

1 Анализ объекта исследования

1.1 Характеристика ERP-системы iiko

Iiko представляет собой комплексное программное обеспечение для автоматизации организаций общественного питания, к примеру, ресторанов, столовых, кафе, кофеен и баров. Iiko позволяет вести учёт запасов готовой продукции и ингредиентов, управлять персоналом, настраивать систему лояльности, создавать отчёты, составлять прогноз деятельности предприятия и т. п. [1, 2]

Iiko состоит из таких компонентов, как iikoServer (центральная часть системы, реализующая основную бизнес-логику и координирующая взаимодействие других частей), iikoOffice (система управления торговым предприятием), iikoFront (кассовая система), iikoChain (система централизованного управления сетью предприятий) и др. Для хранения данных используется СУБД Microsoft SQL Server.

Iiko предоставляет API для интеграции с другим программным обеспечением. При создании информационной системы обеспечения линии раздачи возникла потребность в следующих возможностях API iiko: авторизации, получении номенклатуры и создании OLAP-отчётов о продажах [3].

1.2 Анализ бизнес-процесса обеспечения линии раздачи

Рассмотрим бизнес-процесс обеспечения линии раздачи необходимым количеством блюд. Участниками данного процесса являются администратор и повара.

Администратор в целях максимизации доходов предприятия должен в течение рабочего дня обеспечивать наличие на линии раздачи всех блюд. Кроме того, администратору необходимо предотвращать появление избытка блюд, так как тот может значительно увеличить издержки. Следовательно, администратор в течение смены вынужден периодически оценивать запасы готовой продукции, составлять заказ на приготовление блюд и передавать заказ поварам.

Повара после получения заказов приступают к приготовлению блюд. Выполнив заказ, повара в той или иной форме сообщают администратору о приготовленных блюдах.

Модель данного процесса в виде BPMN-диаграммы представлена на рисунке 1.1. Задачи администратора обозначены как циклический подпроцесс, так как заранее неизвестно, сколько раз понадобится проверять запасы. А задачи поваров по приготовлению блюд представлены в виде многоэкземплярного параллельного подпроцесса, поскольку одновременно на кухне могут готовиться несколько блюд и количество экземпляров процесса известно заранее (с момента поступления заказов).

