Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

**Отчет по лабораторной работе на тему**:

**«Построение функциональной модели IDEF0»**

Выполнил:

студент 4 курса 7 группы ФИТ

Леонов Д.И.

Цель:

Изучение основ методологии структурного моделирования IDEF. Ознакомление с функциональным моделированием на основе методологии IDEF0, получение навыков по применению IDEF0 для построения функциональных моделей на основании требований к информационной системе.

Минск 2023

# 1. Теоретические вопросы

## 1.1 В чем основная сущность структурного подхода?

Сущность структурного подхода к разработке информационной системы заключается в ее декомпозиции на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимоувязаны. При разработке системы «снизу-вверх» от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов.

## 1.2 Дайте расшифровку терминам DFD, IDEF и SADT.

DFD (Data Flow Diagrams) – это нотация, предназначенная для моделирования информационный систем с точки зрения хранения, обработки и передачи данных.

IDEF (ICAM DEFinition) – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов.

SADT (Structured Analysis and Design Technique) – методология структурного анализа и проектирования, интегрирующая процесс моделирования, управление конфигурацией проекта, использование дополнительных языковых средств и руководство проектом со своим графическим языком (диаграммы).

## 1.3 Какие модели строятся с помощью IDEF методологий?

С помощью IDEF методологий строятся функциональные (IDEF0), информационные (IDEF1) и динамические (IDEF2) модели.

Функциональная модель - это структурированное представление функций, действий или процессов в моделируемой системе или объекте.

Информационная модель представляет собой структуру и семантику информации внутри моделируемой системы или объекта.

Динамическая модель представляет собой изменяющиеся во времени поведенческие характеристики моделируемой системы или объекта.

## 1.4 Укажите базовые принципы моделирования в IDEF0.

В IDEF0 реализованы три базовых принципа моделирования процессов:

1) Принцип функциональной декомпозиции.

Данный принцип представляет собой способ моделирования типовой ситуации, когда любое действие, операция, функция могут быть разбиты (декомпозированы) на более простые действия, операции, функции. Т.е., сложная бизнесфункция 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.

2) 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.

3) 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 бизнес-функции необходимо всегда иметь ввиду цель моделирования и точку зрения на модель.

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

Модель «как есть» целесообразно применять в том случае, когда нужно проанализировать конкретные потребности предприятия, какие процессы и операции будут на нём проводиться. Модель «как будет» позволяет уже на стадии проектирования будущей информационной системы определить изменения условий выполнения отдельных операций, структуры деловых процессов и предприятия в целом, как следствие может привести к изменению системы бизнес-правил, используемых на предприятии, модификации должностных инструкций сотрудников. Обычно модель «как будет» создаётся на основе найденных недостатков в модели «как есть».

# 2. Описываемые функциональные требования

## 2.1 Функциональные требования

Возможности приложения:

- Пользователь:

* Регистрация/Авторизация;
* Просмотр тарифов;
* Каталог автомобилей;
* Просмотр карты с расположением автомобилей;
* Просмотр информации о профиле, о предыдущих заказах;
* Бронирование авто;
* Пополнение баланса;

- Администратор:

* Авторизация;
* Просмотр информации о пользователях;
* Мониторинг арендованных автомобилей;
* Проверка фотоконтроля автомобиля;
* Тех-поддержка пользователей;

## 2.2 Основные системные требования

Приложение состоит из 5 основных компонентов:

* Сервер, хранящий информацию в базе данных и предоставляющий API для мобильного приложения;
* Сервис для карты, отображения автомобилей и данные с их бортового компьютера.
* Сервис для работы с оплатой.
* Мобильное приложение, позволяющее выполнять функции роли пользователя;
* Реляционная СУБД PSQL,

Сервер предоставляет REST API для взаимодействия клиентов с системой. Передача данных осуществляется в формате JSON. Сервер написан на Go, так как приложения, написанные на этом языке, можно легко масштабировать. Сервер предоставляет WEB-интерфейс для администратора со следующим функционалом:

* Авторизация;
* Карта с расположением автомобилей;
* Информация с сервиса карт;
* Заявки на техподдержку;
* Страница для проверки фотоконтроля;
* Информация о пользователях.

