

Министерство науки и высшего образования Российской Федерации  
**Муромский институт (филиал)**  
федерального государственного бюджетного образовательного учреждения высшего образования  
**«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»**  
(МИ ВлГУ)

Факультет \_\_\_\_\_ ИТР \_\_\_\_\_

Кафедра \_\_\_\_\_ ИС \_\_\_\_\_

# ОТЧЕТ

по \_\_\_\_\_ преддипломной практике  
(наименование практики)

\_\_\_\_\_ ООО «РЕДСОФТ»  
(место прохождения практики)

Тема «Разработка интерактивного симулятора экономических моделей»

\_\_\_\_\_ (оценка) \_\_\_\_\_  
Руководитель  
Комкова С.В.  
(подпись) (от предприятия)

\_\_\_\_\_ (подпись) (от института)

Студент \_\_\_\_\_ ИС-121  
(группа)

\_\_\_\_\_ Есин Д.И.  
(фамилия, инициалы)

\_\_\_\_\_ (подпись) (дата)

# БЛАНК ЗАДАНИЯ

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 Общие сведения об объекте автоматизации.....	5
2 Программы аналоги .....	7
3 Определение требований к проектируемой информационной системе.....	9
4 Выбор средств реализации поставленной задачи .....	10
5 Проектная часть.....	13
5.1 Определение бизнес-процессов .....	14
5.2 Определение сценариев использования.....	16
5.3 Проектирование архитектуры программы .....	18
5.4 Проектирование взаимодействия модулей.....	19
5.5 Проектирование базы данных.....	22
5.6 Проектирование структуры классов сервера .....	23
5.7 Проектирование структуры классов клиента.....	26
ЗАКЛЮЧЕНИЕ .....	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	29

					МИВУ.09.03.02-04.000									
Изм	Лист	№ докум.	Подп.	Дата										
Студент	Есин Д.И.									Лит.	Лист	Листов		
Руков.	Комкова С.В.											3	29	
Конс.										МИВлГУ ИС-121				
Н.контр.														
Зав.каф.														

## ВВЕДЕНИЕ

В современных экономических условиях эффективность управления предприятиями существенно зависит от точности прогнозирования и корректности оценки экономических моделей, которые применяются для анализа и стратегического планирования. Сегодня одной из ключевых задач бизнеса является оптимизация процессов принятия решений на основе достоверного математического и имитационного моделирования экономических процессов. При этом ручной подход к разработке таких моделей и обработке больших массивов данных приводит к существенным затратам времени и ресурсов, а также увеличивает вероятность ошибок.

В связи с этим, актуальной задачей является разработка специализированного программного комплекса, обеспечивающего возможность создания и анализа экономических и имитационных моделей с применением современных технологий искусственного интеллекта. Использование таких систем позволяет автоматизировать трудоёмкие процессы обработки данных, минимизировать влияние человеческого фактора на результат и, как следствие, повысить точность и оперативность принятия управленческих решений.

Целью данной преддипломной практики является проектирование информационной системы – симулятора экономических моделей, обладающего расширенными возможностями моделирования и анализа данных с интеграцией внешних AI-сервисов. Разработка позволит пользователям создавать и настраивать экономические модели, проводить расчёты и симуляции, а также автоматически сравнивать полученные идеализированные результаты с реальным положением дел на основе загруженной документации и финансовой отчётности.

## 1 Общие сведения об объекте автоматизации

Объектом автоматизации в рамках рассматриваемого проекта является процесс построения и анализа экономических моделей на основе внутренних и внешних данных предприятия. Этот процесс представляет собой одну из ключевых задач в системе управления, планирования и прогнозирования деятельности организаций. Он охватывает как этапы стратегического анализа, так и оперативного управления, обеспечивая руководству возможность принимать обоснованные решения на основе математических и имитационных представлений реальной экономической среды.

Экономическое моделирование используется для отражения различных аспектов деятельности предприятий: финансовых потоков, производственных процессов, взаимодействия с рынком, управления издержками и доходами, оценки эффективности инвестиций, распределения ресурсов и других задач. В зависимости от цели анализа применяются разные типы моделей — от детерминированных математических расчётов до сложных имитационных схем, в которых учитываются неопределённости, поведение агентов и внешние влияния.

Как правило, построение экономических моделей требует сбора значительного объёма данных, их структурирования и приведения к виду, пригодному для последующего анализа. Источниками информации выступают бухгалтерская и управленческая отчетность, финансовые документы, статистика и рыночные исследования. Аналитик, работающий над моделью, должен не только корректно интерпретировать данные, но и построить модель, отражающую закономерности и специфические особенности объекта моделирования.

Значительную роль в этом процессе играет взаимодействие между различными источниками информации и форматами представления данных. Расхождения в структуре отчетности, различия в уровне детализации, наличие нетиповых форматов и необходимость приведения разнородных источников к единому виду требуют от аналитиков высокой квалификации и затрат времени. Кроме того, при использовании моделей на практике возникает потребность в их

					МИВУ.09.03.02-04.000	Лист
						5
Изм	Лист	№ докум.	Подп.	Дата		

постоянной актуализации, учёте новых факторов и переоценке исходных предпосылок, что затрудняет повторное применение моделей и ведёт к дублированию работы.

Дополнительной сложностью становится необходимость согласования моделей между различными участниками внутри организации. Каждое подразделение может иметь собственные приоритеты, методы расчёта и представление об исходных данных, что требует унификации подходов и прозрачности логики моделирования. Нередко между IT-специалистами, аналитиками и управленческим звеном возникают барьеры в интерпретации модели, что приводит к ошибкам, замедляет принятие решений и снижает эффективность внедрения аналитических инструментов в реальную практику управления.

Таким образом, объект автоматизации представляет собой комплексную деятельность по формированию экономических моделей, включающую работу с данными, построение и настройку моделей, проведение анализа и визуализацию результатов. Автоматизация этого процесса направлена на повышение точности, скорости, повторяемости моделирования и снижение зависимости от ручного труда, что делает её актуальной задачей для широкого круга организаций — от предприятий малого бизнеса до крупных корпораций, а также для научных и образовательных учреждений, занимающихся экономическим анализом.

					МИВУ.09.03.02-04.000	Лист
						6
Изм	Лист	№ докум.	Подп.	Дата		

## 2 Программы аналоги

При проектировании информационной системы важно учитывать уже существующие решения, которые выполняют схожие функции. Анализ аналогов позволяет выявить их сильные и слабые стороны, что поможет избежать ошибок при разработке собственной системы. Для оценки преимуществ и недостатков были выбраны три наиболее распространённых аналога: AnyLogic, Bizagi Modeler и Vensim.

AnyLogic – это многофункциональное программное решение для имитационного моделирования бизнес-процессов, социальных и экономических систем. Программа поддерживает различные подходы к моделированию (агентное, системно-динамическое, дискретно-событийное), а также имеет широкие возможности визуализации и анализа данных. Основным недостатком является высокая сложность изучения для неподготовленных пользователей и отсутствие встроенных возможностей для анализа документации с использованием технологий AI.

Bizagi Modeler – популярный инструмент моделирования бизнес-процессов, который позволяет проектировать и анализировать процессы с помощью наглядных BPMN-диаграмм. Программа отличается простотой освоения и хорошими возможностями для экспорта результатов. Недостатком Bizagi Modeler является ограниченность функционала исключительно моделированием бизнес-процессов, а также отсутствие встроенных модулей математических расчётов и интеграции AI.

Vensim – инструмент системно-динамического моделирования экономических и социальных процессов, позволяющий создавать и тестировать сложные математические модели. Программа обладает удобным интерфейсом для разработки моделей и проведения расчётов, а также мощными средствами анализа и визуализации результатов. Недостатком Vensim является отсутствие поддержки имитационного моделирования агентов и невозможность интеграции внешних AI-сервисов.

