Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Проектирование и разработка информационных систем (ПиРИС)

Практическая работа №2

Структурный подход к программированию.

Стадия «Эскизный проект»

Выполнил

студент: гр. 751003 Стубеда В.Д.

Проверил: Грибович А.А.

Минск 2021

# 1 АНАЛИЗ ФУНКЦИОНАЛЬНЫХ И ЭКСПЛУАТАЦИОННЫХ ТРЕБОВАНИЙ

## **1.1 Анализ функциональных требований**

Разрабатываемая информационная система должна предоставлять пользователю следующий функционал:

а) регистрация данных нового пользователя системы;

б) изменение данных пользователя системы;

в) удаление данных пользователя системы;

г) просмотр данных всех зарегистрированных пользователей системы;

д) оформление депозитного договора;

е) оформление кредитного договора.

## **1.2 Анализ эксплуатационных требований**

Разрабатываемая информационная система должна соответствовать следующим эксплуатационным требованиям:

а) пользователь системы должен обладать базовой компьютерной грамотностью;

б) количество пользователей, одновременно использующих систему, должно быть в пределе от 1 до 10000;

в) на сервере хранится резервная копия базы данных.

Для правильной работы информационной системы необходимо минимум два пользователя – системный администратор и клиент банка.

Системный администратор должен поддерживать работоспособность технических средств, устранять неисправности системы, создавать резервные копии базы данных.

Пользователь системы должен обладать достаточным уровнем знаний в предметной области, владеть базовыми знаниями и навыками работы с программным комплексом, использующемся в системе.

г) наличие браузера Chrome 88.0.

# 2 ОСНОВНЫЕ ТЕХНИЧЕСКИЕ РЕШЕНИЯ

**2.1 Инструмент разработки Visual Studio Code**

Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя [отладчик](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D1%87%D0%B8%D0%BA), инструменты для работы с [Git](https://ru.wikipedia.org/wiki/Git" \o "Git), [подсветку синтаксиса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D0%B0), [IntelliSense](https://ru.wikipedia.org/wiki/IntelliSense" \o "IntelliSense) и средства для [рефакторинга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3" \o "Рефакторинг). Имеет широкие возможности для [кастомизации](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D1%82%D0%BE%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F" \o "Кастомизация): пользовательские темы, [сочетания клавиш](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%87%D0%B5%D1%82%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BA%D0%BB%D0%B0%D0%B2%D0%B8%D1%88) и [файлы конфигурации](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB_%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%B8). Распространяется бесплатно, разрабатывается как программное обеспечение с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), но готовые сборки распространяются под [проприетарной лицензией](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5).

**2.2 Платформа Node.js**

Платформа Node.js – это кроссплатформенная среда выполнения для JavaScript, которая работает на серверах. С момента выпуска этой платформы в 2009 году она стала чрезвычайно популярной и в наши дни играет весьма важную роль в области веб-разработки.

Платформа Node.js построена на базе JavaScript движка V8 от Google, который используется в браузере Google Chrome. Данная платформа, в основном, используется для создания веб-серверов, однако сфера её применения этим не ограничивается.

Выбор серверного JavaScript для стороны сервера обеспечивает проекту ряд преимуществ:

а) рост эффективности разработки благодаря использованию одного языка как для стороны клиента, так и для стороны сервера и возможности повторного использования кода;

б) возможность использовать самого большого пакетного менеджера npm;

в) быстрая обработка большого количества запросов одновременно, что обеспечивает быстродействие приложения.

**2.3 Фреймворк React**

React — JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. React может использоваться для разработки одностраничных и мобильных приложений. React может использоваться для разработки [одностраничных](https://ru.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) и мобильных приложений. Его цель — предоставить высокую скорость, простоту и масштабируемость.

**2.4 Система управления базами данных MySQL**

Система управления базами данныхMySQL – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle. MySQL является решением для малых и средних приложений. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Для работы с СУБД используется MySQL Workbench – инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL.

# 3 АРХИТЕКТУРА ПРОГРАММНОГО ПРОДУКТА

## **3.1 Диаграммы, описывающие архитектуру системы**

## 

Общая архитектура программного продукта представлено в виде DFD-модели на рисунке 3.1.



Рисунок 3.1 – Диаграмма потоков данных в системе DFD

Укрупненная схема регистрации клиента банка представлена ниже на рисунке 3.2.

