# Учреждение образования

# «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №2 Проектирование и разработка ПО на языке UML

> Студент гр. 953501: Рисунова Е.В. Руководитель: Гриценко Н.Ю.

# СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ	3
2 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ МОДЕЛИ	4
3 РАЗРАБОТКА МОДЕЛИ ДАННЫХ	5
4 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ	6
4.1 ДИАГРАММЫ КЛАССОВ	
4.2 ДИАГРАММА СОСТОЯНИЙ	8
4.3 ДИАГРАММА АКТИВНОСТИ	9
4.4 ДИАГРАММА КОМПОНЕНТОВ	10
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	11

# 1 ПОСТАНОВКА ЗАДАЧИ

Предметная область: «Игровалютная биржа».

Разработать веб-приложение для иммитации работы онлайн биржи. Тип СУБД – MySQL.

Среда проектирования – draw.io

Языки разработки – Python, JavaScript (фреймворк Angular CLI 13.0.4)

Среда разработки – Visual Studio Code 2020.

# 2 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ МОДЕЛИ

Функциональную модель предметной области представим в виде диаграммы вариантов использования в нотации UML, представляющей систему в виде набора варианта использования и актеров, взаимодействующих с ними.

В рамках предметной области можно выделить одного актера:

– Пользователь;

Диаграмма вариантов использования системы приведена на рисунке 2.1.



Рисунок 2.1 — Диаграмма вариантов использования Согласно приведенной диаграмме пользователь применяет следующие варианты использования:

- Просмотреть графики и курсы коинов;
- Просмотреть доступные предложения по играм;
- Приобрести выбранное предложение;
- Добавить ордер на покупку/продажу коинов;
- Проверить свой баланс.

Роль администратора не предусмотрена, все пользователи в системе равны.

## 3 РАЗРАБОТКА МОДЕЛИ ДАННЫХ

Так как приложение подразумевает под собой постоянный пересчет данных от всех пользователей, данные будут сразу заноситься в базу данных и выводится на UI через обращение API к базе данных.

Преимущество такого проектирования приложения — меньший риск потери данных при ошибке на стороне клиента, актуальное состояние у всех пользователей, а также быстрый пересчет данных на уровне таблиц.

#### 4 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

### 4.1 Диаграммы классов

Разработаем диаграммы классов, взаимодействия с пользователем, приложения с разделением на пакеты.

Диаграмма классов приложения приведена на рисунке 4.1.

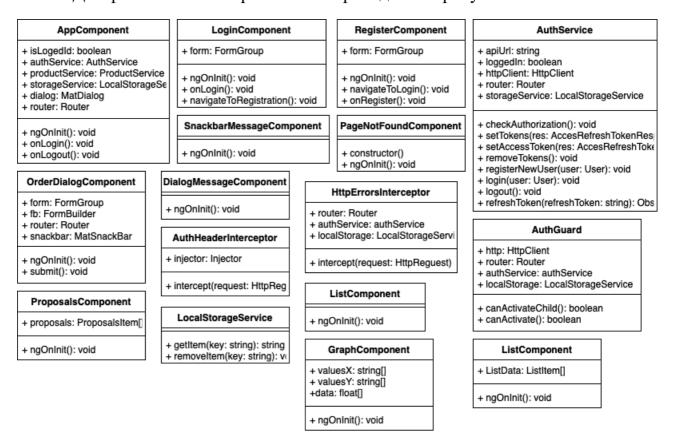


Рисунок 4.1 — Диаграмма классов для взаимодействия с пользователем приложения

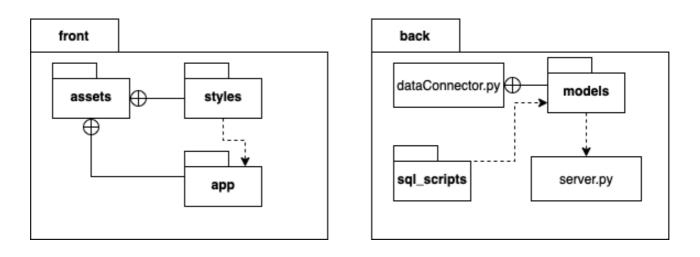


Рисунок 4.2 – Диаграмма пакетов приложения

На диаграмме представлены следующие классы и пакеты:

- Front все классы и пакеты для взаимодействия с пользователем:
  - Assets пакет содержащий все ресурсы приложения.
  - Styles пакет для UI стилей приложения.
  - Арр Классы для взаимодействия с UI.
- Back все классы и пакеты для API сервера:
  - dataConnector.py класс для обращения к базе данных.
  - server.py класс для обращения UI к API с помощью запросов.
  - models пакет для запуска SQL скриптов и формирования тела ответа.
  - sql scripts пакет с sql скриптами.