Сервис для карты реализован на JAVA, так как этот язык обеспечивает надежность приложений. Сервис должен постоянно отправлять уведомления в систему, и сообщать о различных нарушениях или неполадках автомобилей.

Сервис для работы с оплатой вынесен в отдельный компонент, так как важно не ошибиться при транзакциях связанных с балансом пользователей. Сервис реализован на Python.

Для общения между сервисами использована Kafka, так как она является асинхронным способом общения между сервисами, а также позволяет хранить журнал сообщений.

Мобильное приложение используется для выполнения функций пользователя, написано на языке Flutter, а также имеет локальную базу данных SQLite, содержащую в себе данные пользователя и информацию, необходимую для работы приложения (без подключения к сети интернет) и последующей синхронизацией с сервером. Имеет следующие экраны для роли пользователя:

* Регистрация\авторизация;
* Главный экран;
* Карта с расположением автомобилей;
* Бронирование авто;
* Каталог автомобилей;
* Информация о тарифах;
* Текущая поездка;
* Профиль.

## 2.3 Аппаратные требования

* ОЗУ 1024Mb;
* OS Android 8.0 и выше (API 23+);
* Сетевая пропускная способность около 1Mb/s;
* Внутренняя память не менее 128Mb.

# 3. Описание программных средств

Для создания схем используется draw.io –ресурс , с помощью которой появляется возможность для составления графиков, чертежей, диаграмм, блок-схем. Приложение помогает представить графическую информацию в простом и доступном виде.

Это отличный графический редактор, позволяющий работать с диаграммами и схемами. Программа содержит мощный набор инструментов, который будет полезен для работы. Приложение может применяться в разных сферах. Его используют it-специалисты, менеджеры, аналитики. Программа входит в пакет Office, а также может использоваться в виде отдельного компонента.

Программа, имеющая множество полезных инструментов, которые отлично подходят для управления каким-либо проектом. Приложение располагает мощным арсеналом средств. Причем от пользователя не требуются какие-либо профессиональные знания в технической или изобразительной области, а также связанные с этим навыки. Вы можете использовать готовые шаблоны, фигуры и элементы, с помощью которых добьетесь нужного результата.

Какие-либо навыки работы в draw.io для этого не нужны. Разобраться в программе сможет даже неопытный и начинающий пользователь. Приложение имеет удобный и понятный интерфейс. Вы сможете легко создавать нужные диаграммы и схемы, необходимые для решения конкретных задач.

Возможности приложения зависят от версии. Она может быть обычной или полнофункциональной. В первом случае вы можете просматривать диаграммы и выполнять их печать. Во втором – появляется возможность для создания и редактирования диаграмм. Версия с полным функционалом является более привлекательной.

# 4. Описание практического задания

Под рассмотрение в данной лабораторной работе попала функция регистрации авторизации. Для общего и простого представления используется диаграмма IDEF0. Данная диаграмма позволяет рассмотреть какую-либо бизнес-функцию в общих чертах, чтобы иметь представление о функциональных требованиях приложения. На блок у нас поступают входные данные, сверху расположены инструкции, по которым данные проверяются. Внизу мы пишем обработчиков данных. И в конце указываем выходные данные или же действие, которое произойдёт после завершения нашей функции. На рисунке 1.1 представлена контекстная диаграмма регистрации\авторизации

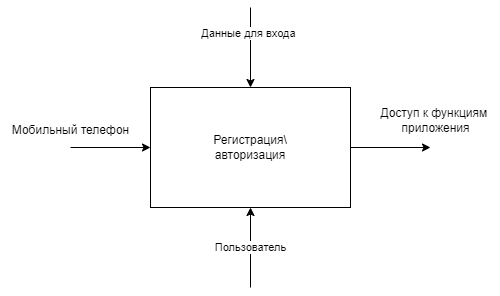


Рисунок 1.1 – Контекстная диаграмма

Дальше диаграмму необходимо декомпозировать, чтобы рассмотреть бизнес-функцию более подробно.

