

Белорусский Государственный Университет
Факультет радиофизики и компьютерных технологий

Отчет по лабораторной работе №3

«Создание UML-диаграмм для графического описания ИС. Проектирование
модели данных»

Подготовили:

Студенты 4 курс 5ПИ,

Равгейша Алексей

Бабарико Виолетта

Петров Егор

Преподаватель:

Ломако А.А.

Минск, 2025

1. UML-диаграмма для описания информационной системы

1.1 Диаграмма сценариев использования (Use Case Diagram)

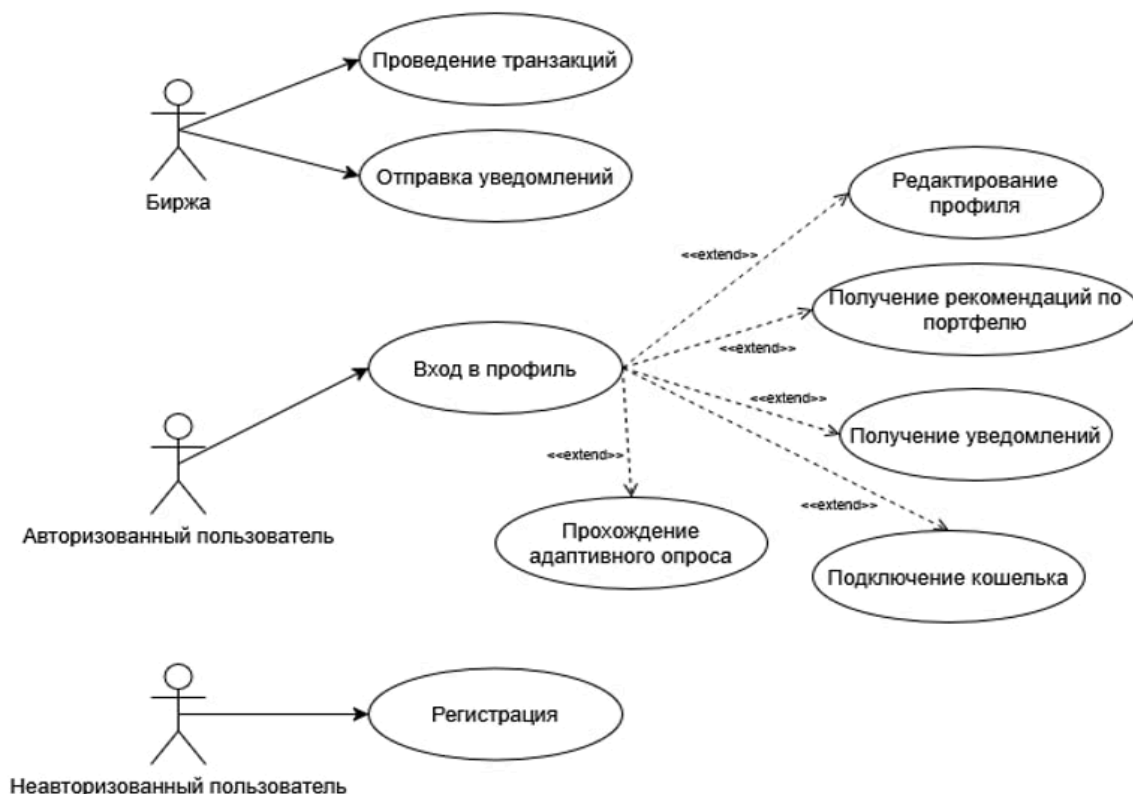


Рисунок 1. Диаграмма сценариев использования

На представленной диаграмме показана диаграмма вариантов использования (Use Case Diagram), описывающая взаимодействие различных участников с системой биржи. В системе участвуют три основных актёра: неавторизованный пользователь, авторизованный пользователь и биржа как отдельный системный компонент. Неавторизованный пользователь имеет доступ только к базовой функции — регистрации, которая позволяет создать учётную запись и получить статус авторизованного пользователя.

