Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа по предмету «Проектирование информационных систем» на тему «Построение функциональной модели IDEF0»

Студент: Коршун Н.И.

ФИТ 4 курс 5 группа

Преподаватель: Якубенко К. Д.

Минск 2024

# 1 Постановка задач

Web-приложение должно быть выполнено с использованием асинхронного программирования, взаимодействовать с базой данных и быть реализовано под разными платформами. Оно должно представлять собой web-приложение с асинхронным пользовательским интерфейсом (UI), разработанным на базе фреймворка Next.js. Отображение, бизнес-логика и хранилище данных должны быть максимально независимы друг от друга, что позволит расширять функционал в будущем. Для проектирования диаграмм вариантов использования следует использовать UML, а также необходимо разработать логическую схему базы данных и структурную схему приложения. Язык разработки проекта — JavaScript (Node.js, Next.js). Развёртывание конечного приложения должно осуществляться с использованием Docker. Web-приложение должно быть логически завершенным и интуитивно понятным для пользователя, с удобным управлением. Код проекта должен содержать комментарии.

Функционально web-приложение должно:

* обеспечивать возможность просмотра информации о различных моделях автомобилей (характеристики, фотографии);
* предоставлять функционал просмотра видеообзоров на автомобили;
* позволять пользователю сравнивать характеристики нескольких автомобилей;
* обеспечивать регистрацию и авторизацию пользователей с возможностью восстановления пароля через электронную почту
* поддерживать роли гостя, пользователя, администратора;
* для гостей доступен просмотр автомобилей и видеообзоров;
* для зарегистрированных пользователей доступен функционал сохранения избранных автомобилей и добавление автомобилей в список для сравнения;
* поддерживать функционал торговой площадки для продажи и покупки автомобилей с возможностью размещения объявлений, загрузки фотографий и контактов;
* администратор может добавлять и редактировать информацию об автомобилях, видеообзорах и управлять пользователями;
* поддерживать фильтрацию и поиск по различным параметрам автомобилей (марка, год выпуска, тип двигателя и т.д.).

Основные задачи приложения:

* разработать удобный интерфейс для поиска информации об автомобилях;
* предоставить функционал для просмотра видеообзоров на автомобили;
* создать систему сравнения характеристик автомобилей для более детального анализа.

# 2 Описание программных средств

Web-приложение разрабатывается с использованием современных технологий, которые обеспечивают асинхронное взаимодействие с базой данных и поддержку различных платформ. Основная часть проекта реализована с использованием языка программирования JavaScript и платформы Node.js, что обеспечивает высокую производительность и гибкость для реализации бизнес-логики приложения. Платформа Node.js позволяет создавать кроссплатформенные приложения и поддерживает независимость между бизнес-логикой, пользовательским интерфейсом и хранилищем данных, что способствует масштабируемости и расширяемости проекта.

Для реализации асинхронного пользовательского интерфейса используется фреймворк Next.js, который позволяет оптимизировать работу приложения за счет серверного рендеринга и статической генерации страниц. Next.js поддерживает обновление данных на стороне клиента без необходимости перезагрузки страницы, что обеспечивает более плавное и интерактивное взаимодействие пользователя с приложением.

Безопасность пользовательских данных обеспечивается с помощью современных алгоритмов хэширования, таких как bcrypt или argon2, что позволяет защитить пароли пользователей даже в случае утечки базы данных. Эти алгоритмы используют соль для каждого пароля, что предотвращает использование радужных таблиц для расшифровки.

Развёртывание конечного веб-приложения выполняется с использованием Docker, что позволяет создать изолированные контейнеры для различных компонентов системы, обеспечивая их гибкость и независимость при развертывании на различных платформах. Docker предоставляет возможность легко масштабировать приложение и управлять его зависимостями.

Моделирование и проектирование системы выполнено с использованием UML для разработки диаграммы вариантов использования, а также логической и структурной схемы базы данных, что позволяет визуализировать основные компоненты системы и их взаимодействие.