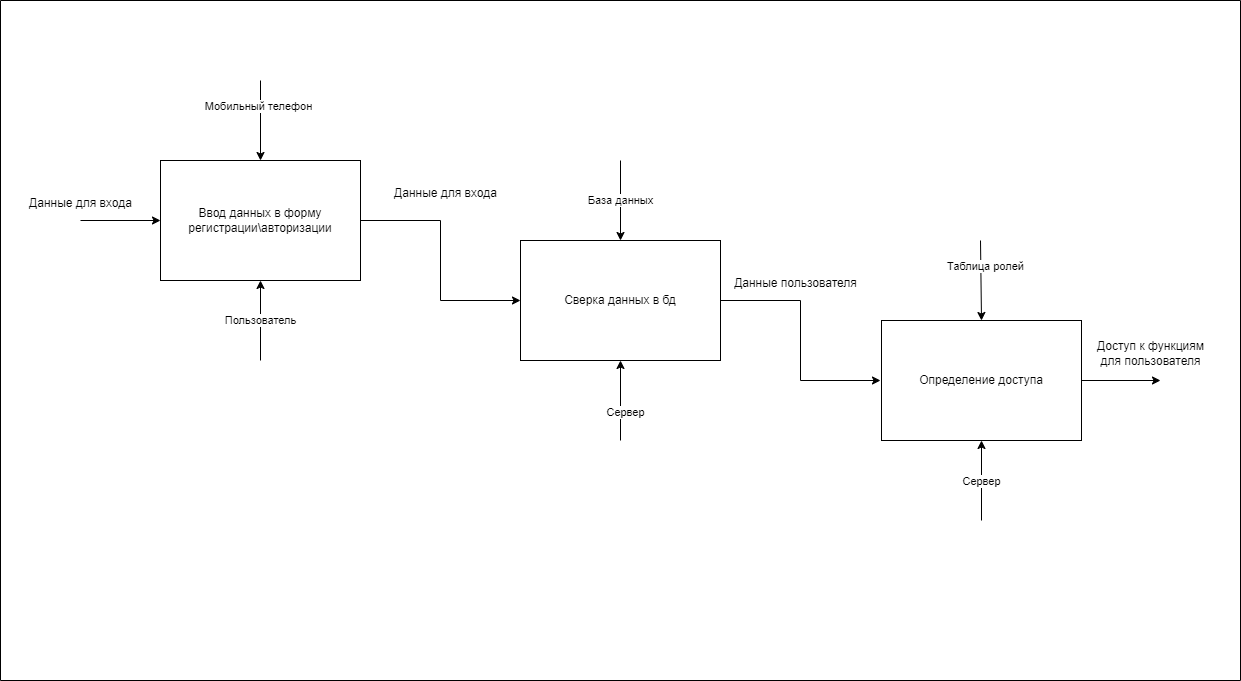


Рисунок 1.2 – Диаграмма первого уровня декомпозиции

Как видно из рисунка 1.2, перед входом в приложение, мы проверяем введённые данные на наличие в базе данных, после чего предоставляем доступ пользователю, если таковой нашёлся в базе данных.