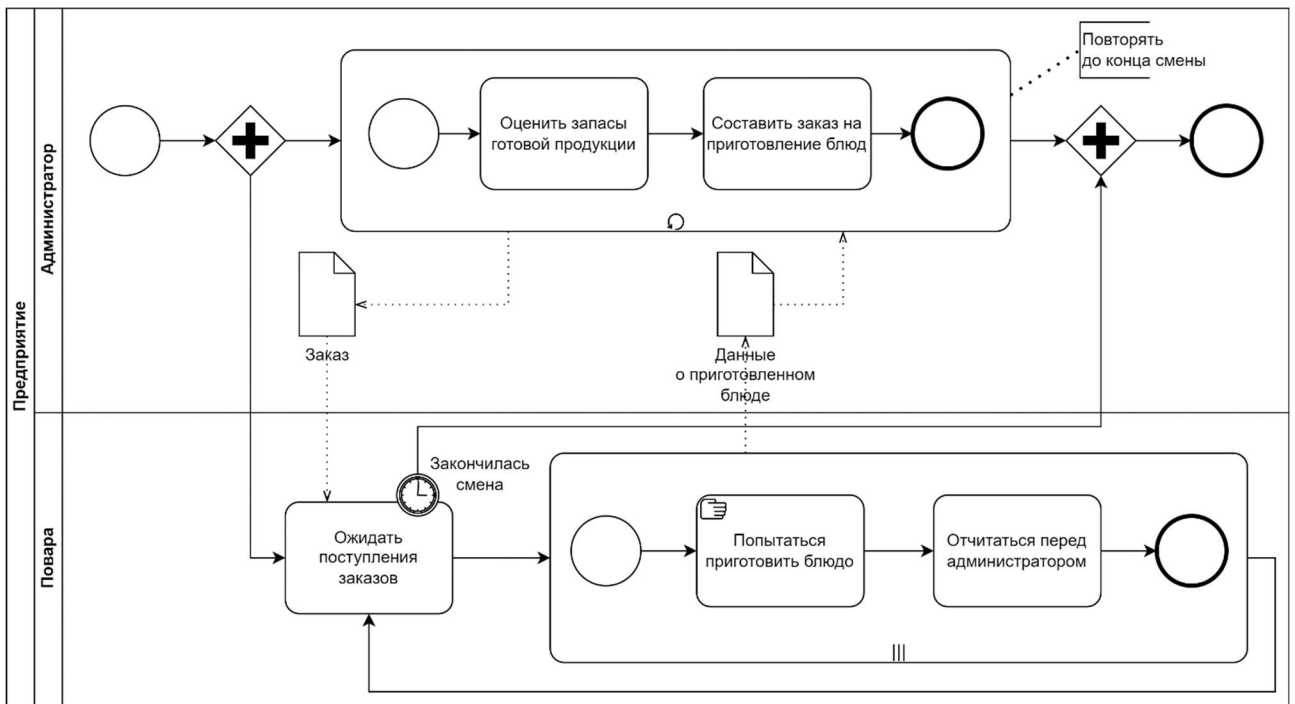


Рисунок 1.1. BPMN-диаграмма «как есть» процесса обеспечения линии раздачи

Критический взгляд на бизнес-процесс позволяет выявить такие его недостатки, как высокая нагрузка на администратора и неудовлетворительная коммуникация.

Действительно, администратор помимо прочих работ по непосредственному управлению предприятием должен время от времени

проверять запасы блюд на линии раздачи и при необходимости составлять заказы. Вследствие высокой трудоёмкости этих задач администратор не всегда может правильно рассчитать размер заказа, что приводит к уменьшению прибыли предприятия.

Суть другой проблемы – неудовлетворительной коммуникации – заключается в том, что повара вынуждены отвлекать администратора от текущих работ либо ждать, пока тот освободится, чтобы сообщить о нештатной ситуации, к примеру, о нехватке ингредиентов или невозможности приготовить заказ в срок. В целях улучшения коммуникации целесообразно добавление в разрабатываемую систему возможности отправки администраторам сообщений о проблемах на кухне.

Перечисленные слабые места бизнес-процесса можно исправить путём внедрения специализированной информационной системы. ООО «Денвик Мир» получил заказ на разработку такого продукта, и генеральный директор «Денвика» Петров Константин Юрьевич предложил автору работы создать эту систему. Требования к системе были представлены в виде заполненной заказчиком анкеты. На основе анкеты была создана модель бизнес-процесса обеспечения линии раздачи в случае внедрения разрабатываемого продукта. Соответствующая BPMN-диаграмма показана на рисунке 1.2.

Важным отличием обновлённой версии процесса является уменьшение нагрузки на администратора: теперь администратору достаточно в начале смены внести в систему сведения о запасах продукции и включить планировщик задач – компонент системы, который в определённое время запускает получение последних продаж и расчёт заказов поварам.

