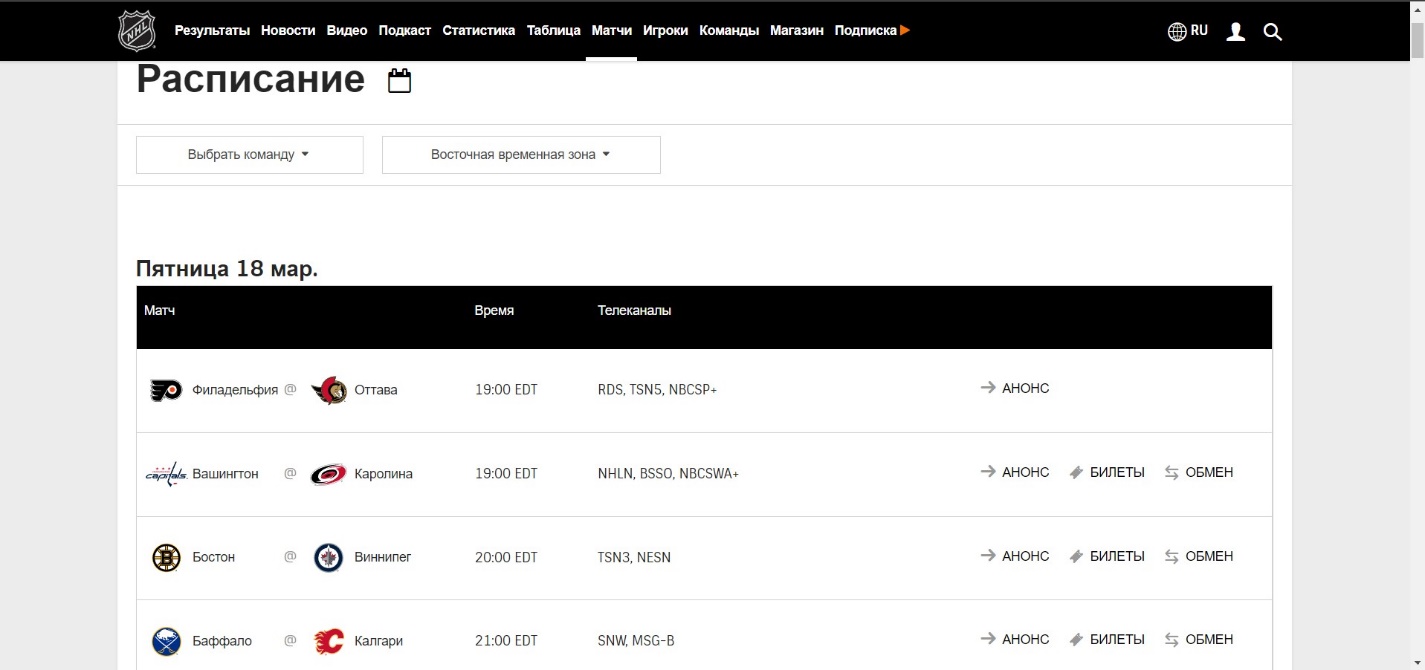


Рисунок 1.4 – Интерфейс приложения «**NHL**»

**КХЛ -** Официальное приложения для КХЛ (Континентальная хоккейная лига), Реализовано почти также как и в предыдущем аналоге (Календарь, покупка билетов на матч, итоги матча и т.д.). Синхронизируется с веб-версией, приложениями для Android, Windows.

Плюсы:

* возможность заказа билета;
* обзор прошедших матчей;
* календарь всех матчей;
* анонс предшествующего матча;
* новости в лиге;
* присутствие магазина;

Минусы:

* + Видны баги;
  + Слишком забитая пустой информацией;

