Технический проект

1. Функциональная структура проекта

Для описания основных автоматизируемых бизнес-процессов были составлены диаграммы различных уровней детализации в нотации IDEF0. Контекстная диаграмма представлена на рисунке 1.

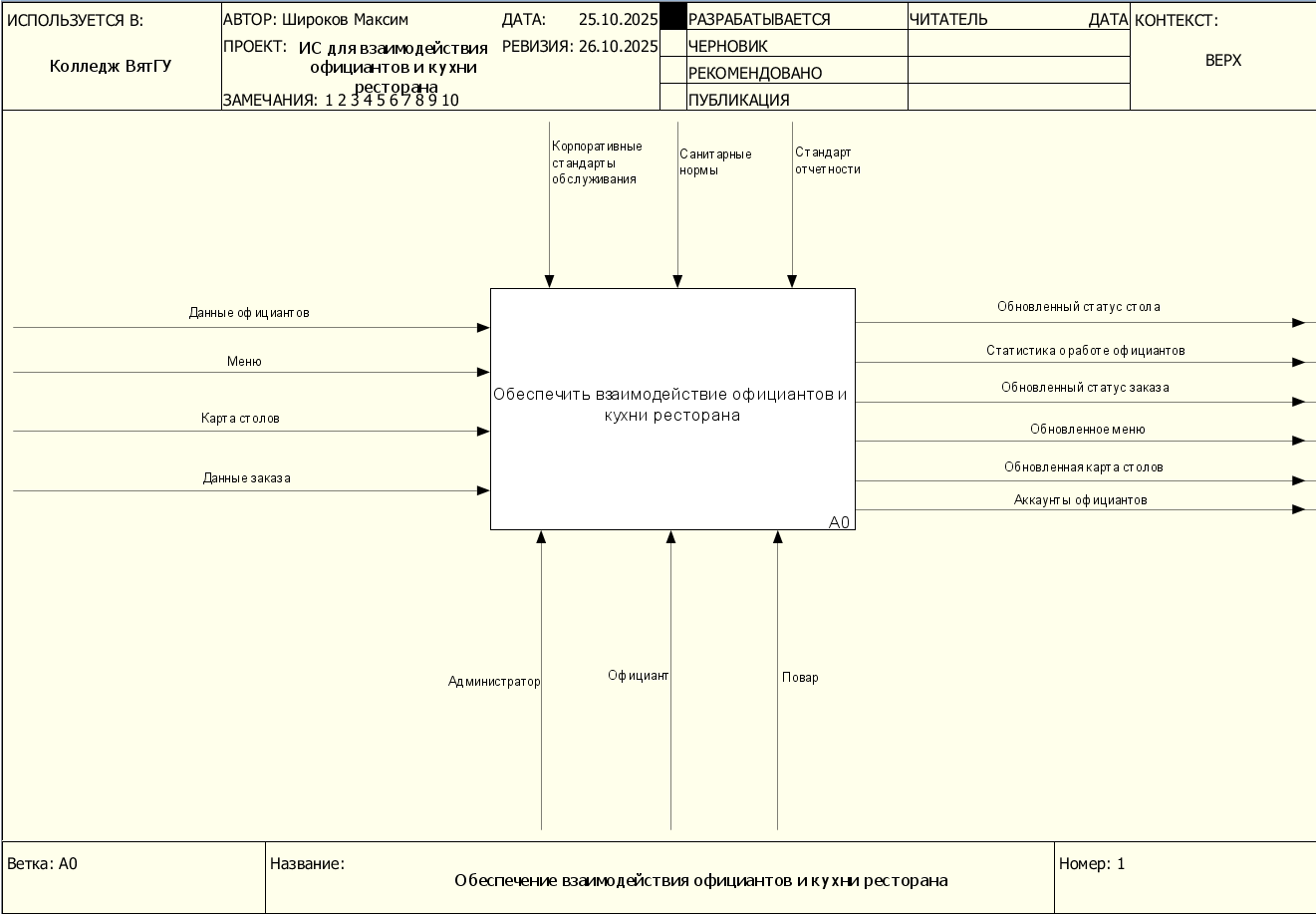


Рисунок 1 – Контекстная диаграмма в нотации IDEF0

Таким образом, взаимодействующими с системой будут пользователи с ролями: администратор (обеспечивает актуальность данных ресторана в системе), официант (управляет заказами клиентов) и повар (обеспечивает информирование о статусе приготовления блюд).

Диаграмма детализации общего бизнес-процесса представлена на рисунке 2.

