

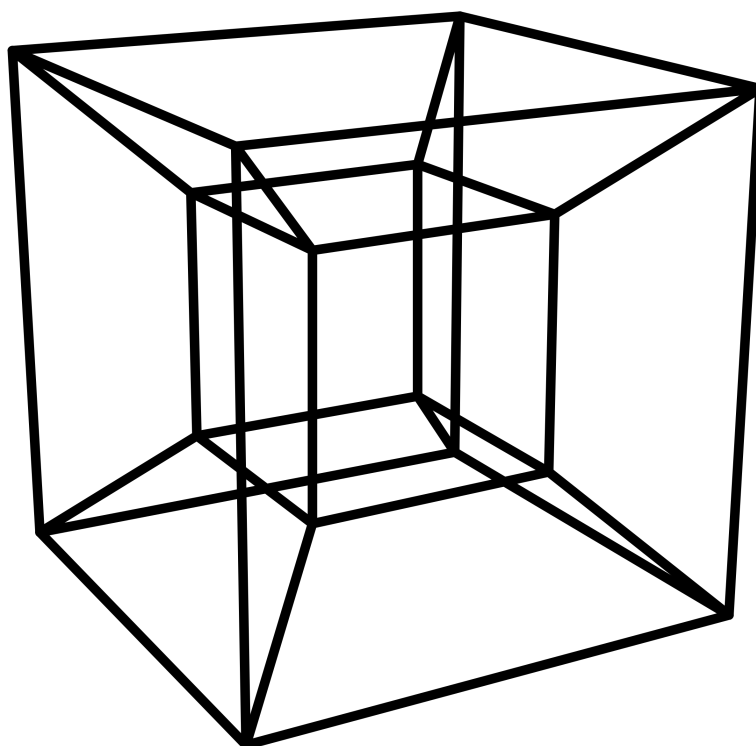
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский политехнический университет Петра Великого»

ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И КИБЕРБЕЗОПАСНОСТИ
ВЫСШАЯ ШКОЛА ПРОГРАММНОЙ ИНЖЕНЕРИИ

HIGH-LEVEL DESIGN

АГРЕГАТОР ЦИФРОВЫХ ФИНАНСОВЫХ АКТИВОВ «ТЕССЕРАКТ»
по дисциплине «Технологии разработки качественного программного обеспечения»



Выполнили студенты группы: 5130904/00104:

Почернин В. С.
Шиляев В. С.
Мурзаканов И. М.
Разукрантов В. Е.

Преподаватель:

Маслаков А. П.

Санкт-Петербург
2023

Содержание

1	Макет дизайна интерфейса	2
1.1	Список страниц	3
2	Архитектура приложения	3
3	Стек технологий	3
3.1	Backend	3
3.2	Frontend	4
4	Диаграмма классов	4
5	Схема базы данных	5
6	API	5
6.1	Аутентификация через логин и пароль	6
6.2	Аутентификация через Google OAuth	6
6.3	Регистрация пользователя	6
6.4	Получить список всех активов	7
6.5	Добавить актив в избранное	7
6.6	Удалить актив из избранного	8
6.7	Получить информацию о конкретном активе	8
6.8	Получить список избранных активов	8
6.9	Получить список диверсификаций	9
6.10	Создать диверсификацию	9
6.11	Получить информацию о конкретной диверсификации	10
6.12	Изменить пароль	10
7	Приложение 1: список страниц (экранов)	11