					МИВУ.09.03.02-04.000	Лист
						7
Изм	Лист	№ докум.	Подп.	Дата		

Таблица 1 - Сравнительный анализ программ аналогов

Программа	Математическое моделирование	Имитационное моделирование	Анализ финансовой отчетности	Интеграция AI-сервисов	Экспорт отчетности
AnyLogic	Да	Да	Нет	Нет	Да
Bizagi Modeler	Нет	Нет	Нет	Нет	Да
Vensim	Да	Нет	Нет	Нет	Да

Анализ существующих решений показал, что большинство программ-аналогов не поддерживают весь комплекс необходимых возможностей, включая интеграцию искусственного интеллекта и автоматический анализ финансовой отчетности. Эти недостатки требуют дополнительных затрат на интеграцию сторонних решений и значительно усложняют рабочий процесс. Таким образом, необходима разработка специализированной информационной системы, объединяющей математическое и имитационное моделирование, а также возможности AI-анализа документации и финансовых показателей в рамках единого программного решения.



### 3 Определение требований к проектируемой информационной системе

Проектируемая информационная система должна предоставлять пользователю возможность создавать, редактировать и управлять экономическими моделями, а также задавать и настраивать их параметры. Система должна позволять проводить математические расчёты и симуляции на основе созданных моделей, сохранять результаты, визуализировать их посредством графиков и диаграмм с возможностью настройки визуализации, а также экспортировать отчёты и графическую информацию.

Важной частью функциональных требований является интеграция внешних AI-сервисов, которые должны обеспечивать автоматический анализ загруженной пользователем документации и финансовой отчетности. Это позволит системе проводить сравнение идеализированных расчётов с реальными бизнес-показателями, повышая точность моделирования и прогнозирования. Кроме того, функционал AI должен обеспечивать генерацию рекомендаций и предложений по улучшению созданных моделей и агентов, а также автоматизировать формирование и настройку агентов для имитационного моделирования.

Система также должна обеспечивать необходимые механизмы защиты данных, авторизации и регистрации пользователей, сессионности, кеширования и параллельного выполнения вычислений. Это позволит обеспечить безопасность, производительность и удобство работы, а также создаст комфортную среду для экономистов, аналитиков и других специалистов, работающих с экономическими моделями и финансовым прогнозированием.

#### 4 Выбор средств реализации поставленной задачи

Для реализации серверной части разрабатываемой информационной системы предъявляются требования к языку программирования, фреймворку, реализации кеширования, а также к системе управления базами данных. Клиентская часть должна быть реализована в виде десктопного приложения с дружелюбным пользовательским интерфейсом.

Выбор языка программирования сервера:

Выбор стоит между Java и Python. Python отличается простой синтаксиса, скоростью прототипирования и большим количеством библиотек для работы с данными и AI. Однако Java превосходит Python по производительности, многопоточности и масштабируемости, особенно актуальным для серверных приложений с интенсивными вычислениями и параллельной обработкой данных. Кроме того, использование новейшей версии Java (jdk-21) позволяет использовать современные механизмы виртуальных потоков (Virtual Threads) и значительно оптимизировать работу приложения. Именно поэтому Java была выбрана для реализации серверной части системы.

Выбор серверного фреймворка:

Выбор стоит между Spring-framework и Jakarta EE. Jakarta EE предоставляет обширные возможности и является стандартом для разработки корпоративных приложений на Java, однако характеризуется более высоким порогом вхождения и сложностью конфигурации по сравнению с Spring Framework. Spring обладает простотой конфигурации, высоким уровнем интеграции с популярными технологиями и большим сообществом разработчиков, что ускоряет разработку, упрощает поддержку и дальнейшее развитие проекта. Эти преимущества стали решающими для выбора Spring Framework.

Выбор реализации кеширования:

Выбор стоял между Ehcache и Redis. Несмотря на простоту и лёгкость интеграции Ehcache, Redis был выбран как надёжное и масштабируемое решение. Redis предоставляет поддержку распределённого кеша, что делает его идеальным

					МИВУ.09.03.02-04.000	Лист
						10
Изм	Лист	№ докум.	Подп.	Дата		

для систем, предполагающих рост нагрузки и горизонтальное масштабирование. Его высокая производительность, устойчивость и активная поддержка сообщества обеспечивают долгосрочную надёжность и гибкость архитектуры. Учитывая потенциал развития проекта и стремление к отказоустойчивости, предпочтение отдано Redis.

Выбор базы данных:

PostgreSQL сравнивался с популярной реляционной базой данных MySQL. MySQL обладает хорошей производительностью, простотой использования и поддерживается большим количеством приложений и сервисов. Однако PostgreSQL предоставляет расширенные возможности работы с JSON, поддерживает сложные запросы и транзакции, которые крайне важны в контексте реализации экономических моделей, хранения расчётов и аналитических данных. Также PostgreSQL имеет более продвинутые механизмы обеспечения целостности и надёжности данных. Именно эти преимущества сделали PostgreSQL предпочтительным вариантом для хранения данных в системе.

Выбор языка программирования клиентской части:

Выбор стоит между Java и C#. Платформа .NET и язык C# обладают удобными инструментами и хорошей производительностью для разработки десктоп-приложений. Однако выбор Java обусловлен необходимостью использовать единый стек технологий на клиенте и сервере, что повышает простоту поддержки. Кроме того, JavaFX предоставляет мощные инструменты и библиотеки для реализации удобного и быстрого пользовательского интерфейса. Поэтому Java была выбрана и для реализации клиентского приложения.

Выбор технологии для клиентского интерфейса:

JavaFX сравнивался с альтернативным решением — Swing. Несмотря на то, что Swing — это зрелая и проверенная временем технология, JavaFX является более современным и функциональным решением, обеспечивающим широкие возможности для создания интерактивных и визуально привлекательных интерфейсов, включая встроенную поддержку графиков, диаграмм и анимаций.

Именно это послужило причиной выбора JavaFX для реализации интерфейса пользователя в клиентском приложении.

#### Выбор системы сборки:

Для реализации сборки и управления зависимостями серверного и клиентского приложений были рассмотрены две наиболее распространённые системы — Gradle и Maven. Maven является широко используемым инструментом, обладающим стандартной и понятной структурой проекта, простой конфигурацией и развитой экосистемой плагинов. Основное преимущество Maven заключается в унифицированном подходе к сборке и строгом соответствии стандартам, однако это же является его недостатком: любая нестандартная задача требует более сложной настройки. В свою очередь, Gradle представляет собой более современную и гибкую систему сборки. Gradle позволяет описывать процесс сборки проекта при помощи сценариев на языке Groovy или Kotlin, что значительно упрощает реализацию сложных и нестандартных задач. Также Gradle эффективнее управляет зависимостями, поддерживает инкрементальную сборку, параллельное выполнение задач и обладает более высокой скоростью сборки крупных проектов за счёт продвинутого механизма кеширования. Исходя из вышеперечисленного, а также учитывая необходимость гибкости, производительности и лёгкости дальнейшего сопровождения проекта, выбор сделан в пользу системы сборки Gradle.

Таким образом, выбранные технологии и архитектурные решения обеспечат надежность, удобство и безопасность работы информационной системы.

					МИВУ.09.03.02-04.000	Лист
						12
Изм	Лист	№ докум.	Подп.	Дата		

## 5 Проектная часть

При проектировании информационной системы, предназначенной для автоматизации экономического моделирования и анализа финансовых данных, необходимо соблюдать ряд общих требований и принципов, обеспечивающих надёжность, эффективность и простоту последующего сопровождения системы.

