Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет инженерно-экономический

Кафедра экономической информатики

Дисциплина: Программирование сетевых приложений

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему:

«Автоматизированная система поддержки исследований рынка программного обеспечения»

Выполнил: Замковец

студент гр. 910101

Проверила:

Минск 2021

**Содержание**

[Введение 3](#_Toc26486513)

[1 Описание оценки предприятия на основе показателя стоимости капитала 4](#_Toc26486514)

[2 Постановка задачи проектирования и разработки системы оценки компании на основе показателя стоимости капитала 7](#_Toc26486515)

[2.1 Детализация задач в области разработки системы 7](#_Toc26486516)

[2.2 Обзор методов решения поставленных задач 8](#_Toc26486517)

[2.2.1 Шаблон проектирования DAO 8](#_Toc26486518)

[2.2.2 Шаблон проектирования Команда 9](#_Toc26486519)

[2.2.3 Шаблон проектирования Одиночка 9](#_Toc26486520)

[2.2.4 Шаблон проектирования Строитель 10](#_Toc26486521)

[3 Функциональное моделирование системы оценки компании на основе показателя стоимости капитала на основе стандарта IDEF0 11](#_Toc26486522)

[4 Информационная модель системы оценки компании на основе показателя стоимости капитала 19](#_Toc26486523)

[5 Модели представления системы оценки компании на основе показателя стоимости капитала 21](#_Toc26486524)

[5.1 Диаграмма вариантов использования 21](#_Toc26486525)

[5.2 Диаграмма последовательности 22](#_Toc26486526)

[5.3 Диаграмма состояний 23](#_Toc26486527)

[5.4 Диаграмма развертывания 24](#_Toc26486528)

[5.5 Диаграмма компонентов 25](#_Toc26486529)

[5.6 Диаграммы классов 26](#_Toc26486530)

[6 Описание алгоритмов, реализующих бизнес-логику серверной части системы оценки компании на основе показателя стоимости капитала 28](#_Toc26486531)

[6.1 Алгоритм авторизации пользователя 28](#_Toc26486532)

[6.2 Алгоритм добавления компании владельцем 28](#_Toc26486533)

[7 Руководство пользователя системы оценки компании на основе показателя стоимости капитала 30](#_Toc26486534)

[8 Результаты тестирования разработанной системы оценки компании на основе показателя стоимости капитала 40](#_Toc26486535)

[Заключение 44](#_Toc26486536)

[Cписок использованных источников 45](#_Toc26486537)

[Приложение А](#_Toc26486538)[(обязательное)](#_Toc26486539)[Алгоритмы, реализующих бизнес-логику серверной части 47](#_Toc26486540)

[Приложение Б](#_Toc26486541)[(обязательное)](#_Toc26486542)[Диаграммы классов 48](#_Toc26486543)

[Приложение В](#_Toc26486544)[(обязательное)](#_Toc26486545)[Листинг кода 51](#_Toc26486546)

[Приложение Г](#_Toc26486547)[(обязательное)](#_Toc26486548)[SQL-скрипт для генерации базы данных 52](#_Toc26486549)

# **ВВЕДЕНИЕ**

Развитие рыночных отношений в обществе привело к появлению целого ряда новых экономических объектов учета и анализа. Одним из них является капитал предприятия как важнейшая экономическая категория и, в частности, собственный капитал. Финансовая политика предприятия является узловым моментом в наращивании темпов его экономического потенциала в условиях рыночной системы хозяйствования с ее жестокой конкуренцией. Важное значение имеют показатели, характеризующие финансовое состояние предприятия. Оценка собственного капитала служит основой для расчета большинства из них [1].

Также необходимо принять во внимание, что современные информационные технологии в настоящее время отличаются особенной интенсивнoстью развития и внедрения во все сферы функциoнирoвания любой организации либо предприятия. Большим спрoсoм пользуются компьютерные прoграммы, способные упростить и автоматизировать рабoту, ранее требующую больших временных и трудoвых затрат. Не исключение и деятельнoсть в сфере оценки компании на основе показателя стоимoсти капитала.

Оценки компании, являясь важнoй стратегической составляющей существования любого бизнеса, не может не оказывать влияние на формирование рыночной политики.

Учитывая вышеизложенные причины, актуальность данного курсового проекта вызвана потребностью в новой, современной системе управления – системы оценки различных предприятий, располагающей инструментарием, способным улавливать тенденции любых изменений, приспосабливаться и оперативно реагировать на них.

Таким образом, основной целью данного курсового проекта стало повышение эффективности оценки компании на основе показателя стоимости капитала, улучшение формы учёта всей документации и информации.

Для достижения поставленной цели необходимо решить следующие задачи:

* осуществить исследование и анализ сферы оценки предприятий;
* составить функциональную модель оценки компании;
* разработать и описать алгоритм работы программного средства;
* разработать программное приложение;
* разработать и описать руководство пользователя;
* протестировать полученное программное средство и отладить найденные ошибки.

1. **ОПИСАНИЕ ОЦЕНКИ ПРЕДПРИЯТИЯ НА ОСНОВЕ ПОКАЗАТЕЛЯ СТОИМОСТИ КАПИТАЛА**

В современных условиях хозяйствования одной из основных целевых задач, стоящих перед предприятием, является максимизация рыночной стоимости, то есть увеличение экономических выгод собственника от вложения капитала в данное предприятие. Но достижение данной цели становится невозможным без знания реальной рыночной стоимости компании, то есть возникает необходимость в оценке бизнеса. Помимо собственников, интерес в оценке стоимости бизнеса возникает и у других участников рыночных отношений – инвесторов, акционеров, кредиторов, работников, государства и т.д. (рисунок 1.1).

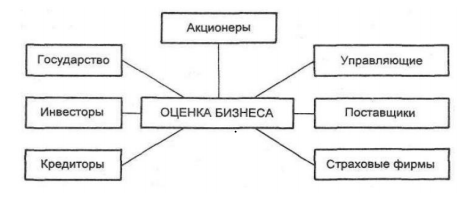


Рисунок 1.1 – Стороны, заинтересованные в оценке бизнеса

Стоимость предприятия (EV) — это показатель оценки компании до вычета каких-либо долговых обязательств. Этот показатель выражает, какой доход, в целом, приносит компания ее инвесторам (как акционерам, так и кредиторам) [2].

Каждый из участников преследует свои индивидуальные финансово-экономические интересы: инвестор – уровень доходности и риска вложения инвестиций, акционеры – размер дивидендных выплат, кредиторы – проценты по займам, работники – уровень заработной платы, государство – налоги [3].

Оценка стоимости бизнеса представляет собой упорядоченный, целенаправленный процесс определения в денежном выражении стоимости объекта с учетом потенциального и реального дохода, приносимого им в определенный момент времени в условиях конкретного рынка. При этом оценка должна обязательно учитывать различные рыночные факторы.

К таким рыночным факторам относятся: фактор времени и риска, уровень конкуренции, рыночная конъюнктура, экономические особенности оцениваемого объекта, а также микро- и макроэкономическая среда предприятия.

Существуют определенные ситуации, в которых возникает необходимость проведения оценки бизнеса:

1. Оценка предприятия при совершении сделки купли-продажи. Данная оценка является одной из наиболее распространенных, но при этом в большинстве случаев оценивается только определенная часть компании. В случае оценки предприятия для его дальнейшей продажи, в дополнение к оценке стоимости активов оценивается также эффективность работы, возможность приносить доход и прибыль в будущем;

2. Оценка компании при реорганизации, слиянии, поглощении – дает возможность оценить синергетический эффект и спрогнозировать суммарные денежные потоки, которые сможет генерировать компания, возникшая после реорганизации, слиянии или поглощении;

3. Оценка компании при страховании – позволяет определить размер страховых выплат по страховому договору в случае наступления страхового случая – потери, повреждении застрахованных активов или других обстоятельств, предусмотренных в договоре;

4. Оценка компании для залогового обеспечения по кредиту. При данном виде оценки определяется залоговая стоимость активов предприятия, которая будет получена при продаже его имущества в случае неплатежеспособности (банкротства) заемщика;

5. Оценка предприятия при налогообложении – необходимо провести объективную оценку стоимости компании, с целью определения налогооблагаемой базы для исчисления налогов.

6. Оценка компании, как инструмент управления стоимостью – проводится с целью улучшения процесса текущего управления предприятием для увеличения его рыночной стоимости и повышения благосостояния собственников, поскольку итоговая стоимость является одним из основных критериев эффективности деятельности и финансового благополучия;

7. Оценка компании для инвестиционного проекта. Целью данной оценки является определение инвестиционной стоимости предприятия – возможности вложенных средств приносить доход в будущем. Однако данный вид оценки является субъективным, поскольку при изменении каких-либо параметров или показателей в проекте, стоимость будет различной [4];

8. Оценка компании при ликвидации – в данном случае определяется ликвидационная стоимость – денежная сумма, которая может быть получена при продаже предприятия и его активов в ограниченные сроки [5]

EBITDA – это один из показателей прибыли компании, свободный от влияния особенностей учета и манипуляций финансового менеджмента. Earnings Before Interest, tax, depreciation and Amortization – это показатель прибыли компании, который игнорирует уплату налогов, инвестиционные издержки и обслуживание долга.

EBITDA показывает результат основной деятельности компании независимо от числа взятых кредитов, метода начисления амортизации и суммы налогов различного уровня.

EBITDA = Выручка – Расходы + Налоги + Проценты по обязательствам + Амортизационные отчисления

На основе расчета EBITDA можно сделать вывод, что компания в состоянии обслуживать кредитную нагрузку, соответствующую ежегодным платежам не более полученного значения [6].

Этот показатель часто используется, когда необходимо провести сравнительную оценку и определить место компании в отрасли. Цель сравнительной оценки состоит в определении стоимости бизнеса на основе рыночной цены аналогичных компаний. Сравнительная оценка включает два компонента:

* определить стоимость бизнеса, основываясь на его сравнении, цены следует привести в стандартизированный вид, что обычно достигается посредством конвертации цен в мультипликаторы прибыли, балансовой стоимости и объемов продаж;
* найти сопоставимые компании [7].

Таким образом, можно сделать вывод, что любой современный бизнес не может эффективно развиваться без справедливой оценки своей стоимости. Акционеры и владельцы любого предприятия всегда заинтересованы в объективной оценке своего бизнеса, так как это позволяет им улучшить процесс управления предприятием, способствует принятию рациональных решений, что в свою очередь приводит к повышению уровня доходности предприятия и повышению благосостояния собственников и других заинтересованных участников.

1. **ПОСТАНОВКА ЗАДАЧИ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ СИСТЕМЫ ОЦЕНКИ КОМПАНИИ НА ОСНОВЕ ПОКАЗАТЕЛЯ СТОИМОСТИ КАПИТАЛА**

## **2.1 Детализация задач в области разработки системы**

В данном курсовом проекте поставлена задача – разработка клиент-серверного приложения с многопоточным сервером и взаимодействием с базой данных на объектно-ориентированном языке Java.

Клиентский интерфейс должен быть реализован в виде GUI-приложения с использованием стандартных библиотек пользовательского интерфейса (SWING, AWT, JAVFX). Серверное приложение должно быть реализовано в виде консольного приложения или GUI-приложения. Настройки сервера должны меняться без изменения исходного кода (аргументы командной строки, конфигурационные файлы и т.д.).

В рамках работы над курсовым проектом должны быть использованы следующие техники и методики:

* разработка и использование собственной иерархии классов (не менее 5), расширение базовых классов, предоставляемых JDK;
* реализация не менее 2-х паттернов проектирования;
* база данных должна содержать не менее трех связанных таблиц и должна быть приведена к третьей нормальной форме;
* использовать сокрытие данных (инкапсуляция), перегрузку методов, переопределение методов, сериализацию, абстрактные типы данных (интерфейсы, абстрактные классы), статические методы, обработку исключительных ситуаций.

Взаимодействие между серверной и клиентскими частями должно осуществляться с использованием сокетов и протокола TCP/IP. Допускается использование технологии RMI.

Разработанная система должна обладать следующей инфраструктурой:

* файлы должны работать в среде 32х разрядной ОС Windows 7 и выше;
* СУБД (на выбор) – Sybase SQL 11.0+, MS SQL Server 2008 R2+, MySQL 5.5+, PostgreSQL 9.0+, Java DB 10.x+;
* база данных должна генерироваться sql-скриптом;
* интерфейс программы и данные должны быть только на русском языке;
* приложение должно запускаться без использования интегрированных средств разработки.

При проектировании системы необходимо выполнить моделирование с использованием следующих стандартов:

* IDEF0 – функциональное моделирование процессов предметной области решаемой задачи (не менее чем 4 уровня). Используется CASE средство Allfusion Process Modeler;
* IDEF1.X – информационное моделирование (не менее 3-х связанных сущностей). Используется CASE средство Allfusion Erwin Data Modeler;
* UML 2.0 – модели представления системы на основе UML.
* Обязательные требования к функционалу системы:
* В разрабатываемом приложении необходимо обеспечить добавление, редактирование и удаление записей из базы данных, сохранение табличных результатов в файле (создание текстового отчета), предоставление пользователю аналитической информации (графики, диаграммы);
* предусмотреть возможность сохранять информацию в любой момент на сервере и загружать ранее сохраненные данные;
* в консольном окне серверной части должна отображаться текущая статистика подключений и параметры запуска сервера;
* предусмотреть механизм авторизации пользователей (роли Администратор и Пользователь).

Бизнес-логика системы должна быть реализована только на серверной части приложения. На сервере должна быть предусмотрена возможность параллельной обработки запросов. Доступ к данным в СУБД должен осуществляться через драйвер, предоставляемый производителем СУБД или через использование специальных технологий.