Авторизованный пользователь, войдя в систему, получает доступ к основным возможностям. Центральным действием для него является вход в профиль, который расширяется (через отношения extend) дополнительными возможностями: редактированием профиля, получением рекомендаций по портфелю, получением уведомлений, а также подключением кошелька. Кроме того, пользователь может пройти адаптивный опрос, что, вероятно, используется для персонализации рекомендаций или интерфейса.

Отдельно выделена роль биржи, которая выполняет системные функции: проведение транзакций и отправку уведомлений пользователям.

Эти процессы представляют внутреннюю работу системы и обеспечивают реализацию бизнес-логики обмена и информирования. Диаграмма описывает структуру взаимодействия между пользователями и системой биржи, показывая, какие действия доступны различным категориям пользователей и как эти действия связаны между собой.

1.2 Диаграмма компонентов (Component Diagram)

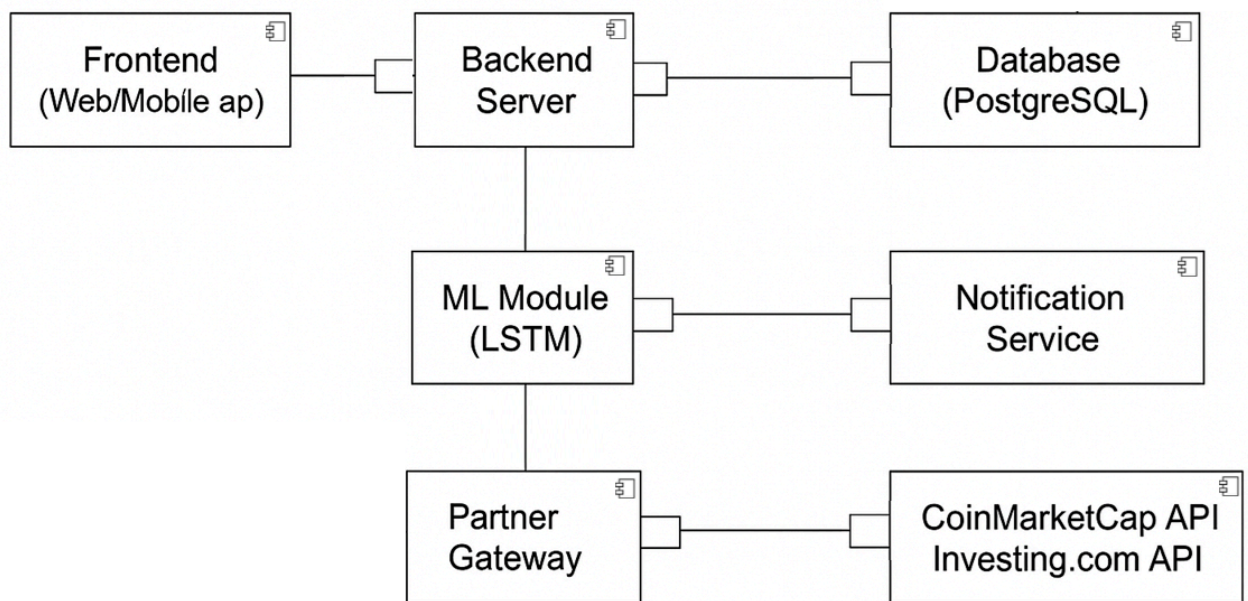


Рисунок 2. Диаграмма компонентов.

Диаграмма компонентов представляет архитектуру системы как набор взаимосвязанных модулей, где компонент "Frontend (Web/Mobile App)" обеспечивает взаимодействие с пользователем через UI/UX интерфейс, включая регистрацию, опрос и просмотр рекомендаций. Связи с "Backend Server" отражают передачу запросов для обработки, что позволяет интегрировать данные о предпочтениях и портфелях для коллаборативной фильтрации. Данная структура подчеркивает разделение обязанностей, обеспечивая масштабируемость и удобство для трейдеров.