1 Макет дизайна интерфейса

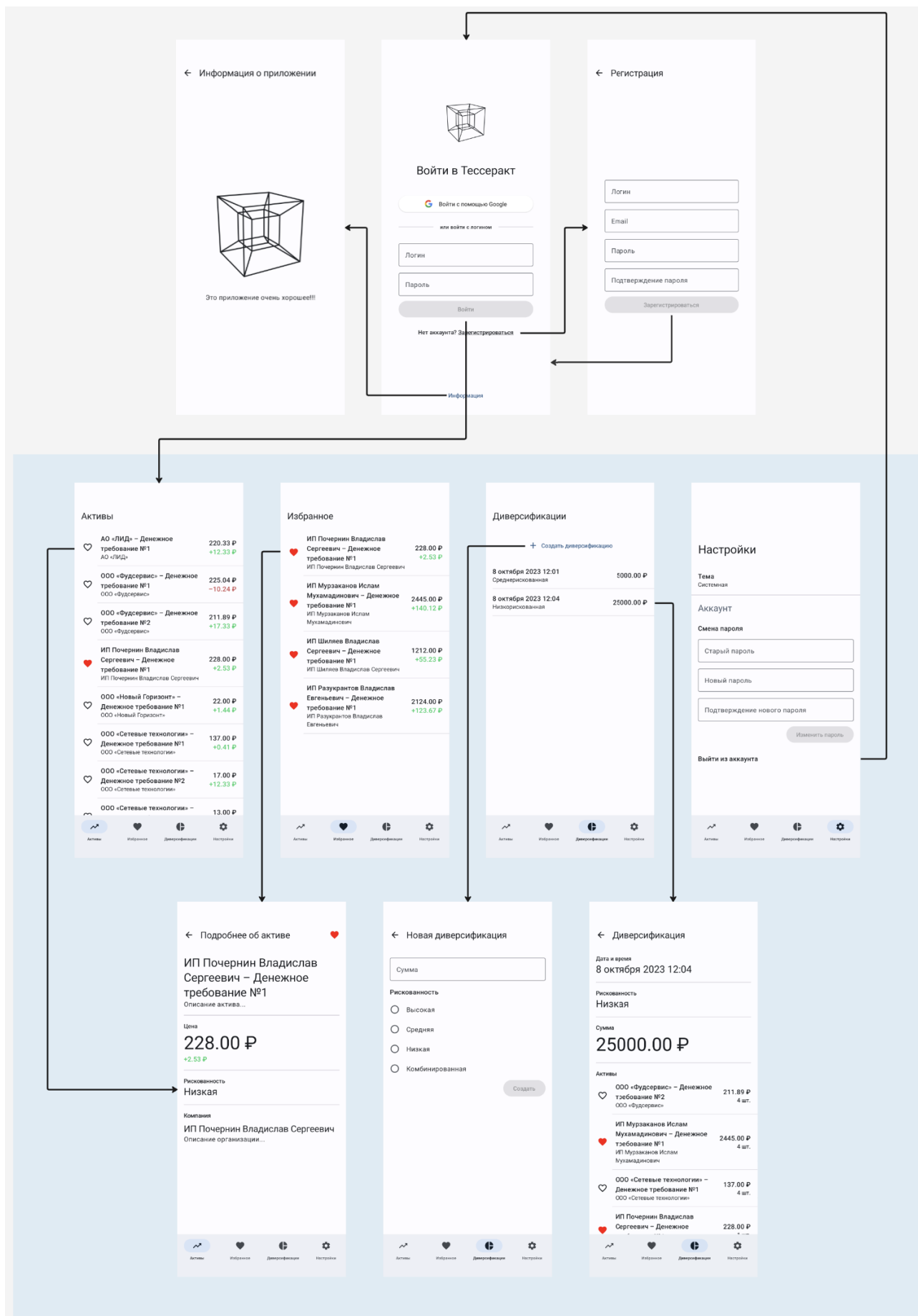


Рис. 1: Схема переходов между страницами

1.1 Список страниц

- 1) Страница входа (LoginPage).
- 2) Страница информации (InfoPage).
- 3) Страница регистрации (RegistrationPage).
- 4) Страница активов (AssetsPage).
- 5) Страница конкретного актива (AssetPage).
- 6) Страница избранных активов (FavouritesPage).
- 7) Страница диверсификаций (DiversificationsPage).
- 8) Страница создания диверсификации (DiversificationCreatePage).
- 9) Страница конкретной диверсификации (DiversificationPage).
- 10) Страница настроек (SettingsPage).

2 Архитектура приложения

Клиентская часть приложения реализует паттерны **Single Activity Application**, а также **MVVM (Model View ViewModel)**.

Пользователь взаимодействует с UI, написанном на **Compose**, который в свою очередь получает данные из **ViewModel** и отправляет действия пользователя во **ViewModel**.

Клиент же общается с сервером посредством предоставляемого сервером **API**, внутри которого происходит исполнение бизнес-логики и обращение к БД.

Используется модель **MVC**.

