Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

**Отчёт**

по предмету «Проектирование программного обеспечения»

лабораторная работа №3

«Моделирование процессов с использованием методологии IDEF3»

тема «Интерфейс сети автопарковок»

Студент: Беласин Д.А.

ФИТ 3 курс 1 группа

Преподаватель: Якубенко К. Д.

Минск 2024

# 1. Постановка задачи

Функциональное назначение интерфейса сети автопарковок «ParkSmart» заключается в предоставлении удобной и многофункциональной платформы для аренды парковочных мест на короткий срок. Интерфейс должен обеспечивать пользователям возможность легко находить доступные парковочные места, арендовать их и управлять своим процессом аренды, а также отслеживать информацию по завершенным арендам.

Программа систематизирует данные о доступных парковочных местах, их местоположении, стоимости аренды и статусе, что позволяет пользователям делать осознанный выбор. С помощью интерактивной карты интерфейс предоставляет информацию о ближайших доступных местах в реальном времени.

Интерфейс предлагает пользователям возможность создавать личные кабинеты, где они могут управлять своими арендами, отслеживать историю использования, пополнять счет и получать уведомления о доступных бонусах и акциях. Функционал корзины дает возможность добавлять места в аренду, резервировать их на определенное время и оплачивать несколько арендуемых мест одновременно.

Для операторов парковок интерфейс предоставляет широкий функционал, включая управление парком парковочных мест, мониторинг их статуса, отслеживание активности пользователей и управление статистикой аренды через специальную панель администратора.

В интерфейсе предусмотрена гибкая система тарифов, основанная на длительности аренды и типе парковочного места, а также возможность возврата автомобиля на любые станции сети, что делает процесс аренды удобным и гибким для пользователей.

# 1.1 Описание объектов системы

Описание объекта «парковочное место»:

* Это пространство, которое можно арендовать через приложение для временной парковки автомобиля.
* Основные атрибуты парковочного места: идентификационный номер, текущее местоположение, доступность для аренды, тип места (стандартное, электрическое), стоимость аренды.
* Связи: Парковочное место принадлежит сети аренды и может быть арендовано пользователями.

Описание объекта «оператор»:

* Это лицо или организация, управляющее сетью парковочных мест, следящее за их состоянием, доступностью и обработкой платежей.
* Основные атрибуты оператора: имя компании, контактная информация, список доступных парковочных мест и станции аренды.
* Связи: оператор создает и управляет парком парковочных мест, отслеживает их состояние и взаимодействует с пользователями через приложение.

Описание объекта «пользователь»:

* Это физическое лицо, которое может арендовать парковочное место, следить за своими поездками и управлять арендой через личный кабинет.
* Основные атрибуты пользователя: имя, контактные данные, история аренды, текущие бронирования.
* Связи: пользователь арендует парковочные места, управляет своими арендами и взаимодействует с платёжной системой.

Описание объекта «аренда»:

* Это процесс, который позволяет пользователю взять парковочное место на определённый срок и оплатить его использование.
* Основные атрибуты аренды: уникальный идентификатор, продолжительность, дата начала и завершения, стоимость аренды, статус (активная, завершена).
* Связи: аренда привязана к пользователю и парковочному месту.

Описание объекта «платёж»:

* Это финансовая транзакция, выполняемая пользователем для оплаты аренды парковочного места.
* Основные атрибуты платежа: сумма, способ оплаты (банковская карта, электронные кошельки), статус (успешный, неудачный), дата проведения.
* Связи: платёж связан с арендой и пользователем.

Программа должна поддерживать следующие функции:

* Регистрация и авторизация пользователей.
* Личный кабинет пользователя должен включать историю аренд и доступ к текущим арендам.
* Отображение доступных парковочных мест на карте с информацией о местоположении, типе и статусе.
* Процесс аренды парковочного места должен включать выбор пункта аренды и типа места с возможностью оплаты через приложение.
* Операторы должны иметь личный кабинет с доступом к управлению парком парковочных мест и мониторингу активности.
* Поиск и бронирование парковочных мест: пользователи смогут искать доступные парковочные места через карту.