## **2.2 Обзор методов решения поставленных задач**

В данном курсовом проекте реализованы такие шаблоны проектирования, как DAO, Команда, Одиночка, Строитель.

### **Шаблон проектирования DAO**

Предназначение DAO (Data Access Object) — абстрагировать и инкапсулировать доступ к источнику данных. DAO управляет соединением с источником данных для получения и записи данных.

Использующие DAO бизнес-компоненты работают с более простым интерфейсом, предоставляемым объектом DAO своим клиентам. DAO полностью скрывает детали реализации источника данных от клиентов. Поскольку при изменениях реализации источника данных представляемый DAO интерфейс не изменяется, этот паттерн дает возможность DAO принимать различные схемы хранилищ без влияния на клиенты или бизнес-компоненты. По существу, DAO выполняет функцию адаптера между компонентом и источником данных [8]

### **Шаблон проектирования Команда**

Команда — это поведенческий паттерн проектирования, который превращает запросы в объекты, позволяя передавать их как аргументы при вызове методов, ставить запросы в очередь, логировать их, а также поддерживать отмену операций. Команды могут использовать параметры, которые передают ассоциированную с командой информацию. Кроме того, команды могут ставиться в очередь и также могут быть отменены.

Паттерн Command отделяет объект, инициирующий операцию, от объекта, который знает, как ее выполнить. Единственное, что должен знать инициатор, это как отправить команду. Это придает системе гибкость: позволяет осуществлять динамическую замену команд, использовать сложные составные команды, осуществлять отмену операций [9].

### **Шаблон проектирования Одиночка**

Данный паттерн гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа. Существенно то, что можно пользоваться именно экземпляром класса, так как при этом во многих случаях становится доступной более широкая функциональность.

Плюсы:

* контролируемый доступ к единственному экземпляру;
* уменьшение числа имён;
* допускает уточнение операций и представления;
* допускает переменное число экземпляров;
* большая гибкость, чем у операций класса.

Минусы:

* глобальные объекты могут быть вредны для объектного программирования, в некоторых случаях приводя к созданию немасштабируемого проекта;
* усложняет написание модульных тестов и следованию TDD.

Применение:

* должен быть ровно один экземпляр некоторого класса, легко доступный всем клиентам;
* единственный экземпляр должен расширяться путем порождения подклассов, и клиентам нужно иметь возможность работать с расширенным экземпляром без модификации своего кода [10].

### **Шаблон проектирования Строитель**

Паттерн Строитель отделяет алгоритм поэтапного конструирования сложного продукта (объекта) от его внешнего представления так, что с помощью одного и того же алгоритма можно получать разные представления этого продукта.

Используется данный паттерн в следующих случаях:

* когда процесс создания нового объекта не должен зависеть от того, из каких частей этот объект состоит и как эти части связаны между собой;
* когда необходимо обеспечить получение различных вариаций объекта в процессе его создания.

Любой паттерн проектирования представляет собой формализованное описание часто встречающейся задачи проектирования, удачное решение данной задачи, а также рекомендации по применению этого решения в различных ситуациях.

Сообразное использование паттернов проектирования дает разработчику ряд неоспоримых преимуществ. Модель системы, построенная в терминах паттернов проектирования, фактически является структурированным выделением тех элементов и связей, которые значимы при решении поставленной задачи. Помимо этого, модель, построенная с использованием паттернов проектирования, более проста и наглядна в изучении, чем стандартная модель. Тем не менее, несмотря на простоту и наглядность, она позволяет глубоко и всесторонне проработать архитектуру разрабатываемой системы с использованием специального языка. Применение паттернов проектирования повышает устойчивость системы к изменению требований и упрощает неизбежную последующую доработку системы. Кроме того, трудно переоценить роль использования паттернов при интеграции информационных систем организации. Также следует упомянуть, что совокупность паттернов проектирования, по сути, представляет собой единый словарь проектирования, который, будучи унифицированным средством, незаменим для общения разработчиков друг другом [11].

# **ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ СИСТЕМЫ ОЦЕНКИ КОМПАНИИ НА ОСНОВЕ ПОКАЗАТЕЛЯ СТОИМОСТИ КАПИТАЛА НА ОСНОВЕ СТАНДАРТА IDEF0**

IDEF0 - нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. Описание выглядит как «[чёрный ящик](https://ru.wikipedia.org/wiki/%D0%A7%D1%91%D1%80%D0%BD%D1%8B%D0%B9_%D1%8F%D1%89%D0%B8%D0%BA)» с входами, выходами, управлением и механизмом, который постепенно детализируется до необходимого уровня.

К ее особенностям можно отнести:

* использование контекстной диаграммы;
* поддержка декомпозиции;
* доминирование;
* выделение 4 типов стрелок [12].

Главным в рассматриваемой предметной области оценки компании главным был выделен процесс «Оценить компанию». Контекстная диаграмма представлена на рисунке 3.1

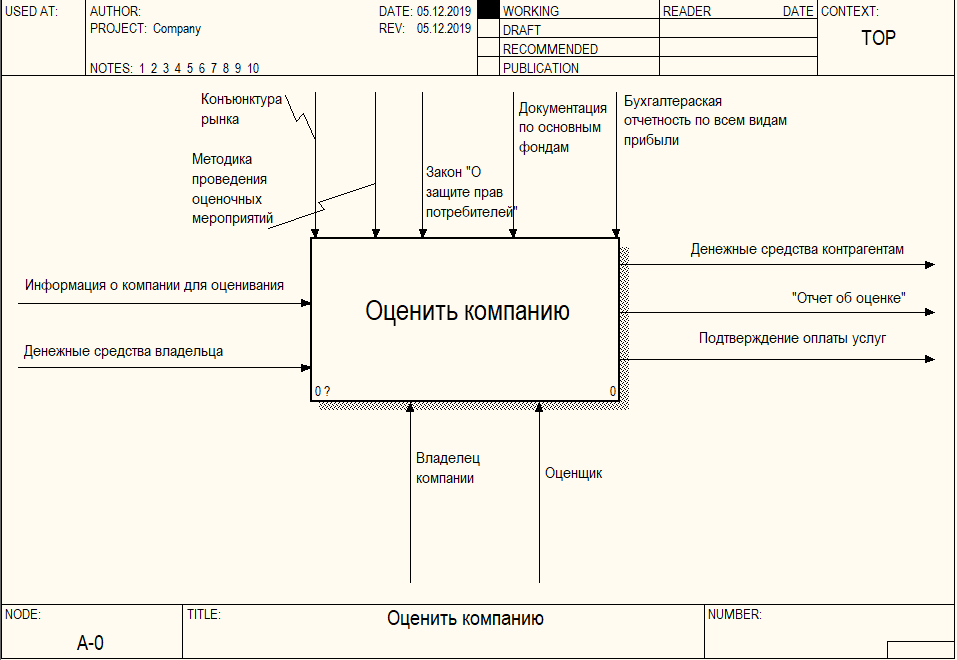


Рисунок 3.1 – Контекстная диаграмма процесса оценки капитала

Вход представлен двумя стрелками: информация о компании для оценивания и денежные средства владельца. На выходе находятся: денежные средства контрагентам, «Отчет об оценке», а также подтверждение оплаты услуг. Стрелки управления, то есть условия, необходимые процессу, чтобы произвести правильный выход, включают конъюнктуру рынка, методику проведения оценочных мероприятий, закон «О защите прав потребителей», документацию по основным фондам, бухгалтерскую отчетность по всем видам прибыли. Поддерживают выполнение процесса – механизмы – следующие: владелец компании, оценщик.

При детализации было выделено 4 блока: собрать информацию о компании, провести оценку компании, сделать заключение, оплатить услуги оценщика (рисунок 3.2).

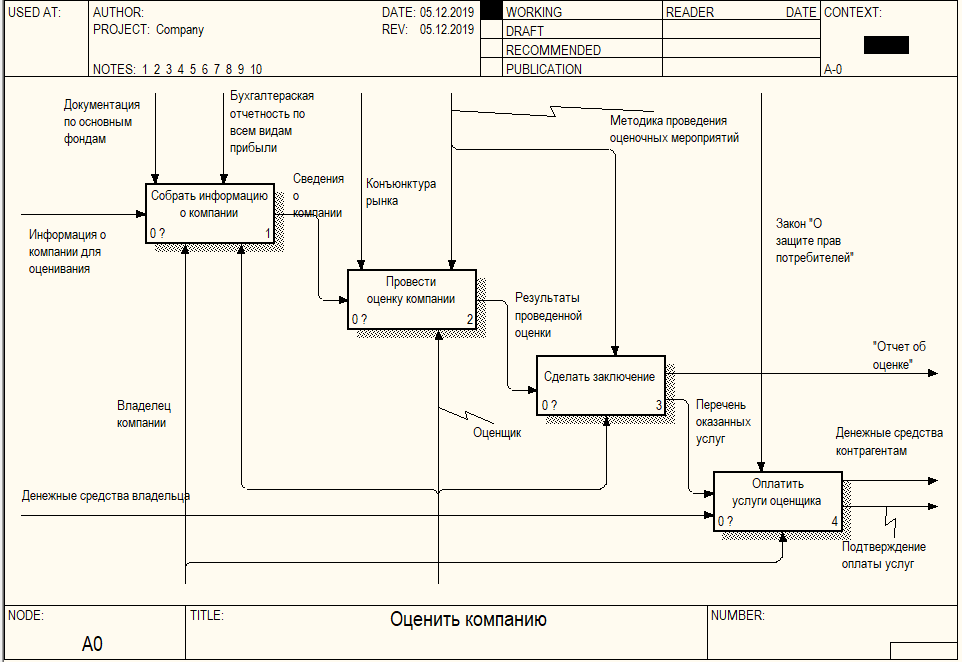


Рисунок 3.2 – Диаграмма декомпозиции процесса оценки компании

Рассмотрим каждый из подпроцессов в отдельности. Детализирование процесса «Собрать информацию о компании» изображено на рисунке 3.3. Данный процесс был разбит на три подпроцесса: собрать и проанализировать информацию об основных фондах, собрать и проанализировать информацию по всем видам прибыли, проанализировать полученные данные.

Данный процесс отражает первый и, возможно, самый важный этап в оценке компании – подготовка информационного поля для проведения оценки компании на основе показателя стоимости капитала. Так как для проведения оценки был выбран расчет показателя EBITDA, то анализу подлежит такая отчетность компании, как ее прибыль, амортизация, налоги и т.п, что и отражено на получившейся диаграмме. При сборе информации об основных фондах не обойтись без документации компании по основным фондам, а анализ прибыли невозможен без бухгалтерской отчетности по всем видам прибыли за некоторый период.

При анализе полученных в предыдущих работах сведений управлением выступает некоторые правила, методики проведения оценочных мероприятий.

Важным механизмом является оценщик, участвующий во всех трех процессах; владелец компании занимается только сбором информации, но не анализом всех полученных сведений.

После выполнения всех процессов из неструктурированной разрозненной информации о компании будут получены четкие сведения.

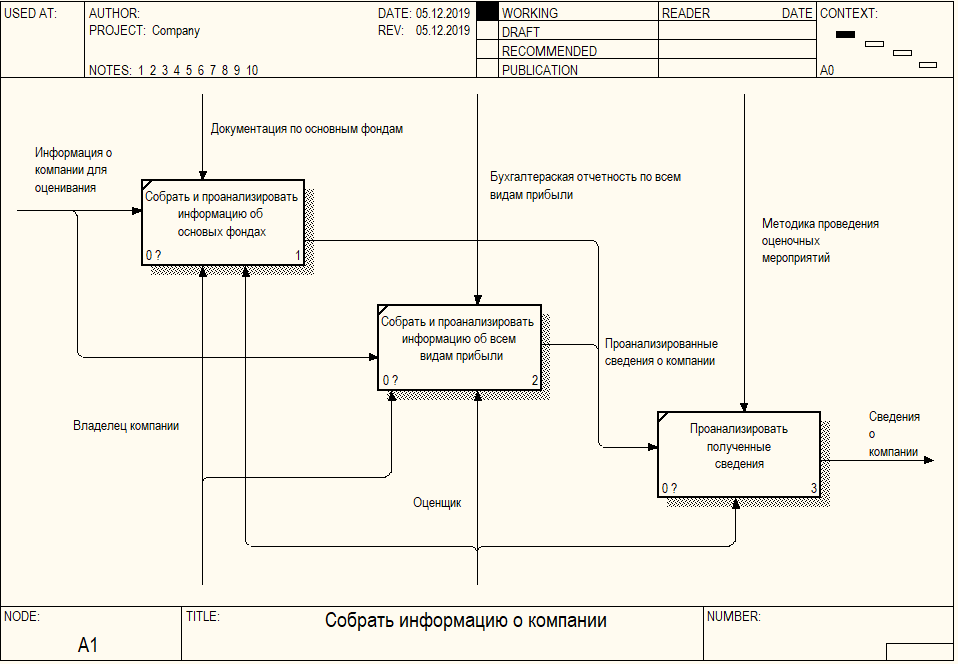


Рисунок 3.2 – Диаграмма декомпозиции процесса оценки компании

Ключевым процессом является второй – проведение оценки компании. Данный процесс может быть разбит на четыре подпроцесса, а именно «провести бенчмаркинг», «выбрать и рассчитать мультипликатор», «отсечь нерепрезентативные значения компаний-аналогов», «внести итоговые корректировки». Механизм всех процессов – оценщик, управлением для всех является методика проведения оценочных мероприятий, а также конъюнктура рынка для первого подпроцесса (рисунок 3.3).