Компонент "Backend Server" выступает в роли центрального хаба, координируя доступ к "Database (PostgreSQL)" для хранения анонимизированных профилей и портфелей, а также к "ML Module" для применения LSTM-моделей в прогнозировании доходности с учетом волатильности. Связи с "External Integrations" обеспечивают реал-тайм данные из CoinMarketCap и Investing.com, что критично для актуальности рекомендаций и уведомлений. Монетизация интегрируется через "Partner Gateway", где транзакции обрабатываются с комиссиями, минимизируя

регуляторные риски. Наконец, "Notification Service" связан с фронтендом для доставки push-уведомлений и Telegram-ботов, триггерируемых рыночными событиями или действиями похожих трейдеров. Общая модель подтверждает ориентацию на MVP, с потенциалом для масштабирования и обработки задержек API в будущем, обеспечивая приватность и производительность системы.

1.3 Диаграмма активностей (Activity Diagram)

Диаграмма активностей моделирует бизнес-процесс анализа портфеля и генерации рекомендаций, начиная с инициации пользователем, что отражает ключевой этап взаимодействия в системе "Интеллектуальная система по рекомендациям инвестиционных портфелей". Начальный узел "Start" переходит к активности "User Initiates Portfolio Analysis", подчеркивая активную роль пользователя в запуске процесса. Разветвление на "Connect to Wallet or Import Account Data" и "Fetch Market Data from External API" демонстрирует параллельную обработку данных из внутренних источников (кошельки, счета) и внешних API (CoinMarketCap, Investing.com), что обеспечивает актуальность анализа. Средняя часть процесса, представленная активностью "Analyze Portfolio and User Preferences", объединяет данные о портфеле и предпочтениях, полученных через адаптивный опрос, для формирования основы рекомендаций. Последующая активность "Generate Recommendation with LSTM Model" использует машинное обучение для прогнозирования доходности с учетом волатильности, а условный узел "Recommendation Accepted?" позволяет адаптировать процесс в зависимости от выбора пользователя, возвращаясь к анализу при необходимости. Это подчеркивает гибкость системы в работе с разными уровнями опыта трейдеров.

Завершающая стадия, включающая "Notify User of Recommendation" и конечный узел "End", отражает доставку результатов пользователю через push-уведомления или Telegram-бот, завершая цикл. Диаграмма акцентирует внимание на образовательном аспекте, минимизируя риски через персонализированный подход, и подтверждает фокус на MVP с интеграцией реал-тайм данных и монетизацией через партнерские транзакции, как описано в исходном описании проекта.

