

МИНОБРАЗОВАНИЯ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук
Кафедра программирования и информационных технологий

Отчет по курсу «Технологии программирования»

Мобильное приложение «Botegy»

Воронеж 2022

Содержание

| | | |
|------|--|----|
| 1 | Анализ предметной области..... | 4 |
| 2 | Постановка задачи | 5 |
| 3 | Функциональные требования | 5 |
| 4 | Технические требования..... | 6 |
| 5 | Требования к интерфейсу | 6 |
| 6 | Макеты интерфейса | 6 |
| 7 | Анализ аналогов..... | 20 |
| 8 | Входные-выходные данные (IDEF0) | 21 |
| 9 | Диаграмма вариантов использования | 22 |
| 9.1 | Диаграмма Игрока | 22 |
| 9.2 | Диаграмма Модератора | 23 |
| 9.3 | Диаграмма Наблюдателя | 24 |
| 10 | Диаграмма состояний | 25 |
| 11 | Структурная схема приложения | 26 |
| 12 | Схема базы данных | 27 |
| 13 | Диаграмма потоков данных..... | 28 |
| 14 | Диаграмма классов | 30 |
| 14.1 | Диаграмма классов сущностей | 30 |
| 14.2 | Диаграмма классов репозиториев..... | 31 |
| 14.3 | Диаграмма классов сервисов..... | 31 |
| 14.4 | Диаграмма классов контроллеров..... | 31 |
| 14.5 | Диаграмма служебных классов..... | 31 |
| 14.6 | Диаграмма классов пакета ButtonScript..... | 32 |
| 14.7 | Диаграмма классов пакета Dialog | 32 |

| | | |
|-------|---|----|
| 14.8 | Диаграмма классов пакета Entity | 33 |
| 14.9 | Диаграмма классов пакета Model | 34 |
| 14.10 | Диаграмма классов пакета SceneManager | 35 |
| 15 | Список планируемых работ..... | 36 |
| 16 | Используемая платформа | 36 |
| 17 | Сценарии воронок..... | 37 |
| 18 | Календарный план | 39 |

1 Анализ предметной области

Издавна человечество старалось упростить свою жизнь. Большая часть сил была направлена на облегчение задач в различных сферах жизни. Огромным шагом в этом направлении является развитие и удешевление ЭВМ, появление и постоянный прогресс смартфонов. Практически у каждого современного человека есть мобильный телефон, у некоторых даже больше одного. Мобильные телефоны нынешнего поколения перестали быть просто средством общения и переросли в нечто большее. Они стали переносимыми мобильными компьютерами, открывающими своим владельцам очень много дополнительных возможностей – доступ к большому каталогу информации, прослушивание музыки, предоставляет разные формы общения, построение оптимального пути, проведение свободного времени в развлечениях и многое другое.

Постоянный прогресс высоких технологий и увеличение их возможностей привели к росту спроса на программы и приложения для этих самых технологий. Среди прочих направлений в этой области отдельное, особое место занимают программы на мобильные устройства. Среди них есть как необходимые для повседневной жизни, упрощающие быт и разные сферы жизнедеятельности, так и приложения – игры (играть/өгрэт/өйрэт – с татарского «обучать». Это форма обучения), без которых процесс обучения был бы сложнее. Чем объясняется популярность игр. Это обусловлено тем, что кому-то хочется скоротать время, кто-то проникается азартом достижения цели и преодоления препятствий, кому-то приятно иметь возможность заниматься тем, что интересно для него в жизни, где и когда угодно (симулятор футбола), кто-то хочет развивать логику или реакцию и так далее. Все эти причины способствуют популярности мобильных игр и стимулированию их к разработке. Наличие бесплатных платформ для создания игр, таких как Unity позволяют разработчикам воплощать в жизнь самые разнообразные свои идеи.

Целью данной работы является разработка мобильной игры под Android на платформе Unity. Особенности игры будут нетипичное косвенное участие пользователя в игровом процессе, а именно написание стратегии, которая будет «играть» за него, концепция PvP, то есть возможность поиграть против стратегий других людей, возможность для анализа эффективности стратегии посредством просмотра визуализации матча.

2 Постановка задачи

Разработать мобильную игру, которая будет решать следующие задачи:

- Обеспечение возможности построения стратегии поведения для юнитов на поле боя при помощи интегрированного редактора;
- Обеспечение возможности пользователя посмотреть матч, чтобы увидеть свою стратегию в бою;
- Обеспечение возможности не смотреть матч, с целью экономии времени, а просто увидеть результат;
- Обеспечение возможности выбора стратегии, против которой игроку хотелось бы попробовать свои силы;
- Обеспечение возможности создания нескольких стратегий;
- Обеспечение возможности редактирования уже существующих стратегий игрока.

3 Функциональные требования

- Производить регистрацию новых учетных записей.
- Производить авторизацию зарегистрированных пользователей.
- Осуществлять поиск ботов по названию.
- Осуществлять создание новых ботов зарегистрированных пользователей, редактирование и удаление уже созданных.
- Осуществлять вывод информации об истории матчей ботов пользователей.
- Осуществлять вывод информации о созданных ботах пользователей.

- Просчитывать результат матча между двумя ботами.
- Выводить ход матча между ботами.
- Осуществлять редактирование данных учетной записи пользователя.
- Осуществлять блокировку учетных записей и назначение пользователей модераторами.

4 Технические требования

Приложение должно обеспечивать:

- Авторизацию его пользователей посредством электронной почты и пароля, которые хранятся и идентифицируются внешней (по отношению к приложению) системой авторизации.
- Удаление ботов пользователя при удалении его учетной записи.
- Вычисление результатов и хода всех матчей на сервере для уменьшения нагрузки на клиентов.

5 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 корректно отображаться при изменении размера экрана. Интерфейс должен поддерживать портретную ориентацию экрана.

Интерфейс должен содержать необходимую для пользователя информацию: информацию о самом пользователе, информацию о пользовательских ботах, информацию о других пользователях, их ботах, матчах. Информация должна находиться в тех местах приложения, где она будет актуальна, то есть, например, при просмотре результатов матча не нужно отображать список ботов текущего пользователя.

6 Макеты интерфейса

Сразу после запуска приложения пользователю будет показан экран, на котором ему будет предложено войти в свою учетную запись, зарегистрировать новую учетную запись или продолжить работу с приложением без входа в учетную запись.



Рисунок 1 - Начальный экран приложения

Рассмотрим случай, когда пользователь в первый раз зашел в приложение и еще не зарегистрировал учетную запись.

В этом случае при нажатии на кнопку «Регистрация» пользователь увидит экран регистрации, где ему будет предложено ввести электронную почту, никнейм для новой учетной записи и пароль. Если будут введены корректные данные и на введенную почту еще не была зарегистрирована учетная запись, то будет создана новая учетная запись, в которую сразу войдет пользователь, и он будет направлен на главный экран приложения.

Кроме введения данных пользователю предлагается создать новую учетную запись с помощью Google.

При нажатии на кнопку «Назад» пользователь вернется на начальный экран приложения.



Рисунок 2 - Экран регистрации

Теперь рассмотрим случай, когда пользователь уже зарегистрирован и хочет войти в свою учетную запись.

После нажатия кнопки «Вход» пользователь увидит экран, на котором ему будет предложено ввести свои учетные данные или войти в учетную запись с помощью Google.

После нажатия кнопки «Войти», если пользователь ввел нужные данные, он увидит главный экран приложения для пользователя, вошедшего в учетную запись.

При нажатии на кнопку «Назад» пользователь вернется на начальный экран приложения.

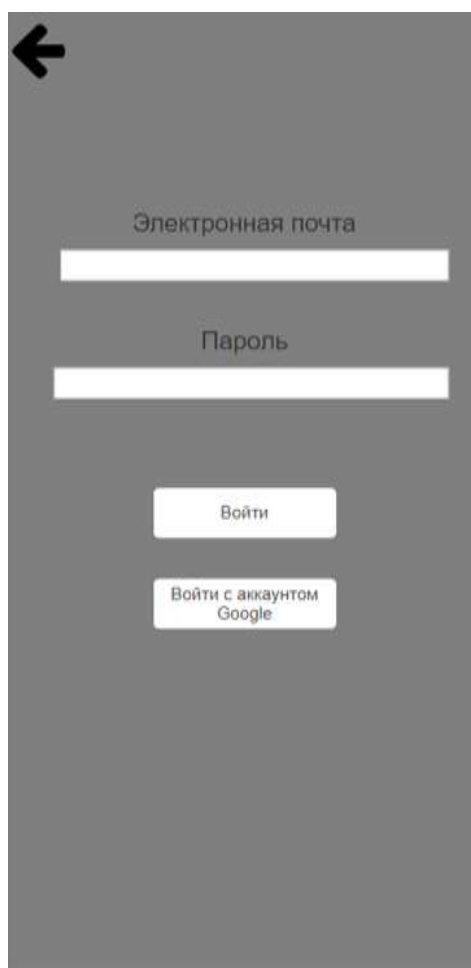


Рисунок 3 - Экран входа в учетную запись

На главном экране находятся кнопки для перехода к экрану с созданными им ботами, экрану поиска ботов других пользователей, экрану просмотра и редактирования информации учетной записи.



Рисунок 4 - Главный экран приложения

Если пользователь нажмет на кнопку с его никнеймом, то он увидит экран, на котором отобразятся никнейм пользователя, электронная почта, используемая для входа в учетную запись. Рядом с никнеймом находится кнопка для редактирования имени пользователя. Во всплывающем окне будет предложено ввести новое имя пользователя.

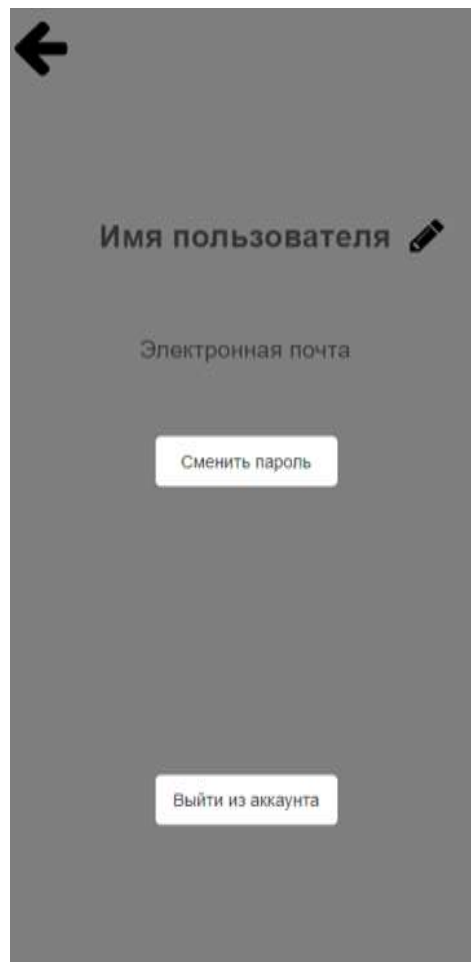


Рисунок 5 - Экран профиля пользователя

Кнопка «сменить пароль» ведет на экран, позволяющий изменить пароль учетной записи. На нем предлагается ввести старый и новый пароли. Если старый пароль введен верно, то пользователь вернется на экран с информацией о его учетной записи.

