**Аннотация**

Приложение "Финансовый трекер" предназначено для отслеживания и управления личными финансами. Пользователи могут регистрировать доходы и расходы, анализировать финансовые данные и получать отчеты. В данной пояснительной записке будут рассмотрены структуры приложения, применяемые технологии и внешний вид интерфейса.

## 

**Содержание**

[**Введение** 4](#_Toc169245292)

[**1. Назначение и область применения** 5](#_Toc169245293)

[**2. Постановка задачи** 6](#_Toc169245294)

[**3. Описание программы** 8](#_Toc169245295)

[**3.1 Общие сведения** 8](#_Toc169245296)

[**3.2 Функциональное назначение** 11](#_Toc169245297)

[**3.3 Описание логической структуры** 11](#_Toc169245298)

[**3.4 Входные и выходные данные:** 17](#_Toc169245299)

[**4. Тестирование приложения** 18](#_Toc169245300)

[**4.1 Объекты испытаний** 18](#_Toc169245301)

[**4.2 Цель испытаний** 18](#_Toc169245302)

[**4.3 Требования к программе** 18](#_Toc169245303)

[**4.4 Методы испытаний** 18](#_Toc169245304)

[**4.5 Тест-кейсы** 19](#_Toc169245305)

[**5. Руководство пользователя** 21](#_Toc169245306)

[**5.1 Работа с приложением** 21](#_Toc169245307)

[**5.2 Сообщение пользователю** 21](#_Toc169245308)

[**6. Мероприятия по информационной безопасности** 22](#_Toc169245309)

[**Приложение A** 24](#_Toc169245310)

[**Источники, использованные при разработке** 27](#_Toc169245311)

**Введение**

Приложение "Финансовый трекер" является интуитивно понятным инструментом для управления личными финансами. Разработка проводилось на основе следующих документов:

* Дипломное задание;
* Техническое задание.

**1 Назначение и область применения**

Программа предназначена для ведения учета финансов, анализа доходов и расходов, составления бюджетов и получения отчетов. Основное назначение - упрощение управления личными финансами для пользователей через удобный интерфейс и аналитические инструменты.

Характеристики области применения:

* Учет доходов и расходов пользователя;
* Визуализация данных на диаграммах;
* Отчетность по транзакциям пользователя;
* Система обратной связи для решения любых проблем и вопросов пользователей;
* Возможность создания напоминаний;
* Возможность создания новых категорий для транзакций;
* Поддержка нескольких групп пользователей с различными уровнями доступа;
* Обеспечение надежного хранения и защиты данных пользователей, включая шифрование и использование системы логинов и паролей.

**2 Постановка задачи**

Программа "Финансовый трекер" разработана для автоматизации процессов управления финансами, регистрации пользователей и оформления сопутствующих услуг.

Программа обладает следующими функциональными характеристиками:

* Регистрация и аутентификация пользователей: пользователи могут регистрироваться в системе, предоставляя свои персональные данные, и идентифицироваться для доступа к функционалу;
* Личные карты: пользователи могут создавать и удалять свои личные карты;
* Создание транзакций по доходам и расходам: пользователи могут созвать необходимые им транзакции по доходам и расходам, выбирая категорий, даты и личные карты;
* Отслеживание истории транзакций: пользователи могут просматривать свои личные транзакции, редактировать их и экспортировать в Excel;
* Напоминания: пользователь может создавать свои личные напоминания по датам;
* Поддержка: пользователь может создать обращение, просмотреть уже созданные и те, на которые ответили;
* Контроль и валидация данных: Программа обеспечивает валидацию входных данных для предотвращения ошибок ввода и повышения точности информации;
* Разграничение доступа: Поддержка нескольких групп пользователей с различными уровнями доступа, таких как пользователи, администраторы и менеджеры;
* Защита данных: Обеспечение надежного хранения и защиты данных пользователей с использованием шифрования, соли и системы логинов и паролей.

**3 Описание программы**

**3.1 Общие сведения**

Наименование приложения: "Финансовый трекер".

* Приложение написано на языке C# в среде разработки Visual Studio;
* БД разработана с помощью Microsoft SQL Server 19;
* Операционная система Windows (7 и выше).

Окна приложения на рисунках (1–3).