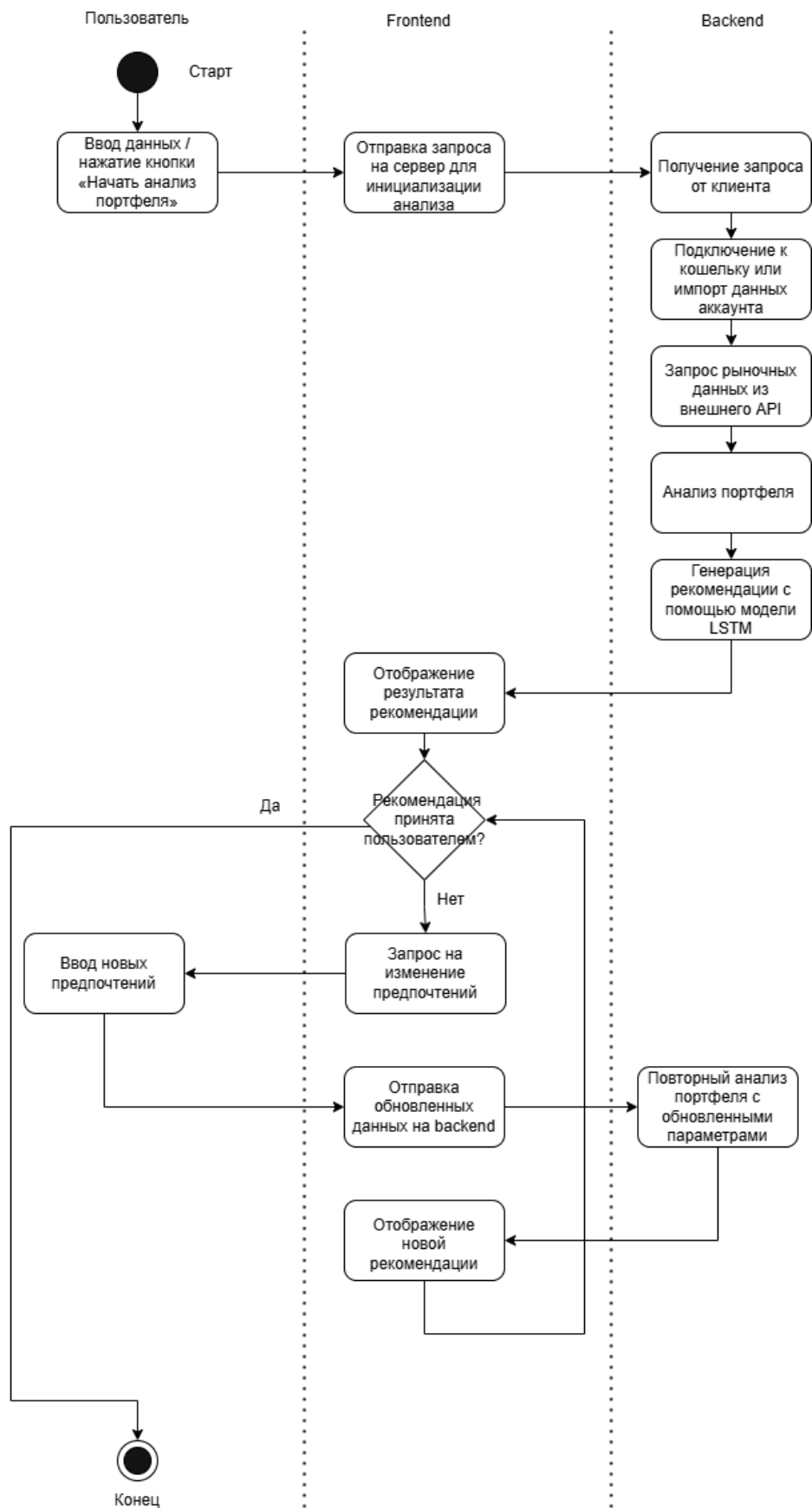


Рисунок 3. Диаграмма активностей.

1.4 Диаграмма последовательностей (Sequence Diagram)

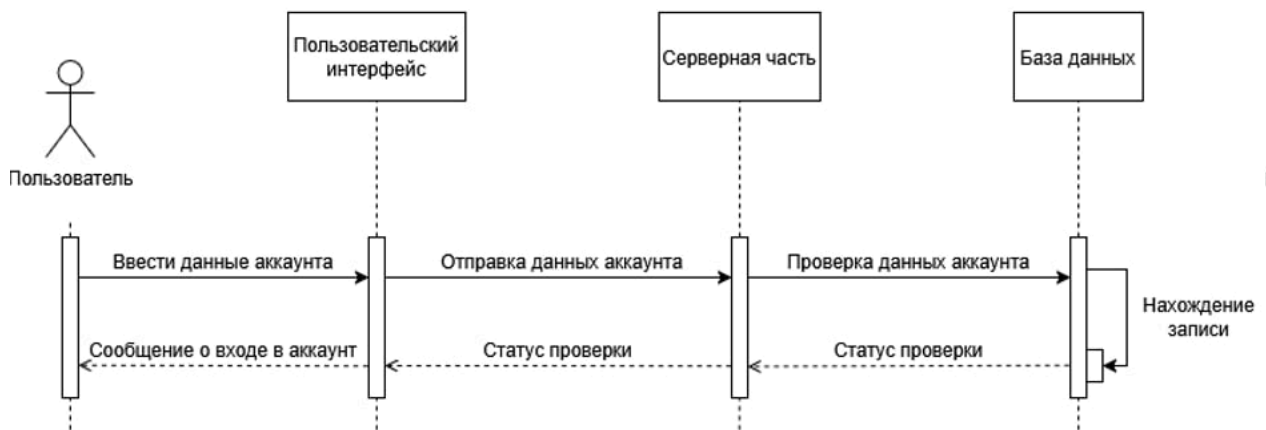


Рисунок 4. Диаграмма последовательностей.

