СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc160196514)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc160196515)

[1.1 Постановка задачи 5](#_Toc160196516)

[1.2 Словарь терминов 6](#_Toc160196517)

[1.3 Описание предметной области 7](#_Toc160196518)

[1.4 Сценарий взаимодействия пользователей с системой 8](#_Toc160196519)

[2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ИНВЕСТИРОВАНИЕ СВОБОДНЫХ СРЕДСТВ» 10](#_Toc160196520)

[2.1 Диаграмма прецедентов и расширенное описание прецедентов 11](#_Toc160196521)

[2.2 Диаграмма классов 14](#_Toc160196522)

[2.3 Диаграмма состояний 16](#_Toc160196523)

[2.4 Проектирование диаграммы последовательности 18](#_Toc160196524)

[2.5 Проектирование диаграммы видов деятельности 19](#_Toc160196525)

[3 РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «ИНВЕСТИРОВАНИЕ СВОБОДНЫХ СРЕДСТВ» 21](#_Toc160196526)

[3.1 Общие принципы организации системы 21](#_Toc160196527)

[3.2 Организация доступа к данным 22](#_Toc160196528)

[3.3 Организация бизнес-логики и интерфейса 23](#_Toc160196529)

[3.4 Организация интерфейса 24](#_Toc160196530)

[ЗАКЛЮЧЕНИЕ 26](#_Toc160196531)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 27](#_Toc160196532)

[ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ 29](#_Toc160196533)

[ПРИЛОЖЕНИЕ Б РЕАЛИЗАЦИЯ ЧАСТИ ФУНКЦИОНАЛА ИНФОРМАЦИОННОЙ СИСТЕМЫ 35](#_Toc160196552)

# ВВЕДЕНИЕ

Инвестирование является одним из ключевых инструментов для эффективного управления финансовыми ресурсами и достижения финансовых целей. В современных условиях развития рыночной экономики и финансовых инструментов, эффективное инвестирование требует информационной системы, которая позволит осуществлять контроль и анализ финансовых операций и поможет принимать обоснованные решения.

Актуальность разработки информационной системы (далее ИС) для инвестирования свободных средств подтверждается тем, что в условиях выбора различных финансовых инструментов и возможности инвестирования в разные активы, важно иметь возможность эффективно отслеживать и анализировать изменения рыночных условий, контролировать свои инвестиционные портфели и принимать своевременные решения для достижения поставленных инвестиционных целей.

Основной задачей данной информационной системы является предоставление пользователю возможности управления своими инвестиционными портфелями, а также анализа финансовых рынков и инструментов, чтобы помочь в принятии обоснованных инвестиционных решений.

Для разработки информационной системы будет использоваться современные информационные технологии и программные инструменты, такие как базы данных, аналитические инструменты и т.д. Это позволит обеспечить пользователю удобный интерфейс, быстрый доступ к необходимым данным и инструментам анализа.

Разработка информационной системы для управления инвестированием свободных средств имеет важное практическое значение, так как обеспечивает эффективное управление финансовыми ресурсами и помогает достигать финансовых целей. Результаты данной работы могут быть использованы как для личного использования, так и для применения в профессиональном финансовом менеджменте.

Цель данного курсового проекта - создание инновационной информационной системы, способной автоматизировать процессы управления инвестициями, повысить эффективность принятия инвестиционных решений и обеспечить надежное хранение и анализ данных об инвестициях.

Задачи курсового проекта по разработке информационной системы для управления инвестированием свободных средств могут быть следующими:

1. Изучение предметной области:

* Изучение основ инвестиций и финансовых рынков;
* Анализ существующих информационных систем для инвестирования и их функциональности;
* Определение требований к разрабатываемой информационной системе.

2. Проектирование информационной системы:

* Определение архитектуры и структуры системы;
* Разработка схемы базы данных и определение необходимых таблиц и связей;
* Проектирование пользовательского интерфейса системы;
* Определение функциональных модулей и их взаимодействия.

3. Разработка информационной системы:

* Создание базы данных и реализация необходимых таблиц и связей;
* Разработка бэкэнд-компонентов системы, включая модули управления инвестиционными портфелями, анализа финансовых рынков, принятия инвестиционных решений и отчетности;
* Реализация пользовательского интерфейса с использованием современных технологий и инструментов.

Структура работы соответствует логике исследования и включает в себя введение, описание предметной области, проектирование информационной структуры приложения, разработка и реализация прототипа ИС, заключение, библиографический список, 2 приложения.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Анализ предметной области включает в себя изучение процесса инвестирования денежных средств компаний в ценные бумаги. Задача руководителя аналитического центра инвестиционной компании заключается в выборе оптимального вида ценных бумаг, который будет приносить прибыль и для компании, и для клиента.

Компания работает с предприятиями, которые хотят управлять своими свободными денежными средствами на определенный период. Предприятия доверяют компании свои деньги в надежде получить прибыль от инвестиций. Поэтому для руководителя аналитического центра важно иметь информацию о предприятии, с которым работает клиент - название, вид собственности, адрес и контактный номер телефона.

Выбор вида ценных бумаг для инвестиций может производиться на основе анализа рынка ценных бумаг, прогнозов экономической ситуации и риска инвестиций. Руководитель аналитического центра должен уметь оценивать риски и потенциальную доходность различных видов ценных бумаг, чтобы принять решение, какие инструменты будут наиболее подходящими для конкретного клиента.

В результате анализа предметной области, руководитель аналитического центра должен иметь хорошее представление о своих клиентах и их потребностях, а также о возможностях и рисках инвестирования на рынке ценных бумаг. Это позволит ему рекомендовать наиболее подходящие виды инструментов для инвестиций и обеспечить прибыль, как для компании, так и для клиента.

## Постановка задачи

Задача руководителя аналитического центра инвестиционной компании состоит в выборе оптимального вида ценных бумаг, который будет приносить прибыль, как для компании, так и для клиента. Для этого необходимо изучить процесс инвестирования денежных средств предприятий в ценные бумаги и оценить риски и потенциальную доходность различных видов инструментов.

Кроме того, руководитель аналитического центра должен обладать информацией о предприятиях, с которыми работает клиент - названии, виде собственности, адресе и контактном номере телефона. Эти данные помогут лучше понять потребности и цели каждого клиента, а также принять во внимание его специфические условия и требования.

Таким образом, основной задачей руководителя аналитического центра является анализ состояния рынка, чтобы иметь полное представление о клиентах, их потребностях и возможностях, а также о рынке ценных бумаг. На основе этого анализа нужно рекомендовать наиболее подходящие виды инструментов для инвестиций, которые обеспечат прибыль, как для компании, так и для клиента.

Подробно постановка задачи и требования к разрабатываемой системе описаны в техническом задании на разработку ИС. Разработанное техническое задание представлено в ПРИЛОЖЕНИИ А

## Словарь терминов

Информационная система - взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Администратор - пользователь ИС, обладающий расширенными правами доступа и ответственный за управление и настройку системы, включая добавление и удаление пользователей, установку прав доступа и другие административные функции.

Доходность - показатель, отражающий прибыльность инвестиций. В приложении инвестирование свободных средств доходность может быть рассчитана на основе полученных доходов от инвестиций и сравнена с целями и ожиданиями инвестора.

Диверсификация - стратегия распределения инвестиций по различным активам или инструментам с целью снижения рисков и повышения потенциальной доходности. В приложении инвестирование свободных средств можно использовать функции диверсификации для оптимизации портфеля.

Инвестиция - вложение денежных средств в какой-либо актив или проект с целью получения прибыли или роста капитала.

Ликвидность - способность быстро и без значительных потерь превратить инвестиции в наличные деньги. В ИС-Инвестирование свободных средств можно отслеживать и анализировать уровень ликвидности инвестиций.

Пользователь - человек, зарегистрированный в ИС-Инвестирование свободных средств и имеющий доступ к функциональности системы, включая возможность внесения и отслеживания инвестиций.

Портфель - набор инвестиций, принадлежащих отдельному инвестору или организации. Портфель может включать различные ценные бумаги, такие как ставка, тип, и другие финансовые инструменты.

План инвестирования - стратегия и план действий по инвестированию, определяющий цели, сроки, ресурсы и методы инвестирования. В ИС-Инвестирование свободных средств можно создавать и отслеживать планы инвестирования.

Риск - вероятность возникновения неблагоприятных событий или потерь при инвестировании. Риск может быть оценен и учтен при принятии решений по инвестированию.

## Описание предметной области

Инвестиционная компания, занимается управлением денежными средствами клиентов на определенный период времени. Задача состоит в выборе наиболее выгодных и безопасных видов ценных бумаг, которые позволят клиентам получить прибыль от своих инвестиций.

Для принятия решений по инвестициям важно иметь информацию о каждом клиенте, такую как название предприятия, вид собственности, адрес и контактный телефон. Это поможет вам оценить потенциальные риски и возможности инвестиций.

Компания предоставляет консультационные услуги клиентам и помогает им определить оптимальный инвестиционный портфель. Для этого необходимо провести анализ финансовых показателей предприятия, оценить текущие тренды на рынке ценных бумаг и использовать специализированную литературу и аналитические методы и инструменты.

Одновременно с управлением инвестициями необходимо осуществлять мониторинг портфеля и оценивать его результаты. Если необходимо, можно проводить оптимизацию портфеля и принимать меры по снижению рисков.

Также важной частью работы является управление и координация работы сотрудников аналитического центра, включая распределение задач, обучение и мотивацию.

Для успешного решения задач могут быть полезны нормативные акты, регулирующие деятельность инвестиционных компаний, а также участие в профессиональных мероприятиях и использование специализированных программ и методик анализа данных.