При нажатии на кнопку «Назад» пользователь вернется на главный экран приложения.

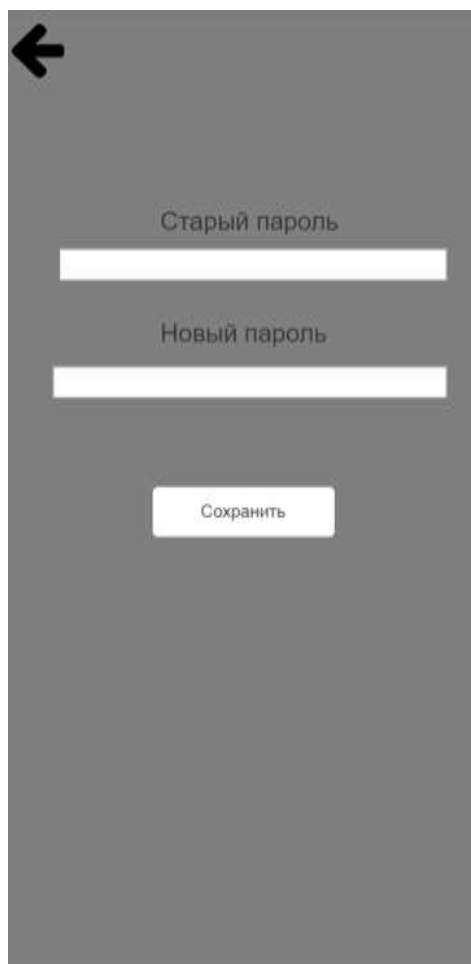


Рисунок 6 - Экран изменения пароля пользователя

Кнопка «Выйти из аккаунта» позволяет выйти из учетной записи и вернуться на начальный экран приложения.

При нажатии на кнопку «Назад» пользователь вернется на экран с информацией о текущем пользователе.

Вернемся к главному экрану приложения для пользователя, вошедшего в учетную запись. Нажатие на кнопку «Мои боты» позволит пользователю перейти к списку ботов, которых он создал.

Кнопки, расположенные слева, позволяют переключаться между ботами для просмотра их названия, истории матчей и кода, написанного пользователем. Кнопки внизу экрана позволяют изменять ботов.

При нажатии на кнопку «Назад» пользователь вернется на главный экран приложения.

При нажатии на кнопку удаления бота пользователь увидит всплывающее окно, на котором пользователю необходимо будет подтвердить, что он действительно хочет удалить бота.

При нажатии на кнопку редактирования бота пользователю будет предложено выбрать, что сделать с ботом: заменить выбранного или создать его копию, чтобы редактировать ее.

При нажатии на кнопку добавления бота будет создан новый бот.

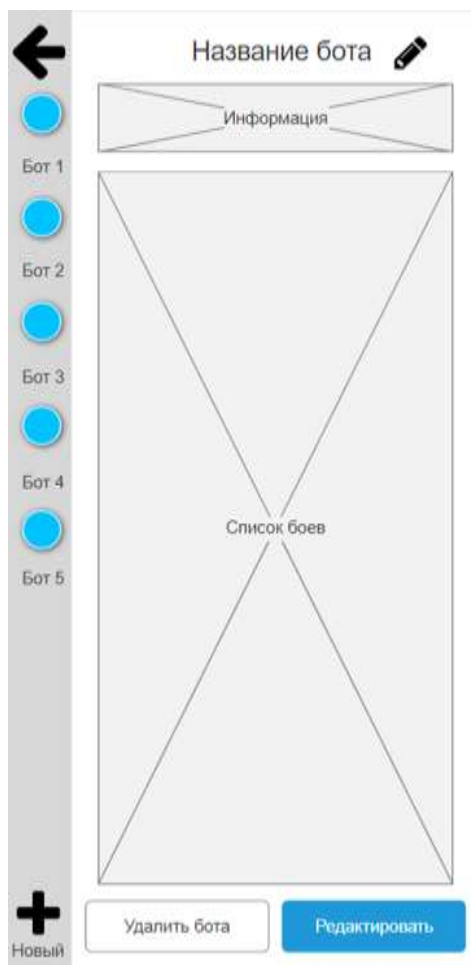


Рисунок 7 - Экран списка ботов пользователя

При редактировании бота и добавлении нового пользователь увидит экран редактора кода бота.

Кнопки с левой стороны экрана вызывают появление меню с теми блоками кода, которые пользователь может добавить в код своего бота, в зависимости от выбранной категории. При выборе блока кода из меню он

появится на экране и его можно будет редактировать: добавлять вложенные блоки, изменять значения, удалять.

При нажатии кнопки сохранения изменения будут сохранены для редактируемого бота, а пользователь вернется на экран со списком ботов.

Нажатие на кнопку «Назад» приведет к тому, что изменения не будут сохранены.

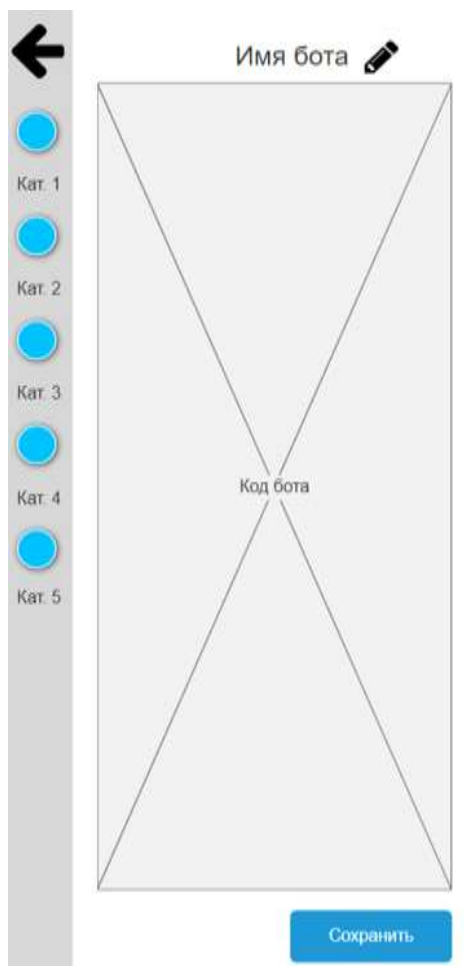


Рисунок 8 - Экран редактирования кода бота

Рассмотрим также случай, когда пользователь решил не входить в учетную запись. В этом случае он будет направлен на экран поиска ботов. На этот же экран направляются зарегистрированные пользователи при нажатии кнопки «Поиск сражения» на главном экране.

Пользователю будет предложено ввести поисковой запрос – название интересующего бота. При нажатии кнопки поиска пользователь увидит список

кнопок с названиями найденных ботов. При нажатии на кнопки пользователь сможет перейти на экраны с информацией о соответствующих ботах.

При нажатии на кнопку «Назад» зарегистрированный пользователь вернется на главный экран приложения, а незарегистрированный – на начальный.



Рисунок 9 - Экран поиска ботов

Экран с информацией о боте содержит название бота, кнопку с никнеймом пользователя, который его создал и кнопки с матчами, в которых бот участвовал.

При нажатии на кнопку с никнеймом создателя пользователь сможет перейти на экран с информацией о другом пользователе.

При нажатии на кнопку матча пользователь сможет посмотреть ход выбранного матча.

При нажатии на кнопку «Назад» пользователь вернется на экран поиска.

Для зарегистрированного пользователя экран бота содержит также кнопку «Сразиться», что позволит одному из ботов пользователя сыграть матч с просматриваемым ботом.

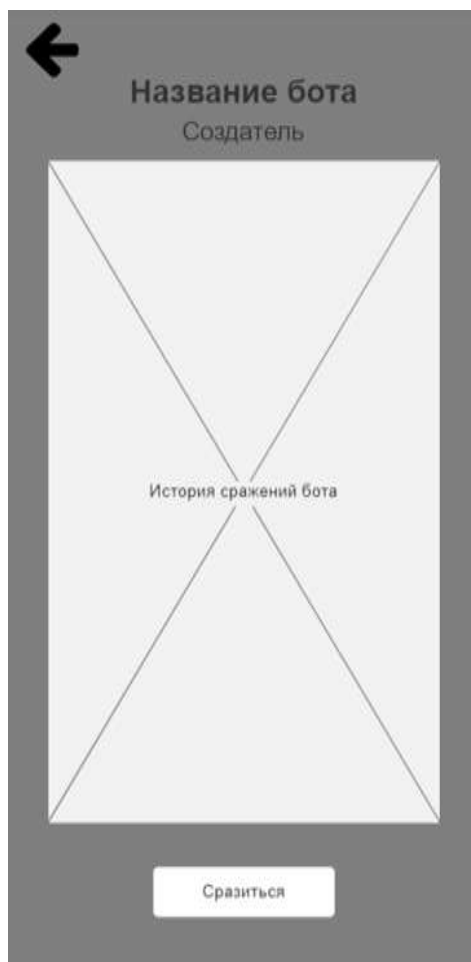


Рисунок 10 - Экран информации о боте другого пользователя

Экран с информацией о другом пользователе содержит никнейм этого пользователя и кнопки с названиями созданных им ботов. Нажатие на эти кнопки ведет на экраны с информацией о соответствующих ботах. Для зарегистрированного пользователя с правами модератора экран содержит кнопки для блокировки учетной записи пользователя и назначения его модератором.

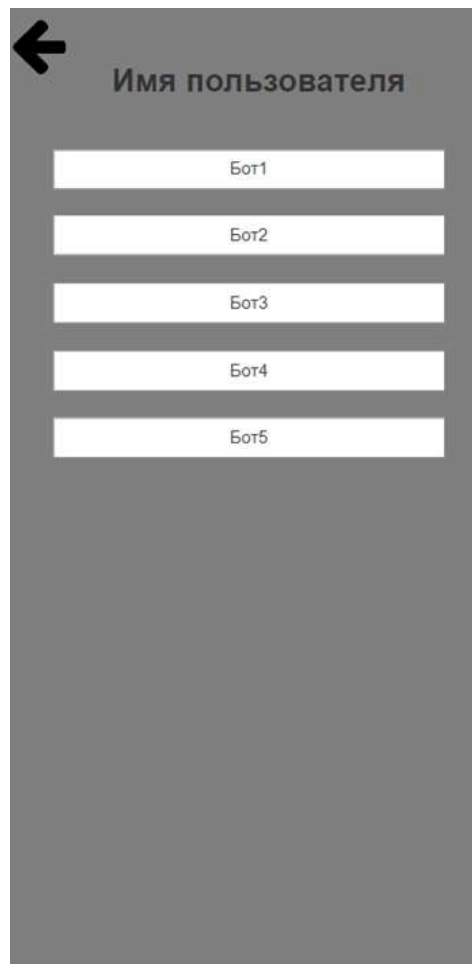


Рисунок 11 - Экран информации о пользователе

Рассмотрим случай, когда зарегистрированный пользователь нажал «Сразиться» на экране бота. Тогда пользователь увидит экран выбора одного из своих ботов для матча.

