

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра Прикладная математика и информатика
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Прикладная информатика в социальной сфере
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка информационной торговой системы для брокерской компании»

Студент

Ф.Р. Акберли
(И.О. Фамилия)

(личная подпись)

Руководитель

к.т.н., О.В. Аникина

(ученая степень, звание, И.О. Фамилия)

Консультант (ы)

А.В. Москалюк

(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

АННОТАЦИЯ

Тема выпускной квалификационной работы: «Разработка информационной торговой системы для брокерской компании».

Целью выпускной квалификационной работы является разработка информационной торговой системы для торгового подразделения брокерской компании.

Пояснительная записка состоит из введения, трех глав, заключения и приложений. Во введении раскрывается общее состояние проблемы, описание предмета и объекта проекта, обосновывается актуальность темы, представлена новизна проекта, определяются цель и задачи проекта, практическое применение результатов проекта. В первой главе дается характеристика текущего состояния фондового рынка, приведен анализ тенденций развития инвестиционной сферы, приводится аналитическое описание правил российского фондового рынка, приведено описание объекта управления, раскрывается экономическая сущность комплекса решаемых задач, а также дается обоснование проектных решений по выбору технического, программного и технологического обеспечения. Вторая глава содержит непосредственную реализацию проектных решений, определенных в первой части. В третьей главе представлены результаты тестирования разработанной информационной торговой системы. В заключении содержатся результаты, полученные в ходе выполнения исследования основных положений бакалаврской работы, даны рекомендации по возможному их использованию в практической деятельности управляющих компаний и финансовых корпораций. В приложении представлены фрагменты кода.

В работе представлено 12 таблиц, 21 рисунок, список использованной литературы содержит 20 источников. Общий объем выпускной квалификационной работы составляет 74 страницы.

ABSTRACT

The theme of the final qualification work: “Development of an information trading system for a brokerage company”.

The aim of the final qualification work is to develop an information trading system for a trading division of a brokerage company.

The explanatory note consists of an introduction, three chapters, conclusion and annexes.

The introduction reveals the general state of the problem, it contains a description of the subject and the object of this work. There is reasonable relevance of the topic, novelty of the project, defined goals and objectives, practical application of the project results.

The first chapter presents a description of the current state of a stock market, an analysis of the trends in the development in the investment sphere, an analytical description of the rules of the Russian stock market. There is also a description of the control object, a revealed economic essence of the complex of tasks to be solved, a justification of the decisions on the choice of technical, software and technological support.

The second chapter contains the direct implementation of the project decisions defined in the first part.

The third chapter presents the test results of the developed information trading system.

The conclusion contains the results obtained in the course of the study of the main provisions of the bachelor’s work. There are also on their possible use in practical activities of management companies and financial corporations.

The appendix contains some code fragments.

In this work there are 12 tables, 21 pictures, a list of used literature, consisting of 20 resources. The total volume of the graduation qualification work is 74 pages.

ОГЛАВЛЕНИЕ

Введение.....	6
Глава 1 Анализ бизнес-процесса проведения торгов на биржевом рынке	9
1.1 Характеристика брокерской компании.....	9
1.1.1 Краткая характеристика торгового подразделения.....	10
1.1.2 Описание информационного и технического обеспечения предприятия.....	11
1.1.3 Выявление существующих бизнес- процессов в брокерской компании.....	13
1.2 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»	16
1.3 Требования к функциональности информационной системы.....	19
1.4 Анализ существующих ИТ-решений	22
1.4.1 Обоснование необходимости и цели использования вычислительной техники для решения комплекса задач	25
1.4.2 Экономическая сущность комплекса задач	27
1.5 Выбор методологии проектирования и архитектуры ИТС	28
1.6 Разработка модели бизнес-процесса «КАК БУДЕТ»	34
1.7 Разработка прототипа интерфейса пользователя	42
1.8 Постановка задачи на разработку информационной торговой системы	45
Выводы по главе 1.....	46
Глава 2 Проектирование информационной торговой системы.....	47
2.1 Выбор платформы разработки.....	47
2.2 Описание и анализ технологии разработки.....	48
2.3 Выбор СУБД и средств объектно-реляционного отображения	52
2.4 Проектирование базы данных информационной торговой системы предприятия	54
2.5 Разработка основных форм приложения	58

Выводы по второй главе.....	64
Глава 3 Тестирование разработанной информационной торговой системы.....	65
3.1 Описание процесса тестирования	65
3.2 Оценка качества пользовательского интерфейса	66
3.3 Требования к программно-аппаратным средствам ИТС	69
Выводы по третьей главе	70
Заключение	71
Список используемой литературы	73
Приложение А Сценарии прецедентов.....	75
Приложение Б Фрагмент программного кода.....	81
Приложение В Скрипты создания таблиц базы данных.....	85

Введение

Последние годы стремительной жизни российского фондового рынка показали впечатляющие результаты: появление современной технологической и регулятивной инфраструктуры, различные циклы – стремительный рост и рекорды падения, становление сообщества профессиональных участников при отсутствии мошеннических схем; модернизация организаторов торговли и как следствие – новые типы бирж, ставших примером для других и национальной отличительной чертой российской экономики; появление доверия к ценным бумагам как инструменту привлечения денежных потоков – инвестиций, создания способов оценки имущественных прав собственности, составляющих основу принятой политики государства в рамках социально-ориентированной рыночной экономики. Не уменьшились риски, присутствующие на фондовом рынке, на сегодняшний день рынок не стал менее волатильным. Как и раньше, конъюнктура рынка акций формировалась денежными ресурсами спекулятивно настроенных инвесторов.

Приведенные аргументы показывают, что необходимо повысить доступность (низкий порог входа) и качество (информационная открытость и финансовая грамотность), снизить транзакционные издержки, это задачи, которые необходимо решать в комплексе, применяя современные информационные технологии в инфраструктуре управляющих компаний.

Новые рекорды, достигнутые информационными технологиями, стремительно меняют отношение к биржевому рынку, время формирования рыночной сделки измеряется в секундах. Все большее понимание появляется у профессиональных участников, они начинают отдавать предпочтение высокотехнологичным решениям ведения торгов, инструментами предлагаемыми организаторами торгов и своими собственными разработками. Управляющие компании уже имеют возможность получать рыночную информацию в реальном времени, но на сегодняшний день встает главный вопрос, как быстро она обновляется.

Существенные законодательные различия и высокая ресурсоемкость внедрения не позволяют использовать зарубежные решения, вынуждая проводить исследовательские и экспериментальные работы самостоятельно. Именно поэтому разработка практических рекомендаций и методик автоматизации торгов на биржевом рынке являются актуальным и значимым для российского фондового рынка и экономики в целом проектом.

Предмет исследования – процесс детального формализованного описания и адаптации финансовых схем учета, научного обоснования методических подходов к проектированию и разработке информационных торговых систем (ИТС), автоматизирующих деятельность на фондовом рынке, их внедрению в бизнес-процессы и интеграции их в единое информационное пространство корпораций. **Объектом** исследования является разрабатываемая информационная торговая система.

Новизна работы заключается в необходимости детализированной, качественной постановки задачи на разработку информационной торговой системы, что требует комплексного обоснованного подхода к моделированию и описанию правил функционирования настоящей системы в рамках торговли на биржевом рынке.

Целью выпускной квалификационной работы является разработка информационной торговой системы для торгового подразделения брокерской компании. Для достижения поставленной цели необходимо решение следующих **задач**:

- изучить процесс ведения вопросов торгов на биржевом рынке и выявление недостатков существующей организации обработки рыночной информации, определяющих необходимость разработки данного проекта;
- представить обоснование выбора основных проектных решений;
- спроектировать информационную торговую систему;
- разработать все виды обеспечивающих подсистем;
- протестировать разработанную информационную торговую систему.

Пояснительная записка состоит из введения, трех глав, заключения и приложений.

Во введении раскрывается общее состояние проблемы, описание предмета и объекта проекта, обосновывается актуальность темы, представлена новизна проекта, определяются цель и задачи проекта, практическое применение результатов проекта;

В первой главе дается характеристика текущего состояния фондового рынка, приведен анализ тенденций развития инвестиционной сферы, приводится аналитическое описание правил российского фондового рынка, приведено описание объекта управления, раскрывается экономическая сущность комплекса решаемых задач, а также дается обоснование проектных решений по выбору технического, программного и технологического обеспечения.

Вторая глава содержит непосредственную реализацию проектных решений, определенных в первой части.

В третьей главе представлены результаты тестирования разработанной информационной торговой системы.

В заключении содержатся результаты, полученные в ходе выполнения исследования основных положений бакалаврской работы, даны рекомендации по возможному их использованию в практической деятельности управляющих компаний и финансовых корпораций.

Работа изложена на 72 страницах, содержит 21 рисунок и 12 таблиц.

Глава 1 Анализ бизнес-процесса проведения торгов на биржевом рынке

1.1 Характеристика брокерской компании

Для исследования была выбрана среднестатистическая брокерская компания, осуществляющая профессиональную деятельность на рынке доверительного управления. Главная задача - эффективное управление активами клиентов. Передача средств в доверительное управление - это установление нового уровня отношений между компанией и её клиентами, отношений на принципах взаимовыгодного сотрудничества и партнёрства. Эти отношения подразумевают, что цели и задачи клиентов компания воспринимает как свои собственные. Компания готова делиться своим опытом, работать на благо своих клиентов и рассчитывать на ответное доверие с их стороны. Компания старается максимально полно раскрывать клиентам особенности своей работы. Компания всегда говорит, что, когда и почему она делает. Поэтому один из главных приоритетов в работе компании - своевременная, качественная и полная отчетность, управление активами, инвестиции и вложения (паевые инвестиционные фонды, закрытые паевые инвестиционные фонды).

Принципиальная стратегия работы с пенсионными резервами обозначена предприятием как консервативная, направлена на максимальную надежность размещения средств, передаваемых в доверительное управление. Доверие со стороны клиентов физических лиц можно объяснить правильно выбранной инвестиционной стратегией компании. Признавая важность создания эффективной системы корпоративного управления для успешного ведения бизнеса и для достижения взаимопонимания с заинтересованными сторонами, компания руководствуется в своей повседневной деятельности основными принципами корпоративного поведения, основанными на стандартах Кодекса корпоративного поведения [6].

Торговое подразделение, которое курирует все основные торговые операции при осуществлении инвестиционной деятельности, в том числе и ведет разработку информационной торговой системы (ИТС). Это

направление - является одним из объектов разработки систем управления качеством доверительного управления, в целом следуя цели автоматизации торгов на биржевом рынке ценных бумаг.

1.1.1 Краткая характеристика торгового подразделения

Торговое подразделение компании за прошедшие годы стало фактически одним из приоритетных направлений в компании и располагает собственными информационными мощностями выраженным в отсутствии ограничения по конечному числу клиентов. На настоящий момент занято обслуживанием двух Негосударственных пенсионных фондов (НПФ), шести паевых инвестиционных фондов (ПИФ) разных типов (открытые, интервальные, закрытые), а также тех проектов, которые не могут быть освоены на основной площадке компании по экономическим причинам.

Задачами торгового подразделения компании являются разработка и «запуск» стратегий торговли, а также отработка технологии механических торговых систем (МТС) для торговли большими объемами или торговля с высокой скоростью осуществления торговых операций. База торгового подразделения включает в себя все бизнес процессы для создания и «запуска» стратегии торговли. Лицензионное программное обеспечение, собственные разработки компании, материальная и техническая база для осуществления торговых операций, договорные отношения от лица компании со всеми необходимыми организациями для проведения торговых операций в реальном времени (специализированный депозитарий, регистратор, оценщик, брокер на фондовом рынке, Интернет компании, договоренности с другими посредниками).

В торговом подразделении компании разработана и внедрена схема сборки и ввода в эксплуатацию стратегий совершения торговых операций с сохранением основных принципов торговли – отвечают требованиям клиентов. Здесь отрабатываются процессы торговли, оценивается качество проектирования и изготовления стратегий торговли. Система менеджмента

качества торгового подразделения соответствует нормам внутреннего корпоративного регламента.

Организационная структура торгового подразделения состоит из следующих подразделений: Сервис поддержки по лицензионному программному обеспечению, Инвестиционный комитет, Комитет по принятию решений, Управляющий активами, Трейдеры, Менеджмент качества.

1.1.2 Описание информационного и технического обеспечения предприятия

Торговое подразделение располагает необходимыми производственными ресурсами, площадями, технологическими, контрольными, испытательными стендами и квалифицированными сотрудниками для выполнения всех процессов на этапах жизненного цикла создания стратегии торговли и осуществлении торговых операций.

Существующая информационная система накапливает и перерабатывает поступающую первичную информацию, существующие регламенты и бизнес процессы в аналитическую информацию, фундамент для развития торгового подразделения, корректировки его целей и вывода планов для нового витка цикла.

В зависимости от степени автоматизации информационных процессов в информационной торговой системе торгового подразделения информационные системы определяются как ручные, автоматические, автоматизированные.

В организации в настоящее время учет сделок ведется с помощью табличного процессора Microsoft Excel, 1С: Предприятие и информационной торговой системой QUIK. Это позволяет осуществлять бесперебойную работу на существующих мощностях и имеющимся количеством поставленных заказчиком клиентов. На основе существующих мощностей анализировать деятельность работы очень сложно, так как в них отсутствуют

необходимые инструменты для реализации торговых стратегий. Ручная и наполовину автоматизированная система не позволяет быстро и качественно подготавливать площадку для тестирования, анализа и оценки тех или иных механических торговых систем. Такой процесс является трудоемким и ненадежным, если не сказать невозможным.

Разработка проекта автоматизации торгов на биржевом рынке ценных бумаг является в настоящее время актуальной необходимостью, так как позволит руководящему блоку управляющей компании расширить линейку предлагаемых инвестиционных продуктов, оперативно получать информацию для анализа и корректировки работы торгового подразделения, а также рассчитывать такие показатели, как рентабельность работы торговой стратегии, соотношение рисков и инвестиционных вложений. Только при наличии такого инструмента руководящий состав сможет своевременно получить материалы для оценки работы торгового подразделения, что в конечном итоге повысит эффективность работы компании в целом.

Законодательная база – это основа работы по созданию регламентированной деятельности торгового подразделения, в которой описываются основные понятия и отношения, возникающие при совершении сделок с ценными бумагами, а также особенности деятельности управляющих компаний, как профессиональных участников рынка ценных бумаг.

Наряду с интенсивным развитием вычислительных средств и систем передачи информации все более актуальной становится проблема обеспечения её безопасности. Меры безопасности направлены на предотвращение несанкционированного получения информации, физического уничтожения или модификации защищаемой информации.

Для защиты информации требуется не просто разработка частных механизмов защиты, а организация комплекса мер, то есть использование специальных средств, методов и мероприятий с целью предотвращения потери информации.

Наиболее простой способ защиты информации – применение паролей. Пользователь вводит свой пароль в открытом виде, компьютерная программа кодирует его, сравнивает с хранящимся эталоном и предоставляет доступ к информации, если пароль введен правильно.

Необходимо отметить, что особую опасность в настоящее время представляет проблема компьютерных вирусов, ибо эффективной защиты против них разработать не удалось. Остальные пути несанкционированного доступа поддаются надежной блокировке при правильно разработанной и реализуемой на практике системе обеспечения безопасности.

Для защиты от вирусов следует систематически проверять компьютер на наличие вирусов с помощью специальных антивирусных программ, а также проверять программы и данные, переписываемые на диск, в случае малейших подозрений об источнике. Сегодня вирусописатели используют все более и более сложные методы, чтобы получать доступ к чужим данным и предпринимать удаленные атаки. Поэтому корпоративная сеть должна быть обеспечена защитой на всех уровнях.

1.1.3 Выявление существующих бизнес- процессов в брокерской компании

Бизнес-процесс — это совокупность взаимосвязанных мероприятий или задач, направленных на создание определенного продукта или услуги для потребителей. Для наглядности бизнес-процессы визуализируют при помощи методологий проектирования [4].

Бизнес-моделированием также называют дисциплину и отдельный подпроцесс в разработке программного обеспечения, в котором описывается деятельность компании и определяются требования к системе - те подпроцессы и операции, которые подлежат автоматизации в разрабатываемой информационной системе [4].

В качестве предметной области в данной работе рассматривается отдельный вид деятельности. Целью функционирования компании –

получение вознаграждения от эффективного доверительного управления на рынке коллективных инвестиций. Доверительное управление переданными средствами клиентов реализуется через отдельный вид работ управляющей компании – проведение торгов на биржевом рынке ценных бумаг.

В рамках данной работы задача автора при анализе бизнес-процессов посвящена формализации, выявлению и устранению выявленных недостатков деятельности компании, выбору автоматизируемых функций и информационного обеспечения, служащих основой для дальнейшего проектирования структуры информационной торговой системы. Дается описание предметной области; производится выбор и описание автоматизируемых функций; выполняется первичное описание информационного обеспечения; выполняется постановка задач проекта; рассматриваются специфические требования к системе: функциональные и нефункциональные требования; предоставляется перечень неэффективности действующей системы, описание таблицы перехода: недостатки – решение; построение формализованной модели «как есть»; выбору путей решения и как итог, модель «как должно быть».

Информация получена в результате анализа опыта работы в соответствующей организации и анализа предполагаемой работы информационной торговой системы (ИТС) на основе оценки уже существующих систем. Обследование реальных или типовых объектов данного класса производилось, все достоинства и недостатки представлены в данной работе.

Направление, в которой функционирует информационная торговая система (ИТС) является моделью взаимодействия, полагающая, что центром всей работы является информация торгов на бирже, а основными направлениями деятельности являются меры по поддержке эффективного управления и обеспечения высокого уровня риск-менеджмента. Поддержка этих бизнес целей в настоящей информационной торговой системе (ИТС) включает ввод заявок, сбор, хранение, изменение и анализ информации об

истории сделок, изменение цен инструментов торговли (ставки, рыночные цены передают представители бирж – брокеры), а также о внутренних процессах, происходящих в компании с денежными потоками. Информационная торговая система (ИТС) показывает пример технологии для поддержки модели биржа-брокер-клиент, необходимой для организации брокерского обслуживания через удаленные пользовательские терминалы и соответствующего программного обеспечения.

Среда функционирования информационной торговой системы (ИТС) включает отдел в управляющих компаниях, коммерческих банках и финансовых компаниях, занимающиеся предоставлением услуг доверительного управления (система отношений, при которой имущество, изначально принадлежащее учредителю, передается в распоряжение доверительного собственника (управляющего или попечителя), но доход с него получают выгодоприобретатели (бенефициары)).

Объект управления представляет собой информацию о денежных средствах, подлежащие выдаче или возврату, и/или инвестированных в инструменты торговли на рынке ценных бумаг; информация об истории сделок; трейдеры (биржевой торговец; подразумевается торговля ценными бумагами (акциями, облигациями, фьючерсами, опционами) на фондовой бирже), которые работают в данной компании; информация о ценных бумагах с которыми работает отдел доверительного управления.

Субъект управления представляет собой персонал, работающий в данной компании и обеспечивающий функционирование информационной торговой системы (ИТС).

Для выявления недостатков существующей информационной торговой системы необходимо составить формальное описание бизнес-процессов, используя одну из методологий проектирования информационных систем. Выявление бизнес-процессов представлено в следующем параграфе через анализ диаграммы организации бизнеса «как есть».

1.2 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»

Для описания модели информационной системы «как есть», т.е. отображающее настоящее состояние, воспользуемся нотацией UML. Диаграмма вариантов использования описывает взаимоотношения и зависимости между группами вариантов использования и действующими лицами, участвующими в процессе. Вариант использования описывает, с точки зрения действующего лица, группу действий в системе, которые приводят к конкретному результату. 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.

Диаграмма вариантов использования «как есть» представлена на рисунке 1.1.