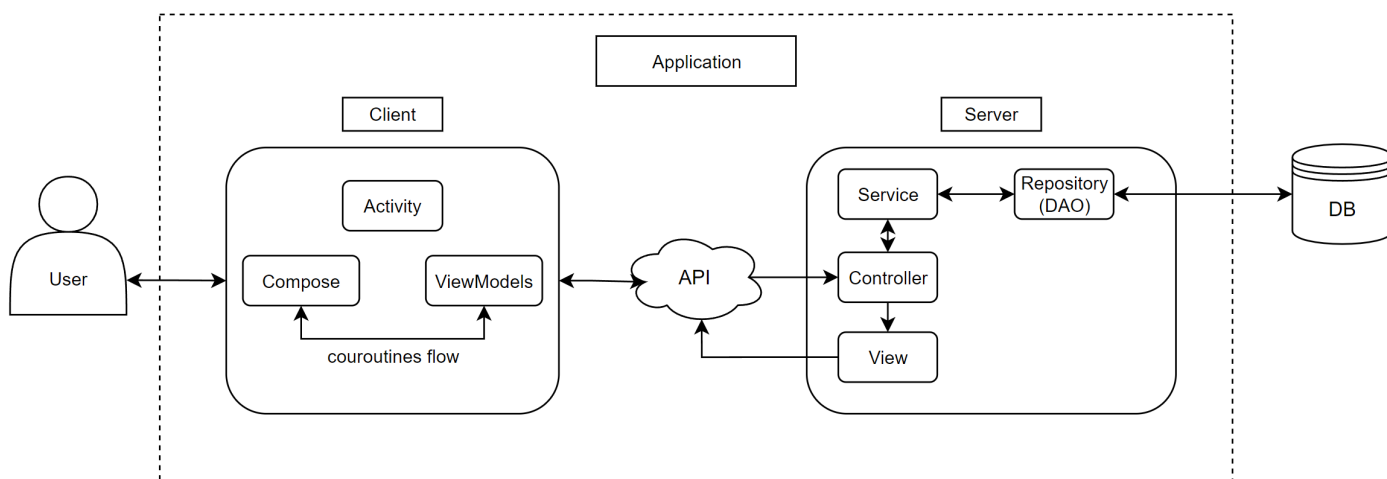


Рис. 2: Архитектура приложения

3 Стек технологий

3.1 Backend

- **Java** - язык программирования.
- **Spring Boot** - основной фреймворк.
- **Spring Web MVC** - веб среда Spring для работы с сетью.

- **Spring Security** - фреймворк, предоставляющий механизмы построения систем аутентификации и авторизации.
- **Spring Data JPA** - механизм для взаимодействия с сущностями базы данных.
- **PostgreSQL** - система управления базами данных.
- **Flyway** - система управления версиями баз данных.
- **Swagger** - инструмент для описания API.

3.2 Frontend

- **Kotlin** - язык программирования.
- **Kotlin Coroutines** - библиотека для асинхронной работы.
- **Ktor** - библиотека для работы с API.
- **Koin** - DI (Dependency Injection) фреймворк.
- **Compose** - библиотека для UI.