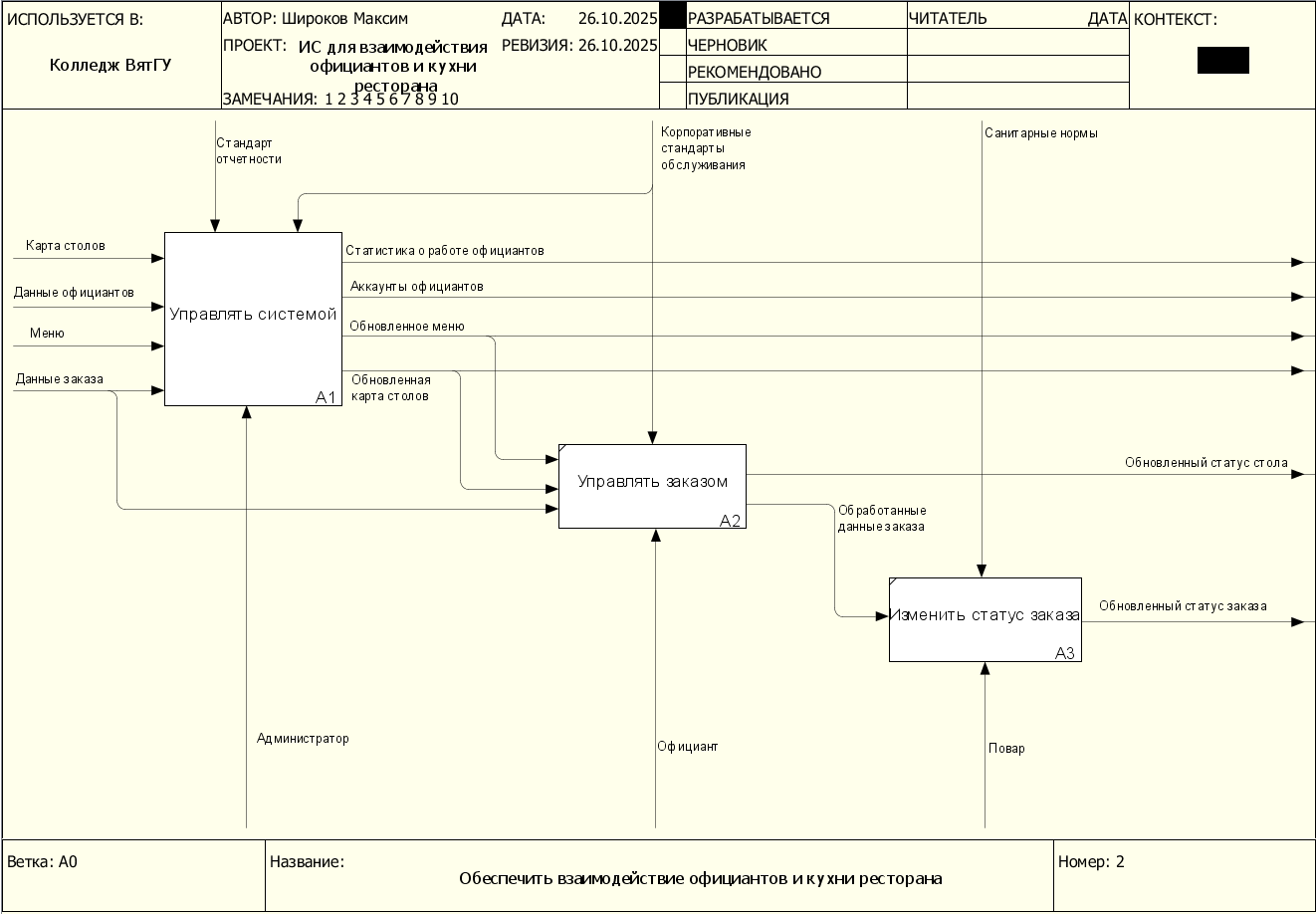


Рисунок 2 – Детализированная диаграмма в нотации IDEF0 для описания процесса обеспечения взаимодействия официантов и кухни ресторана

Основными бизнес-процессами, согласно составленной диаграмме, являются:

* управление системой администратором, включающее изменение карты расположения столов в зале, управление меню и аккаунтами официантов в системе, а также получение статистики о работе официантов;
* управление заказом, проводимое официантом – данный процесс предполагает создание, удаление и редактирование заказов с автоматическим изменением статуса стола при начале/завершении обслуживания столика;
* изменение статуса заказа поваром, подразумевающее изменение сведений о степени готовности блюда при помощи голосовых команд.

Поскольку процессы управления заказом и изменения его статуса являются простыми, в дальнейшей детализации они не нуждались.

Диаграмма детализации процесса управления системой администратором представлена на рисунке 3.