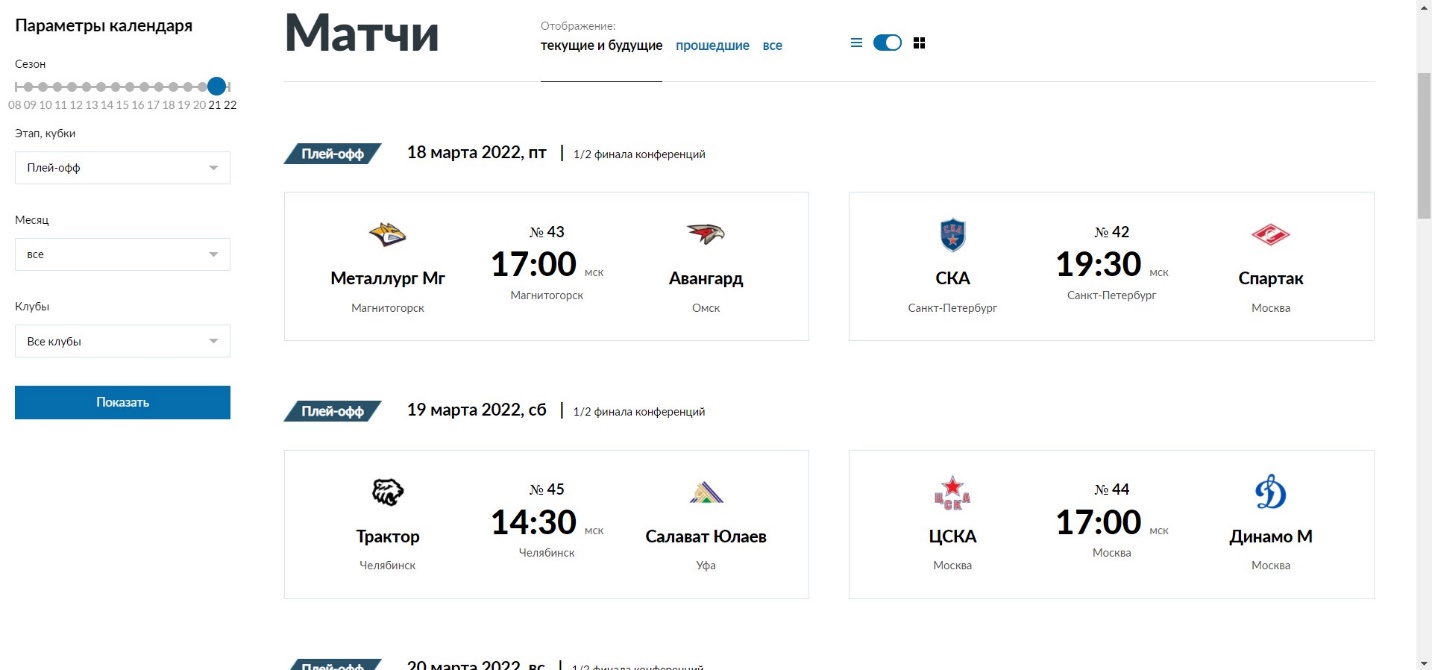


Рисунок 1.5 – Интерфейс приложения «**КХЛ**»

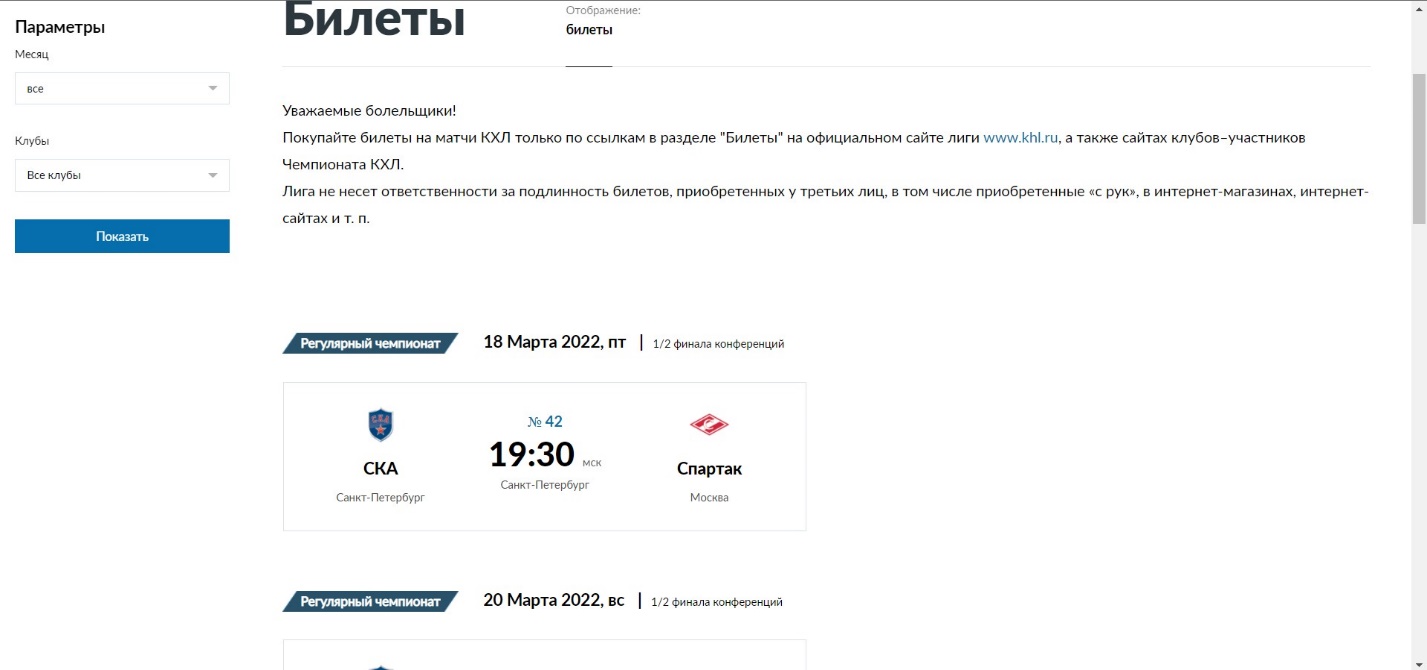


Рисунок 1.6 – Интерфейс приложения «**КХЛ**»

## **1.2 Постановка задачи**

После оценки всех приложений можно сделать вывод, что многие имеют схожий интерфейс и концепцию, но в каждом приложении есть своя изюминка. Просмотрев прототипы приложений для хоккея (хоккейной лиги), я выбрал основную задачу и функциональные требования проекта.