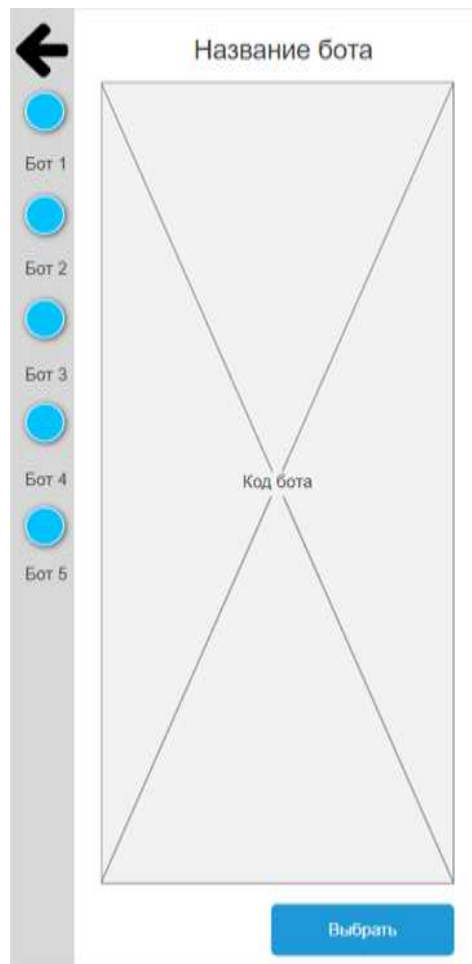


Рисунок 12 - Экран выбора бота для матча

Кнопки, расположенные слева, позволяют переключаться между ботами для просмотра их названия и кода, написанного пользователем.

При нажатии на кнопку «Назад» пользователь вернется на экран просмотра информации о боте другого пользователя.

При нажатии на кнопку выбора бота будет проигран матч между ботом пользователя и выбранным им ботом другого пользователя.

Результат матча будет выведен во всплывающем окне. В окне также расположены кнопки, позволяющие закрыть окно и просмотреть ход матча.



Рисунок 13 - Отображение результатов матча

При нажатии на кнопку «Просмотр сражения» пользователь будет направлен на экран просмотра матча.

Экран просмотра матча содержит названия ботов, участвовавших в матче, и поле, отображающее действия, предпринимаемые ботами на каждом шагу.

Нажатие на кнопки матчей на экранах информации о ботах ведут также на экран просмотра матча.

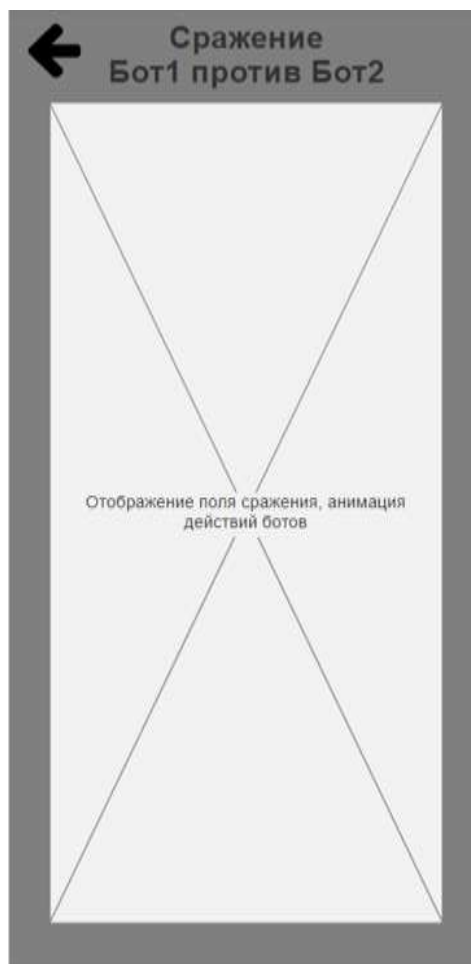


Рисунок 14 - Экран просмотра матча

7 Анализ аналогов

Приступая к разработке новой игры в определенном жанре, необходимо проанализировать проекты, уже существующие внутри него. Рассмотрим их достоинства, недостатки. Насколько они удобны в использовании, содержат ли необходимый функционал. И на основе этого анализа сделаем выводы какие моменты нужно учесть при разработке своего приложения, а именно каким образом будет построен удобно для пользователя интерфейс, и какой функционал будет иметь игра.

GLADIABOTS - AI Combat Arena

Gladiabots – это стратегия о боевых ботах, где игрок создает ИИ для своего отряда и отправляете его в бой.

Основной функционал (Игрок может):

- Создать свой ИИ и наблюдать, как боты выполняют его на арене
- Создать и настроить свой отряд ботов
- Играть в режим одиночной кампании
- Играть в режим многопользовательской онлайн игры, в том числе: рейтинговый, обычный и частные матч
- Попробовать 3 совершенно разных режима игры: устранение, доминирование и сбор
- Участвовать в турнирах или создавать их
- Сражаться с другими игроками даже если они не онлайн, так как присутствует асинхронный мультиплеер
- Использовать Режим песочницы для проверки своих тактик
- Сражаться с равными по силам игроками благодаря системе рангов

Преимущества:

- Простая, но эффективная система программирования ИИ
- Различные режимы игры
- Возможность проведения турниров

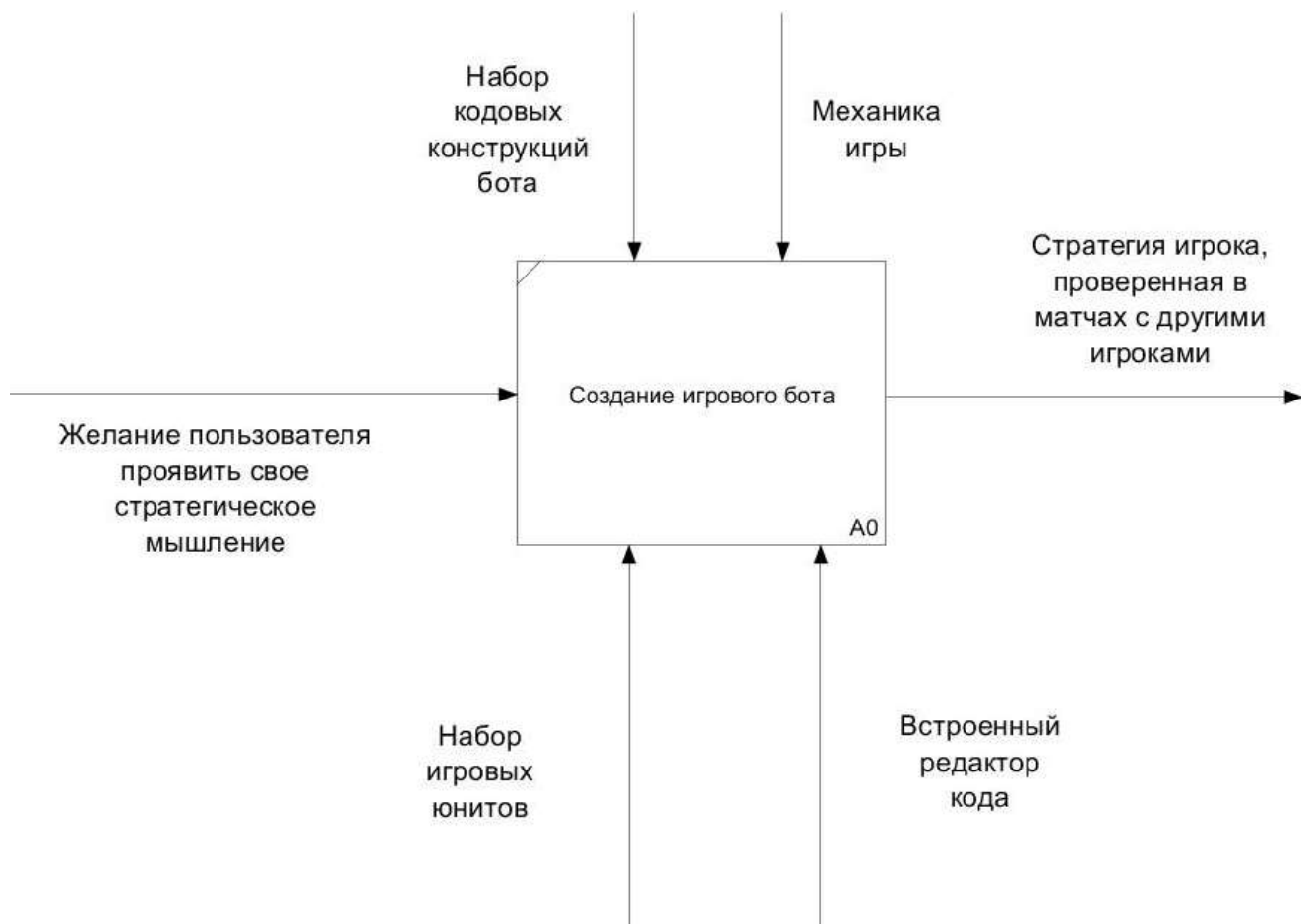
Недостатки:

- Достаточно высокий порог вхождения
- Не очень дружелюбный к пользователю интерфейс
- Чтобы хорошо играть, нужно потратить очень много времени

Вывод:

Игра имеет отличную концепцию, объемный функционал и большое количество активностей внутри нее, но также имеются существенные недостатки, которые перечислены выше.

