Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа по предмету «Проектирование информационных систем» на тему «Моделирование процессов с использованием методологии *IDEF3*»

Студент: Трубач Д. С.

ФИТ 4 курс 5 группа

Преподаватель: Якубенко К. Д.

Минск 2024

# 1 Постановка задач

Задачей проекта является моделирование процессов информационной системы на основе методологии IDEF3, разработка моделей бизнес-процессов и их декомпозиции. Модели должны отражать все функциональные требования, заявленные к информационной системе на основании предыдущих лабораторных работ (например, введение данных в форму, подтверждение количества сданных отходов пользователям).

Web-приложение должно быть выполнено с использованием асинхронного программирования, взаимодействовать с базой данных, реализовано под разными платформами. Web-приложение должно представлять собой web-приложение с асинхронным UI с использованием фреймворка React. Отображение, бизнес-логика и хранилище данных должны быть максимально независимы друг от друга для возможности расширения. Диаграмму вариантов использования разработать на основе UML, также необходимо разработать логическую схему базы данных и структурную схему приложения. Язык разработки проекта JavaScript, платформа «NodeJS». Развертывание конечного приложения для последующего использования должно осуществляться с использованием Docker.

Функционально web-приложение должно:

* поддерживать роли гостя, пользователя, администратора;
* обеспечить возможность регистрации, авторизации;
* обеспечивать возможностью просмотра пунктов вторсырья;
* обеспечивать возможностью обмена своих накопленных баллов на скидки в различных сервисах;
* обеспечивать возможностями создания, изменения и удаления своих статей;
* обеспечивать возможностью оценки статей;
* обеспечивать возможностью получения информации сколько новой продукции будет сделано из сданных отходов.
* обеспечивать возможностью добавления новых пунктов приема вторсырья, изменение и удаление существующих;
* обеспечивать возможностью изменения и удаления видов вторсырья;
* обеспечивать возможностью добавления скидок, которые пользователи могут использовать за накопленные баллы;
* обеспечивать возможностью изменения и удаление статей о раздельном сборе отходов.

Основные задачи приложения:

* разработать удобный интерфейс, который позволит пользователям легко ориентироваться и находить информацию о сортировке отходов;
* предоставить функционал для ознакомления с пошаговыми инструкциями по раздельному сбору мусора;
* создать систему накопления баллов за сортировку, с возможностью их обмена на товары, произведенные из переработанных материалов.

# 2 Описание программных средств

Для выполнения лабораторной работы использовался ряд современных программных средств, каждое из которых играло ключевую роль в разработке и обеспечении функциональности системы.

Node.js (версия 18.17.1) был выбран в качестве основной платформы для разработки серверной части приложения. Это кроссплатформенное решение, поддерживающее работу на таких операционных системах, как Windows, Linux и macOS, что делает его крайне удобным для различных сред разработки и развертывания. Node.js предоставляет высокопроизводительную асинхронную модель обработки событий, которая позволяет эффективно управлять параллельными запросами пользователей, минимизируя задержки и обеспечивая плавную работу системы даже при высоких нагрузках. Особое внимание уделялось масштабируемости серверного решения, поскольку ожидается рост количества пользователей системы. Благодаря архитектуре Node.js, разработчики смогли реализовать систему, которая легко адаптируется к изменениям нагрузки, а также эффективно обрабатывает множество запросов на чтение и запись данных.

Важным преимуществом Node.js является его экосистема npm (Node Package Manager), которая предоставляет доступ к тысячам библиотек и инструментов, значительно ускоряя процесс разработки. Для данного проекта было использовано множество библиотек, что позволило минимизировать время на написание дополнительного кода, сосредоточив внимание на основной логике системы — обработке запросов, управлении авторизацией и взаимодействии с базой данных. Программный код на JavaScript и Node.js был оптимизирован с целью обеспечения высокой производительности и устойчивости системы в условиях масштабирования.

Данные о технологиях:

* Разработчик: OpenJS Foundation.
* Адрес загрузки: nodejs.org.
* Использовался для разработки серверной части системы, обеспечивая обработку запросов пользователей, управление авторизацией и взаимодействие с базой данных.
* Доступность: кроссплатформенная (Windows, Linux, macOS).