## 4.2 Диаграмма состояний

Далее разработаем диаграмму состояний приложения (рисунок 4.3).

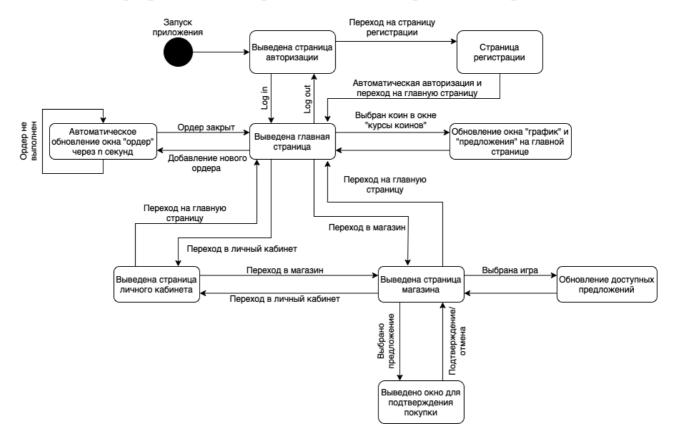


Рисунок 4.3 – Диаграмма состояний приложения

После запуска приложения система принимает состояние «Авторизация». Далее пользователю доступна регистрация либо вход в приложение по адресу электронной почты и паролю. После этого приложение принимает состояние «Главная страница».

Дальнейшие состояния зависят от функции или операции, выбранной пользователем.

Если пользователь создает ордер на покупку/продажу, появляется состояние «Открытый ордер» с автоматическим обновлением и закрытием окна после выполнения ордера.

При выборе коина в окне «Курсы коинов» на главной странице обновляется окно «График» и «Предложения».

При переходе на другие экраны, состояние приложение меняется на состояния «Личный кабинет» или «Магазин».

## 4.3 Диаграмма активности

Разработаем диаграмму активности для варианта использования «Приобрести промокод» (рисунок 4.4).

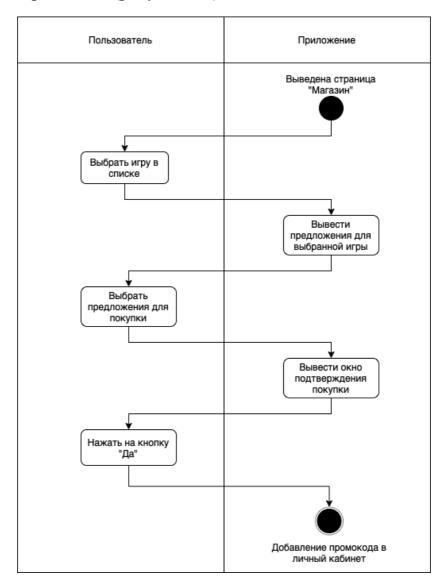


Рисунок 4.4 — Диаграмма активности варианта использования «Приобрести промокод»

Активность предполагает состояние «Выведена страница Магазин» и включает следующие действия:

- Пользователь выбирает игру в списке;
- Приложение выводит доступные предложения по выбранной игре;
- Пользователь выбирает предложение;
- Приложение выводит окно с подтверждением покупки;
- Пользователь нажимает кнопку подтверждения;
- Приложение выводит промокод на экран, добавляет его в личный кабинет

## 4.4 Диаграмма компонентов

Разработаем диаграмму компонентов системы (рисунок 4.5) этапа разработки.

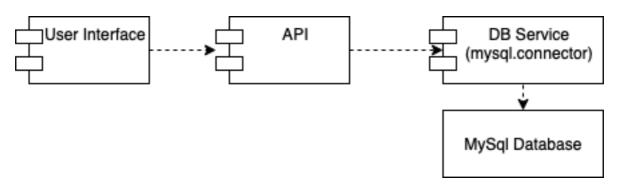


Рисунок 4.5 – Диаграмма компонентов

На этапе разработки система включает следующие компоненты:

- User Interface компонент запускаемого веб-приложения;
- API пакетный компонент, для обращения UI с помощью HTTP запросов;
- DB Service пакетный компонент, обеспечивающий взаимодействие с базой данных;
  - MySql Database компонент базы данных MySql.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Леоненков, А. Самоучитель UML2 / А. Леоненков. СПб.: БХВ-Петербург, 2014. 576 с.
- [2] <a href="https://flask.palletsprojects.com/en/2.2.x/">https://flask.palletsprojects.com/en/2.2.x/</a> Официальный сайт документации по Flask Api.
- [3] <a href="https://www.mysql.com">https://www.mysql.com</a> Официальный сайт документации MySQL.