Процесс аренды и возврата: процесс аренды будет простым и интуитивным: пользователь выбирает парковочное место, завершает бронирование и оплачивает аренду через приложение. Возврат автомобиля будет возможен на любое свободное место в сети.

# 2. Описание программных средств

Описание программного средства Draw.io:

* Название: Draw.io.
* Версия: Актуальная версия (по состоянию на 2024 год). Программа обновляется регулярно.
* Разработчик: JGraph Ltd.
* Адрес загрузки: https://app.diagrams.net/. Draw.io доступен как онлайн-приложение, так и в виде настольного клиента для различных операционных систем.
* Режим использования: Draw.io предоставляет бесплатный доступ ко всем своим функциям без ограничений. Это полностью бесплатный инструмент с открытым исходным кодом, доступный как для личного, так и для корпоративного использования.
* Платформы: Draw.io доступен как веб-приложение, которое работает в любом современном браузере. Также доступны настольные версии для Windows, macOS и Linux.
* Типы моделей, с которыми работает Draw.io: Draw.io используется для создания диаграмм, схем, моделей процессов и прочих визуальных структур. Программа поддерживает работу с блок-схемами, диаграммами UML, IDEF0, сетевыми диаграммами, архитектурными схемами, организационными диаграммами и многими другими визуальными представлениями данных.
* Основные функции: Создание и редактирование диаграмм различных типов (например, UML, блок-схемы, BPMN, ERD). Поддержка импорта и экспорта файлов в различных форматах, включая XML, PNG, SVG и PDF. Возможность совместной работы с другими пользователями через облачные сервисы (Google Drive, OneDrive, Dropbox и др.). Функции автосохранения и интеграции с популярными системами управления версиями. Поддержка версии оффлайн, которая позволяет работать без подключения к интернету. Гибкая система настройки шаблонов и библиотек элементов для ускорения работы.

Draw.io — это удобный, бесплатный инструмент для создания диаграмм и схем, который поддерживает совместную работу и интеграцию с облачными сервисами, обеспечивая гибкость и доступность на различных платформах.

# 3. Описание практического задания

IDEF0 (Integrated Definition for Function Modeling) — это метод моделирования, используемый для описания бизнес-процессов и систем с точки зрения их функций и взаимодействий.

В IDEF0 каждая функция представлена блоком, который описывает действие или процесс. Функции выполняют работу или преобразуют входные данные в выходные.

Контекстная диаграмма – это самый высокий уровень модели IDEF0, который описывает всю систему в целом как одну бизнес-функцию. На этой диаграмме изображается только одна большая функция, представляющая весь процесс или бизнес-систему. Контекстная диаграмма «Система аренды парковочных мест» (рис 3.1) описывает, как система взаимодействует с пользователями и операторами, обрабатывая информацию и услуги.

Пользователи вводят свои данные для регистрации и аренды парковочных мест, операторы добавляют парковочные места, следят за их состоянием и добавляют станции аренды. Система возвращает подтверждения аренды, оплаты и возврата автомобиля. Процессы управляются стандартами безопасности, правилами пользования арендованным имуществом. Включают базы данных пользователей операторов и парковочных мест, системы регистрации, оплаты и поиска доступных мест.