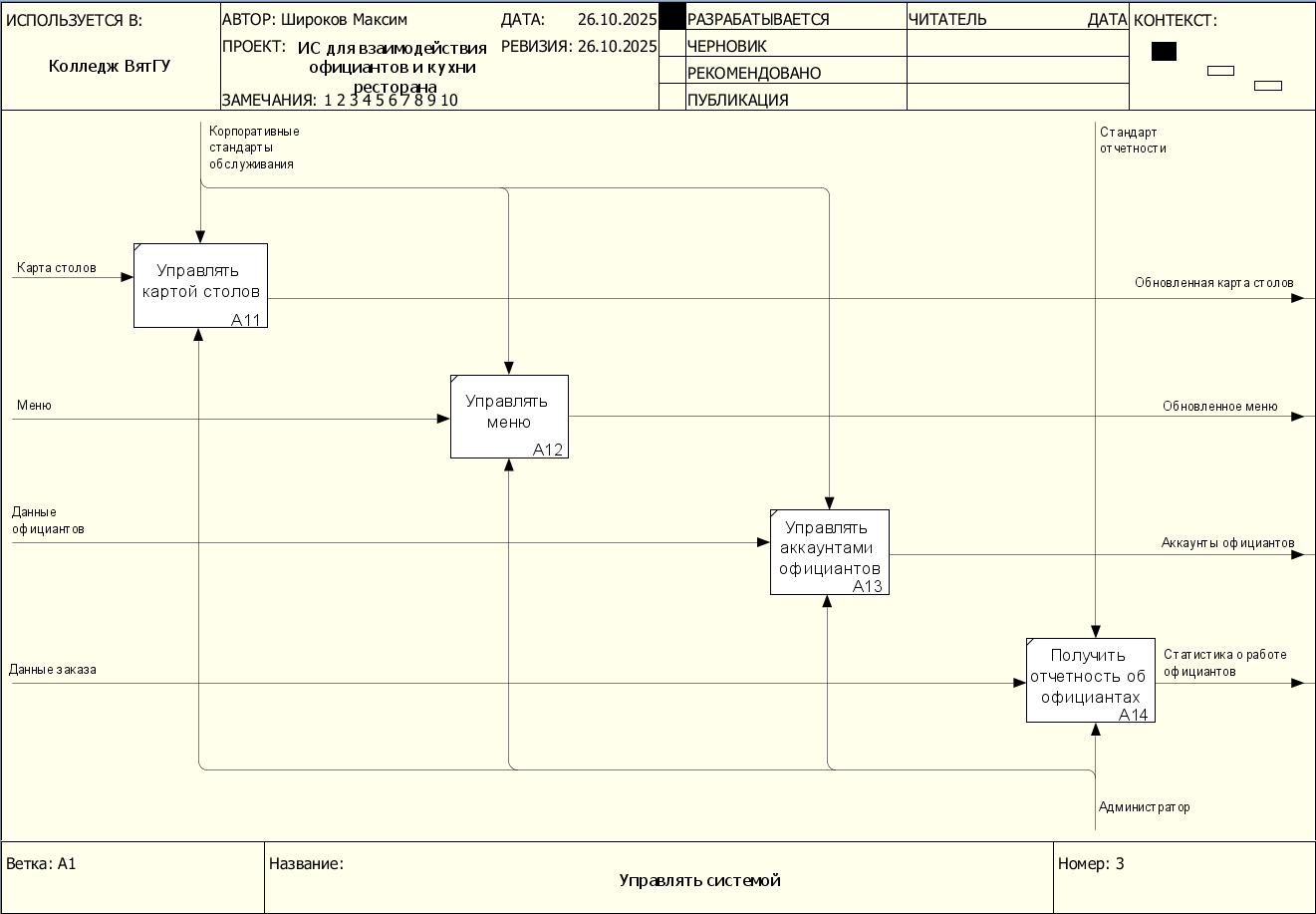


Рисунок 3 – IDEF0 диаграмма детализации процесса управления системой

Согласно составленной диаграмме, администратор имеет все описанные ранее возможности для контроля и обеспечения корректного взаимодействия официантов и кухни ресторана.

Разграничение функциональных возможностей ролей в системе обеспечивается наличием авторизации каждого пользователя.

Для описания данных использующихся и получаемых при работе с системой были составлены диаграммы потоков данных в нотации DFD. Контекстная диаграмма потоков данных представлена на рисунке 4.