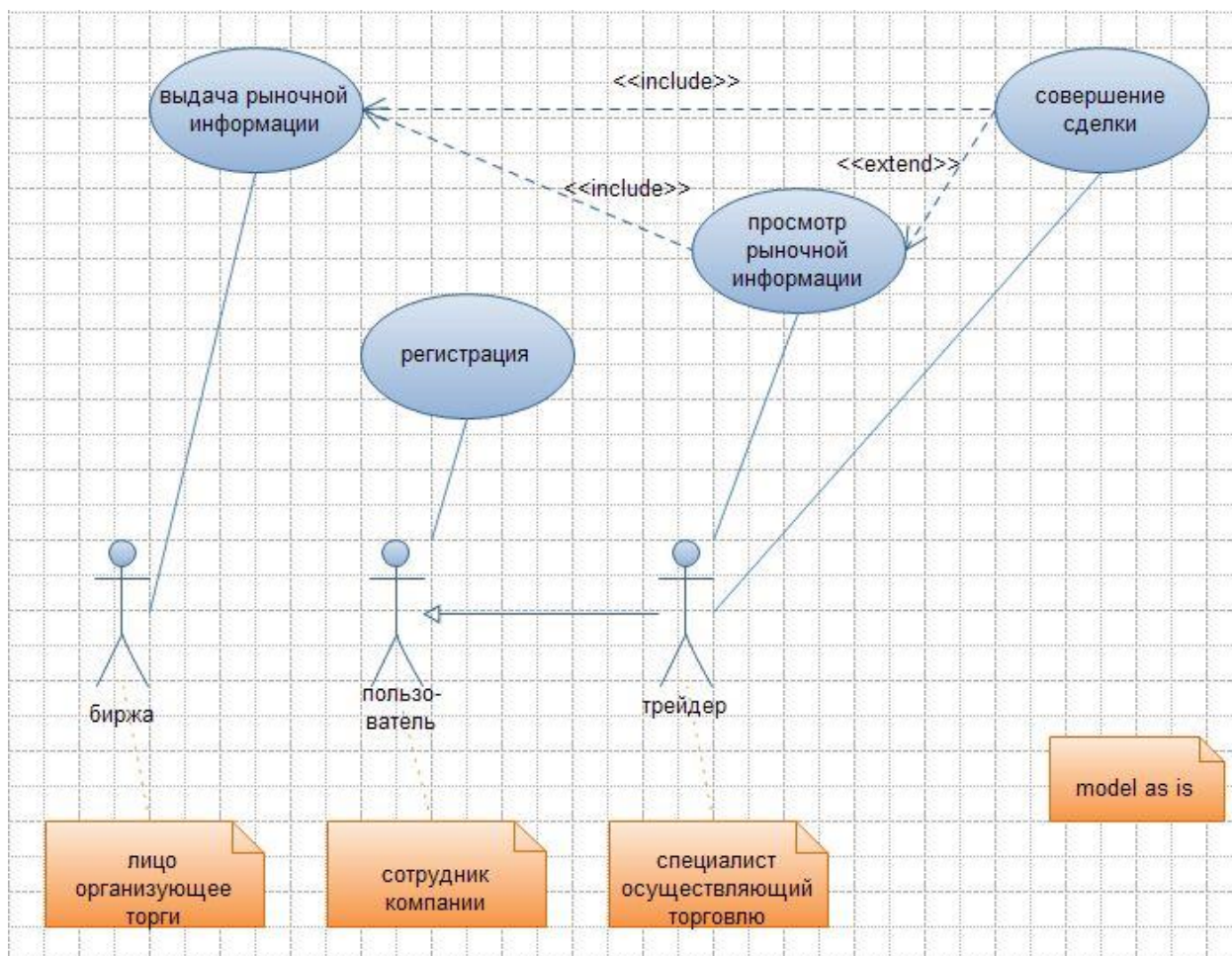


Рисунок 1.1 - Диаграмма вариантов использования «как есть»

«Биржа внешний источник» передает рыночную информацию в информационную систему компании посредством варианта использования «Выдача рыночной информации». Сотрудник компании «Пользователь», специалист отдела, предоставляющего услугу доверительного управления, взаимодействует с информационной системой компании через вариант использования «Регистрация». Специалист «Трейдер», осуществляющий торговлю финансовыми инструментами, денежными средствами, переданными компании через услугу доверительного управления. Трейдер в информационной системе компании – это пользователь удачно прошедший регистрацию. Трейдер взаимодействует с информационной системой через вариант использования: «Просмотр рыночной информации» включающий вариант использования «Выдача рыночной информации». «Совершение сделки» включающий вариант использования «Выдача рыночной информации», возможность включение варианта использования «Просмотр рыночной информации».

Цель и сценарии включенного варианта использования должны иметь смысл независимо друг от друга, чтобы их можно было включать в варианты использования, создаваемые позже.

Смысл включения варианта использования «Выдача рыночной информации» в «Просмотр рыночной информации», что в интерфейсной части при просмотре рыночной информации невозможно без выдачи рыночной информации. Очевидно, что и «Совершение сделки» невозможно без «Выдача рыночной информации» т.к. смысл действия совершения рыночной операции невозможно без конкретной рыночной информации, но не обязательно включает «Просмотр рыночной информации», т.е. визуально пользователю системы не обязательно видеть текущую ситуацию.

Благодаря гибкому подходу в анализе модели «как есть» и анализ предметной области в целом были выявлены недостатки текущей работы в торговом подразделении (Таблица 1.1).

К обработке информации в ИТС предъявляются следующие стандартные требования, предъявляемые в предметной области:

- достоверность информации для реализации торговых стратегий;
- время задержки предоставления входной и выходной информации;
- соответствие внутрикорпоративным регламентам;
- обеспечение полноты информации;
- реализованные затраты при обработке информации не должны превышать получаемый экономический эффект – проект должен быть рентабельным;
- способность изменяться в соответствии с изменяющимся информационным полем, где работает система.

В основном в организации учет сделок ведется с помощью табличного процессора Microsoft Excel; 1С Предприятие и при помощи информационной торговой системы QUIK. В последнее время, в связи с ростом объемов инвестиционного портфеля доверительного управления, растущие запросы к надёжности, и вышеописанный анализ, как следствие повышенные требования к скорости обмена данными возникла потребность в модернизации существующей информационной системы.

В модели информационной системы «как есть» составные части задачи автоматизации решены частично или не решены совсем (Таблица 1.1).

Таблица 1.1 - Задача автоматизации

Порядковый номер	Объект автоматизации	Состояние автоматизации «как есть»
1	Трансферт рыночной информации	Частично решена
2	Анализ инвестиционного портфеля	Не решена
3	Процесс совершения сделки	Частично решена
4	Процесс риск - менеджмента	Не решена

Определение путей устранения недостатков существующей технологии.

Наиболее полно преимущества от использования информационных торговых систем проявляются при получении доступа профессиональными

участниками к таким системам благодаря собственным мощностям для разработки или покупки у компаний разработчика. При этом широкое распространение информационных торговых систем позволяет участникам рынка – управляющим компаниям решить большой комплекс задач:

- получить неограниченное количество клиентов;
- расширить линейку предлагаемых инвестиционных продуктов и решений;
- обеспечить подключение специалистов, использующих удалённый способ работы;
- расширить возможности по количеству торговых площадок для совершения рыночных операций и использование неограниченного количества инструментов;
- снизить издержки при работе с клиентами;
- свести к нулю потери от операционного риска;
- реализовать возможность реализации торговых стратегий (механических торговых систем).

Наблюдаемый активный рост числа систем Интернет-трейдинга на биржевом рынке ценных бумаг обуславливает начало активной конкурентной борьбы в сфере создания информационных торговых систем. Условиями, влияющими на выбор профессиональными участниками той или иной системы, служат не только технические детали реализации системы, но и функциональные возможности, прозрачность алгоритмов работы, качество и надёжность информационной торговой системы в целом.

1.3 Требования к функциональности информационной системы

Требования к программному обеспечению - совокупность утверждений относительно атрибутов, свойств или качества программной системы, подлежащей реализации.

Основные цели автоматизации (нефункциональные требования):

- Надежность (Связь; Колокация; Тестирование; Прозрачность; Ведение логов).

- Устойчивость (Кэширование; Резервирование; Механизм транзакций).

- Масштабируемость (Параллельность выполнения процессов; дополнительные ресурсы доступа к рыночной информации: quik (программный комплекс для организации доступа к биржевым и внебиржевым торговым системам в режиме реального времени), fix protocol (протокол передачи данных, являющийся международным стандартом, для обмена данными между участниками биржевых торгов в режиме реального времени)).

- Гибкость (Модульная система).

- Простота (Помощь; Ссылки).

- Доступность (Тонкий - клиент: web browser и толстый – клиент swing (библиотека для создания графического интерфейса на языке Java) java (объектно-ориентированный язык программирования)).

- Скорость (Стремящая к минимуму обработка транзакций (в качестве эталона некий бенчмарк (benchmark, контрольная задача, необходимая для определения сравнительных характеристик производительности компьютерной системы); трансферт рыночной информации).

Автоматизация технологических процессов информационной торговой системы призвана обеспечить следующие возможности (функциональные требования):

- Контроль рисков (Рыночный риск (англ. Market risk); Риск концентрации на отдельном инструменте (англ. Concentration risk); Риск ликвидности (англ. Risk of liquidity); Операционный риск (англ. Operational risk); Валютный риск (англ. Currency risk); Процентный риск (англ. Percentage risk).

- Торговля «на» бирже (колокация).

- Исполнение умных ордеров (англ. smart order).
- Обезличивание счетов/инвестиционных портфелей.
- Механические торговые системы (МТС).
- Работа со стратегиями торговли.
- Аналитический модуль. (возможность проведения анализа).
- Контроль состояния инвестиционного портфеля.
- Трансферт рыночной информации.
- Анализ инвестиционного портфеля.
- Процесс совершения сделки.
- Процесс риск - менеджмента.

В таблице 1.2 иллюстрированы пути перехода к желаемому состоянию системы по основным направлениям.

Таблица 1.2 - Таблица перехода

Объект автоматизации	Недостаток	Решение
Трансферт рыночной информации	Разрозненность терминалов брокеров.	Использовать единый сервис ввода заявок для разных клиентов у разных брокеров.
	Ограниченная функциональность терминалов брокеров.	Создание системы «умных» ордеров (smart order).
Анализ инвестиционного портфеля	Настройка, здоровье, квалификация и талант специалистов участвующих в ИТС.	Масштабировать принятие торговых решений в режиме реального времени (англ. on-line).
	Отсутствие инструментария анализа.	Создание механизма сбора рыночной информации и обработки её в ИТС.
Процесс риск-менеджмента	Низкая скорость принятия трейдером решений.	Создать систему учета рисков (СУР).
Процесс совершения сделки	Низкая диверсификация направлений торговли.	Обеспечить возможность создания множества торговых стратегий.
Процесс совершения сделки	Количество используемых инструментов.	Масштабировать ресурсы ИТС для работы трейдера с группой счетов или/и ценных бумаг.

Предлагаемая таблица не только описывает рамки комплекса задач, но является наглядным показателем формулировки требований к настоящему

проекту, показателем будущего видения решений, которые описывают специфичные требования к информационной торговой системе при первом приближении.

1.4 Анализ существующих ИТ-решений

В настоящее время рынок удаленного доступа к биржевым торгам для всех желающих достаточно многообразен, что даже создает для неподготовленного специалиста некоторые трудности с выбором того или иного программного комплекса.

В качестве примера рассмотрим программный комплекс QUIK. QUIK (Quickly Updatable Information Kit – быстро обновляемый информационный комплект, англ.) - программный комплекс для организации доступа к биржевым и внебиржевым торговым системам в режиме реального времени. Программный комплекс состоит из серверной части и рабочих мест (терминалов) клиента, взаимодействующих между собой через Интернет.

Изначально QUIK являлся информационной системой, отличавшейся высокой скоростью доставки данных, что и отразилось в названии программы. Сейчас QUIK представляет собой многофункциональную информационно-торговую платформу с различными областями применения. QUIK сегодня — наиболее популярная система для организации брокерского обслуживания через удаленные пользовательские терминалы (Интернет-трейдинга) в России, которая применяется более чем 200 брокерами для обслуживания десятков тысяч клиентов.

Программный комплекс QUIK может применяться для различных целей, предусмотренных используемой конфигурацией.

Проведение собственных операций и брокерское обслуживание клиентов возможно в двух вариантах. Вариант QUIK-Брокер предполагает покупку брокером всего программного комплекса (сервера и рабочих мест клиентов) и его самостоятельную эксплуатацию на собственной технологической базе. Вариант аутсорсинга брокерской системы позволяет

использовать программный комплекс QUIK, размещенный в одном из Технических центров ARQA Technologies.

Проведение собственных операций банка или инвестиционной компании без необходимости обслуживания клиентов предусмотрено системой QUIK-Дилер. QUIK-Дилер — это экономичное решение для участников биржевых торгов, заинтересованных в снижении затрат на технологический доступ к биржевым торговым системам.

Учебная торговая система QUIK-Junior предназначена для овладения навыками биржевой торговли и приемами работы в системе QUIK. QUIK-Junior активно применяется высшими учебными заведениями, брокерами для подготовки клиентов, а также в качестве тестового полигона для отладки собственных программ.

Основная цель создания любой информационной системы - создание такого программного обеспечения, который помогает специалисту выполнять свои повседневные задачи. Для создания новой информационной системы автор работы первым делом определился с требованиями (объекты автоматизации), которым должна удовлетворять система. После анализа предметной области автор составил эти требования, в нашем случае это список функций, по которому трудно судить будет ли будущая информационная система выполнять свое назначение и сможет ли она облегчить специалисту выполнение его работы в принципе.

Главным тезисом, этого параграфа – это понимание создания собственной разработки, т.к. существующий аналог информационная торговая система Quik не подходит по параметрам или по требованиям к будущей системе описанные выше.

Учитывая специфику работы на биржевом рынке и достаточную большую стоимость готовых систем, которые надо в последующем адаптировать под конкретные задачи, автор выбрал путь разработки собственной информационной торговой системы на мощностях торгового подразделения брокерской компании.

При существующей сложности настоящего проекта необходимо конкретизировать комплекс задач, через анализ ситуаций в сегменте рынка ИТС. Иначе, не анализ конкретной компании, а анализ профиля рынка в целом. Автор считает, если предлагаемая информационная торговая система найдёт применение у всех заинтересованных профессиональных участников, то и анализ стоит проводить не отдельной управляющей компании, а анализ ситуации на сегодняшний день. Итак, в бизнесе разработок встраиваемых модулей для автоматизации торговли ценными бумагами, учитывая специфику плагинов (от англ. plugin – подключаемый) торговых серверов, можно услышать следующее: «это у Вас работать не будет, так как если это возможно, то все давно бы внедрились такую разработку» автор предлагает вспомнить историю, времена телеграфа. В то время переход от телеграфа к телефонии был ясен и обоснован. Время, затрачиваемое на получение телеграфной ленты, формулировки торгового решения, не сравнить с разговором по телефону. Следующий скачок технологии учитывая специфику биржевого рынка, это несомненно появление Интернета. Возьмём за образец вышеописанный пример, станет ясен переход от телефонии к Интернету: время, затрачиваемое на постановку торгового решения по телефону и время на мониторинг рыночной ситуации в режиме реального времени не сравнить по скорости.

В современной биржевой торговле осуществление операций с ценными бумагами может осуществляться не только непосредственно человеком, но и в автоматическом (точнее автоматизированном) режиме – для этого есть как математические (возможность разработать и применить разнообразные стратегии поведения на рынке различной степени сложности), так и технические предпосылки (развитые системы брокерского обслуживания и Интернет-трейдинга).

Программный модуль автоматизированной торговли называется торговым роботом, а если он встраивается в программную систему торгового сервера, используется термин – плагин.

Плагин служит двоякой цели: с одной стороны, он реализует какую-то уникальную торговую стратегию или\и обеспечивает взаимодействие с программной системой торгового сервера.

Разрабатываемый плагин – средство создания торговых роботов. Обеспечивает разработчика унифицированными, надежными и эффективными интерфейсами взаимодействия с торговым сервером компании и предлагает гибкие шаблоны создания программных модулей торговых роботов, позволяющие реализовывать торговые стратегии, не отвлекаясь на низкоуровневое программирование.

Автор сделал выбор комплекса задач автоматизации: создание встраиваемых модулей для торгового сервера компании, предназначенных для автоматизации торгов на биржевом рынке ценных бумаг. Необходимости в покупке готового решения нет.

1.4.1 Обоснование необходимости и цели использования вычислительной техники для решения комплекса задач

Биржа в современном мире - это электронная торговая система, в которой с использованием информационных торговых систем вводятся заявки, регистрируются сделки, определяются обязательства участников. Биржевой рынок включает несколько торговых площадок, тысячи участников торгов - зарегистрированных на торговых площадках банков и инвестиционных фондов, инвесторов, а также обширную инфраструктуру электронных торговых систем.

Техника организации и проведения торгов на биржевом рынке схожа на различных торговых площадках, несмотря на некоторые существующие отличия. Участие в торгах на биржах принимают брокеры, в большинстве своем действующие в интересах своих клиентов – управляющие компании, а не самостоятельно. Взаимодействие между управляющими компаниями и брокерами осуществляется обычными средствами связи, такими как телефон или через торговые терминалы. Брокеры принимают участие в биржевых

торгах, используя инфраструктуру и средства организаторов торгов (каналы связи, торговые терминалы, информационные торговые системы).

В основу механизма торговли в рамках основного режима торгов заложен принцип рынка конкурирующих между собой заявок, при котором сделка заключается автоматически при пересечении условий во встречных заявках. В ходе торгов биржевые сделки совершаются по принципу биржевого аукциона. Наиболее распространенным его вариантом является непрерывный двойной встречный аукцион, в ходе которого, исполнение заявок участников биржевых торгов осуществляется непрерывно по мере поступления встречных заявок в соответствии с очередностью.

Заявки на покупку/продажу ценных бумаг с целью заключения сделок принимаются системой торгов к исполнению только после внесения участником необходимых средств в российских рублях и ценных бумагах. При приеме каждой заявки в Системе торгов автоматически фиксируется время ее ввода и ей присваивается идентификационный номер.

При постановке заявки в очередь ее место в очереди определяется ценой, указанной в заявке (первыми в очереди заявок на покупку находятся заявки с большими ценами; первыми в очереди заявок на продажу находятся заявки с меньшими ценами). Среди заявок с равными ценами первыми в очереди находятся заявки, поданные ранее. Заявки всех участников рынка по одной ценной бумаге образуют единую консолидированную очередь. При совпадении цен заявок встречного направления в торговой системе совершается сделка на максимально возможный объем. Остаток по неудовлетворенной сделке остается в системе. В ходе торгов фирма-участник может добавлять, изменять и удалять свои активные заявки.

Заключение сделки по заявке осуществляется при пересечении ее ценовых условий с ценовыми условиями заявок противоположной направленности, принятыми к исполнению в Системе торгов. Пересечение ценовых условий имеет место для поданной заявки:

– на покупку - при наличии заявок на продажу с ценами, не превышающими цены, которая указана в поданной заявке;

– на продажу - при наличии заявок на покупку с ценами, не меньшими, чем цена, которая указана в поданной заявке.

В процессе работы для всех участников торгов рассчитываются текущие и плановые позиции суммарных показателей собственных и клиентских средств по каждому инструменту (деньгам и ценным бумагам).

Создаваемая информационная торговая система в целях автоматизации торгов на биржевом рынке ценных бумаг существенно облегчит деятельность специалиста (бизнес - аналитиков, трейдеров, риск - менеджеров) на всех этапах его деятельности, а в большей степени в процессе работы в торговом подразделении организации. Объединение и систематизация информации о рыночной информации, лимитах и ограничениях, времени и конечных сроках, а также быстрое формирование, оценка и ввод в эксплуатацию торговых стратегий позволит сэкономить транзакционные издержки и позволит решить поставленный комплекс задач. Эту задачу решить невозможно без использования вычислительной техники.

1.4.2 Экономическая сущность комплекса задач

Объектом рассмотрения является процесс автоматизации торгов на биржевом рынке ценных бумаг. Выберем входящую в данный объект задачу, создания встраиваемых модулей для автоматизации торговли ценными бумагами, другими словами, создание среды для развертывания торговых роботов, иначе реализации торговых стратегий. Данная задача относится к классу задач «Автоматизация торгов на биржевом рынке» и необходима для поддержания работы торгового подразделения и тенденции выбранной стратегии работы брокерской компании. Результаты решения данной задачи являются основой для требований менеджмента качества компании в настоящих условиях. Поэтому задача «Создание встраиваемых модулей» является важной и неотъемлемой частью процесса автоматизации торгов на

биржевом рынке ценных бумаг. Информацию для решения задачи получают на этапах проектирования в виде проектов автоматизации, иначе прямых заданий. Результаты решения задачи могут служить идеальной площадкой для тестирования и реализации торговых стратегий.

Отличия рассматриваемого подразделения заключается в том, что это часть всей компании в целом, главная цель которого отвечать требованиям заказчика, иначе соблюдать договорные обязательства компании перед клиентами. Поэтому экономические показатели, направленные на повышение рентабельности бизнеса свойственные типичным коммерческим организациям в нашем случае имеют второстепенное, но не последнее значение, так как торговое подразделение подчиняется требованиям руководящего блока компании и для него также рассчитываются показатели окупаемости проекта. На показатели рентабельности экономической деятельности в первую очередь влияет вознаграждение компании, как и количество самих клиентов, дает весомую часть получаемой прибыли. Как и любые другие показатели, доход этот можно увеличить, путем развития мощностей торгового подразделения, иначе решить выбранный комплекс задач.