Важнейшим требованием является соблюдение модульного принципа построения системы. Компоненты должны разрабатываться максимально независимо друг от друга, что обеспечит высокую масштабируемость и лёгкость внесения изменений. Каждый компонент должен иметь чётко определённые функции и способы взаимодействия с другими компонентами, которые документируются на этапе проектирования. Это необходимо для снижения рисков при дальнейшем развитии и доработке системы.

Кроме того, проектирование должно предусматривать интеграцию с внешними AI-сервисами для расширения возможностей анализа данных и повышения точности моделей. Взаимодействие с внешними системами должно быть реализовано с использованием стандартных интерфейсов и протоколов (например, REST API), что позволит легко изменять или добавлять интеграции по мере необходимости.

Таким образом, соблюдение перечисленных требований обеспечит создание надёжной, эффективной и легко поддерживаемой информационной системы, полностью удовлетворяющей заявленным функциональным и техническим характеристикам.

					МИВУ.09.03.02-04.000	Лист
						13
Изм	Лист	№ докум.	Подп.	Дата		

## 5.1 Определение бизнес-процессов

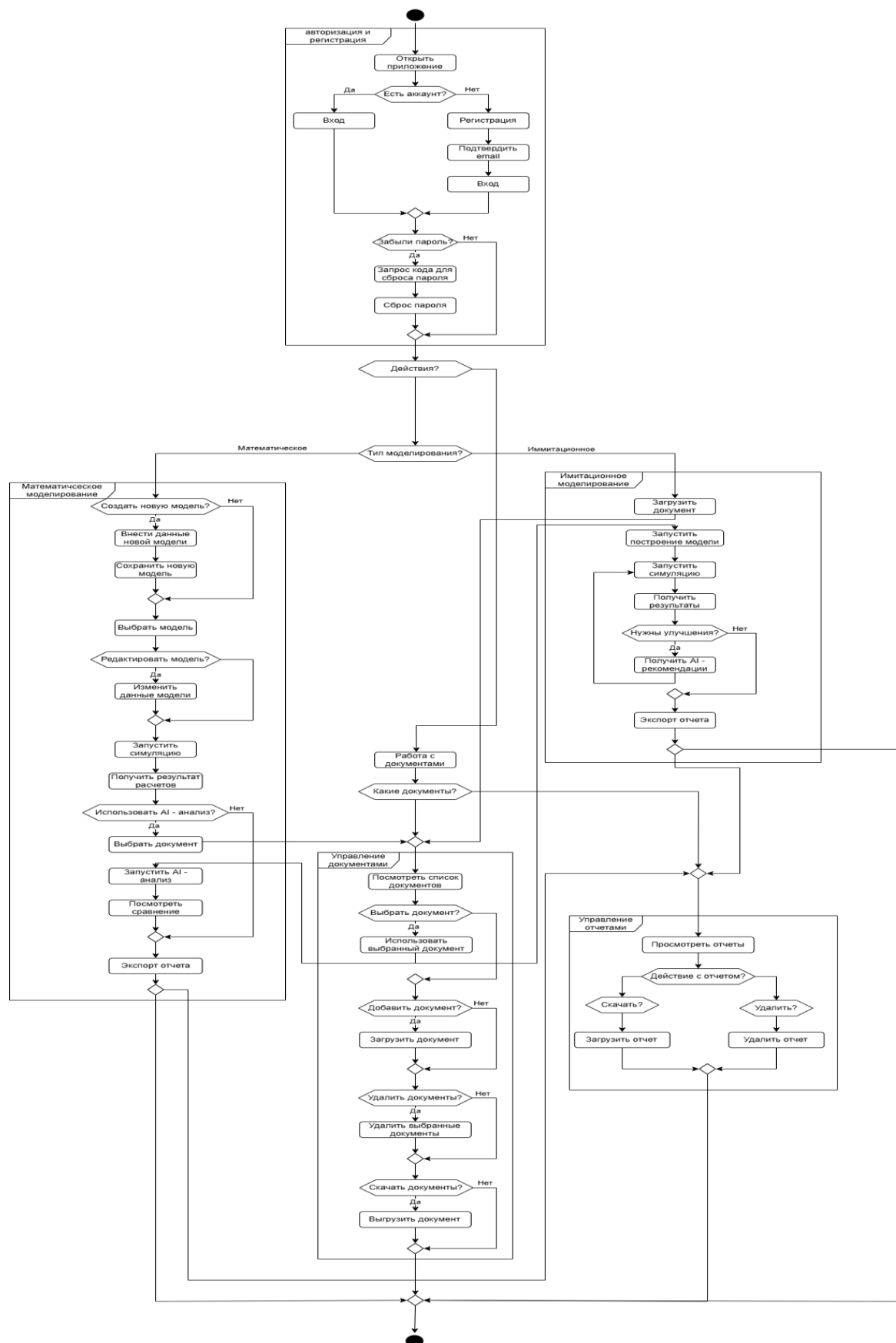


Рисунок 1 - Диаграмма бизнес-процессов

Диаграмма бизнес-процессов , представленная на рисунке 1, отражает логическую последовательность действий, принимаемые решения, а также возможные развилки, определяющие ход работы пользователя с приложением.

#### Авторизация и регистрация:

Пользователь запускает приложение и выбирает, есть ли у него аккаунт. При наличии аккаунта выполняется вход. В противном случае необходимо пройти процедуру регистрации с подтверждением электронной почты. Также предусмотрен механизм восстановления доступа через сброс пароля.

#### Выбор действий и типа моделирования:

После входа пользователь выбирает, какое действие он хочет выполнить — перейти к моделированию или работе с документами и отчетами. При выборе моделирования предлагается два типа: математическое или имитационное.

#### Математическое моделирование:

Пользователь может создать новую модель, внести данные и сохранить её. Далее доступен запуск симуляции и получение результатов. Система предлагает возможность AI-анализа, сравнение с документами и экспорт итогового отчета.

#### Имитационное моделирование:

Пользователь загружает документ, на основе которого строится модель. После запуска симуляции пользователь получает результаты и при необходимости, AI-рекомендации для улучшения. Итоговый отчет также может быть экспортирован.

#### Работа с документами:

В этом разделе реализовано управление документацией: просмотр, выбор, загрузка новых документов, удаление и выгрузка. Эти действия позволяют пользователю использовать внешние источники данных в моделировании.

#### Управление отчетами:

Пользователь просматривает ранее сформированные отчеты и может принять решение — скачать их или удалить. Это упрощает повторный доступ к уже выполненным анализам и моделированиям.

