**Техническое задание для программы ведения расходов и доходов.**

**1. Введение**

Приложение для управления личными финансами разработано с использованием архитектуры «клиент-сервер», где клиентская часть реализована на C# с использованием WinForms для графического пользовательского интерфейса. Приложение взаимодействует с базой данных SQLite для хранения и управления финансовыми данными. Основные компоненты приложения включают:

**2. Общие сведения**

Название проекта: Приложение введения расходов и доходов

Заказчик: [Имя заказчика или название компании]

Исполнитель: Юдина Дарья Алексеевна

Сроки выполнения: 35 часов

Бюджет: Согласно технологии TM, оплата 2.000 рублей/час

**3. Описание функциональности**

**3.1 Основные функции**

**Добавление доходов**

Введите сумму дохода в поле amountTextBox.

Выберите категорию из списка categoryComboBox.

Установите дату транзакции с помощью элемента управления dateTimePicker.

Нажмите кнопку «Добавить доход» (addIncomeButton).

После успешного добавления вы увидите сообщение об успешном добавлении, а таблица транзакций обновится.

**Добавление расходов**

Введите сумму расхода в поле amountTextBox.

Выберите категорию из списка categoryComboBox.

Установите дату транзакции с помощью элемента управления dateTimePicker.

Нажмите кнопку «Добавить расход» (addExpenseButton).

После успешного добавления вы увидите сообщение об успешном добавлении, а таблица транзакций обновится.

Как использовать функции бюджетирования, анализа расходов и напоминаний

**Бюджетирование**

Выберите категорию из списка бюджета (budgetCategoryComboBox).

Введите сумму бюджета в поле budgetAmountTextBox.

Нажмите кнопку «Сохранить бюджет» (saveBudgetButton).

После успешного сохранения вы получите уведомление о том, что бюджет установлен.

**Анализ расходов**

Нажмите кнопку «Сгенерировать отчет» (generateReportButton).

На графике (cartesianChart1) будут отображаться расходы по категориям, что позволит вам визуально оценить, на что уходят ваши деньги.

**Напоминания**

Введите описание напоминания в поле reminderDescriptionTextBox.

Выберите дату напоминания с помощью элемента управления reminderDatePicker.

Нажмите кнопку «Добавить напоминание» (addReminderButton).

Чтобы проверить все напоминания, нажмите кнопку «Проверить напоминания» (checkRemindersButton). Если есть просроченные напоминания, вы получите уведомление с описанием.

**Импорт и экспорт данных**

Импорт данных из CSV

Нажмите кнопку «Импортировать из CSV» (importCsvButton).

Выберите файл CSV для импорта.

Данные будут загружены в приложение, и вы получите уведомление об успешном импорте.

**Экспорт данных в CSV**

Нажмите кнопку «Экспорт в CSV» (exportCsvButton).

Выберите место для сохранения файла.

Данные будут экспортированы, и вы получите уведомление об успешном экспорте.

**4. Технические требования**

**4.1 Платформа**

C# и WinForms: основной язык программирования и фреймворк для создания графического пользовательского интерфейса.

SQLite: легковесная база данных, используемая для хранения финансовых данных в приложении.

LiveCharts: библиотека для визуализации данных в виде графиков и диаграмм, используемая для анализа расходов по категориям.

**4.2 База данных**

База данных SQLite: хранит данные о транзакциях, категориях, бюджетах и напоминаниях.

Элементы управления: текстовые поля, кнопки, выпадающие списки и графики для взаимодействия с пользователем.

**Объяснение структуры базы данных SQLite**

База данных SQLite состоит из трех основных таблиц:

**Категории: Хранит категории расходов и доходов.**

ID: Уникальный идентификатор категории.

Название: название категории (например, «Еда», «Транспорт»).

sql

CREATE TABLE Categories (

Id INTEGER PRIMARY KEY AUTOINCREMENT,

Name TEXT NOT NULL UNIQUE

);

**Транзакции: Хранит записи о финансовых транзакциях.**

ID: Уникальный идентификатор транзакции.

Количество: Сумма транзакции.

CategoryId: Идентификатор категории, к которой относится транзакция (внешний ключ).

Дата транзакции: Дата транзакции.

Тип: Тип транзакции (доход или расход).

sql

CREATE TABLE Transactions (

Id INTEGER PRIMARY KEY AUTOINCREMENT,

Amount REAL NOT NULL,

CategoryId INTEGER NOT NULL,

TransactionDate DATETIME DEFAULT CURRENT\_TIMESTAMP,

Type TEXT CHECK(Type IN ('Income', 'Expense')),

FOREIGN KEY(CategoryId) REFERENCES Categories(Id)

);

**Бюджеты: Хранит информацию о бюджетах по категориям.**

ID: Уникальный идентификатор бюджета.

Идентификатор категории: Идентификатор категории (внешний ключ).

Количество: Сумма бюджета.

Месяц: Месяц бюджета.

Год: Год бюджета.

sql

CREATE TABLE Budgets (

Id INTEGER PRIMARY KEY AUTOINCREMENT,

CategoryId INTEGER NOT NULL,

Amount REAL NOT NULL,

Month INTEGER NOT NULL,

Year INTEGER NOT NULL,

FOREIGN KEY(CategoryId) REFERENCES Categories(Id)

);

**Напоминания: Хранит напоминания о предстоящих платежах.**

ID: Уникальный идентификатор напоминания.

Описание: Описание напоминания.

Двойственный: Дата выполнения напоминания.

sql

CREATE TABLE Reminders (

Id INTEGER PRIMARY KEY AUTOINCREMENT,

Description TEXT,

DueDate TEXT

);

**5. Пользовательский интерфейс**

**5.1 Дизайн**

Приложение имеет интуитивно понятный интерфейс с основными компонентами:

Поля ввода для суммы, категорий и даты транзакций.

Кнопки для добавления доходов и расходов, генерации отчетов и управления напоминаниями.

График анализа расходов по категориям.

Таблица для отображения всех транзакций.

Инструкции по добавлению доходов и расходов

**5.2 Прототипы**

Разработка прототипов экранов приложения для согласования с заказчиком.

**6. Тестирование**

Проведение функционального тестирования всех компонентов системы.

Проведение пользовательского тестирования для проверки удобства интерфейса.

**7. Документация**

Подготовка пользовательской документации и руководства по эксплуатации.

**8. Поддержка и обслуживание**

Обеспечение технической поддержки в течение 3-х месяцев после запуска проекта.

Регулярные обновления системы в соответствии с отзывами пользователей.

**9. Заключение**

Данное техническое задание является основой для разработки многофункционального калькулятора и может быть изменено по мере необходимости в процессе работы над проектом. Все изменения должны быть согласованы с заказчиком.