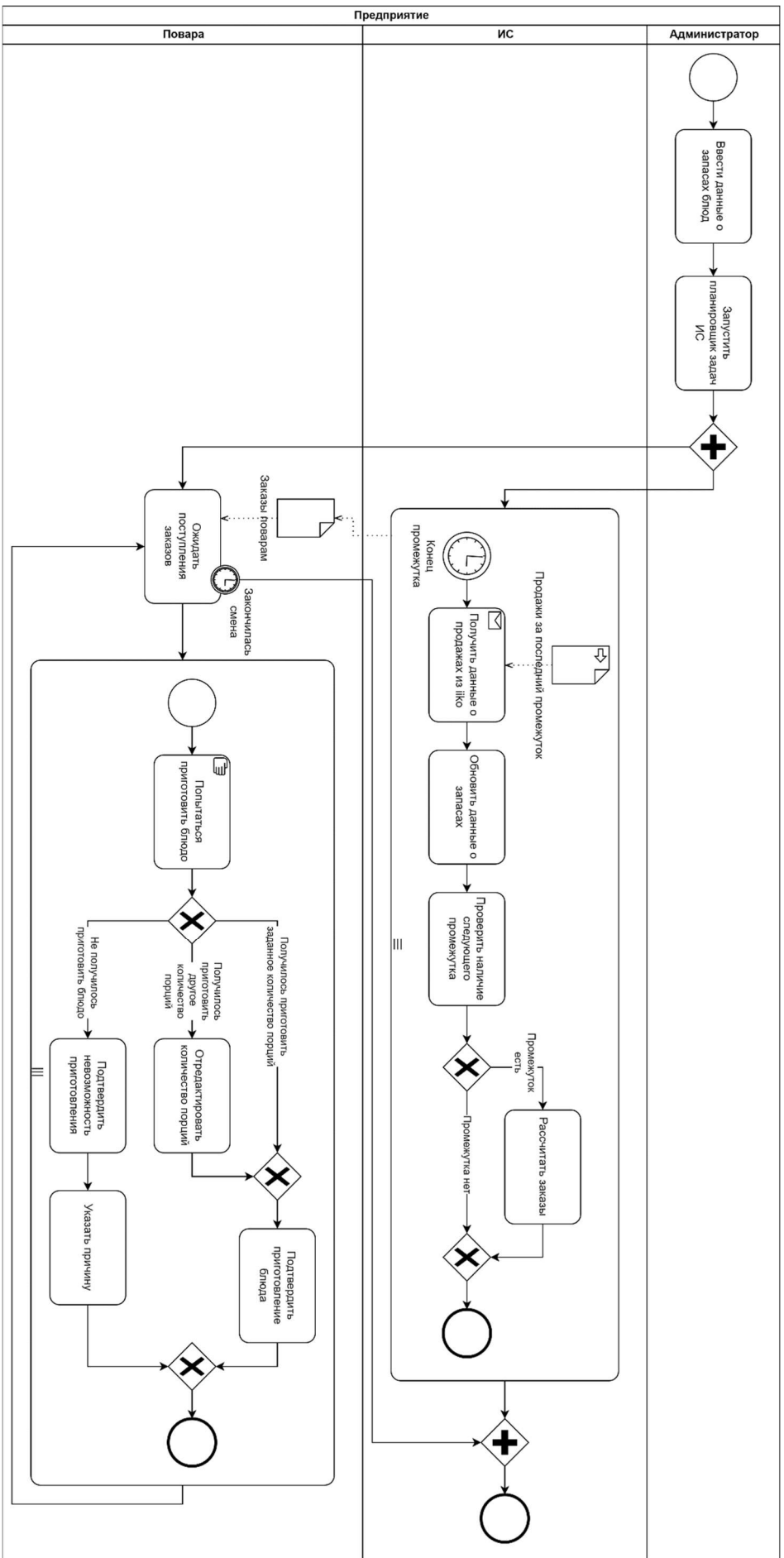


Рисунок 1.2. BPMN-диаграмма «как должно быть» процесса обеспечения линии раздачи

После включения планировщика в системе запускается последовательный многоэкземплярный подпроцесс. Началом каждой итерации подпроцесса служит конец временного промежутка смены (промежутки создаются в соответствии с настройками, заданными администратором). Первым шагом является получение из iiko данных о продажах за последний промежуток. Получив ответ от iiko, система обновляет в своей базе данных сведения о запасах продукции и, если есть следующий промежуток, рассчитывает заказы на приготовление блюд.