## 5.2 Определение сценариев использования

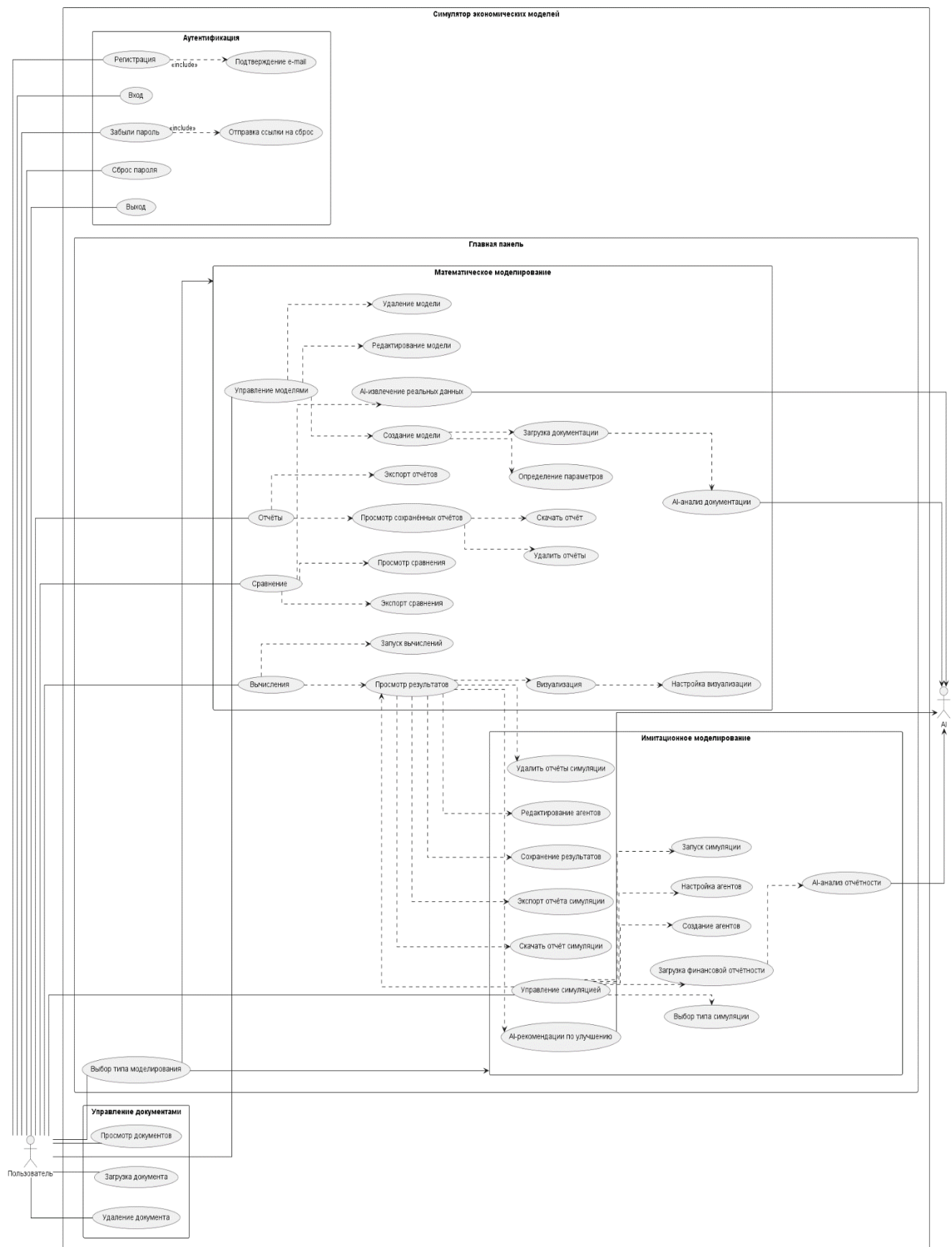


Рисунок 2 - Диаграмма сценариев использования



Диаграмма сценариев использования иллюстрирует взаимодействие пользователя с системой симулятора экономических моделей и охватывает все основные функции, доступные в рамках программного продукта. На диаграмме представлен единственный актер — пользователь, который инициирует действия, связанные с авторизацией, работой с моделями, документами, симуляциями и анализом результатов.

На начальном этапе пользователь проходит аутентификацию, включая регистрацию, подтверждение электронной почты, восстановление доступа и выход из системы. Эти действия обеспечивают безопасный и контролируемый доступ к функционалу приложения.

После авторизации пользователь попадает на главную панель, где выбирает тип моделирования: математическое или имитационное. В рамках математического моделирования пользователь может создавать, редактировать и удалять модели, а также управлять их параметрами. Поддерживается загрузка документации, экспорт и сравнение отчетов, запуск вычислений и настройка визуализации. Для расширенного анализа доступна интеграция с AI, позволяющая автоматически извлекать данные из документов и использовать их для повышения точности моделей.

Имитационное моделирование включает в себя работу с агентами и финансовой отчетностью. Пользователь может настраивать поведение агентов, запускать симуляции и анализировать полученные результаты. При необходимости система предлагает AI-рекомендации для улучшения моделей. Также предусмотрены действия по сохранению и экспорту результатов симуляций.

Дополнительно реализована система управления документами. Пользователь может загружать, просматривать и удалять документы, которые затем используются в процессе моделирования или анализа.

Диаграмма охватывает полный цикл работы пользователя в системе — от первого входа до получения аналитических отчетов, обеспечивая логичную структуру и удобство взаимодействия с каждым функциональным блоком.

### 5.3 Проектирование архитектуры программы

Для проектирования информационной системы был проведён анализ трёхуровневой и микросервисной архитектур. Несмотря на гибкость и масштабируемость микросервисов, они требуют сложной инфраструктуры и высокой трудоёмкости поддержки. Была выбрана трёхуровневая архитектура как оптимальный вариант, сочетающий простоту реализации с достаточной надёжностью, производительностью и возможностью расширения, благодаря чёткому разделению на клиентский, серверный и базовый уровни.

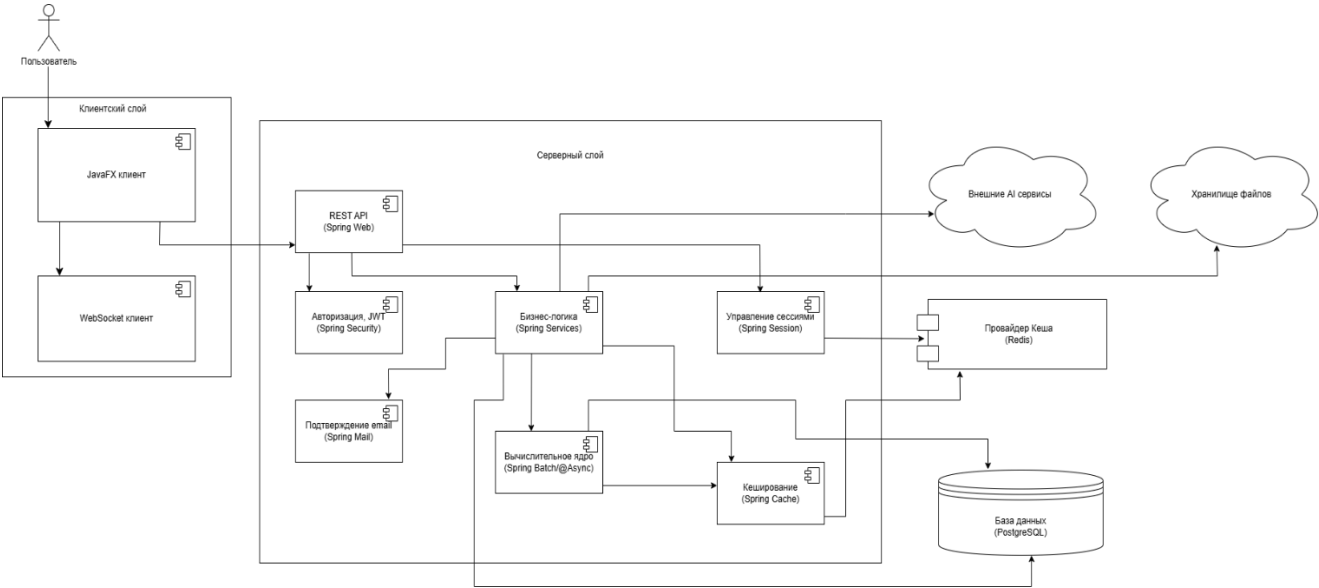


Рисунок 3 – Диаграмма архитектуры

Представленная на рисунке 3 диаграмма демонстрирует трёхуровневую архитектуру системы, построенной на базе Spring Framework. Клиентская часть реализована в виде JavaFX-приложения и WebSocket-клиента, обеспечивающих взаимодействие с пользователем и передачу данных на сервер. Серверный слой включает REST API, модули авторизации и защиты, управление сессиями, обработку email-подтверждений, бизнес-логику и вычислительное ядро. Для ускорения работы используются компоненты кэширования с поддержкой Redis. Хранение данных обеспечивается базой PostgreSQL, а интеграция с внешними AI-сервисами и файловым хранилищем осуществляется через сетевые интерфейсы.