4 Диаграмма классов

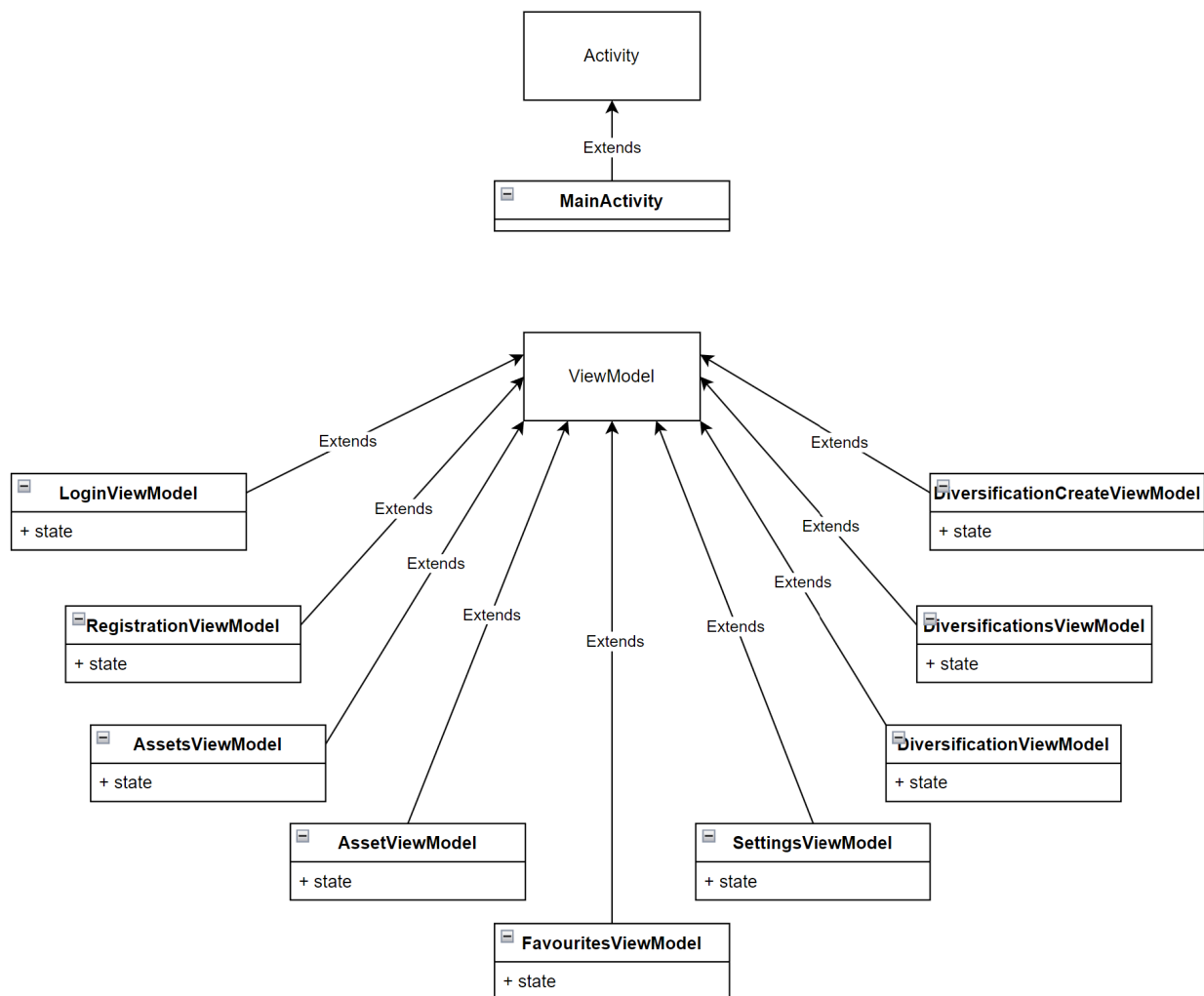


Рис. 3: Диаграмма классов клиентского приложения

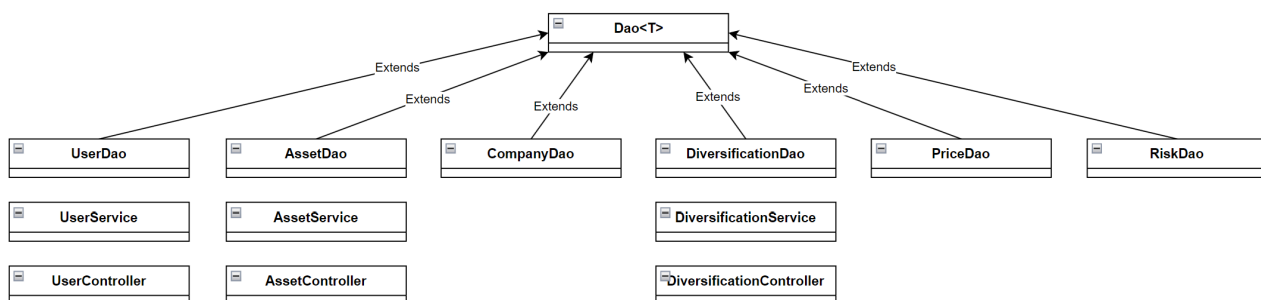


Рис. 4: Диаграмма классов серверного приложения

5 Схема базы данных

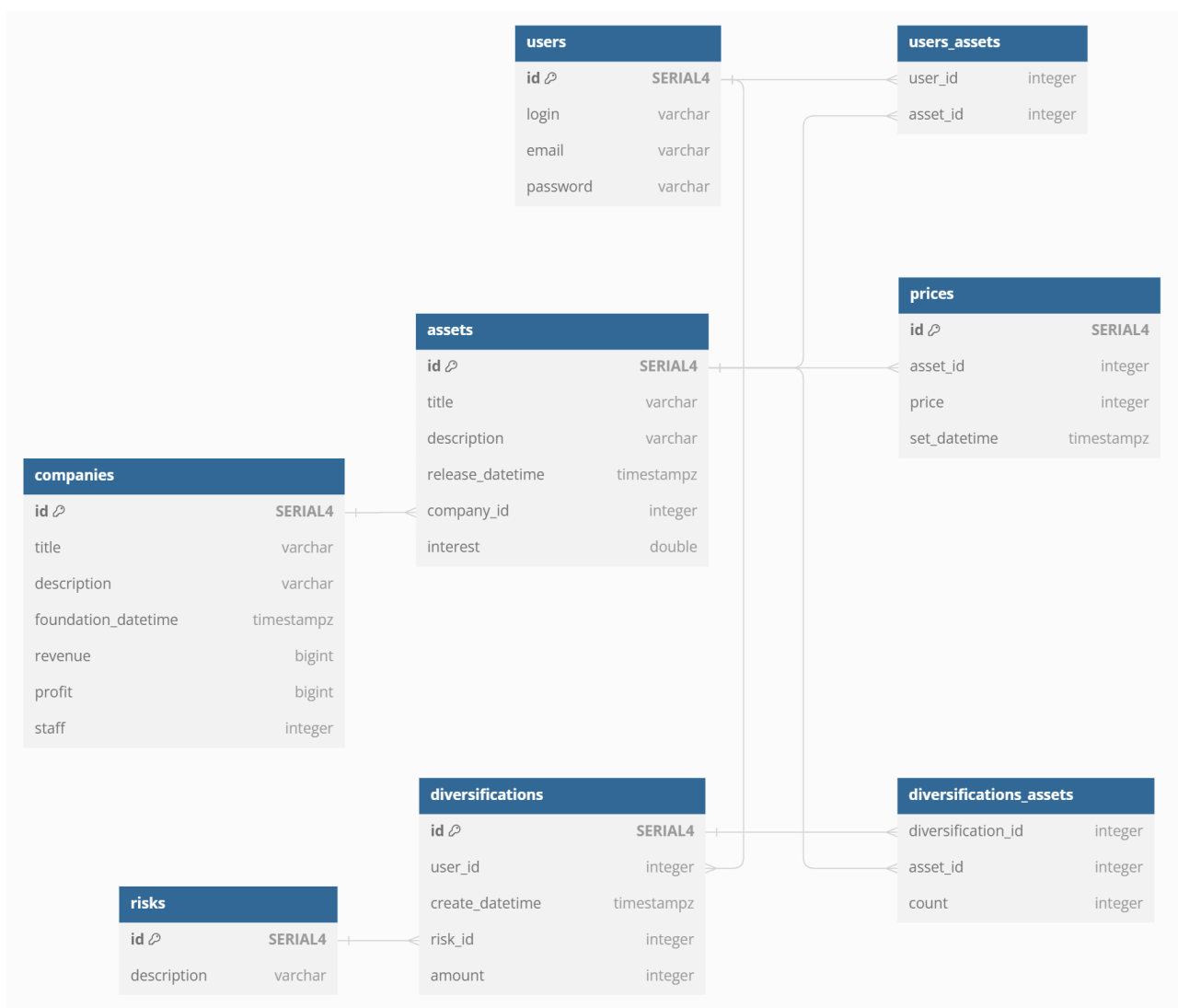


Рис. 5: Схема базы данных

6 API

API будет поддерживать следующие HTTP коды:

- 200 OK - успешные GET, PUT, DELETE или POST без создания.
- 201 Created - успешный POST с созданием.
- 400 Bad Request - некорректное тело запроса.

- 401 `Unauthorized` - отсутствуют данные аутентификации.
- 404 `Not Found` - запрашивается несуществующий ресурс.
- 500 `Internal Server Error` - непредвиденная ошибка.

6.1 Аутентификация через логин и пароль

- HTTP метод: `GET`.
- Путь: `/login/tesseract`.
- Входные параметры:
 - Тело (JSON):
 - * `String login`;
 - * `String password`;
- Выходные параметры:
 - Тело (JSON):
 - * `String token`;
 - HTTP коды:
 - * 200 `OK` - успешная авторизация.
 - * 400 `Bad Request` - некорректное тело запроса.
 - * 401 `Unauthorized` - несуществующая комбинация логин/пароль.
 - * 500 `Internal Server Error` - непредвиденная ошибка.

6.2 Аутентификация через Google OAuth

- HTTP метод: `GET`.
- Путь: `/login/google`.
- Входные параметры:
 - Тело (JSON):
 - * `String googleToken`;
- Выходные параметры:
 - Тело (JSON):
 - * `String token`;
 - HTTP коды:
 - * 200 `OK` - успешная авторизация.
 - * 400 `Bad Request` - некорректное тело запроса.
 - * 401 `Unauthorized` - некорректный Google токен.
 - * 500 `Internal Server Error` - непредвиденная ошибка.

6.3 Регистрация пользователя

- HTTP метод: `POST`.
- Путь: `/users`.
- Входные параметры:
 - Тело (JSON):

- * String login;
- * String password;
- * String email;
- Выходные параметры:
 - HTTP коды:
 - * 201 Created - успешная регистрация.
 - * 400 Bad Request - некорректное тело запроса.