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рис.1 - Окно авторизации

Окно авторизации позволяет пользователю зайти в приложение, используя логин и пароль, при учете того, что раннее пользователь проходил регистрацию. Данный процесс необходим для обеспечения безопасности и индивидуального доступа к функционалу.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис.2 - Окно регистрации

В окне регистрации идет создание нового пользователя, для успешной регистрации необходимо ввести: фамилию, имя, отчество, логин и пароль. Так же имеется возможность обратно перейти на окно авторизации (рис.1). Данный процесс позволяет новым пользователем зарегистрироваться в системе и получить доступ к приложению.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис.3 - Главное окно пользователя

Элементы интерфейса (рис.3) – предназначены для следующего функционала:

* Транзакции: пользователю предоставляется возможность создания транзакции по доходу или расходу. Так же может переводить средства между своими счетами;
* История: пользователь может просмотреть информацию по своим транзакциям, так же присутствует возможность экспортировать данные в Excel.
* Напоминания: пользователь может создать себе напоминание на желаемую дату;
* Профиль: пользователю предоставляется возможность создания своих счетов для дальнейшей работы, также имеется возможность удаления созданных счетов;
* Поддержка: пользователь может обращаться за поддержкой через встроенную систему;
* Выход из аккаунта: пользователь может выйти из своего аккаунта, после чего он вернётся на окно авторизации (рис.1).

**3.2 Функциональное назначение**

В целях обеспечения безопасности и эффективности использования системы, установлено несколько функциональных ограничений.

* Уникальные логины пользователей: логины пользователей должны быть уникальными в системе. При регистрации нового пользователя система должна проверить, не занят ли уже выбранный логин;
* Доступ к данным: Каждый пользователь имеет доступ только к собственным данным и данным, к которым ему был предоставлен доступ.

**3.3 Описание логической структуры**

**Алгоритм приложения "Финансовый трекер".**

**Регистрация и аутентификация:**

* Пользователь вводит свои персональные данные для регистрации;
* Система проверяет корректность введенных данных;
* При успешной регистрации данные сохраняются в базе данных;
* Пользователь аутентифицируется с использованием логина и пароля;
* Администратор может зарегистрировать пользователя с любой ролью.

**Профиль:**

* После аутентификации пользователь создает свои личные карты;
* При необходимости имеет возможность удалить карту;
* Совершенные действия сохраняются в базе данных.

**Главная:**

* С данной формы пользователь совершает все дальнейшие навигационные переходы по формам, таким как – транзакции, история, напоминания, профиль и поддержка;
* Пользователь может выбрать свою личную карту для отображения информации по ней.

**Транзакции:**

* Пользователь может создать нужную ему транзакцию на выбранную карту;
* Пользователь может переводить средства между своими картами;
* Администратор может создавать и удалять категории для транзакций;
* Совершенные действия сохраняются в базе данных.

**История:**

* Пользователь может просмотреть историю по совершенным транзакциям;
* Пользователь может удалить ненужную транзакцию;
* Пользователь может экспортировать данные в Excel;
* Администратор может просматривать историю транзакций всех пользователей;
* Совершенные действия сохраняются в базе данных.

**Напоминания:**

* Пользователь может создать напоминание на определенную дату и просмотреть созданные напоминания;
* Пользователь может удалить ненужные напоминания;
* Совершенные действия сохраняются в базе данных.

**Поддержка:**

* Пользователь может создать обращение, просмотреть ранее созданные обращения и ответы на них;
* Менеджер может просматривать обращения всех пользователей и отвечать на них;
* Совершенные действия сохраняются в базе данных.

**Связи приложения с другими приложениями**

**База данных (Database):**

* Приложение взаимодействует с базой данных для хранения и получения информации о пользователях, картах, транзакций, напоминаний и поддержки;
* Все данные, введенные пользователями, администраторами и менеджерами сохраняются в базе данных;
* Пользователи, администраторы и менеджеры могут извлекать данные из базы данных, исходя из своих ограничений по ролям, для дальнейшего использования функционала приложения.

Изображение выглядит как зарисовка, диаграмма, текст, рисунок

Автоматически созданное описание

Рис. 4 - Диаграмма вариантов

Изображение выглядит как текст, снимок экрана, черно-белый, дизайн

Автоматически созданное описание

Рис. 5 – Блок-схема приложения

База данных состоит из 8 таблиц:

* Card – таблица с картами и балансом пользователей;
* Transaction – таблица с информацией про транзакциям пользователей;
* Category – таблица с категориями транзакций;
* Type – таблица с типами транзакций;
* Inquiry – таблица с информацией по обращениям пользователей и ответами менеджеров;
* User – таблица с информацией о всех пользователей;
* Role – таблица с ролями пользователей;
* Notification – таблица с информацией о напоминаниях пользователей.