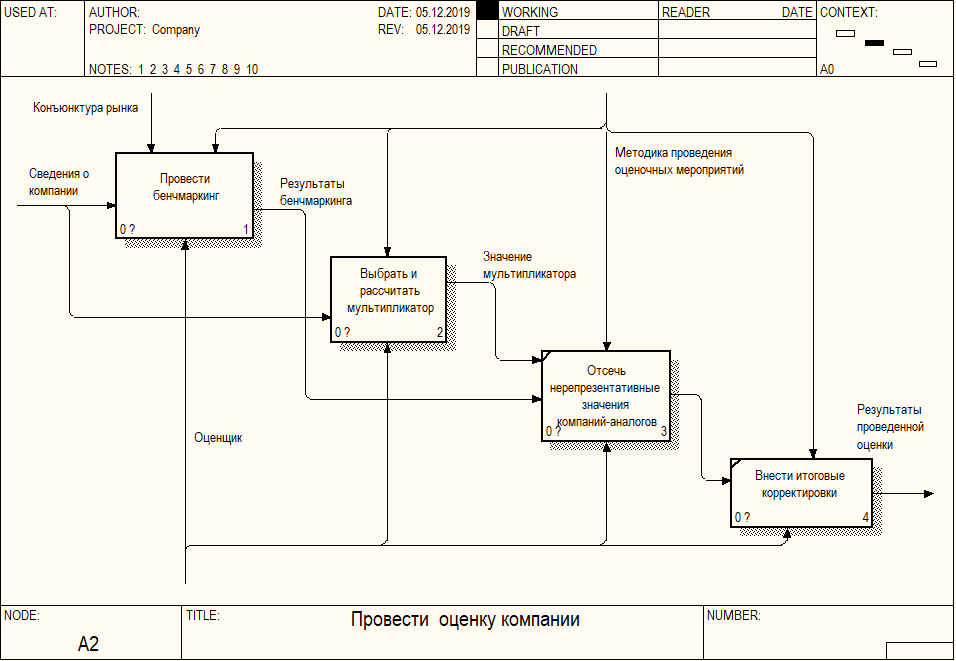


Рисунок 3.3 – Декомпозиция процесса «Провести оценку компании»

Первый процесс «Провести бенчмаркинг» также был детализирован (рисунок 3.4).

Бенчмаркинг (benchmarking) - процесс сравнения продуктов, услуг или процессов одной организации с продуктами, услугами или процессами другой организации. Иногда, проводят сравнение внутри одной и той же организации [13].

Подпроцессы следующие:

* определить широкую выборку компаний рынка по общим критериям, характеризующим компанию;
* сузить список, используя нишевые критерии (сюда относится доля рынка, уровень конкуренции, управленческая команда, потенциал роста, финансовые показатели и т.д.);
* проанализировать полученный список компаний.

Результаты изучения отрасли и других компаний помогут составить общую картину занимаемого оцениваемой компанией места. Также после расчета мультипликатора полученное значение будет сравнивать именно со списком компаний, отобранных на данном этапе, чтобы определить, является ли данная компания удачной для инвестирования.

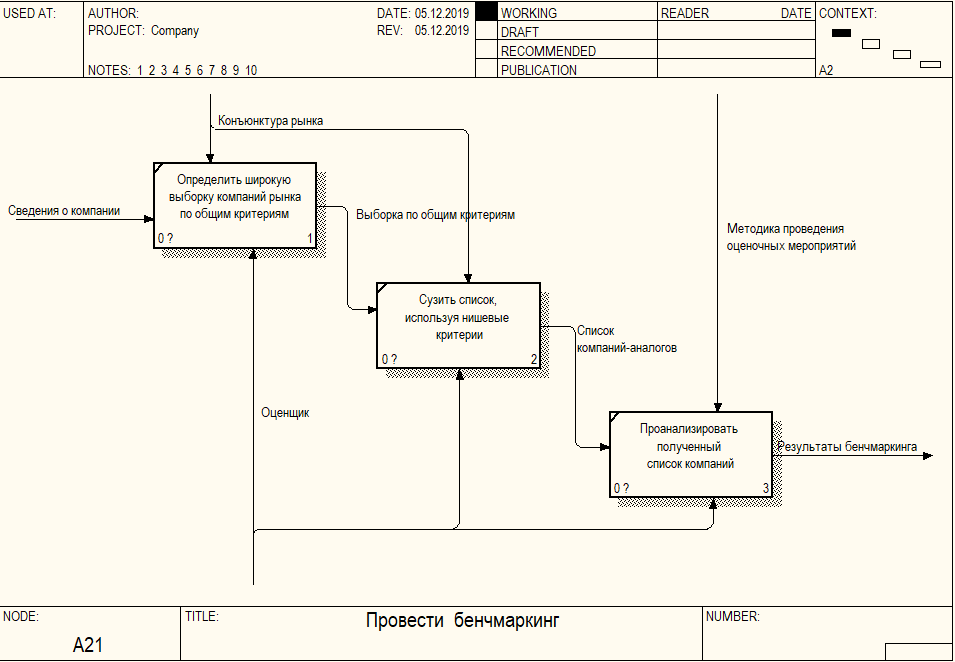


Рисунок 3.4 – Декомпозиция процесса «Провести бенчмаркинг»

Декомпозиция процесса «Выбрать и рассчитать мультипликатор» представлена на рисунке 3.5. Было выделено четыре подпроцесса: рассчитать прибыль, добавить расходы на налоги, прибавить амортизацию, провести окончательные расчеты. Причем четвертый процесс использует данные, полученные при выполнении первых трех. Уже не раз упоминалось, что в данном курсовом проекте используется показатель EBITDA, поэтому данные подпроцессы полностью нацелены на удовлетворения составляющих расчета данного показателя.

Стрелки механизма и управления по одной– оценщик и методика проведения оценочных мероприятий соответственно.

В результате выполнения данного процесса будет получено значение мультипликатора EBITDA для конкретной компании, которая нуждается в оценке.

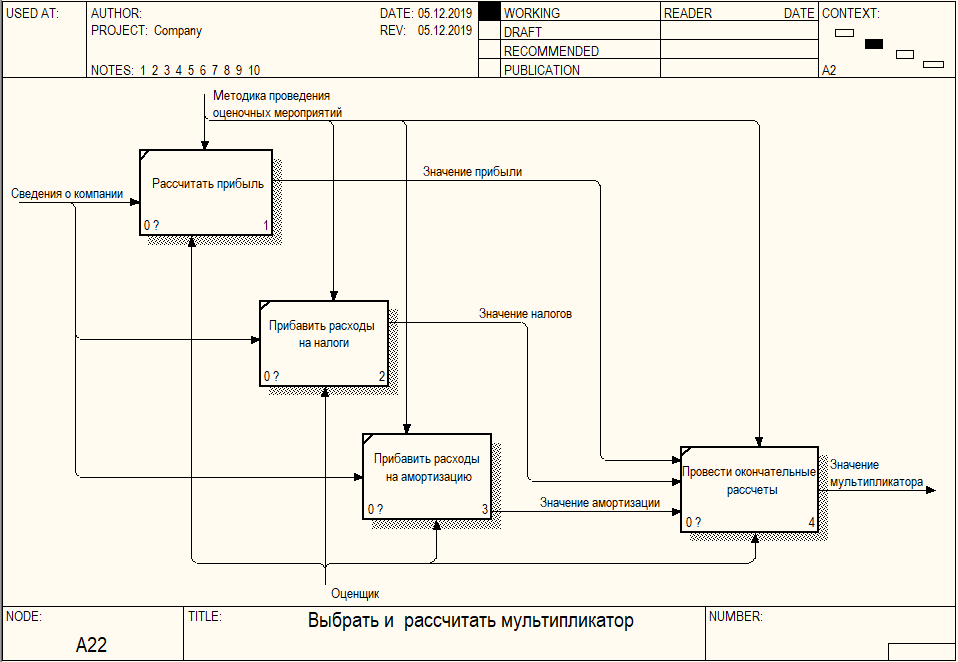


Рисунок 3.5 – Декомпозиция процесса «Выбрать и рассчитать мультипликатор»

После проведения собственно процедуры оценивания и расчетов, оценщику необходимо составить и предоставить владельцу компании «Отчет об оценке». Данные подпроцессы изображены на рисунке 3.6.

Отчет об оценке - документ, предоставляемый экспертом-оценщиком заказчику, в котором обосновывается мнение эксперта по результату оценки. Отчет включает: сопроводительное письмо, заключение, задание на оценку, основную часть, обоснование и расчеты, допущения и ограничения, согласование результатов, полученных различными методами, фотографии и другие справочные материалы, личную подпись, квалификационный аттестат и реквизиты эксперта-оценщика [14].

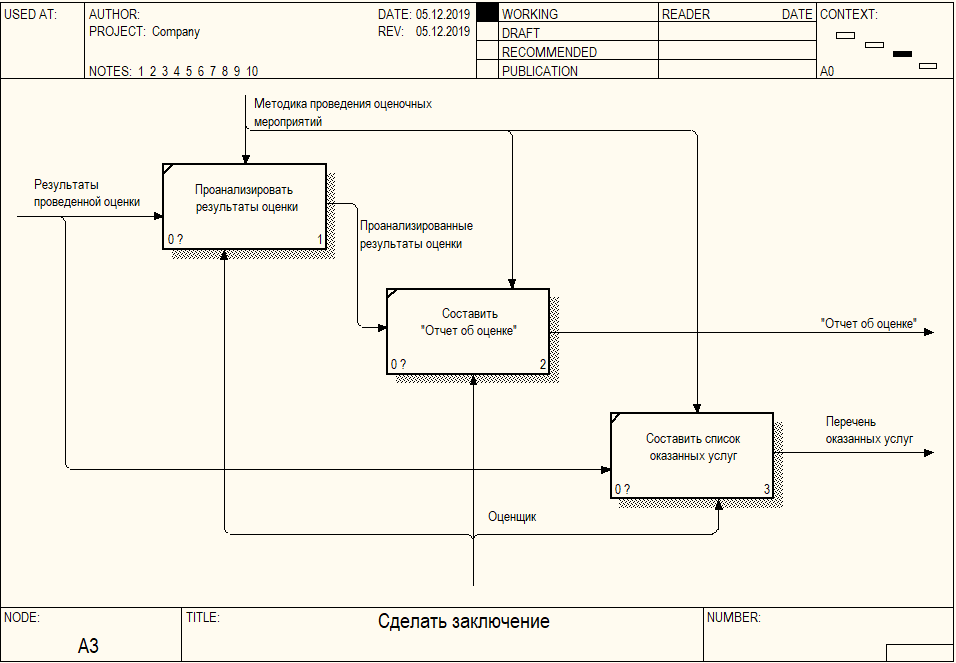


Рисунок 3.6 – Декомпозиция процесса «Сделать заключение»

Процесс «Оплатить услуги оценщика» представлен на рисунке 3.7:

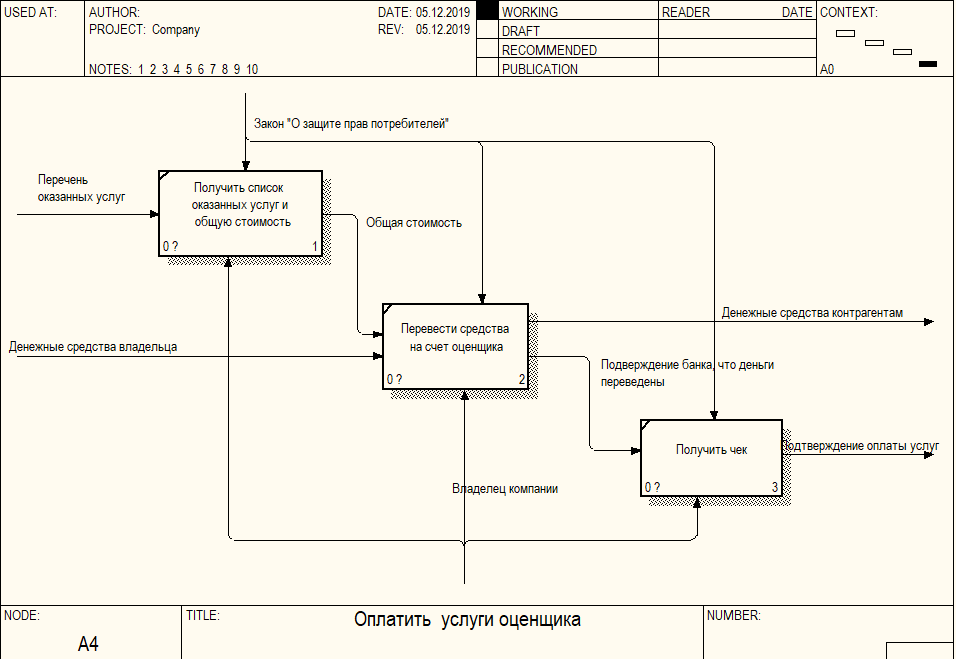


Рисунок 3.7 – Декомпозиция процесса «Оплатить услуги оценщика»

Как и любая работа, работа оценщика-эксперта также должна быть оплачена. В свою очередь заказчик в подтверждение оплаты получает некоторый чек, что потом можно будет использовать в некоторых конфликтных ситуациях, как доказательство совершения оплаты. Управлением всех подпроцессов является закон РБ «О защите прав потребителей».

Построение контекстной диаграммы и диаграмм декомпозиции позволяет предварительно описать проектируемую систему, увидеть и описать все бизнес-процессы, которые нуждаются в автоматизации и улучшении. Причем именно визуальное представление такой информации делает последующую работу понятнее и структурированнее.