1.5 Выбор методологии проектирования и архитектуры ИТС

В настоящее время в области разработки и реализации интеллектуальных систем сложилось следующее положение: с одной стороны, квалификация коллективов разработчиков здесь, как правило, достаточно высока, с другой стороны, одна из сложнейших проблем, препятствующих широкому внедрению ИС, является недостаточное знание системными аналитиками и программистами предметных областей, в рамках которых готовятся проекты. Создание и внедрение систем общения с базами данных и особенно информационных торговых систем (ИТС), и их широкое распространение выдвинуло проблему совершенствования методологии создания информационных систем на передний план.

На текущий момент выделяют две основных методологии проектирования информационных систем: структурные методологии и объектно-ориентированные методологии проектирования.

Многие структурные методологии моделирования бизнес-процессов состоят из методологий: SADT (англ. Structured Analysis and Design Technique) метод структурного анализа и проектирования, семейство стандартов IDEF (Icam DEFINITION, где Icam - это Integrated Computer-Aided Manufacturing) и алгоритмические языки. Основные типы методологий моделирования и анализа бизнес-процессов:

1. SADT методология структурного анализа и проектирования, интегрирующая процесс моделирования, управления конфигурацией проекта, использования дополнительных языковых средств и руководства проектом.

2. IDEF0 модели (англ. ProcessModelling), процессное моделирование в общей нотации IDEF. Изучаемая система или процесс представляются в виде взаимосвязанных функций.

3. IDEF1x модели (англ. InformationModellingExtended), тоже информационное моделирование, основанное на концепции «Сущность-связь». Как правило, используется для моделирования реляционных структур баз данных.

4. IDEF3 модели (англ. ProcessDescriptionCapture), описание и документирование процессов информационной системы, используется для моделирования альтернатив и причинно-следственных связей.

5. Нотация DFD (англ. DataFlowDiagramming), удобное средство для формирования контекстной диаграммы, то есть диаграммы, показывающей разрабатываемую ИТС в коммуникации с внешней средой.

В основе вышеперечисленных типов методологий лежит метод декомпозиции.

Выбранная методология для использования в настоящем проекте должна отвечать требованиям:

- Простота моделирования предметной области.
- Моделирования сложных бизнес-процессов.
- Быть наглядной моделью.
- Выявлять возможность повторного использования кода.

Проведем сравнительный анализ и последующий выбор из существующих на сегодняшний день объектно-ориентированным методологий:

1. SA/SD (англ. StructuredAnalysis/StructuredDesign) содержит несколько вариантов обозначений для формальной спецификации программных систем. В методологии SA/SD организован этап структурного анализа (SA). После структурного анализа начинается этап структурного конструирования (SD), в процессе которого разрабатываются и уточняются более тонкие детали проектируемой системы.

2. JSD (англ. JacksonStructuredDevelopment) объединяет этап анализа требований и этап разработки системы в один общий этап разработки спецификаций проектируемой системы. Методология JSD может быть названа объектно-ориентированной с большой натяжкой: в ней почти не рассматривается структура объектов, мало внимания уделяется их атрибутам.

3. OSA (англ. Object-OrientedSystemAnalysis) обеспечивает объектно-ориентированный анализ программных систем и не содержит возможностей, связанных с поддержкой этапа разработки.

4. UML (англ. UnifiedModelingLanguage) является стандартной нотацией визуального анализа и моделирования программных систем.

При выборе методологии проектирования используем критерии, указанные в таблице 1.3.

Таблица 1.3 - Критерии выбора методологии проектирования

Критерий	Весовой коэффициент
Возможность обнаружения ошибок на начал-х этапах ЖЦ	3
Генерация кода	3
Наглядность	1
Наличие программного продукта (ПП)	2

Опыт работы	3
-------------	---

Оценку методологий по критериям проведем согласно бальной системе: 1-нет реализации, 2-частично реализован критерий, 3-реализован (Таблица 1.4).

Таблица 1.4 - Выбор методологии проектирования

Критерий	к (вес критерия)	SA/SD	JSD	OSA	UML
Возможность обнаружения ошибок на начальных этапах ЖЦ	3	2	2	3	3
Генерация кода	3	1	3	1	3
Наглядность	1	2	3	2	2
Наличие программного продукта (ПП)	2	1	1	1	3
Опыт работы	3	1	1	1	3
Итого (с учетом к – значимости критерия)		16	23	19	36

Однозначно можно сказать, что по вышеперечисленным критериям, отвечающим современным требованиям, автор проекта отдает предпочтение объектно-ориентированным методологиям. Для анализа проектирования ИТС будет применяться объектно-ориентированная нотация UML (англ. Unified Modeling Language - унифицированный язык моделирования).

Автор работы применяет объектно-ориентированный подход к проектированию ИС, так как платформа разработки, применяемая в работе – Java (далее в работе представлено обоснование выбора). Разработка автоматизированной информационной системы ИТС ресурсоемкая, т.е. требует несколько сотен человек-часов, то необходимость использования гибкой технологии проектирования обоснована.

Распространенные гибкие технологии:

eXtremeProgramming (XP). Технология XP, (экстремальное программирование) на сегодня является наиболее известной из гибких технологий. Основой проектной документации считается тщательно прокомментированный код. Большое внимание в технологии уделяется тестированию.

FeatureDrivenDevelopment (FDD). Используемое в FDD понятие функции или свойства системы достаточно близко к понятию прецедента использования, используемому в RUP. Существенное отличие: "каждая функция должна допускать реализацию не более, чем за две недели". FDD включает пять процессов, последние два из которых повторяются для каждой функции: составление общей модели, список функций системы, план работы по функциям, проектирование функции, разработка функции.

RationalUnifiedProcess (RUP). Основа RUP это:

- Ранняя идентификация и непрерывное устранение основных рисков.
- Концентрация на выполнении требований заказчиков к исполняемой программе (модель прецедентов).
- Ожидание изменений в требованиях, проектных решениях и реализации в процессе разработки.
- Компонентная архитектура, реализуемая и тестируемая на ранних стадиях проекта.
- Постоянное обеспечение качества на всех этапах разработки проекта.
- Работа над проектом в команде, ключевая роль в которой принадлежит архитекторам.

Принципы:

- Итерационный подход.
- Планирование и управление проектом на основе функциональных требований к системе (варианты использования).
- Построение системы на базе архитектуры программного обеспечения.

При выборе технологии проектирования используем критерии с коэффициентами (Таблица 1.5).

Таблица 1.5 - Критерии выбора с коэффициентами

Критерий	Весовой коэффициент
Возможность выбора степени формализации разработки	2
Итерационный подход	2
Наглядность	1
Производительность	1
Простота использования	1
Наличие программного продукта (ПП)	2

Оценку технологии по критериям проведем согласно бальной системе: 1-нет реализации, 2-частично реализован критерий, 3-реализован (Таблица 1.6).

Таблица 1.6 - Оценка технологий

Критерий	к (вес критерия)	XP	FDD	RUP
Возможность выбора степени формализации разработки	2	1	1	3
Итерационный подход;	2	3	3	3
Наглядность	1	3	2	3
Производительность	1	3	3	3
Простота использования	1	2	2	2
Наличие ПП	2	1	1	2
Итого (с учетом к – значимости критерия)		18	17	24

Вывод: при проектировании ИТС будет применяться гибкая технология проектирования Rational Unified Process.

Будем понимать под архитектурой ПО внутреннюю структуру продукта (компоненты и их связи), основы пользовательского интерфейса продукта, а также квинтэссенцию знаний и решений, являющихся инструментом разработки и управления проектом. Уровень развития информационных технологий на сегодняшний день при построении информационных систем позволяет выбрать архитектуру из достаточно широкого спектра. При создании ИТС торгового подразделения рассматривались различные архитектуры построения информационных систем: файл-серверная архитектура и архитектура «клиент-сервер».

Файл-серверная архитектура нацелена на использование небольших объемов передаваемой информации и работу небольшого числа пользователей. Системы, построенные на основе указанной архитектуры, применяются в небольших компаниях для решения небольших изолированных задач. Файл-серверная архитектура не подходит для реализации ИТС, призванных работать с большим объемом передаваемой информации при высоких нагрузках на сервер, что не позволяет разрабатывать проектируемую ИТС в соответствии с данной архитектурой.

В архитектуре «клиент-сервер» логика приложения выполняется либо на клиентской стороне («толстый» клиент), либо на серверной стороне, обычно, в виде хранимых процедур базы данных («тонкий» клиент). Основной, помимо сложности масштабирования, проблемой архитектуры клиент-сервер, является необходимость установления соединения между базой данных и каждым работающим с системой пользователем, независимо от того, работает он в настоящий момент или нет (в торговом подразделении предполагается один, максимум два терминала).

Автор остановил свой взвешенный выбор на архитектуре «клиент-сервер», тип двухзвенная, т.е. интерфейсная часть и база данных.

1.6 Разработка модели бизнес-процесса «КАК БУДЕТ»

Ранее были выявлены недостатки текущей модели «как есть», т.е. работы в существующих условиях. Для их устранения необходимо составить модель «как должно быть», в которой учитывались и устранялись все недостатки текущих процессов. В рамках технологии RUP, необходимо помнить о повторяемости итерационного подхода и содержании самих периодов. Стадии RUP: начальная стадия, уточнение, конструирование, внедрение. На примере создание модели «как должно быть» проходим стадию начало:

1. При анализе предметной области, автор проекта сформулировал видение и границы ИТС, было проведено бизнес-моделирование и создана модель «как есть».

2. Создано экономическое обоснование.

3. В таблице перехода, а при повторении итерации в стадии уточнение, разбили на блоки или так называемые плагины. Определили основные требования, ограничения и ключевая функциональность продукта;

4. Создаём базовую версию модели прецедентов.

При соблюдении требований технологии RUP, повторим итерации в стадии уточнения, благодаря чему у нас появится несколько моделей «как должно быть», с конечной моделью.

Для того чтобы более точно понять, как должна работать информационная система, все чаще используется описание функциональности системы через варианты использования (англ. use case или прецеденты). Варианты использования предназначены в первую очередь для определения функциональных требований к системе и управляют всем процессом разработки. Все основные виды деятельности, такие как анализ, проектирование, тестирование выполняются на основе вариантов использования. Во время анализа и проектирования варианты использования позволяют понять, как результаты, которые хочет получить пользователь влияют на архитектуру системы и как должны себя вести компоненты системы, для того чтобы реализовать нужную для специалиста функциональность.

В процессе тестирования, описанные ранее варианты использования позволяют проще оценить точность реализации требований специалистов и позволяют провести пошаговую проверку этих требований.

Стратегия использования прецедентов при определении требований определяет необходимость дополнительно к вопросу "что специалисты ждут от системы?" задавать вопрос "что система должна сделать для конкретного специалиста?". Такой подход позволяет искать функции, которые нужны

многим специалистам и исключать те возможности, которые не могут помочь специалистам выполнять свои повседневные задачи.

Диаграмма вариантов использования состоит из актеров, для которых система производит действие и собственно действия Use Case, которое описывает то, что актер хочет получить от системы. Актер обозначается значком человечка, а Use Case - овалом. Дополнительно в диаграммы могут быть добавлены комментарии.

Вид диаграммы вариантов использования «как должно быть» представлен на рисунке 1.2.

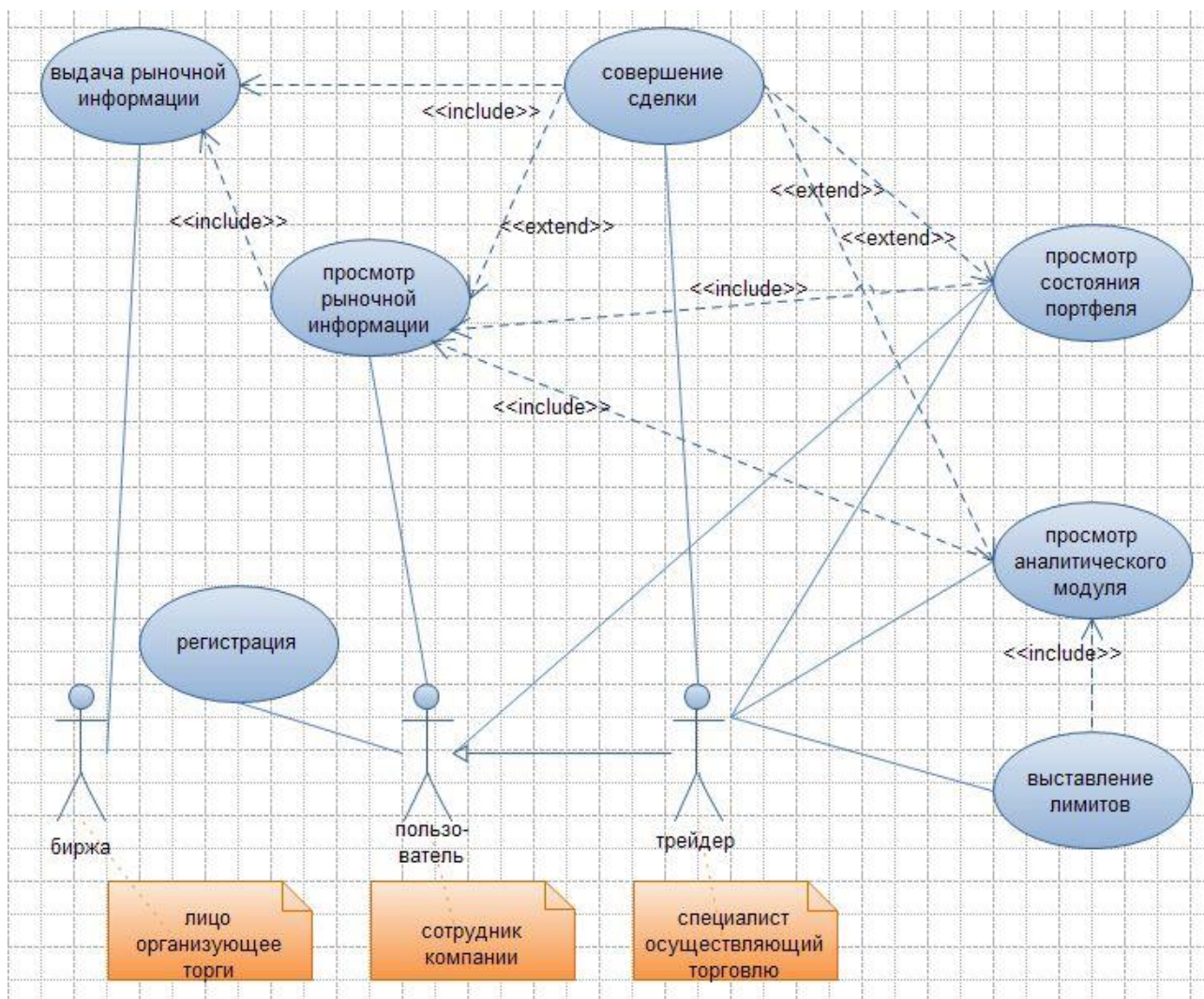


Рисунок 1.2 - Диаграмма вариантов использования «как должно быть», вариант 1

Для реализации требуемого функционала добавлены необходимые прецеденты.

Актер – сотрудник компании, может благодаря информационной торговой системе (ИТС): просматривать рыночную информацию, осуществлять просмотр состояния портфеля.

Актер – специалист, осуществляющий торговлю, может благодаря введению новых прецедентов: выставлять лимиты, работать с аналитическим модулем, просматривать состояние портфеля.

Унифицированный процесс – это процесс, управляемый прецедентами, которые отражают сценарии взаимодействия специалистов. Фактически, это взгляд специалистов на программную систему снаружи. Таким образом, одним из важнейших этапов разработки, согласно RUP, будет этап определения требований, который заключается в сборе всех возможных пожеланий к работе системы, которые только могут прийти в голову специалистам и аналитикам. Простота диаграммы прецедентов позволяет легко общаться с заказчиками в процессе определения требований, выявлять ограничения, налагаемые на систему и на выполнение отдельных требований, такие, например, как время реакции системы, которые в настоящей работе попали в раздел нефункциональных требований.

Также диаграмма прецедентов может использоваться для создания сценариев тестирования, поскольку все взаимодействие пользователей и системы уже определено. Сценарий представляет собой последовательность шагов, описывающих взаимодействие между пользователем и системой. (сценарии прецедентов представлены в приложении 1).

Большой комплекс задач не позволяет полностью разобрать предложенную диаграмму (Рисунок 1.2), в связи с этим разберем диаграмму вариантов использования решающую часть функций информационной торговой системы, представленную на рисунке 1.3.

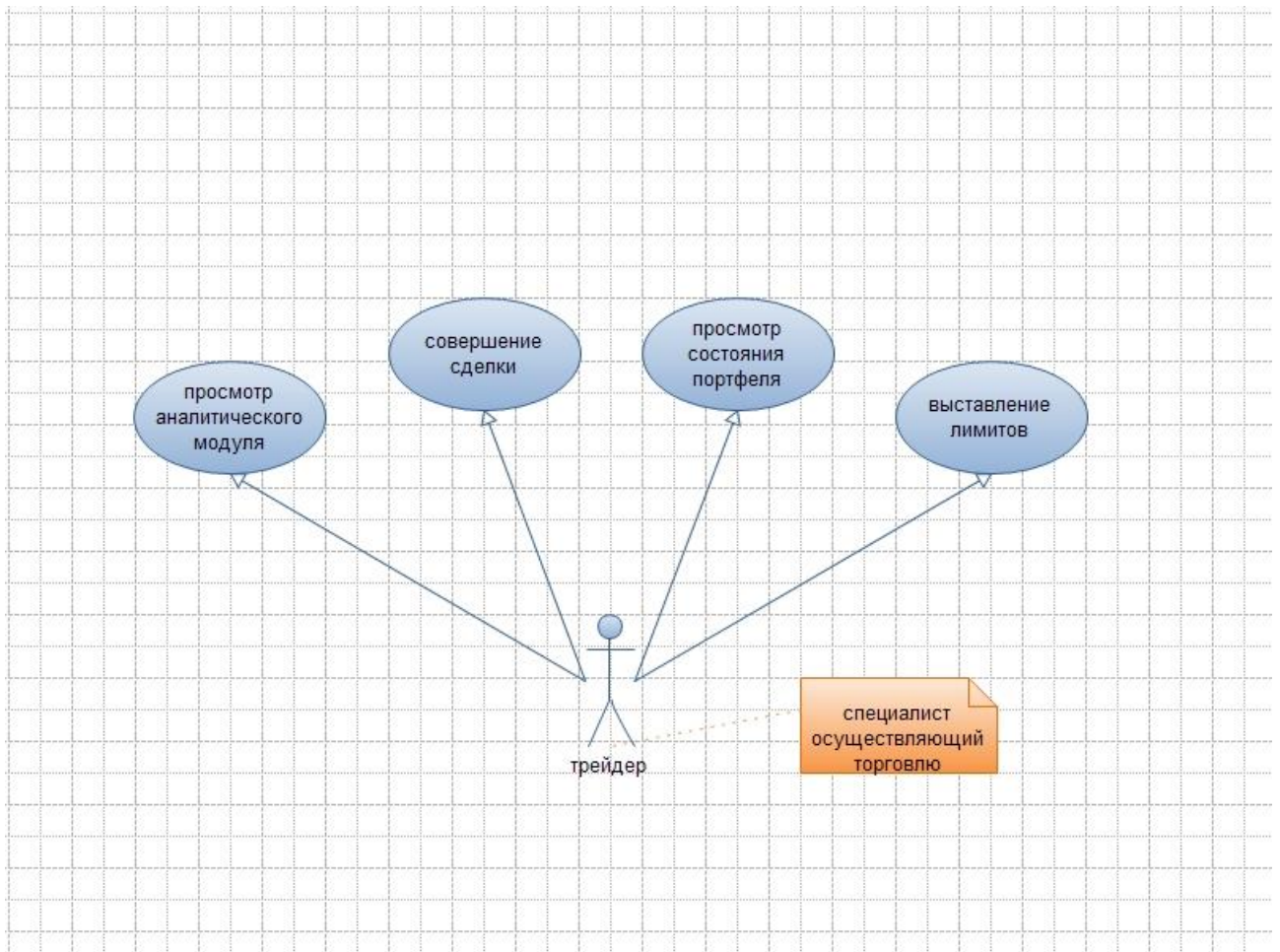


Рисунок 1.3 - Диаграмма вариантов использования «как должно быть», вариант 2

Выделение классов и объектов – одна из самых сложных задач объектно-ориентированного проектирования, которая осуществляется в процессе декомпозиции ключевых абстракций программной системы.

Декомпозиция занимает центральное место в объектно-ориентированном анализе и проектировании программного обеспечения. Под объектно-ориентированной декомпозицией понимается процесс разбиения системы на части, соответствующие объектам предметной области. Практическое применение объектно-ориентированного проектирования приводит к объектно-ориентированной декомпозиции, при которой мы рассматриваем мир, как совокупность объектов, согласованно действующих для обеспечения требуемого поведения.

Автор работы попытался разбить существующие прецеденты с использованием добавления дополнительных вариантов использования, более подробно на рисунке 1.4.