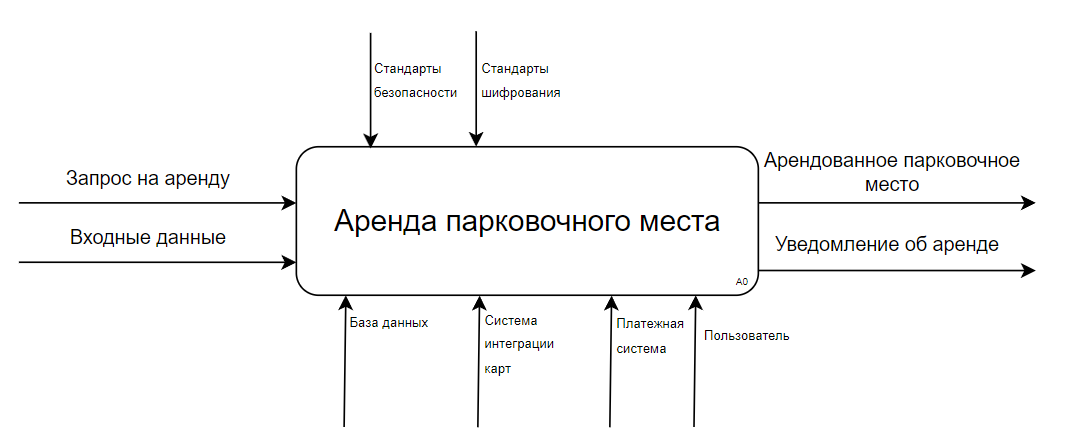


Рисунок 3.1 – Контекстная диаграмма

Стрелки описывают взаимодействие функции с внешними и внутренними элементами системы. Есть четыре типа стрелок:

* Входы (inputs): то, что функция использует для выполнения действия. Они поступают слева к блоку.
* Выходы (outputs): результаты работы функции. Выходы указываются справа от блока.
* Управление (controls): правила, стандарты, инструкции и другие элементы, которые контролируют выполнение функции. Стрелки управления приходят сверху к блоку.
* Механизмы (mechanisms): ресурсы, необходимые для выполнения функции, такие как сотрудники, оборудование или системы. Механизмы приходят снизу к блоку.

Диаграмма 1-го уровня декомпозиции, представленная на рисунке 3.2, является детализацией контекстной диаграммы. Она разбивает ту общую функцию, которая была на контекстной диаграмме, на основные подфункции. Это первый уровень детализации, который позволяет лучше понять, как выполняется общая функция системы.