Основной задачей моего курсового проекта является разработка десктопного приложения, позволяющего пользователю просмотреть результаты матчей, заказ билетов на матчи в будущем, просмотр календаря матчей. Программное средство должно содержать профиль авторизованного пользователя с его контактными данными. Интерфейс должен быть простым и удобным для использования.

# **Анализ и проектирование архитектуры приложения**

**2.1 Описание средств разработки**

На предыдущем этапе курсового проекта были рассмотрены несколько аналогов программных средств для работы с фанатами хоккея. Функционал нашего приложения будет включать в себя совокупность основных возможностей аналогов. Стоит учесть все излишки и проблемы рассмотренных аналогов, а также добавить некоторые дополнительные возможности.

Задачей курсового проекта является разработка приложения, предназначенного для показа информации для пользователей, а именно результаты прошедших игр и возможность покупки билета на будущий матч.

Программный продукт реализован на объектно-ориентированном языке C# с использованием технологий WPF.

Windows Presentation Foundation (WPF) — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Графической технологией, лежащей в основе WPF, является DirectX, в отличие от WindowsForms, где используется GDI/GDI+. Производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX.

База данных была разработана на Microsoft SQL Server.

Для организации баз данных MS SQL Server использует реляционную модель, которая предполагает хранение данных в виде таблиц, каждая из которых состоит из строк и столбцов. Каждая строка хранит отдельный объект, а в столбцах размещаются атрибуты этого объекта.

Для взаимодействия с базой данных применяется язык SQL (Structured Query Language).

Клиент (например, внешняя программа) отправляет запрос на языке SQL должным образом интерпретирует и выполняет запрос, а затем посылает клиенту результат выполнения.

Основной используемый язык запросов – Transact-SQL – реализован на структурированном языке запросов (SQL) с расширениями.

Для осуществления связи между базой данных и приложением на C# необходим посредник, которым будет являться технология Entity Framework. Она основана на платформе .NET Framework и предназначена для работы с данными.

Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Эта технология предоставляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Entity Framework предполагает три возможных способа взаимодействия с базой данных:

* database first: Entity Framework создает набор классов, которые отражают модель конкретной базы данных;
* model first: сначала разработчик создает модель базы данных, по которой затем Entity Framework создает реальную базу данных на сервере;
* code first: разработчик создает класс модели данных, которые будут храниться в базе данных, а затем Entity Framework по этой модели генерирует базу данных и ее таблицы.

В курсовом проектировании для разработки был выбран первый подход.

Приложение необходимо реализовать с использованием шаблона проектирования MVVM и с организацией взаимодействия с базой данных. В качестве клиентской части необходимо реализовать оконное приложение с использованием стандартных и иных библиотек. Приложение должно иметь возможность запускаться без использования интегрированных средств разработки.

**2.2 Определение требований к программному средству**

Для разработанного продукта необходимо предусмотреть следующие задачи:

* удобная система навигации;
* интуитивно понятный интерфейс;
* механизм поиска матчей и таблица с результатами;

Дополнительные задачи для администратора:

* возможность редактирования данных об играх и таблицы результатов;

По итогу работы над продуктом, его необходимо протестировать и отладить, проанализировать его возможности и, по необходимости, внести дополнения или улучшения проекта, с возможностью дальнейшего расширения проекта.

Программное средство предоставляет пользователю следующие функциональные возможности:

- регистрация пользователей;

- авторизация пользователей;

- просмотр расписания и результатов матчей;

- просмотр статистики команд в группах

- бронь места на матче.

Администратор имеет следующие возможности:

- изменение данных об матчах и сохранение их в БД;

- добавление результатов матча дописывая и сохраняя их в БД.

Описание функциональности программного средства представлено на UML-схеме, находящийся в графической части.

**2.3 Описание функциональности программного средства**

Для функциональности ПС необходимо создание базы данных для хранения информации приложения. Подробно о БД описано в главе 3.

В программе “Хоккейная лига” при запуске необходимо реализовать регистрацию и авторизацию пользователей для дальнейшего использования приложения. Для авторизации входными параметрами являются имя и пароль пользователя, которые содержатся в базе данных. Для того чтобы зарегистрироваться необходимо ввести имя, почту, пароль и номер платежной карты.

Основной функцией приложения является просмотр обновлений об матчах и возможности покупки билета на любой матч. Пользователь может просмотреть имеющеюся информацию, а также забронировать место на арене. При бронировании места пользователь выбирает дату игры и место с помощью слайдеров.

Таким образом, в ходе работы над этим разделом были сформулированы основные функциональные требования для проектирования программного средства.

