Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Инженерно-экономический факультет

Кафедра экономической информатики

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе по курсу:

«Программирование сетевых приложений»

на тему:

**«**Разработка системы прогнозирования платежеспособности предприятия на основе модели Фулмера**»**

Студент Купреева С. И.

группа 872302

Руководитель Ассистент кафедры

ЭИ, Мухин Ю. С.

Минск 2020

СОДЕРЖАНИЕ

[Введение 3](#_Toc58864066)

[1 Описание процесса прогнозирования платежеспособности предприятия 4](#_Toc58864067)

[1.1 Общие понятия и термины 4](#_Toc58864068)

[1.2 Анализ модели прогнозирования банкротства Д. Фулмера 6](#_Toc58864069)

[1.3 Определение проблемы, требующей автоматизации 7](#_Toc58864070)

[2 Постановка задачи на разработку программного средства 9](#_Toc58864071)

[3 Функциональное моделирование на основе стандарта IDEF0 12](#_Toc58864072)

[4 Информационная модель системы и ее описание 19](#_Toc58864073)

[5 Спецификация вариантов использования системы прогнозирования платежеспособности предприятия 21](#_Toc58864074)

[6 Модели представления системы прогнозирования платежеспособности предприятия и их описание 23](#_Toc58864075)

[7 Описание применения паттернов](#_Toc58864076) [проектирования 25](#_Toc58864077)

[8 Описание алгоритма реализующего бизнес-логику серверной части системы прогнозирования платежеспособности предприятия методом фулмера и алгоритма обработки запросов клиентов на примере авторизации 29](#_Toc58864078)

[9 Руководство пользователя по развертыванию системы прогнозирования платежеспособности предприятия методом Фулмера 32](#_Toc58864079)

[10 Результаты тестирования системы прогнозирования платежеспособности предприятия методом Фулмера 33](#_Toc58864080)

[Заключение 43](#_Toc58864081)

[Список использованных источников 44](#_Toc58864082)

[Приложение А](#_Toc58864083) [(обязательное)](#_Toc58864084) [Диаграммы развертывания, компонентов, последовательности состояний и классов (к разделу 6) 45](#_Toc58864085)

[Приложение Б](#_Toc58864086) [(обязательное)](#_Toc58864087) [Скрипт генерации базы данных (к разделу 9) 51](#_Toc58864088)

[Приложение В](#_Toc58864089) [(обязательное)](#_Toc58864090) [Листинг алгоритмов, реализующих бизнес-логику 53](#_Toc58864091)

[Приложение Г](#_Toc58864092) [(обязательное)](#_Toc58864093) [Листинг основных элементов программы 58](#_Toc58864094)

# ВВЕДЕНИЕ

В экономической системе современного общества огромную роль играет сфера бизнеса, представленная предприятиями. Любое предприятие является юридическим лицом, имеет законченную систему учета и отчетности, самостоятельный бухгалтерский баланс, расчетный и другие счета, печать с собственным наименованием и товарный знак (марку) [1]. Предприятие является основным звеном всей экономики, поскольку именно на этом уровне создается нужная обществу продукция, оказываются необходимые услуги. От финансового положения предприятий зависит экономика государства в целом.

В условиях рыночной экономики первостепенное значение имеет то, насколько эффективно устроена производственно-хозяйственной деятельность предприятия. Показатели эффективности при этом имеют первостепенное значение. Фактором, который влияет на эффективность является планирование. Планирование, расчет и анализ основных экономических показателей деятельности имеет очень большое значение [2]. Это позволяет понять перспективы изменения и развития предприятия, дать оценку работы, найти методы снижения затрат на ресурсы.

Эффективность работы предприятия напрямую зависит от анализа управления и работы предприятия, рассмотренных со всех сторон деятельности. А сделать это без расчетов экономических показателей представляется затруднительной задачей.

На практике при прогнозировании платежеспособности предприятия чаще всего применяют комбинированные методы, отнимающие намного больше времени. Каждый из методов является весьма трудоемким и сложным как математически, так и экономически, вследствие чего ошибки, вызванные человеческим фактором неизбежны.

Целью выполняемого курсового проекта является сокращение числа ошибок, вызванных человеческим фактором, при прогнозировании платежеспособности предприятия, сокращение количества рабочего времени и в последствии трудового ресурса на предприятии.

Задачами данного курсового проекта являются:

* изучить предметную область;
* разработать базу данных;
* реализовать клиент-серверное приложение;
* создать графический интерфейс приложения для пользователя;
* предоставить возможность просмотра информации в виде диаграмм.

# 1 ОПИСАНИЕ ПРОЦЕССА ПРОГНОЗИРОВАНИЯ ПЛАТЕЖЕСПОСОБНОСТИ ПРЕДПРИЯТИЯ

## Общие понятия и термины

Финансовый анализ – это изучение основных показателей финансового состояния и финансовых результатов деятельности организации с целью принятия заинтересованными лицами управленческих, инвестиционных и прочих решений. Финансовый анализ является частью более широких терминов: анализ финансово-хозяйственной деятельности предприятия и экономический анализ.

На практике финансовый анализ проводят при помощи таблиц MS Excel или специальных программ. В ходе анализа финансово-хозяйственной деятельности производятся как количественные расчеты различных показателей, соотношений, коэффициентов, так и их качественная оценка и описание, сравнение с аналогичными показателями других предприятий. Финансовый анализ включает анализ активов и обязательств организации, ее платежеспособности, ликвидности, финансовых результатов и финансовой устойчивости, анализ оборачиваемости активов (деловой активности). Финансовый анализ позволяет выявить такие важные аспекты, как возможная вероятность банкротства. Финансовый анализ является неотъемлемой частью деятельности таких специалистов, как аудиторы, оценщики [3].

В основе финансового анализа лежит расчет специальных показателей, чаще в виде коэффициентов, характеризующих тот или иной аспект финансово-хозяйственной деятельности организации. Среди самых популярных финансовых коэффициентов можно выделить следующие:

* коэффициент автономии (отношение собственного капитала к общему капиталу (активам) предприятия), коэффициент финансовой зависимости (отношение обязательств к активам);
* коэффициент текущей ликвидности (отношение оборотных активов к краткосрочным обязательствам);
* коэффициент быстрой ликвидности (отношение ликвидных активов, включающих денежные средства, краткосрочные финансовые вложения, краткосрочную дебиторскую задолженность, к краткосрочным обязательствам);
* рентабельность собственного капитала (отношение чистой прибыли к собственному капиталу предприятия);
* рентабельность продаж (отношение прибыли от продаж (валовой прибыли) к выручке предприятия), рентабельность по чистой прибыли (отношение чистой прибыли к выручке).

Разработка прогнозных моделей финансового состояния предприятия необходима для выработки генеральной финансовой стратегии на обеспечение предприятия финансовыми ресурсами, оценки его возможностей в п перспективе. Она должна строиться на основе изучения реальных финансовых возможностей предприятия, внутренних и внешних факторов и охватывать такие вопросы, как оптимизация основных и оборотных смет, собственного и заемного капитала, распределение прибыли, инвестиционная и ценовая политика. Основное внимание при этом уделяется выявлению и мобилизации внутренних резервов увеличения денежных доходов, мак максимальных снижению себестоимости продукции и услуг, выработке правильной политики распределения прибыли, эффективному использованию капитала предприятия на всех стадиях его круговороту [4].

Значение прогнозного анализа финансового состояния заключается в том, что он позволяет заблаговременно оценить финансовую ситуацию с позиции ее соответствия стратегии развития предприятия с учетом изменения внутренних и внешних условий его функционирования. В основе построения прогнозной модели лежит прогноз объема продаж и необходимого объема ресурсов.

Очень часто заинтересованные лица не имеют доступа к внутренним данным организации, поэтому в качестве основного источника информации для финансового анализа выступает публичная бухгалтерская отчетность организации. Основные формы отчетности – Бухгалтерский баланс и Отчет о прибылях и убытках – дают возможность рассчитать все основные финансовые показатели и коэффициенты. Для более глубокого анализа можно использовать отчеты о движении денежных средств и капитала организации, которые составляются по итогам финансового года. Еще более детальный анализ отдельных аспектов деятельности предприятия, например, расчет точки безубыточности, требует исходных данных, лежащих за пределами отчетности (данные текущего бухгалтерского и производственного учета) [5].

Платежеспособность предприятия (организации) – это способность субъекта экономической деятельности полностью и срок погашать свою кредиторскую задолженность. Платежеспособность является одним из ключевых признаков нормального (устойчивого) финансового положений предприятия.

Платежеспособность предприятия складывается из двух факторов:

* наличие активов (имущества и денежных средства), достаточных для погашения всех имеющих у организации обязательств;
* Степень ликвидности имеющих активов достаточная для того, чтобы при необходимости реализовать их, привести в деньги в сумме достаточной для погашения обязательств [6].

При анализе первого изучают наличие у организации чистых активов (собственного капитала). Если у организации отрицательные чистые активы, т.е. отсутствует собственный капитал, то она в принципе не может расплатиться по всем своим обязательствам ввиду превышения суммы обязательства суммой на всех имеющих активов. Такая организация может быть платежеспособной в короткой перспективе, рассчитывать по текущим долгам, но в долгосрочной перспективе велика вероятность банкротства.

## Анализ модели прогнозирования банкротства Д. Фулмера

Американский экономист Джон Фулмер в 1984 году опубликовал свою работу, в которой представил модель классификации банкротства на основании обработки данных 60 предприятий: 30 потерпевших крах и 30 нормально работающих — со средним годовым оборотом в $455 000. Точность прогнозов, сделанных с помощью данной модели на год вперёд — 98 %, на два года — 81 %. Первоначальный вариант модели содержал 40 финансовых коэффициентов, окончательный использует лишь 9 [7].

Формула модели банкротства Фулмера:

H = 5.528 \* K1 + 0.212 \* K2 + 0.073 \* K3 + 1.27 \* K4 + 0.12 \* K5 + 2.235 \* \* K6 + 0.575 \* K7 + 1.083 \* K8 + 0.984 \* K9 – 3.075,

где H — интегральная оценка;

K1 = Нераспределенная прибыль прошлых лет / Активы;

K2 = Выручка от продаж / Активы;

К3 = (Прибыль до налогообложения +Проценты к уплате) / Собственный капитал;

К4 = Денежный поток / (Краткосрочные + Долгосрочные обязательства);

К5 = Долгосрочные обязательства / Активы;

К6 = Краткосрочные обязательства / Активы;

К7 = Log (материальные активы);

К8 = Оборотный капитал / (Долгосрочные + Краткосрочные обязательства);

К9 = Lg[(Прибыль до налогообложения + Проценты к уплате)/ Проценты к уплате] [8].

Результат прогноза приведен в таблице 1.

|  |  |
| --- | --- |
| **Значение Н** | **Вероятность банкротства, %** |
| *<*0 | Максимальная (90-100) |
| 0-0,18 | Высокая (60—80) |
| 0,18-0,32 | Средняя (35—50) |
| 0,32-0,42 | Низкая (15-20) |
| >0,42 | Минимальная (до 10) |

Таблица 1 – «Определение критерия вероятности банкротства предприятия в зависимости от значения модели Н»

Модель Фулмера используется для выявлений рисков хозяйственной деятельности предприятия при разработке собственных стратегических планов, а также для оценки кредитоспособности потенциальных заёмщиков и выявления рисков неплатежеспособности.

Можно сделать следующие выводы: модель Фулмера использует большое количество показателей, а значит демонстрирует больше стабильности, чем другие методики. Кроме того, она учитывает размер фирмы, что справедливо для любой страны.

## Определение проблемы, требующей автоматизации

Банкротство является результатом конкурентной борьбы на рынке товаров и услуг, с одной стороны, и неудовлетворительной работы по финансированию и кредитованию предприятий торговли - с другой. Поэтому предприятия торговли должны постоянно анализировать объем просроченной задолженности перед бюджетом, банками, поставщиками и другими контрагентами. Необходимо постоянно исследовать основные параметры предприятия, чтобы поддерживать его конкурентоспособность путем проведения соответствующих мероприятий. Именно поэтому прогнозирование платежеспособности – обязательное мероприятие для любого предприятия. Но при этом прогноз является сложным и трудоемким процессом, любое отклонение от методики или вычислительная ошибка могут стать фатальными для предприятия, что может в последствии привести к его уничтожению.

Человеческий фактор присутствует в любой сфере деятельности, прогнозирование платежеспособности предприятия не исключение. Именно поэтому процесс расчета коэффициентов и анализ полученных данных следует автоматизировать. Все, что требуется от специалиста – внести начальные данные, которые будут автоматически рассчитаны программой, что способствует снижению количества временного и человеческого ресурсов.

# ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ ПРОГРАММНОГО СРЕДСТВА

Перед программным средством поставлена задача расчета прогноза платежеспособности предприятия на основе модели банкротства Фулмера. Для реализации программной поддержки расчета необходимо определить функционал системы.

Разработка приложения предусмотрена для ролей администратора и пользователя. Программное средство будет предоставлять следующие возможности:

* регистрация;
* авторизация;
* функции администратора: механизм управления пользователями, добавление информации в базу данных, просмотр и удаление уже имеющихся данных, сохранение статистики приложения в виде текстового файла (количество пользователей, количество компаний и количество отчетов);
* функции пользователя: просмотр и добавление данных, просмотр диаграммы изменения результата прогнозирования, просмотр диаграммы изменения коэффициентов прогнозирования, сортировка и фильтрация по результату прогнозирования, поиск по дате отчета;
* обработка исключительных ситуаций.

Для решения поставленной задачи в курсовом проекте используются следующие технологии.

Для создания приложения был выбран язык программирования Java. Ключевой особенностью языка Java является то, что его код сначала транслируется в специальный байт-код, независимый от платформы. А затем этот байт-код выполняется виртуальной машиной JVM (Java Virtual Machine). В этом плане Java отличается от стандартных интерпретируемых языков, код которых сразу же выполняется интерпретатором. В то же время Java не является и чисто компилируемым языком.

Подобная архитектура обеспечивает кроссплатформенность и аппаратную переносимость программ на Java, благодаря чему подобные программы без перекомпиляции могут выполняться на различных платформах - Windows, Linux, Mac OS и т.д. Для каждой из платформ может быть своя реализация виртуальной машины JVM, но каждая из них может выполнять один и тот же код. Также Java — это объектно-ориентированный язык программирования [9].

IntelliJ IDEA — интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains. Community Edition является полностью свободной версией, доступной под лицензией Apache 2.0, в ней реализована полная поддержка Java SE, Kotlin, Groovy, Scala, а также интеграция с наиболее популярными системами управления версиями. Программа содержит полный набор необходимых для создания полноценных приложений компонент: редактор, среда компиляции и выполнения, а также отладчик.

Данная программа помогает специалисту экономить время вследствие глубокого анализа контекста и удаления неподходящих вариантов. Эта и другие детали обеспечивают повышение уровня продуктивности пользователя, одновременно позволяя ему получать больше удовольствия от деятельности.

MySQL это система управления реляционными базами данных с открытым исходным кодом (СУРБД) с моделью клиент-сервер. [СУРБД](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) — это программное обеспечение или служба, используемая для создания и управления базами данных на основе реляционной модели.

Основные цели MySQL: быстродействие и ошибкоустойчивость. Эта СУБД прекрасно справляется с обработкой очень значительных массивов данных. Кроме того, MySQL в запросах к большим таблицам превосходит многие другие системы. Она очень стабильна и превосходно работает даже в случаях, когда несколько сотен пользователей нуждаются в доступе к одним и тем же данным. MySQL также очень масштабируема, работает на всем - от персональных компьютеров до больших многопроцессорных систем. Для взаимодействия с MySQL-сервером можно использовать множество языков программирования [10].

Данный курсовой проект выполнен в архитектуре «клиент-сервер». Эта технология обеспечивает безопасность пользователя и стабильное выполнение процессов, а также согласованность и быструю обработку информации.

Серверное приложение реализовано в виде консольного приложения, а клиентское приложение – это оконное приложение с использованием стандартного инструментария языка Java – JavaFX.

JavaFX представляет собой инструментарий для создания кроссплатформенных графических приложений на платформе Java. Он позволяет создавать приложения с богатой насыщенной графикой благодаря использованию аппаратного ускорения графики и возможностей GPU.

С помощью JavaFX можно создавать программы для различных операционных систем: Windows, MacOS, Linux и для самых различных устройств: десктопы, смартфоны, планшеты, встроенные устройства, ТВ. Приложение на JavaFX будет работать везде, где установлена исполняемая среда Java (JRE). Также JavaFX предоставляет большие возможности по сравнению с рядом других подобных платформ и большой набор элементов управления, и возможности по работе с мультимедиа, двухмерной и трехмерной графикой, декларативный способ описания интерфейса с помощью языка разметки FXML, возможность стилизации интерфейса с помощью CSS, интеграция со Swing и многое другое [11].

# 3 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ НА ОСНОВЕ СТАНДАРТА IDEF0

Для представления процесса прогнозирования платежеспособности предприятия была разработана функциональная модель курсового проекта. Для данной цели был выбран стандарт IDEF0. IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для иллюстрации и описания бизнес-процессов.

Ключевым процессом является прогноз платежеспособности предприятия.

На контекстной диаграмме верхнего уровня (рисунок 3.1) представлена функциональная модель «Прогноз платежеспособности предприятия», а также определены потоки входных и выходных данных, механизмы ограничения и управления данными.