На диаграмме последовательностей изображен процесс авторизации пользователя в системе. Всё начинается с того, что пользователь вводит свои данные — логин и пароль — через пользовательский интерфейс. После ввода этих данных интерфейс отправляет их на серверную часть системы для проверки. Серверная часть, получив данные аккаунта, инициирует обращение к базе данных. В базе данных выполняется поиск записи, соответствующей введенным пользователем данным. Если запись найдена, база данных возвращает на сервер статус проверки, указывающий, совпадают ли введенные данные с хранящимися.

После получения результата серверная часть формирует ответ о статусе проверки и передает его обратно пользовательскому интерфейсу. Интерфейс, в свою очередь, отображает пользователю сообщение о результате авторизации — успешный вход в систему или уведомление об ошибке, например, при неверных данных. Диаграмма показывает последовательное взаимодействие между пользователем, интерфейсом, сервером и базой данных при входе в аккаунт: от момента ввода данных до получения ответа о результате проверки.

2. Организация данных в информационной системе

Анализ организации данных в системе "Интеллектуальная система по рекомендациям инвестиционных портфелей" осуществляется с учетом обеспечения эффективного хранения, обработки и анализа структурированных данных о пользователях, их портфелях, предпочтениях, рекомендациях и внешних рыночных данных. Учитывая специфику проекта, включая интеграцию с внешними API (CoinMarketCap, Investing.com),

применение коллаборативной фильтрации на основе машинного обучения (LSTM-модели), соблюдение анонимизации данных в соответствии с GDPR-подобными стандартами и поддержку реал-тайм уведомлений, данные характеризуются высокой степенью структурированности и множественными связями между сущностями (пользователи, активы, рекомендации). Данный подход предполагает использование реляционной модели данных (RDBMS), например PostgreSQL или MySQL, обеспечивающих поддержку ACID-транзакций, сложных запросов (JOIN для фильтрации по портфелям) и масштабируемости. Альтернативные решения, такие как NoSQL (MongoDB для неструктурированных новостей), могут рассматриваться для хранения исторических данных о ценах или логов уведомлений, однако основа модели базируется на реляционной архитектуре для обеспечения целостности (foreign keys для связей портфель-активы) и производительности запросов машинного обучения. Проведенный анализ указывает на оптимальность гибридного подхода (SQL + NoSQL для аналитики), при этом для этапа MVP акцент делается на реляционной базе данных с индексацией часто запрашиваемых полей (например, риск-толерантность для генерации рекомендаций).

2.1 Диаграмма «сущность-связь» для создания концептуальной модели данных

Связи определяются как 1:N (User has many Portfolios), M:N (Portfolio contains many Assets), 1:N (User receives many Recommendations). Данная структура отражает концептуальную логику системы, где данные централизуются вокруг пользователя, портфеля и рекомендаций, с внешними интеграциями для актуальности рыночных данных.

Концептуальная модель данных подчеркивает центральную роль сущности "User" как отправной точки для всех процессов, включая регистрацию, анализ портфеля и получение рекомендаций. Связи, такие как "has" между пользователем и портфелем, отражают автоматическую оценку уровня вовлеченности по объему портфеля, обеспечивая адаптацию рекомендаций для различных категорий пользователей. Данная структура поддерживает интеграцию данных о предпочтениях, полученных через опрос, с реальными активами, минимизируя дублирование и обеспечивая анонимизацию данных.