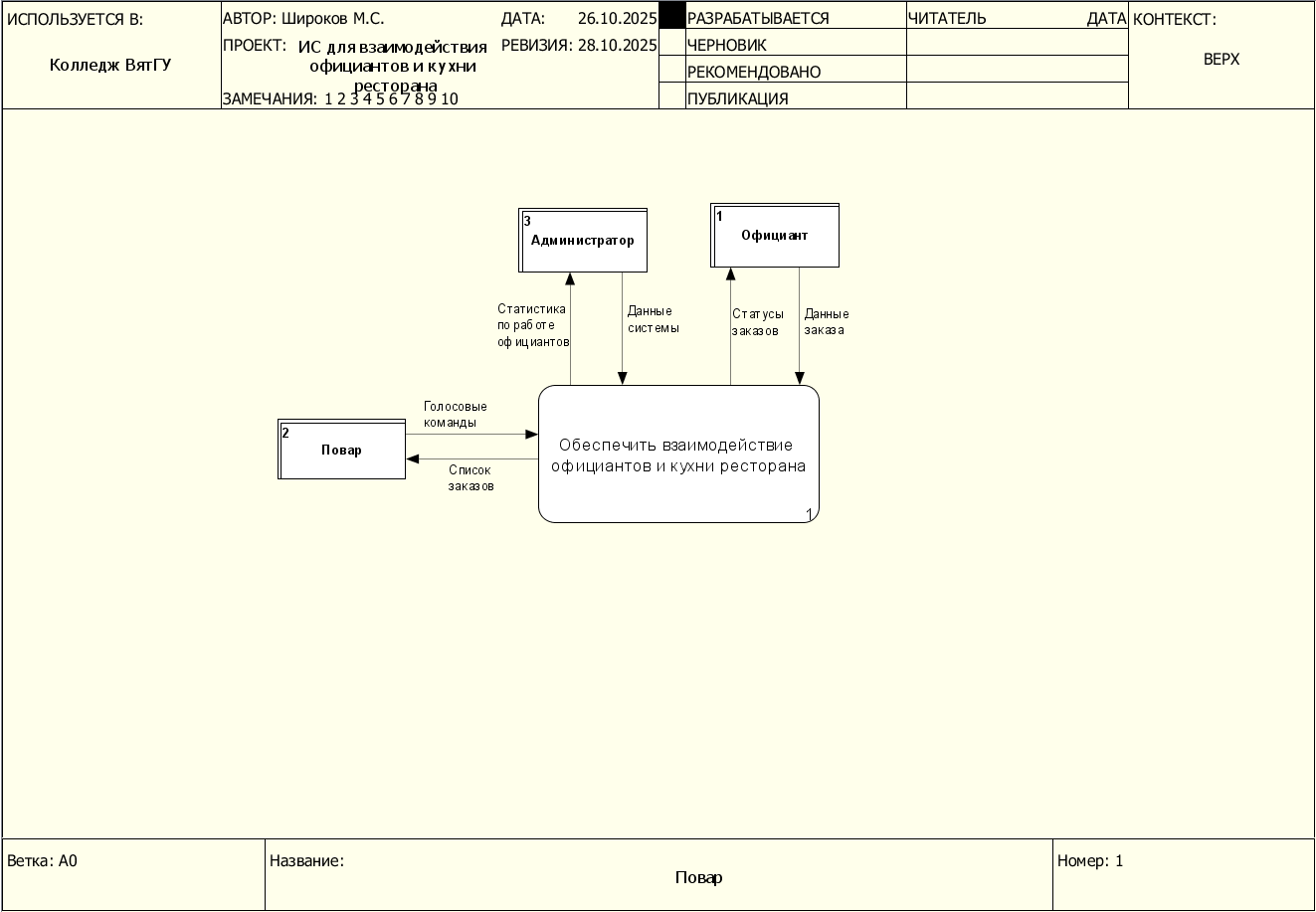


Рисунок 4 – Контекстная диаграмма потоков данных для процесса автоматизации обеспечения взаимодействия официантов и кухни ресторана

Согласно составленной диаграмме, повар, при взаимодействии с системой, получает информацию о текущих заказанных блюдах и подает на вход голосовые команды, которые система использует для управления статусами заказов. Администратор в свою очередь загружает в систему данные о меню, расположении столов и списке официантов, взаимодействующих с системой, а также получает статистическую информацию о их работе. Официант, при взаимодействии с системой, записывает в нее данные о заказах клиентов, а в ответ получает статусы приготовления каждого блюда для организации своего рабочего процесса.

Детализированная диаграмма потоков данных для системы представлена на рисунке 5.