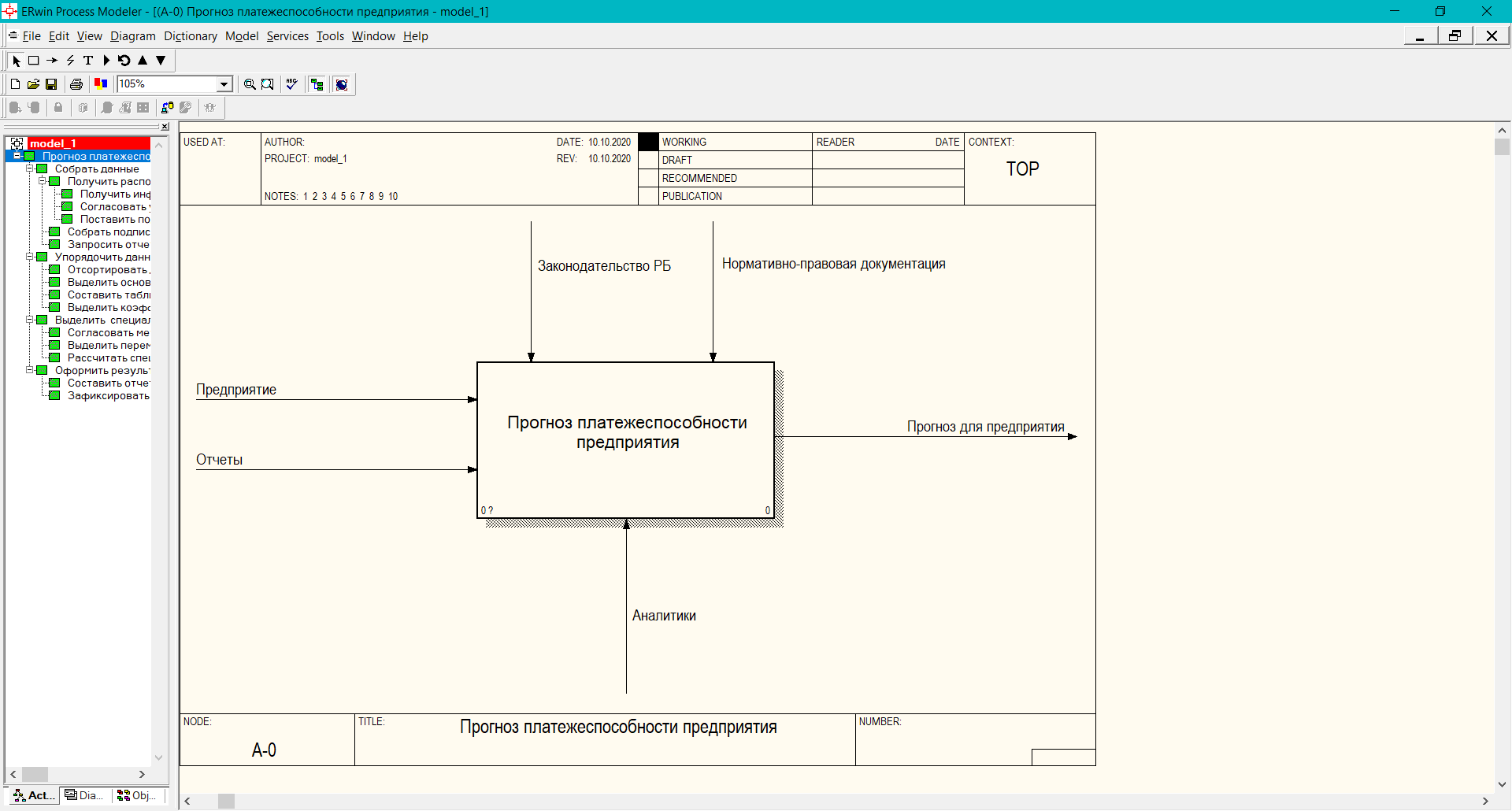


Рисунок 3.1 – Контекстная диаграмма прогноза платежеспособности предприятия

Входными данными являются предприятие и отчеты; управляющими механизмами являются законодательство РБ и нормативно-правовая документация; механизмом являются аналитики. Целью данной бизнес-модели является прогноз для предприятия, который является выходным параметром данной диаграммы.

Декомпозиция данной диаграммы подробно представлена на рисунке 3.2. Она состоит из следующих блоков: «Собрать данные», «Упорядочить данные», «Выделить специальные показатели» и «Оформить результаты».

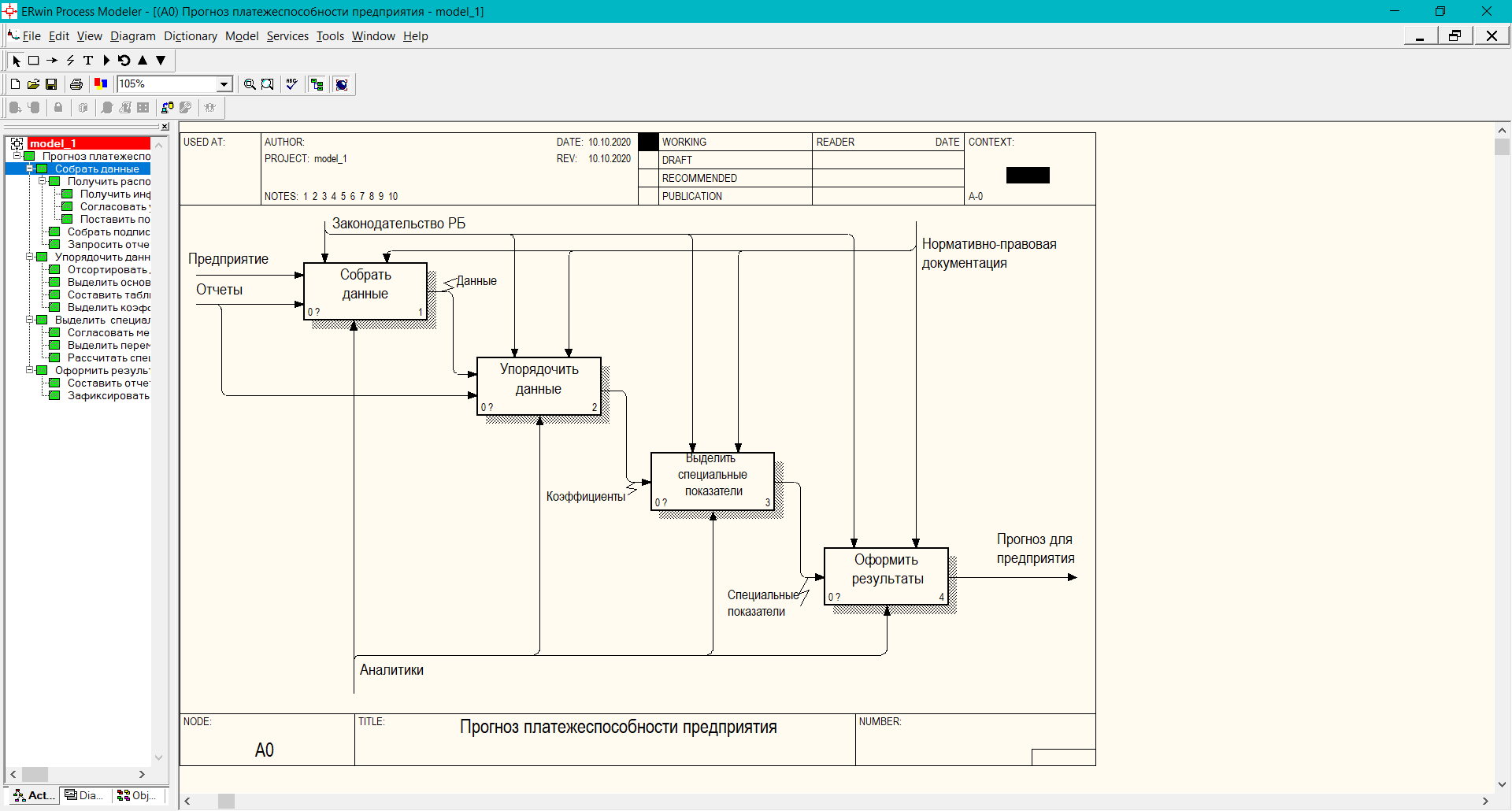


Рисунок 3.2 – Декомпозиция главного блока

Первый блок данной декомпозиции «Собрать данные» подразумевает получение информации, получение распоряжения, сбор подписей и запрос отчетов.

Каждый из данных блоков определяет свою собственную группу действий и процессов, происходящих внутри предприятия. Все процессы зависимы друг от друга и имеют общие входные данные, управляющие механизмы. Но при этом у каждого блока есть уникальная цель, которая впоследствии становится входными данными следующего блока и т. д.

Декомпозиция данного блока представлена на рисунке 3.3.

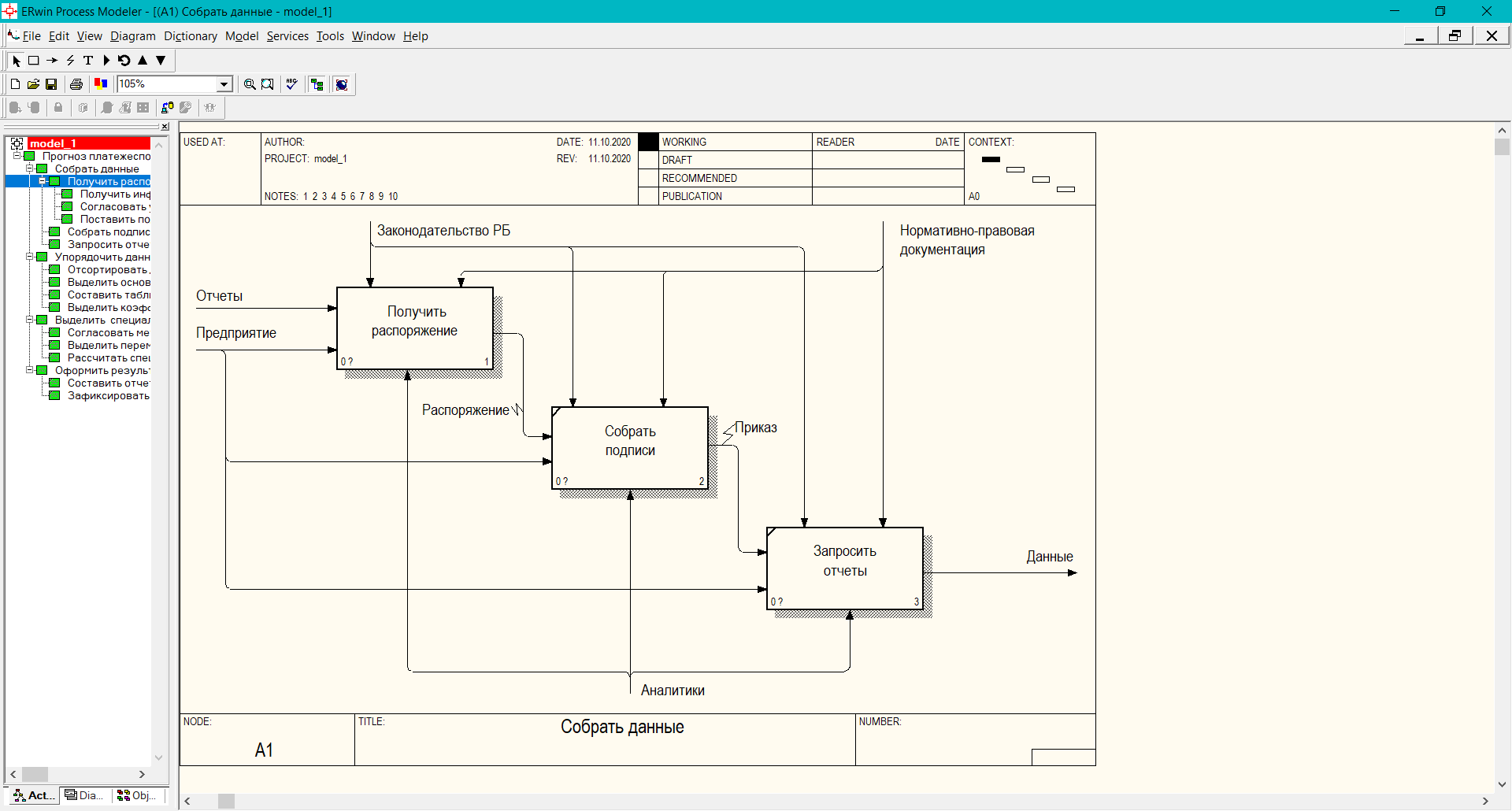


Рисунок 3.3 – Декомпозиция блока «Собрать данные»

Данная декомпозиция состоит из следующих этапов: «Получить информацию о распоряжении», «Согласовать условия», «Поставить подпись».

Как и в других декомпозициях здесь каждый блок имеет одинаковые управляющие механизмы, так как акты нормативно-правовой документации должны соблюдаться в любом процессе, как и законодательство РБ, которое регулирует все экономические процессы в стране. Их нарушение влечет за собой нарушение работоспособности остальных процессов в этом блоке и последующих за ним. Поэтому важно, чтобы управляющие механизмы присутствовали в каждом блоке процесса независимо от его сложности и количества уровней.

Этап «Получить информацию о распоряжении» подразумевает получении информации о содержании распоряжения: сроках выполнения, целях и составе команды аналитиков. Декомпозиция данного этапа представлена на рисунке 3.4.

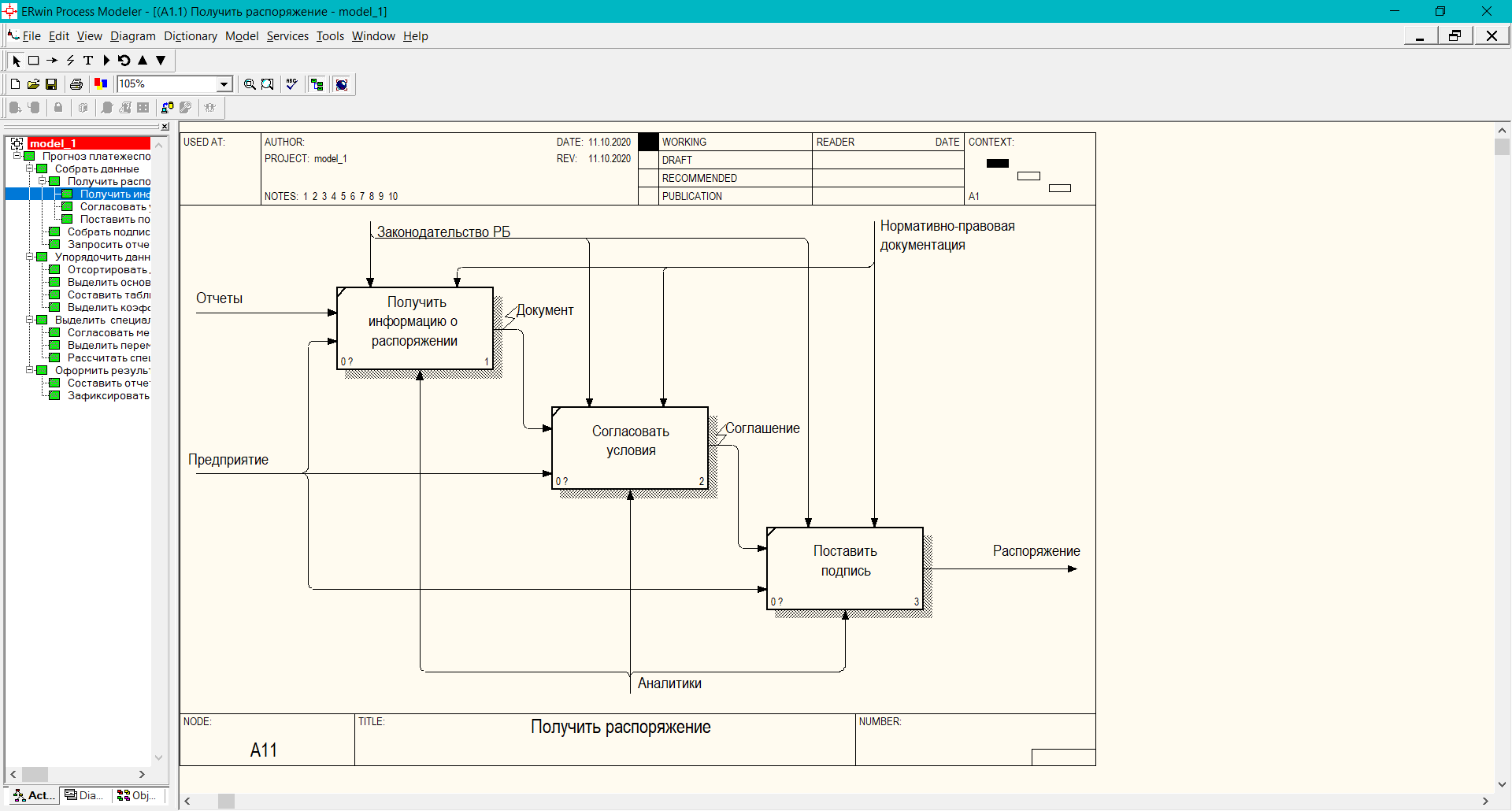


Рисунок 3.4 – Декомпозиция блока «Получить распоряжение»

Вторым блоком декомпозиции главного процесса является «Упорядочить данные», который предполагает подготовку полученных данных для дальнейшего расчета.

Для каждого блока во всем процессе существуют механизмы «Аналитики», которые представляют собой трудовой ресурс, отвечающий за выполнение данного процесса. Так как процесс «Прогноз платежеспособности предприятия» является полностью экономическим процессом, его исполнением должны заниматься люди с экономическим образованием. В предприятиях, обычно, эти люди занимают должность аналитиков, основной их задачей является просчет финансовых действий и прогнозов.

На рисунке 3.5 представлена декомпозиция данного блока.