Каждая таблица имеет первичные ключи с автоинкрементируемым значением.

Ниже представлены таблицы базы данных (рис.6):

Изображение выглядит как текст, диаграмма, снимок экрана, Параллельный

Автоматически созданное описание

Рис. 6 – Схема базы данных

**3.4 Входные и выходные данные:**

**Входные данные:**

**Регистрация и аутентификация пользователей:**

* При регистрации пользователь должен предоставить следующие данные: ФИО, логин и пароль;
* При аутентификации пользователь вводит свой логин и пароль.

**Создание карты:**

* Пользователь указывает название карты.

**Создание транзакций:**

* Пользователь создает необходимую ему транзакцию по доходу или расходу;
* Пользователь переводит средства между своими счетами.

**Создание напоминаний:**

* Пользователь создает напоминание.

**Поддержка:**

* Пользователь создает обращение.

**Создание категорий:**

* Администратор создает новые категории.

**Выходные данные:**

**Статистика и история:**

* Пользователю предоставляется статистика из учета его транзакций, может просмотреть отдельно каждую транзакцию.

**Поддержка:**

* Ответ менеджера пользователю на его обращение.

**Напоминания:**

* Пользователю предоставляется доступ ко всем его напоминаниям.

**4 Тестирование приложения**

**4.1 Объекты испытаний**

Объектом для испытаний является приложение "Финансовый трекер" и его функционал.

**4.2 Цель испытаний**

Испытания проводятся с целью проверки соответствия проекта требованиям, указанным в техническом задании.

**4.3 Требования к программе**

Соответствие программного продукта требованиям, указанным в техническом задании.

**4.4 Методы испытаний**

**Тестирование регистрации и аутентификации пользователей:**

Испытания проводятся для проверки правильности регистрации новых пользователей с различными вариантами данных, включая валидные и не валидные. Это включает в себя проверку корректности введенных данных (ФИО, логин и пароль), а также проверку возможности аутентификации существующих пользователей с использованием правильных и неправильных учетных данных.

**Тестирование добавление новой транзакции.**

Испытания проводятся для проверки возможности добавления новых транзакций в систему. Это включает в себя проверку корректного сохранения введения данных и их отображения.

**4.5 Тест-кейсы**

Тест – кейс №1 "Тестирование регистрации и аутентификации пользователей"

|  |  |  |  |
| --- | --- | --- | --- |
| Номер тест-кейса | 1 | | |
| Название: | Тестирование регистрации и аутентификации пользователей | | |
| Действие | | Ожидаемый результат | Результат теста:  пройден |
| Предусловие: | | | |
| Приложение запущено, сервер с базой данных доступен. | | Приложение запущено без ошибок. Сервер с базой данных доступен. | Пройден |
| Шаги теста: | | | |
| Попытка зарегистрировать нового пользователя с корректными данными (ФИО, логин и пароль). | | При регистрации нового пользователя система успешно создает новую учетную запись и перенаправляет пользователя на страницу авторизации. | Пройден |
| Попытка выполнить аутентификацию с использованием учетных данных вновь зарегистрированного пользователя. | | При аутентификации с использованием учетных данных зарегистрированного пользователя система успешно авторизует пользователя и перенаправляет его на главную страницу приложения. | Пройден |
| Постусловие: | | | |
| Закрыть приложение, выйти в OS | | Приложение закрыто | Пройден |

Тест – кейс №2 "Тестирование добавление новой транзакции"

|  |  |  |  |
| --- | --- | --- | --- |
| Номер тест-кейса | 2 | | |
| Название: | Тестирование добавление новой транзакции | | |
| Действие | | Ожидаемый результат | Результат теста:  пройден |
| Предусловие: | | | |
| Приложение запущено, сервер с базой данных доступен. | | Приложение запущено без ошибок. Сервер с базой данных доступен. | Пройден |
| Шаги теста: | | | |
| Попытка создать транзакцию с корректными данными, заполнением необходимых данных. | | При создании новой транзакции система успешно информирует пользователя об успешном добавлении новой транзакции. | Пройден |
| Попытка создать транзакцию не заполняя данные, написание некорректных данных. | | При создании с некорректными данными и не заполнением всех данных, система выдает ошибку и просит исправить недочёты. | Пройден |
| Постусловие: | | | |
| Закрыть приложение, выйти в OS | | Приложение закрыто | Пройден |

**5 Руководство пользователя**

**5.1 Работа с приложением**

**Загрузка и запуск программы:**

* Пользователь запускает приложение на своей рабочей станции;
* Приложение загружается и отображает начальное окно с авторизацией.