**3. Проектирование программного средства**

Проектирование программного средства – процесс создания проекта программного обеспечения. Целью проектирования является определение внутренних свойств системы и детализации её внешних свойств на основе исходных условий задачи. Исходные условия задачи были сформулированы во втором разделе данной пояснительной записки. Этап проектирования подразумевает их анализ.

**3.1 Архитектура системы**

Программное средство «Хоккейная лига» имеет следующую структуру, представленную на рисунке 3.1.

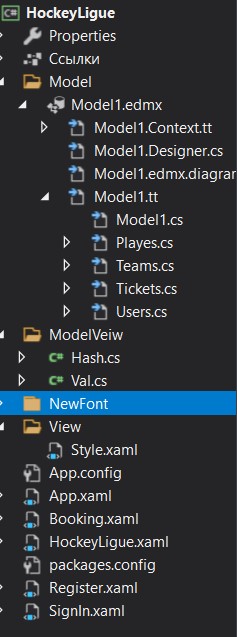


Рисунок 3.1 –Структура проекта

**3.2 Разработка общей схемы приложения**

Общая схема приложения представлена на рисунке 3.2

Рисунок 3.2 – Общая схема приложения

При запуске приложения открывается окно регистрации, где можно зарегистрировать нового пользователя, а также перейти в окно авторизации для входа в существующий аккаунт.

После входа или регистрации открывается главное меню, из которого можно закрыть программу, выйти из аккаунта и перключатся во все оставшиеся страницы приложения

На странице Расписание можно посмотреть содержание всех будущих игр и заказать билет на этот матч. Администратор может изменить время игры.

На странице Таблицы можно посмотреть статистику команд, на каком она месте, сколько очков и сыгранных игр у команд. Администратор может изменять содержимое таблицы, кроме названия команд. При переходе команд мест в группе, просто сохраняются изменения, где после выхода администратора они сохраняются и сортируются по месту в таблице

На странице Результаты можно просматривать итоги всех сыгранных матчей. Администратор может изменять данные о голах команд, с каким счетом они закончили игру.

**3.3 Описание структуры проекта**

Описание структуры проекта приведено в таблице 3.1.

Таблица 3.1 – Описание структурных пакетов проекта

|  |  |
| --- | --- |
| Имя пакета | Описание |
| Model | Здесь описаны модели. |
| ViewModel | Содержит логику, которая позволяет получить данные при помощи View, обработать их, используя при этом Model, после чего передать на сервер. |
| View | Здесь описан визуальный интерфейс |
| Registration, SignIn, Booking | Окна для регистрации, авторизации и бронирования билетов |
| HockeyLigue | Главное окно приложенияя |
| NewFont | Содержит картинки, шрифт приложения |

**3.4 Проектирование базы данных**

Проектирование баз данных – процесс создания схемы базы данных и определения необходимы ограничений целостности.

При проектировании базы данных курсового проекта использовался Entity Framework. Он представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища.

Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ мы можем получать объекты, связанные различными ассоциативными связями.

Логическая модель базы данных представлена в графической части.

Всего в базы данных 4 таблицы, описание которых приведено ниже.

На рисунке 3.2 проиллюстрирована структура таблицы «Playes», которая содержит информацию об играх. В данной таблице поля HOMETEAM и GUESTTEAM, где представлены в nvarchar. Поля HOMEGOALS и GUESTGOALS хранит голы матчей, HOMEIMAGE и GUESTIMAGE – хранит путь до картинок, DATE – хранит дату игры, первичный ключ, TIME — время игры.

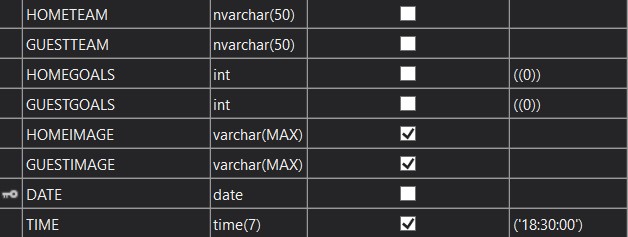


Рисунок 3.2 – Структура таблицы «Playes»

На рисунке 3.3 проиллюстрирована структура таблицы «Teams», которая содержит информацию о командах. В данной таблице поле Play\_ID, через identity(1,1), является первичным ключом. Поле TEAMS\_NAME хранит имя команды, POINTS – очки, PLACE – место в таблице, GROUPAB — содержит информацию о группе, в которой команда, GAMES – количество игр.

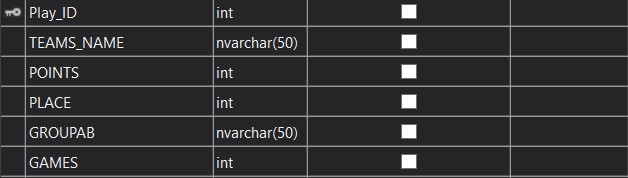


Рисунок 3.3 – Структура таблицы «Teams»

На рисунке 3.4 проиллюстрирована структура таблицы «Users», которая содержит информацию о пользователях. В данной таблице поле ID, через identity(1,1), является первичным ключом. Поле NAME хранит имя пользователя, PASSWORD – пароль, EMAIL – адрес электронной почты, CARD — номер платежной карты, ADMIN – соответственно роль (Yes для администратора, NULL для пользователя).