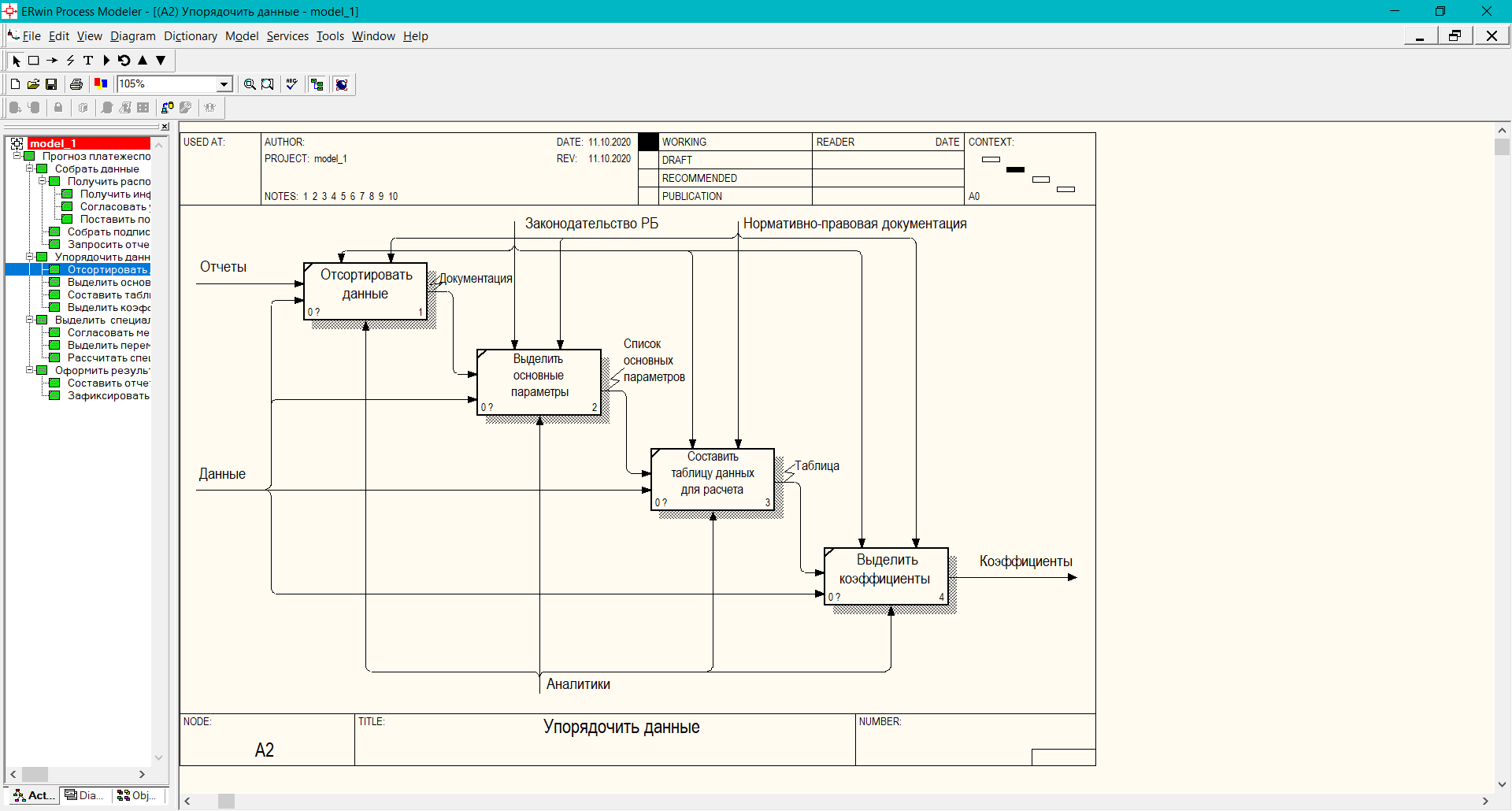


Рисунок 3.5 – Декомпозиция блока «Упорядочить данные»

Данная декомпозиция состоит из следующих блоков: «Отсортировать данные», «Выделить основные параметры», «Составить таблицу данных для расчета» и «Выделить коэффициенты».

Входные данные с именем «Отчеты» входят не во все процессы, так как «Отчеты» — собирательный образ документации, ведущейся в предприятиях. Документация может быть как электронная, так и бумажная, бывает одновременно 2 видов. В предприятии любого вида и типа ведение документации – важный и непрерывный процесс. Благодаря документации можно просчитать прибыль, затраты, издержки, выручку и огромное количество других экономических показателей, характеризующих финансовое положение предприятия. Однако «Отчеты» нужны лишь для получения исходных данных, именно поэтому они входят не во все процессы.

Третьим блоком главного процесса является «Выделить специальные показатели», который подразумевает выбор метода подсчета, выбор переменных и расчет специальных показателей.

На рисунке 3.6 представлена декомпозиция данного блока.

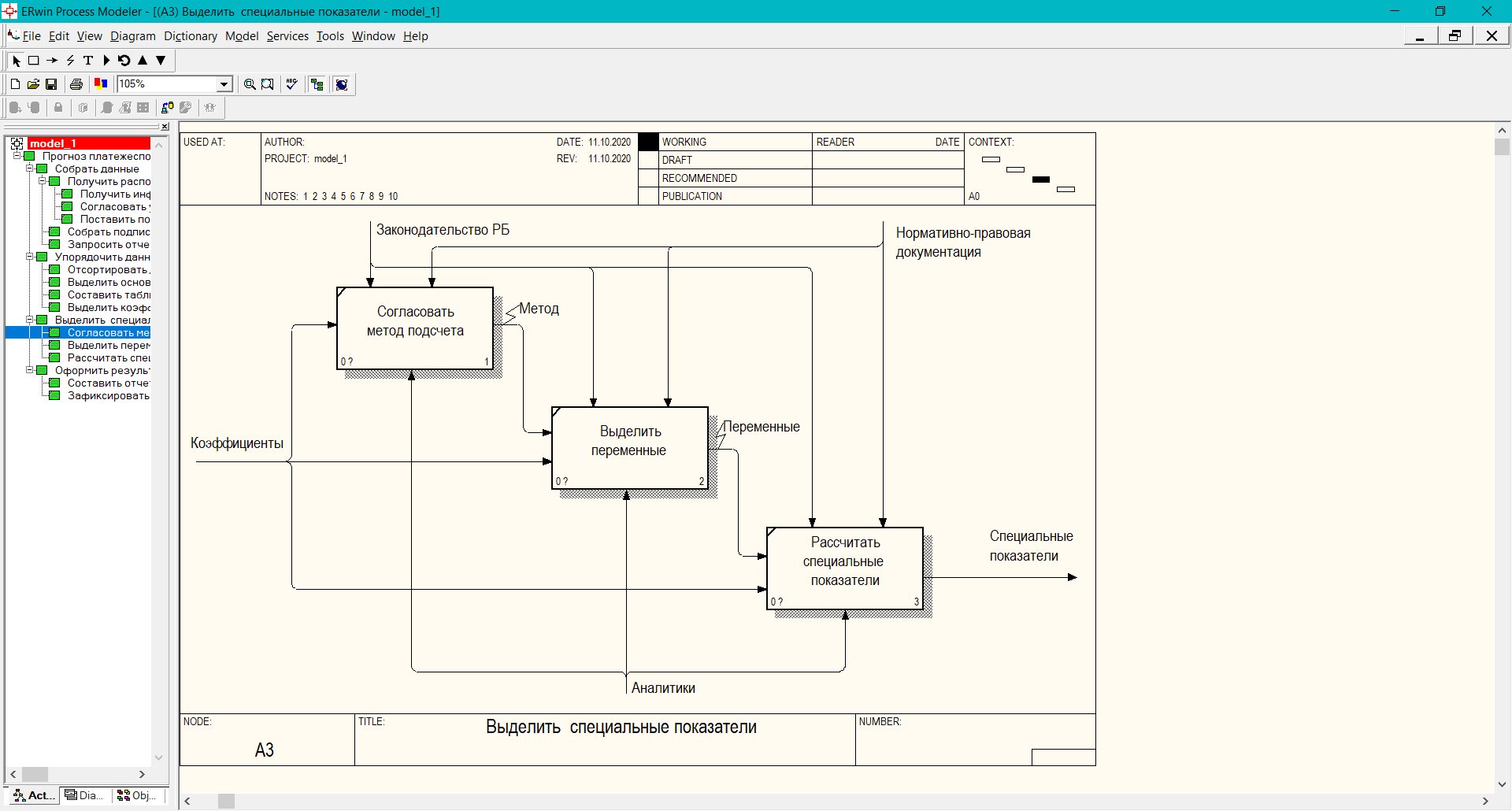


Рисунок 3.6 – Декомпозиция блока «Выделить специальные показатели»

Данная декомпозиция состоит из следующих блоков: «Согласовать метод подсчета», «Выделить переменные» и «Рассчитать специальные показатели».

Входные данные «Предприятие» также как и «Отчеты» являются собирательным образом. Они представляют собой остальные отделы и службы, существующие в предприятии и косвенно влияющие на блоки процессов. Например, при получении информации о распоряжении в блоке процесса присутствует «Предприятие», в данном случае, оно служит источником данных. В реальной жизни это может быть директор отдела аналитики или непосредственный начальник, или же это может быть документ, представляющий собой поручение кого-либо из вышестоящего состава предприятия.

Последним этапом главного процесса является «Оформить результат», который подразумевает составление отчета о проделанной работе и оформление прогноза для предприятия.

Декомпозиция данного процесса представлена на рисунке 3.7.

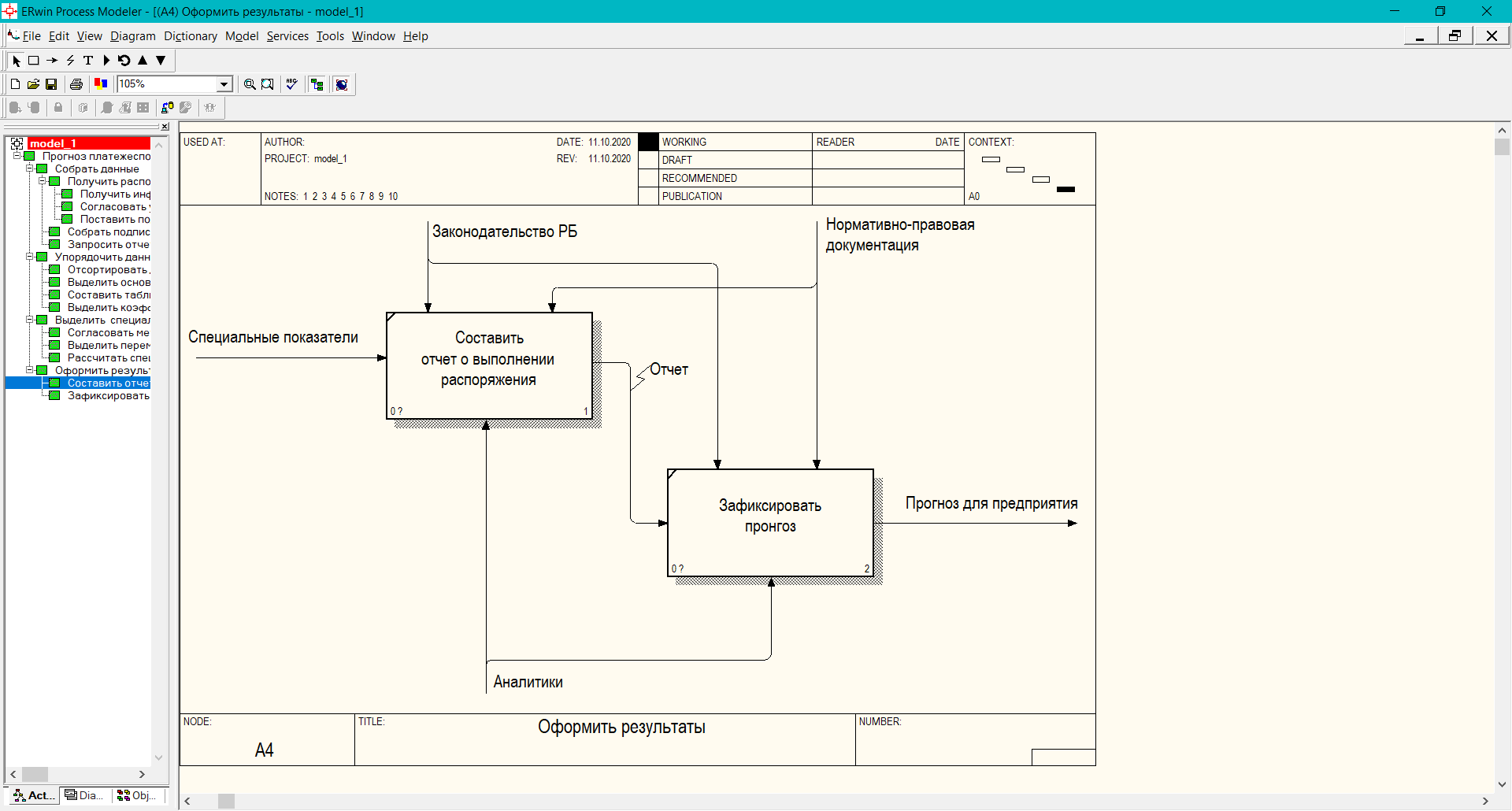


Рисунок 3.7 – Декомпозиция блока «Оформить результат»

Данная декомпозиция состоит из следующих блоков: «Составить отчет о выполнении распоряжения», «Зафиксировать прогноз».

Таким образом, из представленных декомпозиций можно сделать вывод о том, насколько сложным и многоступенчатым процессом является прогнозирование платежеспособности предприятия. Также становится очевидным тот факт, что избежать проблем на любом из этапов прогнозирования почти невозможно. Это еще раз подтверждает необходимость создания приложения для улучшения данного процесса.

# ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ И ЕЁ ОПИСАНИЕ

Информационная модель в данном курсовом проекте была построена с помощью приложения DBeaver. DBeaver — это клиент SQL и средство администрирования базы данных. Для реляционных баз данных он использует JDBC API для взаимодействия с базами данных с помощью драйвера JDBC. Он предоставляет редактор, поддерживающий компиляцию кода и подсветку синтаксиса. Он предоставляет архитектуру плагинов (основанную на архитектуре плагинов [Eclipse](https://wiki.yola.ru/eclipse/eclipse" \o "eclipse:eclipse)), которая позволяет модифицировать поведение большой части приложения для обеспечения специальных функций базы данных, которые не зависят от самой базы данных.

Это настольное приложение, написанное на [Java](https://wiki.yola.ru/java/java" \o "java:java) и основанное на платформе [Eclipse](https://wiki.yola.ru/eclipse/eclipse" \o "eclipse:eclipse). DBeaver Community является бесплатным программным обеспечением с открытым исходным кодом, которое распространяется под лицензией Apache [12].

Сущностями данной модели являются пользователь, компания, отчет, валюта, пользователь-компания и пользователь-отчет.

Сущность «Пользователь» содержит информацию о пользователях, зарегистрированных в базе данных: идентификационный номер пользователя, фамилия, имя, адрес электронной почты, номер телефона, логин и пароль, используемые для входа в систему. Ключевым полем данной сущности является идентификационный номер пользователя.

Сущность «Компания» содержит информацию о компаниях: идентификационный номер компании, название и адрес. Ключевым полем данной сущности является идентификационный номер компании.

Сущность «Отчет» содержит информацию об отчетах, произведенных пользователями: идентификационный номер отчета, идентификационный номер курса валют, идентификационный номер компании, для которой осуществляется прогноз, значение результата прогноза и 9 переменных, описанных в пункте 1.2 (см. выше). Сущность «Отчет» связана с сущностями «Курс» и «Компания» по идентификационному номеру курса валют и идентификационному номеру компании соответственно. Ключевым полем данной сущности является идентификационный номер отчета.

Сущность «Курс» содержит информацию о курсе валют: идентификационный номер курса валют, отношение доллара к белорусскому рублю, отношение доллара к евро, отношение доллара к российскому рублю и дата данного курса валют. Ключевым полем данной сущности является идентификационный номер курса валют.

Сущность «Пользователь-компания» содержит информацию о связи сущности «Пользователь» и «Компания»: идентификационный номер пользователя и идентификационный номер компании соответственно.

Сущность «Пользователь-отчет» содержит информацию о связи сущности «Пользователь» и «Отчет»: идентификационный номер пользователя и идентификационный номер отчета соответственно.

Общая схема связанных сущностей базы данных представлена на рисунке 4.1.

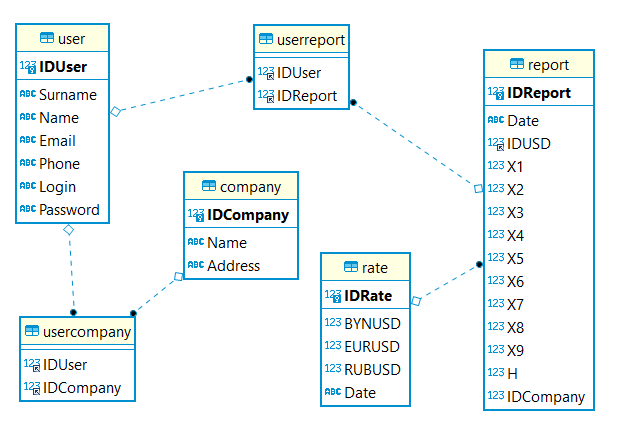


Рисунок 4.1 – Информационная модель базы данных

Для доказательства нахождение таблиц в третьей нормальной форме необходимо воспользоваться следующими понятиями. Таблица находится в первой нормальной форме, если все ее поля имеют простые значения. Таблица находится во второй нормальной форме, если она находится в первой нормальной форме, а каждое не ключевое поле функционально полно зависит от составного ключа. Таблица находится в третьей нормальной форме, если она находится во второй нормальной форме, и каждое не ключевое поле не транзитивно зависит от первичного ключа.