# 5.4 Проектирование взаимодействия модулей

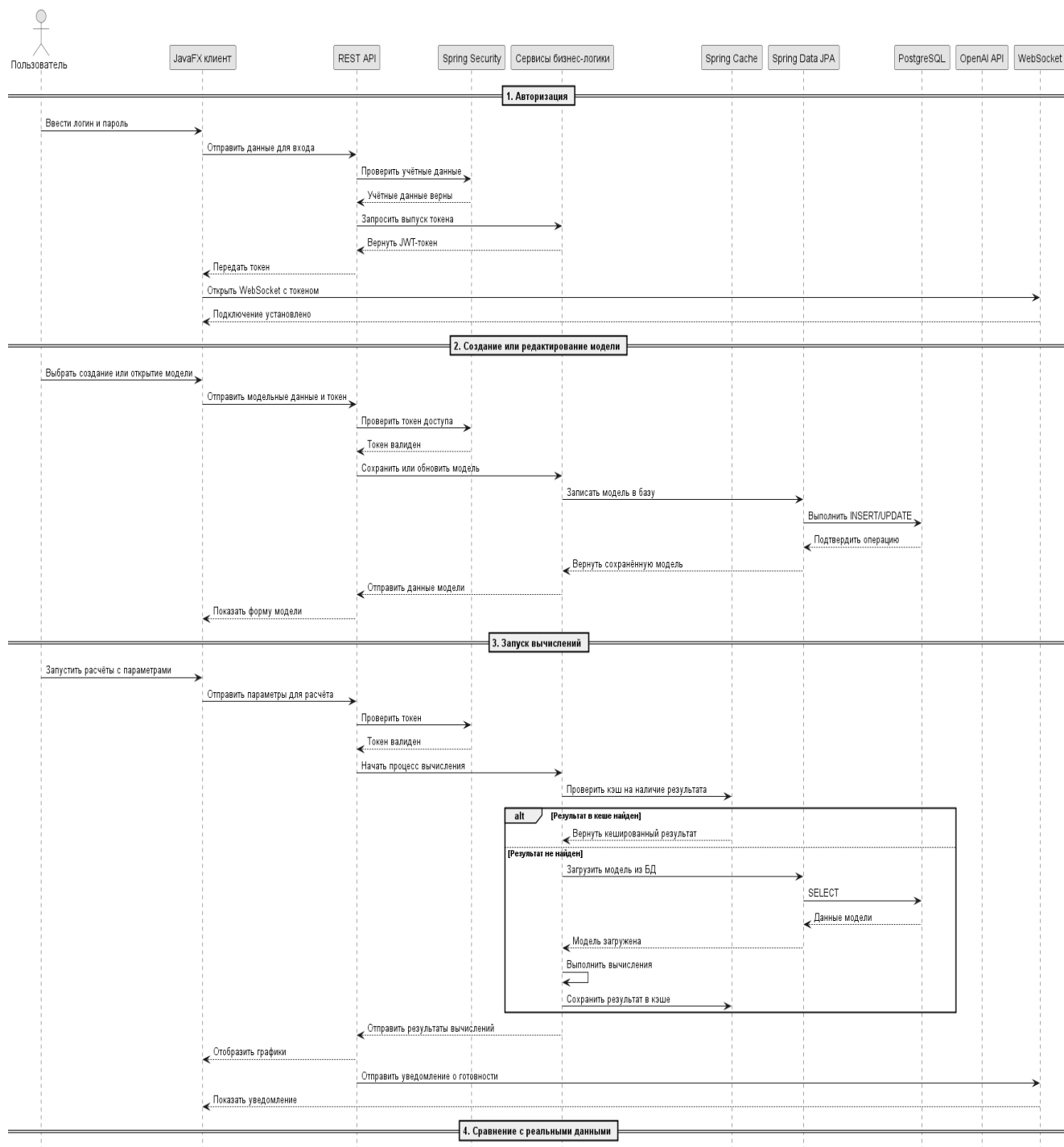


Рисунок 4 - Диаграмма последовательностей

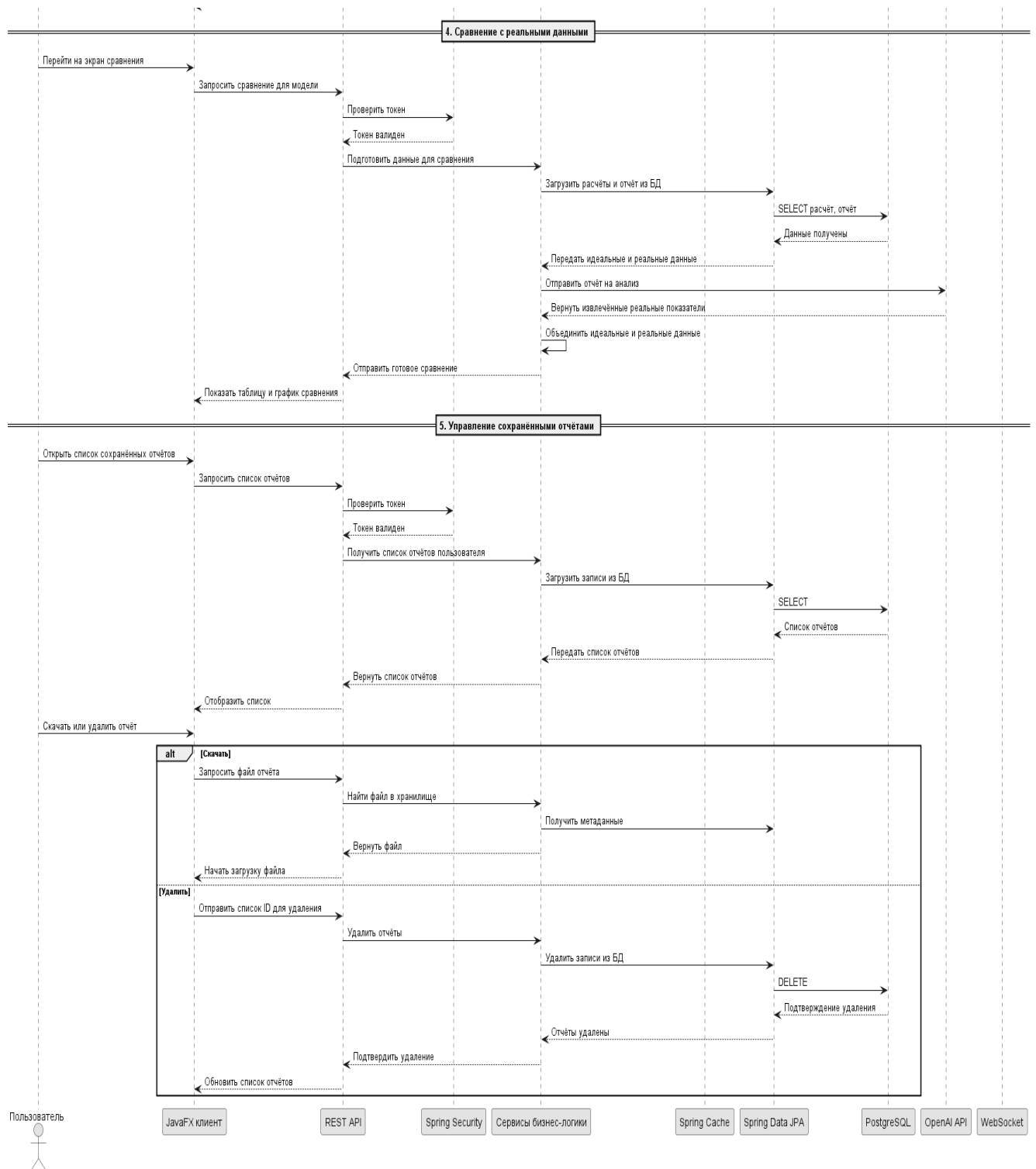


Рисунок 4 – Диаграмма последовательностей (продолжение)

На рисунке 4 показаны пять ключевых сценариев функционирования симулятора экономических моделей:

В сценарии авторизации пользователь вводит логин и пароль, после чего данные проходят валидацию с использованием Spring Security. При успешной

проверке сервер формирует JWT-токен, возвращает его клиенту и устанавливает WebSocket-соединение для последующего обмена данными в реальном времени.