**Регистрация нового пользователя:**

* Пользователь переходит на регистрацию;
* Пользователь вводит свои данные: ФИО, логин и пароль;
* Приложение проверяет введенные данные на корректность и уникальность;
* При успешной регистрации пользователь получает подтверждение и переход на форму авторизации.

**Вход пользователя:**

* Пользователь вводит логин и пароль и нажимает на кнопку вход;
* Приложение проверяет правильность введенных данных;
* При успешной аутентификации пользователь перенаправляется на основное окно приложения.

**5.2 Сообщение пользователю**

**Регистрация с существующем логином:** "Данный логин уже занят, выберите другой".

**Неверный ввод логина или пароля:** "Неверный логин или пароль".

**6 Мероприятия по информационной безопасности**

**6.1 Возможные угрозы информационной безопасности:**

Для обеспечения безопасности информации в приложении "Финансовый трекер" необходимо учитывать следующие возможные угрозы:

* Кража учетных данных
* Вредоносные программы и вирусы
* Несанкционированный доступ к данным
* Подбор чужого пароля

**6.2 Предотвращение угроз информационной безопасности:**

**Шифрование пароля:**

Для повышения безопасности паролей пользователей реализуется механизм шифрования и использование соли. При регистрации нового пользователя пароль хэшируется с использованием криптографических алгоритмов, таких как SHA-256, перед сохранением в базу данных. Это обеспечивает защиту от несанкционированного доступа к хранимым паролям (рис.7).

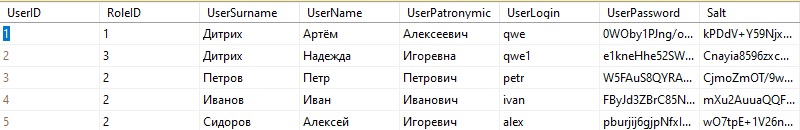


Рис.7 – Пример шифрования из базы данных

**Скрытие ввода пароля:**

При вводе пароля пользователем символы пароля отображаются звездами, чтобы предотвратить его просмотр со стороны посторонних лиц (рис.8).

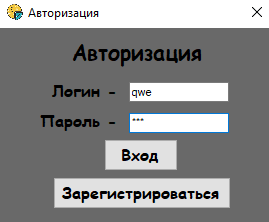


Рис. 8 – Скрытый пароль при авторизации

**Разграничение прав**

Пользователям назначаются минимально необходимые права для выполнения операций в системе. Это уменьшает риски несанкционированного доступа и снижает возможные негативные последствия в случае компрометации аккаунта (рис. 9).

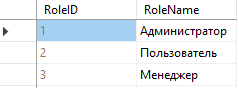


Рис.9 – Разграничение прав

**Резервное копирование базы данных:**

Для обеспечения сохранности и доступности критически важной информации, менеджеры компании могут на регулярной основе выполнять резервное копирование базы данных. Это мероприятие направлено на защиту данных от потерь, вызванных различными угрозами, включая технические сбои, ошибки пользователей, вредоносные атаки и прочие форс-мажорные обстоятельства (рис.10-11).

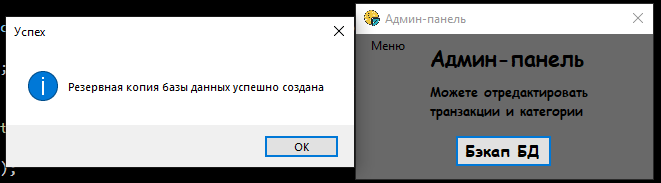


Рис.10 – Бэкап базы данных в приложении

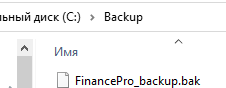


Рис.11 – Сохраненный бэкап на локальной станции

Реализация указанных мероприятий позволит повысить уровень информационной безопасности и обеспечить защиту приложения.

**Заключение**

Программный продукт "Финансовый трекер" полностью соответствует всем требованиям, изложенным в техническом задании. В ходе разработки были реализованы основные функции для управления личными финансами, включая регистрацию и аутентификацию пользователей, создание и управление транзакциями, напоминаниями, а также поддержку пользователей.