Таким образом, база данных приведена к третей нормальной форме, поскольку у каждой таблицы имеется всего один первичный ключ, а каждое не ключевое поле не транзитивно зависит от первичного ключа, т.е. изменив значение в одном столбце не потребуется изменение в другом столбце.

# 5 СПЕЦИФИКАЦИЯ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ ПЛАТЕЖЕСПОСОБНОСТИ ПРЕДПРИЯТИЯ

Диаграмма вариантов использования (сценариев поведения, прецедентов) является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Данная диаграмма состоит из актеров, вариантов использования и отношений между ними. При построении диаграммы могут использоваться также общие элементы нотации: примечания и механизмы расширения.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером называется любой объект, субъект или система, взаимодействующая с моделируемой системой извне. В свою очередь вариант использования – это спецификация сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемых системой при взаимодействии с актером. При этом в модели никак не отражается то, каким образом будет реализован этот набор действий.

Диаграмма вариантов использования данного проекта представлена на рисунке 5.1.

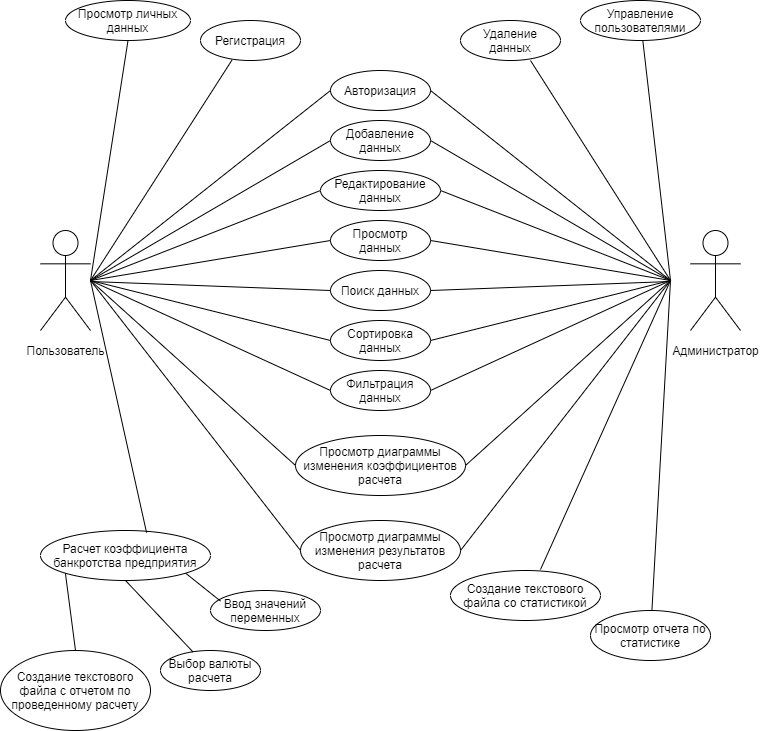


Рисунок 5.1 – Диаграмма вариантов использования

Актерами в данной диаграмме являются пользователь и администратор.

Таким образом, разрабатываемая система позволит разделить роли участников, предоставив им разные возможности.

# модели представления системы прогнозирования платежеспособности предприятия и их описание

Модель представления данных – это множество допустимых типов данных и отношений между ними, ограничений и действий над этими типами данных и отношений. Множество допустимых типов данных и отношений называют структурой данных. Модель данных является ядром базы данных. В данном курсовом проекте рассматриваются следующие модели представления разрабатываемой системы:

− диаграмма развертывания;

− диаграмма компонентов;

− диаграмма последовательности;

− диаграмма состояний;

− диаграмма классов.

Рассмотрим каждую из них.

Диаграмма развертывания для разработанной системы представлена на рисунке А.1. В качестве узлов на ней выступают клиентское и серверное приложения, которые связаны с помощью протокола TCP/IP. Еще одним узлом системы является база данных, соединяющаяся с сервером с помощью драйвера JDBC. Для запуска разрабатываемого приложения необходимо наличие исполняемой среды JDK минимальной версии JDK8 на компьютере пользователя.

На рисунке А.2 изображена диаграмма компонентов приложения. Она иллюстрирует как выглядит модель системы на физическом уровне. Компонентами являются исполняемые файлы MultiThreadedServer.jar и Client.jar, а также содержимое пакетов MultiThreadedServer и Client.

На рисунке А.3 представлена диаграмма последовательности просмотра статистики по отчетам администратором. Для этого администратор отправляет запрос серверу. Сервер, в свою очередь, обращается к базе данных, чтобы получить данные для анализа. Обработав эти данные, сервер отправляет статистику администратору.

Диаграмма состояний отчета изображена на рисунке А.4. Начальным событием является нажатие кнопки «Сохранить отчет .txt», после которого отчет ожидается. Затем происходит отправка запроса на сервер, и состояние отчета определяется как «Запрошен». Если произошла ошибка при получении доступа к базе данных, отчет вернется в состояние «Ожидается». В противном случае происходит извлечение информации из базы данных и отчет переходит в состояние «Обрабатывается». Если произошла ошибка при создании файла, отчет снова будет ожидаться, в противном случае он переходит в состояние «Создан». Далее в отчет записываются полученные данные и он переходит в состояние «Записан», после чего файл сохраняется и процесс формирования отчета завершается.

Так как это приложение создано на основе клиент-серверной архитектуры, то для клиента и для сервера определены свои классы.

Рассмотрим классы серверного приложения.

На рисунке А.5 изображена диаграмма классов основных сущностей, необходимых для реализации проекта. Ими являются User (пользователь), Company (компания), Report (отчет), USD (курс валют). Классы DataHandler и Configs предназначены для взаимодействия с базой данных (рисунок А.6). Класс, реализующий основные расчеты – Report – изображен на рисунке А.7. В нем записываются введенные пользователем данные, высчитывается и сохраняется в базу данных результат расчета. Диаграмма остальных классов серверной части приложения представлена на рисунке А.8.

Рассмотрим классы клиентского приложения.

Рисунок А.9 содержит диаграмму классов, реализующих связь с графическим интерфейсом приложения. Каждой из сущностей соответствует свой класс, предназначенный для взаимодействия с базой данных. Они соответствуют одноименным классам серверной части приложения. При этом для сущностей «Пользователь-компания» и «Пользователь-отчет» созданы специальные классы – Users, CollectionUsers, CollectionCompanies и CollectionReports (рисунок А.10). На рисунке А.11 изображена диаграмма остальных классов серверной части приложения.

Все вышеперечисленные диаграммы являются результатом моделирования данного курсового проекта, и были использованы для его разработки.

# 7 описание применения паттернов

# проектирования

В данном курсовом проекте были реализованы такие паттерны проектирования, как Singleton и MVC.

Singleton (Одиночка) – это порождающий паттерн, который гарантирует существование только одного объекта определённого класса, а также позволяет достучаться до этого объекта из любого места программы. Одиночка имеет такие же преимущества и недостатки, что и глобальные переменные. Его невероятно удобно использовать, но он нарушает модульность кода. Благодаря этому паттерну нельзя использовать класс, зависящий от одиночки в другой программе. Для этого придётся эмулировать присутствие одиночки и там.

Данный паттерн реализован в классах User, Client, CollectionCompanies, CollectionUsers клиентской части приложения (Рисунок 7.1).

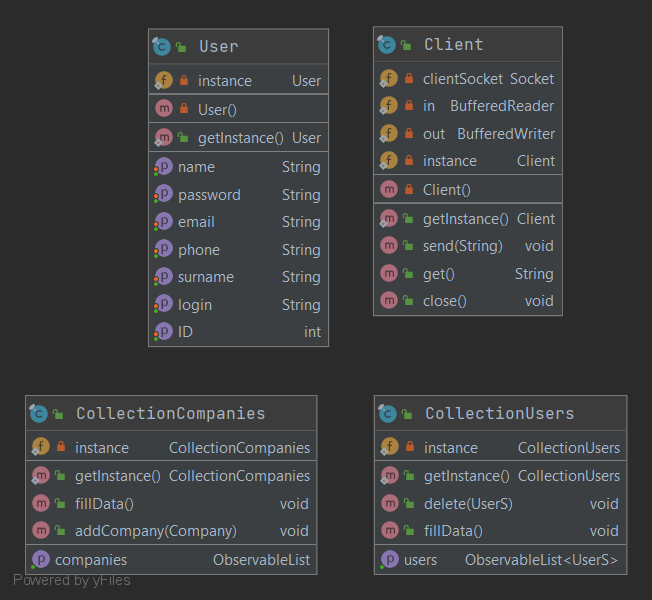


Рисунок 7.1 – Диаграмма классов, использующих паттерн Одиночка (Singleton)

MVC — это паттерн проектирования, который включает в себя несколько более мелких шаблонов. Основная цель применения MVC состоит в разделении данных и бизнес-логики от визуализации. За счет такого разделения повышается возможность повторного использования программного кода: например, добавить представление данных какого-либо существующего маршрута не только в виде HTML, но и в форматах JSON, XML, PDF, XLSX становится очень просто и не требует изменений слоя бизнес-логики исходного маршрута. Также упрощается и сопровождение программного кода: внесение изменений во внешний вид, например, не отражаются на бизнес-логике, а изменения бизнес-логики не затрагивают визуализацию.

Рассмотрим каждый компонент данного паттерна.

Модель (Model) — предоставляет собой объектную модель некой предметной области, включает в себя данные и методы работы с этими данными, реагирует на запросы из контроллера, возвращая данные и/или изменяя своё состояние. При этом модель не содержит в себе информации о способах визуализации данных или форматах их представления, а также не взаимодействует с пользователем напрямую. В данном курсовом проекте реализован в качестве классов клиентской части программы (Рисунок 7.2).

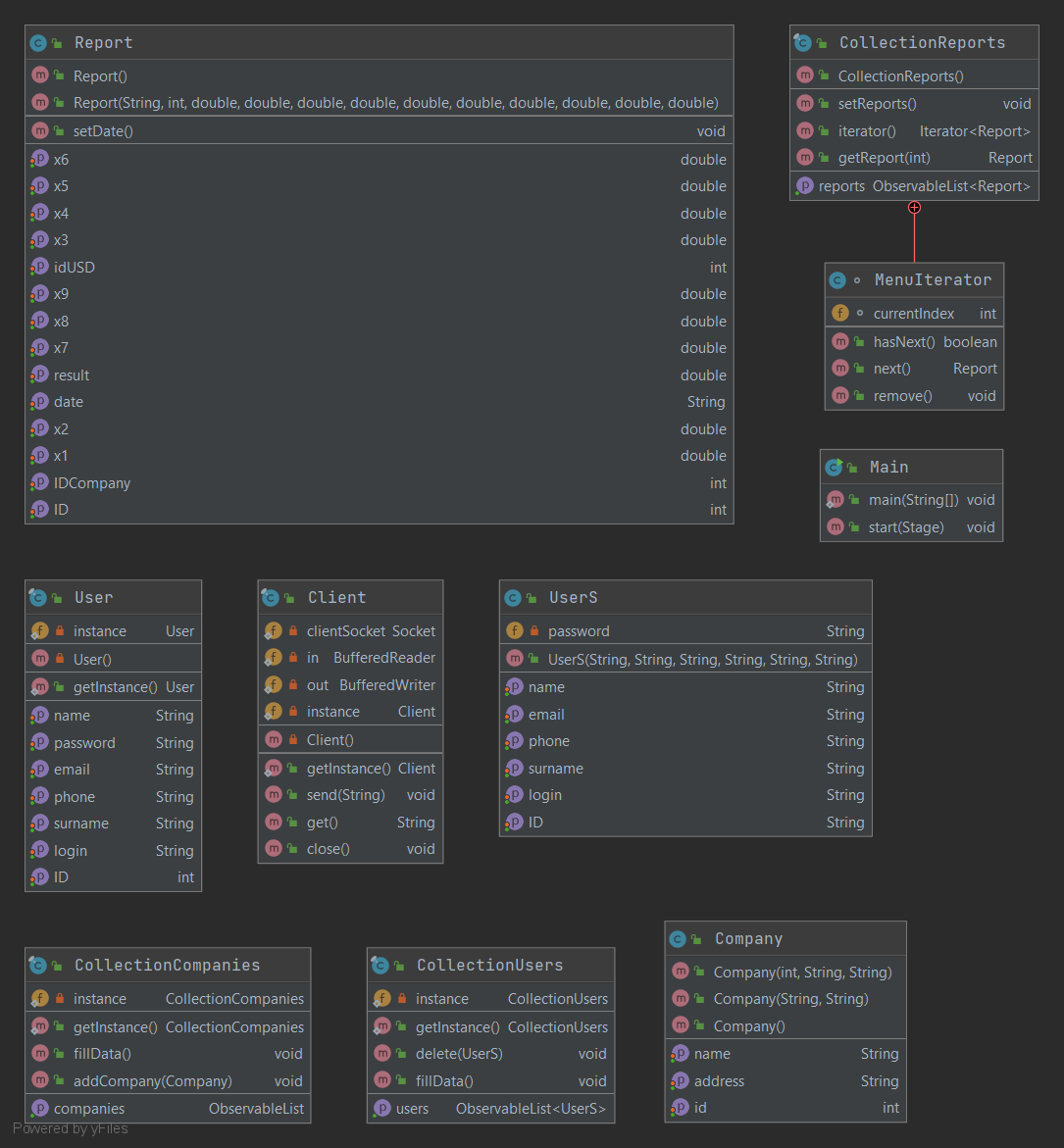


Рисунок 7.2 – Диаграмма классов клиентской части приложения

Представление (View) — отвечает за отображение информации (визуализацию). Одни и те же данные могут представляться различными способами и в различных форматах. Например, коллекцию объектов при помощи разных представлений можно представить на уровне пользовательского интерфейса как в табличном виде, так и списком; на уровне API можно экспортировать данные как в JSON, так в XML или XSLX. В данном курсовом проекте реализован набором FXML файлов, находящихся в пакете «FXML» (Рисунок 7.3).

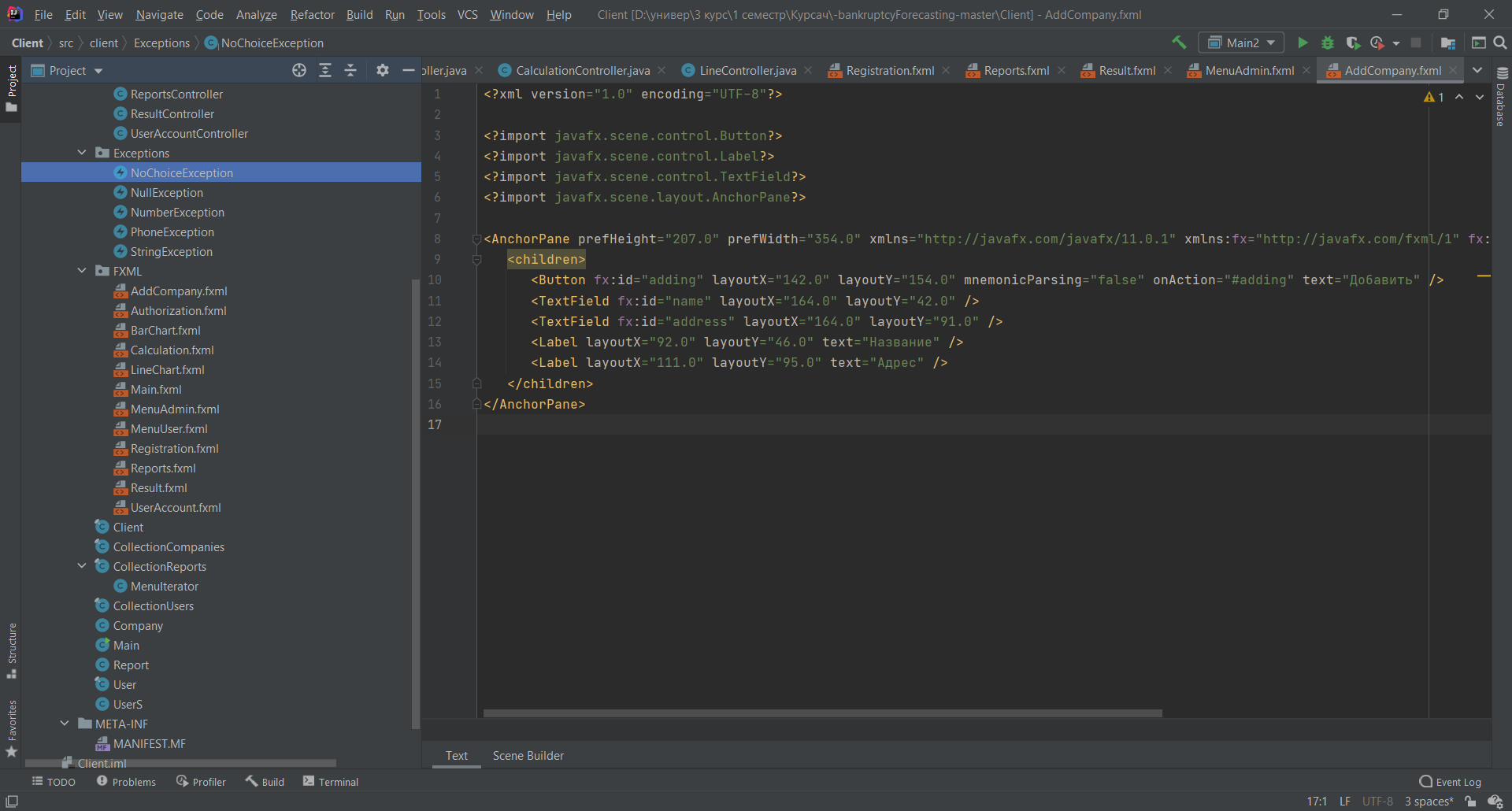


Рисунок 7.3 – Пакет FXML клиентской части приложения

Контроллер (Controller) — обеспечивает связь между пользователем и системой, использует модель и представление для реализации необходимой реакции на действия пользователя. Как правило, на уровне контроллера осуществляется фильтрация полученных данных и авторизация — проверяются права пользователя на выполнение действий или получение информации. Реализация данного компонента представлена пакетом «Controllers» (Рисунок 7.4).