Для поваров бизнес-процесс начинается с ожидания заказов. Получив заказы, повара приступают к приготовлению блюд. Эту задачу, как и в предыдущей версии процесса, можно представить в форме параллельного многоэкземплярного подпроцесса. Каждая его итерация начинается непосредственно с попытки приготовления блюда. Далее повару достаточно лишь подтвердить результат, если удалось приготовить именно рекомендованное системой количество порций. Если на выходе получилось иное количество, повару нужно ввести это число и подтвердить приготовление блюда, чтобы в базе данных содержались достоверные сведения о запасах. Если по какой-то причине невозможно выполнить заказ, повару следует подтвердить невозможность приготовления и указать причину. Информация о случившемся будет отправлена администратору. После выполнения всех заказов повара вновь возвращаются к ожиданию поручений. При этом окончание смены прерывает ожидание и завершает бизнес-процесс с точки зрения поваров.

1.3 Постановка задачи на разработку

На основании пожеланий заказчика и проведённого анализа бизнес-процесса представляется целесообразной разработка информационной системы, удовлетворяющей приведённым ниже требованиям.

Во-первых, поварам в течение смены должен быть доступен список актуальных заказов. Повара получают возможность подтверждать приготовление блюда либо невозможность выполнения заказа. Поварам должен

быть доступен способ быстрого редактирования текущих остатков. Также поварам нужен сервис быстрой отправки сообщений администраторам.

Во-вторых, администраторы должны иметь возможность задавать настройки пополнения линии раздачи; создавать временные промежутки, по окончании которых система запрашивает продажи и рассчитывает заказы; управлять пользователями; импортировать список блюд из iiko и редактировать данные о блюдах; получать сообщения о невозможности выполнения заказов и о проблемах в системе.

Взаимодействие пользователя и системы осуществляется через веб-приложение и модуль отправки сообщений администраторам. Предпочтительной формой реализации последнего, по словам заказчика, является Telegram-бот.

1.4 Сравнение системы с модулем iikoShelfManager

Для обеспечения линии раздачи требуемым количеством блюд может быть использован встроенный в iiko модуль iikoShelfManager. Рассмотрим отличия данного модуля от разрабатываемой системы:

- 1) Модуль доступен по предназначенному для крупных ресторанных сетей тарифу iiko Enterprise, а система требует лишь наличия доступа к API iiko, потому доступна предприятиям, использующим тарифы Pro или Enterprise.
- 2) Модуль, в отличие от разрабатываемой системы, не позволяет задавать требуемое количество порций на линии раздачи.
- 3) Модуль не выдаёт заказы: повара должны сами решать, какие блюда следует готовить.
- 4) Модуль не содержит средств коммуникации между поварами и администраторами.
- 5) Для использования модуля требуется терминал с установленной программой iikoFront, а работа в системе возможна с любого устройства при наличии браузера.

б) Модуль позволяет отслеживать срок хранения блюд на линии раздачи [4, 5].

Таким образом, разрабатываемая система выгодно отличается от встроенного модуля бóльшим набором функций и менее строгими требованиями к тарифу iiko и оборудованию.

2 Описание технологии проектирования и программирования

2.1 Описание функциональности

Принимая во внимание результаты анализа бизнес-процесса обеспечения линии раздачи, целесообразно разделить пользователей системы на две категории: администраторы и повара. Рассмотрим набор функций, доступный каждой из групп.

В обязанности администратора входит настройка системы. Для корректной её работы требуется ввести данные о блюдах, временных промежутках и типах смен, создать пользователей и задать настройки пополнения.

Для удобства редактирования настроек блюд последние могут быть объединены в группы, по аналогии с иерархией номенклатуры в iiko. Администратор может как импортировать перечень блюд и групп из iiko, так и добавлять или удалять элементы номенклатуры вручную.

При редактировании информации о блюдах администратору нужно указать название, текущие запасы, единицу измерения, выбрать родительскую группу, режим пополнения (подробная информация о режимах в конце данного раздела) и запросить идентификатор блюда или группы из iiko. Поскольку зачастую нецелесообразно готовить малые объёмы блюд, администратор должен ввести ещё и кратность – показатель, от которого зависит размер заказа. Так, например, если в организации принято выдавать заказы на приготовление 5, 10, 15 и т. д. шанег за раз, то администратору следует выставить кратность 5 для этого блюда. Чтобы ускорить выполнение ежедневной рутинной процедуры ввода данных о запасах блюд к началу смены, администратор может указать начальные запасы блюд по умолчанию.