 - * 500 Internal Server Error - непредвиденная ошибка.

6.4 Получить список всех активов

- HTTP метод: GET.
- Путь: /assets.
- Входные параметры:
 - Заголовок: данные аутентификации;
 - Строка запроса: id последнего загруженного актива (опционально).
- Выходные параметры:
 - Тело (JSON) - список:
 - * String assetTitle; - название актива.
 - * String companyTitle; - название компании.
 - * Integer assetPrice; - стоимость актива.
 - * Integer assetDiff; - изменение стоимости актива за последний месяц.
 - * Boolean favouriteStatus; - статус избранного актива.
 - HTTP коды:
 - * 200 OK - успешное получение списка активов.
 - * 400 Bad Request - некорректное тело запроса.
 - * 401 Unauthorized - отсутствуют данные аутентификации.
 - * 500 Internal Server Error - непредвиденная ошибка.

6.5 Добавить актив в избранное

- HTTP метод: POST.
- Путь: favourites/{id}.
- Входные параметры:
 - Заголовок: данные аутентификации.
 - Путь: id добавляемого в избранное актива.
- Выходные параметры:
 - HTTP коды:
 - * 200 OK - успешное добавление актива в избранное, если он там уже был.
 - * 201 Created - успешное добавление актива в избранное.
 - * 400 Bad Request - некорректное тело запроса.
 - * 401 Unauthorized - отсутствуют данные аутентификации.
 - * 404 Not Found - указан несуществующий id актива.
 - * 500 Internal Server Error - непредвиденная ошибка.

6.6 Удалить актив из избранного

- HTTP метод: DELETE.
- Путь: `favourites/{id}`.
- Входные параметры:
 - Заголовок: данные аутентификации.
 - Путь: `id` удаляемого из избранного актива.
- Выходные параметры:
 - HTTP коды:
 - * 200 OK - успешное удаление актива из избранного (либо отсутствие соответствующего `id` актива).
 - * 400 Bad Request - некорректное тело запроса.
 - * 401 Unauthorized - отсутствуют данные аутентификации.
 - * 500 Internal Server Error - непредвиденная ошибка.