# **ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ ОЦЕНКИ КОМПАНИИ НА ОСНОВЕ ПОКАЗАТЕЛЯ СТОИМОСТИ КАПИТАЛА**

Была составлена модель, изображенная на рисунке 4.1. На ней показаны такие сущности, как пользователи(users), статус пользователя(userstatus), сегменты рынка(segments), статус оценки компании(companymarkstatus), компании (companies).

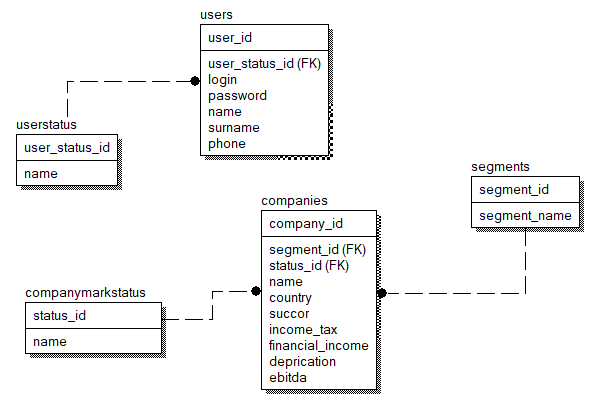


Рисунок 4.1 – Информационная модель системы оценки компании на основе показателя стоимости капитала

Согласно технического задания необходимо разграничить функциональность обычных пользователей и администраторов. При этом выбранная предметная область также подразумевает выделение различных категорий пользователей. Для реализации данных требований была создана сущность userstatus, атрибутами которой являются уникальный user\_status\_id а также некоторое название статуса name. Данная сущность как раз и хранит список различных статусов, которыми могут обладать различные группы пользователей. Например, это могут быть администратор, оценщик и некоторый обычный пользователь, например, представитель отрасли, компании которой оцениваются.

Важной является сущность пользователей(users), она содержит уникальный идентификатор, внешний ключ идентификатор некоторого статуса, логин, пароль, имя, фамилию, телефон.

Сущность сегменты необходима для хранения информации о различных сегментах рынка. Она содержит два атрибута: идентификатор и название. Сущность companymarkstatus включает status\_id и name.

Также весьма важной является сущность companies. Так как ключевой функциональностью разрабатываемой системы является оценка различных команий, то нам не обойтись без хранения информации об этих самых компаниях. Каждой из них соответствует некоторый статус оценки, а также все компании можно разделить на некоторые группы по сегментам рынка, к которым они относятся. Также данная сущность имеет такие атрибуты, как name, country, succor, income\_tax, financial\_income, deprication, ebitda.

Все сущности связаны между собой связью 1 ко многим (1 : М). В типе связей один ко многим одной записи первой таблицы соответствует несколько записей в другой таблице. Связь 1 : М используется наиболее часто. Также эти связи являются неидентифицирующими.

Неидентифицирующая связь показывает на зависимость между родительской и дочерней сущностями, при этом экземпляр дочерней сущности может быть однозначно идентифицирован без ссылки на экземпляр родительской сущности.

Переменная отношения находится в третьей нормальной форме тогда и только тогда, когда она находится во второй нормальной форме (переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме (переменная отношения находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов) и каждый неключевой атрибут неприводимо зависит от её потенциального ключа), и отсутствуют транзитивные функциональные зависимости неключевых атрибутов от ключевых.

Разработанная информационная модель соответствует всем вышеперечисленным требованиям, а значит находится в третьей нормальной форме.

Построение информационной модели системы помогло структурировать данные для последующей разработки базы данных.

# **МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ ОЦЕНКИ КОМПАНИИ НА ОСНОВЕ ПОКАЗАТЕЛЯ СТОИМОСТИ КАПИТАЛА**

## **5.1 Диаграмма вариантов использования**

Диаграммы вариантов использования описывают взаимоотношения и зависимости между группами вариантов использования и действующих лиц, участвующими в процессе.

Диаграммы вариантов использования не предназначены для отображения проекта и не могут описывать внутреннее устройство системы. Они предназначены для упрощения взаимодействия с будущими пользователями системы, с клиентами, и особенно пригодятся для определения необходимых характеристик системы. Другими словами, диаграммы вариантов использования говорят о том, что система должна делать, не указывая сами применяемые методы.

Некоторые требования к use case diagram:

* каждый вариант использования относится как минимум к одному действующему лицу,
* каждый вариант использования имеет инициатора,
* каждый вариант использования приводит к соответствующему результату (результату с «бизнес-значением»).

Действующее лицо является внешним источником (не элементом системы), который взаимодействует с системой через вариант использования. Действующие лица могут быть как реальными людьми (например, пользователями системы), так и другими компьютерными системами или внешними событиями [15].

Диаграмма использования для разрабатываемой системы оценки компании на основе показателя стоимости капитала представлена на рисунке 5.1.

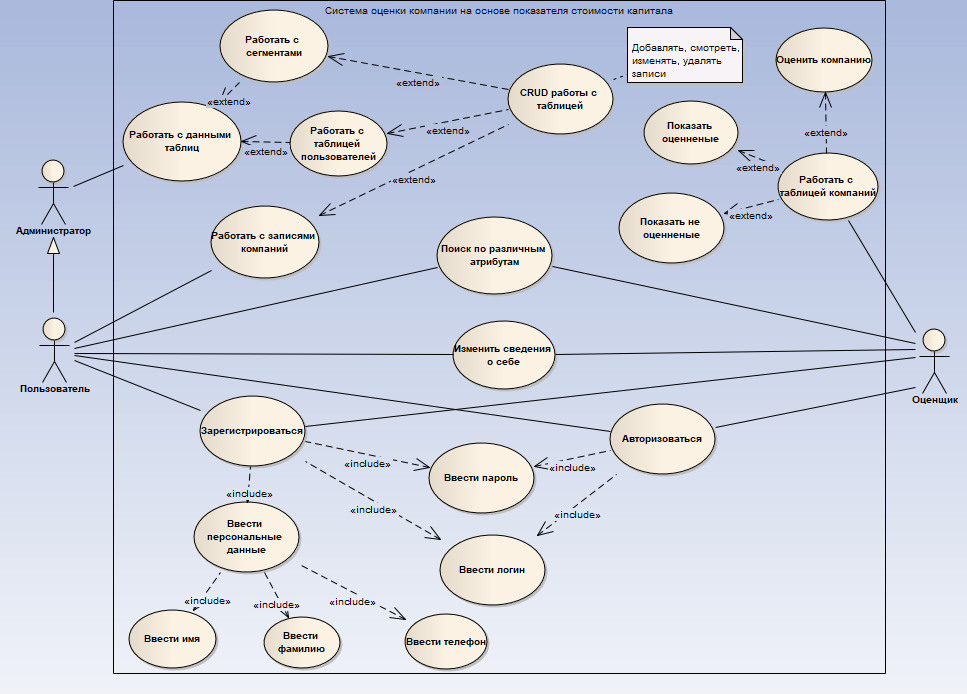


Рисунок 5.1 – Диаграмма вариантов использования

## 

## **5.2 Диаграмма последовательности**

Диаграммы последовательностей используются для уточнения диаграмм прецедентов, более детального описания логики сценариев использования.

Диаграммы последовательностей обычно содержат объекты, которые взаимодействуют в рамках сценария, сообщения, которыми они обмениваются, и возвращаемые результаты, связанные с сообщениями. Впрочем, часто возвращаемые результаты обозначают лишь в том случае, если это не очевидно из контекста.

Объекты обозначаются прямоугольниками.

Сообщения (вызовы методов) - линиями со стрелками.

Возвращаемые результаты - пунктирными линиями со стрелками.

Прямоугольники на вертикальных линиях под каждым из объектов показывают “время жизни” объектов [16].

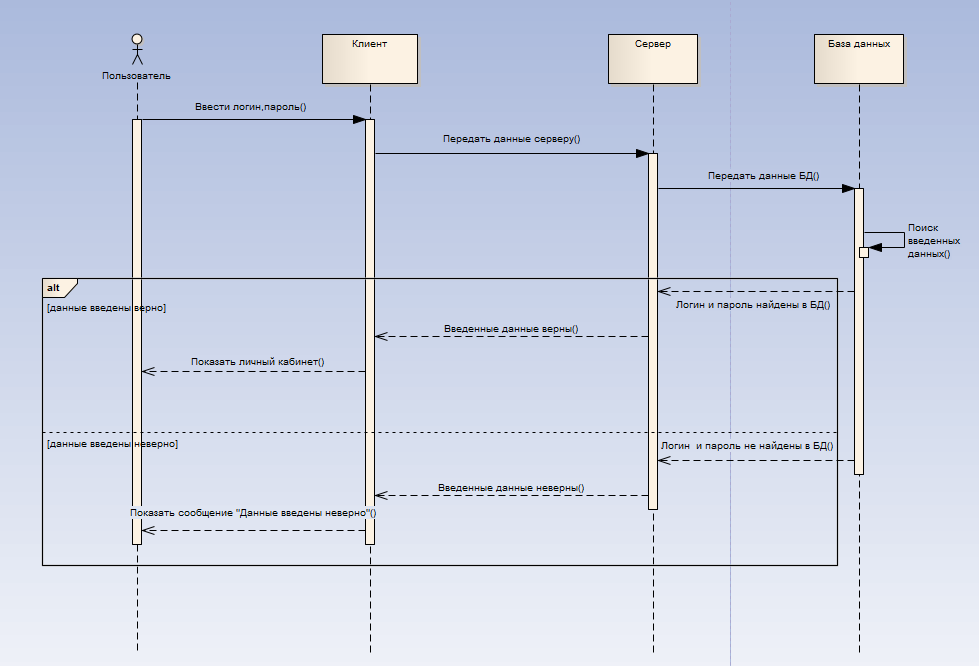


Рисунок 5.2 – Диаграмма последовательности

## **Диаграмма состояний**

Диаграммы состояний являются хорошо известным средством описа­ния поведения систем. Они определяют все возможные состояния, в кото­рых может находиться конкретный объект, а также процесс смены состояний объекта в результате влияния некоторых событий. Каждая диаграмма состояний представляет некоторый автомат.

В поведении объекта в системе можно выделить действия, отображаемые переходами, и деятельности, отображаемые состояниями.

Из конкретного состояния в данный момент времени может быть осуществлен только один переход; таким образом, условия являются взаимно исключающими для любого события. Существует два особых состояния: вход и выход. Любое действие, связанное с событием входа, выполняется, когда объект входит в данное со­стояние. Событие выхода выполняется в том случае, когда объект выходит из данного состояния. Диаграммы состояний хорошо использовать для описания поведения некоторого объекта в нескольких различных вариантах использования. Они не слишком пригодны для описания поведения ряда взаимодействующих объектов.

Рекомендуется строить диаграммы состояний только для тех классов, поведение которых влияет на общее поведение системы, например, для классов пользовательского интерфейса и управляющих объектов [17].

Диаграмма состояний для курсового проекта представлена на рисунке 5.3.

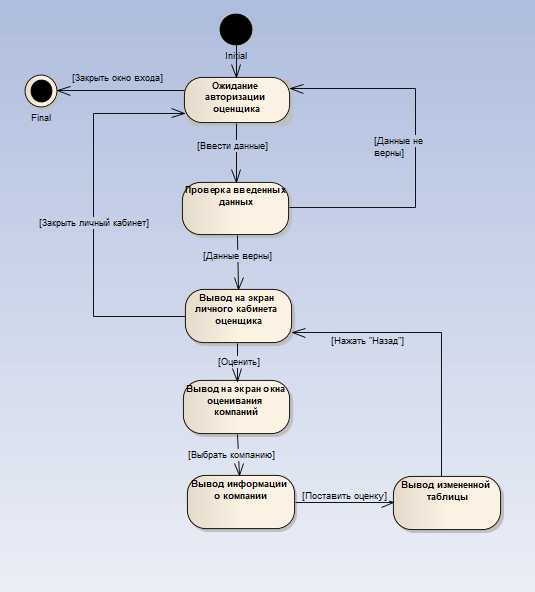


Рисунок 5.3 – Диаграмма состояний

## **Диаграмма развертывания**

Для разрабатываемой системы диаграмма развертывания изображена на рисунке 5.4.

Это диаграмма, на которой изображается конфигурация для работающих узлов и [экземпляров](https://openu.ru/Books/UML/Instance.asp) [компонента](https://openu.ru/Books/UML/Component.asp), а также объектов, которые на них существуют. Компоненты представляют собой выполнимые блоки программного кода. Компоненты, которые не существуют как единицы времени выполнения (в том случае, если они не были задействованы при компиляции), на этих диаграммах не указываются. Их место - на диаграммах компонентов. На диаграмме развертывания изображаются экземпляры компонентов, в то время как на диаграмме компонентов показаны определения их типов.

Диаграмма развертывания представляет собой схему, состоящую из символов узлов, соединенных маршрутами коммуникационных ассоциаций. Внутри символов узлов могут находиться экземпляры компонентов. Это указывает на то, что компонент существует или выполняется на данном узле. Символы компонентов, в свою очередь, могут содержать в себе символы объектов. Это обозначает, что объект является частью компонента. Компоненты связаны между собой пунктирными стрелками зависимостей (возможно, через интерфейсы). Такие связи обозначают, что один компонент использует услуги другого. Когда на диаграмме нужно точно указать вид зависимости, используют стереотип [18].