Создание или редактирование экономической модели начинается с выбора шаблона или загрузки новой модели. Пользователь заполняет форму, отправляет данные через REST API, а сервер проверяет права доступа и сохраняет модель в базе данных PostgreSQL. После успешного сохранения модель возвращается клиенту для дальнейшей работы.

При запуске вычислений модель с заданными параметрами отправляется на сервер, где начинается процесс расчётов. Система проверяет наличие результата в кеше (Spring Cache). Если вычислений ещё не было, данные загружаются из базы, модель рассчитывается, и результат сохраняется как в БД, так и в кеше. Затем результат отправляется клиенту через WebSocket для отображения.

Сравнение с реальными данными осуществляется путём запроса сохранённых расчётов, которые дополняются актуальными показателями из базы данных. Пользователь получает таблицу и график сравнения, формируемые на основе объединённых данных модели и реальности.

В управлении сохранёнными отчётами пользователь может просмотреть, загрузить или удалить отчёт. Клиент запрашивает список доступных записей, которые извлекаются из базы данных. При скачивании отчёта выполняется выгрузка файла, при удалении — очистка метаданных и удаление записи из базы, после чего клиент обновляет список без перезагрузки интерфейса.

Диаграмма охватывает полный жизненный цикл работы пользователя — от авторизации до анализа моделей и управления результатами, подчёркивая взаимодействие между JavaFX-клиентом, REST API, сервисами Spring, базой PostgreSQL, кешем Redis и внешними API.