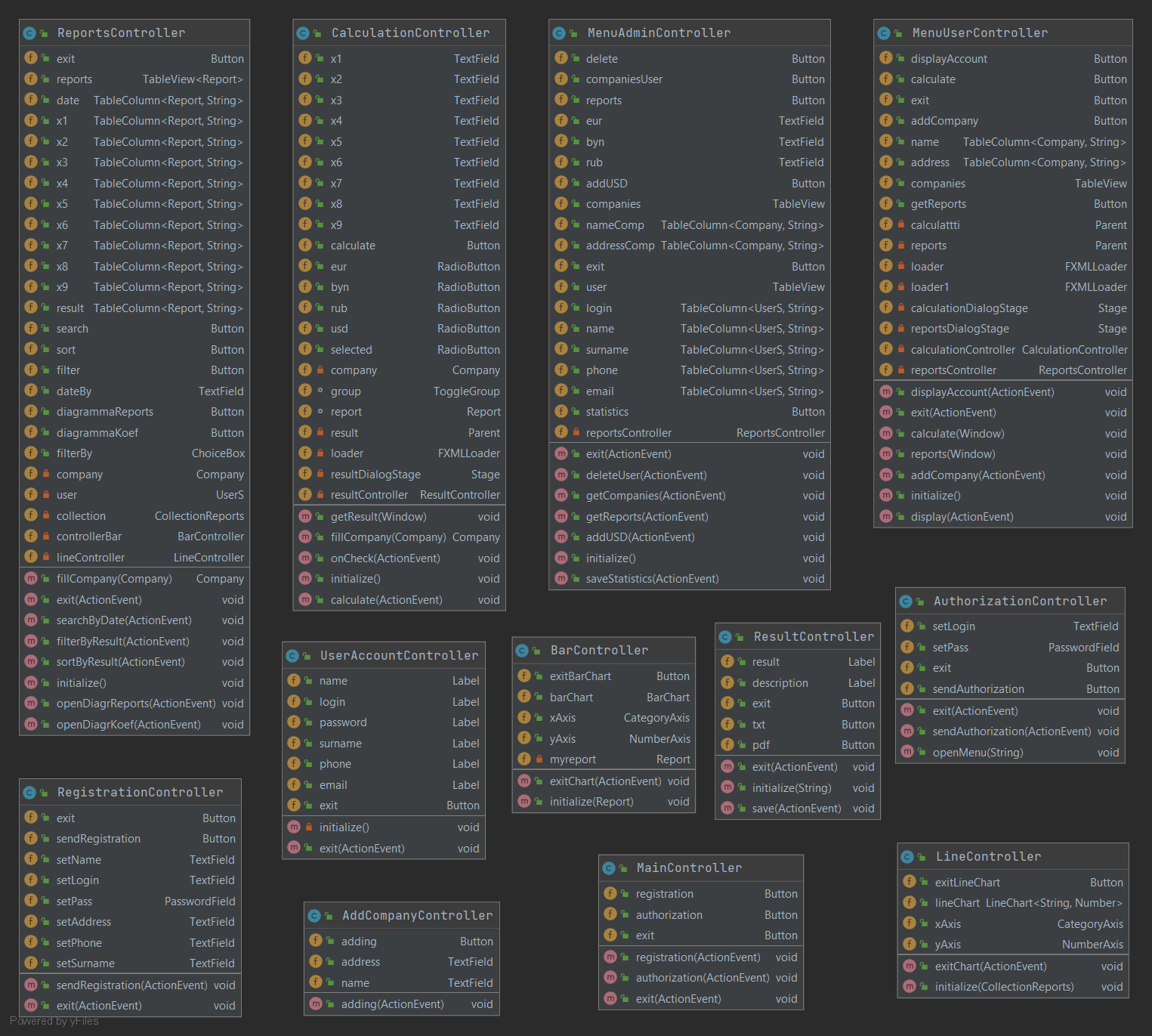


Рисунок 7.4 – Диаграмма классов пакета Controllers клиентской части приложения

# ОПИСАНИЕ АЛГОРИТМа РЕАЛИЗУЮЩего БИЗНЕС-ЛОГИКУ СЕРВЕРНОЙ ЧАСТИ системы прогнозирования платежеспособности предприятия методом фулмера и алгоритма обработки запросов клиентов на примере авторизации

Рассмотрим алгоритм прогнозирования платежеспособности предприятия методом Фулмера. Сначала каждая из переменных просчитывается исходя из таких экономических показателей предприятия, как активы, нераспределенная прибыль прошлых лет, выручка от продаж, прибыль до налогообложения, проценты к уплате, собственный капитал, денежный поток, краткосрочные и долгосрочные обязательства, материальный активы и оборотный капитал. Всего в формуле должно быть 9 переменных и 10 постоянных коэффициентов. Исходя из значений переменных высчитывается значение коэффициента банкротства. После чего происходит оценка коэффициента банкротства и вывод результата, понятного пользователю. Схема алгоритма расчета прогнозирования платедеспособности предприятия методом Фулмера представлена на рисунке 8.1.

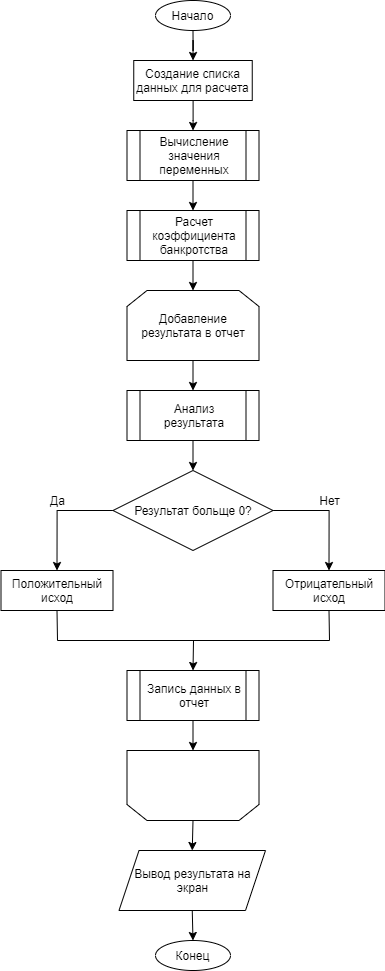


Рисунок 8.1 – Схема алгоритма расчета прогнозирования платедеспособности предприятия методом Фулмера

Так как данный курсовой проект является клиент-серверным приложением для любой работы с данными нужно подключение к базе данных через запросы сервера. Например, для успешной авторизации клиент сначала должен послать запрос серверу на получение нужной информации из базы данных. В случае отказа или неправильно введенных данных у клиента должно быть выведено сообщение об ошибке. В противном случае запрошенная клиентом информация передается на обработку серверу, после чего сервер пересылает их клиенту. Схема процесса авторизации представлена на рисунке 8.2.

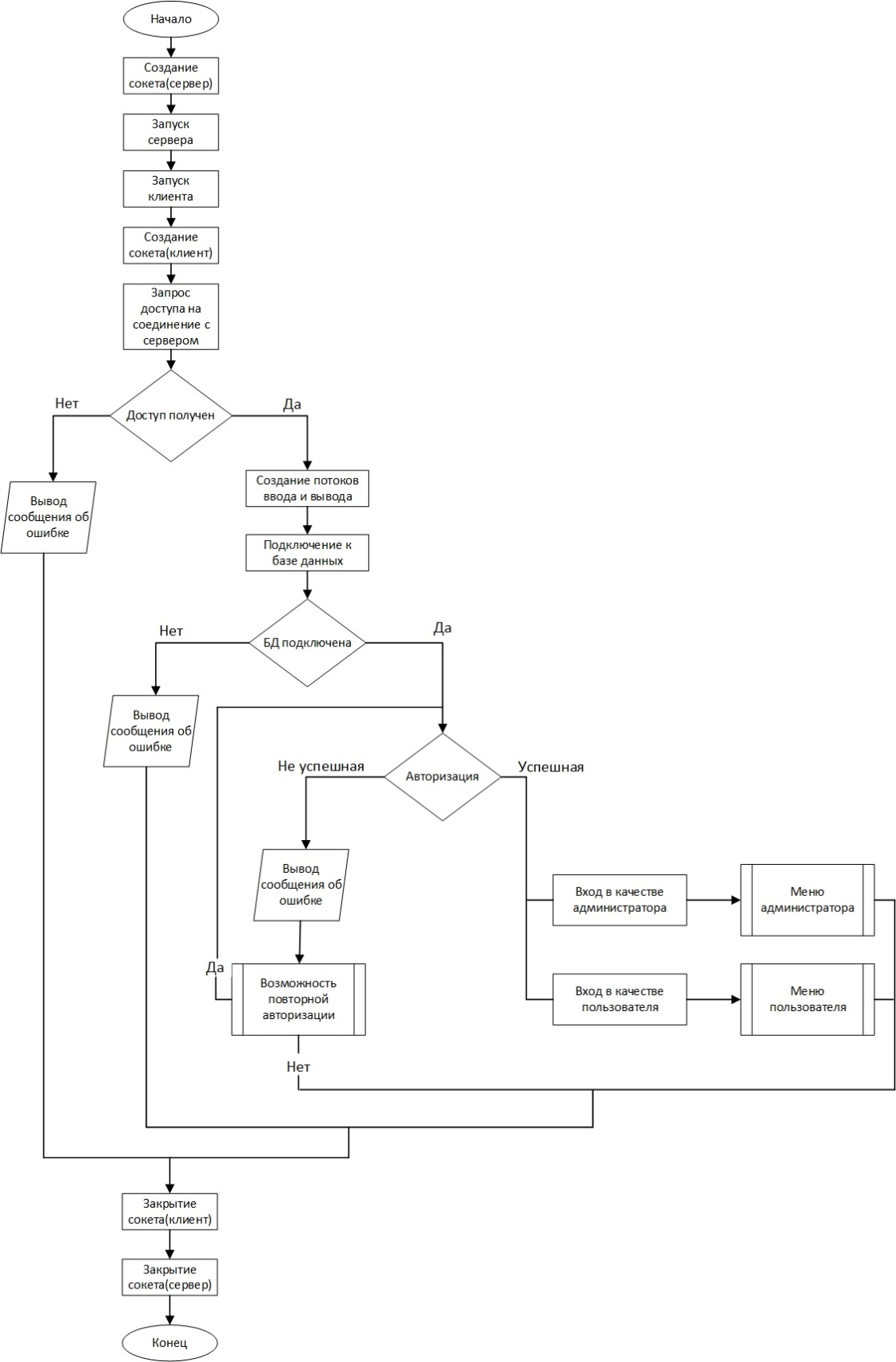


Рисунок 8.2 – Схема процесса авторизации

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПО РАЗВЁРТЫВАНИЮ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ ПЛАТЕЖЕСПОСОБНОСТИ ПРЕДПРИЯТИЯ МЕТОДОМ ФУЛМЕРА

Для корректной работы системы необходимо установить JDK 8, локальный сервер MySQL 8.0.22 и драйвер JDBC для MySQL версии 8.0.22. Скрипт генерации базы данных представлен в приложении Б.

Серверная часть выполнена в виде консольного приложения, клиентская – в виде GUI, и обе эти части упакованы в jar-архивы. Для развертывания системы необходимо запустить 2 файла. Для запуска сервера в командной строке нужно перейти в папку проекта, затем ввести путь “out\artifacts\server”, после чего выполнить команду “java –jar MultiThreadedServer.jar” (рисунок 9.1).

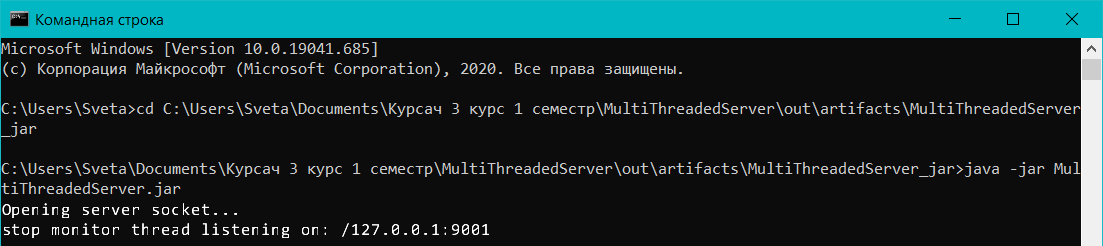


Рисунок 9. 1 – Пример запуска серверной части приложения

Клиентское приложение запустится двойным щелчком левой кнопкой мыши по иконке приложения client.jar. После успешного запуска обоих приложений нужно продолжать работу с client.jar, а MultiThreadedServer.jar может работать в фоновом режиме.

Приложение готово к работе.

# **РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ ПЛАТЕЖЕСПОСОБНОСТИ ПРЕДПРИЯТИЯ МЕТОДОМ ФУЛМЕРА**

При запуске приложения происходит вывод меню первого порядка. Здесь необходимо выбрать действие: авторизация, регистрация или же выход из программы. На рисунке 10.1 представлен скриншот окна меню первого порядка.

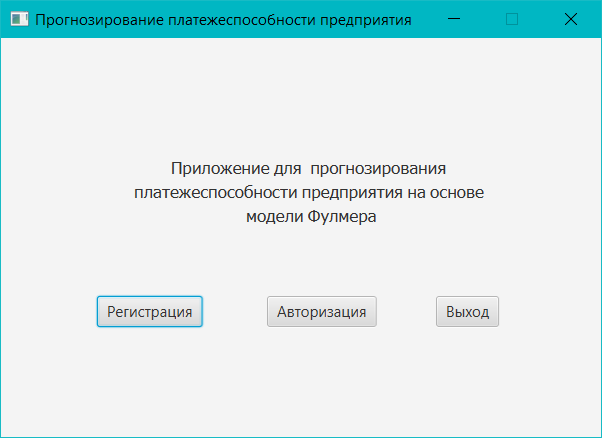


Рисунок 10.1 – Окно меню первого порядка

При выборе пункта «Регистрация» пользователь попадает в окно регистрации (Рисунок 10.2).

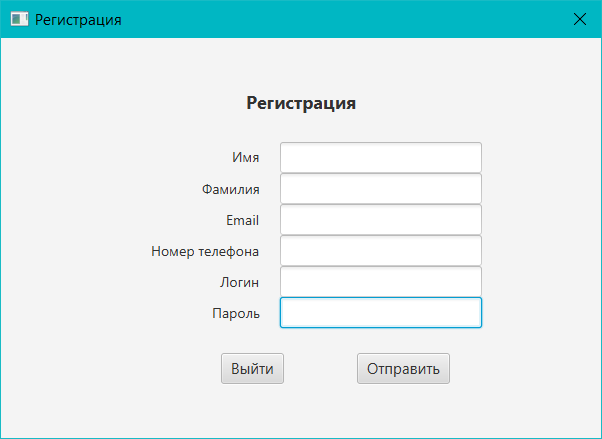


Рисунок 10.2 – Окно регистрации пользователя

Для успешной регистрации следует заполнить все поля, иначе будет выведено соответствующее сообщение об ошибке (Рисунок 10.3).

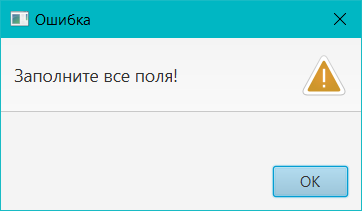


Рисунок 10.3 – Сообщение об ошибке при незаполненном поле

В случае успешной регистрации пользователь попадает в меню пользователя (Рисунок 10.4).

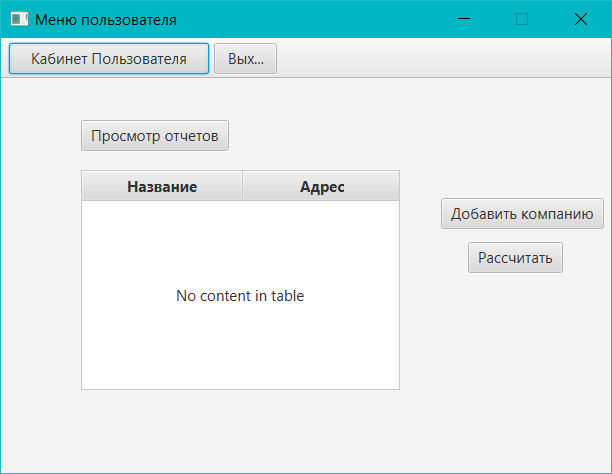


Рисунок 10.4 – Окно меню пользователя

В окне меню пользователя отображается таблица компаний, для которых пользователь рассчитывает данные, также пользователю предоставляются следующие функции: просмотр отчетов, добавление компании, расчет значения для выбранной компании и переход в кабинет пользователя.

При переходе в кабинет пользователя в новом окне отображаются личные данные пользователя (Рисунок 10.5).