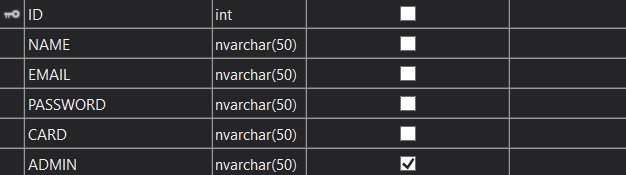


Рисунок 3.4 – Структура таблицы «Users»

На рисунке 3.5 изображена структура таблицы «Tickets», содержащая информацию о билетах. Первичный ключ ID\_PLAY хранит уникальный идентификатор игры, SECTOR – Сектор, ROW –ряд, PLACE –место, DATE – дата матча, USER\_ID – id пользователя, ID\_TICKET – ID билета, первичный ключ.

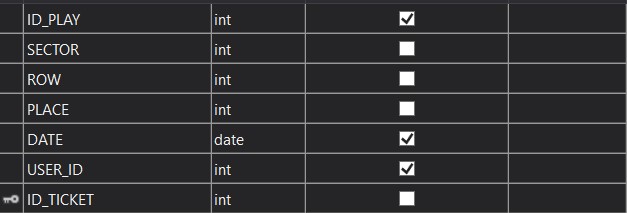


Рисунок 3.5 – Структура таблицы «Tickets»

**3.5 Проектирование логики сценариев использования**

При проектировании приложения были рассмотрены различные сценарии использования. В графической части представлена диаграмма последовательности бронирования билета и изменение информации.

Из диаграммы видно, что пользователь выбирает бронь, приложение делает запрос в базу данных, получая все существующие брони. И он будет уведомлен, свободно ли место.

4. Создание (реализация) программного средства

Следующим этапом разработки приложения является непосредственная реализация программного решения в соответствии с уже сформированными требованиями и шаблонами.

# **Основные классы программного средства**

Для выполнения технических задач программного средства «Хоккейная лига» должны быть реализованы следующие функции и соответствующие им классы и методы:

* Выполнение входа;
* Регистрация;
* Просмотр результатов;
* Просмотр будущих матчей;
* Заполнение формы бронирования билета;
* Просмотр таблицы с командами;
* Функционирование панели администратора.

Далее подробно рассмотрены каждые из необходимых для выполнения технических задач функции, а также созданные для их выполнения классы и методы и их функционал и реализация.

# **Описание классов и методов программного средства**

## **Выполнение входа и регистрации**

Необходимый функционал для возможности входа и регистрации пользователя в аккаунт, а также для проверки корректности данных.

Метод выполняется при нажатии пользователя на кнопку «Войти» или регистрация при заполнении всех полей корректно. При регистрации реализовано хэширование пароля и при входе пароль хэширунтся и сравнивается.

При успешной валидации введённых пользователем входных данных выполняется проверка на существование введённого логина в базе данных, и при нахождении значения выполняется проверка на соответствие введённого пароля и хранящегося в базе данных пароля. При успешном прохождении всех проверок пользователь перенаправляется на главную страницу, а противном случае – получает сообщение о соответствующей ошибке.

В окне регистрации есть переход на авторизацию как и в авторизации на регистрацию.

## **Просмотр Расписания**

После успешной авторизации пользователь перенаправляется на основную страницу программного средства, где есть список будущих матчей.

Пользователь может пролистать список и в любой строке выбрать заказ билета, где он может если что поменять дату и выбрать место для бронирования на матч.

Администратор может менять время игры.

## **Заполнение формы бронирования билета**

После вышеописанного выбора необходимого расписания пользователь перенаправляется на форму бронирования билета после нахатия на кнопку «Купить билет».

После нажатия пользователем на кнопку «Забронировать» выполняется проверка на корректность введённых в форму данных, показывая пользователю при совпадении места для бронирования мест. После успешной валидации метод вносит изменения в таблицы базы данных: добавляет в таблицу с заказами новый заказ, и изменяет количество доступных мест в таблице с расписанием.

## **Просмотр Таблицы**

Пользователь может только смотреть результаты в таблице, и знать положение команды на данный момент.

Администратор может менять место, игры и очки команд. После правильного ввода, изменения сохраняются и после выхода сортируются по месту.

## **Просмотр Результаты**

Пользователь может только смотреть результаты матчей и пролистывать их и знать итоги игры.

Администратор может записывать голы. После правильного ввода, изменения сохраняются и разделение смотрит на текущую дату, если позже, то в результатах, если день еще не наступил, то в расписании.

5. Тестирование, проверка работоспособности и анализ полученных результатов

При тестировании данного приложения, были применены сценарии, которые могли бы привести к ошибке. В этой главе мы рассмотрим некоторые такие сценарии и посмотрим на их обработку.