## Сценарий взаимодействия пользователей с системой

Сценарий взаимодействия пользователей с системой - это описание последовательности действий, которые пользователь выполняет при работе с информационной системой. Он описывает, как пользователь взаимодействует с системой, какие операции выполняет и какие результаты ожидает:

* Пользователь начинает взаимодействие с системой, открывая приложение для инвестирования свободных средств;
* Система предлагает пользователю зарегистрироваться или войти в уже созданный аккаунт;
* Пользователь выбирает вариант регистрации или входит в свой аккаунт введя необходимые данные;
* После успешной регистрации или входа в аккаунт, система предоставляет пользователю доступ к основной функциональности;
* Пользователь может просмотреть информацию о доступных инвестиционных инструментах (ценные бумаги, инвестиции и т.д.), их доходности и рисках;
* Пользователь может создать портфель инвестиций, выбирая нужные инструменты и указывая желаемое соотношение между ними;
* Система предоставляет пользователю информацию о прогнозируемой доходности и рисках созданного портфеля;
* Пользователь может совершить операцию по покупке или продаже выбранных инвестиционных инструментов;
* После совершения операции, система обновляет портфель пользователя и показывает текущую стоимость инвестиций;
* Пользователь может отслеживать изменения стоимости своего портфеля и проводить регулярный анализ своих инвестиций;
* В случае необходимости, пользователь может сделать дополнительные вклады или снятия со своего счета;
* Пользователь также может обращаться в службу поддержки для получения консультации или помощи по вопросам инвестирования;
* Пользователь может изучать различные инвестиционные стратегии и получать рекомендации от системы на основе своего профиля и финансовых целей;
* В случае необходимости пользователь может закрыть свой аккаунт или запросить вывод всех своих средств;
* Система обрабатывает запрос пользователя и осуществляет процедуру закрытия аккаунта или вывода средств;
* Взаимодействие пользователей с системой может продолжаться в дальнейшем, включая регулярное отслеживание и обновление портфеля, проведение новых операций и получение информации от системы для принятия решений об инвестировании свободных средств.

Вывод по первой главе

В данной главе был проведен анализ предметной области, в ходе которого была выполнена постановка задачи, составлен словарь терминов, описана предметная область разработки и спроектирован сценарий взаимодействия пользователей с ИС «Инвестирование свободных средств».

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ИНВЕСТИРОВАНИЕ СВОБОДНЫХ СРЕДСТВ»

В данной главе будет рассмотрено построение различных диаграмм, таких как диаграмма прецедентов, классов, деятельности, последовательности и состояний. Диаграммы при проектировании приложения нужны для визуализации и описания структуры приложения, а также для определения отношений и зависимостей между различными элементами системы. Они помогают лучше понять задачи и требования к приложению, а также облегчают процесс разработки и коммуникации между участниками проекта.

Первым шагом в проектировании системы является создание диаграммы прецедентов, которая позволяет визуализировать взаимодействие актеров и системы. Диаграмма прецедентов помогает определить основные функциональные возможности системы и их взаимосвязи.

На диаграмме прецедентов будут представлены различные прецеденты, такие как "Регистрация пользователя", "Авторизация пользователя", "Просмотр списка доступных инвестиций", "Совершение инвестиции", "Просмотр информации о своих инвестициях" и другие. Для каждого прецедента будет предоставлено расширенное описание, в котором будут указаны основные действия актеров и системы.

Далее будет разработана диаграмма классов, которая позволяет определить структуру системы, включая классы, их атрибуты и методы, а также связи между классами.

На диаграмме классов будут представлены классы, связанные с пользователем, инвестициями, финансовыми операциями и другими сущностями. Также будут показаны связи между классами, такие как наследование и ассоциация.

После этого будет разработана диаграмма состояний, которая позволяет визуализировать различные состояния объектов и переходы между ними.

На диаграмме состояний будут представлены состояния объектов, связанных с инвестициями и финансовыми операциями. Для каждого состояния будет указано, какие операции могут быть выполнены в данном состоянии и какие переходы между состояниями возможны.

В конце данной главы будет проведено обсуждение результатов проектирования информационной системы "Инвестирование свободных средств" и описаны дальнейшие шаги в разработке системы.

## Диаграмма прецедентов и расширенное описание прецедентов

На рисунке 1 изображена диаграмма прецедентов для прототипа информационной системы «Инвестирование свободных средств». На данной диаграмме изображены все варианты использования разрабатываемой ИС.

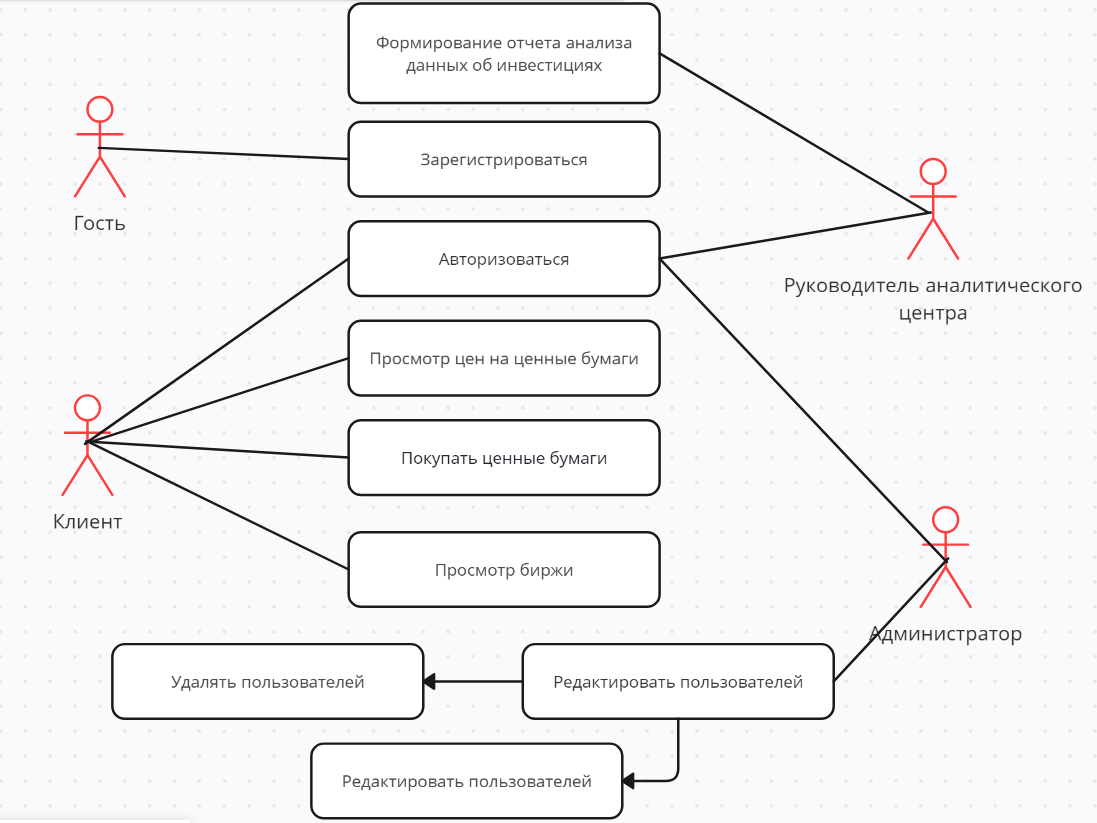
**

Рисунок 1 – Диаграмма прецедентов

Далее рассмотрим детальное описание прецедентов, использованных при разработке прототипа таблица 1.

Таблица 1 – Описание прецедентов

|  |  |  |
| --- | --- | --- |
| № | Наименование | Действие |
| 1 | Регистрация нового пользователя | * Пользователь открывает приложение системы "Инвестирование свободных средств". * Пользователь нажимает на кнопку "Регистрация". * Система отображает форму регистрации, в которую пользователь вводит свои данные, такие как имя, логин, пароль и.т.д. * Пользователь заполняет форму регистрации и нажимает на кнопку "Зарегистрироваться". * Система проверяет введенные данные на корректность и наличие уже существующего пользователя с такими же данными. * Если данные заполнены, верно, и такого пользователя нет, система создает новую учетную запись для пользователя и перенаправляет его на страницу входа. |
| 2 | Вход в систему | * Пользователь открывает приложение системы "Инвестирование свободных средств". * Пользователь нажимает на кнопку "Войти". * Система отображает форму входа, в которую пользователь вводит логин, пароль. * Пользователь заполняет форму входа и нажимает на кнопку "Войти". * Система проверяет введенные данные на правильность и существование пользователя с такими данными. * Если данные введены правильно и пользователь существует, система авторизует пользователя и открывает ему доступ к функционалу системы. |
| 3 | Создание нового инвестиционного портфеля | * Пользователь входит в систему. * Пользователь переходит на страницу управления инвестиционными портфелями. * Пользователь нажимает на кнопку "Создать новый портфель". * Система отображает форму создания нового портфеля, в которой пользователь указывает название портфеля и выбирает начальный баланс. * Пользователь заполняет форму создания портфеля и нажимает на кнопку "Создать". * Система создает новый инвестиционный портфель с указанными данными и отображает его пользователю. * Система проверяет введенные данные на корректность и сохраняет новый инвестиционный портфель в базе данных. * Система отображает пользователю подтверждение успешного создания портфеля и предлагает перейти к просмотру и управлению портфелем. |
| 4 | Просмотр и управление инвестиционными портфелями | * Пользователь входит в систему. * Пользователь переходит на страницу управления инвестиционными портфелями. * Система отображает список всех созданных пользователем инвестиционных портфелей. * Пользователь выбирает портфель, который он хочет просмотреть или управлять. * Система отображает детальную информацию о выбранном портфеле, такую как его наименование, цель, срок, текущий баланс и список вложенных активов. * Пользователь может выполнить следующие действия: * Добавить новый актив в портфель, указав его тип, наименование, количество и стоимость; * Удалить актив из портфеля; * Редактировать информацию о портфеле, например, изменить его наименование, цель или срок; * Посмотреть подробную информацию об активе, такую как его тип, стоимость и долю в портфеле. * Система сохраняет все внесенные изменения в портфель и обновляет информацию о нем в базе данных. |
| 5 | Просмотр, ввод и обновление финансовых показателей | * Пользователь входит в систему. * Пользователь переходит на страницу просмотра инвестиционной информации. * Система отображает различные виды инвестиционной информации, такие как текущие цены активов, исторический график изменения цен, отчеты о доходности и рисках и др. * Пользователь выбирает интересующую его информацию, например, конкретный актив или отчет. * Система отображает выбранную информацию. * Пользователь может проанализировать данную информацию, сравнить данные активы или применить фильтры для получения нужных результатов. * Пользователь может сохранить интересующую его информацию или экспортировать ее в виде файла. |
| 6 | Оформление инвестиций | * Пользователь входит в систему. * Пользователь переходит на страницу торговли активами. * Система отображает доступные для торговли активы, их текущие цены и информацию о ликвидности. * Пользователь выбирает актив, на котором он хочет совершить сделку. * Пользователь указывает тип сделки (покупка или продажа) и желаемое количество активов. * Пользователь подтверждает сделку. * Система проверяет доступность активов для торговли и достаточность средств на счете пользователя. * Если все условия выполнены, система оформляет сделку, обновляет информацию о портфеле и списывает или зачисляет средства соответственно. * Система отображает пользователю подтверждение совершенной сделки и обновленные данные активов и портфеля. |

## Диаграмма классов

В процессе анализа диаграммы прецедентов были выделены 9 сущностей, на основании которых была построена диаграмма классов, изображённая на рисунке 2.