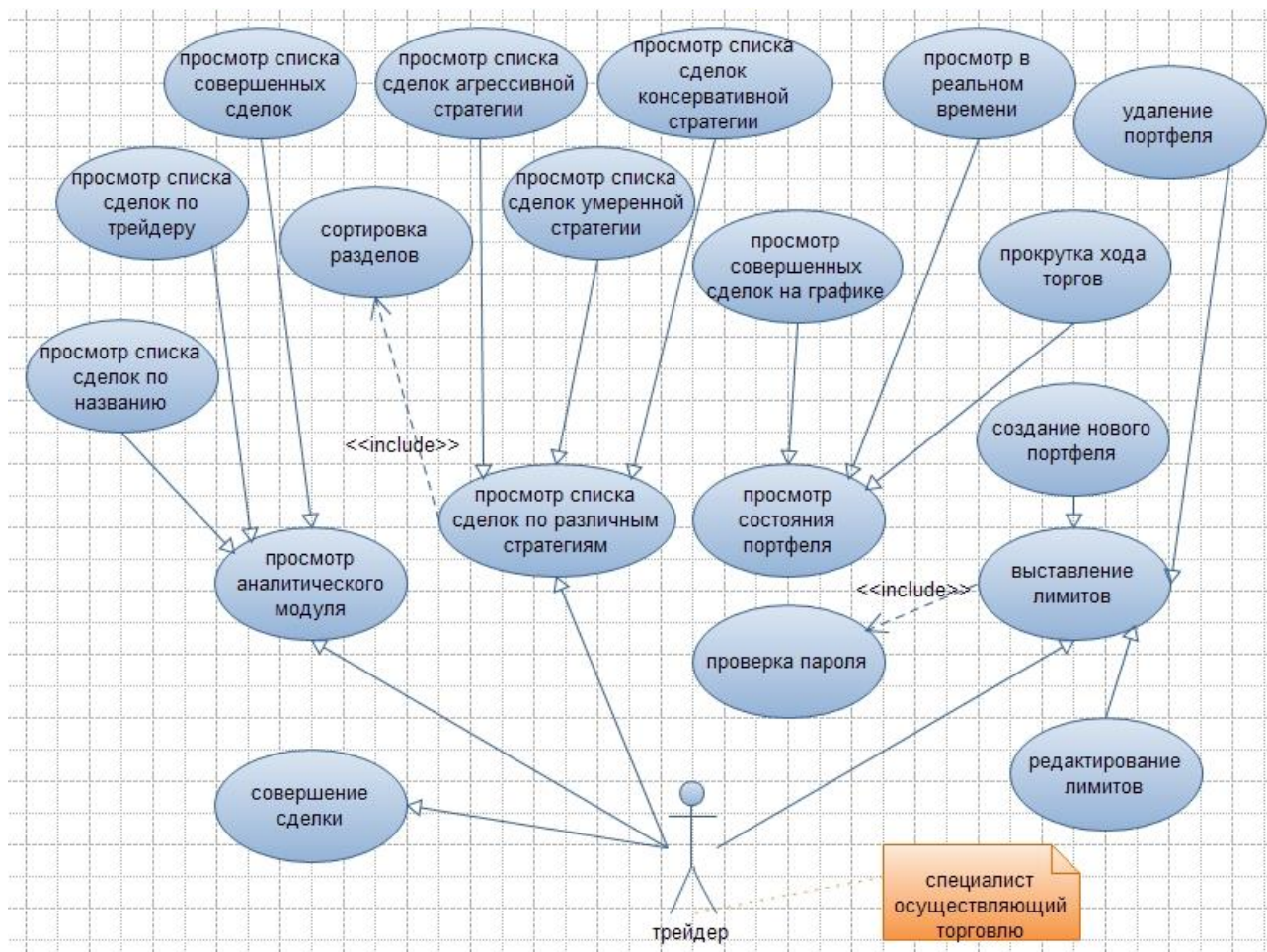


Рисунок 1.4 - Диаграмма вариантов использования «как должно быть», вариант 3

При проектировании и разработке комплекса задач торгового подразделения учитывалась необходимость присутствия свойств открытой системы. Свойства открытой системы, такие как масштабируемость, модульность структуры, использование локальных сетей.

Выполним постановку целей автоматизации путем выделения следующих решений-блоков:

- Автоматическое открытие позиции по стоп-ордеру.
- Автоматическое закрытие позиции по стоп-ордеру.

- Трейлинг-стоп по закрытию позиции.
- Трейлинг-стоп по открытию позиции.
- Модуль подключения сервера к торговой системе.
- Модуль динамического экспорта информации для учетных систем.
- Подключения к торговому шлюзу компании ММВБ (Московская межбанковская валютная биржа).

- Торговые роботы.

Все эти модули (плагины) более подробно рассмотрим ниже.

Сервер. В результате планирования информационной системы было принято решение реализовать проект на платформе Java по типу клиент-серверной архитектуры и модульному принципу. Центральное звено, под названием «сервер», к которому подключаются внешние модули. Модули представляют собой функциональные блоки с различным назначением. Одни модули исполняют ордера, которые выдает сервер. Другие модули эти ордера генерируют. Третьи модули управляют параметрами генерации ордеров. Четвертые модули экспортируют информацию.

Блок автоматического открытия позиции. Позволяет трейдеру войти в рынок при наступлении условия достижения ценой определенных уровней:

1. Для открытия длинных позиций - это движение цены вверх.
2. Для открытия коротких позиций - движение цены вниз.

Блок автоматического закрытия позиции по стоп - ордеру.

Модуль позволяет трейдеру ограничить потенциальные убытки, и тем самым, избежать досадных просадок по счету. Поведение блока аналогично открывающему блоку.

Блоки трейлинг-стоп. Позволяют трейдеру в случае наступления более выгодных условий на рынке, так же автоматически поменять условиях входа/выхода. Т.е. если мы находимся в длинной позиции, и трейдер ставит стоп-ордер, ограничивающий убыток, на уровне 2%, то в случае движения рыночной цены в сторону нашей позиции, модуль трейлинг-стоп будет

двигать стоп-ордер в сторону уменьшения убытков и далее в сторону прибыли в размере 2% от рынка.

Графический клиент. Этот модуль осуществляет подключение к серверу по сети и осуществляет мониторинг и контроль процессов, происходящих на сервере. Масштабируемость достигается путем расширения функционала сервера и модификация клиента под нужды пользователей.

Модуль подключения сервера к торговой системе. Позволяет (равно как и все остальные модули подключения к торговым системам) торговому серверу исполнять стоп-ордера, и торговые приказы в торговой системе. Без этого модуля торговый сервер будет просто информационной системой.

Модуль динамического экспорта данных для учетных систем. Необходимость данного модуля обусловлена медленной работой брокеров. Благодаря этому модулю отпадёт необходимость ждать отчета брокера, и появится возможность обрабатывать информацию о сделках, произведенных торговым сервером сразу – из экспортных файлов.

Ручное выставление ордеров. Данный модуль необходим для того, чтобы трейдер, когда он сам выставляет ордер, мог моментально (перед выставлением ордера) посчитать параметры риска (на счет, на актив, на сделку) и т.о. согласовать их с параметрами рынка, не нанеся ущерба системе риск-менеджмента.

Модуль группировки счетов. Позволит трейдеру управлять всеми счетами системы одним касанием – т.е. выставлением одного (для трейдера) ордера, который выльется во множество (для торговых систем) ордеров. Таким образом, достигается механизация труда трейдера, а также увеличивается скорость его реагирования на условия рынка.

Подключение к торговому шлюзу компании ММВБ. Позволит в полной мере задействовать все возможности торговой системы ММВБ, минуя ограничения, накладываемые сервером QUIK.

Модуль тейк-профит ордеров. Позволит забирать прибыль с рынка до того, как позиции закроются по стоп-ордеру. Таким образом, мы можем задавать для своих позиций желаемую прибыль, при достижении рынком которой эта позиция будет закрыта.

Механические торговые системы (МТС). Данный блок является базисом для создания непосредственно роботов. Условия существования блока: МТС будет извлекать прибыль, он еще должен быть по максимуму надежен с точки зрения сохранения ранее заработанной прибыли и ограничения возможных убытков, которые он может нанести счету. Именно этими функциями должен заниматься модуль МТС.

1.7 Разработка прототипа интерфейса пользователя

При разработке информационной торговой системы автору необходимо подумать об одной из основных задач разработки – создать интерфейс, чтобы он был прост для понимания конечному пользователю (трейдеру). Учитывая, что на интерфейсных формах не должно быть ничего лишнего. Именно с помощью интерфейса происходит коммуникация конечного пользователя программного обеспечения с ИТС. Для выявления прецедентов бизнес-логики разрабатываемой ИТС и составления диаграммы вариантов использования требуется создать прототипы пользовательского интерфейса. Это поможет выявить скрытые прецеденты взаимодействия пользователя и системы, определить структуру системы, понять какие сущности нужно отобразить в модель предметной области.

Прототипы интерфейсов пользователя требуются для распределения компонентов управления по экранным формам оптимальным образом. Они позволяют проектировать логику приложения отталкиваясь от того, как будет устроен пользовательский интерфейс.

Взаимодействие пользователя (в настоящем проекте – трейдер) с интерфейсной частью решено использовать режим меню и упрощенная структура окон, не требующего от пользователей много ресурсов в виде

времени, однако в дальнейшем предполагается создание среды сборки со множеством возможностей под конкретную специфику информационной торговой системы. Технологический процесс в виде блок-схемы при работе с меню представлен на рисунке 1.5.

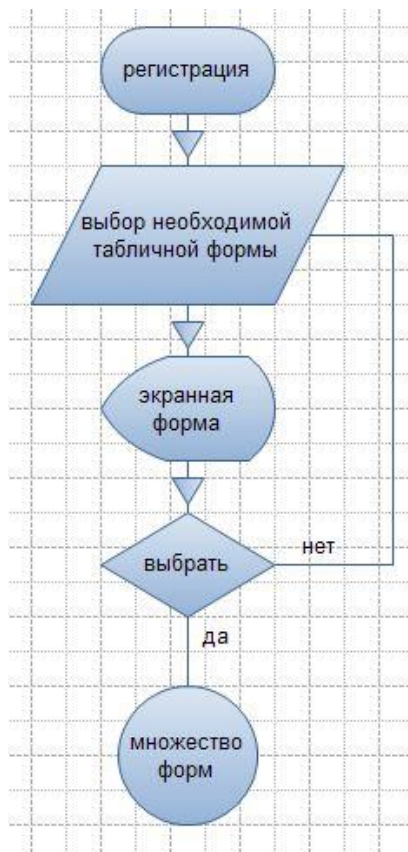


Рисунок 1.5 - Блок-Схема процесса «работа с меню»

Для начала работы с информационной торговой системой работы в торговом подразделении трейдер должен пройти процедуру подключения. Исходя из этого в меню «файл» - «подключиться» пользовательского интерфейса будет создана экранная форма, позволяющую ввести адрес и порт сервера, а также выбрать имя пользователя и пароль, чтобы пройти процесс авторизации. Прототип страницы авторизации пользователя изображен на рисунке 1.6.

При нажатии кнопки «Подключиться» должен начать работу механизм аутентификации. Если трейдер входит в ИТС, то он переходит на основную экранную форму. Здесь должны располагаться меню и соответствующее множество экранных форм.

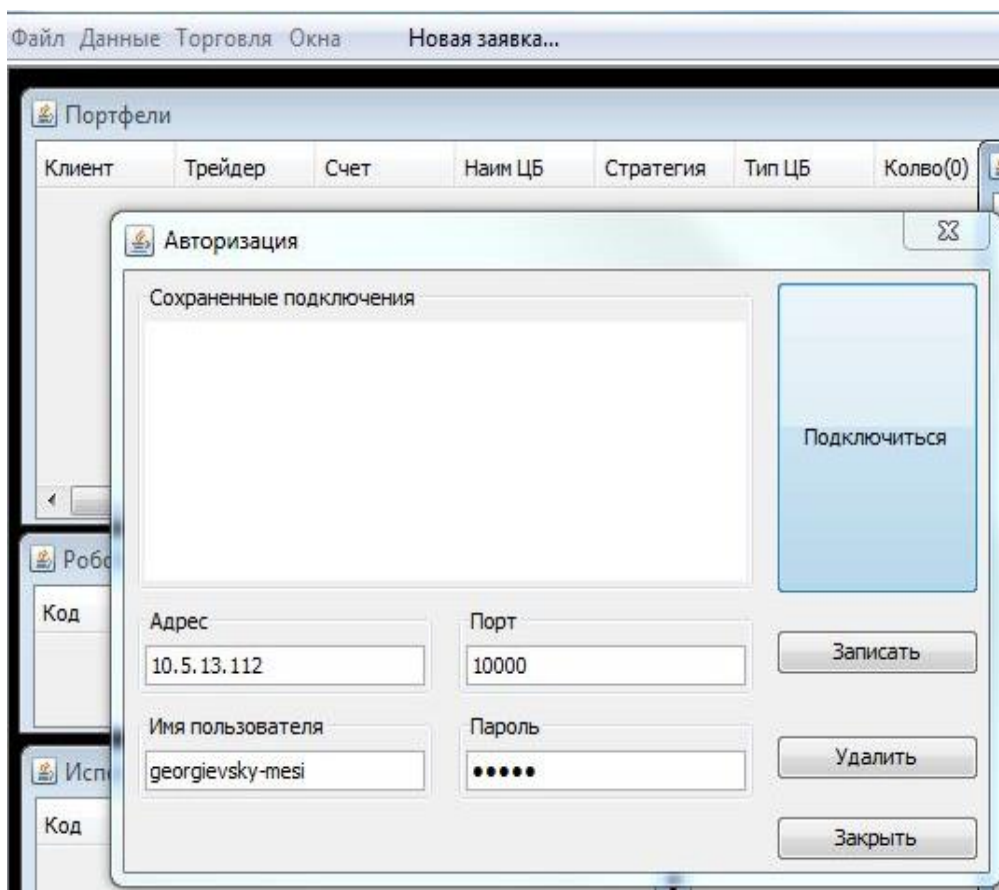


Рисунок 1.6 - Экранная форма «Авторизация»

При нажатии в меню «Данные» появится выпадающий список для выбора необходимой таблицы данных (рисунок 1.7).

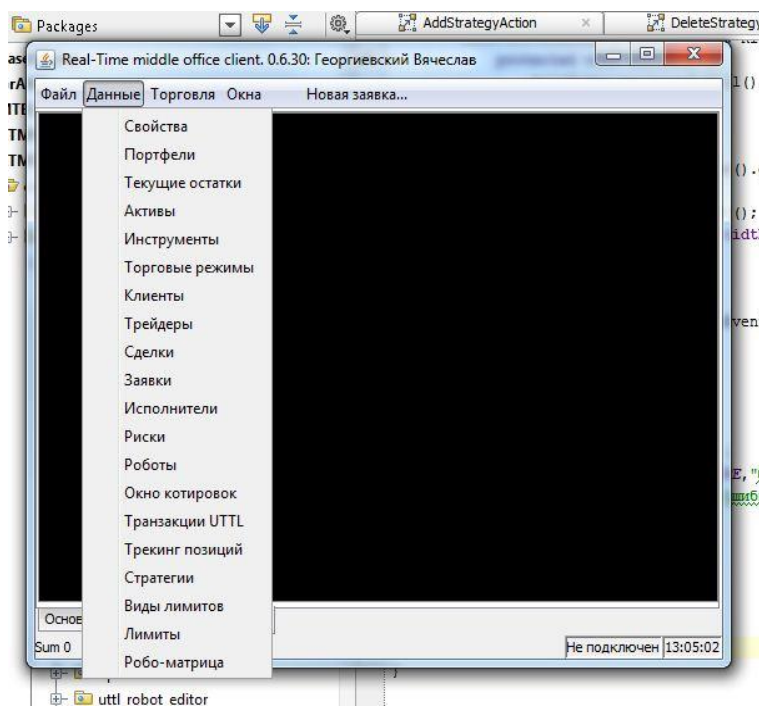


Рисунок 1.7 - Выпадающий список таблиц

На рисунке 1.8 представлена иллюстрация фильтра таблицы.

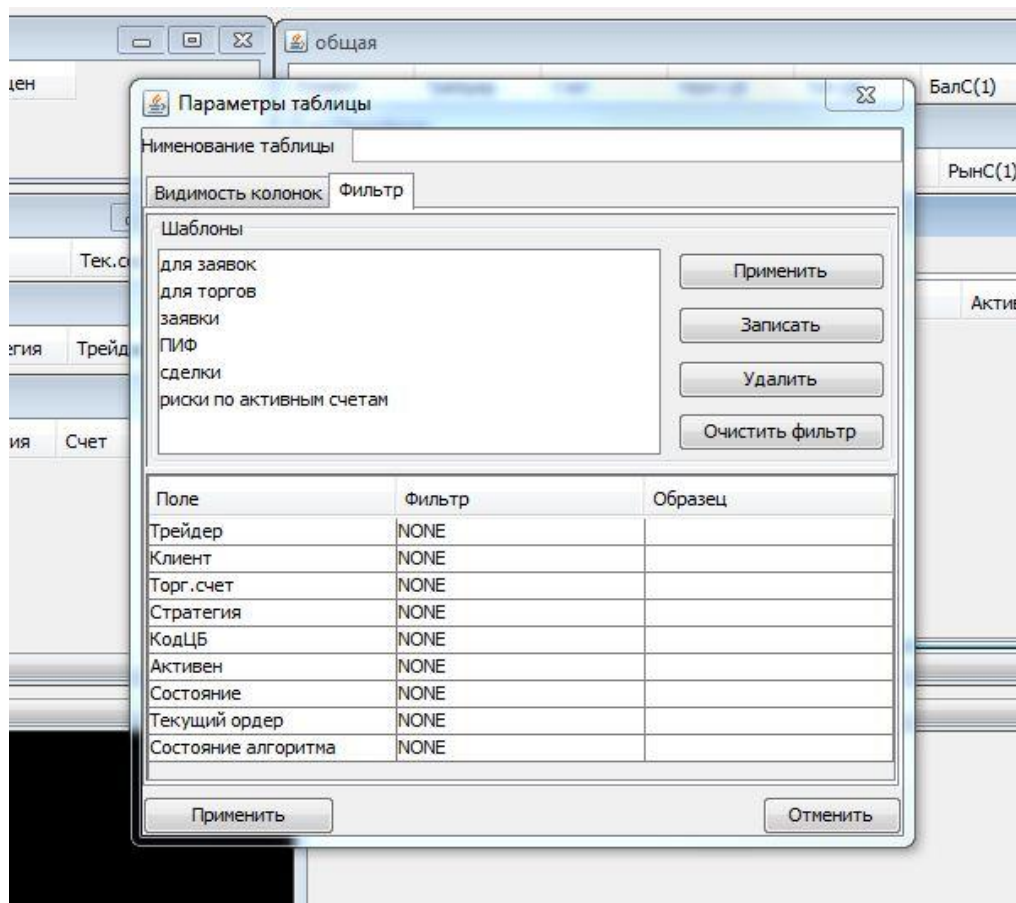


Рисунок 1.8 - Фильтр таблицы

Процесс контроля данных, решает их централизованное хранение, а также доступ к оперативной информации через систему меню.

1.8 Постановка задачи на разработку информационной торговой системы

В процессе проектирования были сформированы следующие исходные данные, необходимые для реализации ИТС осуществления торгов на биржевом рынке в торговом подразделении брокерской компании:

- ИТС создается для автоматизации расчетов при совершении сделок с инструментами биржевого рынка и для автоматизации работы трейдеров управляющей компании.

- ИТС должна быть реализована с использованием двухуровневой архитектуры, отвечающей всем поставленным задачам перед организацией базы данных.
- Серверная часть ИТС должна быть реализована с использованием технологий, позволяющих переносить ее на различные платформы.
- Клиентская часть ИТС должна быть реализована с использованием технологий, позволяющих выполнять приложение на мощностях торгового подразделения.
- Клиентская и серверная части должны взаимодействовать с использованием стека протоколов TCP/IP.
- ИТС должна предоставлять трейдеру удобный и интуитивно понятный интерфейс.
- ИТС должна поддерживать соответствие баз данных с внешними базами данных.

Выводы по главе 1

В результате проведенных в первой главе настоящего проекта исследований и работ:

- Выполнено описание предметной области.
- Наглядно показана актуальность, новизна, практическая ценность предлагаемой информационной торговой системы.
- Проведен сравнительный анализ технологии, методологии и архитектуры проектирования моделей информационных торговых систем в области работы на биржевом рынке.
- Выполнен анализ существующей технологии, определение путей устранения недостатков, характеристика существующих процессов.
- Выполнен переход от состояния «как есть» к состоянию «как должно быть», представлен комплекс задач и возможная таблица перехода.
- Выполнена постановка цели и подзадач автоматизации.