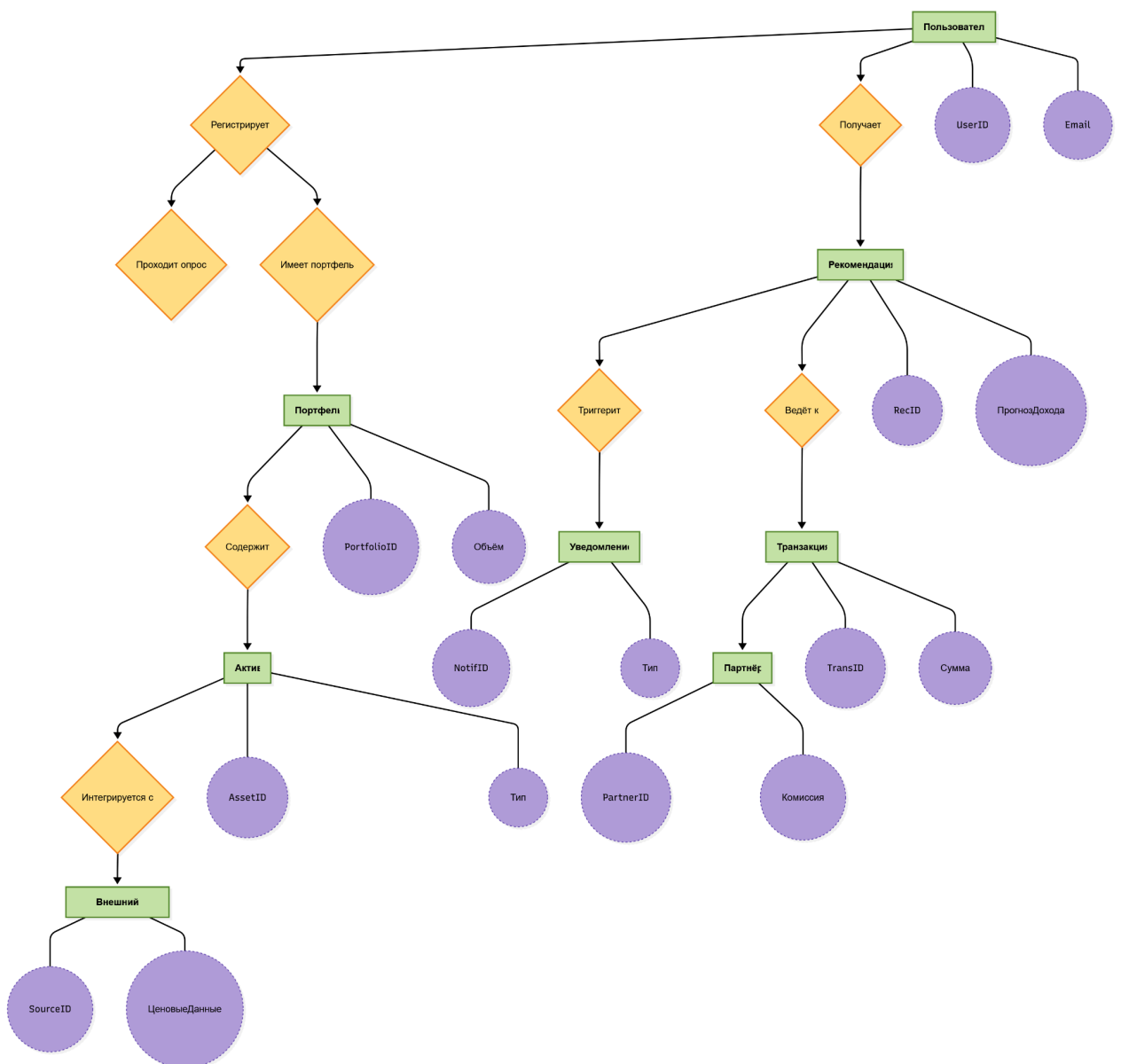


Рисунок 5. Диаграмма «сущность-связь» для создания концептуальной модели данных

Вторая ключевая связь "contains" между портфелем и активами позволяет моделировать диверсифицированные инвестиции в криптовалюты и акции с учетом внешних источников для реал-тайм данных, что способствует реализации коллаборативной фильтрации. Монетизационная модель интегрируется через связи "leads_to" и "processes", связывая рекомендации с транзакциями у партнеров без нарушения приватности. Уведомления и внешние интеграции ("triggers" и "fetched_from") добавляют динамику, обеспечивая реагирование системы на рыночные события в реальном времени. Общая структура подтверждает целесообразность использования реляционной базы данных для управления множественными отношениями и обеспечения масштабируемости.