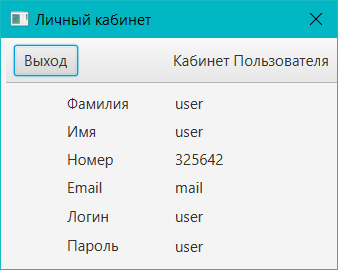


Рисунок 10.5 – Окно кабинета пользователя

При нажатии кнопки «Добавить компанию» открывается новое окно для добавления компании в таблицу (Рисунок 10.6).

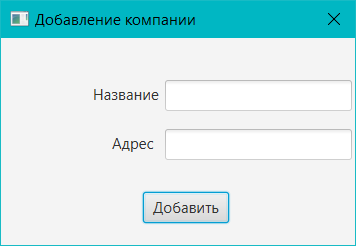


Рисунок 10.6 – Окно добавления компании

Если данные введены правильно, то в таблице меню пользователя появится новая строка с компанией. При выборе строки таблицы и нажатии на кнопку «Рассчитать», пользователь переходит в окно ввода значений переменных и выбора валюты производимого расчета (Рисунок 10.7).

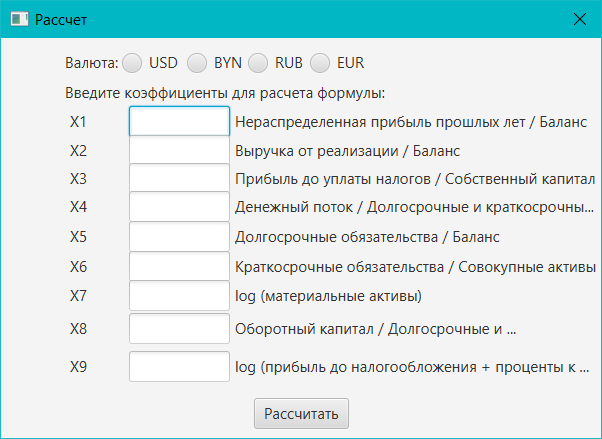


Рисунок 10.7 – Окно ввода значения переменных и выбора валюты

Если в хотя бы одно из полей было введено значение, не соответствующее цифровому формату, то приложение выведет сообщение об ошибке (Рисунок 10.8).

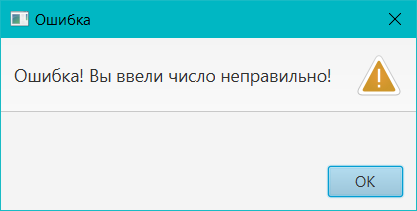


Рисунок 10.8 – Сообщение об ошибке

Если же данные были введены правильно, то открывается окно с выводом результата расчета (Рисунок 10.9).

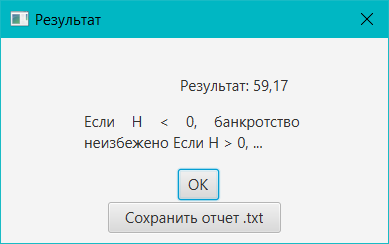


Рисунок 10.9 – Окно вывода результата расчета

При нажатии кнопки «Сохранить отчет .txt» приложение сохраняет отчет о расчете в текстовом файле. Далее пользователь переходит в окно просмотра отчетов. Также в него можно перейти и при нажатии кнопки «Просмотр отчетов» в меню пользователя (Рисунок 10.10).

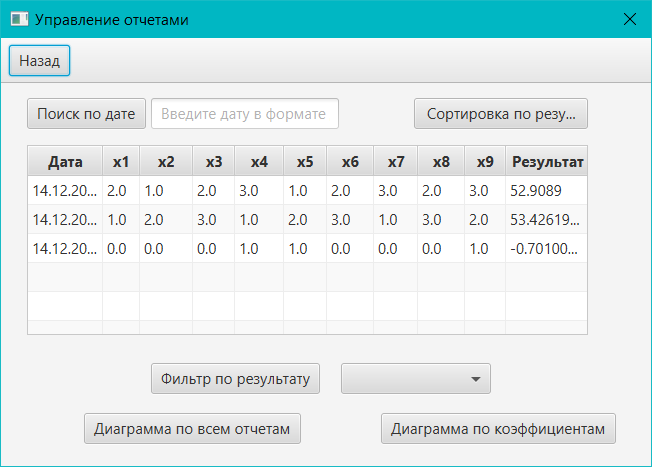


Рисунок 10.10 – Окно просмотра отчетов

Окно просмотра отчетов имеет несколько функций: поиск по введенной дате, сортировка по результату, фильтрация по результату расчета и вывод на экран 2 видов диаграмм.

При вводе в текстовое поле даты в формате дд.мм.гггг и нажатии на кнопку «Поиск по дате», в таблице отображаются только те строки, которые были добавлены в тот день, который был введен в текстовом поле (Рисунок 10.11).

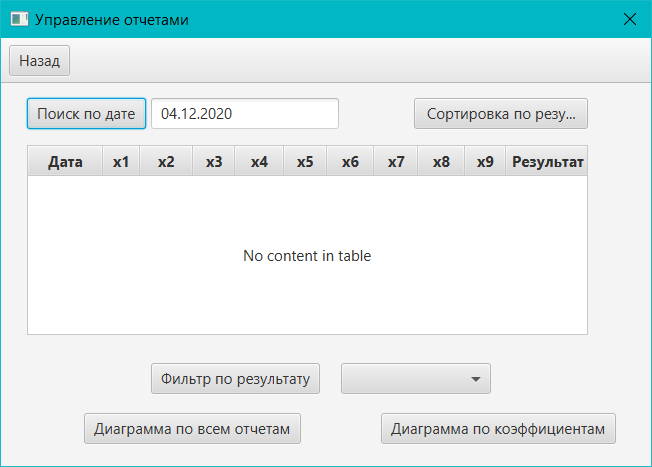


Рисунок 10.11 – Результат поиска по дате

Для фильтрации данных нужно выбрать из выпадающего списка пункт и нажать на кнопку «Фильтр по результату». В таблице отображены только те строки, которые соответствуют параметрам фильтрации (Рисунок 10.12).

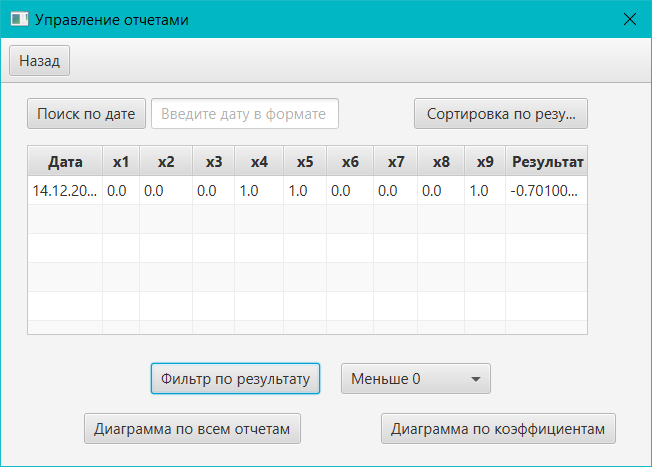


Рисунок 10.12 – Результат выполнения фильтрации данных

Для сортировки достаточно нажать кнопку «Сортировка по результату» и в таблице отобразится результат (Рисунок 10.13).

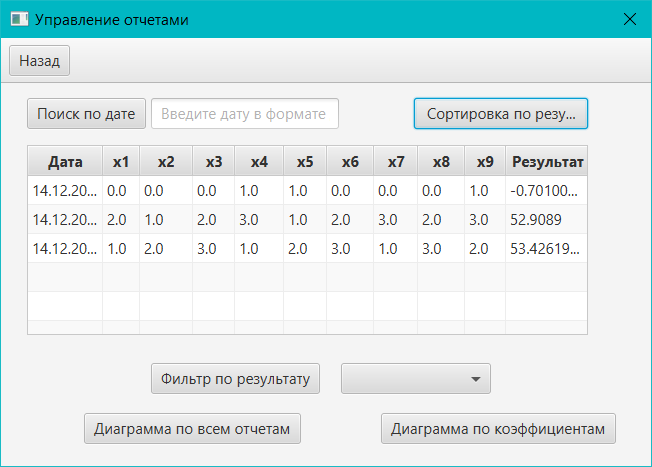


Рисунок 10.13 – Результат выполнения сортировки данных

При нажатии на кнопку «Диаграмма по всем отчетам» открывается новое окно, в котором отображается диаграмма изменения значения результата расчета (Рисунок 10.14).

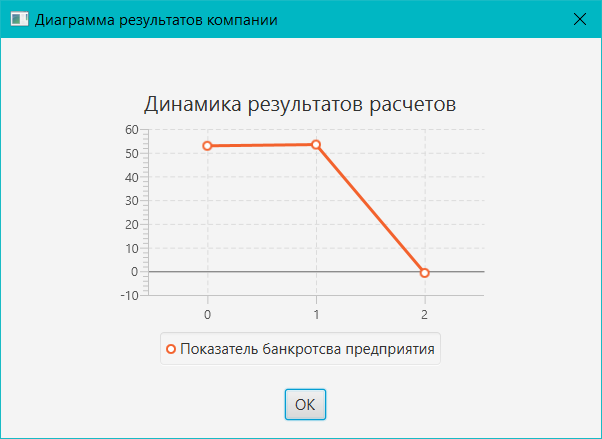


Рисунок 10.14 – Окно диаграммы по всем отчетам

При выборе строки таблицы и нажатии на кнопку «Диаграмма по коэффициентам» открывается новое окно, в котором отображается диаграмма, показывающая изменение значений коэффициентов в выбранном отчете (Рисунок 10.15).

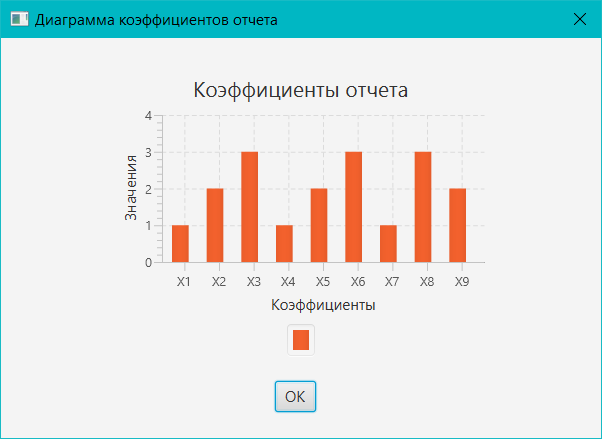


Рисунок 10.15 – Окно диаграммы по коэффициентам выбранного отчета

По нажатию кнопки «Выход» в меню пользователя возвращаемся в меню первого порядка. При нажатии на кнопку «Авторизация» открывается окно авторизации (Рисунок 10.16).

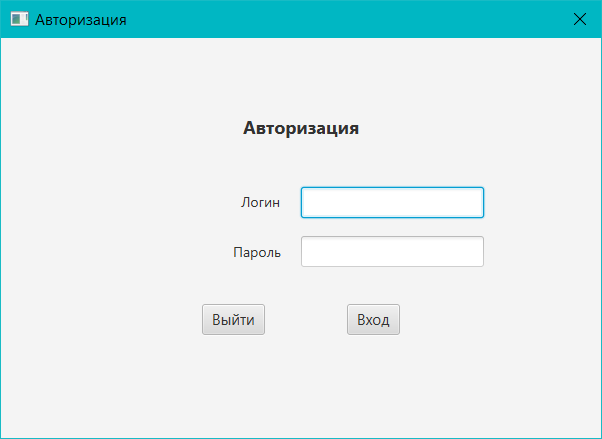


Рисунок 10.16 – Окно авторизации

Если данные введены неверно или же такой пользователь не зарегистрирован, приложение выводит соответствующую ошибку (Рисунок 10.17).

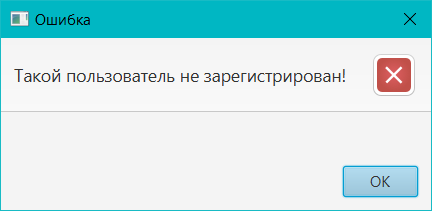


Рисунок 10.17 – Сообщение об ошибке авторизации

В противном случае открывается окно пользователя или администратора в соответствии с введенными данными.

Рассмотрим меню администратора (Рисунок 10.18).

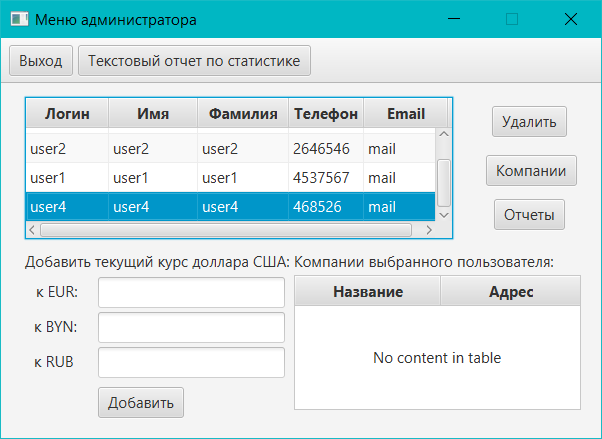


Рисунок 10.18 – Окно меню администратора

При нажатии кнопки «Текстовый отчет по статистике» приложение создает новый текстовый файл, в котором прописывается количество пользователей, количество компаний и количество отчетов, существующих в базе данных.

В таблице пользователей выведена информация о всех зарегистрированных в системе пользователях. При выборе строки таблицы и нажатии на кнопку «Удалить» пользователь будет удален из системы (Рисунок 10.19).

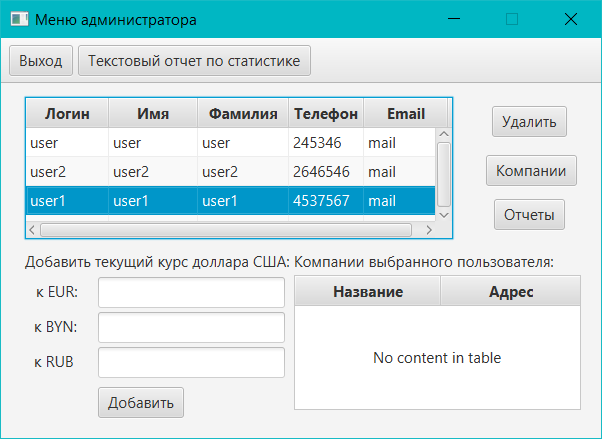


Рисунок 10.19 – Успешное удаление пользователя

При выборе строки таблицы пользователей и нажатии на кнопку «Компании» в таблице компаний будут отображены все компании, которые были добавлены выбранным пользователем (Рисунок 10.20).

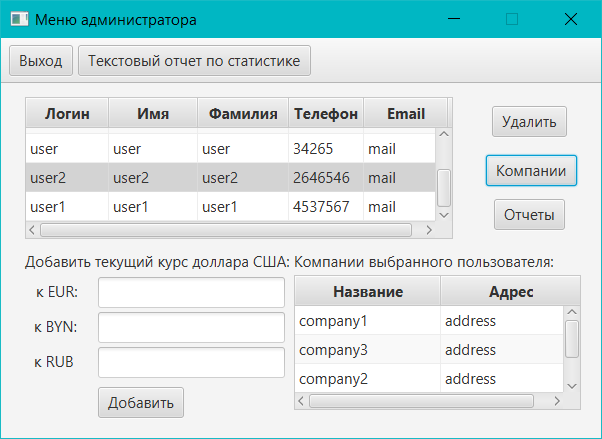


Рисунок 10.20 – Вывод компаний выбранного пользователя

При выборе строки таблицы компаний и нажатии на кнопку «Отчеты» открывается окно «Просмотр отчетов» для выбранной компании, функционал которого был описан выше (см. выше).

Чтобы добавить курс валют нужно ввести необходимые данные в текстовые поля и нажать кнопку «Добавить». При неправильно введенных данных приложение выводит сообщение об ошибке (Рисунок 10.21).

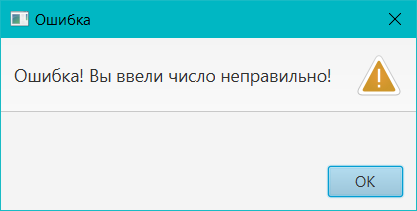


Рисунок 10.21 – Сообщение об ошибке

В противном случае происходит добавление данных в базу данных, при этом проставляется дата добавления строки таблицы (Рисунок 10.22).

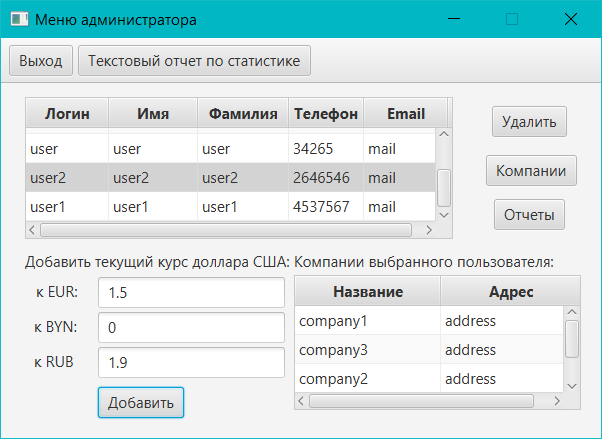


Рисунок 10.22 – Успешное добавление курса валют

# ЗАКЛЮЧЕНИЕ

В ходе данной работы был изучен процесс прогнозирования платежеспособности предприятия на основе модели банкротства Фулмера. Также были сделаны выводы о важности прогнозирования на основе экономических показателей предприятия как основа для статистических данных предприятия. Можно выделить важной составляющей работу аналитиков по подсчету и сбору данных для дальнейшего расчета результата прогнозирования.