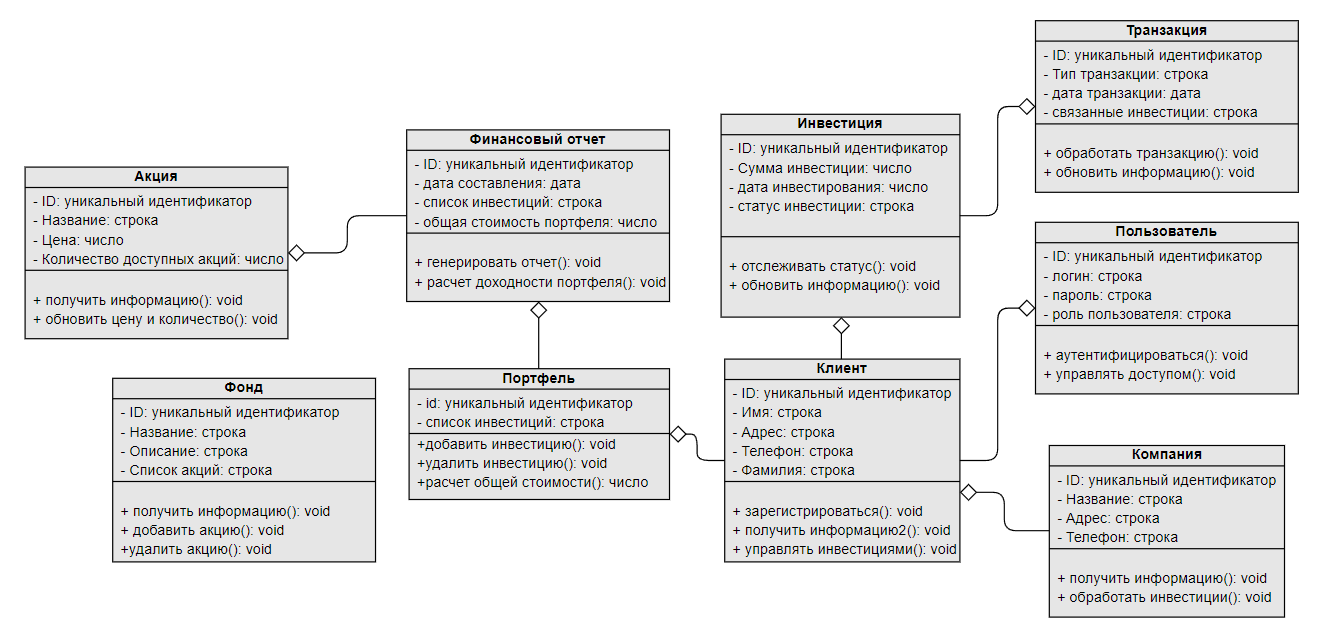


Рисунок 2 - Диаграмма классов ИС «Инвестирование свободных средств»

Опишем более подробно основные сущности в таблице 2.

Таблица 2 – Описание сущностей

|  |  |
| --- | --- |
| Название | Описание |
| Название сущности | Пользователь |
| Словесное описание | Входить в свой профиль |
| Атрибуты | ID: Уникальный идентификатор.  Логин: String;  Пароль: String;  Роль: String; |
| Операции | Аутентификация(): void  Управление доступом(): void |
| Название сущности | Портфель |
| Словесное описание | Просмотр инвестиций |
| Атрибуты | ID: Уникальный идентификатор.  Список инвестиций: String; |
| Операции | Добавить инвестицию(): void  Удалить инвестицию(): void  Расчёт общей стоимости: Integer |
| Название сущности | Компания |
| Словесное описание | Просмотр информации о компании |
| Атрибуты | ID: Уникальный идентификатор.  название: String;  Адрес: String;  Телефон: String; |
| Операции | Получить информацию(): void  Обработать инвестицию(): void |
| Название сущности | Клиент |
| Словесное описание | Информация о клиентах |
| Атрибуты | ID: Уникальный идентификатор.  Имя: String;  Адрес: String;  Телефон: String;  Фамилия: String. |
| Операции | Зарегистрироваться(): void  Получить информацию(): void  Управление инвестициями(): void |
| Название сущности | Фонд |
| Словесное описание | Информация о фонде |
| Атрибуты | ID: Уникальный идентификатор.  название: String;  Описание: String;  Список акций: String; |
| Операции | Получить информацию(): void  Добавить акцию(): void  Удалить акцию(): void |
| Название сущности | Транзакция |
| Словесное описание | Информация о фонде |
| Атрибуты | ID: Уникальный идентификатор.  Тип: String;  Дата: Дата;  Связанные инвестиции: String; |
| Операции | Обработать транзакцию(): void  Обновить информацию(): void |
| Название сущности | Инвестиции |
| Словесное описание | Информация об Инвестициях |
| Атрибуты | ID: Уникальный идентификатор.  Сумма: Integer;  Список: String;  Дата: дата; |
| Операции | Отлеживать статус(): void  Обновить информацию(): void |
| Название сущности | Финансовый отчёт |
| Словесное описание | Просмотр финансового отчёта |
| Атрибуты | ID: Уникальный идентификатор.  Дата составления: String;  Список инвестиций: String;  Общая стоимость профиля: Integer; |
| Операции | Генерировать отчёт(): void  Расчёт доходности портфеля(): void |
| Название сущности | Акция |
| Словесное описание | Информация о Акции |
| Атрибуты | ID: Уникальный идентификатор.  название: String;  Цена: Integer;  Количество допустимых акций: String; |
| Операции | Получить информацию(): void  Обновить цену и количество(): void |

## Диаграмма состояний

Диаграмма состояний (State diagram) - это диаграмма, используемый для моделирования поведения системы, показывающий различные состояния, в которых может находиться объект или система, и переходы между этими состояниями в ответ на определенные события. Диаграмма состояний состоит из состояний, переходов между этими состояниями и событий, вызывающих эти переходы.

На рисунке 3 изображена диаграмма состояний для прецедента, она показывает состояния, которые он может иметь

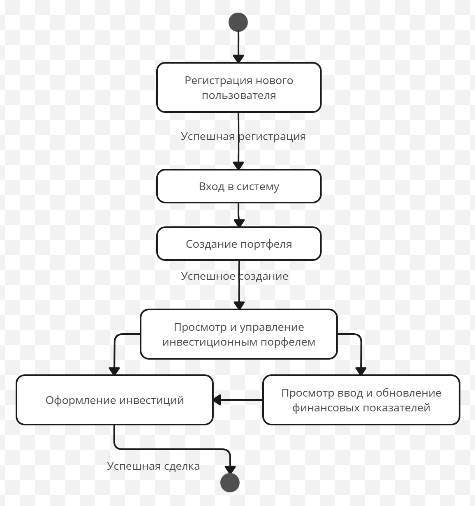


Рисунок 3 - Диаграмма состояний для прецедента

Диаграмма состояний представляет собой модель, которая отображает различные состояния системы и переходы между ними в зависимости от определенных событий или условий. В данном случае, диаграмма состояний для системы "Инвестирование свободных средств" включает следующие основные состояния и переходы:

* Начальное состояние стартовая точка диаграммы, система находится в ожидании начала действий от пользователя;
* Регистрация нового пользователя: Пользователь начинает процесс регистрации, вводя свои данные. После успешной регистрации система переходит к состоянию входа в систему;
* Вход в систему: Пользователь авторизуется в системе с помощью логина и пароля. После успешного входа, система предоставляет доступ к функционалу;
* Создание нового инвестиционного портфеля: Пользователь создает новый инвестиционный портфель, указывая необходимую информацию. После успешного создания портфеля, система переводит пользователя к просмотру и управлению портфелем;
* Просмотр и управление инвестиционными портфелями: Пользователь может просматривать информацию о портфелях, управлять ими, добавлять или удалять активы. Пользователь также может перейти к оформлению инвестиций или просмотру финансовых показателей;
* Просмотр, ввод и обновление финансовых показателей: Пользователь может просматривать различные финансовые показатели, анализировать информацию и обновлять данные. Это состояние также позволяет пользователю экспортировать или сохранить интересующую информацию;
* Оформление инвестиций: Пользователь может совершать инвестиции, торгуя активами. Система проверяет условия сделки и обновляет информацию о портфеле и активах после успешной сделки.

## Проектирование диаграммы последовательности

Диаграммы последовательностей, обычно используемые разработчиками, моделируют взаимодействия между объектами в едином сценарии использования. Они иллюстрируют, как различные части системы взаимодействуют друг с другом для выполнения функции, а также порядок, в котором происходит взаимодействие при выполнении конкретного случая использования.

Диаграмма последовательности представлена на рисунке 4.