5.1 Мануальное тестирование

В момент регистрации, возможна такая ситуация, в которой пользователь ничего не ввёл или ввёл неверные данные. Обработка данного сценария приведена на рисунке 5.1

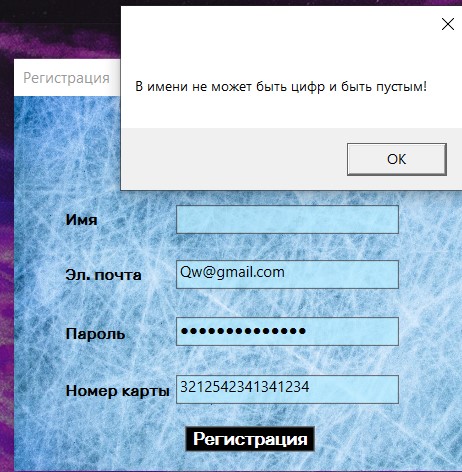


Рисунок 5.1 – Любое поле оставлено пустым

Сценарий, при котором введена почта, но неверно, рис. 5.2.

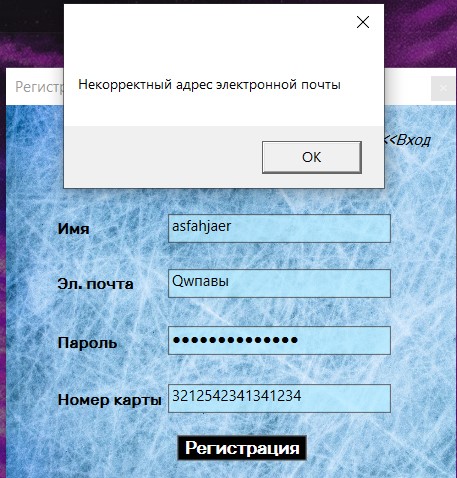


Рисунок 5.2 – неверно введенная почта

Сценарий, когда пользователь пытается войти под несуществующим логином рис. 5.3.

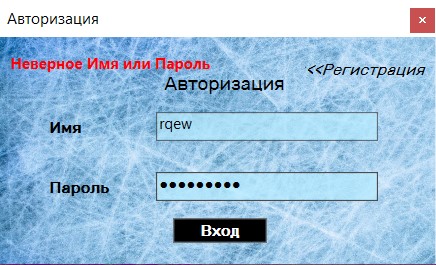


Рисунок 5.3 – Вход под несуществующим именем или неверно введенном пароле

6. Руководство по использованию

При запуске приложения у вас появится окно регистрации, где вы можете перейти на авторизацию или зарегистрировать нового пользователя. Для смены регистрации и входа существует отдельная кнопка.

Окно регистрации и авторизации представлены на рисунках 6.1 и 6.2.

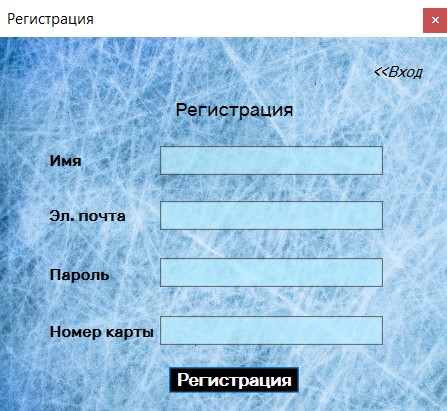


Рисунок 6.1 – Окно регистрации

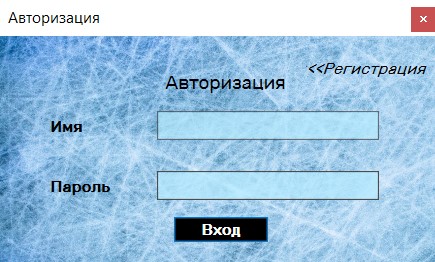


Рисунок 6.2 – Окно авторизации

После успешного входа под вашей учётной записью у вас появится главное меню, которое изображено на рисунке 6.2.