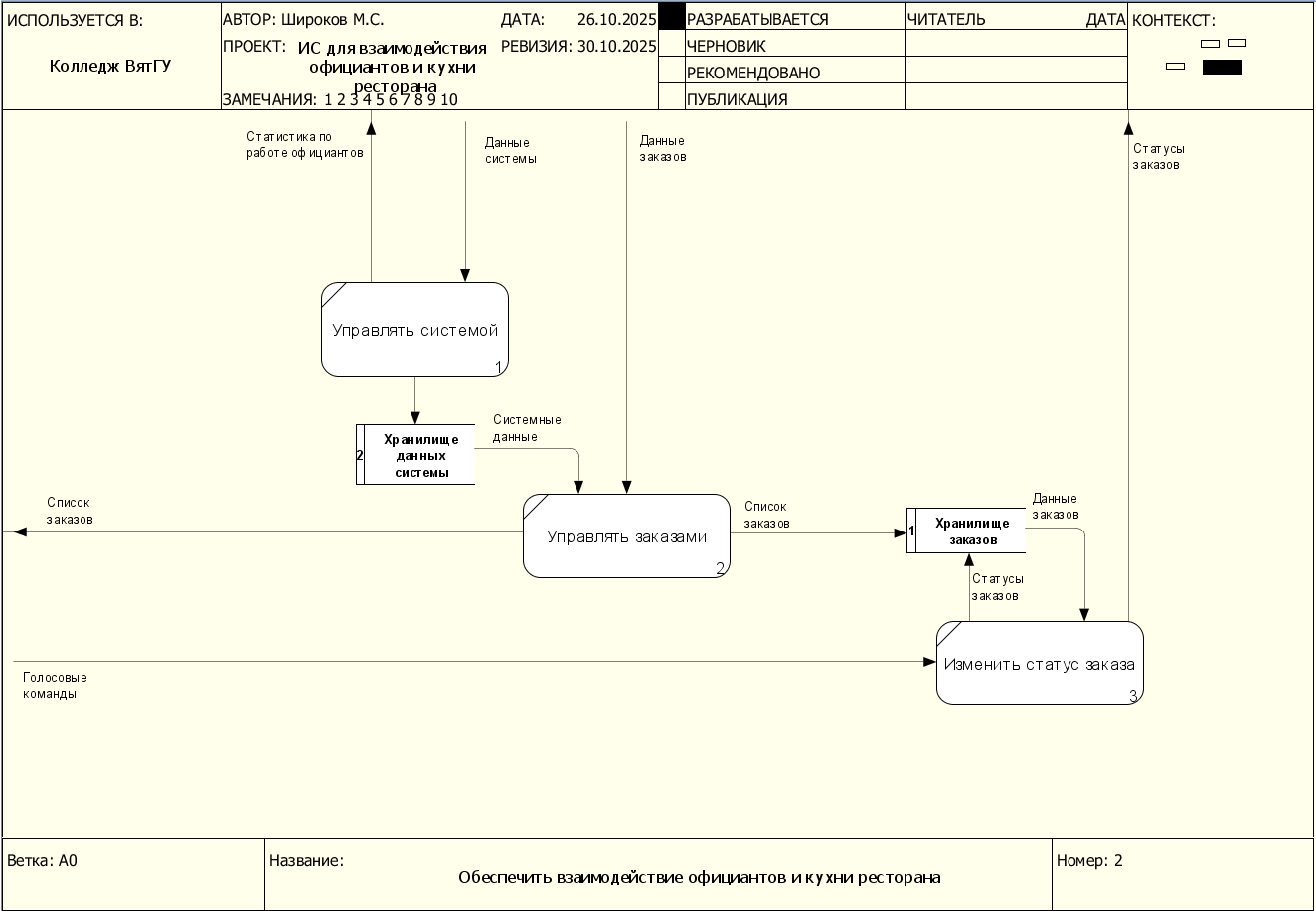


Рисунок 5 – Диаграмма детализации потоков данных в системе автоматизации взаимодействия официантов и кухни ресторана

Согласно полученной диаграмме, процесс управления системой требует только данных от администратора, в свою очередь полученные системой данные о предприятии сохраняются и используются при управлении заказами официантом, в частности официант выбирает столы на схеме зала и позиции из меню. Полученная при управлении заказами информация сохраняется, выводится на клиент поваров, и используется для изменения статусов блюд в заказе. Полученные изменения также сохраняются.

1. Спецификация процессов

Основными функциональными процессами в разрабатываемой системе являются:

* создание заказа в системе;
* редактирование существующего заказа;
* изменение статуса приготовления блюда;
* редактирование меню;
* редактирование схемы зала;
* получение статистики о работе официантов.

На рисунке 6 представлена BPMN-модель процесса создания заказа, доступного официанту. Целью данного процесса является создание в системе заказа на приготовление блюд с учетом пожеланий клиентов. Первым этапом официант выбирает один или несколько столов на схеме зала, затем выбирает нужные позиции из меню и указывает их количество, при необходимости добавляя комментарий к блюду. Затем официант подтверждает создание заказа и управление переходит к системе. Система отмечает выбранные столы обслуживаемыми и блокирует их для создания нового заказа, сохраняет данные заказа и обновляет список заказов на устройствах кухни и администратора.



Рисунок 6 – Модель процесса создания заказа

На рисунке 7 представлена BPMN-модель процесса редактирования заказа, доступного официанту. Целью данного процесса является внесение изменений в существующий заказ. Первым этапом процесса является выбор официантом нужного заказа (выбор осуществляется через схему зала путем нажатия на нужный столик) и переход в режим его редактирования. Далее система отображает список блюд, которые еще не взяты в приготовление поварами, чтобы избежать ситуации изменения уже выполняющегося заказа. Следующим этапом официант вносит необходимые изменения в заказ и подтверждает их. Затем система сохраняет изменения и обновляет список заказов на устройствах кухни и администратора.



Рисунок 7 – Модель процесса редактирования заказа

На рисунке 8 представлена BPMN-модель процесса изменения статуса блюда, доступного повару. Целью данного процесса является поддержание в актуальном состоянии статуса блюда, чтобы оптимизировать процесс информирования официанта о готовности заказа. Данный процесс предполагает открытие окна с списком заказов. В начале процесса повар произносит команду в формате «заказ 1 взят в работу/приготовлен». Далее система переводит голосовую команду в текст и пытается ее распознать. Если программа распознана статус блюда изменяется и обновляется на устройствах официантов и администратора.