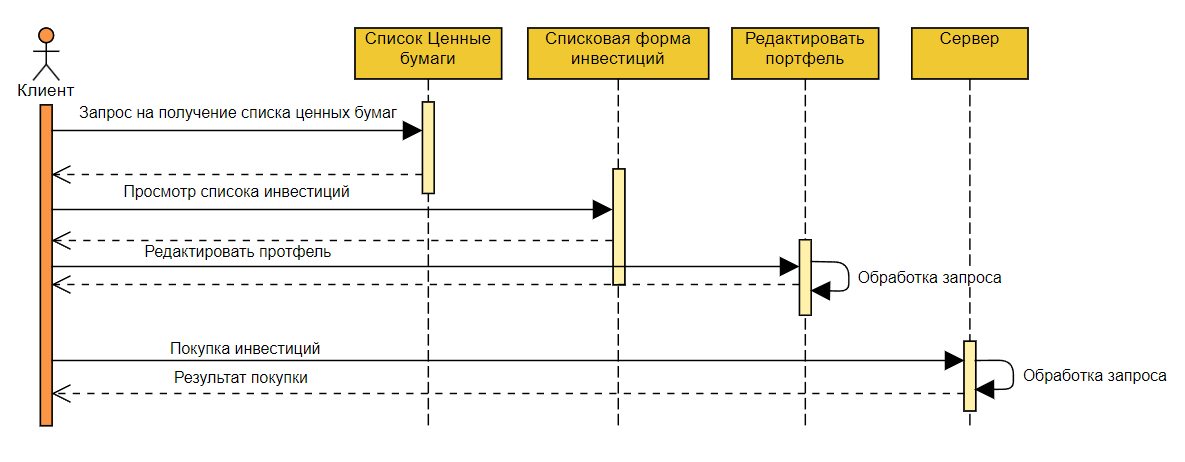


Рисунок 1 - Диаграмма последовательности

На диаграмме представлена последовательность шагов для достижения цели «оформление инвестиции»:

* Первый шагом для достижения цели является обращение клиента к списку ценных бумаг с просьбой оформить ценные бумаги;
* Следующий шаг это переход к списку инвестиций;
* Далее следует добавление инвестиции;
* В результате этих действий достигается конечная цель – Клиент оформляет инвестицию.

## Проектирование диаграммы видов деятельности

Диаграмма видов деятельности - представляет собой тип диаграммы потоков управления, которая используется для моделирования потока действий в системе или процессе. Она показывает последовательность действий, выполняемых системой, и их взаимосвязи.

Диаграмма видов деятельности представлена на рисунке 5.

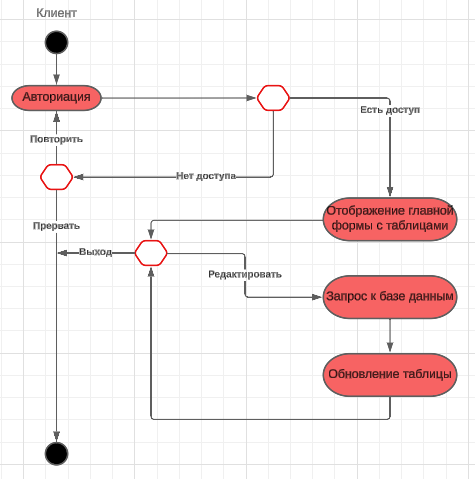


Рисунок 5 - Диаграмма деятельности

Для выполнения Оформления инвестиции необходимо выполнить ряд действий:

* Выполнить авторизацию пользователя. В случае если при авторизации произошла ошибка необходимо выполнить операцию повторно. При возникновении ошибки повторно прервать выполнение авторизации;
* После успешной авторизации в системе пользователю отображается главная форма;
* На главной форме необходимо нажать кнопку «инвестици». Из представленных данных выбрать необходимую запись;
* После выбора необходимой записи нажать кнопку «Сформировать отчёт». После короткого промежутка времени на экран выведется сформированный отчёт об инвестиции.

ВЫВОД

В данной главе было выполнено проектирование ИС, результатом чего стало построение различных диаграмм:

* Диаграмма прецедентов: Определяет внешние события, которые могут инициировать действия в системе, а также актёров, участвующих в этих событиях;
* Диаграмма классов: Описывает классы объектов, которые существуют в системе, а также их атрибуты, операции и взаимосвязи;
* Диаграмма деятельности: Представляет поток деятельности в системе, показывая последовательность действий, выполняемых для достижения конкретной цели;
* Диаграмма последовательности: Изображает взаимодействие между объектами в системе в хронологическом порядке, показывая, какие сообщения отправляются и принимаются;
* Диаграмма состояний: Описывает различные состояния, в которых может находиться объект, а также переходы между этими состояниями.

# РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «ИНВЕСТИРОВАНИЕ СВОБОДНЫХ СРЕДСТВ»

Разработка прототипа информационной системы (ИС) - это процесс создания предварительной версии системы, которая демонстрирует основные функциональные возможности и интерфейс системы. Прототип разрабатывается в начальной стадии проекта, чтобы позволить заказчику или пользователям оценить и проверить концепцию и функциональность системы.

## Общие принципы организации системы

Общие принципы организации информационной системы (ИС) для инвестирования свободных средств могут включать следующие аспекты:

* Автоматизация процессов: ИС должна автоматизировать основные процессы, связанные с управлением инвестициями, включая выбор ценных бумаг, анализ рисков, мониторинг портфеля и генерацию отчетов. Автоматизация позволяет ускорить и упростить процессы, снизить вероятность ошибок и повысить эффективность работы;
* Централизованное хранение данных: ИС должна иметь централизованную базу данных, в которой хранится информация о клиентах, предприятиях, ценных бумагах, операциях и других важных данных. Централизованное хранение обеспечивает доступность и надежность данных, а также облегчает анализ и отчетность;
* Безопасность данных: ИС должна обеспечивать высокий уровень безопасности данных, особенно учитывая конфиденциальность и чувствительность информации о клиентах и их средствах. Это может включать меры защиты данных, авторизацию и аутентификацию пользователей, резервное копирование и мониторинг системы на предмет возможных угроз;
* Интеграция с внешними системами: ИС может взаимодействовать с другими внешними системами, такими как системы биржевой торговли, системы учета и отчетности, а также системы коммуникации с клиентами. Интеграция позволяет обмениваться данными и автоматически выполнять операции, упрощая процессы и повышая точность;
* Аналитика и отчетность: ИС должна предоставлять возможности анализа данных и генерации отчетов, которые помогут принимать информированные решения по управлению инвестициями. Это может включать анализ рынка, оценку рисков, отслеживание производительности портфеля и создание пользовательских отчетов;
* Пользовательский интерфейс: ИС должна иметь удобный и интуитивно понятный пользовательский интерфейс, который позволит пользователям легко взаимодействовать с системой, выполнять операции, просматривать данные и получать необходимую информацию;
* Обучение и поддержка: Разработчики и пользователи ИС должны быть обучены использованию системы, а также обеспечена поддержка и техническая помощь в случае возникновения проблем или вопросов.

## Организация доступа к данным

Данные информационной системы хранятся в базе данных реляционного типа. Таким образом, прежде выделенные сущности были преобразованы в таблицы, с небольшой корректировкой первых. На рисунке 6 представлена схема базы данных.

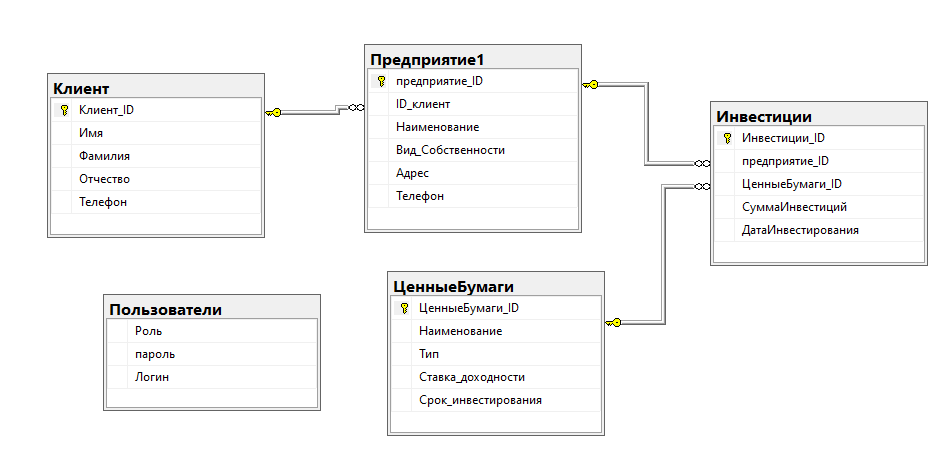


Рисунок 6 - Схема БД и взаимосвязи между её таблицами

## Организация бизнес-логики и интерфейса

Структура ИС инвестирования свободных средств может включать следующие компоненты:

Аутентификация и авторизация:

* Регистрация и аутентификация пользователей с использованием уникальных учетных записей;
* Управление правами доступа на основе ролей пользователей (например, администратор, аналитик);
* Защита данных и обеспечение конфиденциальности информации.

Управление клиентами:

* Создание и обновление профилей клиентов с информацией о предприятии, включая название, вид собственности, адрес и контактный телефон;
* Управление, списком клиентов, включая добавление, поиск и фильтрацию;
* Возможность просмотра и редактирования информации о клиенте.

Управление ценными бумагами:

* Создание и обновление записей о ценных бумагах, включая название, тип и уникальный код;
* Управление списком ценных бумаг, включая добавление, поиск и фильтрацию;
* Возможность просмотра и редактирования информации о ценных бумагах.

Управление портфелями:

* Создание и обновление портфелей клиентов, включая выбор ценных бумаг для инвестирования и распределение средств;
* Отслеживание производительности портфелей и вычисление прибыли и рисков;
* Возможность просмотра и анализа данных о портфелях клиентов.

Аналитика и отчетность:

* Генерация отчетов о производительности портфелей, включая графики, диаграммы и числовые показатели;
* Анализ рыночной ситуации и тенденций для принятия обоснованных решений по инвестициям;
* Предоставление пользователю возможности настройки и фильтрации отчетов.

Взаимодействие с клиентами:

* Возможность обмена сообщениями и консультаций между аналитиками и клиентами;
* Предоставление информации о новых инвестиционных возможностях и обновлениях

## Организация интерфейса

Организация интерфейса приложения в ИС инвестирования свободных средств играет важную роль в обеспечении удобства использования и эффективности работы пользователей. При разработке интерфейса следует учитывать несколько ключевых аспектов.

Во-первых, необходимо продумать навигацию, чтобы пользователи могли легко перемещаться по различным разделам приложения. Логическая структура навигации, такая как меню, вкладки или боковая панель, поможет предоставить доступ к основным функциям и разделам.

Во-вторых, интерфейс должен быть интуитивно понятным. Размещение элементов интерфейса должно быть продуманным, чтобы они были легко обнаруживаемы пользователями. Использование знакомых и понятных иконок и меток поможет пользователям быстро понять функциональность и действия.