## 5.6 Проектирование структуры классов сервера

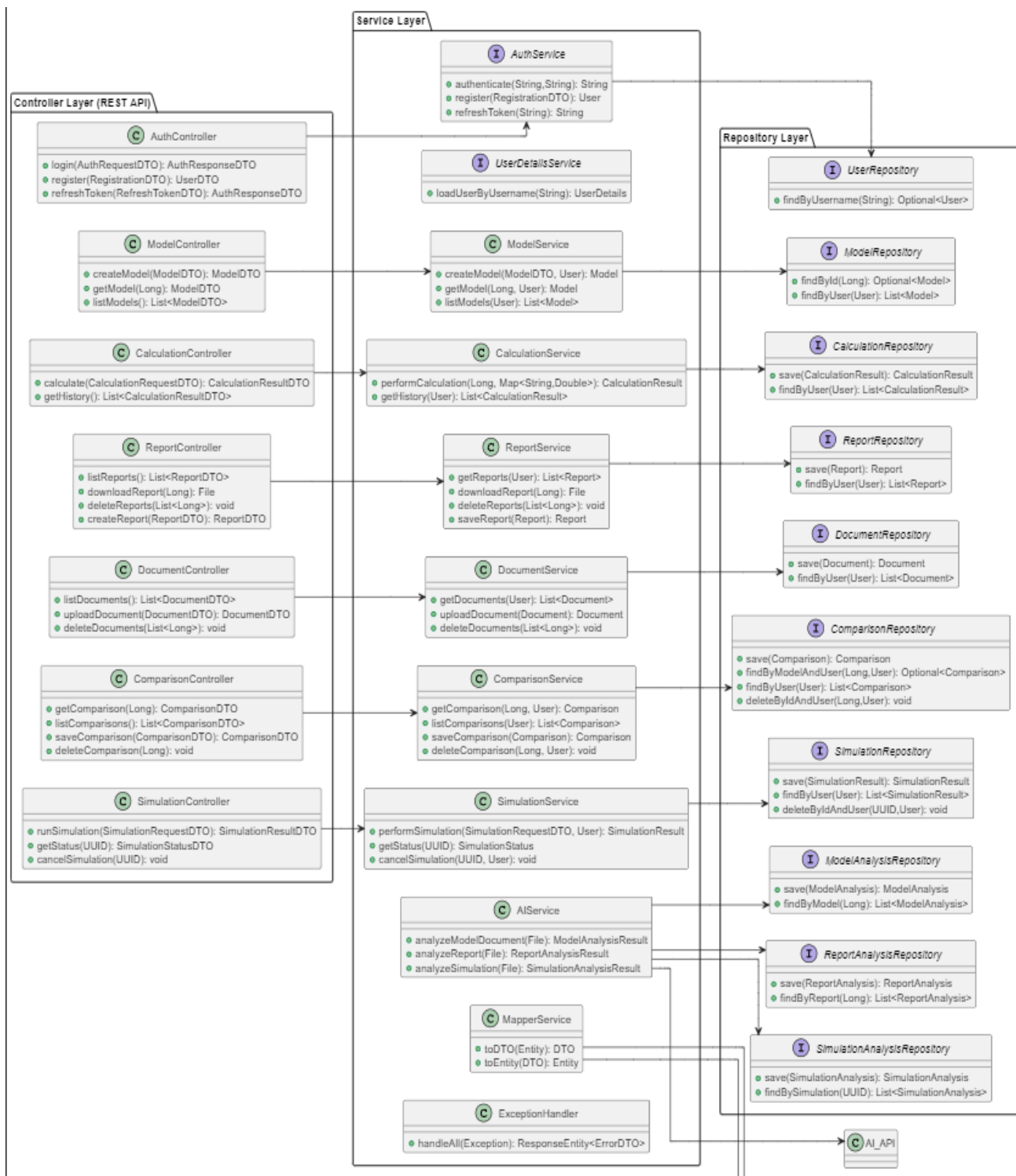


Рисунок 6 - Диаграмма классов сервера

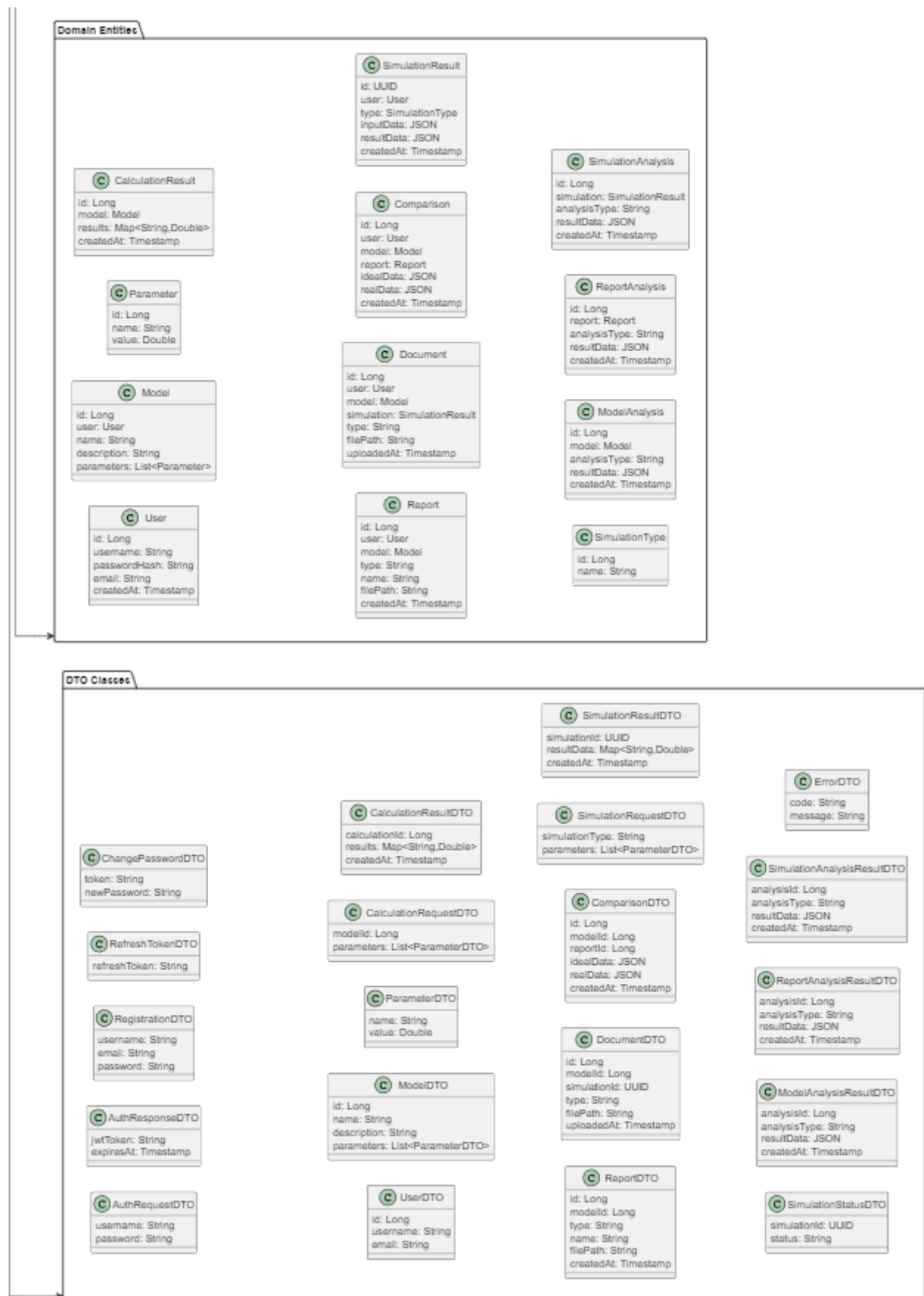


Рисунок 6 - Диаграмма классов сервера (продолжение)



Диаграмма классов, представленная на рисунке 6, серверная часть информационной системы реализована в соответствии с трёхуровневой архитектурой, включающей слой контроллеров (Controller Layer), бизнес-логики (Service Layer) и доступа к данным (Data Access Layer). Контроллеры принимают REST-запросы от клиентского приложения, выполняют базовую обработку входных данных и передают управление на сервисный уровень. Для обмена данными между клиентом и сервером используются DTO-классы, которые инкапсулируют только необходимые поля, обеспечивая безопасность и изоляцию внутренних структур приложения.

Сервисный слой содержит основную бизнес-логику приложения: управление моделями, симуляциями, отчетами, взаимодействие с внешними API-сервисами и запуск вычислений. Этот слой координирует работу между контроллерами и репозиториями, а также управляет кэшированием и асинхронными задачами. Сервисы используют объекты доменной области и обрабатывают их согласно бизнес-правилам, что делает код централизованным и переиспользуемым.

Слой доступа к данным реализован с помощью Spring Data JPA и работает с доменными сущностями, соответствующими структуре реляционной базы данных. Репозитории обеспечивают выполнение CRUD-операций и позволяют гибко взаимодействовать с данными без необходимости писать SQL-запросы вручную. Благодаря чёткому разграничению ответственности между слоями, архитектура остаётся масштабируемой, поддерживаемой и легко расширяемой.