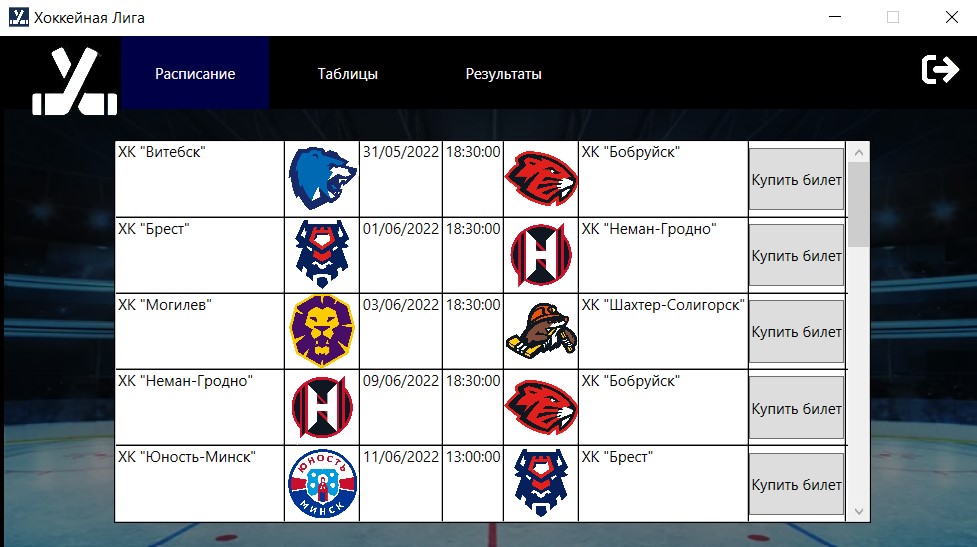


Рисунок 6.3 – Главное меню приложения с расписанием матчей

В нем изображены вкладки, ведущие на все страницы приложения.

Нажав на кнопку “Купить билет” мы перейдём в окно бронирования билета, которое изображено на рисунке 6.4.

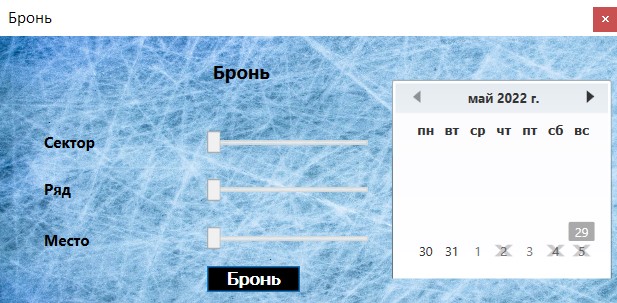


Рисунок 6.4 – Окно бронирования билетов

В этом окне мы можем выбрать день, сектор, ряд и место.

При нажатии на кнопку “Бронь” окно закроется в случае успешного бронирования.

Дальше мы посмотрим вкладки, такие как Таблицы и Результаты рис. 6.5 и 6.6

Там мы можем просмотреть информацию о командах, их успехах и итогов.