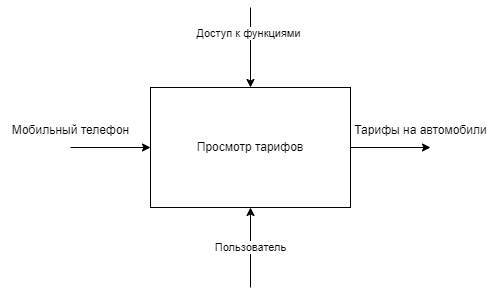


Рисунок 1.3. – Контекстная диаграмма просмотра тарифов

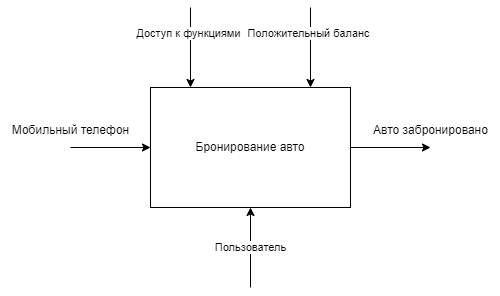


Рисунок 1.4. – Контекстная диаграмма бронирования авто

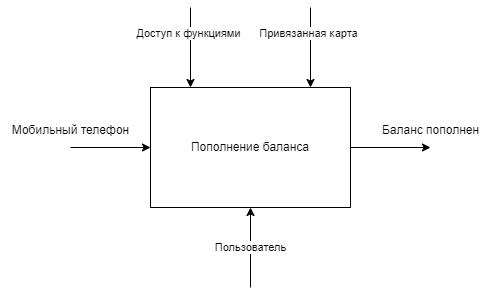


Рисунок 1.5. – Контекстная диаграмма пополнения баланса

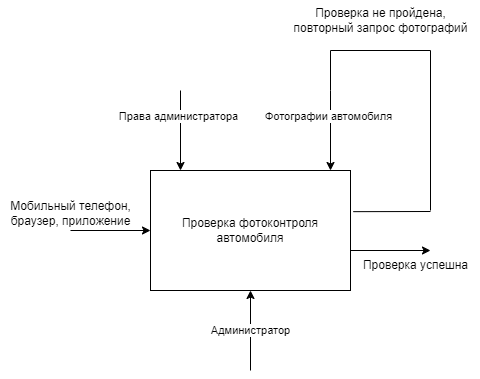


Рисунок 1.5. – Контекстная диаграмма проверки фотоконтроля автомобиля

На рисунке 1.6 представлена диаграмма уровня декомпозиции, которая описывает один из возможных сценариев пользователя. После регистрации/авторизации пользователь может просмотреть доступные автомобили, а затем тарифы к ним. После выбора тарифа авто можно забронировать, однако обязательным условием этого является положительный баланс. Если же баланс отрицательный, пользователя перенаправит на страницу с полонением баланса, где возможно два исхода: баланс пополнен или операция не выполнена.

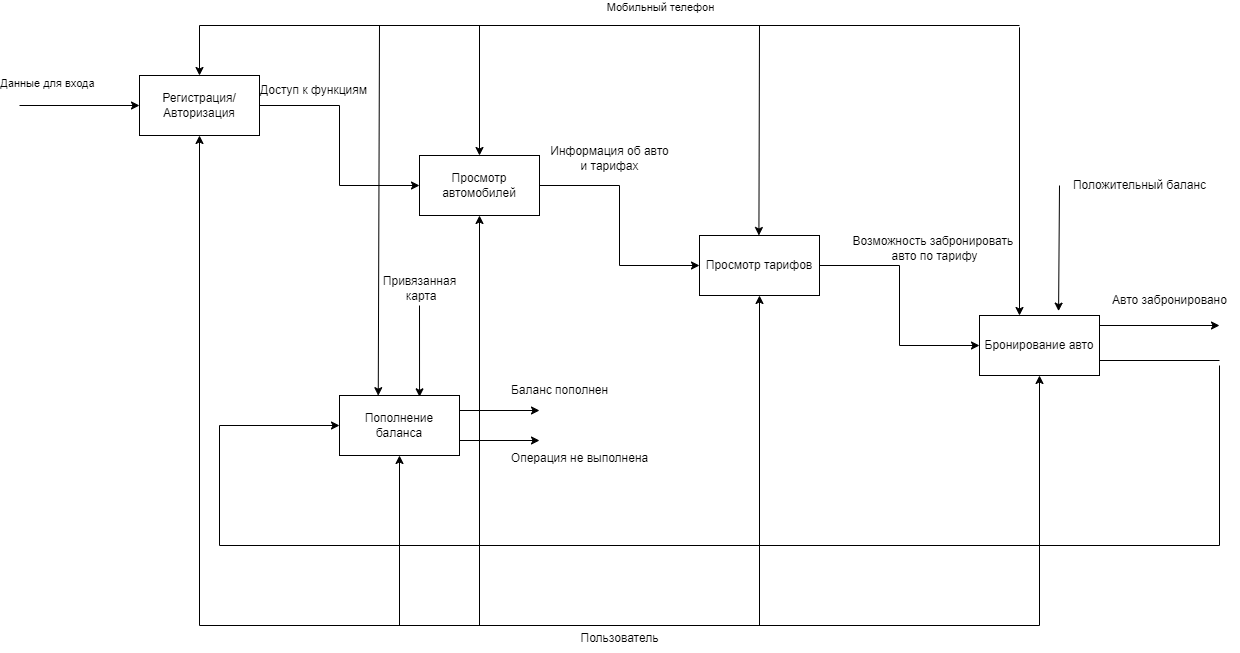
**

Рисунок 1.6. – Диаграмма первого уровня декомпозиции