## 5.7 Проектирование структуры классов клиента

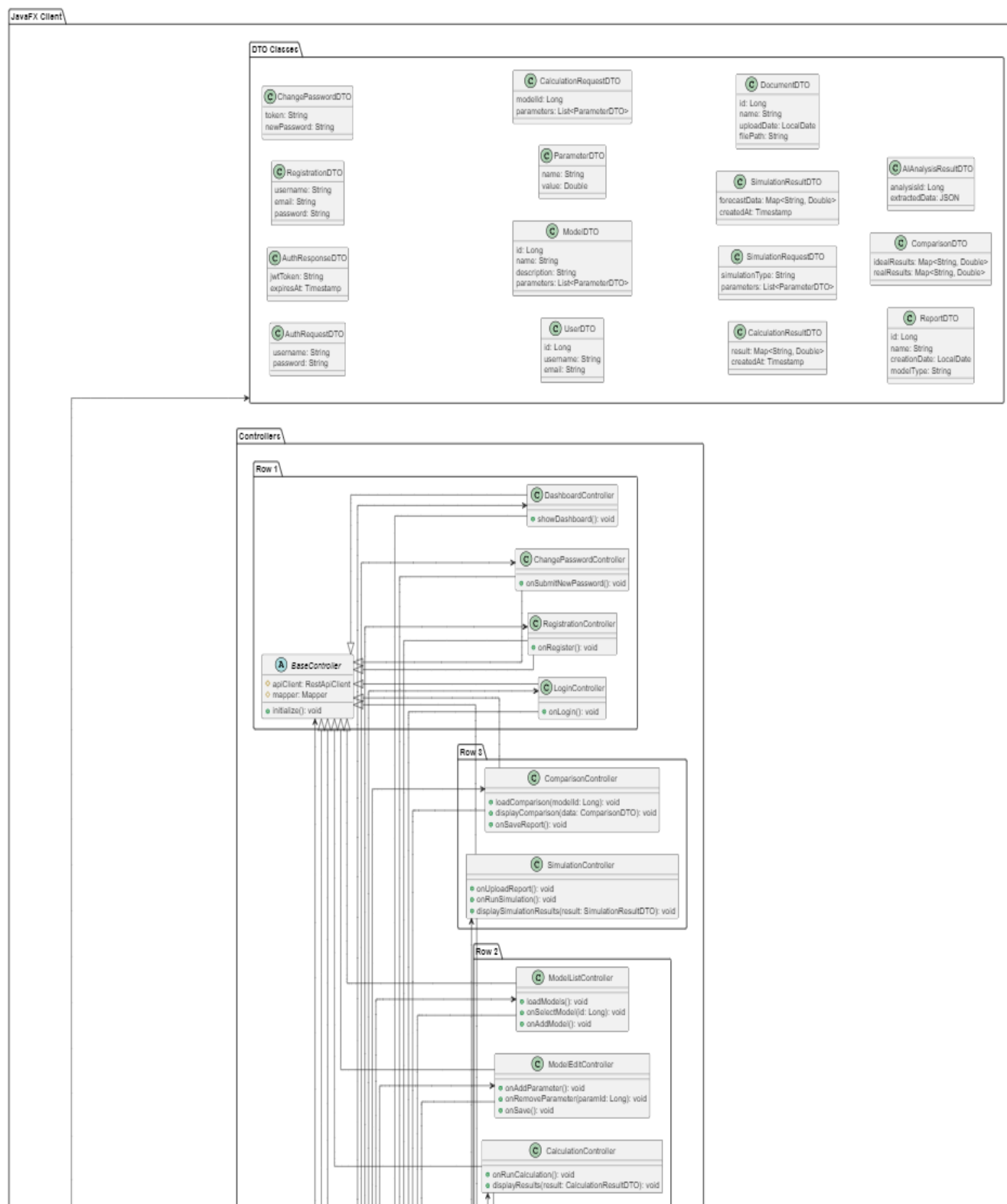


Рисунок 7 – Диаграмма классов клиента

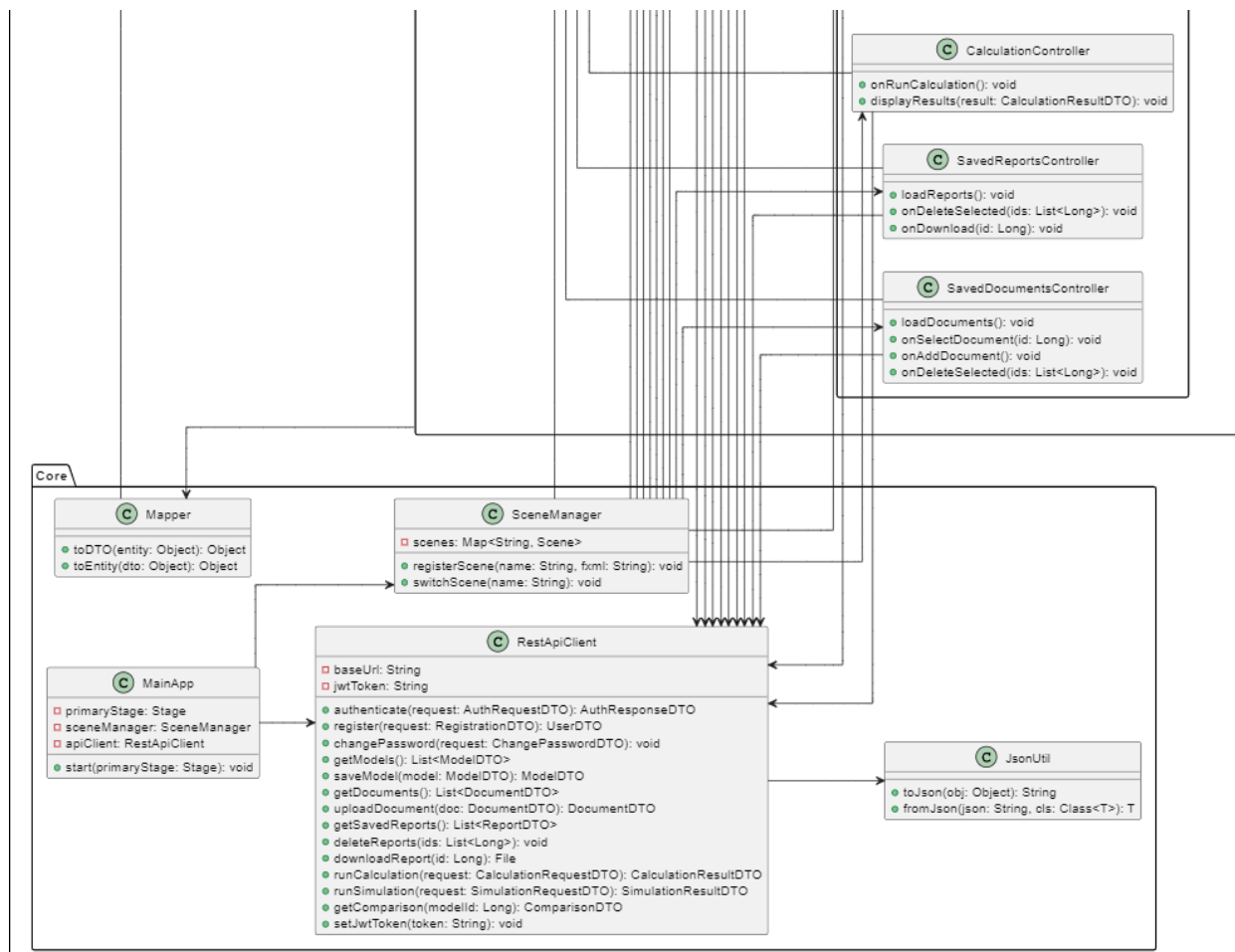


Рисунок 7 – Диаграмма классов клиента (продолжение)

Диаграмма, представленная на рисунке 7, построена с опорой на архитектурные принципы разделения ответственности. Представлены контроллеры JavaFX-интерфейса, компоненты навигации, классы для обработки пользовательских действий и классы, обеспечивающие взаимодействие с сервером через REST-запросы и WebSocket-соединения. Каждый экран приложения представлен отдельным контроллером, который управляет отображением, реакцией на действия пользователя и обращением к сервисным компонентам.

Классы API-клиента обрабатывают сетевые запросы и ответы, преобразуют данные из JSON в объекты и наоборот, используя DTO-структуры, синхронизированные с серверной частью. Также реализованы вспомогательные классы для управления сессиями и валидации пользовательского ввода. Такая структура обеспечивает модульность, упрощает поддержку интерфейса и позволяет легко добавлять новые экраны и сценарии взаимодействия.

## ЗАКЛЮЧЕНИЕ

В ходе прохождения преддипломной практики на базе ООО «РедСофт» была выполнена масштабная работа по проектированию информационной системы, предназначенной для автоматизации процессов построения и анализа экономических моделей. Основное внимание было уделено исследованию предметной области, формализации требований к разрабатываемому программному продукту и выбору оптимальных архитектурных и технологических решений, позволяющих обеспечить надёжность, масштабируемость и эффективность будущей системы.

Были изучены и проанализированы существующие программные аналоги, выявлены их сильные и слабые стороны, что позволило обосновать необходимость разработки нового программного решения с интеграцией AI-сервисов и поддержкой как математического, так и имитационного моделирования. В рамках проектной части были построены диаграммы бизнес-процессов, сценариев использования, архитектуры, взаимодействия компонентов, структуры базы данных и классов клиентской и серверной части, что заложило прочную основу для последующей реализации проекта.

Практика способствовала углублению профессиональных знаний в области проектирования информационных систем, взаимодействия с базами данных, работы с Java, Spring Framework и JavaFX, а также позволила на практике применить навыки анализа требований и построения архитектурных решений. Полученные результаты будут использованы при написании и защите выпускной квалификационной работы, в рамках которой будет завершена разработка функционального программного продукта. Прохождение практики подтвердило актуальность выбранной темы и продемонстрировало высокий уровень подготовки к реализации реальных ИТ-проектов.

					МИВУ.09.03.02-04.000	Лист
						28
Изм	Лист	№ докум.	Подп.	Дата		

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Oracle Java SE Documentation URL: <https://docs.oracle.com/en/java/javase>.
- 2 Open JFX Documentation – URL: <https://open.jvix.io>.
- 3 Spring Framework Documentation – URL: <https://spring.io/projects/spring-framework>.
- 4 Мэнкив, Н. Г. Принципы экономики / Н. Г. Мэнкив. — М.: ДМК Пресс, 2020. — 592 с.
- 5 Самуэльсон, П. А., Нордхаус, У. Д. Экономика / П. А. Самуэльсон, У. Д. Нордхаус. — М.: ДМК Пресс, 2020. — 784 с.
- 6 Гарипов, Р. Р. Spring Framework: разработка корпоративных приложений на Java / Р. Р. Гарипов. — М.: ДМК Пресс, 2020. — 352 с.
- 7 Гончаров, А. Ю. JavaFX: разработка клиентских приложений на Java / А. Ю. Гончаров. — М. : Лань, 2019. — 288 с.
- 8 Лаура, Л. PostgreSQL: основные понятия и практическое применение / Л. Лаура. — СПб.: БХВ-Петербург, 2020. — 416 с.