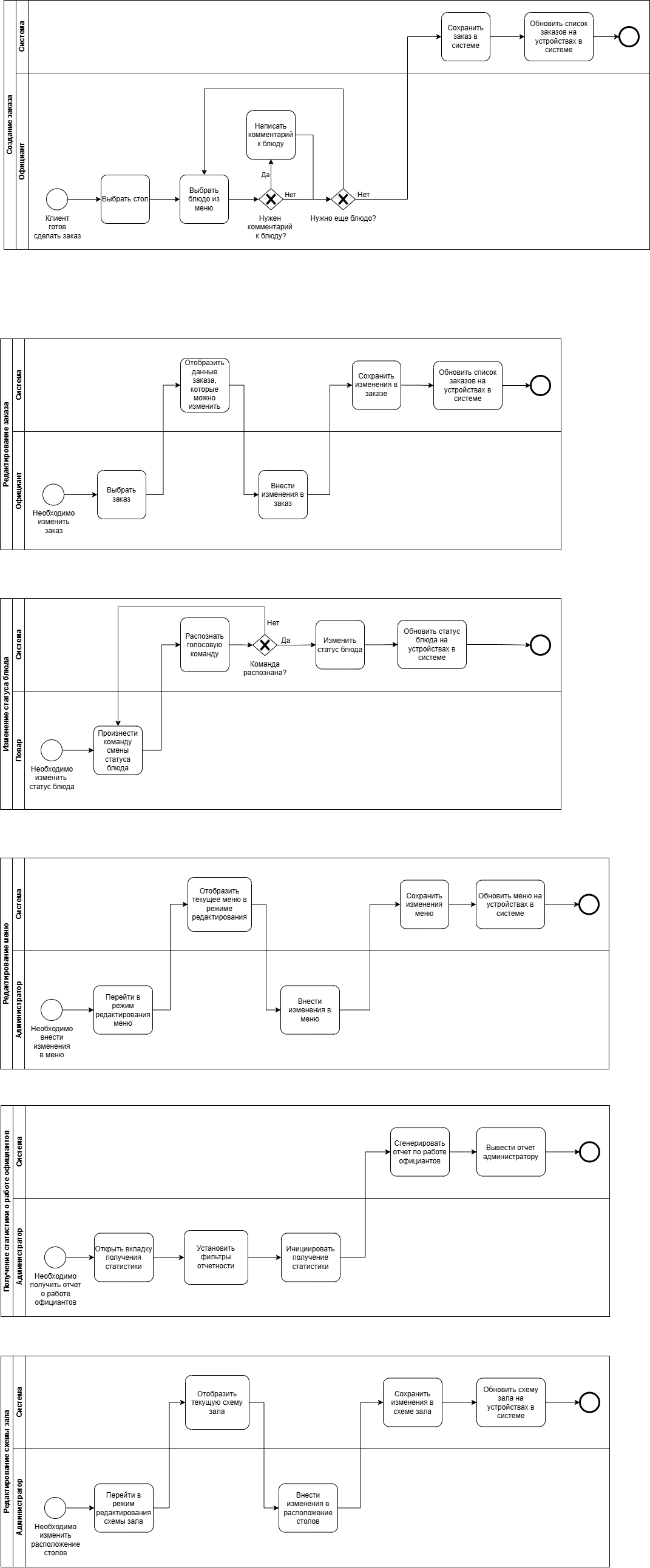


Рисунок 8 – Модель процесса изменения статуса блюда

На рисунке 9 представлена BPMN-модель процесса редактирования меню, доступного администратору. Целью данного процесса является поддержание меню в актуальном состоянии. Первым шагом процесса администратор открывает меню и переходит в режим его редактирования. Далее система отображает текущее меню и предоставляет возможности его редактирования. Затем администратор вносит требуемые изменения в меню и подтверждает изменения. После подтверждения система сохраняет изменения и обновляет меню на устройствах официантов.



Рисунок 9 – Модель процесса редактирования меню

На рисунке 10 представлена BPMN-модель процесса получения статистики о работе официантов, доступного администратору. Целью данного процесса является обеспечение администратору возможности оценки работы официантов, зарегестрированных в системе, по статистическим данным о их деятельности. Первым этапом процесса является открытие администратором раздела получения статистики и установка фильтров, в частности дат сбора данных и списка официантов. Далее администратор подтверждает получение отчета, затем система собирает статистику, генерирует отчет и выводит результат администратору.