# 3 Описание практического задания

Разработка веб-приложения включала в себя создание функциональной модели, состоящей из нескольких уровней, каждый из которых описывает важные аспекты системы. Все уровни функциональной модели выполнены с использованием схем, которые демонстрируют основные элементы системы, их взаимодействие и потоки данных.

Первым шагом было создание диаграммы вариантов использования с помощью UML, которая визуализирует взаимодействие пользователя с системой. В качестве основных участников были выделены пользователи, администраторы и гости, что позволило определить основные функции приложения, такие как регистрация, авторизация, просмотр автомобилей, управление видеообзорами, сравнение характеристик автомобилей и управление пользовательскими ролями.

Следующим этапом была разработка логической схемы базы данных, включающей основные таблицы и их взаимосвязи. В этой схеме отражены такие ключевые сущности, как пользователи, автомобили, видеообзоры, отзывы. Каждая таблица представлена с учетом всех необходимых полей и связей, что обеспечивает целостность данных и их корректное хранение.

Также была создана структурная схема приложения, отражающая его архитектуру и взаимодействие компонентов. На этой схеме представлены все основные функциональные блоки приложения, включая клиентскую часть на Next.js, серверную часть на платформе Node.js, а также базу данных. Особое внимание уделено независимости между блоками отображения, бизнес-логики и хранилища данных, что позволяет легко масштабировать и расширять систему. Асинхронный интерфейс пользователя на Next.js взаимодействует с сервером через API, что обеспечивает быструю передачу данных в реальном времени.

Функциональные блоки системы разделены по задачам, таким как обработка авторизации и регистрации, управление информацией об автомобилях, обработка видеообзоров и управление пользовательскими данными. Взаимодействие между блоками реализовано через API, что позволяет выполнять обмен данными в реальном времени. Потоки данных в приложении проходят между пользователем, сервером и базой данных через стандартные HTTP-запросы и SQL-запросы для хранения и извлечения данных.

Таким образом, разработанная функциональная модель обеспечивает наглядное понимание системы, а описание блоков и потоков данных демонстрирует их взаимодействие и роль в общей архитектуре приложения.

# 4 Контекстная диаграмма и диаграмма 1-го уровня декомпозиции

Контекстная диаграмма — это верхнеуровневая диаграмма потоков данных (DFD, Data Flow Diagram), которая представляет всю информационную систему в виде одного процесса и его взаимодействие с внешними сущностями. Моделирование делового процесса начинается с построения контекстной диаграммы. На этой диаграмме отображается только один блок – главная бизнес-функция моделируемой системы. Если речь идет о моделировании целого предприятия, то главная бизнес-функция не может быть сформулирована как, например, «продавать продукцию».

Главная бизнес-функция системы – это «миссия» системы, ее значение в окружающем мире. Нельзя правильно сформулировать главную функцию предприятия, не имея представления о его стратегии. При определении главной бизнес функции необходимо всегда иметь ввиду цель моделирования и точку зрения на модель.

Контекстная диаграмма в функциональной модели еще «фиксирует» границы моделируемой бизнес-системы, определяя то, как моделируемая система взаимодействует со своим окружением. Это достигается за счет описания дуг, соединенных с блоком, представляющим главную бизнес-функцию.

Рассмотрим диаграмму получения информации – рисунок 4.1. Входными данными являются данные пользователя (почта и пароль). Механизмы системы включают базу данных (для хранения и поиска информации), систему оплаты, пользователя и администратора приложения, которые взаимодействуют с функциями web-приложения. На выходе система предоставляет пользователю уведомление о результате оплаты и опубликованное объявление.