2.2 Детализированная ER-диаграмма для создания логической модели данных

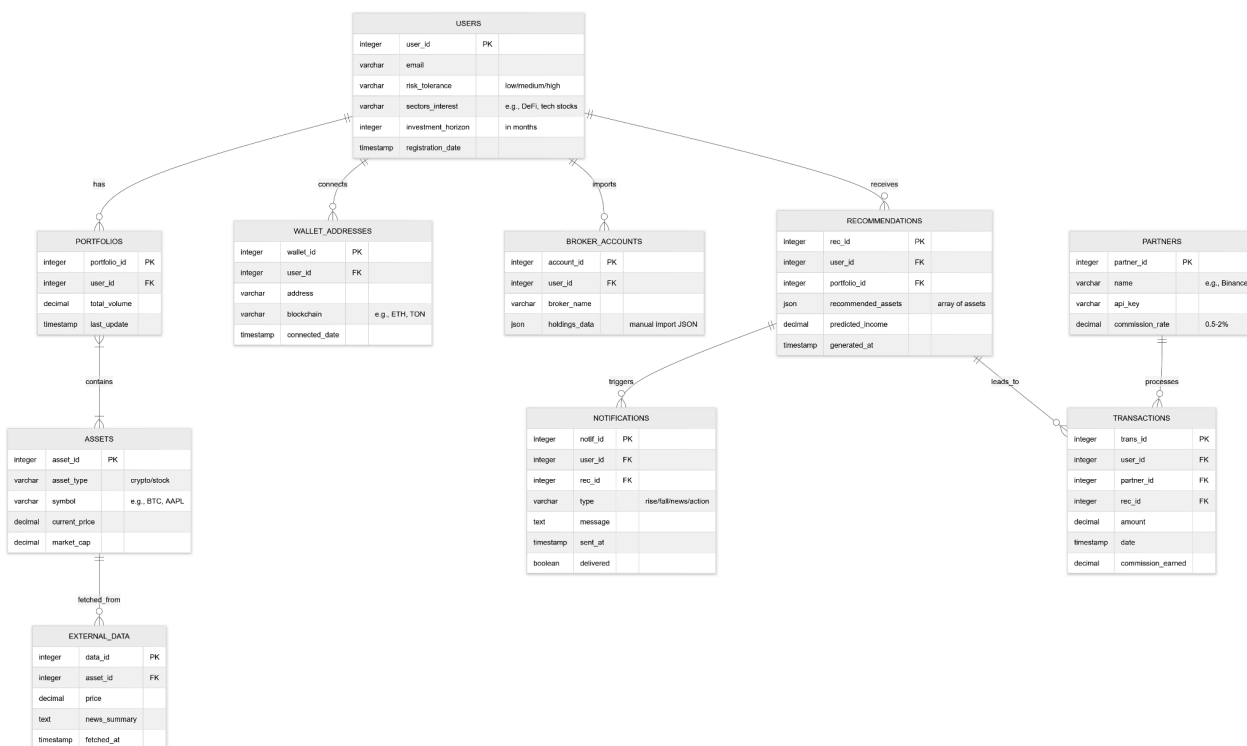


Рисунок 6. Детализированная ER-диаграмма для создания логической модели данных

Логическая модель данных детализирует структуру таблиц, где сущность "USERS" включает атрибуты профиля, такие как `risk_tolerance` и `sectors_interest`, полученные из адаптивного опроса, что обеспечивает основу для генерации персонализированных рекомендаций. Внешние ключи (FK) поддерживают связи с таблицами портфелей, кошельков и счетов, обеспечивая автоматическое считывание данных через публичные viewers и ручной импорт, с использованием `timestamp` для отслеживания обновлений, что гарантирует целостность данных и оптимизирует запросы для оценки уровня вовлеченности по атрибуту `total_volume`.

Таблицы "ASSETS" и "EXTERNAL_DATA" предназначены для хранения информации об активах и рыночных данных (цены, новости), интегрируемых из CoinMarketCap и Investing.com, с использованием `decimal` для точных финансовых расчетов. Связи типа M:N ("contains") моделируют сложные портфели, а поле `json` в "RECOMMENDATIONS" (рекомендованные активы) упрощает хранение массивов для реализации коллаборативной фильтрации и LSTM-прогнозов, включая атрибут `predicted_income` с учетом волатильности. Монетизационная модель реализована через таблицы "TRANSACTIONS" и "PARTNERS", где атрибут `commission_earned` рассчитывается на основе

реферальных ставок, с использованием внешних ключей для отслеживания происхождения рекомендаций.