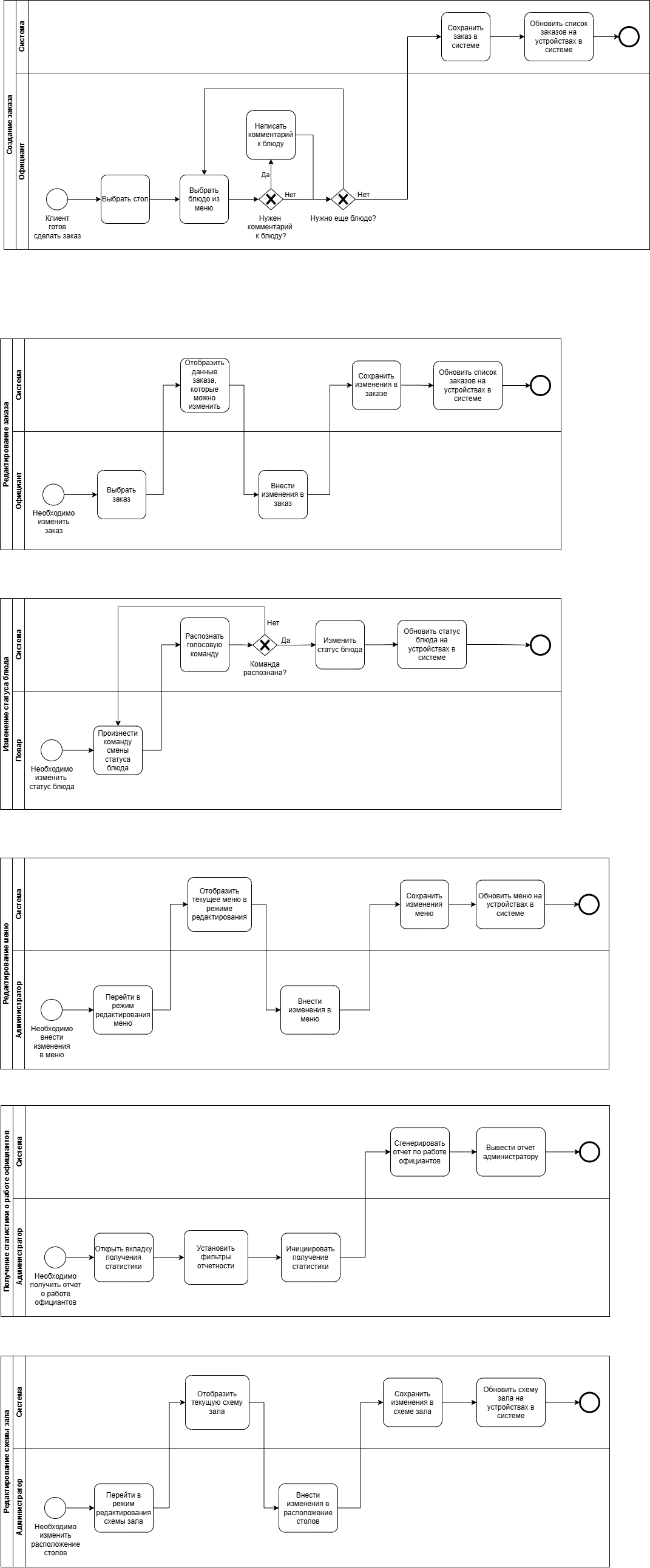


Рисунок 10 – Модель процесса получения статистики о работе официантов

На рисунке 11 представлена BPMN-модель процесса редактирования схемы зала, доступного администратору. Целью данного процесса является поддержание в актуальном состоянии схемы расположения столов в зале. Первым этапом процесса администратор переходит в режим редактирования схемы зала, далее система отображает ему текущую схему зала с возможностями редактирования. Затем администратор вносит изменения в карту столов и подтверждает изменения. В завершение система сохраняет изменения и обновляет схему зала на устройствах в системе.