Глава 2 Проектирование информационной торговой системы

2.1 Выбор платформы разработки

Понятие платформы и платформенно-ориентированного построения приложений в настоящее время является общепринятым. Платформа - это среда исполнения и набор технологий, используемые в качестве основы для построения определенного круга приложений. Платформа предоставляет разработчику некоторую модель, которая, как правило, изолирует его от понятий и подробностей технологий и платформ более низкого уровня.

Рассмотрим наиболее распространенные платформы: Java Platform Enterprise Edition и Microsoft .NET Framework.

Java Platform Enterprise Edition (Java EE) - набор спецификаций и соответствующей документации для языка Java, описывающей архитектуру серверной платформы для задач средних и крупных предприятий.

Спецификации детализированы настолько, чтобы обеспечить переносимость программ с одной реализации платформы на другую. Основная цель спецификаций - обеспечить масштабируемость приложений и целостность данных во время работы системы. Java EE во многом ориентирована на использование её через веб как в Интернете, так и в локальных сетях. Вся спецификация создаётся и утверждается через JCP (Java Community Process) в рамках инициативы Sun Microsystems Inc (Sun).

Java EE является промышленной технологией и в основном используется в высокопроизводительных проектах, в которых необходима надежность, масштабируемость, гибкость.

Популярности Java EE также способствует то, что Sun предлагает бесплатный комплект разработки, SDK, позволяющий предприятиям разрабатывать свои системы, не тратя больших средств. В этот комплект входит сервер приложений с лицензией для разработки [7].

Microsoft .NET Framework - программная технология, предназначенная для создания, как обычных программ, так и веб-приложений (в качестве платформы для разработок впервые предложена корпорацией Microsoft).

Одной из основных идей Microsoft .NET является совместимость различных служб, написанных на разных языках. Например, служба, написанная на C++ для Microsoft .NET, может обратиться к методу класса из библиотеки, написанной на Delphi. Каждая библиотека (сборка) в .NET имеет сведения о своей версии, что позволяет устранить возможные конфликты между разными версиями сборок. .NET является патентованной технологией корпорации Microsoft.

Оценку платформ по критериям проведем согласно бальной системе: 1-нет реализации, 2-частично реализован критерий, 3-реализован (Таблица 2.1).

Таблица 2.1 - Оценка платформ

Критерий	k	Java	Microsoft .Net
Кроссплатформенность	3	3	1
Многоязычность	2	1	3
Транзакции	2	3	3
Управление удаленными объектами	3	3	3
Веб-сервисы XML	3	3	3
Доступ к данным	3	3	2
Среда исполнения	3	3	3
Стандарты	1	3	2
Дополнительные инструменты	2	3	2
Поддержка со стороны производителя	3	3	2
Стоимость программного продукта	3	3	1
Итого (с учетом k)		80	63

Наибольшее количество баллов по критериям оценки платформ соответствует платформе Java. Следовательно, по результатам проведения сравнительного анализа платформ, для разработки ИТС предпочтение отдано Java Platform Enterprise Edition.

2.2 Описание и анализ технологии разработки

Используется гибкий подход (англ. Agile Development) к созданию программного обеспечения. Целью такого подхода является минимизация рисков, путём сведения разработки к серии коротких циклов, называемых итерациями, которые обычно делятся одну-две недели. Каждая итерация сама

по себе выглядит как программный проект в миниатюре, и включает все задачи, необходимые для выдачи мини-прироста по функциональности.

В качестве основных подходов к моделированию, реализации и тестированию используются Model Driven Architecture (MDA) и разработка через тестирование (Test Driven Development — TDD).

Сутью MDA является управление кодом через модель и кодогенерация. Это означает, что с помощью соответствующих программных средств строится метамодель всей информационной системы, описывающая статические и динамические аспекты системы по правилам, устанавливаемыми конкретной реализацией продукта, позволяющего генерировать код. Далее средство кодогенерации анализирует метамодель и генерирует соответствующий код.

Разработка через тестирование (TDD) предполагает создание модульных тестов до написания кода. Созданный тест ограничивает минимальный функционал метода или класса, и сборка проекта не произойдет до тех пор, пока удачно не выполнится весь набор тестов. Таким образом гарантируется, что собранный проект будет удовлетворять условиям установленных бизнес-правил. Кроме того, прохождение всего набора тестов при добавлении новых классов или методов класса реализует регрессионное тестирование. При передаче заказчику очередной версии системы проводятся интеграционные тесты, проверяющие взаимодействие всех модулей ИТС в виде единого целого.

Гибкие методологии предполагают итеративную разработку, в свою очередь, каждая итерация разработки требует определенной последовательности шагов, направленных на реализацию функционала разрабатываемой информационной системы. Типичная последовательность действий в итерации выглядит следующим образом:

1. Анализ предметной области с учетом функционала, добавленного на предыдущей итерации.

2. Выявление наиболее приоритетных задач и добавление их в предстоящую итерацию.

3. Оценка примерной трудоемкости каждой из запланированных задач.

4. Проектирование архитектуры (если это требуется) для каждой запланированной задач и анализ общей архитектуры разрабатываемой ИТС на предмет возможного улучшения (рефакторинга).

5. Реализация функционала с параллельным юнит – тестированием.

6. Интеграция.

7. Интеграционное тестирование.

8. Выпуск.

Структура модели предметной в гибких проектах быстро меняется. Для того чтобы учесть все вносимые изменения в начале итерации проводится анализ предметной области. Эта операция входит в этап планирования.

Выявление наиболее приоритетных задач так же является частью планирования для того, чтобы выполнять в планируемой итерации только те задачи, решение которых является наиболее перспективным для развития или задач, не требующих отлагательств.

Оценка трудоемкости требуется для составления графиков производительности команды разработчиков. На этом этапе каждый участник команды называет примерное время, за которое он бы смог справиться с задачей. После чего берется либо усредненное время, либо наименьшее время. Необходимо отметить, что время, в которое оценена та или иная задача, является примерной оценкой и не всегда задача выполняется за него. Сумма оценок выполненных задач за итерацию учитывается при планировании следующей итерации, являясь, своего рода, обратной связью по перебору или недобору задач.

Интеграция предполагает сборку всех частей, как уже реализованных ранее, так и тех, что были созданы на данной итерации, воедино. Сборка происходит при помощи автоматизированных средств сборки,

преимущественно на специализированной ЭВМ — сервере непрерывной интеграции (build server).

После сборки составляется план тестирования и выполняется интеграционное тестирование, целью которого является выявление возможных ошибок в собранном проекте. В случае обнаружения ошибок составляется отчет об ошибках, которые подлежат устранению. После устранения всех ошибок очередная версия программы сдается заказчику.

Создание крупных проектов требует применения определенного набора инструментов разработки. Ниже описаны средства разработки, применяемые при работе над настоящим проектом.

Для облегчения труда разработчиков предназначены интегрированные средства разработки (англ. Integrateddevelopmentenvironment - IDE), позволяющие автоматизировать рутинные операции, такие как форматирование кода, автоматическое создание стандартных методов класса. Обычно среда разработки включает в себя текстовый редактор, компилятор и/или интерпретатор, средства автоматизации сборки и отладчик. Иногда также содержит средства для интеграции с системами управления версиями и разнообразные инструменты для упрощения конструирования графического интерфейса пользователя.

Среда разработки для создания информационной торговой системы является IntelliJ IDEA - коммерческая IDE для Java компании JetBrains.

Первая версия IntelliJ IDEA появилась в январе 2001 года и быстро приобрела популярность, как первая Java IDE с широким набором интегрированных инструментов для рефакторинга, которые позволяли автору быстро реорганизовывать исходные тексты программ и логично соотносится с итерациями используя выбранную технологию проектирования RUP. Дизайн среды ориентирован на продуктивность работы, позволяя сконцентрироваться на разработке функциональности, в то время как IntelliJ IDEA берет на себя выполнение рутинных операций.

Начиная с шестой версии продукта, IntelliJ IDEA предоставляет интегрированный инструментарий для разработки графического пользовательского интерфейса.

Начиная с версии 9.0, IntelliJ IDEA доступна в двух версиях: Community Edition и Ultimate Edition. Community Edition является полностью open-source версией.

2.3 Выбор СУБД и средств объектно-реляционного отображения

Создание информационных систем с помощью выбранных выше технологий и инструментов разработки не только не налагает ограничение на используемую базу данных, но и расширяет круг СУБД, которые можно использовать для реализации ИТС. Исходя из этого требуется выбрать СУБД.

Для реализации модели создаваемой ИТС требуется реляционная база данных. Были сформированы следующие критерии отбора:

- Наличие бесплатной версии системы (freeware) или версии под лицензиями GNU/GPL (GeneralPublicLicence).
- Поддержка транзакций при высокой производительности.
- Доступность документации и информации о СУБД в Интернете.
- Кроссплатформенность.
- Возможность доступа к данным через интерфейс JDBC.

Исходя из требований, выбор СУБД для использования в рамках данного проекта велся среди бесплатного ПО. Были рассмотрены следующие СУБД:

- MySQL.
- PostgreSQL.
- Interbase (версия 6.0, доступная в исходных кодах).

MySQL - популярная бесплатная СУБД уровня малого бизнеса. К ее достоинствам можно отнести простоту и высокую производительность. До последних версий системы ни один вид таблиц не поддерживал транзакций, за счет чего достигается высокая производительность.

PostgreSQL, как и MySQL, является бесплатной и доступна в исходных кодах. Это система уровня малого и среднего бизнеса. СУБД имеет множество положительных черт. Разработчик периодически размещает на своем сайте новые версии системы и любой желающий может изучить систему комментариам в исходных кодах и документации.

Interbase - продукт компании Borland для систем среднего и крупного бизнеса, предоставляющая механизм транзакций и имеющая объемлющую документацию. Данная СУБД широко используется для создания информационных систем под управлением ОС Windows. С релизом версии 7 данного продукта, версия 6 была предоставлена под лицензией GNU/GPL в исходных кодах.

В таблице подведен итог характеристик, рассматриваемых СУБД и их соответствие указанным критериям. Степень соответствия каждому критерию указана оценкой по пятибалльной шкале и представлена в таблице 2.2.

Таблица 2.2 - Характеристики СУБД

Критерий	СУБД		
	MySQL	PostgreSQL	Interbase
Поддержка транзакций и высокая производительность	3	5	4
Наличие документации и справочной информации	5	5	5
Кроссплатформенность	5	5	5
Поддержка интерфейса JDBC	5	5	4
Итого	18	20	18

При суммировании баллов видно, что СУБД PostgreSQL набрала наибольшее количество баллов, для реализации уровня хранения данных в ИТС будет использована последняя версия СУБД PostgreSQL - PostgreSQL 8.4.

2.4 Проектирование базы данных информационной торговой системы предприятия

На данном этапе реализации требуется составить концептуальную и физическую модели базы данных разрабатываемой ИТС и провести генерацию схемы базы данных выбранными средствами реализации.

Концептуальная модель базы данных представлена на рисунке 2.1.

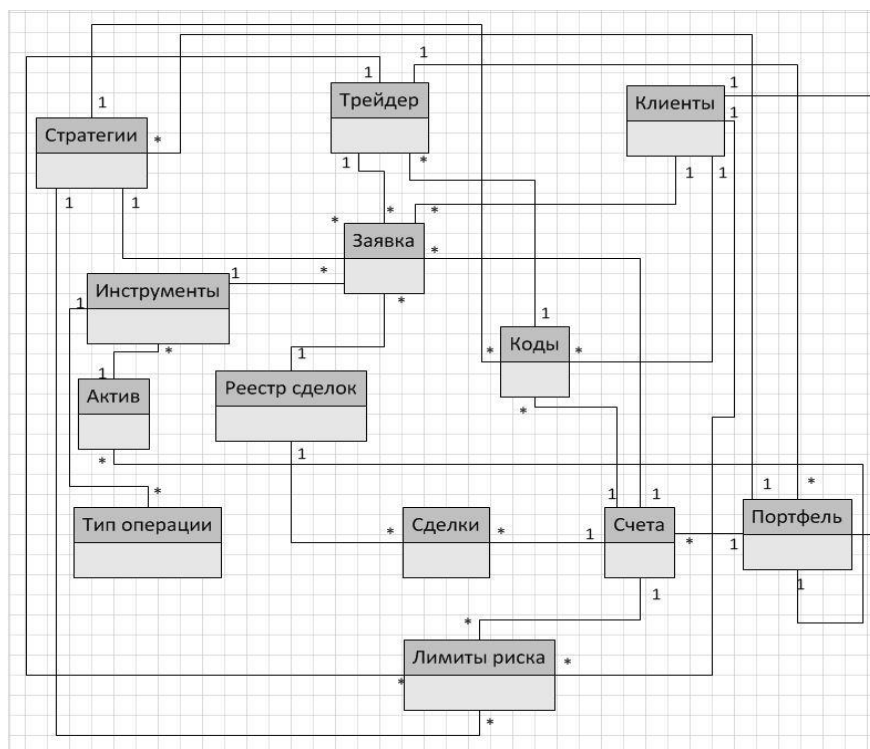


Рисунок 2.1 - Концептуальная модель базы данных

В составе информационного обеспечения рассматриваемого комплекса задач важное место отводится классификаторам рыночной информации, представленных ниже.

Кодированием называется процесс присвоения объектам кодовых обозначений. Основная цель кодирования состоит в однозначном обозначении объектов, а также в обеспечении необходимой достоверности кодируемой информации.

При проектировании кодов предъявляется ряд требований:

- Охват всех объектов, подлежащих кодированию, и их однозначное обозначение.

- Возможность расширения объектов кодирования без изменения правил их обозначения.
- Максимальная информативность кода при минимальной его значности.

Выбор системы кодирования в основном зависит от количества классификационных признаков и разработанной системы классификации. В многоаспектных системах классификации применяется параллельно несколько независимых признаков в качестве классификационных, т. е. исходное множество рассматривается одновременно в разных аспектах.

В соответствии с приведенными требованиями к кодам в настоящем проекте применяется серийная система кодирования, позволяющая кодировать установившееся несложные множества объектов, учитывая возможность расширения кодируемого множества и разбиение по одному признаку классификации.

К оперативной относится информация, которая меняется для каждого фиксированного случая ее использования. Для решения задач автоматизации торгов на биржевом рынке и формирования таких выходных форм как сведения о рыночной информации используется такая интерфейсная часть как представление инструменты. Периодичность формирования документа – по запросу. Макет размещения информации представлен в таблице 2.4.

Таблица 2.4 - Инструменты

п/п №	Наименование поля	Длина поля	Тип поля
1	Описание	1000	текстовый
2	Минимальный шаг цены	19,6	числовой
3	Лот	19	числовой
4	Сколько знаков после запятой	19,6	числовой
5	Последняя цена	19,6	числовой
6	Дата последней цены	Маска: гг/мм/дд 00:00	Дата/время

Характеристика результатной информации. Результатная информация хранится в таблицах базы данных. Таблица базы данных представлена в виде атрибутов и идентификаторов полей. Представление базы данных

представлено в приложение 2. В результате получена база данных, физическая модель которой представлена на рисунке 2.2.

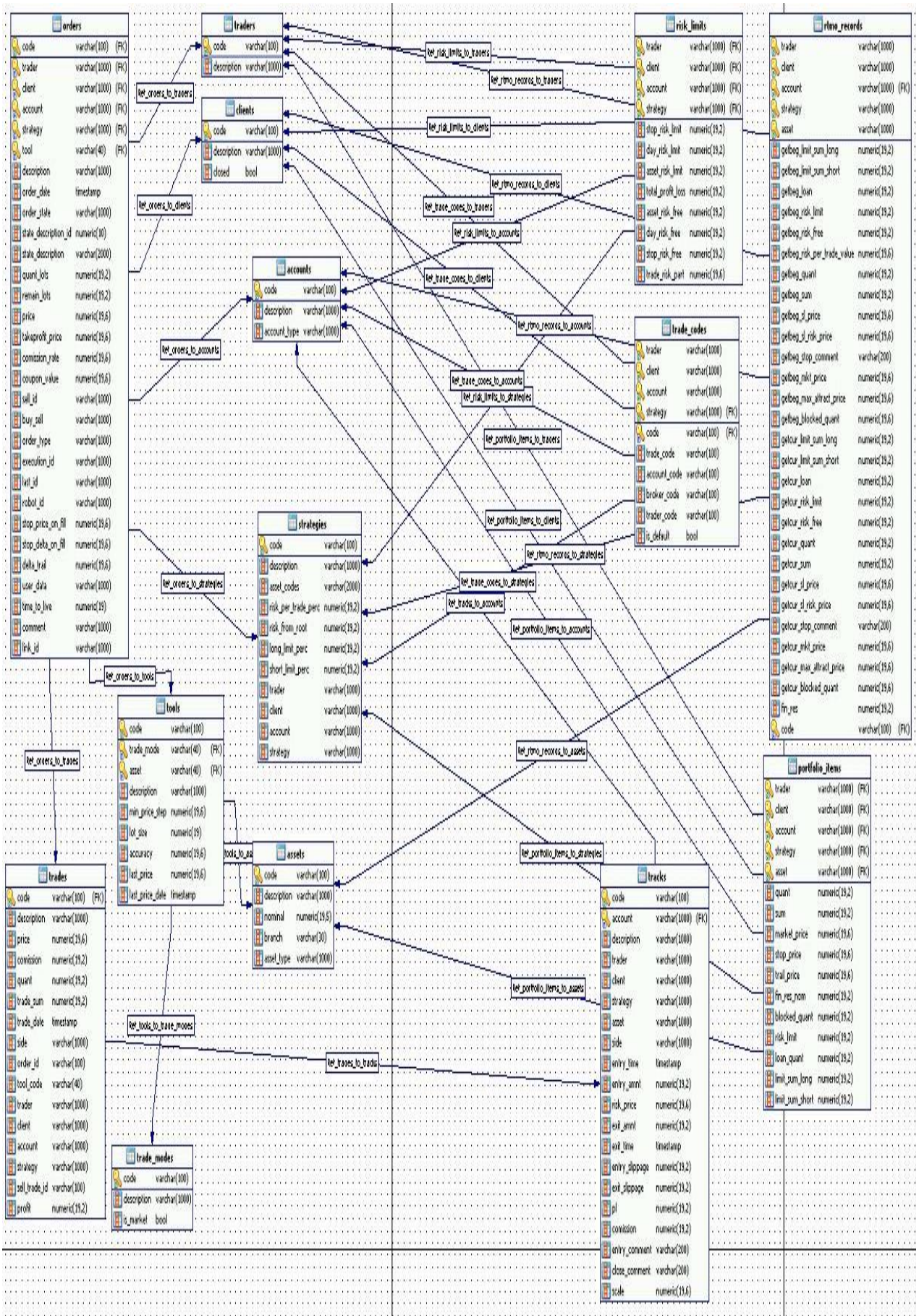


Рисунок 2.2 - Физическая модель базы данных

На рисунке 2.3 представлен класс «Условия снятия позиции».

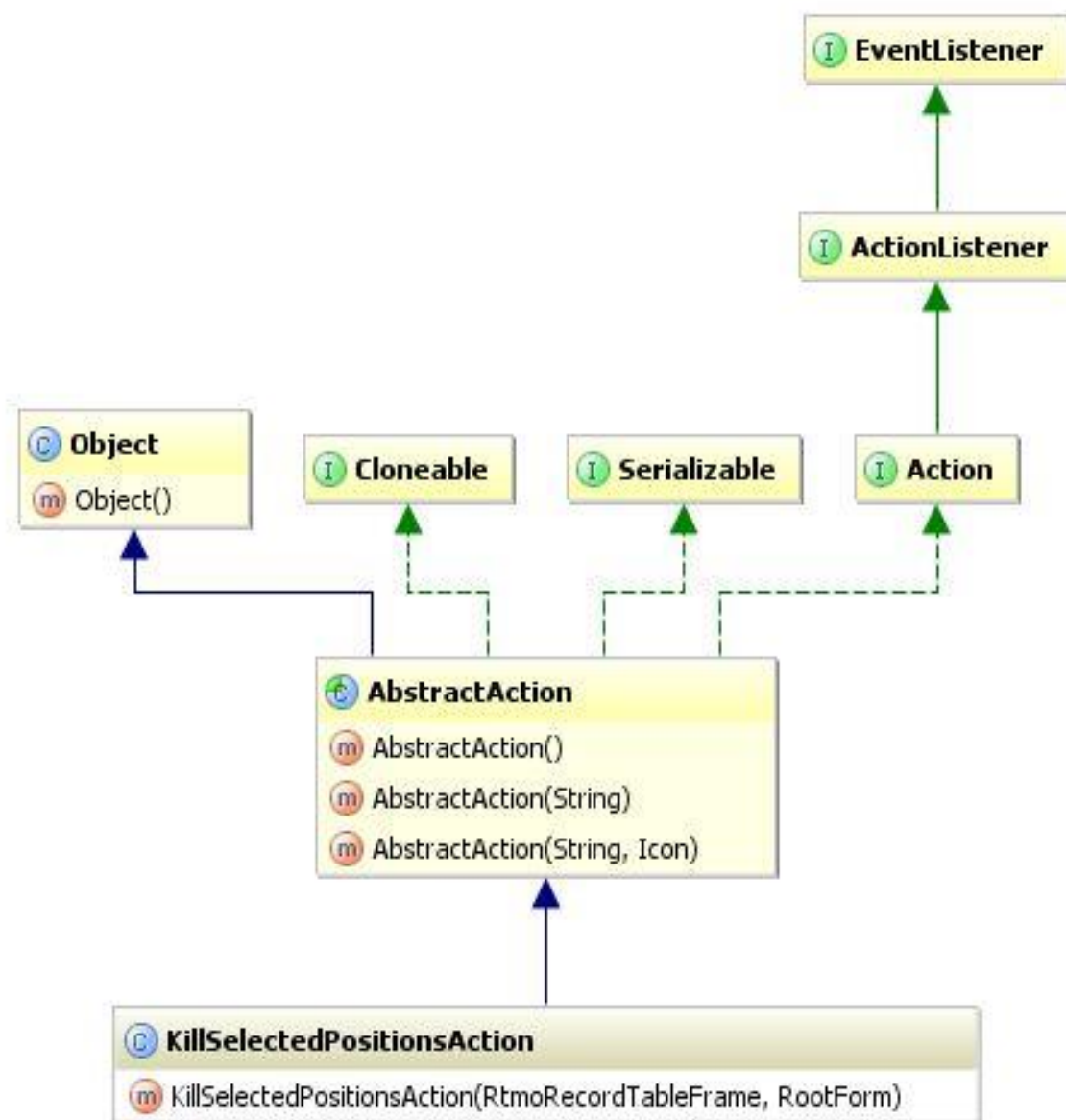


Рисунок 2.3 - Класс «Условия снятия позиции»

В приложении №3 представлен код предложенного класса.