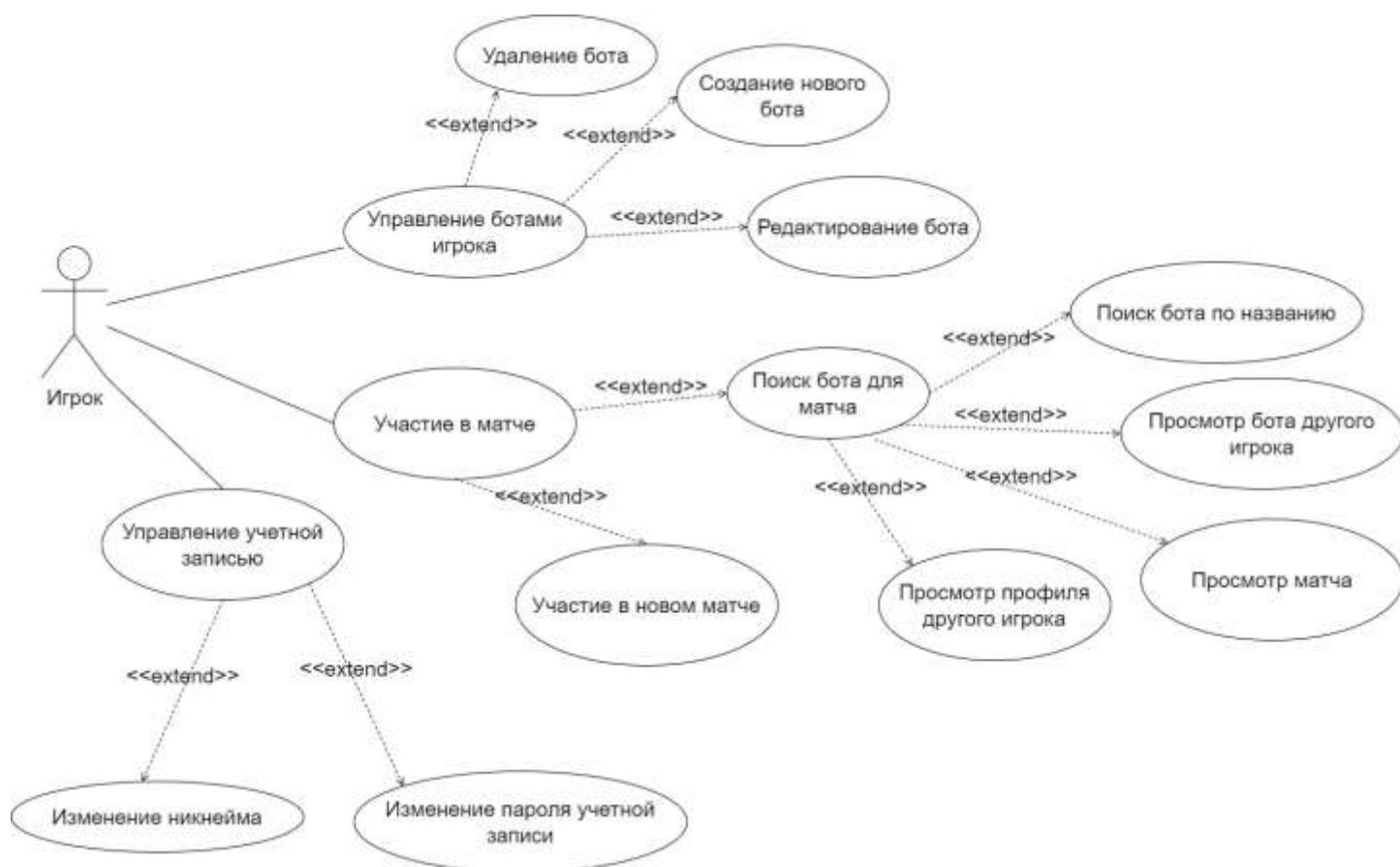
8 Входные-выходные данные (IDEF0)



9 Диаграмма вариантов использования

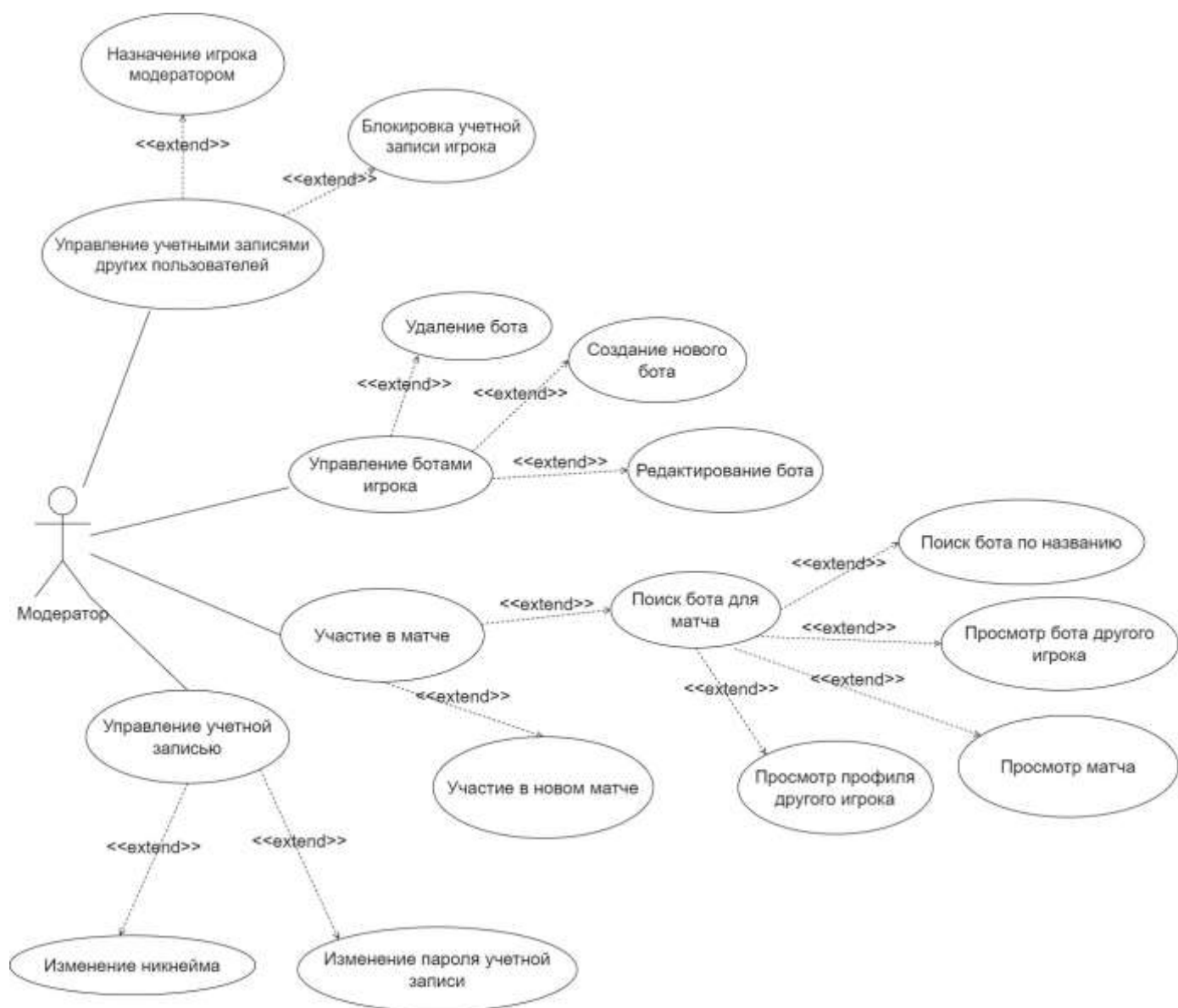
9.1 Диаграмма Игрока

Игрок – авторизованный пользователь, не обладающий правами Модератора.



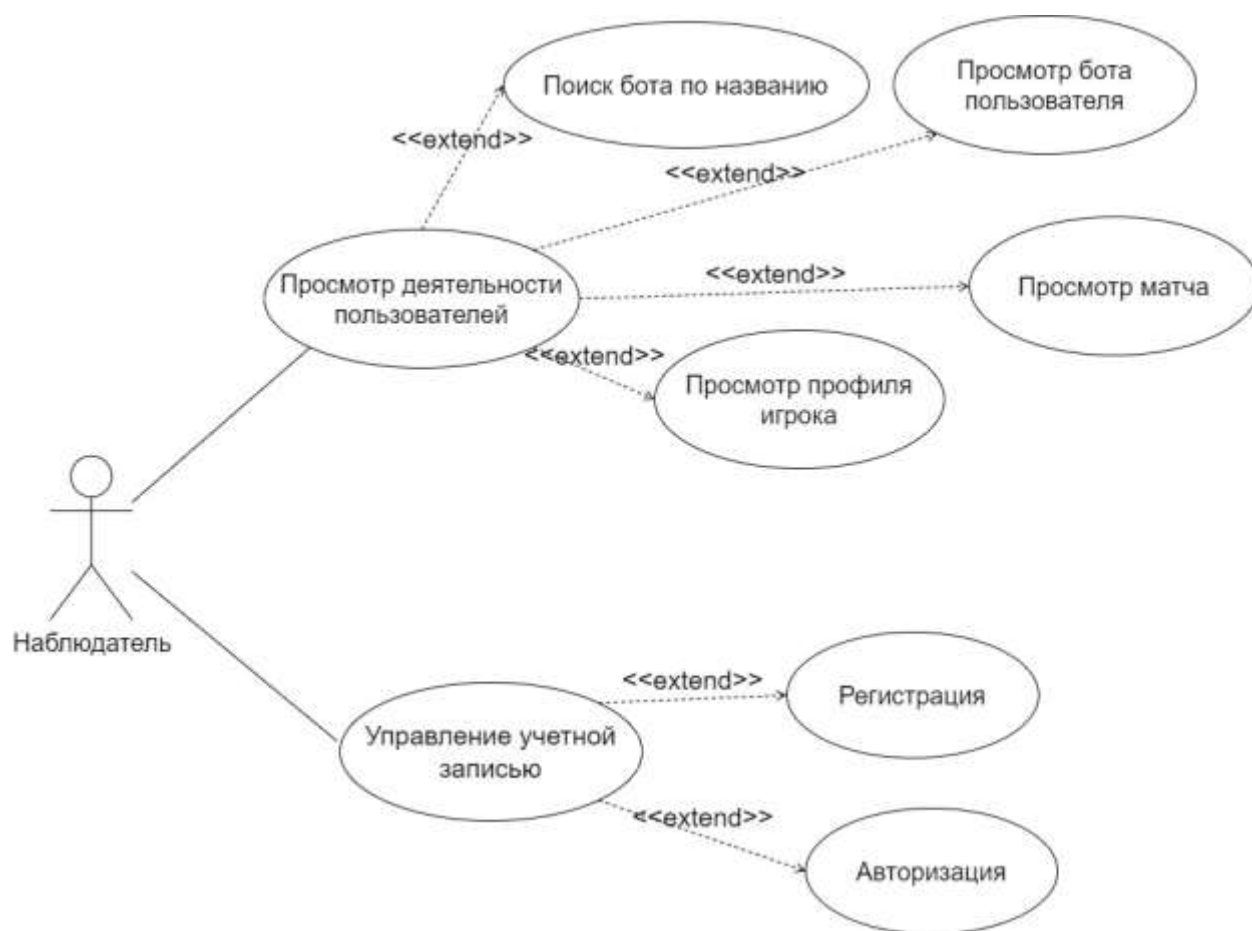
9.2 Диаграмма Модератора

Модератор – авторизованный пользователь, обладающий правами Модератора.