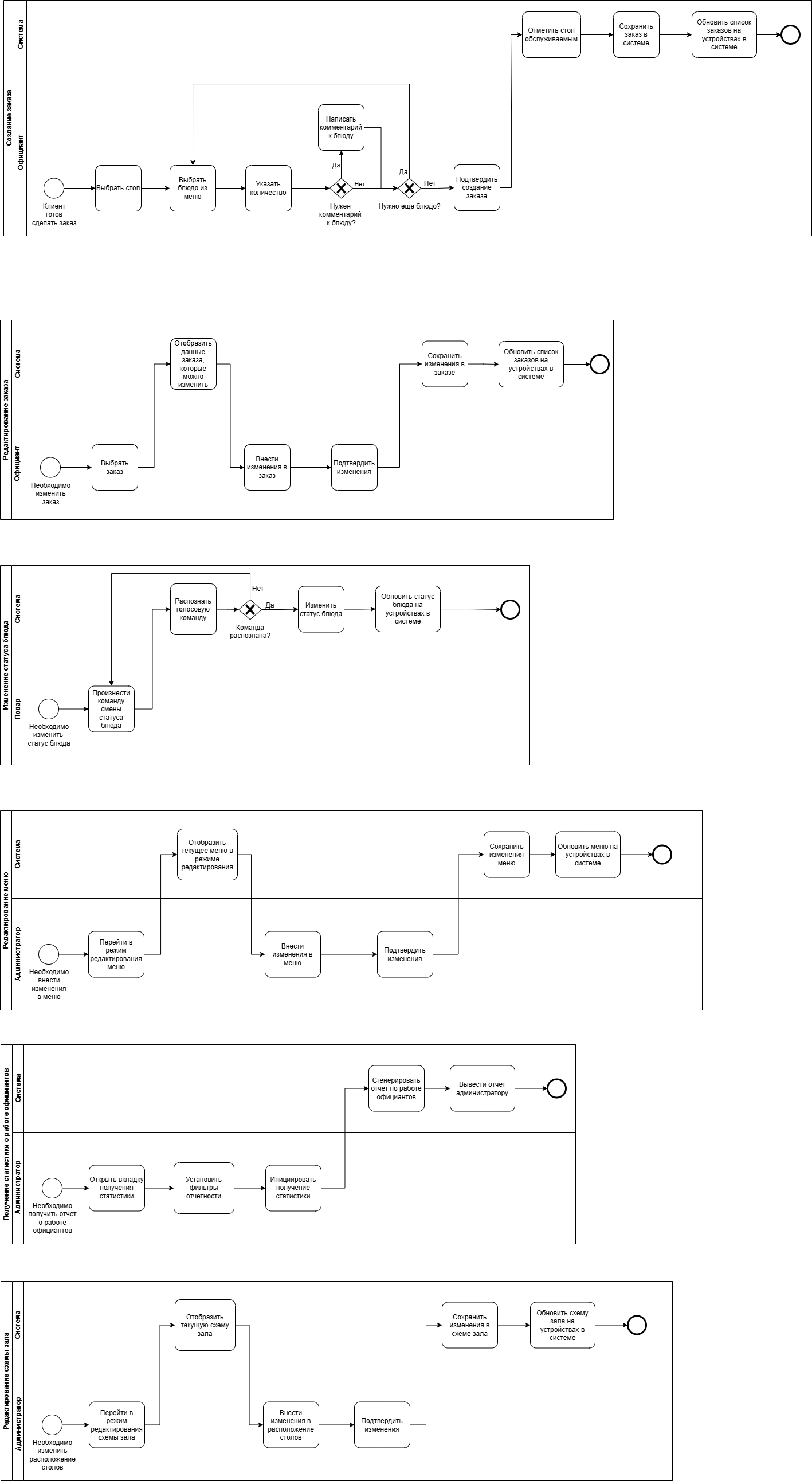


Рисунок 11 – Модель процесса редактирования схемы зала

1. Структура хранимой информации

На рисунке 12 представлена логическая схема базы данных для разрабатываемой системы в нотации IDEF1X.

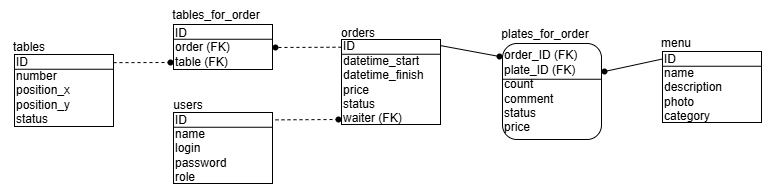


Рисунок 12 – Схема БД в нотации IDEF1X

Таблица tables в описанной БД предназначена для хранения информации о столах (поля ID и number), их расположении (поля position\_x и position\_y) и статусах (поле status). Таблица users хранит информацию о пользователях, включая их роль в системе (поле role) для проведения авторизации. Таблица orders позволяет хранить информацию о созданных официантами заказах (отслеживается по полю waiter), каждый заказ привязывается к одному или нескольким столам через сводную таблицу tables\_for\_order, поле price хранит стоимость заказа, в поле status указывается статус заказа (текущий/завершен), также отслеживается время начала и конца исполнения заказа. Хранение позиций и категорий меню организовано в таблице menu, где поле category с возможными значениями True/False, обеспечивает разделение блюд от их категорий. Блюда для каждого заказа сохраняются в сводной таблице plates\_for\_menu, также в данной таблице для каждого блюда указывается его количество (поле count), цена (поле price), статус приготовления (поле status) и комментарий к блюду (поле comment).

1. Архитектурно-структурное решение

Для реализации проекта была выбрана клиент-серверная модульная монолитная архитектура. Такой подход позволяет разделить монолитный код на модули с определенными границами, которые могут разрабатываться отдельно и отвечают за определенную область объекта автоматизации.

Система представляет собой единое приложение, содержащее следующие основные модули: модуль управления заказами, модуль работы с меню, модуль управления картой столов, модуль обработки голосовых команд, модуль аутентификации и авторизации, а также модуль отчетности и аналитики.

База данных реализована при помощи СУБД PostgreSQL с четко структурированной схемой, где каждая бизнес-область имеет свои таблицы и ограничения.

Мобильные клиенты для официантов, поваров и администраторов взаимодействуют с системой через единый API шлюз, который является частью монолитного приложения. Для коммуникации в реальном времени, в частности реализации уведомления о готовности заказов, используется WebSocket соединение, также интегрированное в основное приложение.

1. Особенности взаимодействия с пользователем

Требования к пользовательскому интерфейсу были предъявлены в техническом задании.