В-третьих, информативные дашборды являются важной частью интерфейса. Они предоставляют пользователю обзорную информацию о его портфеле, активности и результативности инвестиций. Графики, диаграммы и числовые показатели используются для наглядного отображения данных.

В-четвертых, важно предоставить возможность настройки интерфейса. Пользователям следует дать возможность настраивать интерфейс в соответствии с их предпочтениями. Это может включать выбор отображаемых данных, типов графиков и уведомлений.

В-пятых, интерфейс должен обеспечивать простоту взаимодействия. Он должен быть интуитивно понятным и простым в использовании. Инструкции и подсказки помогут пользователям выполнить операции, а минимальное количество кликов упростит выполнение задач.

В-шестых, следует учесть адаптивность и доступность интерфейса. Он должен быть адаптирован к различным типам устройств и экранам, чтобы пользователи могли использовать приложение на компьютерах, планшетах и мобильных устройствах. Также важно обеспечить доступность для пользователей с ограниченными возможностями, учитывая принципы универсального доступа.

Наконец, проведение тестирования пользовательского опыта позволит оценить удобство использования интерфейса и выявить возможные проблемы или улучшения. Обратная связь пользователей может помочь внести корректировки и улучшить интерфейс.

Макет интерфейса приложения для инвестирования в ПРИЛОЖЕНИИ Б разработан с учетом принципов удобства использования и эффективности.

Вывод

В данной главе было выполнено общее описание принципа организации системы, организация доступа к данным, где была представлена физическая модель данных.

# ЗАКЛЮЧЕНИЕ

Прототип информационной системы "Инвестирование свободных средств" представляет собой первоначальную версию будущего программного продукта, который демонстрирует базовые функции и возможности предполагаемой системы. Несмотря на свою небольшую функциональность, этот прототип уже способен показать потенциальную ценность и преимущества, которые могут быть достигнуты при полной реализации и внедрении системы.

Информационная система "Инвестирование свободных средств" направлена на улучшение процессов работы с инвесторами, обеспечивая более быструю и эффективную коммуникацию, анализ данных и принятие решений. Автоматизированные функции этой системы могут значительно упростить взаимодействие с инвесторами, ускорить процессы принятия инвестиционных решений и снизить вероятность ошибок и потери данных.

Использование данного прототипа позволяет не только оценить потенциал будущей информационной системы, но и выявить возможные улучшения и доработки, необходимые для ее оптимального функционирования. Постепенное развитие и доработка этой системы могут привести к созданию мощного инструмента для эффективного управления инвестициями и улучшения финансовых результатов компании.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Информационные технологии в профессиональной деятельности: учебник / Филимонова Е.В. — Москва: КноРус, 2021. — 482 с.
2. Э. Дастин, Тестирование программного обеспечения. Внедрение, управление и автоматизация / Э. Дастин, Д. Рэшка, Д. Пол; Пер. с англ. М. Павлов. — М.: Лори, 2017. — 567 c.
3. И.Г.Семакин, Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования / И.Г.Семакин, А.П.Шестаков. — М.: Издательский центр «Академия», 2017. — 304
4. Иванов, А.А. Автоматизация технологических процессов и производств: Учебное пособие / А.А. Иванов. — М.: Форум, 2018. — 224 c
5. Компьютерное моделирование и исследование систем автоматического управления: Учебно-методическое пос : учебнометодическое пособие / Федоров С.Е. — Москва : Русайнс, 2020. — 92 с.

Маркин, А. В. Программирование на sql в 2 ч. Часть 2: учебник и практикум для бакалавриата и магистратуры / А. В. Маркин. — М.: Издательство Юрайт, 2019. — 292 с.

1. Ангарский промышленно-экономический техникум – ГБПОУ ИО «АПЭТ» [Электронный ресурс]: Официальный сайт ГБПОУ ИО «Ангарский промышленно-экономический техникум», 2008-2022. URL: <http://a-pet.ru/> (дата обращения: 22.02.2024).
2. Добро пожаловать в интегрированную среду разработки Visual Studio [Электронный ресурс] – Microsoft, 2022. Режим доступа: https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022 (дата обращения: 22.02.2024
3. Microsoft Visual Studio [Электронный ресурс] – Wikipedia, 2022. Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_Visual\_Studio (дата обращения: 22.02.2024).
4. SQL [Электронный ресурс] – htmlacademy.ru, 2022. Режим доступа: https://htmlacademy.ru/tutorial/php/sql (дата обращения 22.02.2024).
5. Создание макетов [Электронный ресурс] – maket-master.ru, 2022. Режим доступа: http://maket-master.ru (дата обращения: 22.02.2024).
6. Расчет экономической эффективности [Электронный ресурс] – business.ru, 2022. Режим доступа: https://www.business.ru/article/1810-raschet-ekonomicheskoy-effektivnosti (дата обращения: 22.02.2024).
7. Интеграция приложений [Электронный ресурс] – osp.ru, 2022. Режим доступа: https://www.osp.ru/os/2006/09/3776464 (дата обращения: 22.02.2024).
8. Электронный фонд правовых и нормативно-технических документов [электронный ресурс] URL: https://docs.cntd.ru/document/1200082859(дата обращения: 22.02.2024).
9. IT MASTER GURU [электронный ресурс] URL:https://itmaster.guru/nastrojka-interneta/kak-po-lokalnoy-seti-podklyuchit printer.html(дата обращения: 22.02.2024).
10. Справочник [электронный ресурс] URL:https://spravochnick.ru/informatika/soprovozhdenie\_i\_obsluzhivanie\_progra mnogo\_obespecheniya\_kompyuternyh\_sistem(дата обращения: 22.02.2024).

# ПРИЛОЖЕНИЕ А

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Министерство образования Иркутской области

Государственное бюджетное профессиональное образовательное учреждение

Иркутской области

**«Ангарский промышленно-экономический техникум»**

(ГБПОУ ИО «АПЭТ»)

|  |
| --- |
| Автоматизация системы инвестирование свободных средств |
| Наименование ИС |
| ИНВЕСТИРОВАНИЕ СВОБОДНЫХ СРЕДСТВ |
| Сокращенное наименование ИС |
| **ТЕХНИЧЕСКОЕ ЗАДАНИЕ** |
| Действует с 27.02.2024 |

|  |
| --- |
| Составитель |
| Коновалов Сергей Валадимирович |
| ФИО сотавителя |
| Разработчик |
| Должность составителя |

г. Ангарск, 2024 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ. 3](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338856)

[1.Основание для разработки 31](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338857)

[2.Назначение разработки 32](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338858)

[3.Технические требования к программе или программному изделию 32](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338859)

[3.1. Требования к функциональным характеристикам 33](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338860)

[3.2. Требования к надежности 33](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338861)

[3.3. Условия эксплуатации 33](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338862)

[3.4. Требования к составу и параметрам технических средств 34](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338863)

[3.5. Требования к информационной и программной совместимости 34](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338864)

[3.6. Требования к транспортированию и хранению 34](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338865)

[3.7. Специальные требования 34](file:///\\apetfs\ИСПП-3\Коновалов_СВ\4%20курс\Учебная%20практика\ИС%20Автотрансопорт\Доки\Приложения\1%20ТЗ..docx#_Toc28338866)

# ВВЕДЕНИЕ

В настоящее время автоматизация и создание информационных систем считаются одними из самых трудоемких областей деятельности в технологическом обществе. Одной из главных причин такого активного развития этой области является то, что автоматизация является основой для революционных изменений в управленческих процессах, которые играют ключевую роль в деятельности людей и общества.

В результате возникают системы управления, которые направлены на поддержание и улучшение работы объекта с помощью устройства управления (комплекса средств сбора, обработки, передачи информации и формирования управляющих сигналов или команд).

Информационная система, в свою очередь, представляет собой систему, предоставляющую авторизованному персоналу данные или информацию, связанную с организацией.

Основная цель информационных систем заключается в том, чтобы производить необходимую информацию для эффективного управления всеми ресурсами организации.

# Основание для разработки

Основанием для работы служит договор между организациями. Темой разработки является «Автоматизация системы инвестирование свободных средств».

# Назначение разработки

Актуальность разработки автоматизированных систем управления определяется необходимостью внедрения системы электронного документооборота на предприятиях, вызванной большими объемами работы с документами, поиском, утверждением, согласованием документов, автоматизацией движения конструкторско-технологической документации, экономией времени, обеспечением информационной безопасности и повышением исполнительской дисциплины для контролируемости технологических процессов.

# Технические требования к программе или программному изделию

## Требования к функциональным характеристикам.

* Система должна обеспечивать возможность регистрации пользователей и авторизации для доступа к функциональности;
* Система должна предоставлять информацию о доступных инвестиционных инструментах, их доходности и рисках;
* Пользователи должны иметь возможность создавать инвестиционные портфели, определять соотношение между инструментами и проводить операции купли-продажи;
* Система должна предоставлять информацию о прогнозируемой доходности и рисках созданных портфелей;
* Пользователи должны иметь возможность отслеживать изменения стоимости своих портфелей и проводить анализ инвестиций;
* Система должна обеспечивать возможность внесения дополнительных вкладов или снятия средств со счета;
* Пользователи должны иметь доступ к консультациям и рекомендациям по инвестированию;
* Система должна обеспечивать возможность закрытия аккаунта или вывода всех средств по запросу пользователя.

## Требования к надежности

Для обеспечения надежности необходимо проверять корректность вводимых данных инвестором.

## Условия эксплуатации

Данный программный продукт предполагается эксплуатировать в условиях, благоприятных для оператора ЭВМ. Предполагается, что обслуживать данный программный продукт будет один человек –администратор. Администратор должен обеспечивать нормальное функционирование программного продукта и ввести базу об инвесторах пользующийся этим программным продуктом.

## Требования к составу и параметрам технических средств

Для нормального функционирования программного средства должны соответствовать:

* процессор Intel Pentium III с тактовой частотой от 75 Мгц и выше;
* оперативная память от 128 Мб;
* жесткий диск c объемом памяти не менее 60 Мб свободного дискового пространства;
* видеокарта, с объемом оперативной памяти 512 Мб.

## Требования к информационной и программной совместимости

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы:

* Windows NT 10.0 (2018);
* Windows NT 10.0 (2019);
* Windows NT 10.0 (2022).