React.js был выбран для создания клиентской части приложения, что позволяет пользователям взаимодействовать с системой через удобный веб-интерфейс. React.js — это библиотека для разработки динамических пользовательских интерфейсов, которая оптимизирует работу с компонентами и виртуальным DOM, что позволяет интерфейсу быстро реагировать на изменения данных без необходимости перезагрузки страницы. Благодаря гибкости React, разработчикам удалось легко интегрировать асинхронные запросы, обеспечивая плавное взаимодействие пользователя с сервером. Данные о технологии:

* Разработчик: Facebook.
* Адрес загрузки: reactjs.org.
* Использовался для создания клиентской части приложения, асинхронного пользовательского интерфейса и взаимодействия с сервером.
* Доступность: кроссплатформенная (поддержка браузеров).

MySQL (версия 8.0.28) выполняет роль основной базы данных для хранения данных пользователей, статей, пунктов приема вторсырья и статистической информации. Этот реляционный менеджер баз данных был выбран за его высокую производительность, надежность и расширяемость. MySQL поддерживает мощные инструменты для обработки больших объемов данных, включая индексацию, параллельную обработку запросов и надежные транзакции. Это позволяет системе обрабатывать множество запросов на чтение и запись данных одновременно, обеспечивая стабильную работу даже при увеличении числа пользователей. Также стоит отметить, что MySQL кроссплатформенен и поддерживает различные операционные системы, что делает его гибким в использовании.

Наконец, Docker (версия 20.10) был использован для контейнеризации всех компонентов системы, что позволяет разворачивать приложение на различных платформах с минимальными усилиями. Docker обеспечивает изоляцию и независимость окружений, что упрощает процесс разработки, тестирования и развертывания приложения. Каждый компонент системы был упакован в отдельные контейнеры, что позволяет легко управлять зависимостями и масштабировать систему. Docker также предоставляет возможность горизонтального и вертикального масштабирования приложения, что особенно важно для поддержки высоких нагрузок в продакшене.

Таким образом, использование этих технологий обеспечило эффективную и масштабируемую разработку системы, соответствующую всем требованиям по функциональности и безопасности. Каждая технология сыграла свою уникальную роль в создании целостной архитектуры, которая способна адаптироваться к изменениям нагрузки и масштабироваться без ущерба для производительности. Благодаря контейнеризации с использованием Docker, разработчики смогли обеспечить гибкость развертывания системы, а также ее быструю настройку в различных окружениях. Кроссплатформенные возможности Node.js и MySQL, а также гибкость React.js позволили обеспечить взаимодействие серверной и клиентской частей, при этом сохраняя высокий уровень производительности и отказоустойчивости.

# 3 Описание практического задания

Задачей является построение модели бизнес-процессов на основе методологии IDEF3. Необходимо:

* Построить IDEF0-модель на основе предыдущих этапов разработки.
* Разработать IDEF3-модель, представляющую функциональные блоки системы (например, обработку сообщений, авторизацию пользователей, управление чатами).
* Описать связи между блоками, потоки данных, хранилища и внешние объекты, которые участвуют в процессах.

Пример описания IDEF3 процесса:

* **Работа 1**: заполнение формы. Входные данные: заполненные поля формы (тип сдаваемых отходов, количество сданных отходов в кг., фото чека). Процесс: проверка введенных данных через валидаторы. Выходные данные: сообщение о успешном заполнении формы.
* **Работа 2**: подтверждение введенных данных. Входные данные: данные, пришедшие от пользователя по средствам формы. Процесс: проверка достоверности представленного чека пользователя. Выходные данные: статус отправленной ранее формы.

Draw.io – это бесплатное онлайн-приложение для создания диаграмм и схем. Оно позволяет пользователям создавать диаграммы благодаря широкому набору инструментов и функций.

Работа заполнение формы представлена на диаграмме, представленной на рисунке 3.1.