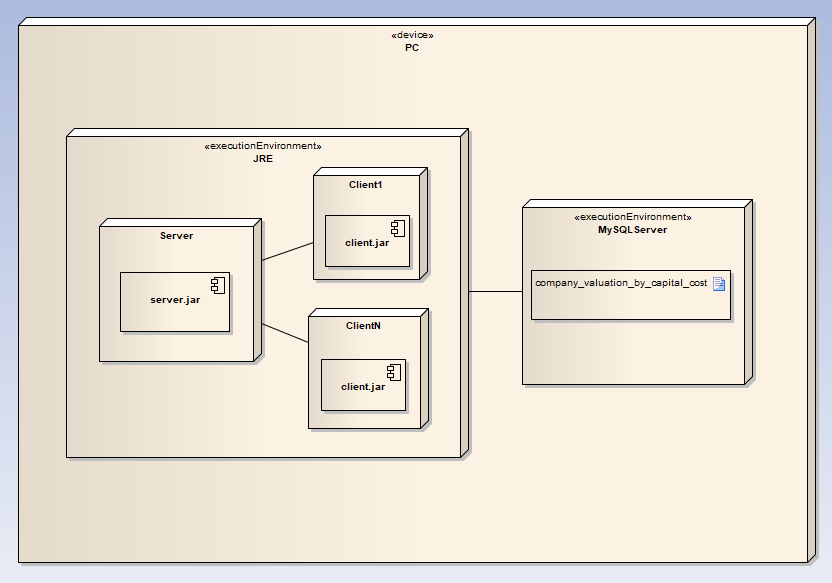


Рисунок 5.4 – Диаграмма развертывания

## **Диаграмма компонентов**

Диаграмма компонентов описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Во многих средах разработки модуль или компонент соответствует файлу. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними [18].

Для данной системы диаграмма компонентов представлена на рисунке 5.5.

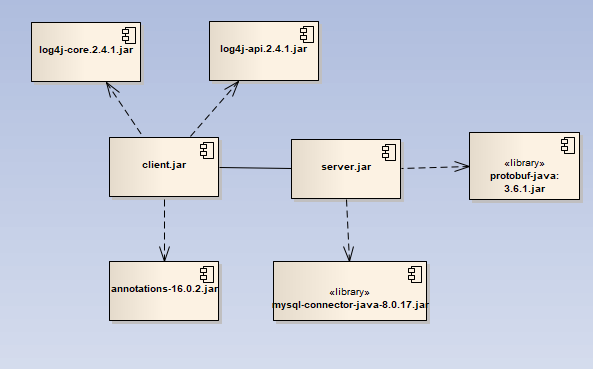


Рисунок 5.5 – Диаграмма компонентов

## **Диаграммы классов**

На рисунке Б.1 приложения Б представлены такие классы как Segment, Company, User, которые наследуются от класса AbstractEntity, а также их поля и методы. Выделение родительского класса обусловлено тем, что все три потомка имеют такие одинаковые поля как serialVersionID и id, а также соответствующие методы. При этом каждый из классов-наследников обладает некоторыми уникальными атрибутами и методами, которые характерны только каждому из них.

На рисунке Б.2 приложения Б изображен интерфейс ConnectionPool и класс DatabaseConnectionPool, который его реализует. Такое решение обусловлено тем, что данные данной системы хранятся в базе данных, а для доступа к ней необходимо создавать объект Connection. Создавать соединение при каждом обращении к базе данных приведет к необусловленной трате ресурсов, в том числе и временны. ConnectionPool подразумевает, что в нашем распоряжении имеется некоторый набор («пул») соединений к базе данных. Когда будет запрошен новый доступ к БД, будет выдано уже открытое соединение из этого пула. Если все открытые соединения уже заняты, создаётся новое. При освобождении одного из уже существующих соединений, оно становится доступно для других пользователей. Если соединение долго не используется, оно закрывается. Также можно задать default-размер такого пула.

Класс-строитель для упомянутого выше класса Company изображен на рисунке Б.3 приложения Б. Для реализации паттерна Строитель был создан класс, который позволяет создать объект класса Company по частям, частей столько же, сколько полей в классе. С помощью строителя появилась возможность создавать только те части, которые необходимы, в любом порядке.

Весомую часть классов клиентской части составляют классы-контроллеры. Например, на рисунке Б.4 приложения Б изображен контроллер для таблицы компаний. Имеющиеся поля и методы позволяют обрабатывать все необходимые нажатия кнопок, просмотр и заполнение таблицы и т.д.

В данном курсовом проекте были использованы различные виды диаграмм UML. Это позволило описать спроектированную систему с точки зрения составных частей, функций и компонентов.

# **ОПИСАНИЕ АЛГОРИТМОВ, РЕАЛИЗУЮЩИХ БИЗНЕС-ЛОГИКУ СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ ОЦЕНКИ КОМПАНИИ НА ОСНОВЕ ПОКАЗАТЕЛЯ СТОИМОСТИ КАПИТАЛА**

## **Алгоритм авторизации пользователя**

На рисунке А.1 приложения А изображен алгоритм авторизации пользователя.

Если пользователь попытается совершить вход без подключения к серверу, система уведомит его о том, что он не подключен и, следовательно, никакая работа по оценке компаний выполнена быть не может.

Если соединение было успешно установлено, то для авторизации пользователю необходимо ввести свои логин и пароль. Полученные данные будут обработаны контроллером и отправлены на сервер. Сервер, в свою очередь, сделает запрос к базе данных на поиск введенных данных в таблице users. После чего сервер полученные результаты поиска передаст на клиент. Если введенные пользователем данные были найдены в базе данных, то в зависимости от его роли будет открыто соответствующий личный кабинет: кабинет клиента, оценщика либо администратора. В случае, когда введенных данных в базе данных обнаружено не будет, на экране появится сообщение «Неверный логин или пароль».

## **Алгоритм добавления компании владельцем**

Система не имела бы смысла, если бы нельзя было добавлять новые компании для оценивания. Алгоритм добавления компании обычным пользователем (например, владельцем компании) изображен на рисунке 6.1.

После нажатия соответствующей кнопки, перед клиентом открывается окно с полями для заполнения. Пользователю обязательно нужно заполнить следующую информацию: название компании, страну, сегмент рынка, данные о прибыли, налогах, амортизации. После чего система на основе введенных данных сама рассчитывает показатель EBITDA.

Если все данные были введены верно, то данные о новой компании отправляются на сервер и добавляются в базу данных, на экране появляется сообщение о том, что операция добавления новой компании прошла успешно.

Если же при вводе данных пользователь в каком-то из полей допустил ошибку, система выдаст сообщение «Данные введены неверно».



Рисунок 6.1 – Алгоритм добавления компании

Описанные бизнес-процессы отражают суть бизнес-логики системы оценки компании на основе показателя стоимости капитала: авторизация пользователя и добавление своей компании с целью ее оценки после расчета определенного показателя. Причем сделать это можно легко, просто и удобно, что и доказывает оправданность такой системы по сравнению с ручным документированием подобных процессов.

# **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ СИСТЕМЫ ОЦЕНКИ КОМПАНИИ НА ОСНОВЕ ПОКАЗАТЕЛЯ СТОИМОСТИ КАПИТАЛА**

Конфигурационные данные для сервера и доступа к базе данных описаны в файле bd\_info.properties.

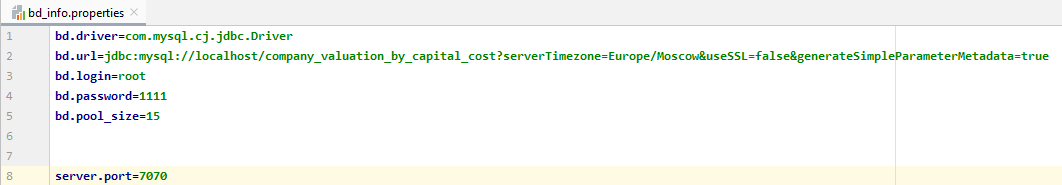


Рисунок 7.1 – Содержимое файла конфигурации сервера

Сначала необходимо запустить сервер, а затем клиент.

После чего пользователь увидит следующее окно (рисунок 7.2):

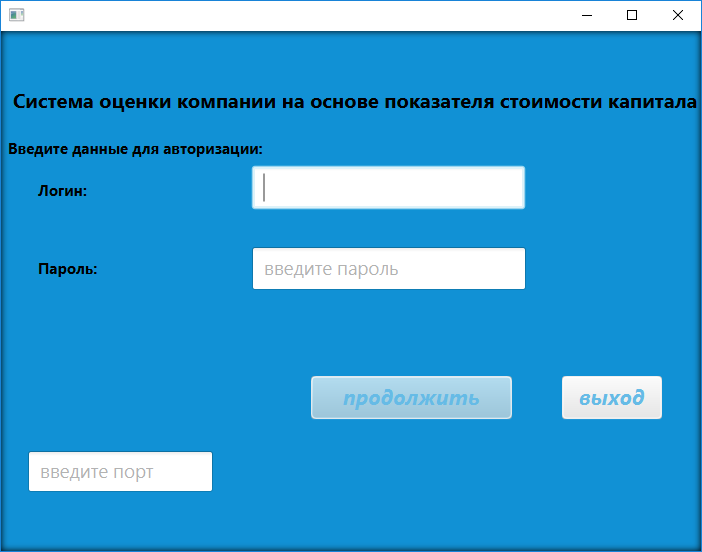


Рисунок 7.2 – Первое окно приложения

Здесь указано, что за система будет использоваться, можно ввести данные для авторизации. Однако первое, что необходимо сделать, это подключиться к серверу. Для этого необходимо ввести номер порта в текстовое поле снизу слева, а также нажать Enter. Если порт будет введен правильно, на экране появится сообщение об этом (рисунок 7.3), а также появится возможность зарегистрироваться (рисунок 7.4).

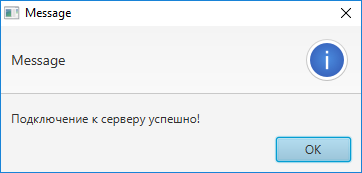


Рисунок 7.3 – Подключение к серверу прошло успешно

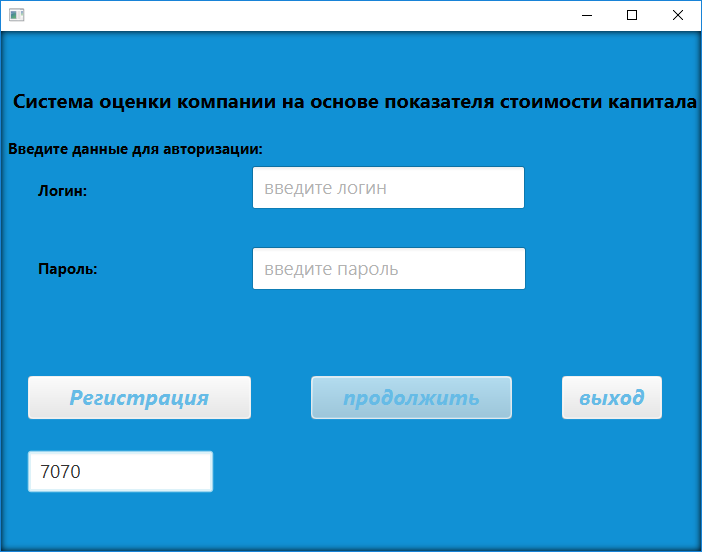


Рисунок 7.4 – Добавление кнопки «Регистрация»

Для регистрации пользователю надо ввести логин, пароль, имя, фамилию и телефон (рисунок 7.5).

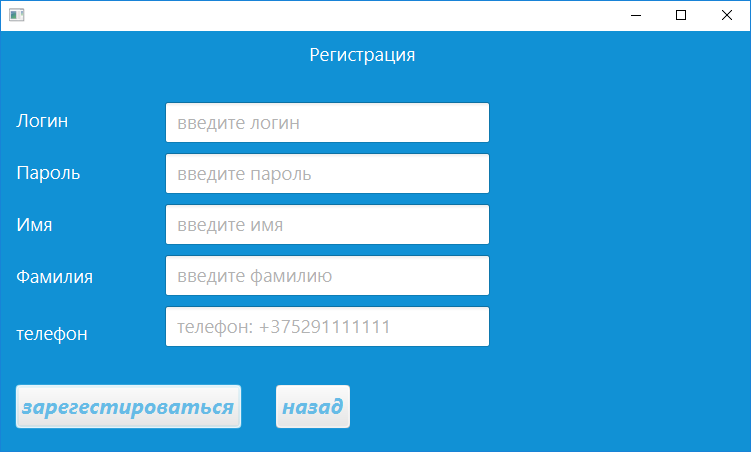


Рисунок 7.5 – Окно регистрации нового пользователя

Для входа в систему необходимо ввести логин и пароль. Причем вход будет осуществлен в личный кабинет согласно статуса пользователя. Функция изменения статусов пользователей есть только у администратора.

Личный кабинет обычного пользователя выглядит следующим образом (рисунок 7.6):

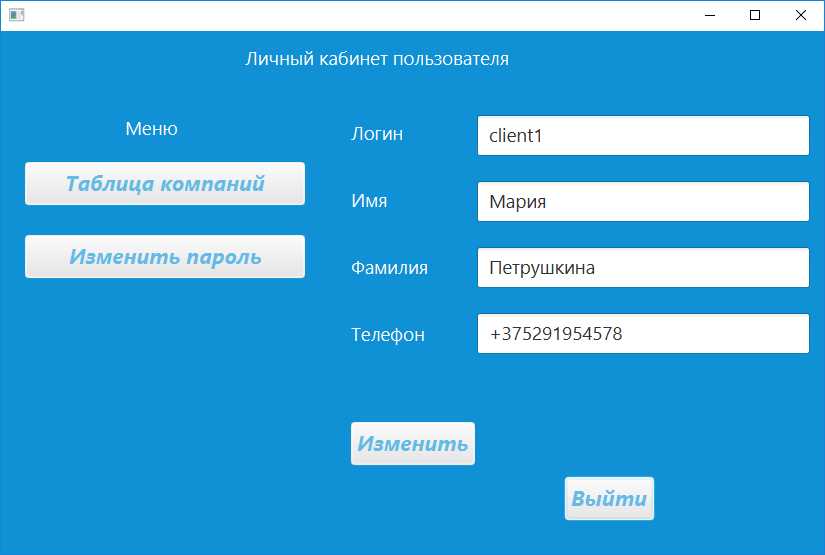


Рисунок 7.6 – Личный кабинет пользователя

Справа находится персональная информация пользователя, которую, при желании, можно изменить.