2.5 Разработка основных форм приложения

На рисунке 2.4 представлена иллюстрация «принятие торгового решения» при взаимодействии с интерфейсной частью.

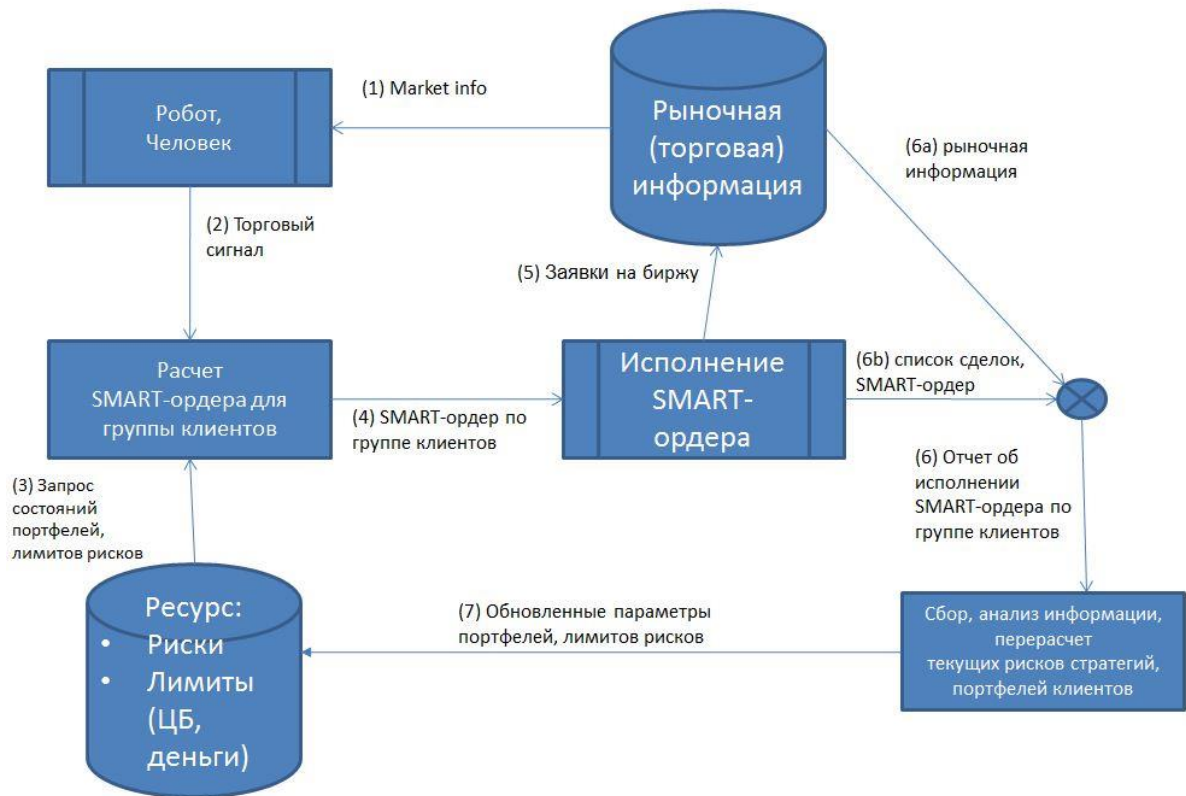


Рисунок 2.4 - Схема технологического процесса «Принятие решения»

The registration form contains the following elements:

- Сохраненные подключения:** A list box containing the text `stPresets:JList`.
- Подключиться:** A large button to initiate the connection.
- Адрес:** A text input field for the server address.
- Порт:** A text input field for the server port.
- Записать:** A button to save the current configuration.
- Имя пользователя:** A text input field for the username.
- Пароль:** A text input field for the password.
- Удалить:** A button to delete the current configuration.
- Закреть:** A button to close the form.

Рисунок 2.5 - Форма регистрации

На рисунке 2.5 иллюстрирована форма подключения к серверу данных. На следующем рисунке 2.6 представлено взаимодействие двух классов для формы регистрации.

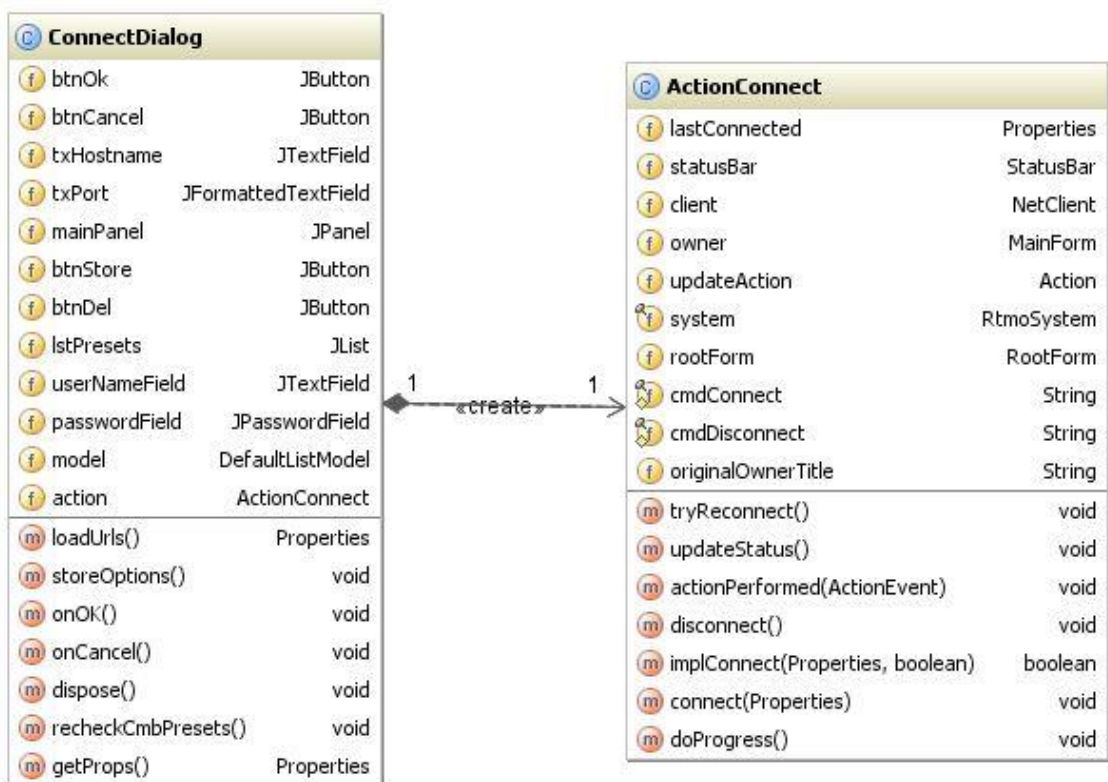


Рисунок 2.6 - Взаимодействие классов

Класс «ConnectDialog» при подключении вызывает класс «ActionConnect». Необходимость создания отдельного класса, а не кодировки действия подключения в форме, обусловлено модульностью системы. Существует возможность использования класса «ActionConnect» в других формах, а также процесс отладки и изменения кода упрощается.

На рисунке 2.7 иллюстрирована форма удаления позиции.

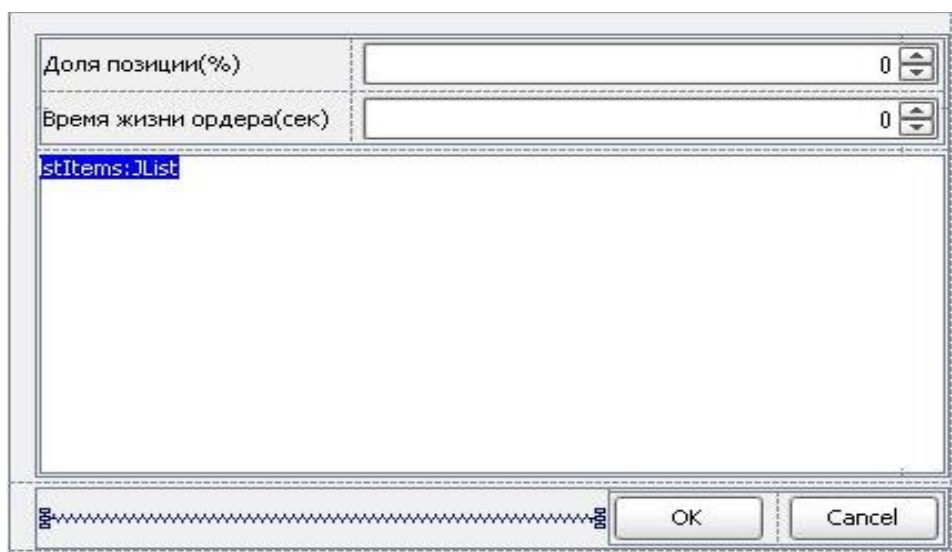


Рисунок 2.7 - Форма удаления позиции

На следующем рисунке 2.8 представлено взаимодействие классов для формы удаление позиции.

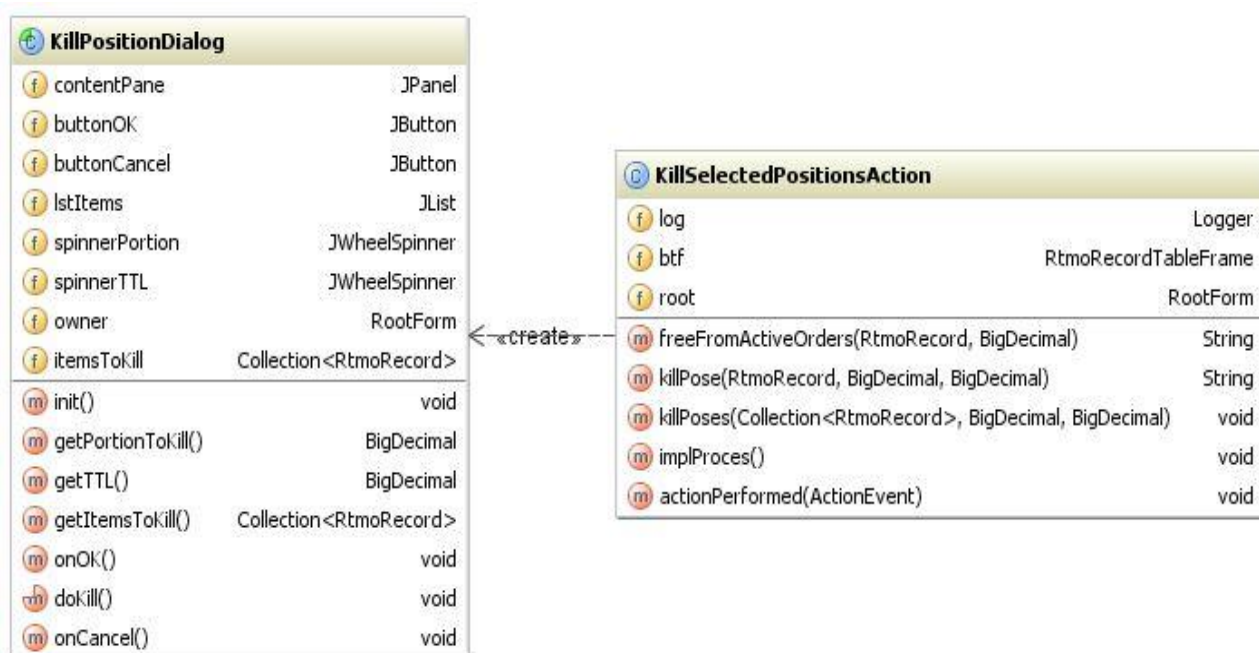


Рисунок 2.8 - Взаимодействие классов

Далее поочередно представлена пара примеров кодирования классов. Класс «KillSelectedPositionsAction» иллюстрирован на рисунке 2.9 и класс «KillPositionDialog» на рисунке 2.10.

На рисунке 2.11 иллюстрирована форма выставления лимита по позиции.

```
UpdateStrategyAction.java x StrategyDialog.java x KillPositionDialog.form x
KillSelectedPositionsAction.java x KillPositionDialog.java x StrategyDialog.form x Package strategy x AddStrategyAction.java x

package com.tht.bench.actions.kill_position;

import ...

/**
 * Kill all selected poses
 * User: georgievsky
 * Date: 10.04.2010
 * Time: 7:06:29
 */
public class KillSelectedPositionsAction extends AbstractAction {
    private Logger log = Logger.getLogger(getClass().getName());
    private RtmoRecordTableFrame btfr;
    private RootForm root;
    public KillSelectedPositionsAction(RtmoRecordTableFrame btfr, RootForm rootForm) {
        super("Установить stop на актив");
        this.btfr = btfr;
        this.root = rootForm;
    }
    private String freeFromActiveOrders(RtmoRecord record, BigDecimal quantToFree) {...}
    private String killPose(RtmoRecord record, BigDecimal portion, BigDecimal ttl) {...}
    private void killPoses(Collection<RtmoRecord> itemsToKill, BigDecimal portion, final BigDecimal ttl) {...}
    private void implProces() {...}

    public void actionPerformed(ActionEvent event) {
        new Thread(){
            @Override
            public void run() {
                try {
                    implProces();
                } catch (Exception e) {
                    log.log(Level.SEVERE, "Ошибка постановка ордеров: " + e.getMessage(), e);
                    root.showMessageDialog("Ошибка постановки ордеров: \n" + e.getMessage(), "ОШИБКА", JOptionPane.ERROR_MESSAGE);
                }
            }
        }.start();
    }
}
```

Рисунок 2.9 - Класс «KillSelectedPositionsAction»


```

KillSelectedPositionsAction.java x KillPositionDialog.java x StrategyDialog.form x Package strategy x
package com.tht.bench.actions.kill_position;
import ...
public abstract class KillPositionDialog extends JDialog {
private JPanel contentPane;
private JButton buttonOK;
private JButton buttonCancel;
private JList lstItems;
private JWheelSpinner spinnerPortion;
private JWheelSpinner spinnerTTL;
private RootForm owner;
private Collection<RtmoRecord> itemsToKill;
    public KillPositionDialog(RootForm owner, Collection<RtmoRecord> itemsToKill) (...)
private void init(){
    setTitle("Вы уверены, что хотите закрыть по рынку следующие позиции?");
    DefaultListModel model = new DefaultListModel();
    for(RtmoRecord record : itemsToKill){
        Split split = record.getKey();
        StringBuffer sb = new StringBuffer();
        sb.append("Счет "+split.getTradeAccount()+"; Трейдер "+split.getTrader()
            +"; Клиент "+split.getClient()+"; Актив "+split.get(Asset.class)
            +"; Колво "+record.getCur().getFreeQuant()+"\n");
        model.addElement(sb.toString());
    }
    lstItems.setModel(model);
    spinnerPortion.setModel(new SpinnerBigDecimalModel(100,0,100,0.01));
    spinnerTTL.setModel(new SpinnerBigDecimalModel(1,0,3600,1));
}
public BigDecimal getPortionToKill(){
    return new BigDecimal(spinnerPortion.getValue().toString());
}
public BigDecimal getTTL(){
    return new BigDecimal(spinnerTTL.getValue().toString());
}
public Collection<RtmoRecord> getItemsToKill() {
    return itemsToKill;
}
private void onOK() (...)

protected abstract void doKill();

private void onCancel() {
// add your code here if necessary
    dispose();
}
}

```

Рисунок 2.10 - Класс «KillPositionDialog»

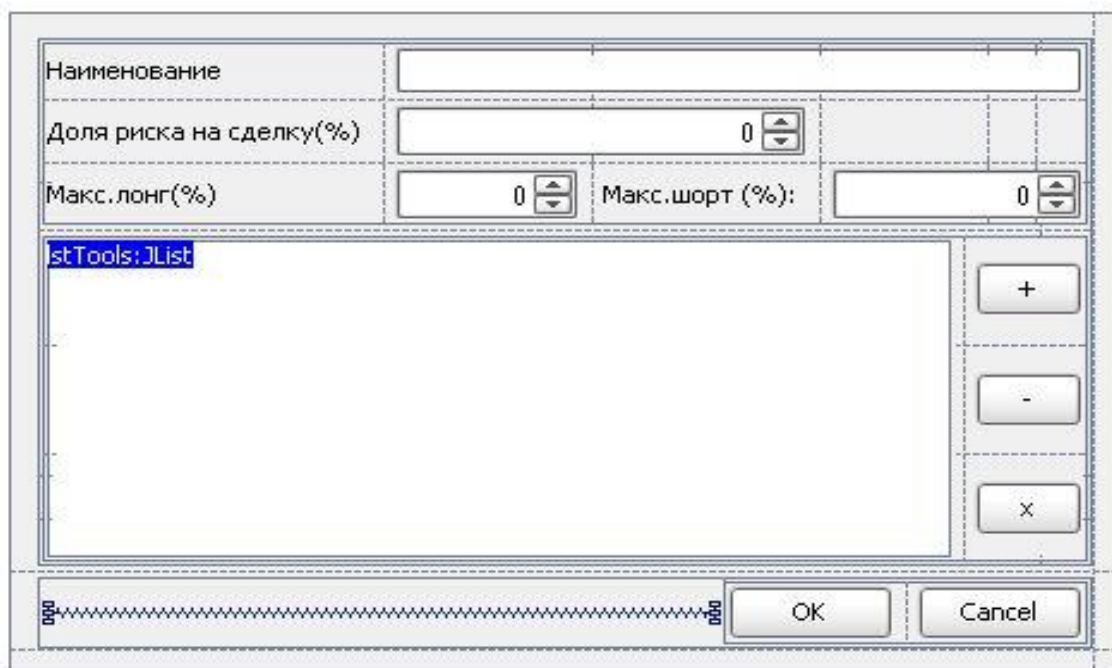


Рисунок 2.11 - Форма выставления лимита по позиции

Выводы по второй главе

В результате проведенных во второй главе настоящего проекта исследований и работ:

- Выполнено графическое представление интерфейсной части.
- Наглядно показана база данных предлагаемой информационной торговой системы.
- Выполнен ряд мер по предоставлению скриптов генерации базы данных.
- Представлены выборочные формы и кодировка классов.

Глава 3 Тестирование разработанной информационной торговой системы

3.1 Описание процесса тестирования

Тестирование программного обеспечения - процесс исследования программного обеспечения с целью получения информации о качестве продукта. Существует множество подходов к решению задачи тестирования и верификации ПО. Качество программных средств можно определить, как совокупную характеристику исследуемого ПО с учётом следующих составляющих: надёжность, сопровождаемость, практичность, эффективность, мобильность, функциональность.

На сегодняшний день выделяют ряд тестов для проверки разработанной системы, ниже представлены некоторые из них:

Модульное тестирование - тестируется минимально возможный для тестирования компонент, например, отдельный класс или функция. Часто модульное тестирование осуществляется разработчиками ПО.

Регрессионное тестирование - после внесения изменений в очередную версию программы, регрессионные тесты подтверждают, что сделанные изменения не повлияли на работоспособность остальной функциональности приложения. Регрессионное тестирование может выполняться как вручную, так и средствами автоматизации тестирования.