Ещё администратор должен создать временные промежутки, указав для каждого время начала и конца. Поскольку в разные дни ожидаемый объём продаж может быть неодинаковым, имеет смысл привязывать каждый временной промежуток к определённому типу смены. Единственным параметром типа смены, который должен указать администратор, является название.

После ввода информации о блюдах и временных промежутках администратор может задать настройки пополнения блюд, а именно указать, какое наименьшее и наибольшее приемлемое количество порций выбранного блюда может быть на линии раздачи в течение определённого временного промежутка.

Кроме того, администратор должен иметь возможность управлять пользователями информационной системы: менять имя в системе (логин), настоящее имя, адрес электронной почты, номер телефона, пароль и роли пользователя.

Ещё администратор может подписаться на получение от Telegram-бота сообщений о проблемах в системе (к примеру, об отрицательных остатках блюд после получения продаж) и о действиях поваров (об отказе готовить заказ и о приготовлении количества порций, отличного от требуемого). Благодаря боту администратор может мгновенно получать информацию о таких ситуациях и оперативно решать проблемы.

Наконец, администратор в начале каждой смены должен запускать планировщик задач. При этом от администратора требуется указать тип сегодняшней смены.

И повар, и администратор должны иметь возможность редактировать текущие запасы блюд. Это необходимо для того, чтобы система могла правильно определить остатки блюд после получения продаж и рассчитать актуальные заказы.

Также поварам и администраторам нужна функция получения списка заказов: первым – для приготовления блюд, вторым – для контроля правильности настройки системы.

Наконец, обе роли могут закрывать заказы. Для этого они могут либо подтвердить приготовление блюда, либо подтвердить невозможность приготовления. В первом случае пользователь может отредактировать количество приготовленных порций, чтобы в системе хранились корректные

сведения о запасах. Во втором – выбрать причину и (или) добавить комментарий для администраторов.

Предусмотрены различные варианты применения системы на кухне. Приложение могут одновременно использовать несколько поваров – каждый со своего устройства. Также возможен сценарий, при котором несколько поваров работают с одного устройства, к примеру, терминала или планшета. Наконец, систему может использовать и один шеф-повар, который распределяет заказы между подчинёнными.

Рассмотрим функции, которые информационная система выполняет относительно самостоятельно. Во-первых, после запуска планировщика задач система может получать из iiko продажи за последний промежуток в соответствии с расписанием, созданным планировщиком. Во-вторых, сразу после обработки полученного от iiko отчёта о продажах и уменьшения остатков на количество проданных порций приложение может рассчитать заказы на следующий период, если такой существует. В-третьих, система способна разослать сообщение о возникшей проблеме всем администраторам, оформившим подписку на уведомления в Telegram.

Описанную функциональность системы можно изобразить в виде диаграммы прецедентов, представленной на рисунке 2.1.

Далее рассмотрим, какие режимы пополнения должны быть реализованы в приложении. Первый вариант – режим «продажи». В таком случае размер заказа равен количеству порций, которое было продано за последний промежуток. Другой режим – «до максимума». Если выбран данный режим и запасы блюда меньше минимального количества, то система выдаст заказ приготовить такое количество порций, чтобы после приготовления запасы были как можно больше, но не превышали максимальное количество. При этом размер заказа должен быть кратен показателю «кратность» данного блюда. Таким образом, если выбран режим «до максимума», размер заказа может быть рассчитан по формуле (1). Расчёт размера заказа по данной формуле возможен, если выполняется неравенство (2).

$$\text{заказ} = \left\lfloor \frac{\text{максимум} - \text{запасы}}{\text{кратность}} \right\rfloor \times \text{кратность} \quad (1)$$

$$\text{кратность} \leq \text{максимум} - \text{минимум} \quad (2)$$



Рисунок 2.1. Диаграмма прецедентов с описанием функциональности системы