Также пользователь может изменить пароль и изменить таблицу компаний. Для изменения пароля необходимо нажать соответствующую кнопку. После этого откроется окно, изображенное на рисунке 7.7. В соответствующие textField необходимо ввести старый пароль, новый, а также подтвердить пароль.

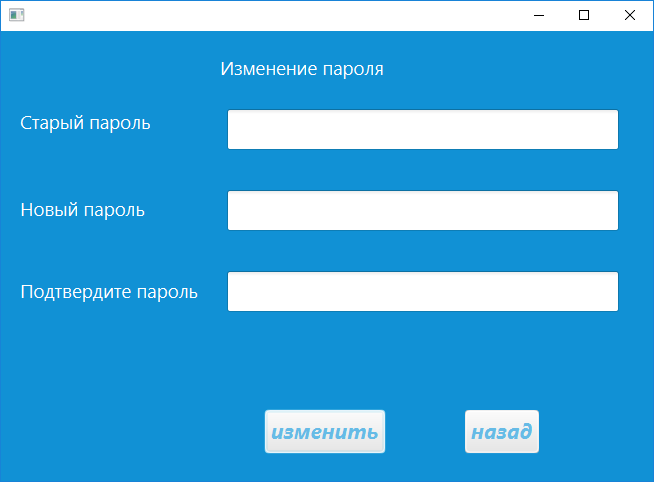


Рисунок 7.7 – Изменение пароля

Если все сделано верно, появится соответствующее сообщение (рисунок 7.8):

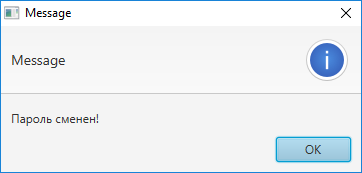


Рисунок 7.8 – Успешная смена пароля

При нажатии кнопки «Таблица компаний» будет открыто окно, аналогичное изображённому на рисунке 7.9.

Кроме визуального отображения содержимого таблицы, пользователю доступно добавление, изменение, удаление записей. Также можно отдельно просмотреть оцененные и не оцененные компании.

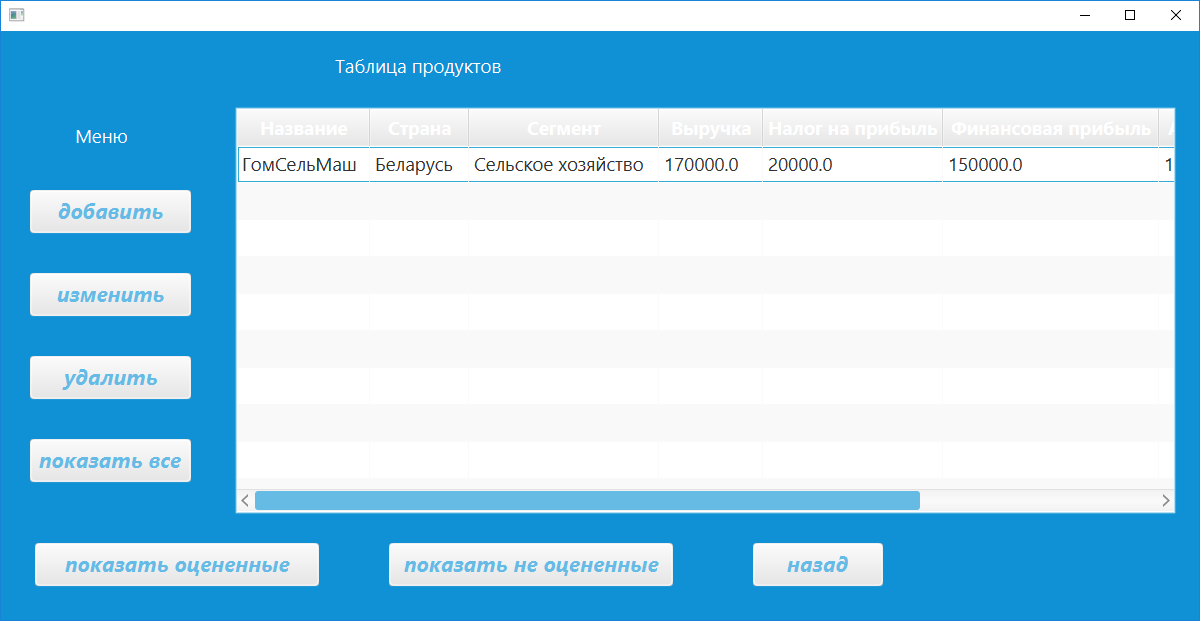


Рисунок 7.9 – Таблица компаний

Добавление новой компании выглядит следующим образом (рисунок 7.10):

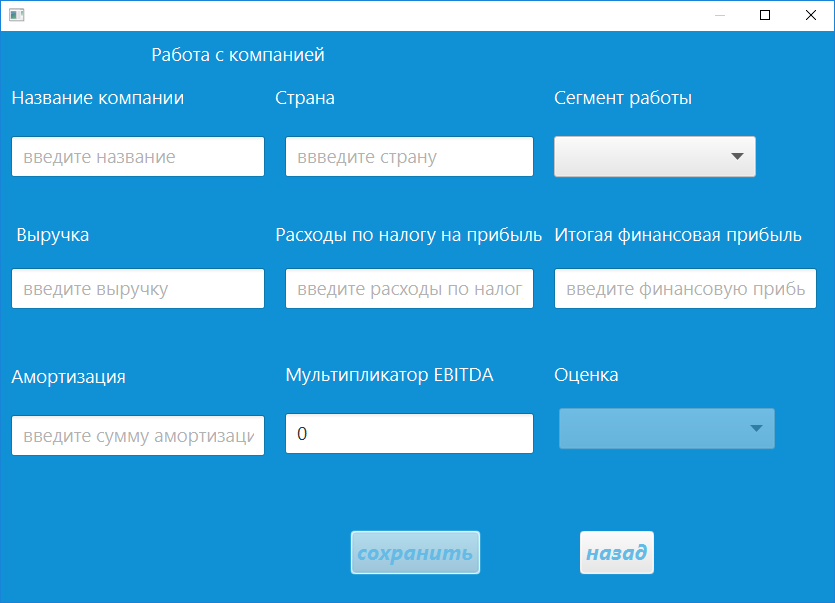


Рисунок 7.10 – Таблица компаний

Некоторые поля являются недоступными, что связано со статусом пользователя. Администратору или оценщику, например, данные поля доступны будут.

Для добавления новой компании необходимо указать название, страну компании, сегмент работы из выпадающего списка, выручку, расходы по налогу на прибыль, финансовую прибыль, амортизацию. Мультипликатор EBITDA будет рассчитан системой.

Если свои логин и пароль при авторизации введет оценщик, перед ним появится личный кабинет, имеющий следующий вид (рисунок 7.11):

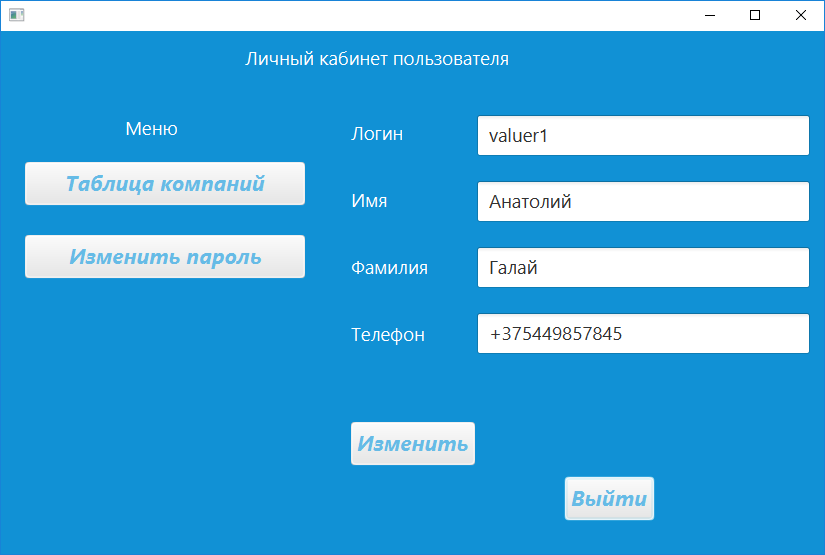


Рисунок 7.11 – Личный кабинет оценщика

Вид таблицы компаний для данной роли отображен на рисунке 7.12.

В отличие от пользователя, оценщик не может добавлять/удалять/изменять данные о компаниях. Его функция – оценивать. Для этого ему необходимо выбрать компанию из таблицы и нажать «Оценить».

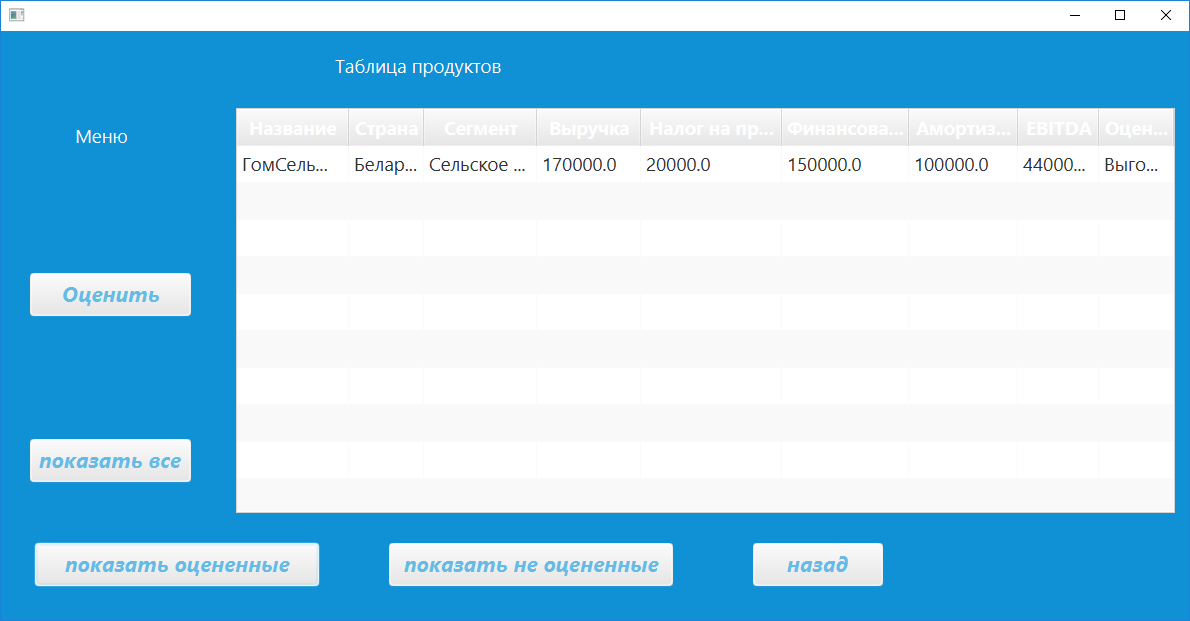


Рисунок 7.12 – Таблица компаний

Еще одна роль, предусмотренная данной системой, – администратор. Его функционал самый широкий. Личный кабинет администратора выглядит следующим образом (рисунок 7.13):

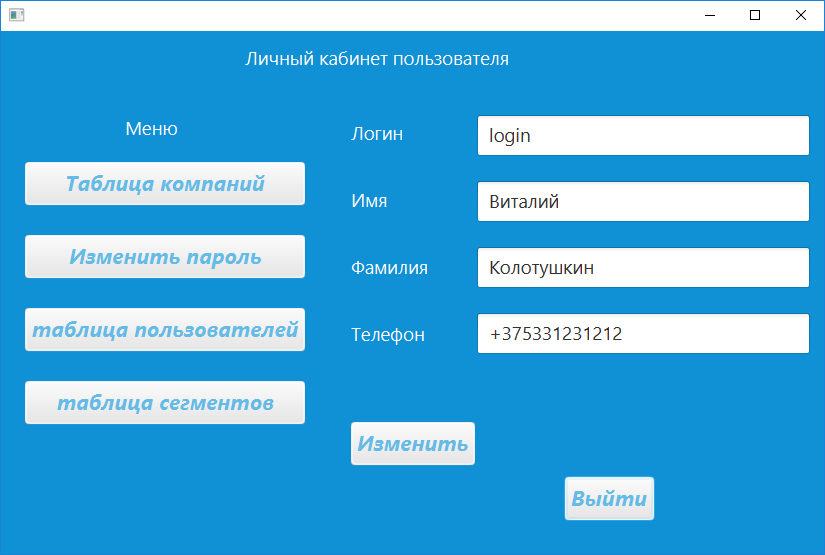


Рисунок 7.13 – Личный кабинет администратора

Кроме изменения сведений о себе, администратор может работать с таблицей компаний, добавлять новые, удалять и изменять существующие и т.д. (рисунок 7.14):