9.3 Диаграмма Наблюдателя

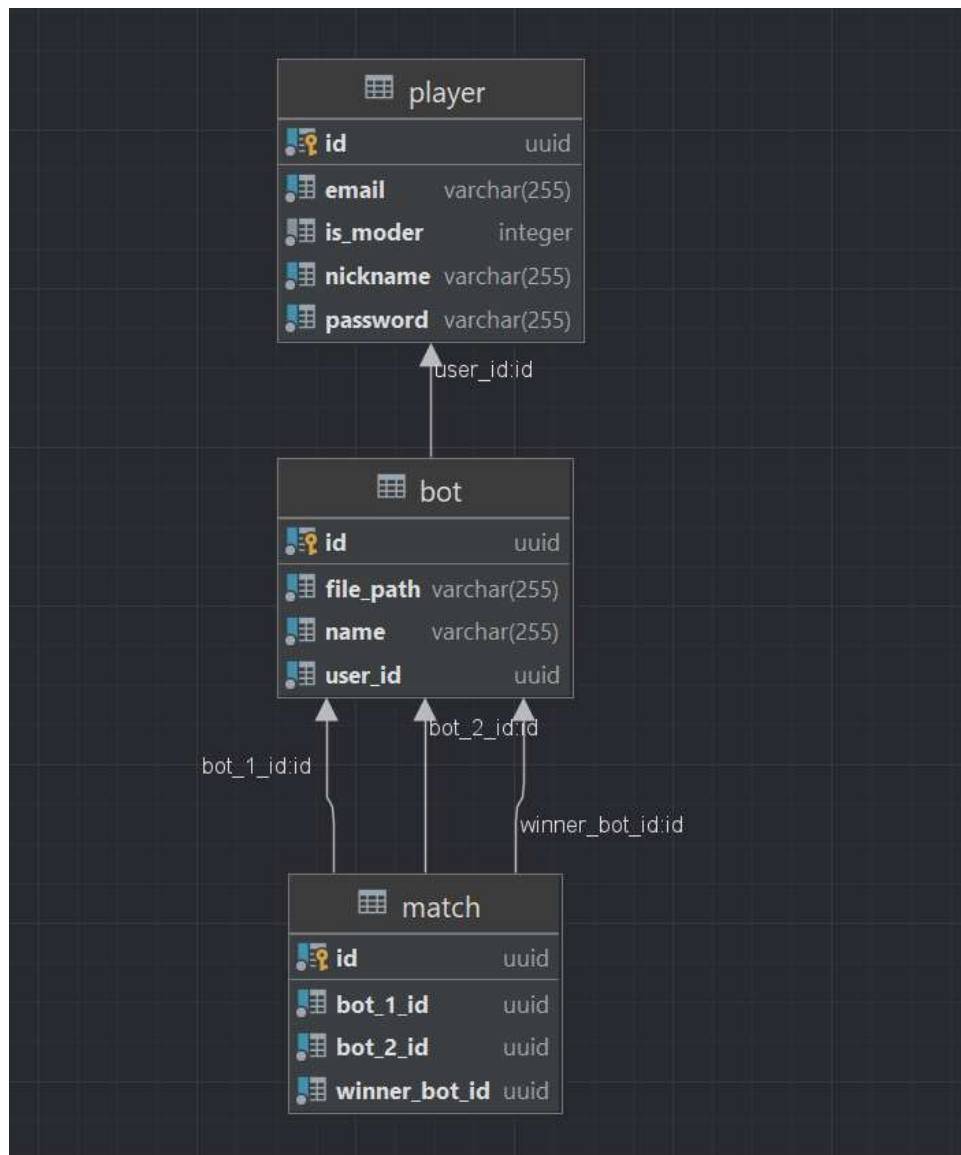
Наблюдателем считается пользователь, решивший пользоваться приложением без входа в учетную запись.



10 Диаграмма состояний



12 Схема базы данных

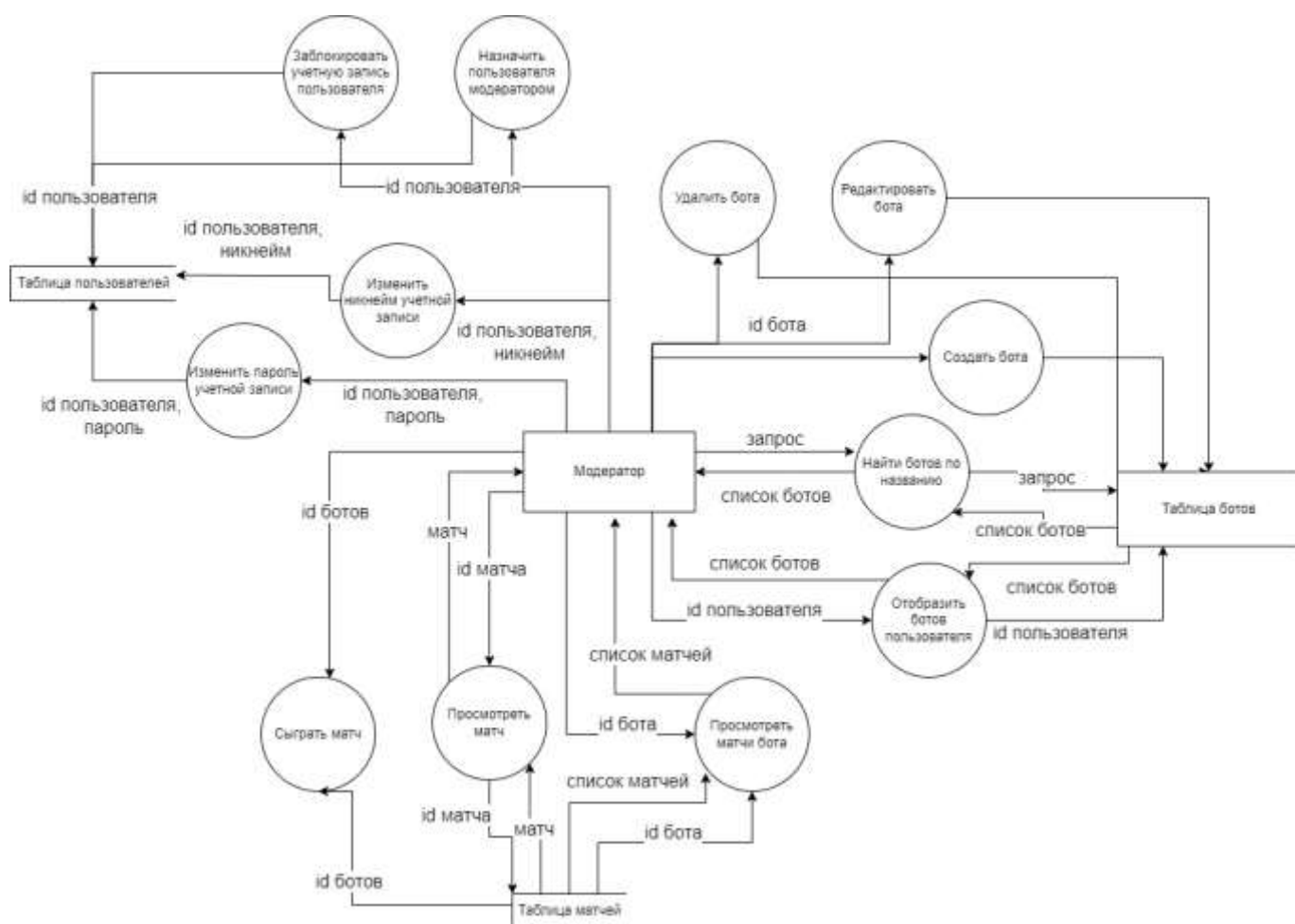
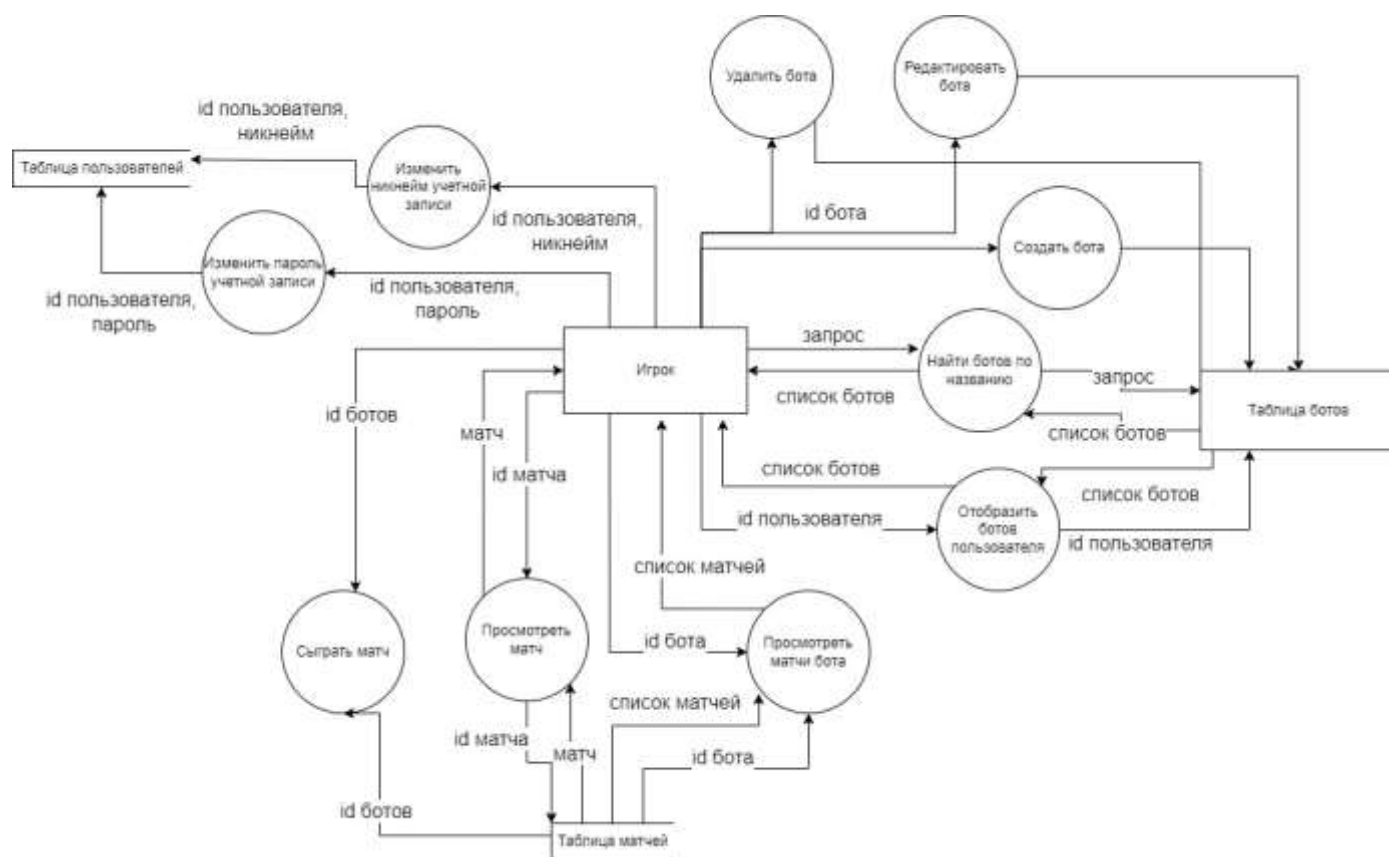