Приложение разработано на языке программирования C# с использованием среды разработки Visual Studio, а база данных создана с помощью Microsoft SQL Server.

Важной частью системы является обеспечение надежного хранения и защиты данных пользователей, включая шифрование паролей и строгую систему контроля доступа. Приложение прошло тестирование, что подтвердило его работоспособность и соответствие заявленным требованиям.

"Финансовый трекер" предоставляет пользователям интуитивно понятный интерфейс и мощные аналитические инструменты для управления личными финансами. В будущем планируется расширение функционала, включая интеграцию новых функций для улучшения пользовательского опыта и повышения безопасности данных.

Таким образом, поставленные при проектировании задачи выполнены, и приложение готово к использованию для упрощения и автоматизации управления личными финансами.

**Приложение A**

**Текст программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Security.Cryptography;

namespace FinancePro.Forms

{

public partial class GlavAutoruz : Form

{

private const string connectionString = "Data Source=DESKTOP-2JK9QLP;Initial Catalog=FinancePro;User ID=qwe;Password=qwe;Encrypt=False;TrustServerCertificate=True";

public GlavAutoruz()

{

InitializeComponent();

}

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

{

string login = txtLogin.Text;

string password = txtPassword.Text;

// Проверка правильности введенных данных

if (!IsValidUser(login, password))

{

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

return;

}

// Получение роли и ID пользователя

int roleID = GetUserRole(login);

int userID = GetUserID(login);

// В зависимости от роли, открываем соответствующую форму

if (roleID == 1)

{

AdminPanel adminPanel = new AdminPanel(login, roleID, userID.ToString());

adminPanel.Show();

}

else if (roleID == 2)

{

Ocnovnay ocnovnay = new Ocnovnay(login, roleID, userID.ToString());

ocnovnay.Show();

}

else if (roleID == 3)

{

ManagerPanel managerPanel = new ManagerPanel(login, roleID, userID.ToString());

managerPanel.Show();

}

else

{

MessageBox.Show("Неизвестная роль пользователя.");

return;

}

this.Hide();

}

private bool IsValidUser(string login, string password)

{

bool result = false;

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT UserPassword, Salt FROM [User] WHERE UserLogin = @Login";

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

{

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

connection.Open();

using (SqlDataReader reader = command.ExecuteReader())

{

if (reader.Read())

{

string storedPassword = reader.GetString(0);

string salt = reader.GetString(1);

string hashedPassword = HashPassword(password, salt);

if (storedPassword == hashedPassword)

{

result = true;

}

}

}

}

}

return result;

}

private int GetUserRole(string login)

{

int roleID = -1;

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT RoleID FROM [User] WHERE UserLogin = @Login";

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

{

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

connection.Open();

object result = command.ExecuteScalar();

if (result != null)

{

roleID = Convert.ToInt32(result);

}

}

}

return roleID;

}

private int GetUserID(string login)

{

int userID = -1;

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT UserID FROM [User] WHERE UserLogin = @Login";

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

{

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

connection.Open();

object result = command.ExecuteScalar();

if (result != null)

{

userID = Convert.ToInt32(result);

}

}

}

return userID;

}

private string HashPassword(string password, string salt)

{

using (var sha256 = SHA256.Create())

{

byte[] saltedPassword = Encoding.UTF8.GetBytes(password + salt);

byte[] hash = sha256.ComputeHash(saltedPassword);

return Convert.ToBase64String(hash);

}

}

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

{

Registration registration = new Registration();

registration.Show();

this.Hide();

}

}

}

**Источники, использованные при разработке**

* ГОСТ Р ИСО\_МЭК 25051-2017 Требования к качеству готового к использованию программного продукта (RUSP) и инструкции по тестированию.
* ГОСТ Р 2.105–2019 Общие требования к текстовым документам.
* ГОСТ Р 59795–2021 Комплекс стандартов на автоматизированные системы.
* ЕСПД Единая система программной документации.

**Internet – ресурсы**

* Руководство по C# - <https://metanit.com/sharp/>
* Руководство по программированию Windows Forms - <https://metanit.com/sharp/windowsforms/>
* MySQL [Электронный ресурс] - <https://www.mysql.com/>
* GitHub [https://github.com/] Онлайн-платформа для хостинга и совместной разработки проектов. Позволяет отслеживать изменения в коде.
* Microsoft Documentation [https://docs.microsoft.com/] Официальная документация от Microsoft. Использовалась для изучения технических деталей, API и рекомендаций по разработке на платформе .NET.