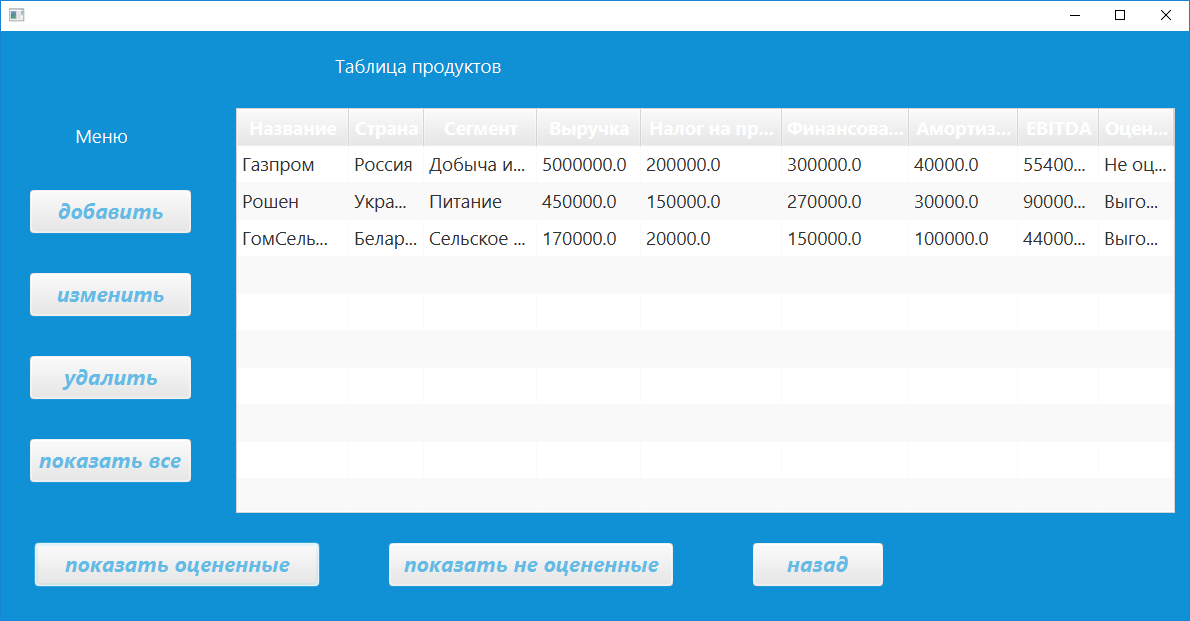


Рисунок 7.14 – Таблица компаний

При добавлении новой компании ему доступны все поля (рисунок 7.15):

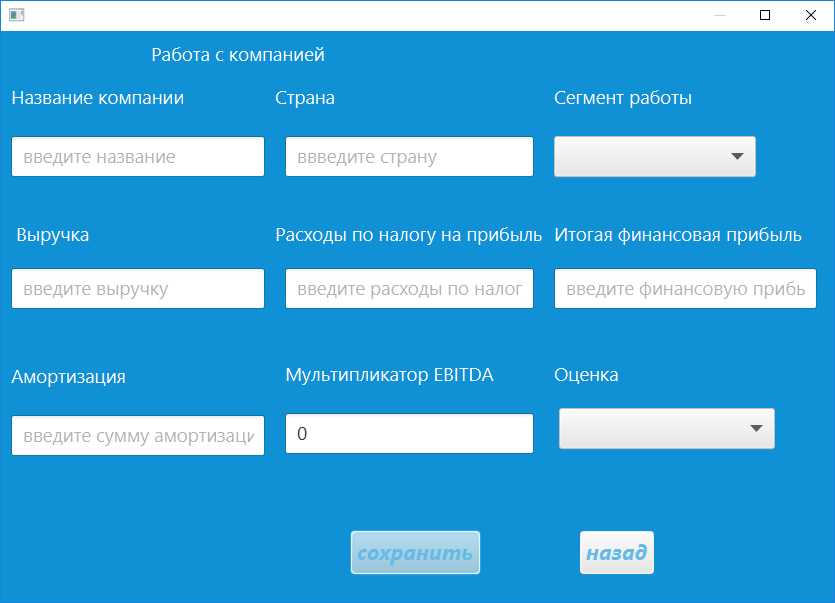


Рисунок 7.15 – Добавление новой компании

Для удаления необходимо выбрать строку таблицы и нажать «Удалить», причем оцененные компании удалить нельзя.

Доступной только администратору является функция работы с таблицей всех пользователей (рисунок 7.16). Зарегистрироваться в систему можно только обычным пользователем, и только администратор в своем личном кабинете может как-то изменить этот статус (рисунок 7.17):

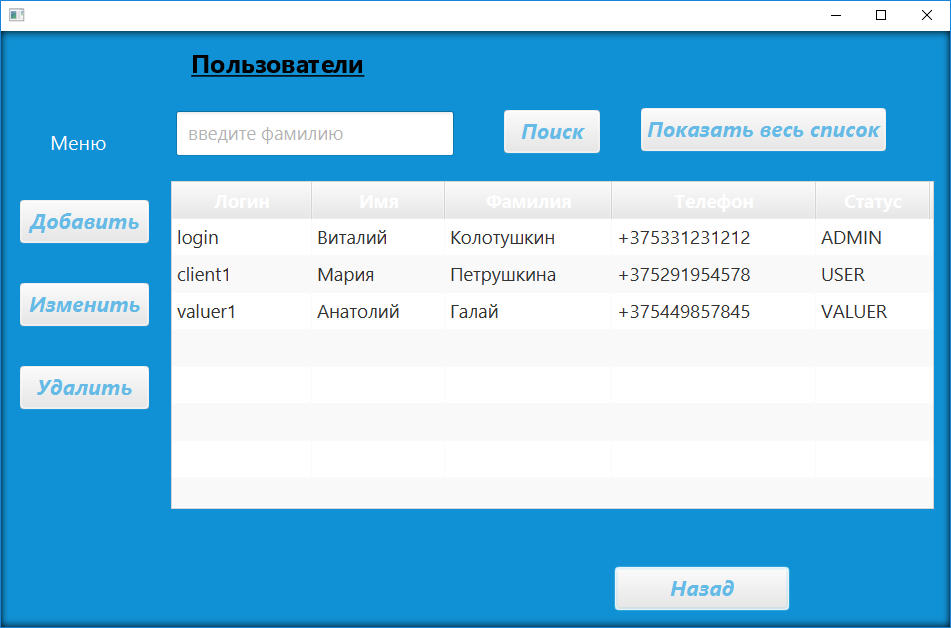


Рисунок 7.16 – Таблица всех пользователей

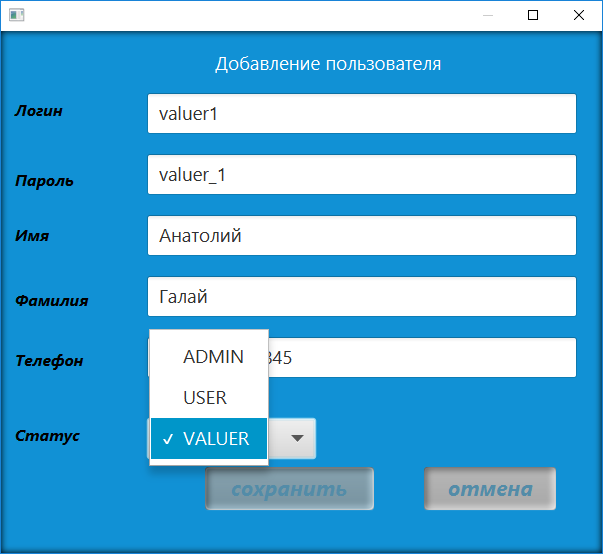


Рисунок 7.17 – Изменение пользователя

Также администратор может работать с таблицей сегментов (рисунок 7.18).

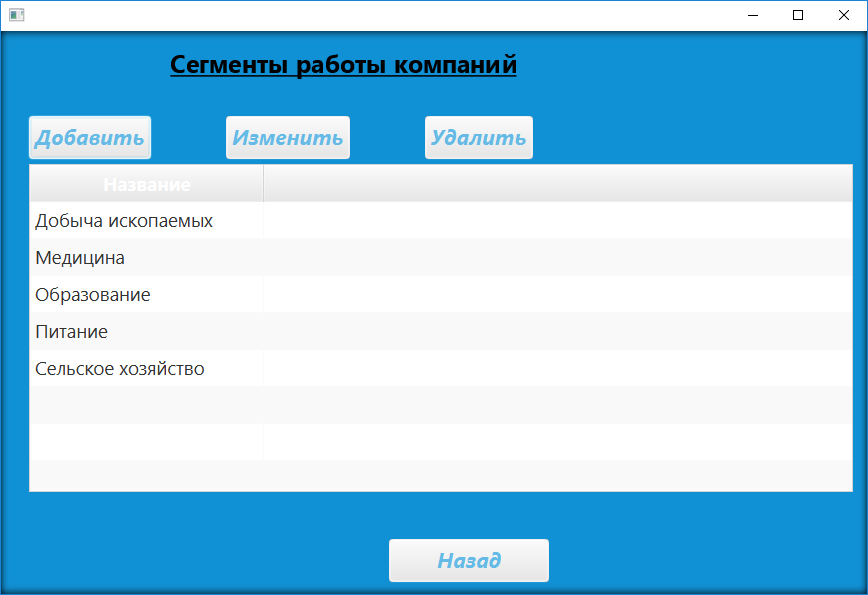


Рисунок 7.18 – Таблица сегментов компаний

Разработанный интерфейс системы является простым, понятным и красивым, что является большим преимуществом. Работать с системой без труда сможет пользователь с любым опытом работы с различными системами, все возможности являются интуитивно понятными.

# **РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ РАЗРАБОТАННОЙ СИСТЕМЫ ОЦЕНКИ КОМПАНИИ НА ОСНОВЕ ПОКАЗАТЕЛЯ СТОИМОСТИ КАПИТАЛА**

Первое, что необходимо сделать при работе с системой – подключиться к серверу, если этого не сделать, на экран будет выведено сообщение об ошибке (рисунок 8.1):

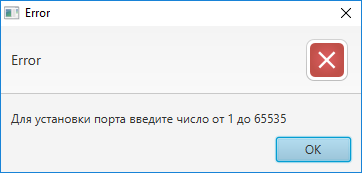


Рисунок 8.1 – Отсутствие подключения к серверу

Если ввести неправильный порт, также будет показано окно с ошибкой (рисунок 2):

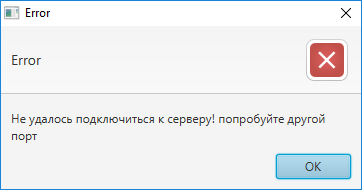


Рисунок 8.2 – Неправильно введенный порт

Если при регистрации не заполнить какое-либо из полей, программа уведомит об этом (рисунок 8.3). Также важно не просто заполнить поля, а заполнить их правильно. Например, при написании телефона не в нужном формате, будет выдан alert, который скажет об этом (рисунок 8.4):

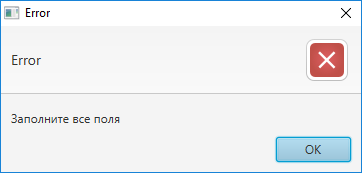


Рисунок 8.3 – Некоторые поля не были заполнены

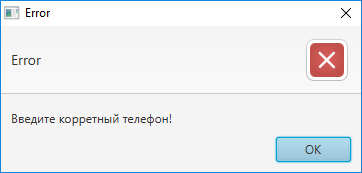


Рисунок 8.4 – Некорректный ввод телефона

Также нельзя зарегистрироваться, используя уже занятый логин (рисунок 8.5):

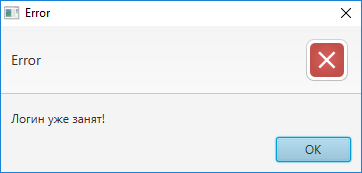


Рисунок 8.5 – Использование занятого логина

При авторизации необходимо ввести как логин, так и пароль, иначе можно увидеть такое сообщение (рисунок 8.6):

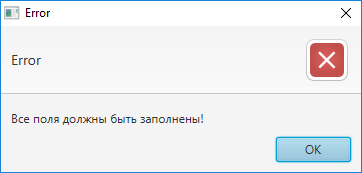


Рисунок 8.6 – Все поля должны быть заполнены

Если при авторизации ввести неправильные данные, войти в личный кабинет не получится (рисунок 8.7):

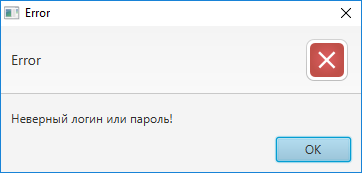


Рисунок 8.7 – Неверный логин или пароль

Если при работе с различными таблицами при желании удалить строку ничего не выбрать, пользователь увидит, что его действия были ошибочными (рисунок 8.8).

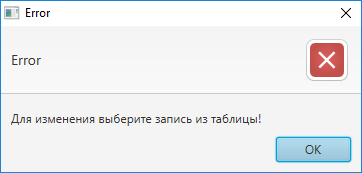


Рисунок 8.8 – Необходимо выбрать строку таблицы

Аналогично авторизации, при добавлении каких-либо данных также необходимо заполнять все доступные поля.

В поля, где подразумеваются только числовые значения, сама система не даст ввести ничего кроме цифр.

Если попытаться изменить пароль, при этом неправильно ввести текущий, программа сообщит об ошибке (рисунок 8.9):

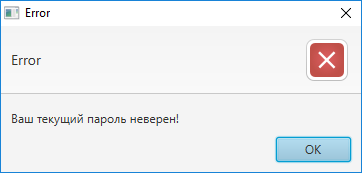


Рисунок 8.9 – Неверный текущий пароль

Также программа сообщит об ошибке, если новый и подтверждающий пароли не совпадут (рисунок 8.10)

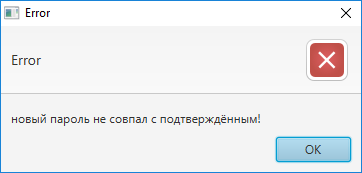


Рисунок 8.10 – Новый пароль не совпал с подтвержденным

Создание ситуаций, в которых потенциально могли возникнуть ошибки, позволило учесть такие моменты и реализовать соответствующие проверки. Таким образом, работа пользователя с системой стала еще удобнее и проще.