6.7 Получить информацию о конкретном активе

- HTTP метод: GET.
- Путь: `/assets/{id}`.
- Входные параметры:
 - Заголовок: данные аутентификации.
 - Путь: `id` запрашиваемого актива.
- Выходные параметры:
 - Тело (JSON):
 - * String `assetTitle`; - название актива.
 - * String `assetDescription`; - описание актива.
 - * Integer `assetPrice`; - стоимость актива.
 - * Integer `assetDiff`; - изменение стоимости актива за последний месяц.
 - * String `companyTitle`; - название компании.
 - * String `companyDescription`; - описание компании.
 - * Integer `risk`; - степень рискованности актива.
 - * Boolean `favouriteStatus`; - статус избранного актива.
 - HTTP коды:
 - * 200 OK - успешное получение информации об активе.
 - * 400 Bad Request - некорректное тело запроса.
 - * 401 Unauthorized - отсутствуют данные аутентификации.
 - * 404 Not Found - отсутствие запрашиваемого `id` актива.
 - * 500 Internal Server Error - непредвиденная ошибка.

6.8 Получить список избранных активов

- HTTP метод: GET.
- Путь: `/favourites`.
- Входные параметры:

- Заголовок: данные аутентификации.
- Строка запроса: id последнего загруженного избранного актива (опционально).
- Выходные параметры:
 - Тело (JSON) - список:
 - * String assetTitle; - название актива.
 - * String companyTitle; - название компании.
 - * Integer assetPrice; - стоимость актива.
 - * Integer assetDiff; - изменение стоимости актива за последний месяц.
 - * Boolean favouriteStatus; - статус избранного актива.
 - HTTP коды:
 - * 200 OK - успешное получение списка активов.
 - * 400 Bad Request - некорректное тело запроса.
 - * 401 Unauthorized - отсутствуют данные аутентификации.
 - * 500 Internal Server Error - непредвиденная ошибка.

6.9 Получить список диверсификаций

- HTTP метод: GET.
- Путь: /diversifications.
- Входные параметры:
 - Заголовок: данные аутентификации.
 - Строка запроса: id последней загруженной диверсификации (опционально).
- Выходные параметры:
 - Тело (JSON) - список:
 - * Timestamp createDatetime; - дата и время создания диверсификации.
 - * Integer risk; - степень рискованности диверсификации.
 - * Integer amount; - сумма диверсификации.
 - HTTP коды:
 - * 200 OK - успешное получение списка диверсификаций.
 - * 400 Bad Request - некорректное тело запроса.
 - * 401 Unauthorized - отсутствуют данные аутентификации.
 - * 500 Internal Server Error - непредвиденная ошибка.

6.10 Создать диверсификацию

- HTTP метод: POST.
- Путь: /diversifications.
- Входные параметры:
 - Заголовок: данные аутентификации.
 - Тело (JSON):
 - * Integer amount; - сумма создаваемой диверсификации.
 - * Integer risk; - степень рискованности создаваемой диверсификации.
- Выходные параметры:

- HTTP коды:
 - * 201 Created - успешное создание диверсификации.
 - * 400 Bad Request - некорректное тело запроса.
 - * 401 Unauthorized - отсутствуют данные аутентификации.
 - * 500 Internal Server Error - непредвиденная ошибка.

6.11 Получить информацию о конкретной диверсификации

- HTTP метод: GET.
- Путь: /diversifications/{id}.
- Входные параметры:
 - Заголовок: данные аутентификации.
 - Путь: id запрашиваемой диверсификации.
- Выходные параметры:
 - Тело (JSON):
 - * Timestamp createDatetime; - дата и время создания диверсификации.
 - * Integer risk; - уровень рискованности диверсификации.
 - * Integer amount - сумма диверсификации.
 - * Список:
 - String assetTitle; - название актива.
 - String companyTitle; - название компании.
 - Integer priceSum; - **старая** цена актива (то есть цена на момент создания диверсификации).
 - Integer count; количество активов данного вида.
 - HTTP коды:
 - * 200 OK - успешное получение диверсификации.
 - * 400 Bad Request - некорректное тело запроса.
 - * 401 Unauthorized - отсутствуют данные аутентификации.
 - * 404 Not Found - отсутствует диверсификация с таким id (либо запрашиваем не свою диверсификацию).
 - * 500 Internal Server Error - непредвиденная ошибка.