## Требования к транспортированию и хранению

Программа поставляется на разных носителях таких как Usb флеш накопителе или на лазерном носителе информации.

## Специальные требования.

Разрабатываемый программный продукт должен обладать следующими требованиями:

* программа должна иметь простой интуитивно понятный интерфейс;
* редактировать устаревшую информацию;
* обеспечивать целостность хранимой информации.

# ПРИЛОЖЕНИЕ Б

РЕАЛИЗАЦИЯ ЧАСТИ ФУНКЦИОНАЛА ИНФОРМАЦИОННОЙ СИСТЕМЫ «ИНВЕСТИРОВАНИЕ СВОБОДНЫХ СРЕДСТВ»

Реализация части функционала информационной системы (ИС) включает в себя создание и разработку необходимых компонентов и модулей, которые обеспечивают желаемый функционал системы. Этот процесс включает в себя анализ требований, проектирование архитектуры системы, написание кода и тестирование функционала.

На этапе анализа и проектирования определяются требования к функционалу ИС, проводится изучение бизнес-процессов, и определяются основные функции, которые должны быть реализованы. Затем проектируется структура системы и разрабатывается план реализации.

После этого команда разработчиков приступает к созданию необходимых компонентов и модулей, используя выбранные технологии и инструменты. Они пишут код, который реализует нужный функционал и обеспечивает взаимодействие с другими компонентами системы.

После завершения разработки проводится тестирование функционала ИС. Это включает в себя проверку соответствия требованиям, исправление ошибок и проверку работоспособности и стабильности системы. Тестирование может включать модульные тесты, интеграционное тестирование и системное тестирование.

После успешного завершения тестирования функционал ИС готов к внедрению. На этом этапе система разворачивается на целевой инфраструктуре, настраивается и готовится к использованию. Пользователи обучаются работе с новым функционалом и начинают его использование.

После внедрения функционала ИС организуется его сопровождение и поддержка. Это включает в себя обновление и исправление ошибок, а также обеспечение бесперебойной работы системы. При необходимости может выполняться дальнейшая разработка и расширение функционала.

Реализация части функционала информационной системы требует планирования, согласованной работы разработчиков и тестировщиков, а также контроля качества. Она основана на анализе требований, проектировании, разработке, тестировании, внедрении и поддержке системы.

Создание макетов

Макет - это визуальное представление дизайна приложения. Он служит основой для разработки и реализации окончательного продукта.

На рисунке 7 представлен макет формы авторизации

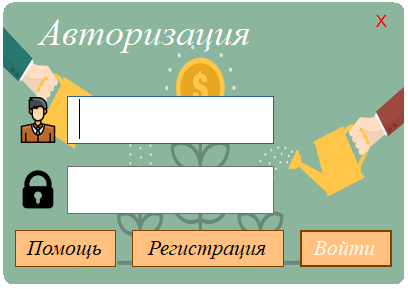


Рисунок 7 – Макет авторизации

На рисунке 8 представлен макет формы регистрации

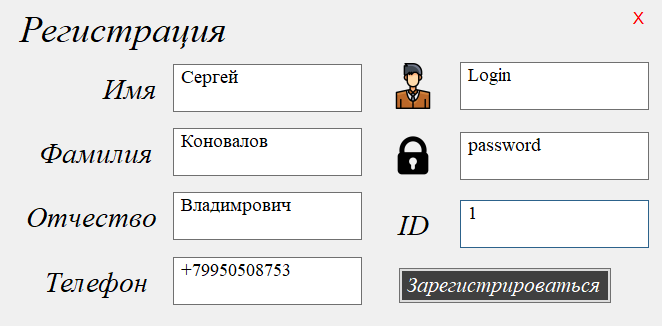


Рисунок 8 – Макет регистрации

На рисунке 9 представлен макет формы администратора

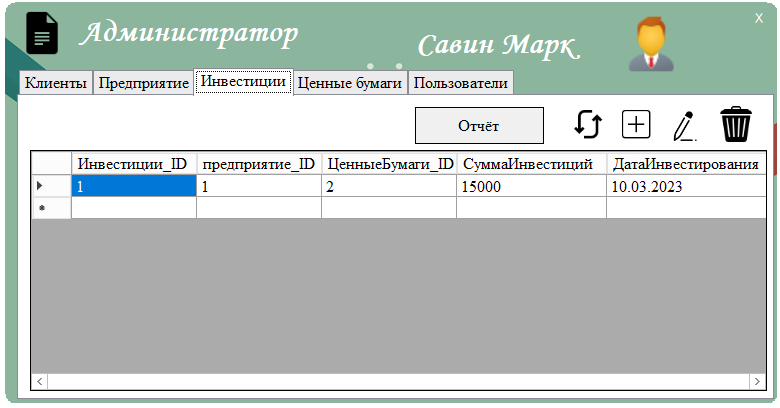


Рисунок 9 –Макет главной формы администратора

На рисунке 10 представлен макет формы добавление портфеля

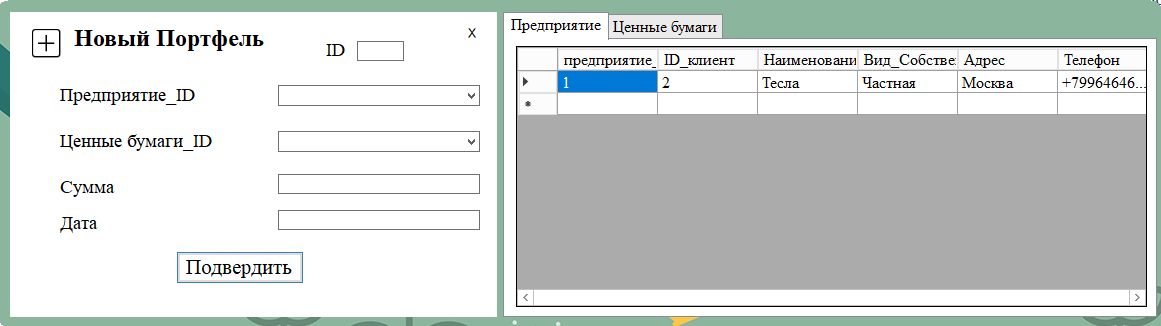


Рисунок 10 –Макет формы добавление портфеля

Соединение с базой данных

Файл sys.syslogins хранит в себе информацию о подключении к базе данных, далее следует листинг фрагмента этого файла, который отвечает за связь с БД.

public static string rol = "";

public string CmdText = "SELECT login,Name\_ak,Pas,buy + sale AS 'Стоимость аккаунта',guard,phon,phon\_date FROM [ak]";

public string ConnString = "Data Source=192.168.1.3;" +

"Initial Catalog=Коновалов\_СВ\_курсовая;" +

"User id=КоноваловСергей;" +

"Password=1202;";//Глобальная БД

Описание классов

Класс в объектно-ориентированном программировании представляет собой шаблон или описание, по которому создаются объекты. Объект, в свою очередь, является конкретным экземпляром класса, обладающим своим уникальным состоянием. Поля класса, также известные как атрибуты, представляют переменные, хранящие данные, относящиеся к объекту. Методы класса, или операции, определяют поведение объекта и могут изменять его состояние.

В данном проекте находятся следующие Классы Avto.cs, registration.cs, Админестратор.cs. Инвестиции.cs. Добавление портфеля.cs. Далее будет приведён частичных листинг этих Классов.

Класс Avto

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Бюро\_технической\_инвентаризации

{

public partial class Авторизация : Form

{

public System.Windows.Forms.Timer notificationTimer;

public System.Windows.Forms.Timer systemTimeTimer;

public Авторизация()

{

InitializeComponent();

notificationTimer = new System.Windows.Forms.Timer();

notificationTimer.Interval = 30 \* 60 \* 1000; // Interval set to 30 minutes

notificationTimer.Tick += NotificationTimer\_Tick;

notificationTimer.Start();

systemTimeTimer = new System.Windows.Forms.Timer(); // Initialize the timer

systemTimeTimer.Interval = 1000; // Set the interval to 1 second

systemTimeTimer.Tick += SystemTimeTimer\_Tick; // Set the Tick event handler

systemTimeTimer.Start(); // Start the timer

}

private void NotificationTimer\_Tick(object sender, EventArgs e)

{

// Show a message box every 30 minutes

MessageBox.Show("С момента последнего действия прошло 30 минут!!!");

}

private void SystemTimeTimer\_Tick(object sender, EventArgs e)

{

// Update the system time display every 1 second

label3.Text = DateTime.Now.ToString();

}

private void Client\_Load1(object sender, EventArgs e)

{

// Initially display the system time.

label3.Text = DateTime.Now.ToString();

}

protected override void OnPaint(PaintEventArgs e)

{

Rectangle bounds = new Rectangle(0, 0, Width, Height);

int CornerRadius = 20;

GraphicsPath path = new GraphicsPath();

path.AddArc(bounds.X, bounds.Y, CornerRadius, CornerRadius, 180, 90);

path.AddArc(bounds.X + bounds.Width - CornerRadius, bounds.Y, CornerRadius, CornerRadius, 270, 90);

path.AddArc(bounds.X + bounds.Width - CornerRadius, bounds.Y + bounds.Height - CornerRadius, CornerRadius, CornerRadius, 0, 90);

path.AddArc(bounds.X, bounds.Y + bounds.Height - CornerRadius, CornerRadius, CornerRadius, 90, 90);

path.CloseAllFigures();

Region = new Region(path);

base.OnPaint(e);

}

public static string rol = "";

public string CmdText = "SELECT login,Name\_ak,Pas,buy + sale AS 'Стоимость аккаунта',guard,phon,phon\_date FROM [ak]";

public string ConnString = "Data Source=192.168.1.3;" +

"Initial Catalog=Коновалов\_СВ\_курсовая;" +

"User id=КоноваловСергей;" +

"Password=1202;";//Глобальная БД

private void label2\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

public static string connectString = @"Data Source=sqlLite;Initial Catalog=Коновалов\_СВ\_курсовая;Integrated Security=true";

public static string a, b;

private SqlConnection myConnection;

public SqlDataAdapter da = null;

public DataSet ds = new DataSet();

public SqlConnection connP = new SqlConnection();

private void button1\_Click\_1(object sender, EventArgs e)