Интеграционное тестирование - тестируются интерфейсы между компонентами, подсистемами. При наличии резерва времени на данной стадии тестирование ведётся итерационно, с постепенным подключением последующих подсистем.

Системное тестирование - тестируется интегрированная система на её соответствие требованиям.

Альфа-тестирование - имитация реальной работы с системой штатными разработчиками, либо реальная работа с системой потенциальными пользователями/заказчиком. Чаще всего альфа-тестирование проводится на ранней стадии разработки продукта, но в

некоторых случаях может применяться для законченного продукта в качестве внутреннего приёмочного тестирования.

Бета-тестирование - в некоторых случаях выполняется распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц, с тем чтобы убедиться, что продукт содержит достаточно мало ошибок. Иногда бета-тестирование выполняется для того, чтобы получить обратную связь о продукте от его будущих пользователей.

Предполагается использование метода разработки через тестирование (Test Driven Development — TDD) в качестве метода тестирования программного продукта. Этот метод тестирования предполагает создание модульных тестов до написания кода. Созданный тест ограничивает минимальный функционал метода или класса, и сборка проекта не произойдет до тех пор, пока удачно не выполнится весь набор тестов. Таким образом гарантируется, что собранный проект будет удовлетворять условиям установленных бизнес-правил. Кроме того, прохождение всего набора тестов при добавлении новых классов или методов класса реализует регрессионное тестирование. При передаче заказчику очередной версии системы проводятся интеграционные тесты, проверяющие взаимодействие всех модулей ИТС в виде единого целого.

Интеграционное тестирование в качестве входных данных использует модули, над которыми было проведено модульное тестирование, группирует их в более крупные множества, выполняет тесты, определённые в плане тестирования для этих множеств, и представляет их в качестве выходных данных в виде отчета тестирования.

3.2 Оценка качества пользовательского интерфейса

В ходе проделанной работы была создана информационная торговая система для осуществления операций на биржевом рынке. На очередном этапе необходимо дать оценку эффективности данной разработки. Для этого

воспользуемся квалиметрическим анализом (КА) и экспертным методом на основе опроса, проведённого среди специалистов торгового подразделения. Экспертная оценка коэффициентов весомости определяется по 5 бальной шкале. Показатели эффективности сведены в таблицу 3.1.

Таблица 3.1 - Расчет коэффициентов весомости

Показатель эффективности ИТ	Оценка экспертов					Сумма	Весовой коэффициент
	1	2	3	4	5		
Функциональность пользовательского интерфейса ИТС	4	5	4	4	3	20	0,13
Расположение основных элементов	5	4	5	5	4	23	0,15
Удобство расположение меню	4	4	5	5	5	23	0,15
Информативность	5	5	4	5	4	23	0,15
Цветовая гамма	4	5	4	4	3	20	0,13
Эргономичность	5	4	5	4	4	22	0,14
Эстетичность	4	3	5	4	3	19	0,12
Общая сумма:						150	1

Сравним с эталоном:

1 – показатель ярко выражен, то есть имеет наиболее важное значение при использовании интерфейса, 0 – показатель не выражен, то есть при использовании интерфейса показатель менее важен. Полученные значения перемножаются с соответствующими весовыми коэффициентами, определенными в таблице 3.1. Затем суммируем полученные данные произведений. В итоге получаем тот уровень эффективности, который соответствует оценке данных (Таблица 3.2).

Таблица 3.2- Оценка эффективности интерфейса пользователя

Показатель качества	Коэффициент весомости	Оценка	Эталон
1	2	3	4
Функциональность интерфейса ИТС	0,133333	1	0,133333
Расположение основных элементов	0,153333	1	0,153333
Удобство расположения кнопок	0,153333	1	0,15
Информативность	0,153333	1	0,153333
Цветовая гамма	0,133333	1	0,133333
Эргономичность	0,146666	1	0,146666
Эстетичность	0,126666	1	0,126666

Уровень эффективности	0,873331	8	
-----------------------	----------	---	--

Уровень эффективности по показателям для разработанного интерфейса пользователя равен 0,873331 ~ 0,87, это свидетельствует об удовлетворённости качественных эргономических, эстетических и функциональных показателей, при работе пользователя с данной ИТС.

В результате проведения интеграционных тестов были выявлены и устранены ошибки, не выявленные при модульном тестировании. Отчет о тестировании приведен на рисунке 3.1.

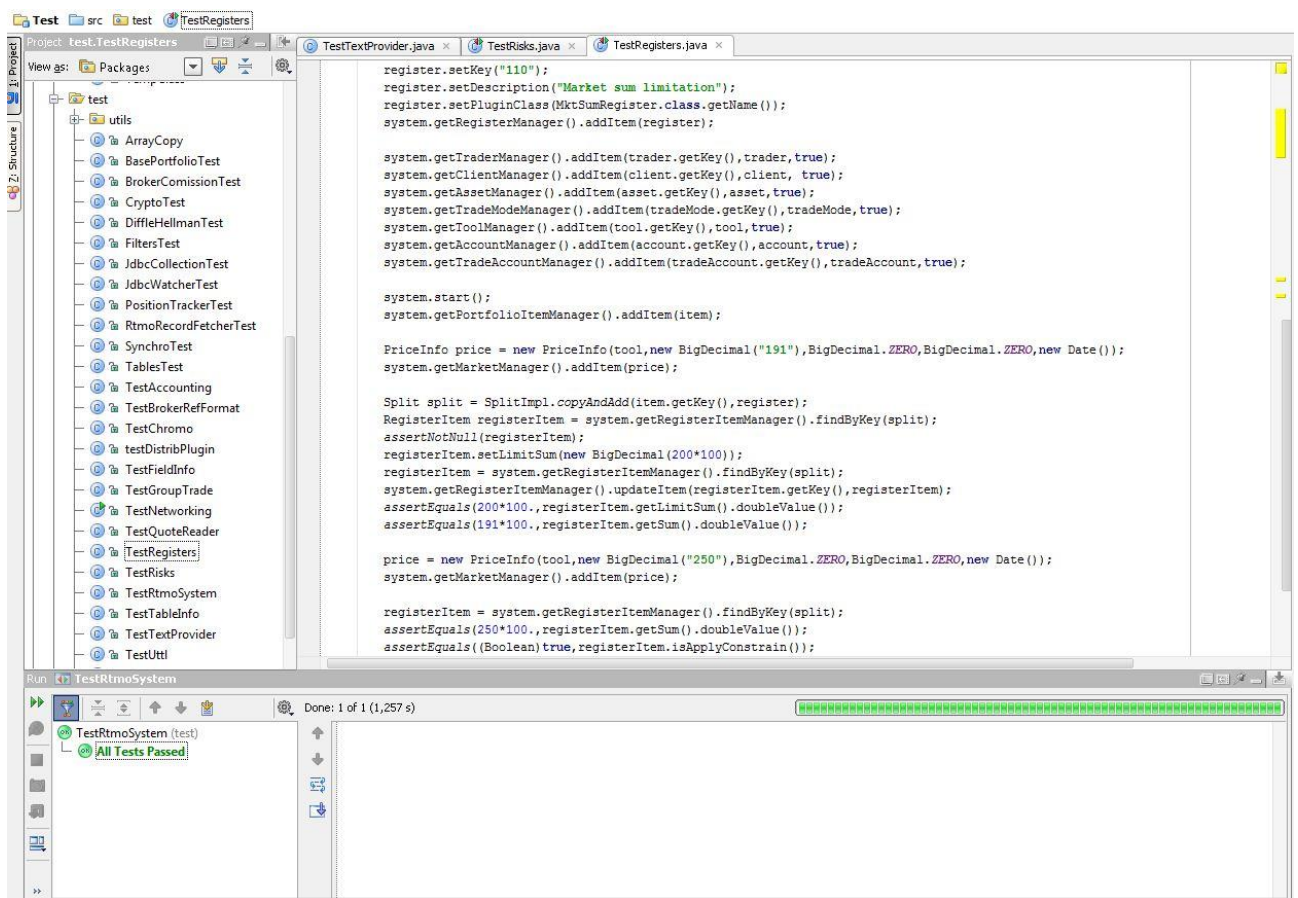


Рисунок 3.1 - Пример теста

На рисунке 3.2 представлен пример отладки программы по результатам теста.

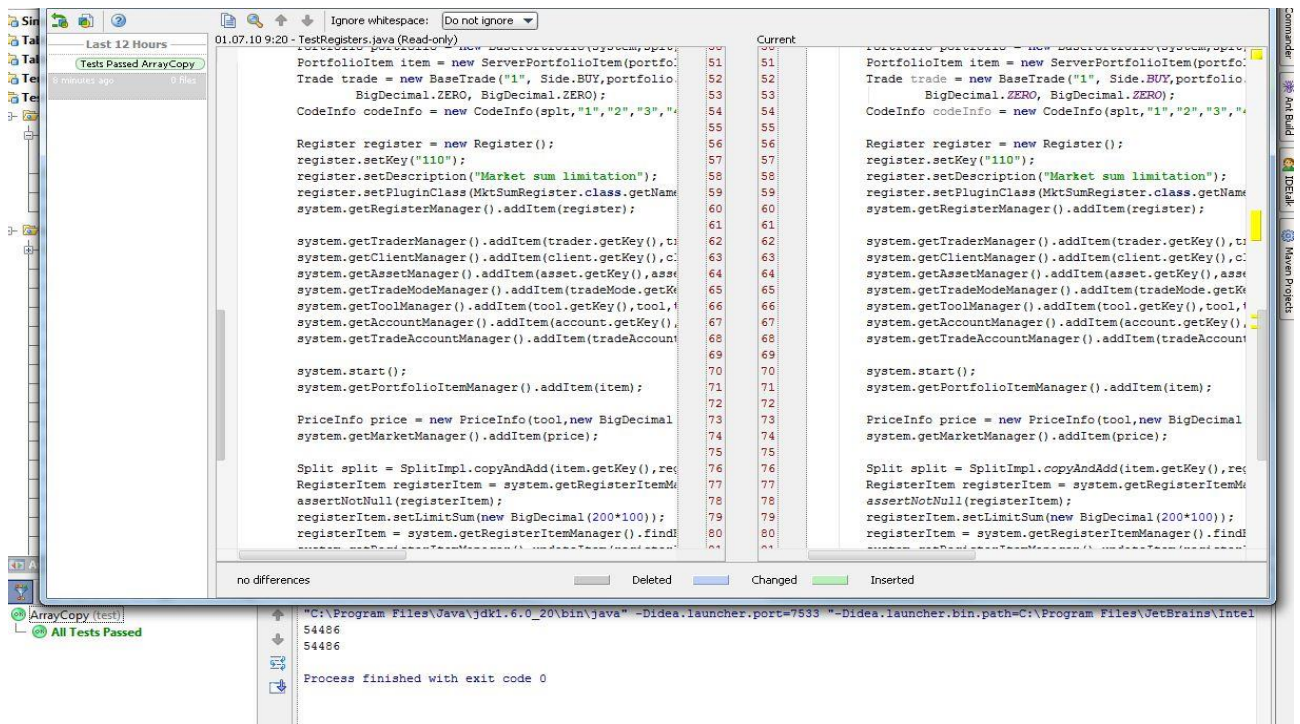


Рисунок 3.2 - Отладка кода

В результате проведения интеграционных тестов были выявлены и устранены ошибки, примеры тестов приведены на рисунках выше.

3.3 Требования к программно-аппаратным средствам ИТС

На данном этапе требуется описать требования к платформе, на которой предполагается развертывание созданной информационной торговой системы.

Исходя из выбранных технологий реализации ИТС для функционирования системы потребуются следующие программные средства на стороне сервера:

- Java Runtime Environment.
- PostgreSQL Server.
- Операционная система, поддерживающая работу со стекком протоколов TCP/IP и протоколом SSH.
- Фильтрация трафика.

Техническое обеспечение (ТО) представляет собой комплекс технических средств, предназначенных для обработки данных в рамках информационной торговой системы. В состав комплекса входят электронно-вычислительный комплекс, осуществляющий обработку рыночной информации, средства сбора и регистрации информации, средства накопления, хранения данных и выдачи результатной информации, вспомогательное оборудование. Анализ использования аппаратных ресурсов сервера с выполняющейся на нем ИТС при альфа-тестировании продукта показал, что для внедрения данного проекта потребуется несколько персональных компьютеров (минимум 3шт.) со следующими характеристиками (Таблица 3.3).

Таблица 3.3 - Технические требования

Наименование	Характеристика (рекомендуется использовать современные решения)
Центральный процессор	ЦП с частотой не менее 2GHz
Оперативная память	4GB
Жесткий диск	Система RAID 1. Ёмкость – не менее 500GB
Периферия (устройства ввода/вывода)	стандарт
Монитор	15 дюймов для серверной части и 21 дюйм (широкоформатный) для пользовательской части (2шт.)

Выводы по третьей главе

В результате проведенных в третьей главе настоящего проекта исследований и работ:

- Иллюстрирована технология тестирования и пример проведения тестов.
- Сформулированы требования к программно-аппаратным средствам ИТС.

Заключение

В результате проведенных исследований и работ:

- Разработана модель, описывающая правила обмена и учета рыночной информации на фондовом рынке.
- Разработана модель определения плановых значений финансовых показателей при использовании системы управления рисками, защищающая интересы торгового подразделения и компании в целом.
- Предложены стандартные методологии проектирования моделей информационных систем для области финансовых приложений. С их помощью разработан необходимый комплект информационных моделей информационной торговой системы.
- Проведен сравнительный анализ архитектур и реализации информационных торговых систем.
- Обоснован выбор методологии проектирования, путем сравнительного анализа.
- Разработаны проектные решения по информационному, программному, техническому и технологическому обеспечению комплекса задач.
- Разработаны и представлены интерфейсные формы.
- Представлены кодировка реально действующей ИТС.

Таким образом, можно сделать вывод, что поставленные задачи выпускной квалификационной работы выполнены в полной мере. Результаты настоящего проекта информационной направленности могут быть использованы в разработках подобного рода, как при разработке новых корпоративных информационных систем предприятий, так и для повышения эффективности эксплуатации и модернизации существующих. Полученные методики учета финансовой деятельности, проектирования и внедрения ИТС применимы как для комплексного решения по автоматизации, так и по частям, повышая эффективность доверительного управления клиентов, и тем

самым, способствуя дальнейшему развитию инвестиционного рынка в России.

Список используемой литературы

1. Открытые системы: Журнал/Издательство «Открытые системы» - Москва.: №06 2003.
2. Современный экономический словарь./Под редакцией Райзберг Б.А., Лозовский Л.Ш., Стародубцева Е.Б.– М.: Инфра-М., 2006.
3. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2006. — 736 с.
4. Бизнес-процесс [Электронный ресурс] .2011.— Режим доступа: <http://ru.wikipedia.org/wiki/Бизнес-процесс>, свободный. — Загл. с экрана. — Яз. Рус.
5. Волатильность [Электронный ресурс] .2011.— Режим доступа: <http://ru.wikipedia.org/wiki/Волатильность>, свободный. — Загл. с экрана. — Яз. Рус.
6. Распоряжение от 04.04.2002 № 421/р «О рекомендации к применению кодекса корпоративного поведения», распоряжение ФКЦБ РФ №421/р от 04.04.2002.
7. Java 2 Enterprise Edition [Электронный ресурс] .2007.— Режим доступа: <http://ru.wikipedia.org/wiki/J2EE>, свободный. — Загл. с экрана. — Яз. Рус.
8. Проектирование информационных систем [Электронный ресурс] .2009.— Режим доступа: <http://www.intuit.ru/department/se/devis/6/>, свободный. — Загл. с экрана. — Яз. Рус.
9. Ларман К. Применение UML 2.0 и шаблонов проектирования. Практическое руководство. 3-е издание.[Текст]: Пер с англ. - М.: ООО «И.Д. Вильямс», 2009. - 736 с.: ил. - Парал. тит. англ.
10. Унифицированный язык моделирования [Электронный ресурс] .2008.— Режим доступа: <http://ru.wikipedia.org/wiki/UML>, свободный. — Загл. с экрана. — Яз. Рус.
11. Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализа прецедентов[Текст]: Пер. с англ. - М.: ДМК

Пресс, 2002. - 160 с.: ил.(Серия «Объектно-ориентированные технологии программирования»)

12. Амблер С. Гибкие технологии: экстремальное программирование и унифицированный процесс разработки. Библиотека программиста [Текст]: - Спб.: Питер, 2005. - 412 с.: ил.

13. .NET Framework [Электронный ресурс] .2008.— Режим доступа: http://ru.wikipedia.org/wiki/.NET_Framework, свободный. — Загл. с экрана. — Яз. Рус.

14. Ruby on Rails [Электронный ресурс] .2008.— Режим доступа http://ru.wikipedia.org/wiki/Ruby_on_Rails, свободный. — Загл. с экрана. — Яз. Рус.

15. DotGNU [Электронный ресурс] .2009.— Режим доступа <http://ru.wikipedia.org/wiki/DotGNU>, свободный. — Загл. с экрана. — Яз. Рус.

16. Бурькин Д.В. Компьютерное моделирование на фондовом рынке // Сб. "Компьютерное моделирование. Экономика". - М.: "Вузовская книга", 2000.

17. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. - СПб.: Бинум, Невский Диалект, 1998. - 560 с.

18. Буч Г. Объектно-ориентированное проектирование с примерами применения / Пер. с англ. - М.: Конкорд, 1992.

19. Вендров А. М. CASE-технологии. Современные методы и средства проектирования информационных систем. - М.: Финансы и статистика, 1998

20. Игнатович Н. В. Брокер интеграции приложений/Открытые системы, - № 9. - 2003

Приложение А

Сценарии прецедентов

Сценарий № 1.

Прецедент: Просмотр аналитического модуля.

Актёры: Треjder.

Описание: ИТС предоставляет пользователю список сделок по введённому параметру.

Цель: Предоставление трейдеру списка рыночных сделок с конкретным параметром.

Тип прецедента: Базовый.

Ссылки:

Потомки прецедента:

1. Просмотр списка совершенных сделок.
2. Просмотр списка сделок по трейдеру.
3. Просмотр списка сделок по названию.

Типичный ход событий:

Действия актёров	Отклик ИТС
Треjder нажимает на кнопку меню «Просмотр по ...» (совершенным сделкам, трейдеру), соответствующую необходимому параметру.	ИТС предлагает трейдеру ввести параметр (совершенные сделки, трейдер).
Треjder вводит параметр списка сделок. Треjder нажимает на кнопку «ОК».	ИТС отображает список сделок с заданным параметром.

Исключения:

Исключение № 1: Треjder не ввёл никакого параметра.

Исключение № 2: В базе ИТС нет списка сделок с заданным параметром.

Исключение № 3: Треjder нажал на кнопку «Отмена».

Исключение № 1: Треjder не ввёл никакого параметра.

Действия актёров	Отклик ИТС
	ИТС выводит сообщение об отсутствии параметра. ИТС предлагает трейдеру ввести параметр списка сделок.

Исключение № 2: В базе ИТС нет списка сделок с заданным параметром.

Действия актёров	Отклик ИТС
	ИТС выводит сообщение об отсутствии в базе ИТС списка с заданным параметром. ИТС предлагает трейдеру ввести параметр списка сделок.

Исключение № 3: Треjder нажал на кнопку «Отмена».

Действия актёров	Отклик ИТС
	ИТС выходит из режима просмотра по параметру.

Сценарий № 2.

Прецедент: Просмотр списка сделок по различным стратегиям.

Актёры: Треjder.

Описание: ИТС показывает трейдеру список сделок по выбранной стратегии.

Цель: Предоставление трейдеру списка сделок определённой стратегии.

Тип прецедента: Потомок прецедента «Просмотр базы ИТС».

Ссылки:

Включает в себя прецеденты:

Продолжение Приложения А

1. Сортировка списка сделок по разделам.

Является базовым для прецедентов:

1. Просмотр списка сделок агрессивной стратегии.
2. Просмотр списка сделок умеренной стратегии.
3. Просмотр списка сделок консервативной стратегии.

Типичный ход событий:

Действия актёров	Отклик ИТС
Треjder нажимает на кнопку меню «Просмотр стратегии ...» (агрессивная, умеренная, консервативная), соответствующую необходимому разделу.	Выполняется прецедент «Сортировка разделов». ИТС отображает список сделок выбранной стратегии.

Исключения:

Исключение № 1: Отсутствие в разделе списка сделок.

Исключение № 2: Треjder нажал на кнопку «Отмена».

Исключение № 1: Отсутствие в выбранной стратегии списка сделок.

Действия актёров	Отклик ИТС
	ИТС выводит сообщение об отсутствии списка сделок в выбранной стратегии.

Исключение № 2: Треjder нажал на кнопку «Отмена».

Действия актёров	Отклик системы
	ИТС выходит из режима просмотра списка сделок определённой стратегии.

Сценарий № 3.

Прецедент: Сортировка разделов.

Актёры: Отсутствуют.

Описание: Список сделок сортируется по стратегиям (агрессивная, умеренная, консервативная).