Целью курсового проекта являлось сокращение человеческого и временного ресурса предприятия при обработке данных для подсчета результата прогнозирования платежеспособности предприятия путем автоматизации данного процесса. Для этого были разработаны функциональная модель основного процесса, основные алгоритмы и функции для работы с данными, а также создан программный продукт и описание к нему. Таким образом, поставленная цель была достигнута, а задачи выполнены.

Были закреплены знания в области объектно-ориентированного программирования и паттернах проектирования. Также был изучен инструментарий JavaFX и работа с ним. Изучена модель банкротства Фулмера.

Итогом данного проекта является функционирующее клиент-серверное приложение, имеющее соединение с базой данных и позволяющее прогнозировать платежеспособность предприятия на основе модели банкротства Фулмера.

Разработанную программу в перспективе можно модифицировать, расширив возможности работы с пользователями, добавив новые возможности для администратора.

# список использованных источников

[1] Экономический портал Российской федерации [Электронный ресурс] // URL: [https://economy-ru](https://economy-ru.) (дата обращения: 07.10.2020)

[2] Портал Киберленинка [Электронный ресурс] // URL: https://cyberleninka.ru (дата обращения: 07.10.2020)

[3] Портал ИТ Аудит [Электронный ресурс] // URL:  [https://www.audit-it.ru](%20https://www.audit-it.ru) (дата обращения: 10.10.2020)

[4] Портал ЦФМТ [Электронный ресурс] // URL: http://www.cfmt.ru/ (дата обращения: 10.10.2020)

[7] Википедия [Электронный ресурс] // URL: <https://ru.wikipedia.org> (дата обращения: 13.11.2020)

[8] Экономический портал ФинЗнания [Электронный ресурс] // URL: https://finzz.ru (дата обращения: 14.11.2020)

[9] Электронный портал Метанит [Электронный ресурс] // URL: https://metanit.com (дата обращения: 14.11.2020)

[10] Электронный форум СитФорум [Электронный ресурс] // URL: http://citforum.ru (дата обращения: 09.12.2020)

[11] Электронный форум МетанитФорум [Электронный ресурс] // URL: http://metanitforum.ru (дата обращения: 13.12.2020)

[12] Электронный портал Вики Ела [Электронный ресурс] // URL: https://wiki.yola.ru (дата обращения: 13.12.2020)

# ПРИЛОЖЕНИЕ А

# (обязательное)

# Диаграммы развертывания, компонентов, последовательности состояний и классов (к разделу 6)

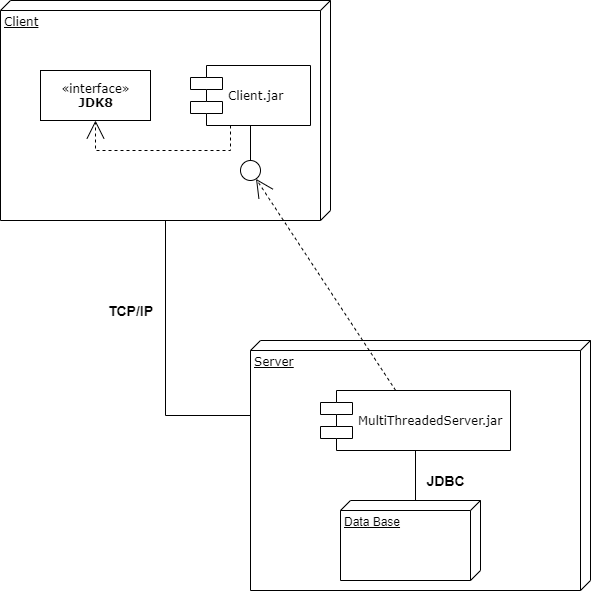
****

Рисунок А.1 – Диаграмма развертывания системы прогнозирования платежеспособности предприятия методом Фулмера

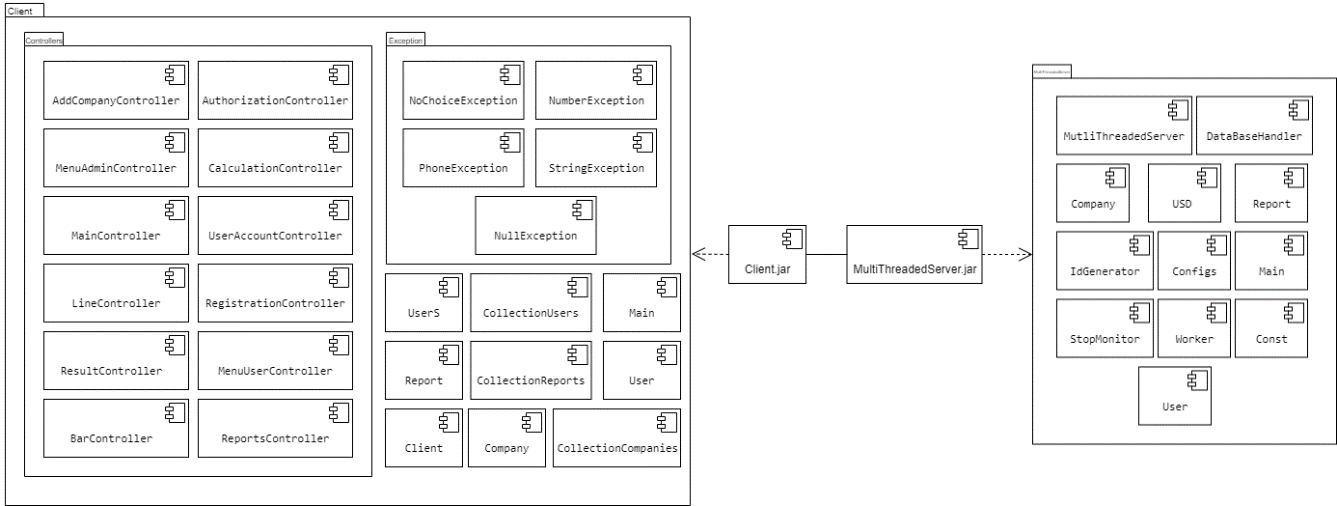
****

Рисунок А.2 – Диаграмма компонентов системы расчета коэффициента банкротства модели Фулмера

Продолжение приложения А

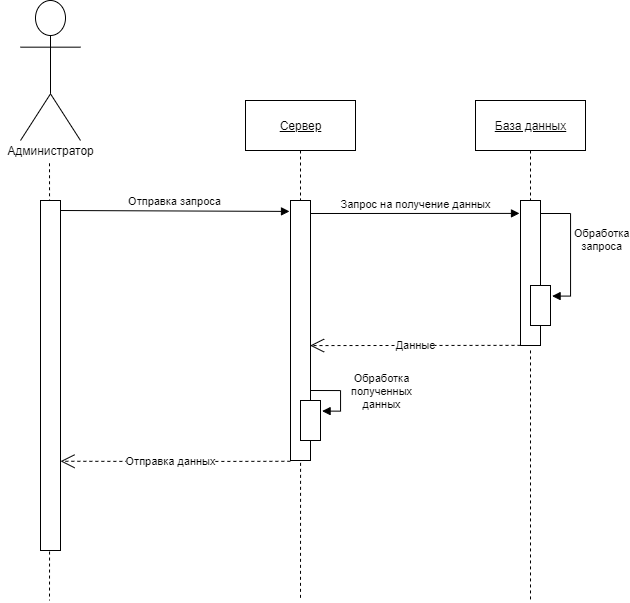


Рисунок А.3 – Диаграмма последовательности просмотра статистики администратором

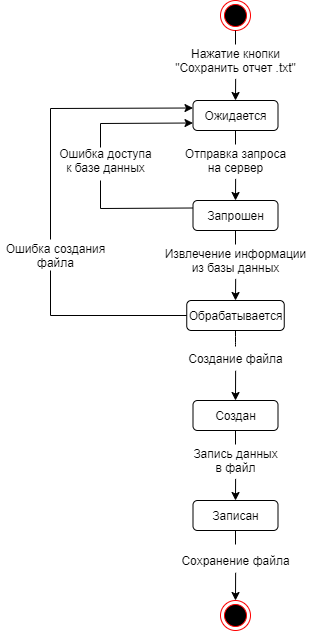


Рисунок А.4 – Диаграмма состояний отчета о расчете прогноза платежеспособности предприятия методом Фулмера