{

if (loginmenu.Text == "" || pasportmenu.Text == "")

{

MessageBox.Show("Все поля должны быть заполнены");

}

else

{

myConnection = new SqlConnection(connectString);

myConnection.Open();

string query = "SELECT Логин, пароль, Роль From Пользователи";

DataTable dt = new DataTable();

da = new SqlDataAdapter(query, myConnection);

da.Fill(dt);

Boolean flag = false;

foreach (DataRow row in dt.Rows)

{

if (row["Логин"].ToString() == loginmenu.Text && row["пароль"].ToString() == pasportmenu.Text)

{

flag = true;

break;

}

}

if (flag == true)

{

this.Hide();

Админестратор frm2 = new Админестратор();

frm2.Show();

}

else

{

MessageBox.Show("Неверный логин или пароль");

}

myConnection.Close();

}

}

private void Авторизация\_Load(object sender, EventArgs e)

{

loginmenu.MaxLength = 10;

pasportmenu.MaxLength = 10;

}

private void button1\_MouseMove(object sender, MouseEventArgs e)

{

Button button = (Button)sender;

button.ForeColor = Color.Black;

}

private void button1\_MouseLeave(object sender, EventArgs e)

{

Button button = (Button)sender;

button.ForeColor = Color.White;

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Help frm = new Help();

frm.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Hide();

registration frm = new registration();

frm.Show();

}

}

}

Класс registration

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Бюро\_технической\_инвентаризации

{

public partial class registration : Form

{

public registration()

{

InitializeComponent();

}

SqlConnection con = new SqlConnection(@"Data Source=sqlLite;Initial Catalog=Коновалов\_СВ\_курсовая;Integrated Security=true");

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

try

{

if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "" || textBox4.Text == "" || textBox5.Text == "" || textBox6.Text == "" || textBox7.Text == "")

{

MessageBox.Show("Заполните все поля");

}

else

{

string connectionString = @"Data Source=sqlLite;Initial Catalog=Коновалов\_СВ\_курсовая;Integrated Security=True";

string query = "INSERT INTO [Пользователи] ([Роль],[пароль],[Логин]) VALUES (@rol, @par, @log)";

using (SqlConnection connection = new SqlConnection(connectionString))

using (SqlCommand command = new SqlCommand(query, connection))

{

string rol = "инвестор";

string login = Convert.ToString(textBox2.Text);

string parol = Convert.ToString(textBox3.Text);

command.Parameters.AddWithValue("@log", login);

command.Parameters.AddWithValue("@par", parol);

command.Parameters.AddWithValue("@rol", rol);

connection.Open();

command.ExecuteNonQuery();

}

string query1 = "INSERT INTO [Клиент] ([Клиент\_ID],[Имя],[Фамилия],[Отчество],[телефон]) VALUES (@id, @name, @fam, @otch, @tel)";

using (SqlConnection connection1 = new SqlConnection(connectionString))

using (SqlCommand command1 = new SqlCommand(query1, connection1))

{

string ID = Convert.ToString(textBox6.Text);

string name = Convert.ToString(textBox1.Text);

string fam = Convert.ToString(textBox5.Text);

string otch = Convert.ToString(textBox4.Text);

string tel = Convert.ToString(textBox7.Text);

command1.Parameters.AddWithValue("@id", ID);

command1.Parameters.AddWithValue("@name", name);

command1.Parameters.AddWithValue("@fam", fam);

command1.Parameters.AddWithValue("@otch", otch);

command1.Parameters.AddWithValue("@tel", tel);

connection1.Open();

command1.ExecuteNonQuery();

}

this.Close();

Авторизация frm = new Авторизация();

frm.Show();

}

}

catch

{

MessageBox.Show("Используёте другой ID");

}

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

Класс Админестратор

using System;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Runtime.InteropServices;

using System.Windows.Forms;

using System.Windows.Threading;

namespace Бюро\_технической\_инвентаризации

{

public partial class Админестратор : Form

{

public Админестратор()

{

InitializeComponent();

}

protected override void OnPaint(PaintEventArgs e)

{

Rectangle bounds = new Rectangle(0, 0, Width, Height);

int CornerRadius = 20;

GraphicsPath path = new GraphicsPath();

path.AddArc(bounds.X, bounds.Y, CornerRadius, CornerRadius, 180, 90);

path.AddArc(bounds.X + bounds.Width - CornerRadius, bounds.Y, CornerRadius, CornerRadius, 270, 90);

path.AddArc(bounds.X + bounds.Width - CornerRadius, bounds.Y + bounds.Height - CornerRadius, CornerRadius, CornerRadius, 0, 90);

path.AddArc(bounds.X, bounds.Y + bounds.Height - CornerRadius, CornerRadius, CornerRadius, 90, 90);

path.CloseAllFigures();

Region = new Region(path);

base.OnPaint(e);

}

private void Админестратор\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.Пользователи);

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.ЦенныеБумаги". При необходимости она может быть перемещена или удалена.

this.ценныеБумагиTableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.ЦенныеБумаги);

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.Инвестиции". При необходимости она может быть перемещена или удалена.

this.инвестицииTableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.Инвестиции);

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.Предприятие1". При необходимости она может быть перемещена или удалена.

this.предприятие1TableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.Предприятие1);

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.Клиент". При необходимости она может быть перемещена или удалена.

this.клиентTableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.Клиент);

}

private void pictureBox7\_Click(object sender, EventArgs e)

{

Клиент gg = new Клиент();

gg.Show();

}

private void label3\_Click(object sender, EventArgs e)

{

this.Close();

}

SqlConnection con = new SqlConnection(@"Data Source=sqlLite;Initial Catalog=Коновалов\_СВ\_курсовая;Integrated Security=true");

private void pictureBox8\_Click(object sender, EventArgs e)

{

try

{

string commandText = "select \* from Клиент";

SqlDataAdapter itm = new SqlDataAdapter(commandText, con);

DataTable dt2 = new DataTable();

itm.Fill(dt2);

dataGridView1.DataSource = dt2;

con.Close();

}

catch

{

MessageBox.Show("Нечего обновлять");

}

}

private void pictureBox4\_Click(object sender, EventArgs e)

{

try

{

string commandText1 = "select \* from Предприятие1";

SqlDataAdapter itm1 = new SqlDataAdapter(commandText1, con);

DataTable dt21 = new DataTable();

itm1.Fill(dt21);

dataGridView2.DataSource = dt21;

con.Close();

}

catch

{

MessageBox.Show("Нечего обновлять");

}

}

private void pictureBox11\_Click(object sender, EventArgs e)

{

try

{

string commandText = "select \* from Инвестиции";

SqlDataAdapter itm = new SqlDataAdapter(commandText, con);

DataTable dt2 = new DataTable();

itm.Fill(dt2);

dataGridView3.DataSource = dt2;

con.Close();

}

catch

{

MessageBox.Show("Нечего обновлять");

}

}

private void pictureBox23\_Click(object sender, EventArgs e)

{

try

{

string commandText = "select \* from ЦенныеБумаги";

SqlDataAdapter itm = new SqlDataAdapter(commandText, con);

DataTable dt2 = new DataTable();

itm.Fill(dt2);

dataGridView6.DataSource = dt2;

con.Close();

}

catch

{

MessageBox.Show("Нечего обновлять");

}

}

private void pictureBox9\_Click(object sender, EventArgs e)

{

Добавить\_помещения gg = new Добавить\_помещения();

gg.Show();

}

private void pictureBox12\_Click(object sender, EventArgs e)

{

Инвестиции gg = new Инвестиции();

gg.Show();

}

private void pictureBox24\_Click(object sender, EventArgs e)

{

добавить\_земленой\_участок gg = new добавить\_земленой\_участок();

gg.Show();

}

private void pictureBox6\_Click(object sender, EventArgs e)

{

Редоктирование gg = new Редоктирование();

gg.Show();

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

Редоктирование\_Помещения gg = new Редоктирование\_Помещения();

gg.Show();

}

private void pictureBox10\_Click(object sender, EventArgs e)

{

Редактирование\_литера gg = new Редактирование\_литера();

gg.Show();

}

private void pictureBox22\_Click(object sender, EventArgs e)

{

редактировать\_земельный\_уч gg = new редактировать\_земельный\_уч();

gg.Show();

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

удаление\_д gg = new удаление\_д();

gg.Show();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

Удалить\_п gg = new Удалить\_п();

gg.Show();

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

удалить\_л gg = new удалить\_л();

gg.Show();

}

private void pictureBox21\_Click(object sender, EventArgs e)

{

удалить\_зу gg = new удалить\_зу();

gg.Show();

}

private void pictureBox20\_Click(object sender, EventArgs e)

{

//добавить\_инженера gg = new добавить\_инженера();

//gg.Show();

}

private void pictureBox18\_Click(object sender, EventArgs e)

{

редактировать\_инженер gg = new редактировать\_инженер();

gg.Show();

}

private void pictureBox26\_Click(object sender, EventArgs e)

{

редактировать\_пользователей gg = new редактировать\_пользователей();

gg.Show();

}

private void pictureBox17\_Click(object sender, EventArgs e)

{

удалить\_и gg = new удалить\_и();

gg.Show();

}

private void pictureBox25\_Click\_1(object sender, EventArgs e)

{

try

{

string commandText = "select \* from Пользователи";

SqlDataAdapter itm = new SqlDataAdapter(commandText, con);

DataTable dt2 = new DataTable();

itm.Fill(dt2);

dataGridView7.DataSource = dt2;

con.Close();

}

catch

{

MessageBox.Show("Нечего обновлять");

}

}

private void Выход\_Click(object sender, EventArgs e)

{

this.Close();

Авторизация gg = new Авторизация();

gg.Show();

}

private void pictureBox13\_Click(object sender, EventArgs e)

{

Удаление\_С gg = new Удаление\_С();

gg.Show();

}

private void pictureBox14\_Click\_1(object sender, EventArgs e)

{

добавление\_Сотрудников gg = new добавление\_Сотрудников();

gg.Show();

}

private void button10\_Click(object sender, EventArgs e)

{

this.Hide();

Help2 frm = new Help2();

frm.Show();

}

private void pictureBox15\_Click\_1(object sender, EventArgs e)

{

редактировать\_пользователей frm = new редактировать\_пользователей();

frm.Show();