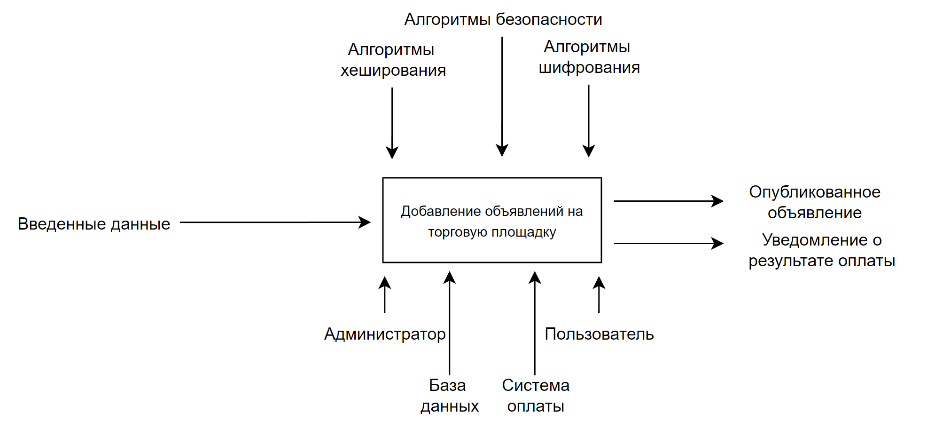


Рисунок 4.1 – Контекстная диаграмма управления торговой площадкой

Принцип функциональной декомпозиции представляет собой способ моделирования типовой ситуации, когда любое действие, операция, функция могут быть разбиты (декомпозированы) на более простые действия, операции, функции. То есть сложная бизнес-логика представлена в виде совокупности элементарных блоков.

При построении диаграммы 1-го уровня декомпозиции бизнес-функция приложения должна быть представлена в виде совокупности элементарных функций. Количество блоков на диаграмме должно быть не менее двух, но и не более шести. Соблюдение этого принципа приводит к тому, что функциональные процессы, представленные в виде IDEF0 модели, хорошо структурированы, понятны и легко поддаются анализу.

На рисунке 4.2 представлена диаграмма декомпозиции 1-го уровня (А0).

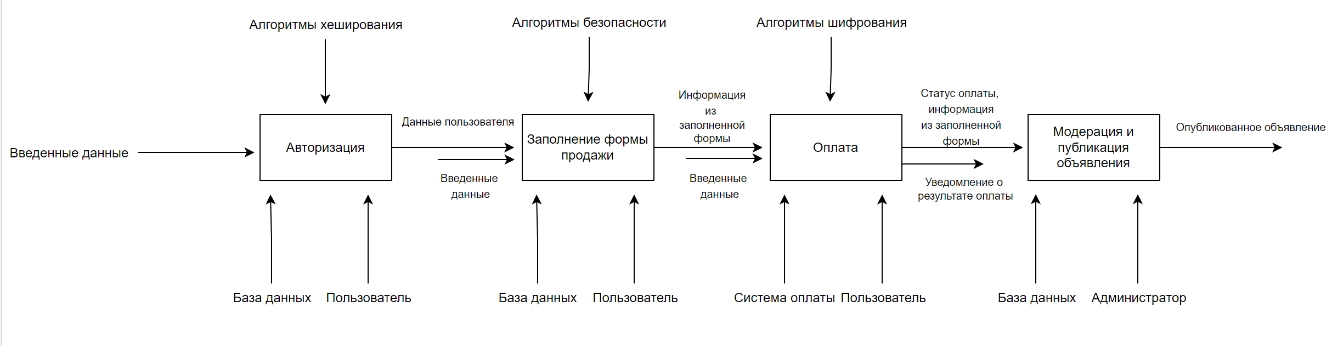


Рисунок 4.2 – Диаграмма А0

На входе в первый блок «Авторизация» поступают данные пользователя, которые он вводит для входа в систему. Пользователь вводит логин и пароль для доступа к системе. На выходе этого блока система подтверждает успешную авторизацию или выдает сообщение об ошибке, если данные некорректны.