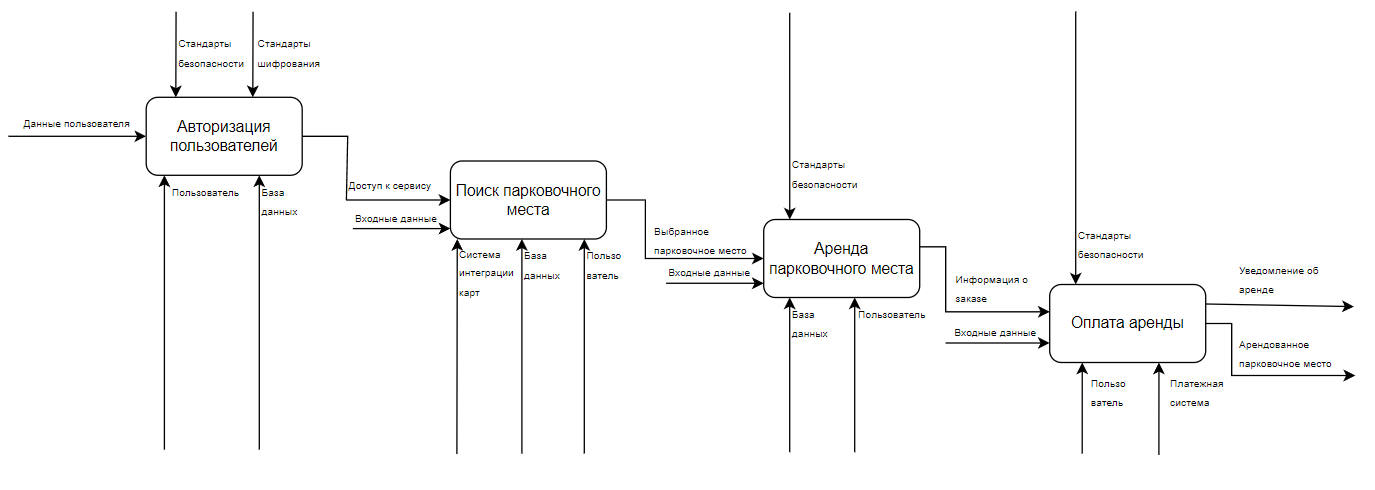


Рисунок 3.2 – Диаграмма 1-го уровня декомпозиции

Диаграмма разбивает процесс аренды парковочных мест на несколько ключевых этапов. Сначала происходит авторизация пользователей, где система проверяет их данные и предоставляет доступ к личному кабинету. Пользователи могут искать парковочные места используя интегрированную карту. После выбора места начинается этап аренды. На этом этапе пользователи выбирают условия аренды, время аренды и переходят к оплате. Каждый этап связан с определёнными входами, управлением и выходами, что обеспечивает плавную работу системы.

Каждый из этих этапов связан с определёнными входами, управлением и выходами, что обеспечивает плавную и слаженную работу системы. Входные данные включают в себя введённую информацию о предпочтениях аренды, тогда как управление подразумевает координацию.

Для информационной системы «Интерфейс сети автопарковок» были разработаны следующие бизнес-процессы:

**1.Авторизация пользователей**

* Ввод данных пользователя.
* Проверка пльзовательских данных.
* Определение роли пользователя.
* Доступ к системе.

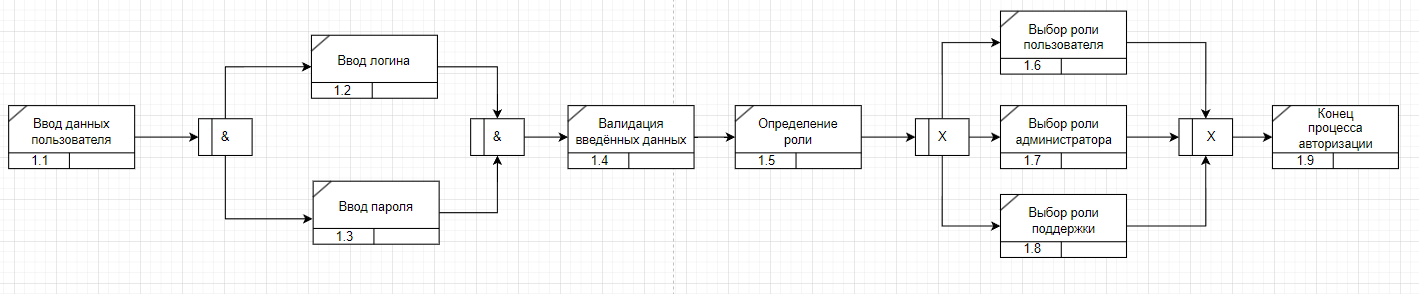


Рисунок 3.3 – Процесс авторизации пользователя

**2.Поиск доступных парковочных мест:**

* Получение данных о парковках.
* Поиск парковочных мест.
* Фильтрация по параметрам.
* Сортировка результатов.
* Поиск по введенным данным.
* Выбор парковочного места.