}

}

}

Класс Инвестиции

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Бюро\_технической\_инвентаризации

{

public partial class Инвестиции : Form

{

public Инвестиции()

{

InitializeComponent();

}

protected override void OnPaint(PaintEventArgs e)

{

Rectangle bounds = new Rectangle(0, 0, Width, Height);

int CornerRadius = 20;

GraphicsPath path = new GraphicsPath();

path.AddArc(bounds.X, bounds.Y, CornerRadius, CornerRadius, 180, 90);

path.AddArc(bounds.X + bounds.Width - CornerRadius, bounds.Y, CornerRadius, CornerRadius, 270, 90);

path.AddArc(bounds.X + bounds.Width - CornerRadius, bounds.Y + bounds.Height - CornerRadius, CornerRadius, CornerRadius, 0, 90);

path.AddArc(bounds.X, bounds.Y + bounds.Height - CornerRadius, CornerRadius, CornerRadius, 90, 90);

path.CloseAllFigures();

Region = new Region(path);

base.OnPaint(e);

}

SqlConnection sqlConnection = new SqlConnection(@"Data Source=sqlLite;Initial Catalog=Коновалов\_СВ\_курсовая;Integrated Security=true");

private void button1\_Click(object sender, EventArgs e)

{

try

{

if (string.IsNullOrWhiteSpace(textBox1.Text) || string.IsNullOrWhiteSpace(textBox2.Text) ||

string.IsNullOrWhiteSpace(textBox5.Text) || string.IsNullOrWhiteSpace(textBox4.Text))

{

MessageBox.Show("Заполните все поля!");

}

else

{

string pred = Convert.ToString(textBox1.Text);

string chen\_bum = Convert.ToString(textBox2.Text);

int cumma = Convert.ToInt32(textBox4.Text);

string y = Convert.ToString(textBox5.Text);

string date = Convert.ToString(textBox3.Text);

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sqlConnection;

sqlConnection.Open();

using (SqlCommand command = new SqlCommand("Insert into [Инвестиции](Инвестиции\_ID, предприятие\_ID, ЦенныеБумаги\_ID, СуммаИнвестиций, ДатаИнвестирования)values(@y, @pred, @chen\_bum, @cumma, @date)", sqlConnection))

{

command.Parameters.AddWithValue("@pred", pred);

command.Parameters.AddWithValue("@chen\_bum", chen\_bum);

command.Parameters.AddWithValue("@cumma", cumma);

command.Parameters.AddWithValue("@y", y);

command.Parameters.AddWithValue("@date", date);

command.ExecuteNonQuery();

sqlConnection.Close();

this.Close();

}

}

}

}

catch

{

MessageBox.Show("Возможно вы вели неправильно");

}

}

private void Инвестиции\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.ЦенныеБумаги". При необходимости она может быть перемещена или удалена.

this.ценныеБумагиTableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.ЦенныеБумаги);

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.Предприятие1". При необходимости она может быть перемещена или удалена.

this.предприятие1TableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.Предприятие1);

}

}

}

Класс Редактирование портфеля

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Бюро\_технической\_инвентаризации

{

public partial class Редактирование\_литера : Form

{

public Редактирование\_литера()

{

InitializeComponent();

}

protected override void OnPaint(PaintEventArgs e)

{

Rectangle bounds = new Rectangle(0, 0, Width, Height);

int CornerRadius = 20;

GraphicsPath path = new GraphicsPath();

path.AddArc(bounds.X, bounds.Y, CornerRadius, CornerRadius, 180, 90);

path.AddArc(bounds.X + bounds.Width - CornerRadius, bounds.Y, CornerRadius, CornerRadius, 270, 90);

path.AddArc(bounds.X + bounds.Width - CornerRadius, bounds.Y + bounds.Height - CornerRadius, CornerRadius, CornerRadius, 0, 90);

path.AddArc(bounds.X, bounds.Y + bounds.Height - CornerRadius, CornerRadius, CornerRadius, 90, 90);

path.CloseAllFigures();

Region = new Region(path);

base.OnPaint(e);

}

SqlConnection con = new SqlConnection(@"Data Source=sqlLite;Initial Catalog=Коновалов\_СВ\_курсовая;Integrated Security=true");

private void button1\_Click(object sender, EventArgs e)

{

//try

//{

string pred = Convert.ToString(textBox1.Text);

string chen\_bum = Convert.ToString(textBox2.Text);

int cumma = Convert.ToInt32(textBox4.Text);

string y = Convert.ToString(textBox5.Text);

string date = Convert.ToString(textBox3.Text);

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

string zap = "UPDATE [Инвестиции] SET [предприятие\_ID] = @pred, [ЦенныеБумаги\_ID] = @chen\_bum, [СуммаИнвестиций] = @cumma, [ДатаИнвестирования] = @date WHERE [Инвестиции\_ID] = @y";

cmd.CommandText = zap;

cmd.Parameters.AddWithValue("@pred", pred);

cmd.Parameters.AddWithValue("@chen\_bum", chen\_bum);

cmd.Parameters.AddWithValue("@cumma", cumma);

cmd.Parameters.AddWithValue("@y", y);

cmd.Parameters.AddWithValue("@date", date);

cmd.ExecuteNonQuery();

con.Close();

this.Close();

//}

//catch

//{

// MessageBox.Show("Вы ввели неправильно, нельзя редактировать если в БД нет записи");

//}

}

private void label7\_Click(object sender, EventArgs e)

{

this.Close();

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar <= 48 || e.KeyChar >= 59) && e.KeyChar != 8)

e.Handled = true;

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar <= 48 || e.KeyChar >= 59) && e.KeyChar != 8)

e.Handled = true;

}

private void Редактирование\_литера\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.Предприятие1". При необходимости она может быть перемещена или удалена.

this.предприятие1TableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.Предприятие1);

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.ЦенныеБумаги". При необходимости она может быть перемещена или удалена.

this.ценныеБумагиTableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.ЦенныеБумаги);

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.Клиент". При необходимости она может быть перемещена или удалена.

this.клиентTableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.Клиент);

// TODO: данная строка кода позволяет загрузить данные в таблицу "коновалов\_СВ\_курсоваяDataSet.Инвестиции". При необходимости она может быть перемещена или удалена.

this.инвестицииTableAdapter.Fill(this.коновалов\_СВ\_курсоваяDataSet.Инвестиции);

textBox1.MaxLength = 50;

}

private void textBox5\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar <= 48 || e.KeyChar >= 59) && e.KeyChar != 8)

e.Handled = true;

}

}

}

Создание Форм

Для создания Форм использовался редактор VisualStudio. Визуальное отображение этих форм можно увидеть на рисунках, Интерфейс данных форм представлен на рисунках 10-13.

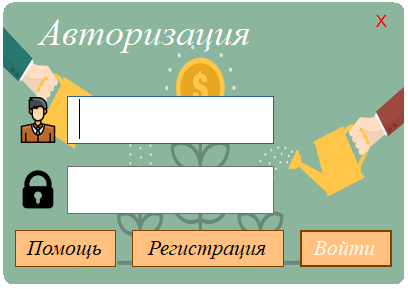


Рисунок 10 – Авторизация

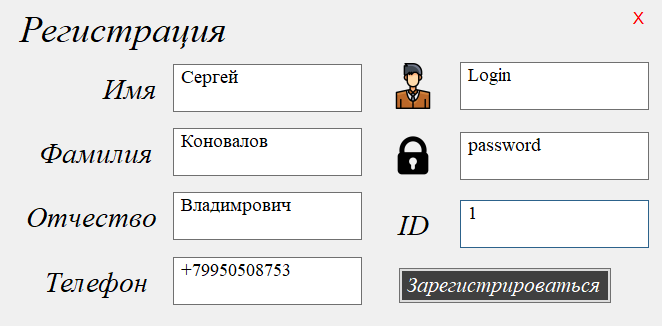


Рисунок11 – Регистрация

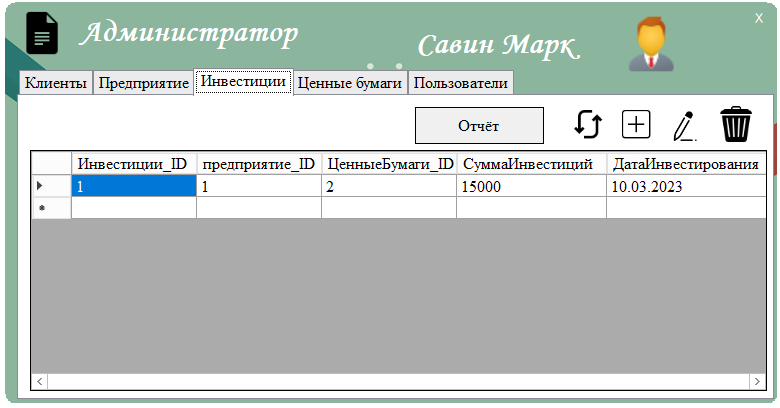


Рисунок 12 – Главная форма администратора

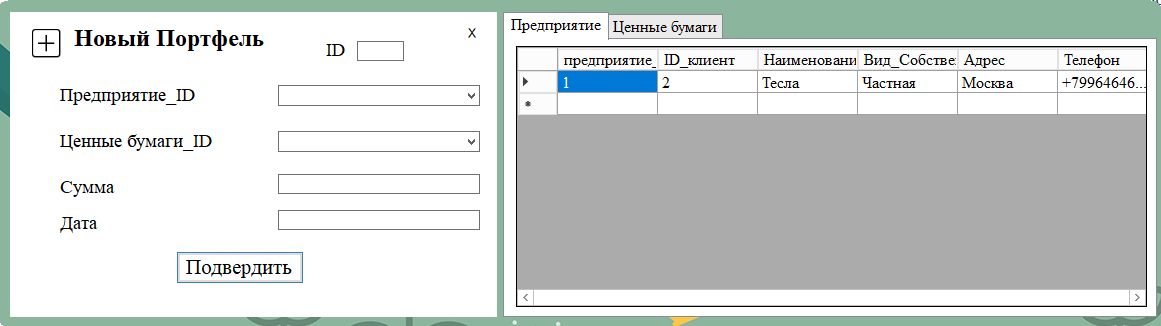


Рисунок 13 –форма добавление портфеля