Таблица "NOTIFICATIONS" с атрибутами типа и сообщения обеспечивает поддержку реал-тайм оповещений, с использованием boolean для подтверждения доставки (push или Telegram). Общая структура логической модели подтверждает применимость реляционной базы данных как основы, с нормализацией для повышения эффективности (избежание избыточности в данных об активах) и возможностью индексации по часто используемым полям (например, user_id для персонализации). Данная архитектура обеспечивает быстрые запросы на этапе MVP, с перспективой интеграции NoSQL для обработки больших объемов исторических данных.

2.3 Выводы о необходимой модели хранения данных

На основе разработанных нотаций концептуальная модель данных акцентирует внимание на центральной роли сущности пользователя и портфеля как основы для формирования рекомендаций, что требует применения реляционной базы данных для эффективного выполнения сложных JOIN-запросов (например, выборка пользователей с portfolio_volume \geq текущего для коллаборативной фильтрации). Логическая модель подтверждает данное решение, включая атрибуты для поддержки машинного обучения (риск, сектора) и интеграций (external_data для реал-тайм цен), с нормализацией до 3NF для предотвращения дублирования данных (отдельные таблицы для assets и external_data). Рекомендуется использование PostgreSQL: поддержка JSON для гибких данных (holdings), индексы для ускорения поиска по рискам, триггеры для автоматизации уведомлений. В дальнейшем целесообразно рассмотреть добавление NoSQL (Redis для кэширования рекомендаций) для обеспечения масштабируемости, однако на этапе MVP реляционная база данных обеспечит соблюдение требований приватности (анонимизация через views) и производительности (partitioning для исторических данных). Данный подход оптимален для финтех-приложений с акцентом на аналитику и монетизацию.

Заключение

Разработанные UML-диаграммы — диаграмма сценариев использования, диаграмма компонентов и диаграмма активностей — формируют взаимосвязанную систему, обеспечивающую комплексное моделирование и реализацию проекта "Интеллектуальная система по рекомендациям инвестиционных портфелей". Диаграмма сценариев использования определяет функциональные требования, отражая ключевые взаимодействия

актора "User (Trader)" с системой, включая регистрацию, анализ портфеля и выполнение транзакций, а также роли внешних акторов ("External API" и "Partner"). Это обеспечивает основу для определения бизнес-процессов, детализированных в диаграмме активностей, где процесс анализа портфеля и генерации рекомендаций демонстрирует последовательность шагов, начиная с инициации пользователем и заканчивая уведомлением, с учетом интеграции внешних данных и адаптации под предпочтения.

Диаграмма компонентов дополняет указанные модели, представляя архитектурную структуру системы, где "Frontend", "Backend", "Database" и "External Services" образуют модульную основу для реализации сценариев и процессов. Связи между компонентами, такие как "User Requests" и "Fetch Market Data", соответствуют шагам в диаграмме активностей (например, "Fetch Market Data from External API"), обеспечивая техническую реализацию функциональности, описанной в сценариях использования. Такая взаимосвязь подтверждает, что система эффективно поддерживает образовательный фокус и монетизацию через партнеров.

Интеграция диаграмм подчеркивает последовательный подход к разработке: диаграмма сценариев использования задает цели и взаимодействия, диаграмма активностей детализирует бизнес-логику (например, использование LSTM-моделей для рекомендаций), а диаграмма компонентов обеспечивает архитектурную основу для их реализации. Эта взаимосвязь оптимизирует проектирование MVP, обеспечивая масштабируемость, приватность данных (через анонимизацию в базе) и производительность, что соответствует требованиям финтех-приложения с акцентом на анализ и реал-тайм уведомления.

Исполнитель



Равгейша А. Д.

(подпись)

(Ф.И.О)

Исполнитель



Бабарико В. Д.

(подпись)

(Ф.И.О)

Исполнитель



Петров Е. А.

(подпись)

(Ф.И.О)