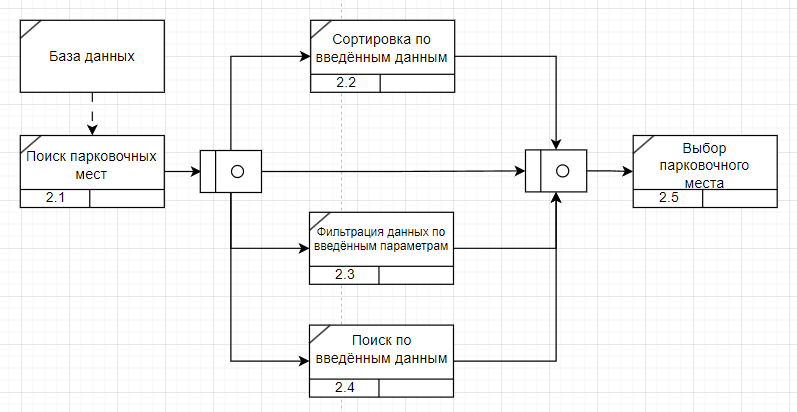


Рисунок 3.4 – Процесс поиска доступных парковочных мест

Пользователь инициирует поиск свободных парковочных мест, вводя нужные параметры, такие как местоположение, дата и время. Система извлекает информацию о текущих доступных местах и отображает результаты поиска, позволяя пользователю быстро находить подходящие варианты.

В результате, пользователю предоставляются результаты поиска, наглядно отображая все подходящие варианты. Это позволяет пользователю быстро и эффективно находить наиболее удобные и подходящие парковочные места, экономя время и усилия. Такой подход значительно упрощает процесс поиска и делает его более комфортным.

3.Бронирование парковочного места:

* Просмотр корзины.
* Проверка корзины.
* Удаление парковочного места.
* Создание записи бронирования.

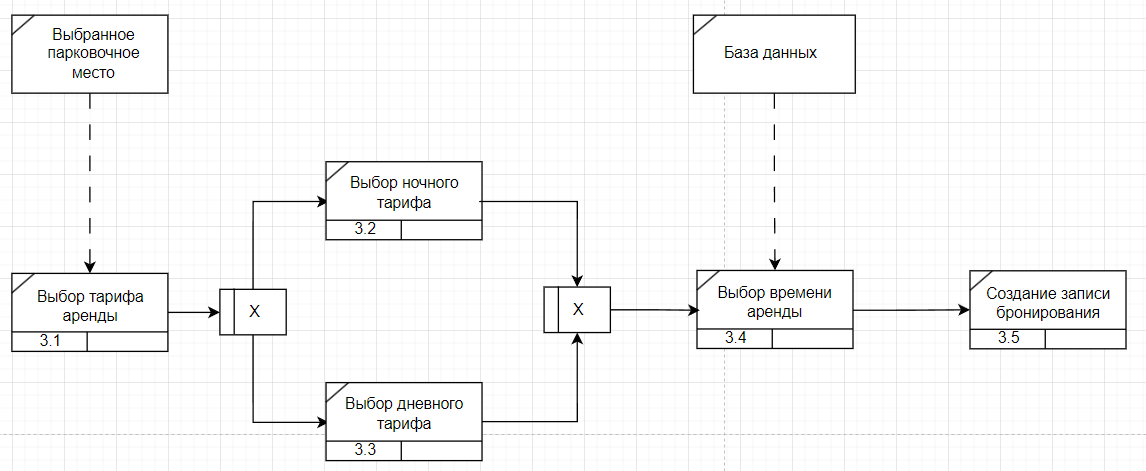


Рисунок 3.5 – Процесс аренды парковочного места

После того как пользователь выбрал доступное парковочное место, он переходит к следующему этапу процесса — подтверждению своего намерения забронировать это место. На этом этапе пользователь имеет возможность внимательно просмотреть информацию о выбранном парковочном месте, включая его расположение, доступные сроки аренды и стоимость. Это позволяет ему убедиться, что именно это место соответствует его ожиданиям и требованиям.

4.Оплата парковочного места:

* Получение записи бронирования.
* Выбор способа оплаты.
* Ввод платежной информации.
* Проверка наличия средств.
* Оплата.
* Ожидание подтверждения оплаты.
* Завершение оплаты.