Во второй блок «Заполнение формы продажи» поступают параметры автомобиля, которые пользователь вводит в соответствующие поля формы. Данные проверяются на корректность через механизм валидации. На выходе этого блока получается информация о заполненной форме, готовая для передачи на модерацию.

Третий блок «Оплата» принимает данные из заполненной формы. Пользователь вводит данные для оплаты и производит платеж. На выходе этого блока получается статус оплаты и информация из заполненной формы.

Четвертый блок «Модерация объявления» принимает информацию из заполненной формы. Администратор проверяет объявление на соответствие требованиям площадки. На выходе этого блока получается опубликованное объявление.

# 5 Ответы на теоретические вопросы

1. В чем основная сущность структурного подхода?

Сущность структурного подхода к разработке модели состоит в расчленении анализируемой системы на части («черные ящики») и их иерархической организации.

2. Дайте расшифровку терминам DFD, IDEF и SADT.

DFD – Data Flow Diagrams – диаграммы потоков данных.

IDEF – Integration Definition Methodology – объединение методологических понятий.

SADT – Structured Analysis and Design Technique – методология структурного анализа и проектирования.

3. Какие модели строятся с помощью IDEF методологий?

С помощью методологии IDEF строятся функциональные модели, описывая бизнес-функции и контекст поведения.

4. Укажите базовые принципы моделирования в IDEF0.

В IDEF0 реализованы три базовых принципа моделирования процессов:

Принцип функциональной декомпозиции представляет собой способ моделирования типовой ситуации, когда любое действие, операций, функция могут быть разбиты (декомпозированы) на более простые действия, операции, функции. Т.е., сложная бизнес-функция может быть представлена в виде совокупности элементарных функций. Представляя функции графически, в виде блоков, можно «заглянуть внутрь» блока и детально рассмотреть ее структуру и состав.

Принцип ограничения сложности. При работе с IDEF0 диаграммами существенным является условие их разборчивости и удобочитаемости. Суть принципа ограничения сложности состоит в том, что количество блоков на диаграмме должно быть не менее двух и не более шести. Практика показывает, что соблюдение этого принципа приводит к тому, что функциональные процессы, представленные в виде IDEF0 модели, хорошо структурированы, понятны и легко поддаются анализу.

Принцип контекстной диаграммы. Моделирование делового процесса начинается с построения контекстной диаграммы. На этой диаграмме отображается только один блок – главная бизнес-функция моделируемой системы. Если речь идет о моделировании целого предприятия, то главная бизнес-функция не может быть сформулирована как, например, "продавать продукцию". Главная бизнес-функция системы – это "миссия" системы, ее значение в окружающем мире. Нельзя правильно сформулировать главную функцию предприятия, не имея представления о его стратегии. При определении главной бизнесфункции необходимо всегда иметь ввиду цель моделирования и точку зрения на модель.

5. В каких случаях целесообразно применять построение модели «как есть, в а каких «как будет»?

Построение модели “как есть”. Обследование предприятия является обязательной частью любого проекта создания или развития корпоративной информационной системы. Построение функциональной модели “как есть” позволяет четко зафиксировать, какие деловые процессы осуществляются на предприятии, какие информационные объекты используются при выполнении деловых процессов и отдельных операций. Модель “как есть” является отправной точкой для анализа потребностей предприятия, выявления проблем и "узких" мест и разработки проекта совершенствования деловых процессов.

Построение модели “как будет”. Создание и внедрение корпоративной информационной системы приводит к изменению условий выполнения отдельных операций, структуры деловых процессов и предприятия в целом. Это приводит к необходимости изменения системы бизнес-правил, используемых на предприятии, модификации должностных инструкций сотрудников. Модель “как будет” позволяет уже на стадии проектирования будущей информационной системы определить эти изменения. Применение функциональной модели “как будет” позволяет не только сократить сроки внедрения информационной системы, но также снизить риски, связанные с невосприимчивостью персонала к информационным технологиям.