6.12 Изменить пароль

- HTTP метод: PUT.
- Путь: /users.
- Входные параметры:
 - Заголовок: данные аутентификации.
 - Тело (JSON):
 - * String oldPassword; - старый пароль.
 - * String newPassword; - новый пароль.
- Выходные параметры:
 - HTTP коды:
 - * 200 OK - успешное изменение пароля.
 - * 400 Bad Request - некорректное тело запроса.
 - * 401 Unauthorized - отсутствуют данные аутентификации.
 - * 500 Internal Server Error - внутренняя ошибка сервера.

7 Приложение 1: список страниц (экранов)

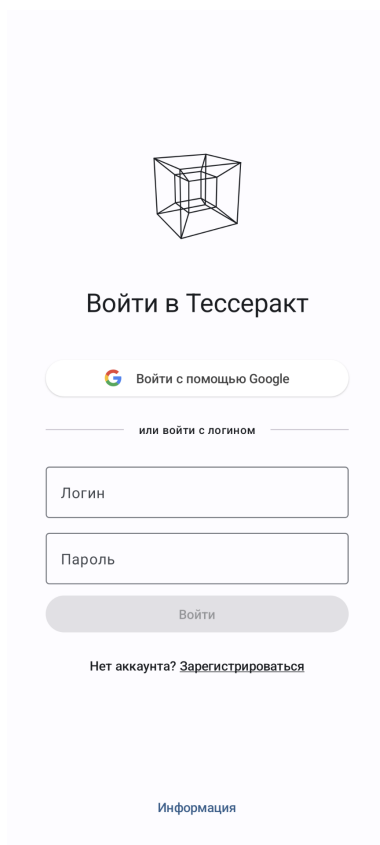


Рис. 6: Страница входа (LoginPage)



Рис. 7: Страница информации (InfoPage)

← Регистрация

Логин

Email

Пароль

Подтверждение пароля

Зарегистрироваться

Рис. 8: Страница регистрации (RegistrationPage)

Активы		
♡	АО «ЛИД» – Денежное требование №1 АО «ЛИД»	220.33 ₽ +12.33 ₽
♡	ООО «Фудсервис» – Денежное требование №1 ООО «Фудсервис»	225.04 ₽ -10.24 ₽
♡	ООО «Фудсервис» – Денежное требование №2 ООО «Фудсервис»	211.89 ₽ +17.33 ₽
♥	ИП Почернин Владислав Сергеевич – Денежное требование №1 ИП Почернин Владислав Сергеевич	228.00 ₽ +2.53 ₽
♡	ООО «Новый Горизонт» – Денежное требование №1 ООО «Новый Горизонт»	22.00 ₽ +1.44 ₽
♡	ООО «Сетевые технологии» – Денежное требование №1 ООО «Сетевые технологии»	137.00 ₽ +0.41 ₽
♡	ООО «Сетевые технологии» – Денежное требование №2 ООО «Сетевые технологии»	17.00 ₽ +12.33 ₽
⋮	ООО «Сетевые технологии» –	13.00 ₽
📈	Ибранное	Диверсификации
⚙️	Настройки	

Рис. 9: Страница активов (AssetsPage)

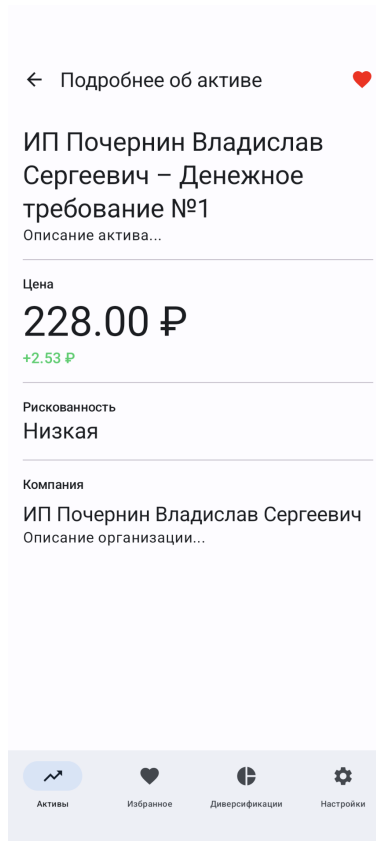


Рис. 10: Страница конкретного актива (AssetPage)

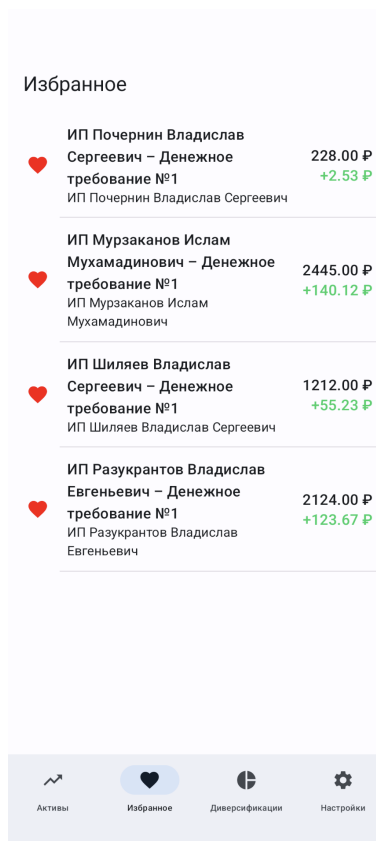


Рис. 11: Страница избранных активов (FavouritesPage)