Продолжение приложения А

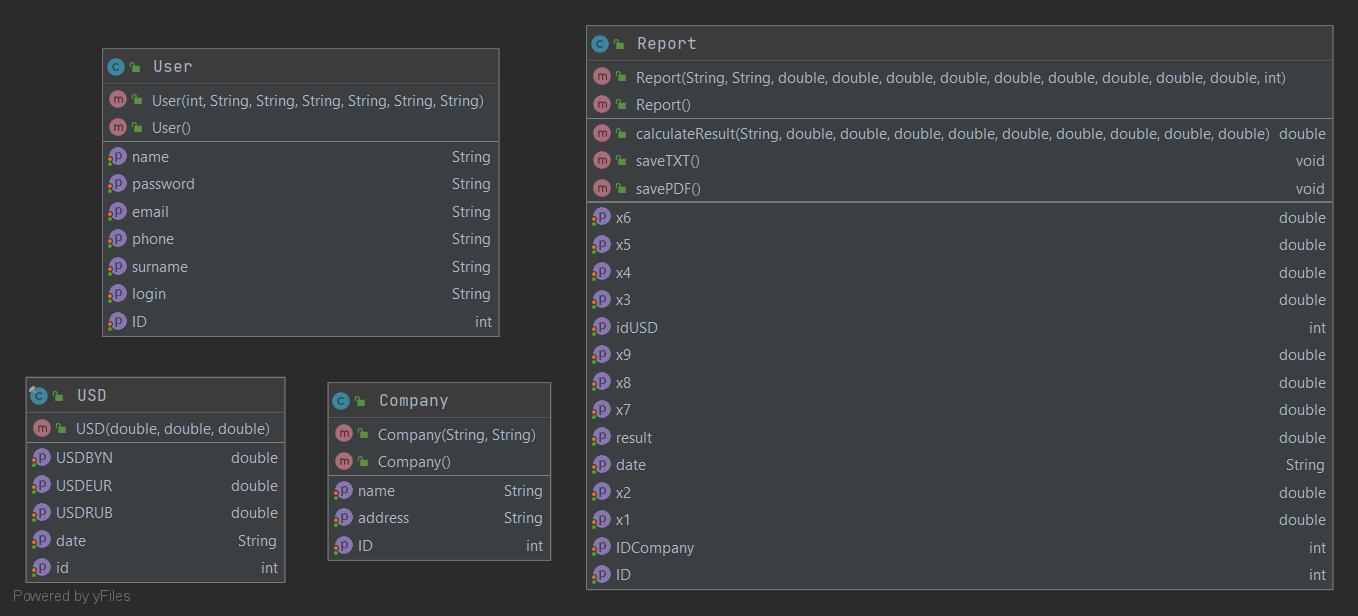


Рисунок А.5 – Диаграмма классов сущностей базы данных

Продолжение приложения А

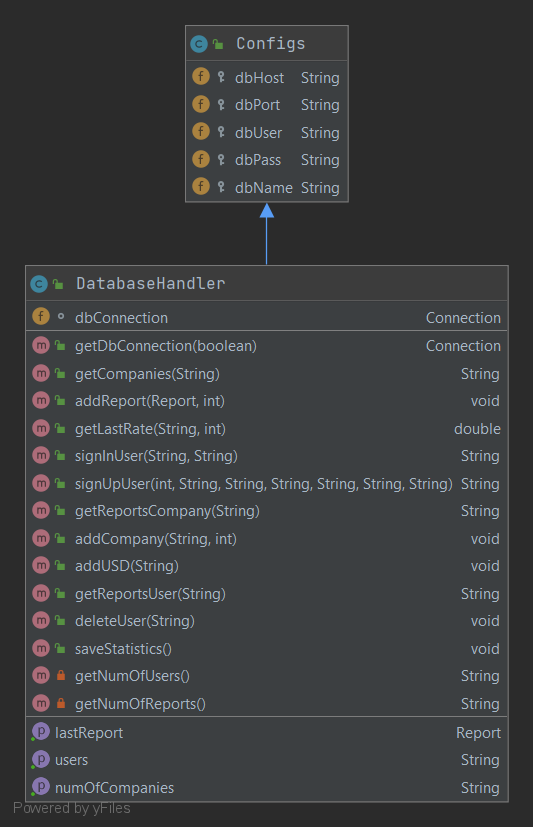


Рисунок А.6 – Диаграмма классов, взаимодействующих с базой данных

Продолжение приложения А

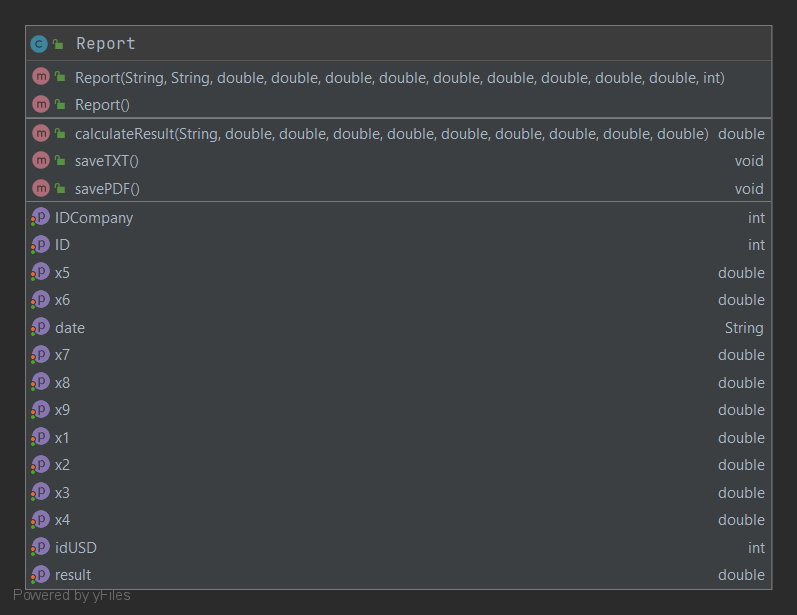


Рисунок А.7 – Диаграмма класса, реализующего бизнес-логику

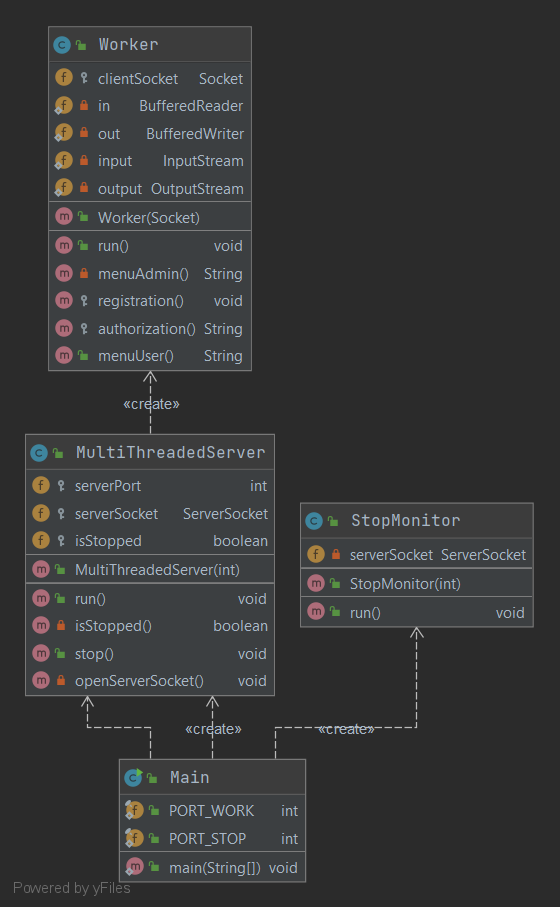


Рисунок А.8 – Диаграмма классов серверной части приложения

Продолжение приложения А

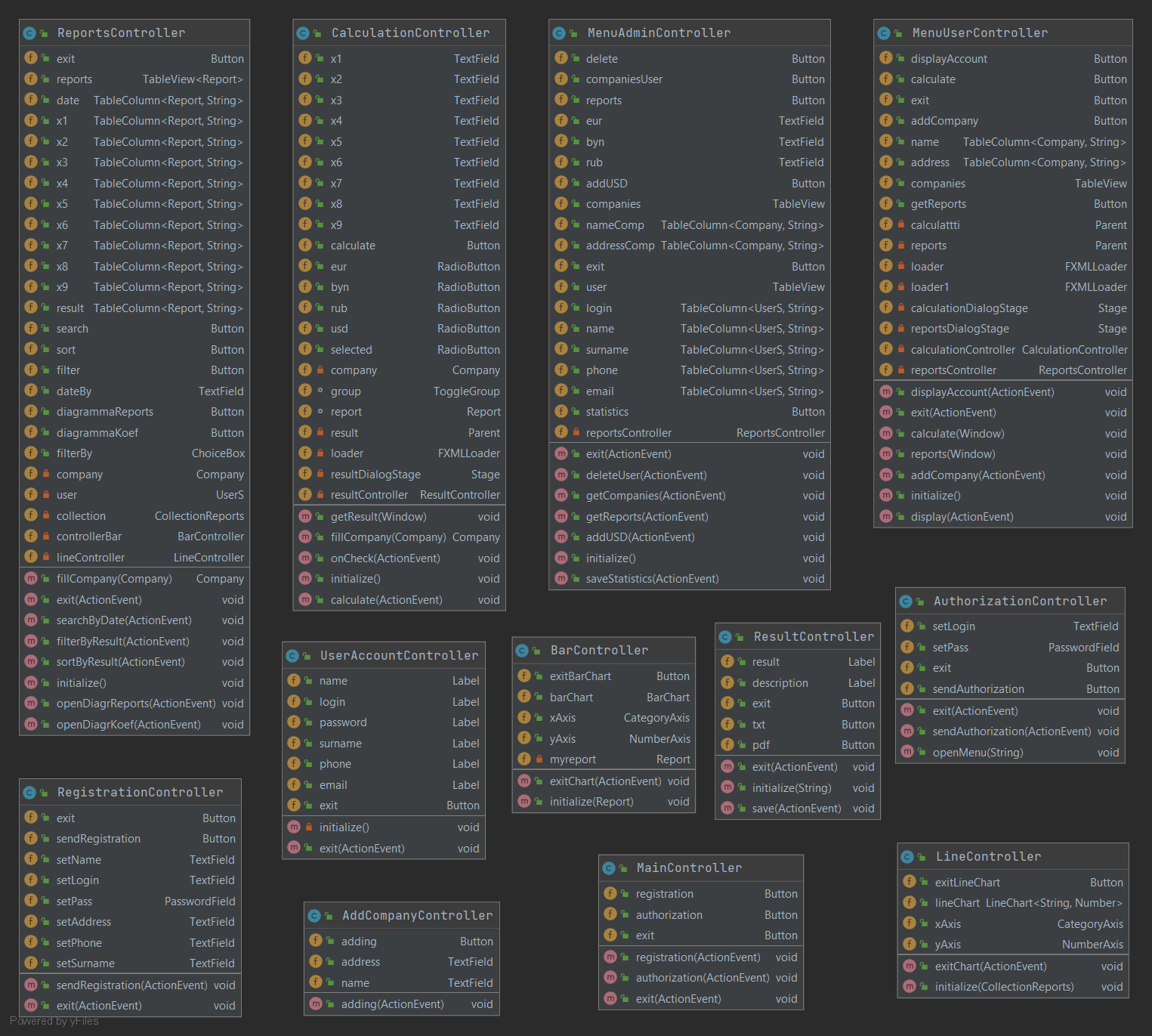


Рисунок А.9 – Диаграмма сущностей контроллеров диалоговых окон

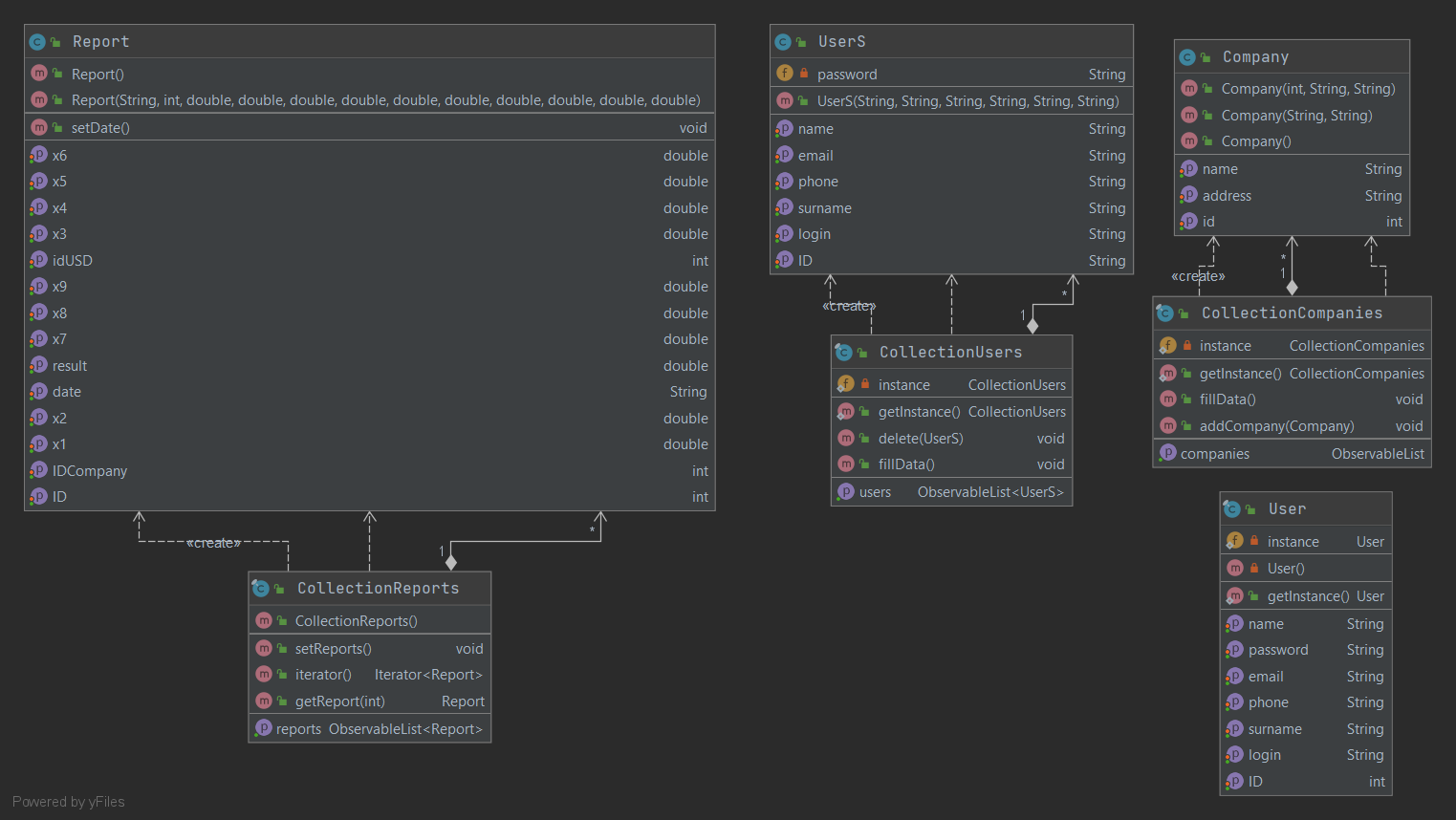


Рисунок А.10 – Диаграмма классов, соответствующих классам на сервере

Продолжение приложения А

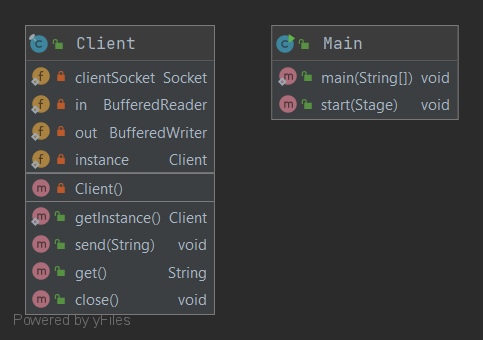


Рисунок А.11 – Диаграмма классов клиентской части приложения

# ПРИЛОЖЕНИЕ Б

# (обязательное)

# Скрипт генерации базы данных (к разделу 9)

create table if not exists user(

IDUser int primary key,

Surname varchar(20) not null,

Name varchar(20) not null,

Email varchar(20) not null,

Phone varchar(20) not null,

Login varchar(20) not null,

Password varchar(20) not null

);

create table if not exists Report(

IDReport int primary key,

Date varchar(20) not null,

IDRate int not null,

X1 double not null,

X2 double not null,

X3 double not null,

X4 double not null,

X5 double not null,

X6 double not null,

X7 double not null,

X8 double not null,

X9 double not null,

H double not null,

foreign key(IDRate) references Rate(IDRate),

IDCompany int not null

);

create table if not exists Company(

IDCompany int primary key,

Name varchar(20) not null,

Address varchar(20) not null

);

create table if not exists Rate(

BYNUSD int not null,

EURUSD int not null,

RUBUSD int not null,

IDRate int primary key,

Продолжение приложения Б

Date varchar(20) not null

);

create table if not exists UserCompany(

IDUser int not null,

IDCompany int not null,

foreign key(IDUser) references User(IDUser),

foreign key(IDCompany) references Company(IDCompany)

);

create table if not exists UserReport(

IDUser int not null,

IDReport int not null,

foreign key(IDUser) references User(IDUser),

foreign key(IDReport) references Report(IDReport)

);

# ПРИЛОЖЕНИЕ В

# (обязательное)

# Листинг алгоритмов, реализующих бизнес-логику

public class Report {

private String date;

private int ID;

private double x1, x2, x3, x4, x5, x6, x7, x8, x9;

private double result;

private int IdUSD;

private int IDCompany;

public Report(String date,String type, double x1, double x2, double x3, double x4, double x5,

double x6, double x7, double x8, double x9, int IDCompany ) {

this.date = date;

this.ID = IdGenerator.getInstance( "report" ).getNextId();

this.x1 = x1;

this.x2 = x2;

this.x3 = x3;

this.x4 = x4;

this.x5 = x5;

this.x6 = x6;

this.x7 = x7;

this.x8 = x8;

this.x9 = x9;

this.result = calculateResult(type, x1, x2, x3, x4, x5, x6, x7, x8, x9);

this.IdUSD = IdGenerator.getInstance( "usd" ).getNextId();

this.IDCompany = IDCompany;

}

public Report(){};

public double calculateResult(String type, double x1, double x2, double x3, double x4, double x5, double x6, double x7, double x8, double x9) {

DatabaseHandler handler = new DatabaseHandler();

double coef = 1;

if(type.equals( "byn" )) {

coef = 2.8;

} else if(type.equals( "eur" )) {

coef = 2.1;

} else if(type.equals( "rub" )) {

coef = 2.4;

Продолжение приложения В

} else coef = 1;

double result = 5.529\*( x1\*coef ) + 0.212\*( coef\*x2 ) + 0.073\*(coef\* x3 )+

1.27\*(coef \* x4 ) + 0.12\*(coef\* x5 ) +2.235\*(coef\* x6 )+ 0.575\*(coef\*x7 )+

1.083\*( coef\* x8 ) + 0.984\*(coef\*x9 ) - 3.075;

return result;

}

public String getDate() {

return date;

}

public void setDate(String date) {

this.date = date;

}

public int getID() {

return ID;

}

public void setID(int ID) {

this.ID = ID;

}

public double getX1() {

return x1;

}

public void setX1(double x1) {

this.x1 = x1;

}

public double getX2() {

return x2;

}

public void setX2(double x2) {

this.x2 = x2;

}

public double getX3() {

Продолжение приложения В

return x3;

}

public void setX3(double x3) {

this.x3 = x3;

}

public double getX4() {

return x4;

}

public void setX4(double x4) {

this.x4 = x4;

}

public double getX5() {

return x5;

}

public void setX5(double x5) {

this.x5 = x5;

}

public double getX6() {

return x6;

}

public void setX6(double x6) {

this.x6 = x6;

}

public double getX7() {

return x7;

}

public void setX7(double x7) {

this.x7 = x7;

}

public double getX8() {

return x8;

}

Продолжение приложения В

public void setX8(double x8) {

this.x8 = x8;

}

public double getX9() {

return x9;

}

public void setX9(double x9) {

this.x9 = x9;

}

public String getResult() {

return Double.toString( result );

}

public void setResult(double result) {

this.result = result;

}

public int getIdUSD() {

return (IdUSD-2);

}

public void setIdUSD(int idUSD) {

IdUSD = idUSD;

}

public int getIDCompany() {

return IDCompany;

}

public void setIDCompany(int IDCompany) {

this.IDCompany = IDCompany;

}

public void saveTXT() {

File file = new File("Отчет "+ date +" " +ID+".txt");

try {

if (!file.exists()) {

file.createNewFile();

FileWriter add = new FileWriter(file, true);

Продолжение приложения В

add.write("Сгенерированный отчет №"+ID+'\n');

add.write( "X1 = "+ x1+'\n');

add.write( "X1 = "+ x2+'\n');

add.write( "X1 = "+ x3+'\n');

add.write( "X1 = "+ x4+'\n');

add.write( "X1 = "+ x5+'\n');

add.write( "X1 = "+ x6+'\n');

add.write( "X1 = "+ x7+'\n');

add.write( "X1 = "+ x8+'\n');

add.write( "X1 = "+ x9+'\n');

add.write( "Результат = "+ result + '\n' );

add.flush();

}

} catch (IOException e) {

e.printStackTrace();

}

System.out.println( "Сохранил отчет " );

}

public void savePDF() throws IOException, COSVisitorException {

}

}

# ПРИЛОЖЕНИЕ Г

# (обязательное)

# Листинг основных элементов программы

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import org.apache.ibatis.jdbc.ScriptRunner;

import java.io.\*;

import java.sql.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

public class DatabaseHandler extends Configs {

Connection dbConnection;

public Connection getDbConnection(boolean flag)

throws ClassNotFoundException, SQLException, FileNotFoundException {

String connectionString = "jdbc:mysql://" + dbHost + ":"

+ dbPort + "/" + dbName +"?useUnicode=true&serverTimezone=UTC&allowMultiQueries=true";

Class.forName("com.mysql.cj.jdbc.Driver");

dbConnection = DriverManager.getConnection(connectionString,

dbUser, dbPass);

if(flag) {

System.out.println("Генерация скрипта базы данных:");

ScriptRunner runner = new ScriptRunner(dbConnection);

runner.runScript(new BufferedReader(new FileReader("course.sql")));

}

return dbConnection;

}

public String getNumOfCompanies(){

ResultSet rs = null;

int count =0;

try {

String select = "SELECT COUNT(1) FROM "+ Const.COMPANY\_TABLE;

PreparedStatement prep = getDbConnection(false).prepareStatement(select);

rs = prep.executeQuery();

while (rs.next()) {

count = rs.getInt( 1 );

}

} catch (SQLException e) {

e.printStackTrace();

}

catch (ClassNotFoundException e) {

e.printStackTrace();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

return String.valueOf(count);

}

public String getCompanies(String userId){

Company company1;

JSONObject companyJson;

JSONArray companies = new JSONArray( );

try {

String selectFirst = "SELECT "+ Const.USERCOMPANY\_COMPANY +" FROM "+Const.USERCOMPANY\_TABLE +" WHERE "+ Const.USERCOMPANY\_USER+

" = "+userId;

PreparedStatement prep1 = getDbConnection(false).prepareStatement( selectFirst );

ResultSet rs1 = prep1.executeQuery();

while (rs1.next()){

int id = rs1.getInt( "IDCompany" );

String select = "SELECT \* FROM course\_schema.company WHERE IDCompany=" + id;

PreparedStatement prep = getDbConnection(false).prepareStatement( select);

ResultSet rs = prep.executeQuery();

while (rs.next()) {

company1 = new Company();

company1.setID( rs.getInt( "IDCompany" ) );

company1.setName( rs.getString( "Name" ) );

company1.setAddress( rs.getString( "Address" ) );

companyJson = new JSONObject();

companyJson.put( "id", company1.getID() );

companyJson.put( "name", company1.getName() );

companyJson.put( "address", company1.getAddress() );

companies.put( companyJson );

}

}

} catch (SQLException | ClassNotFoundException | JSONException | FileNotFoundException e) {

e.printStackTrace();

}

return companies.toString();

}

public void addReport(Report report, int IDUser) throws ClassNotFoundException, SQLException, JSONException, FileNotFoundException {

addUSD("{eur:2.1,byn:2.3,rub:2.4}");

String insert = "INSERT INTO "+ Const.REPORT\_TABLE +"("+

Const.REPORTS\_ID+","+ Const.REPORTS\_DATE +"," +

Const.REPORTS\_IDUSD+","+ Const.X1+","+

Const.X2+","+Const.X3+","+Const.X4+","+Const.X5+","+

Const.X6+","+ Const.X7+","+Const.X8+","+Const.X9+","+

Const.REPORTS\_RESULT +"," + Const.REPORTS\_IDCOMPANY +")"+

"VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

try {

PreparedStatement prSt = getDbConnection(false).prepareStatement(insert);

prSt.setInt(1, report.getID());

prSt.setString(2, report.getDate());

prSt.setInt(3, 8);

prSt.setDouble(4, report.getX1());

prSt.setDouble(5, report.getX2());

prSt.setDouble(6, report.getX3());

prSt.setDouble(7, report.getX4());

prSt.setDouble(8, report.getX5());

prSt.setDouble(9, report.getX6());

prSt.setDouble(10, report.getX7());

prSt.setDouble(11, report.getX8());

prSt.setDouble(12, report.getX9());

prSt.setString(13, report.getResult());

prSt.setInt(14, report.getIDCompany());

prSt.executeUpdate();

String insert2 = "INSERT INTO "+ Const.USERREPORT\_TABLE+"("+

Const.USERREPORT\_IDUSER+","+Const.USERREPORT\_IDREPORT+")"+

"VALUES(?,?)";

PreparedStatement prep = getDbConnection(false).prepareStatement( insert2 );

prep.setInt( 1, IDUser );

prep.setInt( 2, report.getID() );

prep.executeUpdate();

} catch (SQLException | ClassNotFoundException | FileNotFoundException e) {

e.printStackTrace();

}

}

public double getLastRate(String type, int id){

String name = null;

switch (type){

case "byn": {name = "BYNUSD"; break;}

case "ryb": {name = "RUBUSD"; break;}

case "eur": {name = "EURUSD"; break;}

}

double rate = 0;

String select = "SELECT "+name +" FROM " + Const.USD\_TABLE+ " WHERE "+ Const.USD\_ID+" = " +id;

try {

PreparedStatement prep = getDbConnection(false).prepareStatement(select);

ResultSet rs = prep.executeQuery();

while (rs.next()){

rate = rs.getDouble( 1 );

}

} catch (SQLException | ClassNotFoundException | FileNotFoundException e) {

e.printStackTrace();}

}