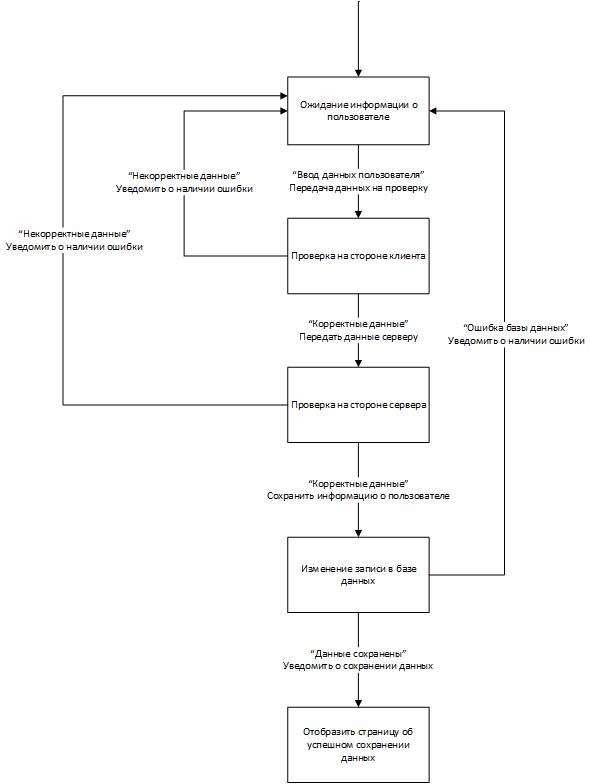


Рисунок 3.2 – STD Диаграмма потоков данных процесса регистрации клиента

Схема процесса обновления данных клиента представлена ниже на рисунке 3.3.



Рисунок 3.3 – DFD Диаграмма потоков данных процесса обновления данных клиента

## **3.2 Объекты и их атрибуты**

Установлены следующие входные документы и сообщения в терминах объектов предметной области:

- клиент;

- основной счет клиента;

- депозитный счет клиента;

- кредитный счет клиента.

Объект «Клиент» должен обладать следующими атрибутами:

- идентификатор клиента;

- фамилия;

- имя;

- отчество;

- дата рождения;

- серия паспорта;

- номер паспорта;

- орган, выдавший паспорт;

- дата выдачи паспорта;

- место рождения;

- город фактического проживания;

- адрес фактического проживания;

- мобильный телефон;

- домашний телефон;

- адрес прописки;

- семейное положение;

- гражданство;

- инвалидность;

- ежемесячный доход.

Объект «Основной счет клиента» должен обладать следующими атрибутами:

- идентификатор основного счета клиента;

- идентификатор клиента;

- дебет;

- кредит;

- сальдо;

- валюта;

- дата открытия счета.

Объект «Депозитный счет клиента» должен обладать следующими атрибутами:

- идентификатор депозитного счета;

- идентификатор клиента;

- вид депозитного счета;

- сумма депозита;

- дебет;

- кредит;

- сальдо;

- валюта;

- дата открытия счета;

- дата окончания действия счета.

Объект «Кредитный счет клиента» должен обладать следующими атрибутами:

- идентификатор кредитного счета;

- идентификатор клиента;

- дебет;

- кредит;

- сальдо;

- валюта;

- вид кредитного счета;

- дата открытия счета;

- дата окончания действия счета.

# 4 СЛОВАРЬ ТЕРМИНОВ

Клиент банка – юридическое или физическое лицо, пользующееся услугами банка.

Форма регистрации/редактирования – HTML-форма на странице сайта, в которой есть поля для ввода информации о клиенте.

Проверка – проведение верификации указанных данных клиента на соответствие формату (телефон моб., дом., серия, номер, идентификационный номер паспорта, даты).

Пользователь – физическое или юридическое лицо, которое использует действующую систему для выполнения конкретной функции.

Транзакция – сделка с использованием банковского счета. Различают онлайн-транзакции, выполняющиеся в режиме реального времени между всеми заинтересованными сторонами, и офлайн-транзакции.

Дебет – левая сторона бухгалтерского счета. По активным и активно-пассивным счетам: увеличение дебета означает увеличение стоимости имущества или имущественных прав организации. По пассивным счетам: увеличение дебета означает уменьшение собственных средств организации (источников).

Кредит – правая сторона бухгалтерского счета. По активным и активно-пассивным счетам: увеличение кредита означает уменьшение стоимости имущества или имущественных прав организации. По пассивным счетам: увеличение кредита означает увеличение собственных средств организации (источников).