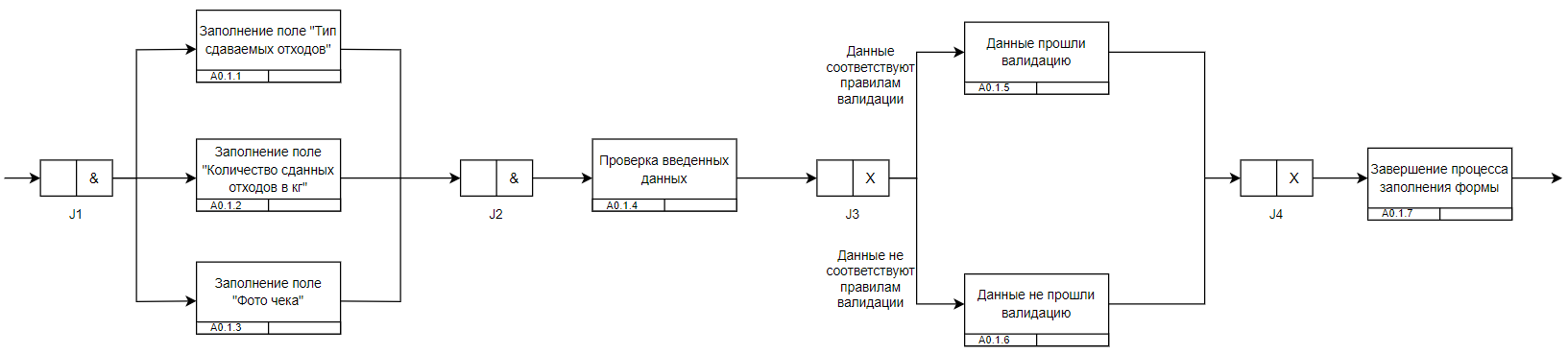


Рисунок 3.1 – Диаграмма процесса заполнение формы

Данная диаграмма отображает процесс заполнение формы о сдаче вторсырья в информационной системе. Входные потоки содержат поля формы (количество сданных отходов, вид сдаваемых отходов и чек подтверждающий факт сдачи отходов. Потоки управления содержат поток валидации формы. Выходные потоки содержат информацию, которая прошла через валидацию. В качестве механизмов выступают пользователь веб-приложения и администратор.

Описание основных элементов:

* Заполнение поле формы – начальный этап, где пользователь вводит свои данные.
* Проверка введенных данных – система проверяет данные, чтобы определить, корректны ли они.
* В зависимости от результатов, система может вывести одно из следующих сообщений: данные прошли валидацию, данные не прошли валидацию.

Эта диаграмма подчеркивает логику работы системы, проверяя введенные данные для дальнейшего взаимодействия с приложением.

Диаграмма процесса отправки подтверждения введенных данных представлена на рисунке 3.2.

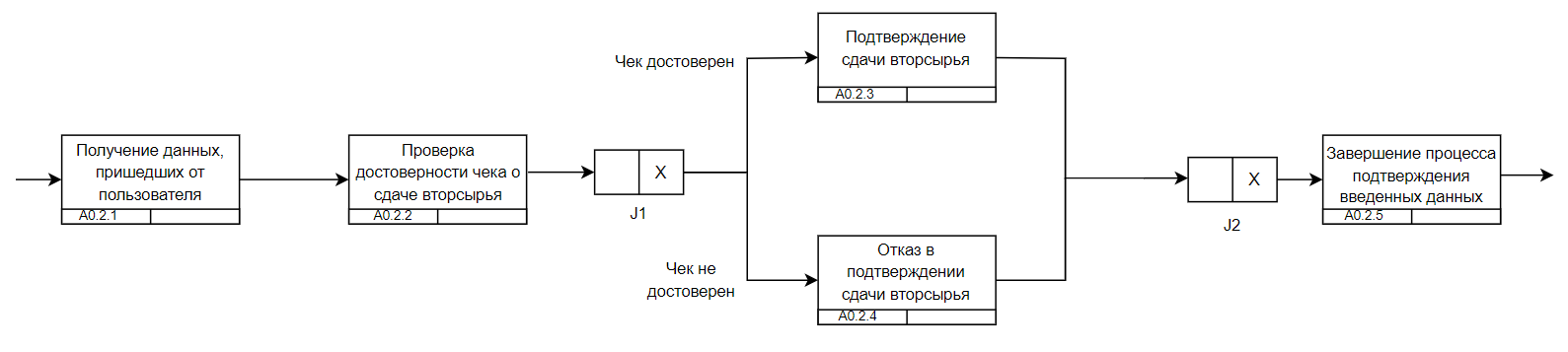


Рисунок 3.2 – Диаграмма процесса подтверждения введенных данных

На данной диаграмме описывается процесс подтверждения введенных данных. Он состоит из пяти функциональных блоков:

* Первый блок – получение данных от пользователя. Отвечает за получение данных из формы, когда пользователь сдал вторсырье.
* Второй блок – поиск проверка достоверности чека о сдаче вторсырья. На данном этапе осуществляется проверка действительного акта сдачи вторсырья.
* Третий блок – подтверждение сдачи вторсырья.
* Четвертый блок – отказ в подтверждении сдачи вторсырья.
* Пятый блок –завершение процесса подтверждения введенных данных.

Диаграммы процессов продемонстрировали, как происходит обработка данных пользователя через различные функциональные блоки системы. Это позволило выявить основные потоки информации, взаимодействие с внешними объектами и механизмами, а также определить точки контроля данных, что в будущем облегчит поддержку и развитие системы.

# Ответы на теоретические вопросы

1. Дайте описание термину «процесс»?

Процесс ­– это последовательность действий или операций, направленных на достижение определенной цели или результата.

2. Какие основные методы входят в IDEF3?

[**Основные методы IDEF3**:](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin10,udsdlpconsent,udsfrontload,cspgrd,&shellsig=764419830fc0a1213ff1bc0bfd4f2796cbf4f76a&setlang=ru&darkschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C0%7C0f10b50b-0d92-4ac0-88af-176327a5ff43)

* **Старшая связь (Precedence Link)**: показывает, что одна работа должна завершиться до начала другой.
* [**Отношения (Relational Link)**: используются для отображения связей между работами и объектами](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin10,udsdlpconsent,udsfrontload,cspgrd,&shellsig=764419830fc0a1213ff1bc0bfd4f2796cbf4f76a&setlang=ru&darkschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C1%7C0f10b50b-0d92-4ac0-88af-176327a5ff43).
* **Потоки объектов (Object Flow)**: описывают использование объекта в нескольких работах.

3. Какие элементы являются центральными компонентами модели IDEF3?

**Центральные компоненты модели IDEF3**:

* [**Единицы работы (Unit of Work, UOW)**: основные элементы, представляющие действия или процессы](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin10,udsdlpconsent,udsfrontload,cspgrd,&shellsig=764419830fc0a1213ff1bc0bfd4f2796cbf4f76a&setlang=ru&darkschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C3%7C0f10b50b-0d92-4ac0-88af-176327a5ff43).
* [**Связи (Links)**: показывают взаимоотношения между единицами работы](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin10,udsdlpconsent,udsfrontload,cspgrd,&shellsig=764419830fc0a1213ff1bc0bfd4f2796cbf4f76a&setlang=ru&darkschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C4%7C0f10b50b-0d92-4ac0-88af-176327a5ff43).
* [**Перекрестки (Junctions)**: используются для отображения логики взаимодействия стрелок при слиянии и разветвлении](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin10,udsdlpconsent,udsfrontload,cspgrd,&shellsig=764419830fc0a1213ff1bc0bfd4f2796cbf4f76a&setlang=ru&darkschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C5%7C0f10b50b-0d92-4ac0-88af-176327a5ff43).

4. В чём смысл использования перекрёстков в IDEF3?

Перекрестки отображают логику взаимодействия стрелок при слиянии и разветвлении, помогая визуализировать последовательность и условия выполнения процессов.

5. В чём отличия IDEF0 и IDEF3? Когда и как их целесообразно использовать?

[**Отличия IDEF0 и IDEF3**:](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin10,udsdlpconsent,udsfrontload,cspgrd,&shellsig=764419830fc0a1213ff1bc0bfd4f2796cbf4f76a&setlang=ru&darkschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C2%7C0f10b50b-0d92-4ac0-88af-176327a5ff43)

**IDEF0**: используется для функционального моделирования, фокусируется на функциях и их взаимодействиях.

**IDEF3**: ориентирован на моделирование процессов и последовательностей действий.

**Применение**: IDEF0 целесообразно использовать для описания функций системы, а IDEF3 ­– для детального описания бизнес-процессов и их последовательностей.