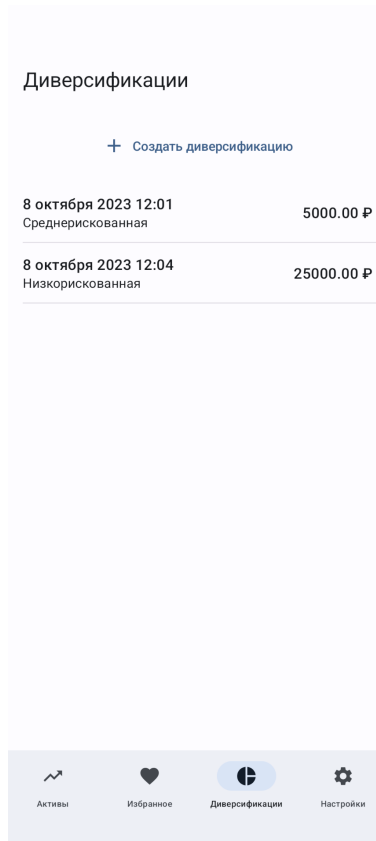


Рис. 12: Страница диверсификаций (DiversificationsPage)

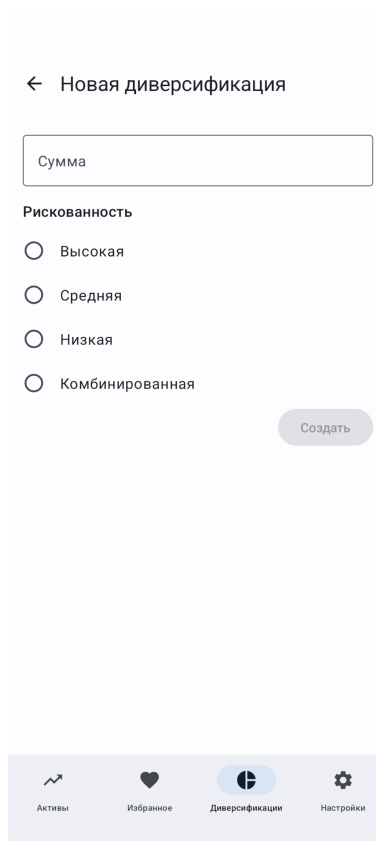


Рис. 13: Страница создания диверсификации (DiversificationCreatePage)

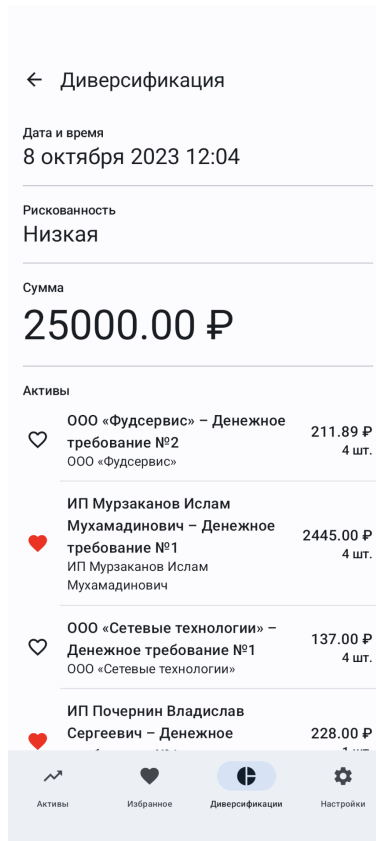


Рис. 14: Страница конкретной диверсификации (DiversificationPage)

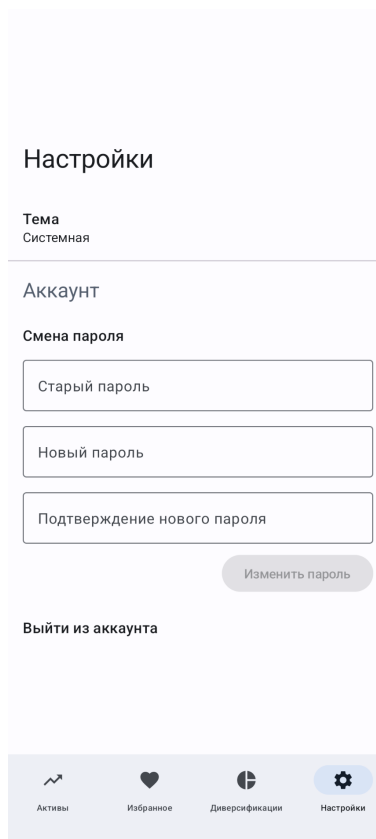


Рис. 15: Страница настроек (SettingsPage)