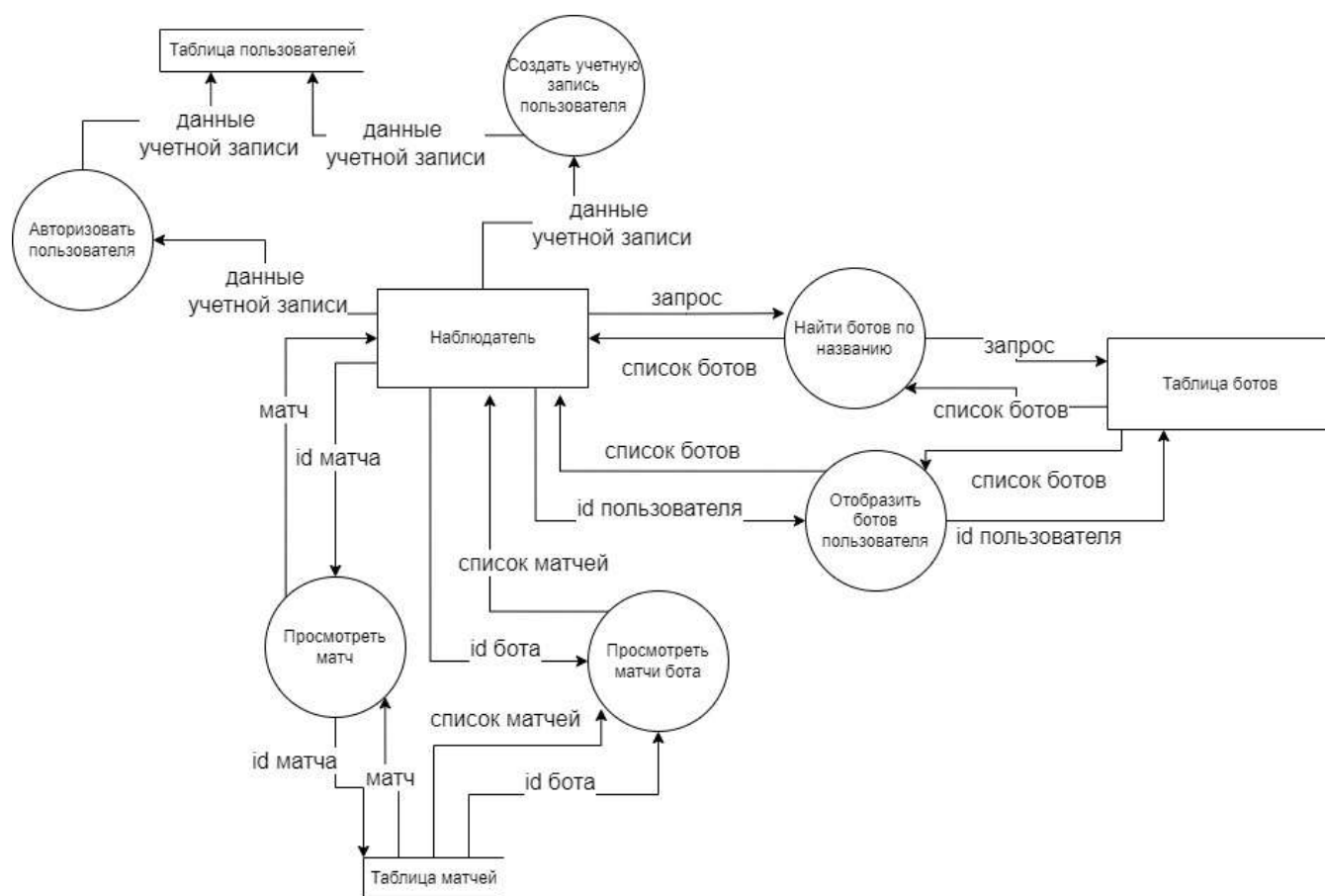
player - таблица, содержащая информацию о зарегистрированных пользователях приложения

bot - таблица, содержащая информацию о стратегиях, которые создали зарегистрированные пользователи.

match - таблица, содержащая информацию о проведённых матчах между ботами.

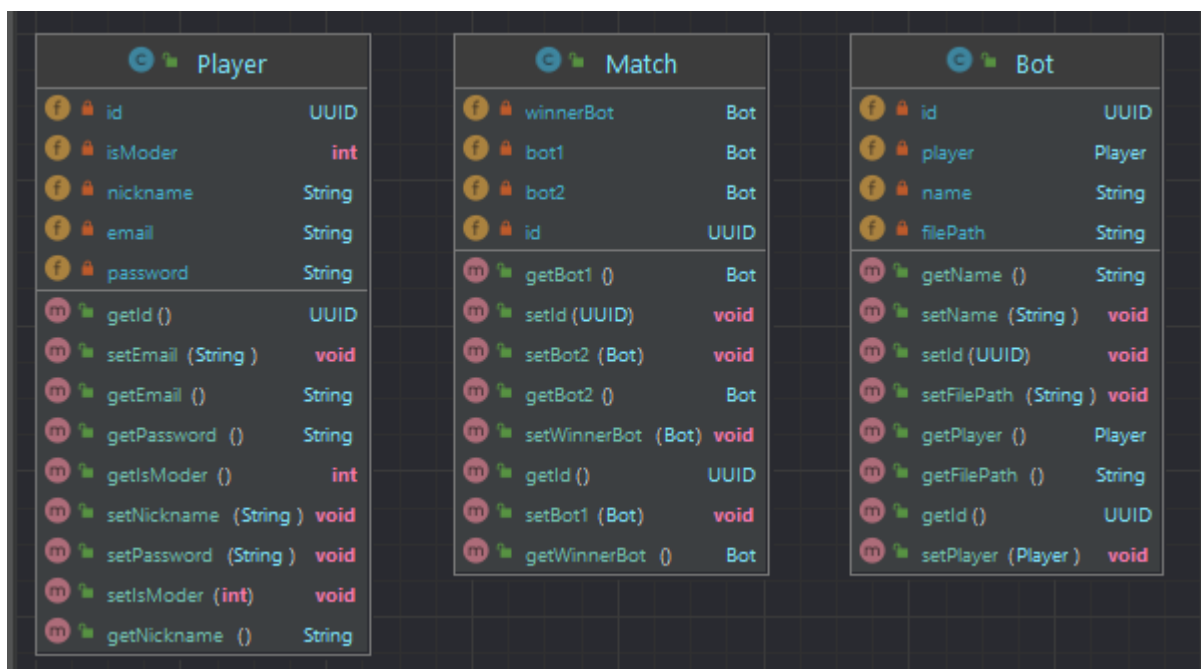
13 Диаграмма потоков данных





14 Диаграмма классов

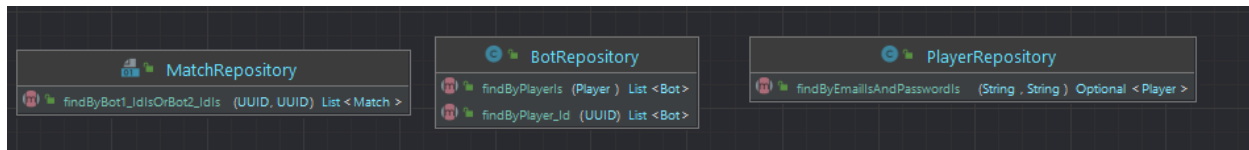
14.1 Диаграмма классов сущностей



Поля каждого из этих классов эквивалентны атрибутам одноименных таблиц в БД.

Player - сущность пользователя, Match - сущность “игры”, Bot - сущность стратегии пользователя.

14.2 Диаграмма классов репозиториев



Каждый из этих классов относится к слою доступа к данным, т.е. обеспечивает взаимодействие приложения с базой данных.

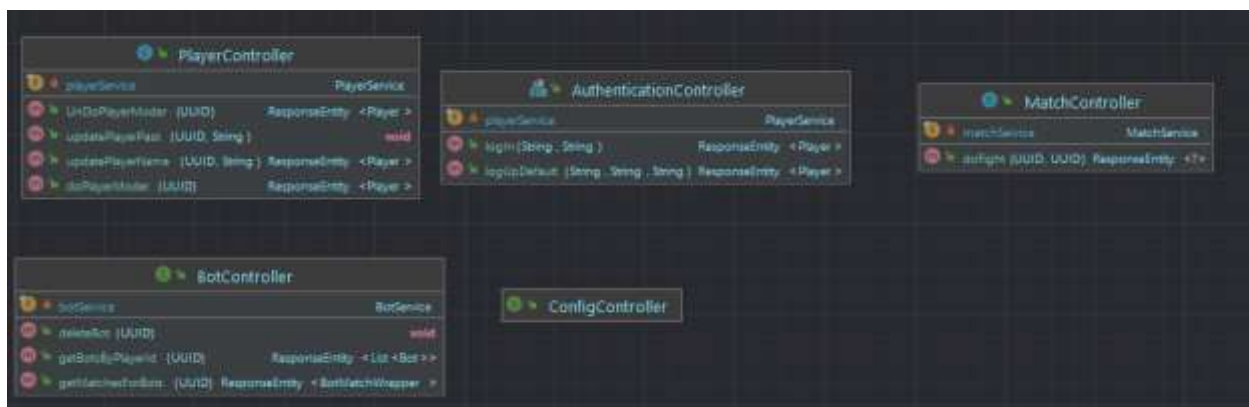
Через Match, Bot и Player репозитории обеспечивается взаимодействие с одноименными таблицами БД.

14.3 Диаграмма классов сервисов



Каждый из этих классов является частью слоя бизнес-логики, т.е. выступают связующим звеном между слоем доступа к данным и контроллерами. Все вычисления и алгоритмы находятся в этих классах.

14.4 Диаграмма классов контроллеров



Эти классы необходимы для общения с клиентом. Они получают запросы, вызывают методы слоя бизнес-логики и отправляют ответы назад на клиент.