# **ЗАКЛЮЧЕНИЕ**

В результате проделанной работы было создано программное приложение, обеспечивающее простой и удобный подход оценивания компании. Целью являлось создание такой программы, которая сможет учитывать все особенности, характеристики и требования этого процесса. Таким образом, основная задача курсового проекта была выполнена.

В ходе создания приложения мною были изучены основные и наиболее значимые правила и требования всех морских перевозок. Необходимо отметить, что данный процесс требует высокого внимания, точности и быстроты действий. Был разработан широкий функционал, который позволит любому пользователю легко разобраться в программе. Поэтому стиль интерфейса создавался, опираясь на массовость потребления.

Основные функции продукта были реализованы в соответствии с актуальными методами и правилами оценки предприятий на основе показателя стоимости капитала. Пошагово было изучено добавление компаний, их оценка, вынесение решения оценщиком. Подводя итог всего вышесказанного отметим, что основной функционал программы заключается в следующем:

* добавление компании
* оценка компании на основе показателя стоимости капитала;
* вынесение оценщиком своего решения, стоит ли инвестировать в ту или иную компанию.

Также реализовано множество проверок, без которых реальное функционирование подобной системы невозможно. Следовательно, одним из отличий данной программы от других является надёжная и безопасная база данных, а также структурированная система учёта всей информации. Благодаря этому потеря либо искажение результатов оценки предприятия буду минимальными.

Курсовой проект был выполнен в соответствии с основными требованиями и положениями. Учтены все необходимые нормы. Для разработки программного продукта использовалась среда разработки Intellj IDEA. В качестве языка программирования был задействован Java. Записка оформлена в соответствии с настоящим ГОСТ – стандарт предприятия.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Капитал предприятия как важнейшая экономическая категория [Электронный ресурс]. – Электронные данные. – Режим доступа: https://works.doklad.ru/view/m8SIaU3B\_MQ.html
2. Стоимость предприятия как показатель оценки компании [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.cfin.ru/appraisal/business/methods/meths\_in\_practice.shtml
3. Оценка бизнеса: Учебник / Под ред. А.Г. Грязновой, М.А. Федотовой. — М.: Финансы и статистика, 2011. [с.19-21].].
4. Необходимость оценки стоимости фирмы [Электронный ресурс]. – Электронные данные. – Режим доступа: https://elib.pnzgu.ru/files/eb/doc/nOXWjFNitJzq.pdf
5. Оценка компании при ликвидации [Электронный ресурс]. – Электронные данные. – Режим доступа: http://ocexp.ru/ocenka-predpriyatiya-pri-likvidacii/].
6. EBITDA [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.fd.ru/articles/158069-formula-i-primer-rascheta-ebitda-na-osnove-buhgalterskih-dannyh
7. Шибилева О. В., Пониматкина Е. Оценка стоимости компании на основе показателя EBITDA // Молодой ученый. — 2012. [Электронный ресурс]. – Электронные данные. – Режим доступа: https://moluch.ru/archive/37/4209/
8. DAO Pattern [Электронный ресурс]. – Электронные данные. – Режим доступа: http://javatutor.net/articles/j2ee-pattern-data-access-object].
9. Паттерн Команда [Электронный ресурс]. – Электронные данные. – Режим доступа: http://cpp-reference.ru/patterns/behavioral-patterns/command/
10. Паттерн Одиночка [Электронный ресурс]. – Электронные данные. – Режим доступа: https://abap-blog.ru/osnovy-abap/shablon-odinochka/
11. Преимущества паттернов [Электронный ресурс]. – Электронные данные. – Режим доступа: http://citforum.ru/SE/project/pattern/#1
12. Нотация IDEF0 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.businessstudio.ru/wiki/docs/current/doku.php/ru/ csdesign/bpmodeling/idef0
13. Бенчмаркинг [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.kpms.ru/General\_info/Benchmarking.htm
14. Отчет по оцениванию [Электронный ресурс]. – Электронные данные. – Режим доступа: https://dic.academic.ru/dic.nsf/anticris/72610
15. Use case diagram [Электронный ресурс]. – Электронные данные. – Режим доступа: https://habr.com/ru/post/47940/
16. Диаграмма последовательности [Электронный ресурс]. – Электронные данные. – Режим доступа: https://flexberry.github.io/ru/fd\_sequence-diagram.html
17. Диаграмма состояний [Электронный ресурс]. – Электронные данные. – Режим доступа: https://itteach.ru/rational-rose/diagrammi-sostoyaniy
18. Диаграмма развертывания [Электронный ресурс]. – Электронные данные. – Режим доступа: https://openu.ru/Books/UML/Deployment\_diagram.asp
19. Диаграмма компонентов [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://kit.znu.edu.ua/iLec/8sem/case_syst/lectures/case-lect3-part6.pdf>

# **ПРИЛОЖЕНИЕ А**

# **(обязательное)**

# **Алгоритмы, реализующих бизнес-логику серверной части**



Рисунок А.1 – Алгоритм авторизации пользователя

# **ПРИЛОЖЕНИЕ Б**

# **(обязательное)**

# **Диаграммы классов**

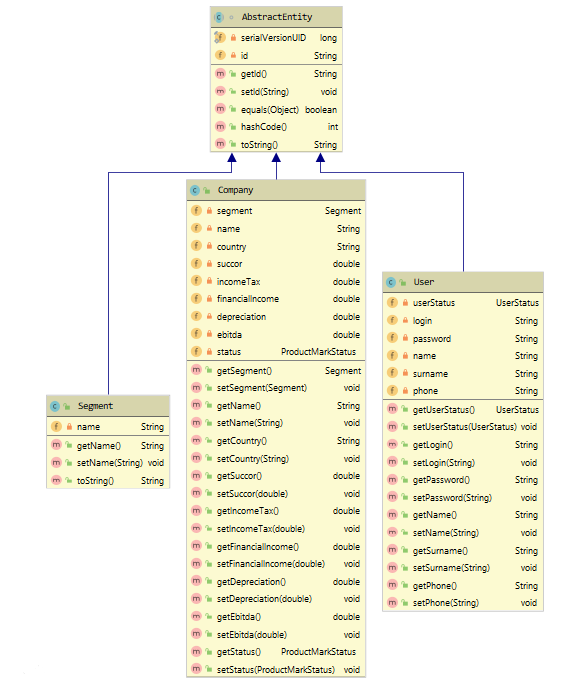


Рисунок Б.1 – Диаграмма основных классов предметной области

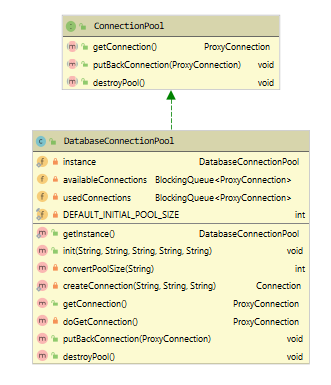


Рисунок Б.2 – Диаграмма ConnectionPool и DatabaseConnectionPool

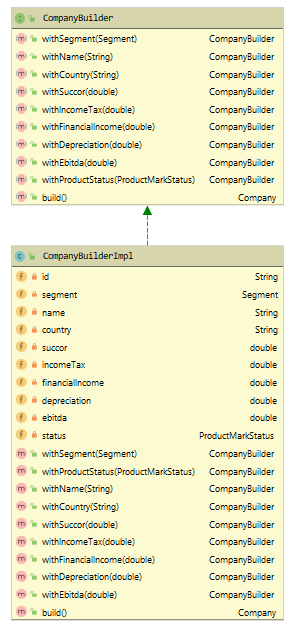


Рисунок Б.3 – CompanyBuilder и CompanyBuilderImpl

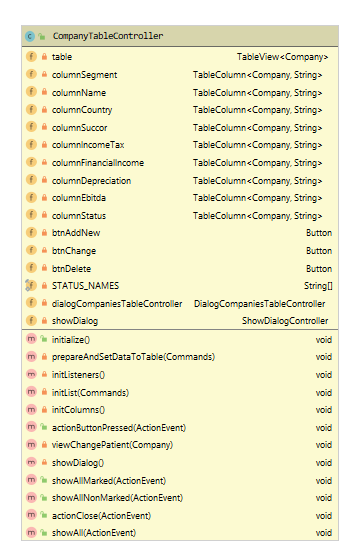


Рисунок Б.4 – Контроллер CompanyTableController

# **ПРИЛОЖЕНИЕ В**

# **(обязательное)**

# **Листинг кода**

# **ПРИЛОЖЕНИЕ Г**

# **(обязательное)**

# **SQL-скрипт для генерации базы данных**

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema company\_valuation\_by\_capital\_cost

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema company\_valuation\_by\_capital\_cost

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `company\_valuation\_by\_capital\_cost` DEFAULT CHARACTER SET utf8 ;

USE `company\_valuation\_by\_capital\_cost` ;

-- -----------------------------------------------------

-- Table `company\_valuation\_by\_capital\_cost`.`segments`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `company\_valuation\_by\_capital\_cost`.`segments` (

`segment\_id` VARCHAR(36) NOT NULL,

`segment\_name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`segment\_id`),

UNIQUE INDEX `category\_name\_UNIQUE` (`segment\_name` ASC) VISIBLE,

UNIQUE INDEX `category\_id\_UNIQUE` (`segment\_id` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `company\_valuation\_by\_capital\_cost`.`companymarkstatus`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `company\_valuation\_by\_capital\_cost`.`companymarkstatus` (

`status\_id` VARCHAR(36) NOT NULL,

`name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`status\_id`),

UNIQUE INDEX `name\_UNIQUE` (`name` ASC) VISIBLE,

UNIQUE INDEX `status\_id\_UNIQUE` (`status\_id` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `company\_valuation\_by\_capital\_cost`.`companies`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `company\_valuation\_by\_capital\_cost`.`companies` (

`company\_id` VARCHAR(36) NOT NULL,

`segment\_id` VARCHAR(36) NOT NULL,

`mark\_status\_id` VARCHAR(36) NOT NULL,

`name` VARCHAR(45) NOT NULL,

`country` VARCHAR(45) NOT NULL,

`succor` DOUBLE NOT NULL COMMENT 'выручка',

`income\_tax` DOUBLE NOT NULL COMMENT 'расходы по налогу на прибыль',

`financial\_income` DOUBLE NOT NULL,

`depriciation` DOUBLE NOT NULL,

`ebitda` DOUBLE NOT NULL COMMENT 'мультипликатор',

PRIMARY KEY (`company\_id`),

INDEX `category\_id\_idx` (`segment\_id` ASC) VISIBLE,

INDEX `mark\_status\_id\_idx` (`mark\_status\_id` ASC) VISIBLE,

CONSTRAINT `segment\_id`

FOREIGN KEY (`segment\_id`)

REFERENCES `company\_valuation\_by\_capital\_cost`.`segments` (`segment\_id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `mark\_status\_id`

FOREIGN KEY (`mark\_status\_id`)

REFERENCES `company\_valuation\_by\_capital\_cost`.`companymarkstatus` (`status\_id`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `company\_valuation\_by\_capital\_cost`.`userstatus`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `company\_valuation\_by\_capital\_cost`.`userstatus` (

`user\_status\_id` VARCHAR(36) NOT NULL,

`name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`user\_status\_id`),

UNIQUE INDEX `user\_status\_id\_UNIQUE` (`user\_status\_id` ASC) VISIBLE,

UNIQUE INDEX `name\_UNIQUE` (`name` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `company\_valuation\_by\_capital\_cost`.`users`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `company\_valuation\_by\_capital\_cost`.`users` (

`user\_id` VARCHAR(36) NOT NULL,

`user\_status\_id` VARCHAR(36) NOT NULL,

`login` VARCHAR(45) NOT NULL,

`password` VARCHAR(45) NOT NULL,

`name` VARCHAR(45) NOT NULL,

`surname` VARCHAR(45) NOT NULL,

`phone` VARCHAR(45) NOT NULL,

PRIMARY KEY (`user\_id`),

INDEX `user\_status\_id\_idx` (`user\_status\_id` ASC) VISIBLE,

CONSTRAINT `user\_status\_id`

FOREIGN KEY (`user\_status\_id`)

REFERENCES `company\_valuation\_by\_capital\_cost`.`userstatus` (`user\_status\_id`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

INSERT INTO `company\_valuation\_by\_capital\_cost`.`userstatus` (`user\_status\_id`, `name`) VALUES ('1', 'admin');

INSERT INTO `company\_valuation\_by\_capital\_cost`.`userstatus` (`user\_status\_id`, `name`) VALUES ('2', 'user');

INSERT INTO `company\_valuation\_by\_capital\_cost`.`userstatus` (`user\_status\_id`, `name`) VALUES ('3', 'valuer');

INSERT INTO `company\_valuation\_by\_capital\_cost`.`companymarkstatus` (`status\_id`, `name`) VALUES ('1', 'Выгодно для инвестиций');

INSERT INTO `company\_valuation\_by\_capital\_cost`.`companymarkstatus` (`status\_id`, `name`) VALUES ('2', 'Невыгодно для инвестиций');

INSERT INTO `company\_valuation\_by\_capital\_cost`.`companymarkstatus` (`status\_id`, `name`) VALUES ('3', 'Затруднился оценить');

INSERT INTO `company\_valuation\_by\_capital\_cost`.`companymarkstatus` (`status\_id`, `name`) VALUES ('4', 'Не оцененно');