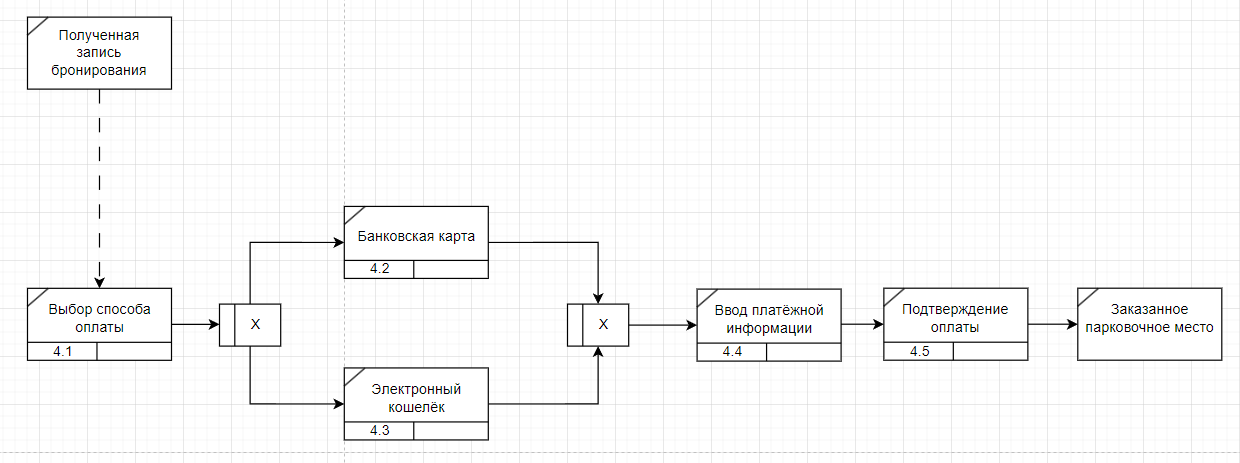


Рисунок 3.6 – Процесс оплаты парковочного места

На данном этапе пользователь производит оплату выбранного им парковочного места и получает желанное место.

В представленных схемах были использованы различные типы перекрестков, которые играют важную роль в организации и управлении потоками информации в системе управления бронированием лекарств.

# 4. Теоретические вопросы подготовки к лабораторной работе:

1) В чем основная сущность структурного подхода?

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.

2) Дайте расшифровку терминам DFD, IDEF и SADT.

DFD - Data Flow Diagram

IDEF - Integrated Definition for Function Modeling

SADT - Structured analysis and design technique

3) Какие модели строятся с помощью IDEF методологий?

IDEF0 — Функциональное моделирование.

IDEF1 — Информационное моделирование.

IDEF1X — Моделирование реляционных данных

IDEF2 — Моделирование динамики систем.

IDEF3 — Моделирование процессов и сценариев.

IDEF4 — Проектирование объектно-ориентированных систем.

IDEF5 — Моделирование онтологий.

IDEF6 — Моделирование дизайна системы (устарела).

IDEF14 — Моделирование сетевых взаимодействий.

4) Укажите базовые принципы моделирования в IDEF0.

1. Функциональная декомпозиция.

2. Блоки функций (функциональные блоки).

3. Контекстная диаграмма.

4. Связи через стрелки.

* Входы (inputs): то, что функция использует для выполнения действия. Они поступают слева к блоку.
* Выходы (outputs): результаты работы функции. Выходы указываются справа от блока.
* Управление (controls): регламенты которые контролируют выполнение функции. Стрелки управления приходят сверху к блоку.
* Механизмы (mechanisms): ресурсы, необходимые для выполнения функции. Механизмы приходят снизу к блоку.

5. Иерархическая структура.

6. Моделирование как черно-белого ящика.

7. Сосредоточение на том, "что" делает система.

5) «Как есть» полезно для выявления ошибок/слабых мест в программе при её модернизации, улучшении или просто для документации, текущей систему. «Как будет» тоже полезно для модернизации как «скелет» для будущей системы, в котором можно будет просмотреть будущие слабые места, нуждающиеся в оптимизации или переосмыслении. Так же такая диаграмма полезна при осмыслении программы с нуля.