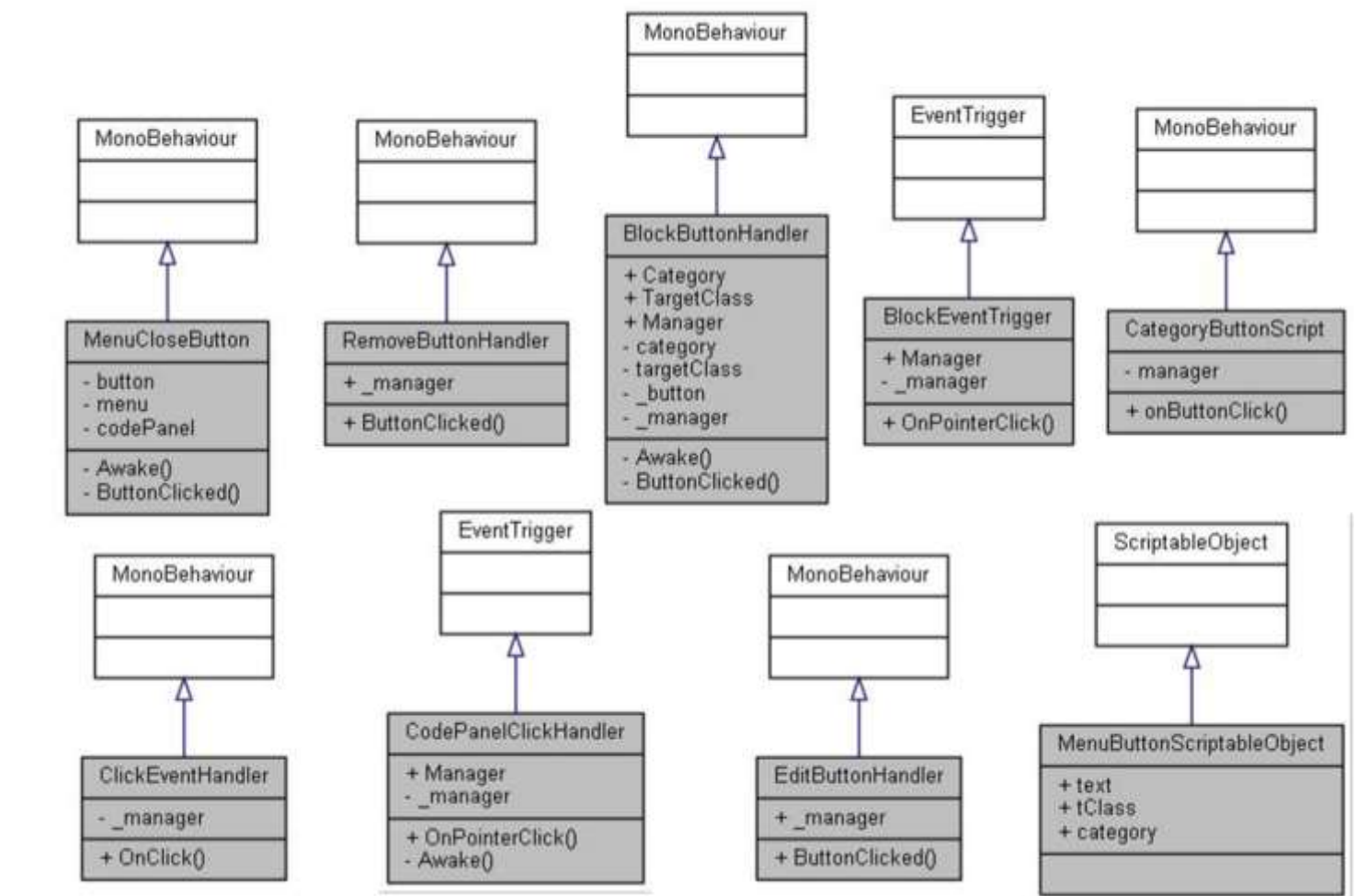
14.5 Диаграмма служебных классов



Класс `BotegeBackApplication` является точкой входа для приложения, он запускает цепочку загрузки нужных зависимостей и классов, благодаря которым приложение функционирует.

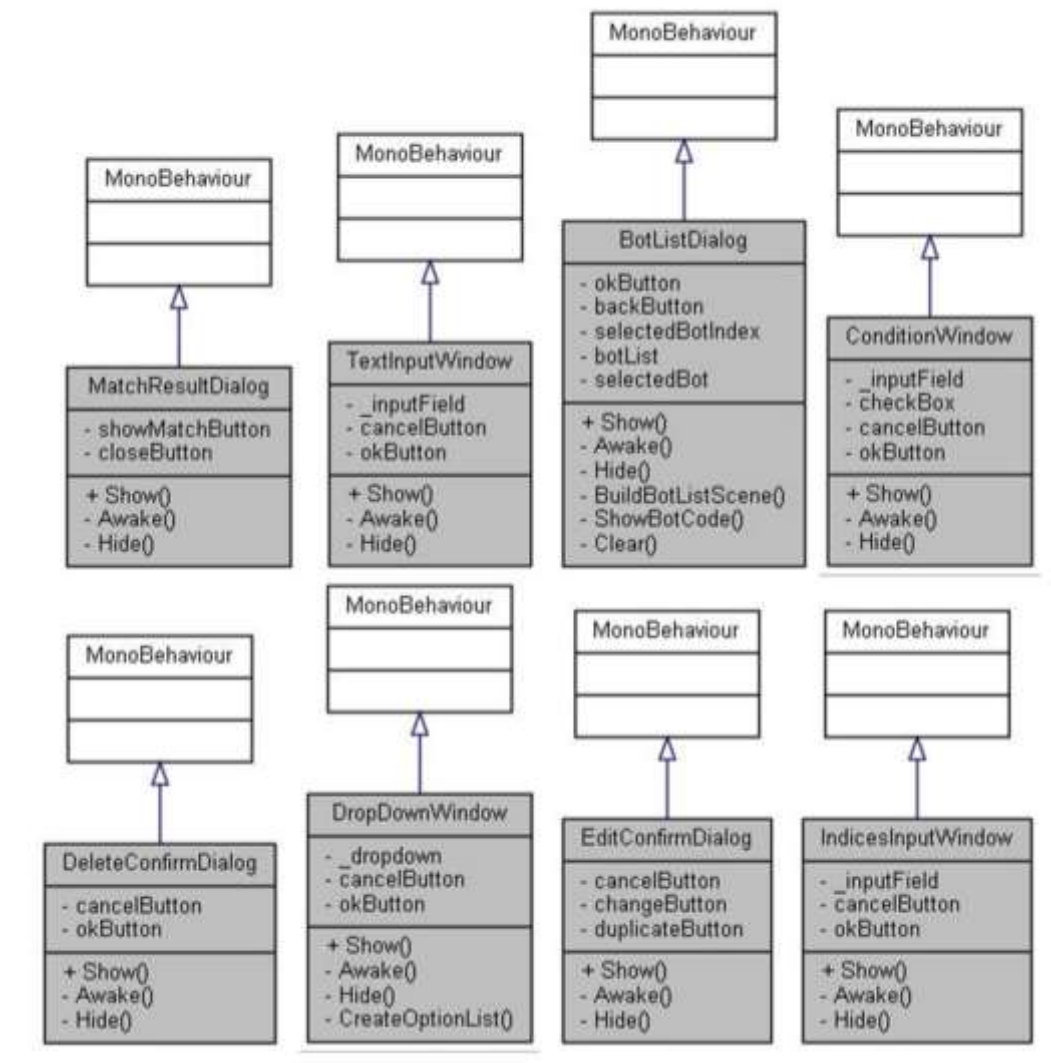
Класс `SwaggerConfig` является классом конфигурация для инструмента создания документации API Swagger.

14.6 Диаграмма классов пакета `ButtonScript`



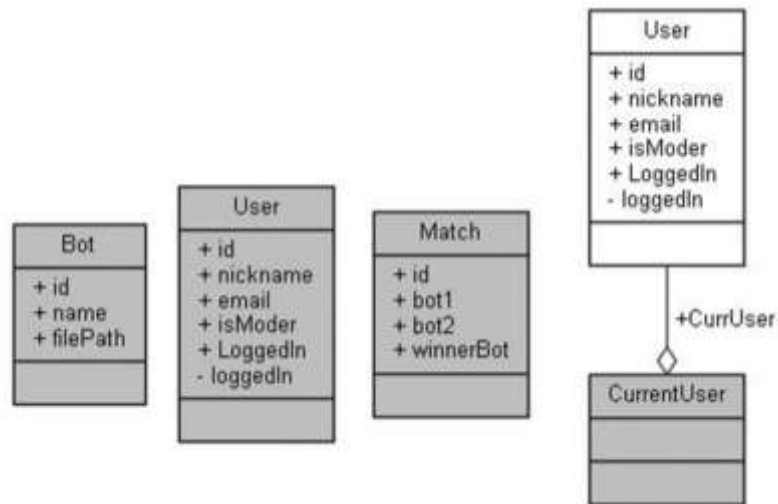
В этот пакет входят классы обработчиков событий клиента. Каждый класс отвечает за свой элемент интерфейса.

14.7 Диаграмма классов пакета `Dialog`



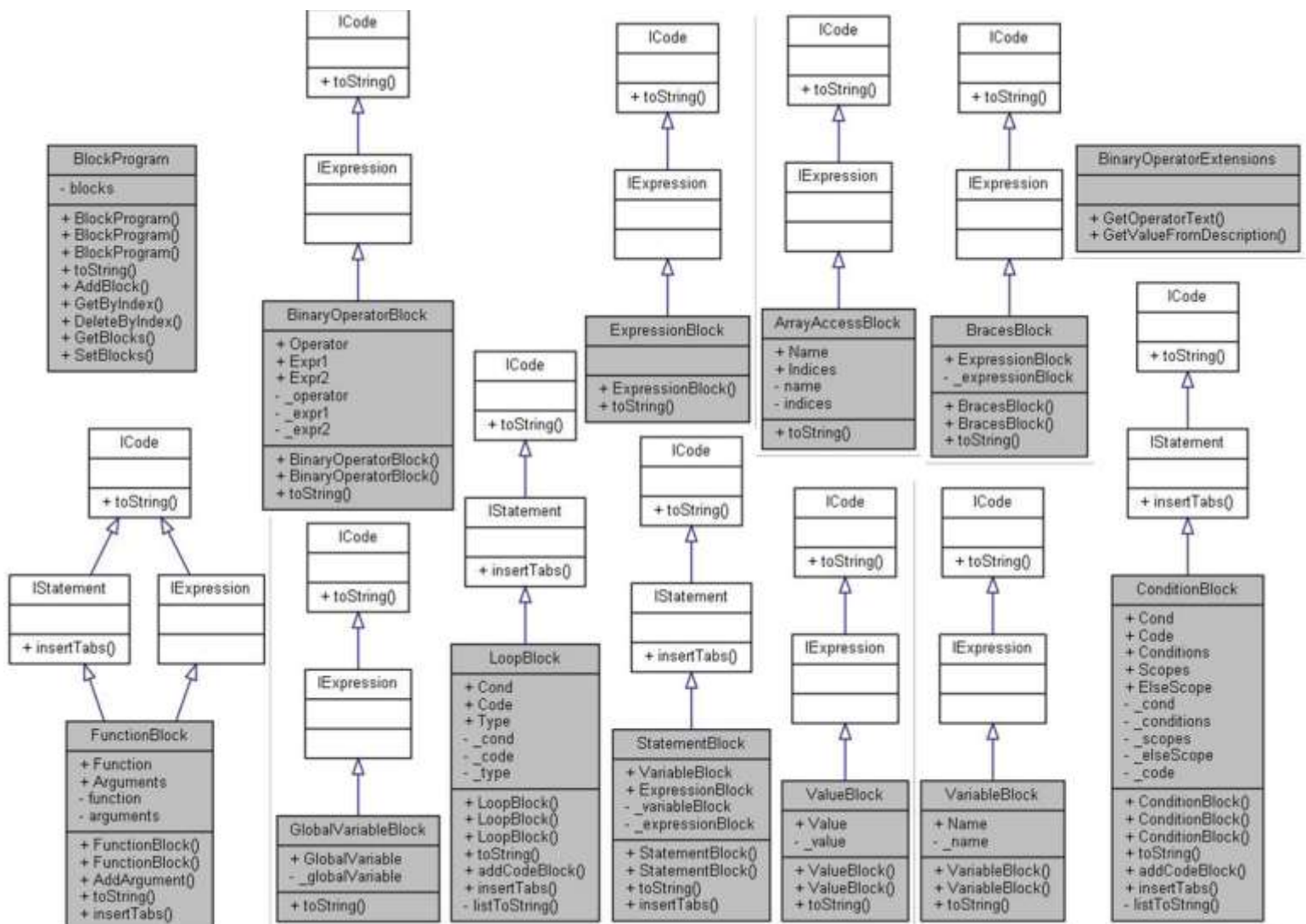
В этот пакет входят классы всплывающих окон интерфейса. Каждое окно обладает своими элементами управления и отвечает за ввод какой-либо информации пользователем: подтверждение действия, ввод текста в текстовое поле, выбор из выпадающего списка.