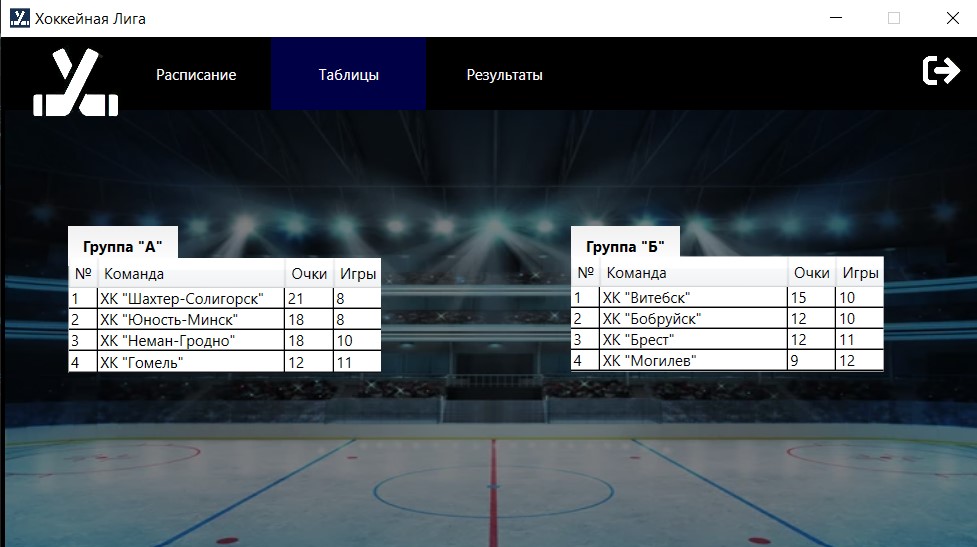


Рисунок 6.5 –Таблицы с командами

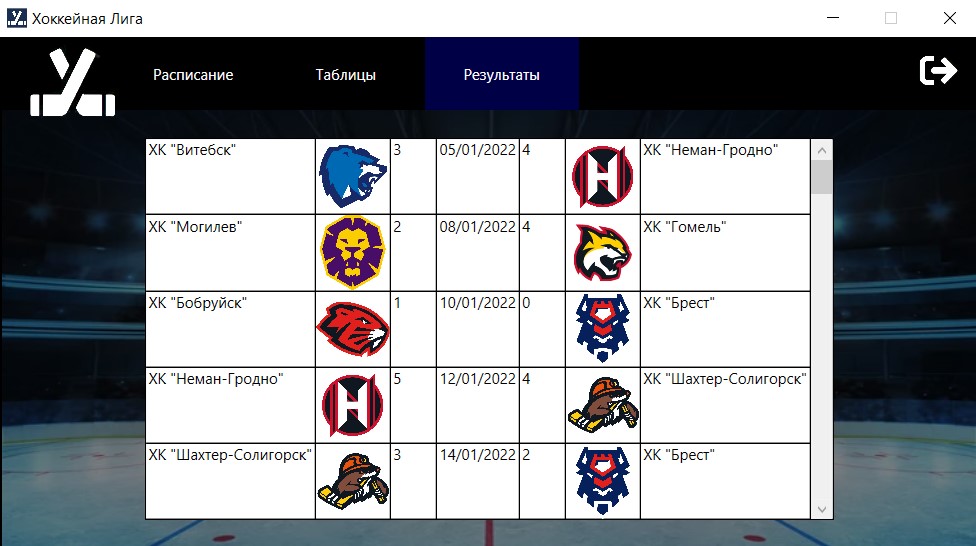


Рисунок 6.6 –Результаты матчей

ЗАКЛЮЧЕНИЕ

В данном курсовом проекте было разработано приложение «Хоккейная лига» при помощи языка программирования C#, API-интерфейса Windows Presentation Foundation, технологии Entity Framework и базы данных Microsoft SQL Server. При выполнении курсового проекта использовались принципы и приёмы объектно-ориентированного программирования.

Разработанное программное средство предоставляет пользователю следующие функциональные возможности:

- регистрация пользователей;

- авторизация пользователей;

- просмотр расписания и результатов матчей;

- просмотр статистики команд в группах

- бронь места на матче.

Администратор имеет следующие возможности:

- изменение данных об матчах и сохранение их в БД;

- добавление результатов матча дописывая и сохраняя их в БД.

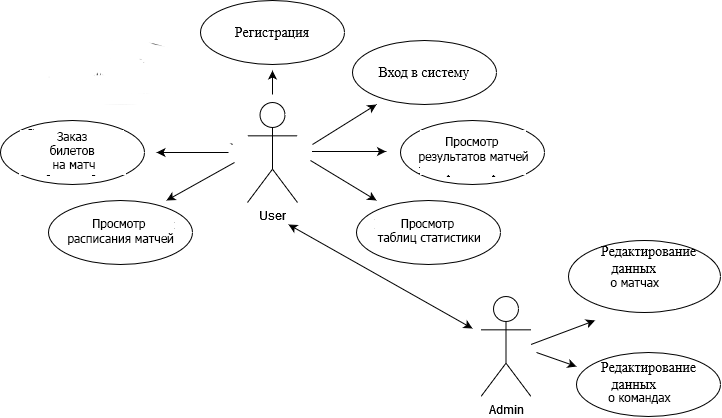
Приложение корректно работает с базой данных. В ходе тестирования не было выявлено нарушений в работе с базой данных.

Приложение было успешно протестировано, что означает пригодность этого приложения для своих целей.

# **Список литературы**

1. Microsoft Visual Studio [Электронный ресурс] – https://ru.wikipedia.org/wiki/Microsoft\_Visual\_Studio – Дата доступа 23.02.2022
2. Полное руководство по языку программирования C# 7.0 и платформе .NET 4.7. Режим доступа: https://metanit.com/sharp/tutorial/ – Дата доступа: 14.03.2022
3. Пацей, Н.В. Курс лекций по языку программирования C# / Н. В. Пацей. – Минск: БГТУ, 2018. – 175 с.
4. Руководство по WPF // [Электронный ресурс]. – Режим доступа: https://metanit.com/sharp/wpf/ – Дата доступа: 28.03.2022
5. Работа с Entity Framework Core [Электронный ресурс] – https://professorweb.ru/my/entity-framework/6/level1/ – Дата доступа 03.05.2022
6. Блинова, Е.А. Курс лекций по Базам данным / Е.А. Блинова. – Минск: БГТУ, 2019. – 175 с.

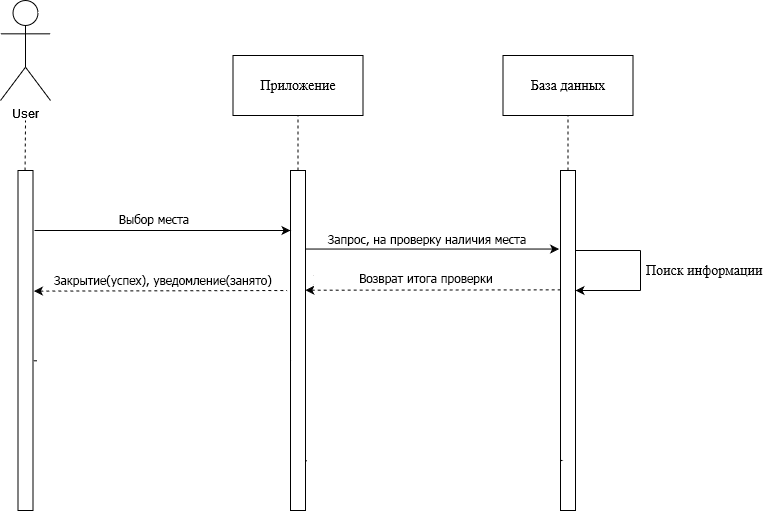
**Приложение А**



**Приложение Б Логическая схема базы данных**



**Приложение В**



**Приложение Г**