Цель: Разобрать список сделок по стратегиям.

Тип прецедента: Включаемый в прецедент «Просмотр списка сделок по различным стратегиям».

Ссылки:

Включается в прецеденты: Просмотр списка сделок по различным стратегиям.

Сценарий № 4.

Прецедент: Совершение сделки.

Актёры: Треjder.

Описание: ИТС предоставляет трейдеру возможность совершить сделку.

Цель: Предоставление трейдеру инструмента совершения сделки.

Тип прецедента: Базовый.

Ссылки: Отсутствуют.

Типичный ход событий:

Действия актёров	Отклик ИТС
Треjder выбирает ценную бумагу из списка. Треjder нажимает на кнопку «ОК».	ИТС показывает трейдеру подробную информацию по совершенной сделке.

Сценарий № 5.

Прецедент: Прокрутка хода торгов.

Актёры: Треjder.

Продолжение Приложения А

Описание: Поэтапно ИТС показывает трейдеру сделки на графике.

Цель: Просмотр трейдером сделок на графике

Тип прецедента: Потомок прецедента «Просмотр состояния портфеля с применением графиков».

Ссылки:

Потомок прецедента: Просмотр состояния портфеля с применением графиков.

Типичный ход событий:

Действия актёров	Отклик ИТС
Треjder выбирает список сделок из списка.	ИТС отображает ход торгов на графике.
Треjder нажимает на кнопку «Поэтапно - график».	ИТС показывает трейдеру характеристики хода торгов.

Исключения:

Исключение № 1: Треjder нажимает на кнопку «Стоп».

Исключение № 1: Треjder нажимает на кнопку «Стоп».

Действия актёров	Отклик ИТС
	ИТС выходит из режима поэтапного хода торгов.

Сценарий № 6.

Прецедент: Просмотр совершенных сделок.

Актёры: Треjder.

Описание: ИТС показывает трейдеру ход совершаемых сделок поэтапно на графике.

Цель: Предоставление трейдеру возможности посмотреть ход торгов, совершаемых ИТС или другим трейдером.

Тип прецедента: Потомок прецедента «Просмотр состояния портфеля с применением графиков».

Ссылки:

Потомок прецедента: Просмотр состояния портфеля с применением графиков.

Типичный ход событий:

Действия актёров	Отклик ИТС
Треjder выбирает список сделок из списка.	ИТС предлагает трейдеру выбрать стратегию, которую он будет поэтапно смотреть.
Треjder нажимает на кнопку «Совершенные сделки».	ИТС отображает на графике ход торгов по совершенным сделкам.
Треjder выбирает стратегию, которую он будет поэтапно смотреть.	ИТС отображает на графике ход торгов по совершенным сделкам.
Треjder делает пометки, когда были совершены рискованные сделки.	ИТС сравнивает сделки трейдера с предложенными пометками. ИТС показывает трейдеру полученный результат на графике.

Исключения:

Исключение № 1: Треjder сделал неверную пометку.

Исключение № 2: Треjder нажимает на кнопку «Стоп».

Исключение № 1: Треjder сделал неверную пометку.

Действия актёров	Отклик ИТС
	ИТС выводит сообщение об ошибочной пометке. ИТС показывает варианты правильных пометок.

Исключение № 2: Трейдер нажимает на кнопку «Стоп».
Продолжение Приложения А

Действия актёров	Отклик ИТС
	ИТС выходит из режима поэтапного хода торгов.

Сценарий № 7.

Прецедент: Просмотр состояния в реальном времени.

Актёры: Трейдер.

Описание: ИТС показывает трейдеру состояние портфеля в реальном времени.

Цель: Просмотр трейдером состояния в реальном времени.

Тип прецедента: Потомок прецедента «Просмотр состояния портфеля с применением графиков».

Ссылки:

Потомок прецедента: Просмотр состояния портфеля с применением графиков.

Типичный ход событий:

Действия актёров	Отклик ИТС
Треjder выбирает портфель из списка. Треjder нажимает на кнопку «Просмотр портфеля в реальном времени».	ИТС производит транзакции по обработке. ИТС отображает портфель на графике с комментариями и своими пометками.

Сценарий № 8.

Прецедент: Создание нового портфеля.

Актёры: Трейдер.

Описание: ИТС предоставляет возможность добавить новый инвестиционный портфель.

Цель: Добавление нового портфеля к списку.

Тип прецедента: Потомок прецедента «Выставление лимитов».

Ссылки:

Потомок прецедента: Выставление лимитов.

Типичный ход событий:

Действия актёров	Отклик ИТС
Треjder выбирает стратегию из списка. Треjder нажимает на кнопку «Новый портфель».	Выполняется прецедент «Проверка пароля». ИТС создаёт пустую форму для ввода информации о портфеле и предлагает трейдеру её заполнить.
Треjder заполняет необходимые поля формы. Треjder нажимает на кнопку «Сохранить».	ИТС проверяет поля партии. ИТС сохраняет внесённые изменения.

Исключения:

Исключение № 1: Выполнение прецедента «Проверка пароля» завершилось ошибкой.

Исключение № 2: Трейдер нажал на кнопку «Отмена».

Исключение № 3: Трейдер не заполнил предложенные поля формы «новый портфель».

Исключение № 1: Выполнение прецедента «Проверка пароля» завершилось ошибкой.

Действия актёров	Отклик ИТС
	ИТС выходит из режима создания нового портфеля.

Исключение № 2: Трейдер нажал на кнопку «Отмена».

Действия актёров	Отклик ИТС
	ИТС выходит из режима создания нового портфеля.

Продолжение Приложения А

Исключение № 3: Трейдер не заполнил предложенные поля формы «новый портфель».

Действия актёров	Отклик ИТС
	ИТС предлагает трейдеру заполнить обязательные поля.

Сценарий № 9.

Прецедент: Редактирование лимитов.

Актёры: Трейдер.

Описание: ИТС предоставляет возможность изменить информацию о портфеле.

Цель: Изменение информации о портфеле.

Тип прецедента: Потомок прецедента «Выставление лимитов».

Ссылки:

Потомок прецедента: Выставление лимитов.

Типичный ход событий:

Действия актёров	Отклик ИТС
Трейдер выбирает портфель из списка. Трейдер нажимает на кнопку «Выставление лимитов».	Выполняется прецедент «Проверка пароля». ИТС отображает поля портфеля, доступные для редактирования.
Трейдер изменяет необходимые поля формы. Трейдер нажимает на кнопку «Сохранить».	ИТС проверяет поля портфеля. ИТС сохраняет внесённые изменения.

Исключения:

Исключение № 1: Выполнение прецедента «Проверка пароля» завершилось ошибкой.

Исключение № 2: Трейдер нажал на кнопку «Отмена».

Исключение № 3: Трейдер не заполнил предложенные поля формы «новый портфель».

Исключение № 1: Выполнение прецедента «Проверка пароля» завершилось ошибкой.

Действия актёров	Отклик ИТС
	ИТС выходит из режима редактирования портфеля.

Исключение № 2: Трейдер нажал на кнопку «Отмена».

Действия актёров	Отклик ИТС
	ИТС выходит из режима редактирования портфеля.

Исключение № 3: Трейдер не заполнил предложенные поля формы «новый портфель».

Действия актёров	Отклик ИТС
	ИТС предлагает трейдеру заполнить обязательные поля.

Сценарий № 10.

Прецедент: Удаление портфеля.

Актёры: Трейдер.

Описание: ИТС предоставляет возможность удалить портфель.

Цель: Удаление портфеля из списка.

Тип прецедента: Потомок прецедента «Выставление лимитов».

Ссылки:

Потомок прецедента: Выставление лимитов.

Типичный ход событий:

Продолжение Приложения А

Действия актёров	Отклик ИТС
Треjder выбирает портфель из списка. Треjder нажимает на кнопку «Удалить».	Выполняется прецедент «Проверка пароля». ИТС удаляет портфель. ИТС сохраняет внесённые изменения.

Исключения:

Исключение № 1: Выполнение прецедента «Проверка пароля» завершилось ошибкой.

Исключение № 1: Выполнение прецедента «Проверка пароля» завершилось ошибкой.

Действия актёров	Отклик системы
	ИТС выходит из режима удаления портфеля.

Сценарий № 11.

Прецедент: Проверка пароля.

Актёры: Треjder.

Описание: ИТС открывает/не открывает трейдеру доступ к некоторым функциям системы.

Цель: Прохождение проверки пароля для доступа к некоторым функциям.

Тип прецедента: Включаемый в прецедент «Выставление лимитов».

Ссылки:

Включается в прецеденты: Выставление лимитов.

Типичный ход событий:

Действия актёров	Отклик ИТС
	ИТС предлагает пользователю ввести пароль.
Треjder вводит пароль. Треjder нажимает на кнопку «ОК».	ИТС проверяет пароль. ИТС сохраняет сведения о вводе правильного пароля ИТС выводит сообщение о том, что пароль принят.

Исключения:

Исключение № 1: Треjder уже вводил правильный пароль.

Исключение № 2: Треjder ввёл неправильный пароль.

Исключение № 1: Треjder уже вводил правильный пароль.

Действия актёров	Отклик ИТС
	ИТС выходит из режима проверки пароля.

Исключение № 2: Треjder ввёл неправильный пароль.

Действия актёров	Отклик ИТС
	ИТС выводит сообщение о неверном пароле. ИТС выходит из режима проверки пароля.

Приложение Б

Фрагмент программного кода

```
public class KillSelectedPositionsAction extends AbstractAction {
    private Logger log = Logger.getLogger(getClass().getName());
    private RtmoRecordTableFrame btf;
    private RootForm root;
    public KillSelectedPositionsAction(RtmoRecordTableFrame btf, RootForm rootForm) {
        super("Установить stop на актив");
        this.btf = btf;
        this.root = rootForm;
    }
    private String freeFromActiveOrders(RtmoRecord record, BigDecimal quantToFree){
        StringBuffer buffer = new StringBuffer("");
        int sgn = record.getCur().getQuant().signum();
        List<RtmoOrder> ordersToCancel = new ArrayList<RtmoOrder>();
        BigDecimal totalQnt = BigDecimal.ZERO;
        for(RtmoOrder order : root.getSystem().getOrderManager()){
            switch (order.getState()){
                case ACTIVE:
                case NOT_SENT:
                    break;
                default:
                    continue;
            }
            if(!order.getSplit().isInner(record.getKey())) continue;
            if(!record.getStock().equals(order.getTool().getAsset())) continue;
            if((sgn>0 && order.getSide()==Side.SELL) ||
                (sgn<0 && order.getSide()==Side.BUY)){
                ordersToCancel.add(order);
                totalQnt = totalQnt.add(order.getQuant());
                if(totalQnt.compareTo(quantToFree)>=0){
                    break;
                }
            }
        }
        for(RtmoOrder order : ordersToCancel){
            Future<Boolean> task = root.getSystem().getTransactor().cancelOrder(order.getKey());
            try {
                task.get(60,TimeUnit.SECONDS);
                buffer.append("\nОтменил ордер: ").append(order.getKey());
            } catch (InterruptedException e) {
                buffer.append("\nПрервано ожидание отмены ордера: ").append(order.getKey());
            } catch (ExecutionException e) {
                buffer.append("\nНе смог отменить ордер: ").append(order.getKey());
            } catch (TimeoutException e) {
                buffer.append("\nНе дождался отмены ордера: ").append(order.getKey());
            }
        }
        return buffer.toString();
    }
    private String killPose(RtmoRecord record, BigDecimal portion, BigDecimal ttl){
```

Продолжение Приложения Б

```

Tool tool = root.getSystem().getToolManager().findMarketTool(record.getStock());
if(tool==null){
return "\n\tНе нашел рыночного инструмента "+record.getStock();
}
int sgn = record.getCur().getQuant().signum();
if(sgn==0) return "\n\tКоличество 0";
Split split = record.getKey();
StringBuilder report = new StringBuilder();
BigDecimal absQnt = record.getCur().getQuant().abs();
BigDecimal absFreeQnt = record.getCur().getFreeQuant().abs();
BigDecimal quantToClose = absQnt.multiply(portion);
BigDecimal busyQuant = quantToClose.subtract(absFreeQnt);
if(busyQuant.compareTo(BigDecimal.ZERO)>0){
freeFromActiveOrders(record,busyQuant);
}
record = root.getSystem().getRtmoRecordManager().findByKey(record.getKey());
absQnt = record.getCur().getQuant().abs();
absFreeQnt = record.getCur().getFreeQuant().abs();
quantToClose = absQnt.multiply(portion).min(absFreeQnt);
Side side = sgn>0?Side.SELL:Side.BUY;
Split portfolioSplit = new SplitImpl(split); portfolioSplit.remove(Asset.class);
quantToClose = quantToClose.divide(tool.getLotSize(),0,RoundingMode.FLOOR);
if(quantToClose.compareTo(BigDecimal.ZERO)==0){
return "\n\tОтменять нечего. Количество 0.";
}
BaseRtmoOrder order =
OrderFactory.createMarketOrder(portfolioSplit,tool,side,quantToClose.intValue());
order.setTimeToLive(ttl);
try {
root.getSystem().getTransactor().sendSyncOrder(order);
report.append("\n\tПозиция закрыта.");
} catch(Exception e){
log.log(Level.WARNING,"Can't close pose: "+e.getMessage(),e);
report.append("\n\tОшибка закрытия позы.");
}
return report.toString();
}
private void killPoses(Collection<RtmoRecord> itemsToKill, BigDecimal portion, final
BigDecimal ttl){
int totalQnt = itemsToKill.size();
ProgressMonitor progressMonitor = new ProgressMonitor(root.getJFrame(),"Снимаем
позиции...", "", -1,totalQnt);
progressMonitor.setMillisToDecideToPopup(100);
progressMonitor.setProgress(0);
ExecutorService executor = Executors.newCachedThreadPool();
Map<RtmoRecord,Future<String>> tasks = new
Hashtable<RtmoRecord,Future<String>>();
final BigDecimal FPortion = portion.divide(new
BigDecimal("100"),6,RoundingMode.HALF_DOWN);
for(final RtmoRecord record : itemsToKill){
Future<String> task = executor.submit(new Callable<String>(){

```

Продолжение Приложения Б

```
public String call() throws Exception {
return killPose(record,FPortion,ttl);
}
});
tasks.put(record,task);
}
StringBuffer report = new StringBuffer("Отчет по закрытию позы:");
while(tasks.size()>0){
for(Map.Entry<RtmoRecord,Future<String>> entry : tasks.entrySet()){
Future<String> task = entry.getValue();
if(task.isCancelled()||task.isDone()){
report.append("\nПоза ").append(entry.getKey().getKey()).append(":");
try {
report.append("\n\t").append(task.get());
} catch (InterruptedException e) {
report.append("\n\tпрервано: ").append(e.getMessage());
} catch (ExecutionException e) {
report.append("\n\tошибка ").append(e.getCause().getMessage());
}
tasks.remove(entry.getKey());
break;
}
}
try {
Thread.sleep(500);
} catch (InterruptedException e) {
break;
}
progressMonitor.setProgress(totalQnt - tasks.size());
}
progressMonitor.close();

root.showMessage(report.toString(),"Отчет",JOptionPane.INFORMATION_MESSAGE);
}
private void implProces(){
Collection<RtmoRecord> allRecords = btf.getSelectedRowsCollection();
if(allRecords ==null) return;
List<RtmoRecord> itemsToKill = new ArrayList<RtmoRecord>();
ExecutionManager manager = root.getSystem().getExecutionManager();
for(RtmoRecord record : allRecords){
if (record.getStock().getType()== AssetType.MONEY)
continue;
Split portfolioSplit = new SplitImpl(record.getKey());
portfolioSplit.remove(Asset.class);
if(manager.findExecution(portfolioSplit)== null)
continue;
if(record.getCur().getQuant().compareTo(BigDecimal.ZERO)==0)
continue;
itemsToKill.add(record);
}
if(itemsToKill.size()==0){
```

Продолжение Приложения Б

```
root.showMessageDialog("Нет нашел ни одной позы к закрытию или они все
пусты", "Нечего закрывать", JOptionPane.ERROR_MESSAGE);
return;
}
KillPositionDialog dialog = new KillPositionDialog(root, itemsToKill){
protected void doKill() {
killPoses(getItemsToKill(), getPortionToKill(), getTTL());
}
};
Rectangle rect = root.getJFrame().getBounds();
dialog.pack();
Rectangle ob = dialog.getBounds();
dialog.setBounds(rect.x+(rect.width-ob.width)/                2,rect.y+(rect.height-ob.height)/
2,ob.width,ob.height);
dialog.setVisible(true);
}
public void actionPerformed(ActionEvent event) {
new Thread(){
@Override
public void run() {
try {
implProces();
} catch (Exception e) {
log.log(Level.SEVERE, "Ошибка постанвка ордеров: "+e.getMessage(),e);
root.showMessageDialog("Ошибка                                постанвки
ордеров:\n"+e.getMessage(),"ОШИБКА",JOptionPane.ERROR_MESSAGE);
}
}
}.start();
}
}
```

Приложение В

Скрипты создания таблиц базы данных

```
SQL редактор | Графический конструктор запросов |
-- Table: portfolio_items

-- DROP TABLE portfolio_items;

CREATE TABLE portfolio_items
(
  trader character varying(1000) NOT NULL,
  client character varying(1000) NOT NULL,
  account character varying(1000) NOT NULL,
  strategy character varying(1000) NOT NULL,
  asset character varying(1000) NOT NULL,
  quant numeric(19,2) DEFAULT 0,
  sum numeric(19,2) DEFAULT 0,
  acc_inc numeric(19,2) DEFAULT 0,
  market_price numeric(19,6) DEFAULT 0,
  market_price_acc_inc numeric(19,2) DEFAULT 0,
  stop_price numeric(19,6) DEFAULT 0,
  trail_price numeric(19,6) DEFAULT 0,
  fin_res_nom numeric(19,2) DEFAULT 0,
  fin_res_acc_inc numeric(19,2) DEFAULT 0,
  blocked_quant numeric(19,2) DEFAULT 0,
  risk_limit numeric(19,2) DEFAULT 0,
  risk_per_trade_part numeric(19,6) DEFAULT 0,
  tp_price numeric(19,6) DEFAULT 0,
  stop_comment character varying(1000),
  loan_quant numeric(19,2) DEFAULT 0,
  limit_sum_long numeric(19,2) DEFAULT 0,
  limit_sum_short numeric(19,2) DEFAULT 0,
  CONSTRAINT portfolio_items_pkey PRIMARY KEY (trader, client, account, strategy, asset)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE portfolio_items OWNER TO sa;
```

Рисунок П4.1 - Скрипт создания таблицы «Портфель»

```
SQL редактор | Графический конструктор запросов |
-- Table: strategies

-- DROP TABLE strategies;

CREATE TABLE strategies
(
  code character varying(100) NOT NULL,
  description character varying(1000),
  asset_codes character varying(2000),
  risk_per_trade_perc numeric(19,2) DEFAULT 0,
  risk_from_root numeric(19,2) DEFAULT 0,
  long_limit_perc numeric(19,2) DEFAULT 0,
  short_limit_perc numeric(19,2) DEFAULT 0,
  CONSTRAINT strategies_pkey PRIMARY KEY (code)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE strategies OWNER TO sa;
```

Рисунок П4.2 - Скрипт создания таблицы «Стратегии»

Продолжение Приложения В

```
SQL редактор | Графический конструктор запросов |
-- Table: orders
-- DROP TABLE orders;

CREATE TABLE orders
(
  code character varying(100) NOT NULL,
  description character varying(1000),
  trader character varying(1000),
  client character varying(1000),
  account character varying(1000),
  strategy character varying(1000),
  order_date timestamp without time zone,
  tool character varying(40),
  order_state character varying(1000),
  state_description_id numeric(10,0) DEFAULT 0,
  state_description character varying(2000),
  quant_lots numeric(19,2) DEFAULT 0,
  remain_lots numeric(19,2) DEFAULT 0,
  price numeric(19,6) DEFAULT 0,
  takeprofit_price numeric(19,6) DEFAULT 0,
  comission_rate numeric(19,6) DEFAULT 0,
  coupon_value numeric(19,6) DEFAULT 0,
  sell_id character varying(1000),
  buy_sell character varying(1000),
  order_type character varying(1000),
  execution_id character varying(1000),
  last_id character varying(1000),
  robot_id character varying(1000),
  stop_price_on_fill numeric(19,6) DEFAULT 0,
  stop_delta_on_fill numeric(19,6) DEFAULT 0,
  delta_trail numeric(19,6) DEFAULT 0,
  user_data character varying(1000),
  time_to_live numeric(19,0) DEFAULT 0,
  "comment" character varying(1000),
  link_id character varying(1000),
  CONSTRAINT orders_pkey PRIMARY KEY (code)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE orders OWNER TO sa;
```

Рисунок П4.3 - Скрипт создания таблицы «Заявки»