14.8 Диаграмма классов пакета Entity



В этот пакет входят классы сущностей, данные которых клиент получает по запросу к серверу.

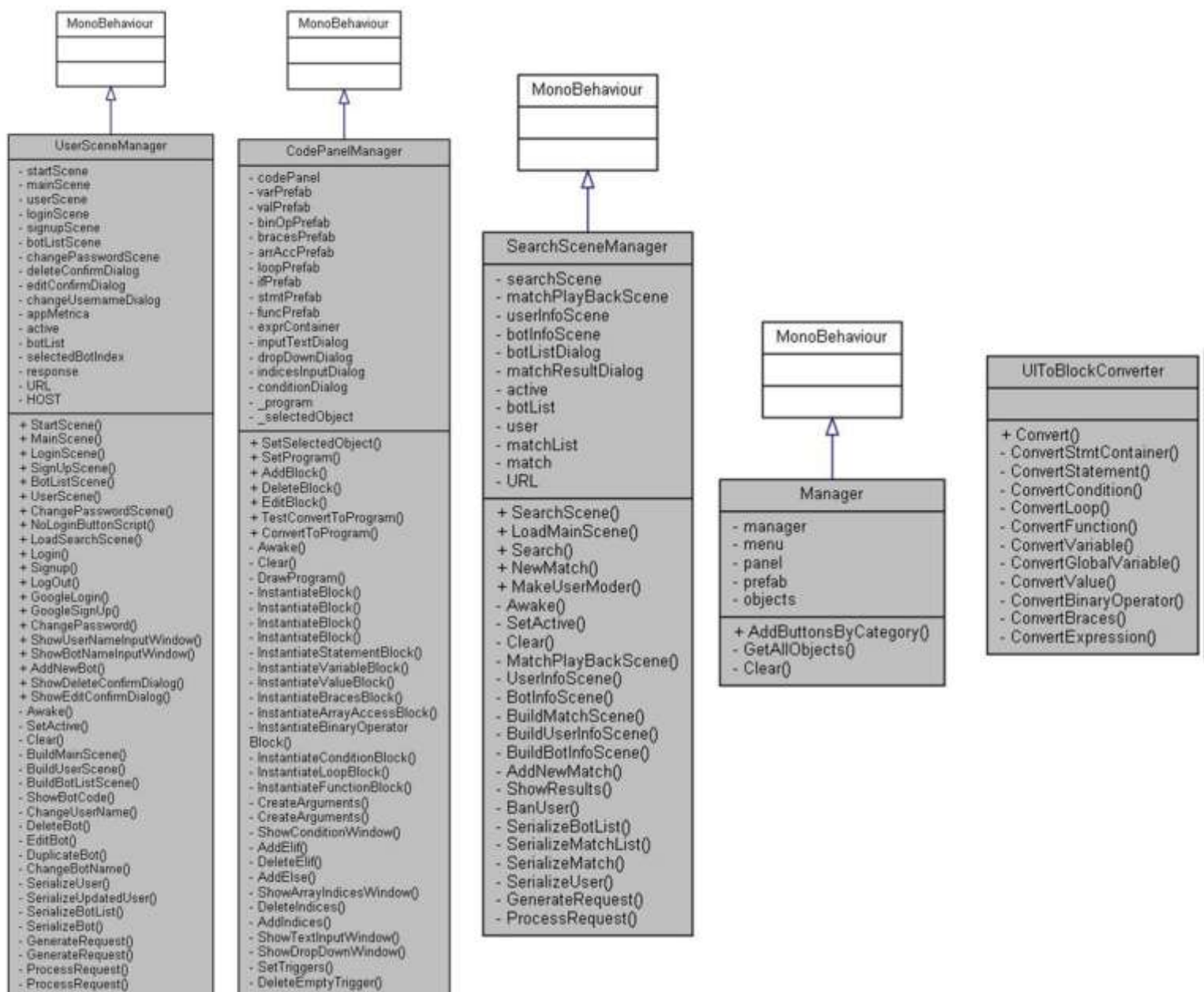
14.9 Диаграмма классов пакета Model



В этот пакет входят классы, моделирующие кодовые конструкции, необходимые для редактора кода бота. По коду бота строится BlockProgram, состоящая из блоков кода IStatement и IExpression. Полученная программа выводится на экран редактора кода.

При сохранении ботов выстроенные на экране блоки преобразуются в BlockProgram для сохранения в базе данных. Модель повторяет модель, используемую интерпретатором при симуляции матчей между ботами.

14.10 Диаграмма классов пакета SceneManager



В этот пакет входят классы необходимые для построения сцен пользовательского интерфейса и переключения между ними.

15 Список планируемых работ

- Анализ предметной области
- Анализ аналогов
- Постановка задачи
- Разработка макетов интерфейса
- Определение используемой платформы
- Построение use case диаграмм
- Создание доски Trello и репозитория GitHub
- Корректировка технического задания
- Реализация слоя доступа к БД
- Реализация интерфейса
- Построение UML диаграмм
- Подключение swagger
- Разработка модели кода бота
- Создание сценариев воронок
- Реализация интерпретатора хода бота
- Реализация модуля симуляции матчей ботов
- Размещение backend-части и БД на хостинге
- Реализация модуля авторизации

16 Используемая платформа

Сервер:

- ОС Windows 10
- Frameworks: Spring boot 2.6.6, Hibernate, Lombok.
- БД - PostgreSQL
- Язык разработки - Java v. 17
- Используемая IDE: IntelliJ IDEA 2021.3.2 (Ultimate Edition)
- Инструмент ведения документации API Swagger
- Система контроля версий - Git
- Хостинг исходного кода - Heroku

Клиент:

- ОС Windows 10
- Microsoft .NET Framework 4.5
- Язык разработки - C#
- Игровой движок Unity 2020.3.32f1
- Используемая IDE: IntelliJ Rider
- Система контроля версий - Git
- Макеты интерфейсов - AxureRP

17 Сценарии воронок

Регистрация после предпросмотра

1 Вход без авторизации Событие: WithOutLoginButtonClicked

2 Возвращение на старто Событие: ReturnToStartSceneButtonClicked

3 Регистрация Событие: SuccessfulSignUp

Добавить шаг

Детальные настройки

События между шагами

Разрешены

Запрещены

Окно воронки

Без ограничений

Ограничить

Регистрация событий

Из любой сессии

Из одной сессии

Подсчет метрик

По пользователям

По устройствам

Сценарий воронки регистрации после предпросмотра определяет, какой процент пользователей решит создать учетную запись после использования приложения в качестве наблюдателя.

Редактирование бота после матча

1

Участие в матче

Событие: MatchFinished

+

2

Редактирование бота

Событие: BotEditorOpened

+

+

Добавить шаг

Детальные настройки ^

События между шагами ⓘ

Разрешены

Запрещены

Окно воронки ⓘ

Без ограничений

Ограничить

Регистрация событий ⓘ

Из любой сессии

Из одной сессии

Подсчет метрик ⓘ

По пользователям

По устройствам

Сценарий воронки редактирования бота после матча призвана определить, какое количество пользователей решит отредактировать своего бота после участия этого бота в матче.

Просмотр хода матча после его завершения

1

Участие в матче

Событие: MatchFinished

+

2

Просмотр хода матча

Событие: MatchPlayBackButtonClicked

+

+

Добавить шаг

Детальные настройки ^

События между шагами ⓘ

Разрешены

Запрещены

Окно воронки ⓘ

Без ограничений

Ограничить

Регистрация событий ⓘ

Из любой сессии

Из одной сессии

Подсчет метрик ⓘ

По пользователям

По устройствам

Сценарий воронки просмотра хода матча определяет, какое количество пользователей решит проанализировать матч, посмотрев его ход после получения результата.

18 Календарный план

| Задача | Дата начала | Дата окончания | Длительность |
|--|-------------|----------------|--------------|
| АТТЕСТАЦИЯ 1 | | | |
| | 25.02.2022 | 20.03.2022 | 23 |
| Анализ предметной области | 02.03.2022 | 10.03.2022 | 8 |
| Анализ аналогов | 02.03.2022 | 04.03.2022 | 2 |
| Постановка задачи | 02.03.2022 | 06.03.2022 | 4 |
| Разработка макетов интерфейса | 10.03.2022 | 16.03.2022 | 6 |
| Определение используемой платформы | 06.03.2022 | 07.03.2022 | 1 |
| Построение use case диаграммы | 06.03.2022 | 08.03.2022 | 2 |
| Создание доски Trello и репозитория GitHub | 07.03.2022 | 13.03.2022 | 6 |
| Корректировка технического задания | 19.03.2022 | 20.03.2022 | 1 |
| АТТЕСТАЦИЯ 2 | | | |
| | 24.03.2022 | 29.04.2022 | 36 |
| Реализация слоя доступа к БД | 24.03.2022 | 25.04.2022 | 32 |
| Реализация интерфейса | 04.04.2022 | 24.04.2022 | 20 |
| Построение UML диаграмм | 25.04.2022 | 28.04.2022 | 3 |
| Подключение swagger | 19.04.2022 | 22.04.2022 | 3 |
| Разработка модели кода бота | 04.04.2022 | 20.04.2022 | 16 |
| Создание сценариев воронок | 24.04.2022 | 28.04.2022 | 4 |

