## Введение

Программный продукт "Kitchen Boss" предназначен для автоматизации бизнес-процессов ресторана. Приложение позволяет управлять списками клиентов, блюд, сотрудников, заказов и столов, а также обеспечивает контроль над действиями пользователей системы. Объектом применения является ресторан или кафе, где требуется эффективное управление заказами и персоналом. Система стремится оптимизировать операции, уменьшить количество ошибок и повысить общую эффективность работы заведения.

## 1 Основания для разработки

Разработка ведётся на основании задания к курсовому проекту по профессиональному модулю ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем» МДК 01.01 «Разработка программных модулей».

## 2 Назначение разработки

Функциональное назначение:

* Управление клиентами, блюдами, сотрудниками, заказами и столами.
* Контроль статусов заказов.
* Разграничение прав доступа для сотрудников.
* Автоматизация типовых операций в бизнес-процессах, таких как расчет суммы заказа и формирование отчетов.
* Обеспечение удобного и интуитивно понятного интерфейса для персонала ресторана.

Эксплуатационное назначение:

* Повышение эффективности работы ресторана за счёт автоматизации бизнес-процессов.
* Уменьшение количества ошибок при оформлении заказов и обслуживании клиентов.
* Улучшение качества обслуживания клиентов за счет более быстрого и точного выполнения заказов.
* Оптимизация управления ресурсами ресторана (например, управление складскими запасами, бронирование посадочных мест и т.п.).
* Предоставление руководству ресторана инструментов для анализа и принятия решений на основе данных (например, отчеты о продажах).

## 3 Требования к программе

### 3.1 Требования к функциональным характеристикам

Данный программный продукт должен удовлетворять следующим функциональным требованиям:

* Авторизацию пользователей с разграничением прав (менеджер, официант, шеф-повар). Для каждого типа пользователей должны быть определены следующие права доступа:
  1. **Менеджер:** Полный доступ к управлению всеми аспектами системы, включая клиентов, сотрудников, меню, заказы, столы и отчеты.
  2. **Официант:** Возможность принимать заказы, управлять столами, просматривать меню и статусы заказов.
  3. **Шеф-повар:** Возможность просматривать заказы и изменять их статус (например, "Готовится", "Готов").
* Просмотр и редактирование данных о клиентах, блюдах, сотрудниках, заказах и столах.
* Управление категориями блюд и ингредиентами.
* Контроль статусов заказов (Принят, Готовится, Готов, Подается, Закрыт, Отменен). Система должна обеспечивать возможность отслеживания изменений статуса заказа и уведомления заинтересованных сторон (например, официанта при готовности блюда).
* Автоматический расчёт суммы заказа, включая возможность применения скидок и налогов (если применимо).
* Формирование отчетов:
  1. Отчеты о продажах за определенный период.
  2. Отчеты по заказам (например, по официантам, по столам).
  3. Отчеты по клиентам (например, история заказов).
  4. Отчеты по сотрудникам (например, производительность).
* Пользовательский интерфейс должен быть интуитивно понятным и удобным в использовании для персонала с различным уровнем компьютерной грамотности.
* Система должна обеспечивать возможность быстрого поиска и фильтрации данных (например, поиск клиента по имени, поиск блюда по названию).
* Поддержка работы с несколькими заказами одновременно (для официантов).
* Возможность добавления комментариев к заказу (например, особые пожелания клиента).

Входные данные:

* Данные о клиентах: ФИО, контактные данные, история заказов.
* Данные о блюдах: название, описание, категория, цена, ингредиенты.
* Данные о сотрудниках: ФИО, должность, дата приема на работу, зарплата, контактные данные, учетные данные для входа в систему.
* Данные о заказах: список заказанных блюд, количество, стол, официант, статус заказа, время создания, время выполнения.
* Данные о столах: номер стола, вместимость, расположение.
* Учётные данные пользователей: логин, пароль, роль.
* Данные об ингредиентах: название, ед. измерения, количество на складе.

Выходные данные:

* Отчёты о заказах, клиентах и сотрудниках.
* Общая сумма заказов.
* Подтверждения о создании, изменении и удалении данных.
* Уведомления о статусах заказов.
* Печатные формы заказов и счетов.

### 3.2 Требования к надёжности

Обязательные требования:

* Обеспечение устойчивой работы приложения при корректных входных данных. Система должна обрабатывать исключения и ошибки, не приводя к аварийному завершению работы.
* Контроль целостности данных в БД. Должны быть реализованы механизмы обеспечения целостности данных, такие как транзакции, ограничения (constraints) и проверки входных данных.
* Резервное копирование данных: Предусмотреть ручное резервное копирование базы данных.
* Журналирование событий: Вести журнал всех важных событий в системе, таких как вход пользователей в систему, создание и изменение заказов, для целей аудита и отладки.
* Отказоустойчивость: Обеспечить минимальное время простоя системы в случае сбоев оборудования или программного обеспечения.

### 3.3 Требования к информационной и программной совместимости

* Наличие СУБД: MS SQL Server 18.
* Наличие .NET Framework 4.7.2.
* Операционная система: Windows 10 или более поздняя.
* Требования к аппаратному обеспечению:
  + Процессор: Core i5
  + Оперативной памяти: 8 ГБ
  + Дисковое пространство: 30 ГБ

## 4 Требования к программной документации

В процессе проектирования должны быть разработаны следующие документы:

* Пояснительная записка (с необходимыми приложениями).

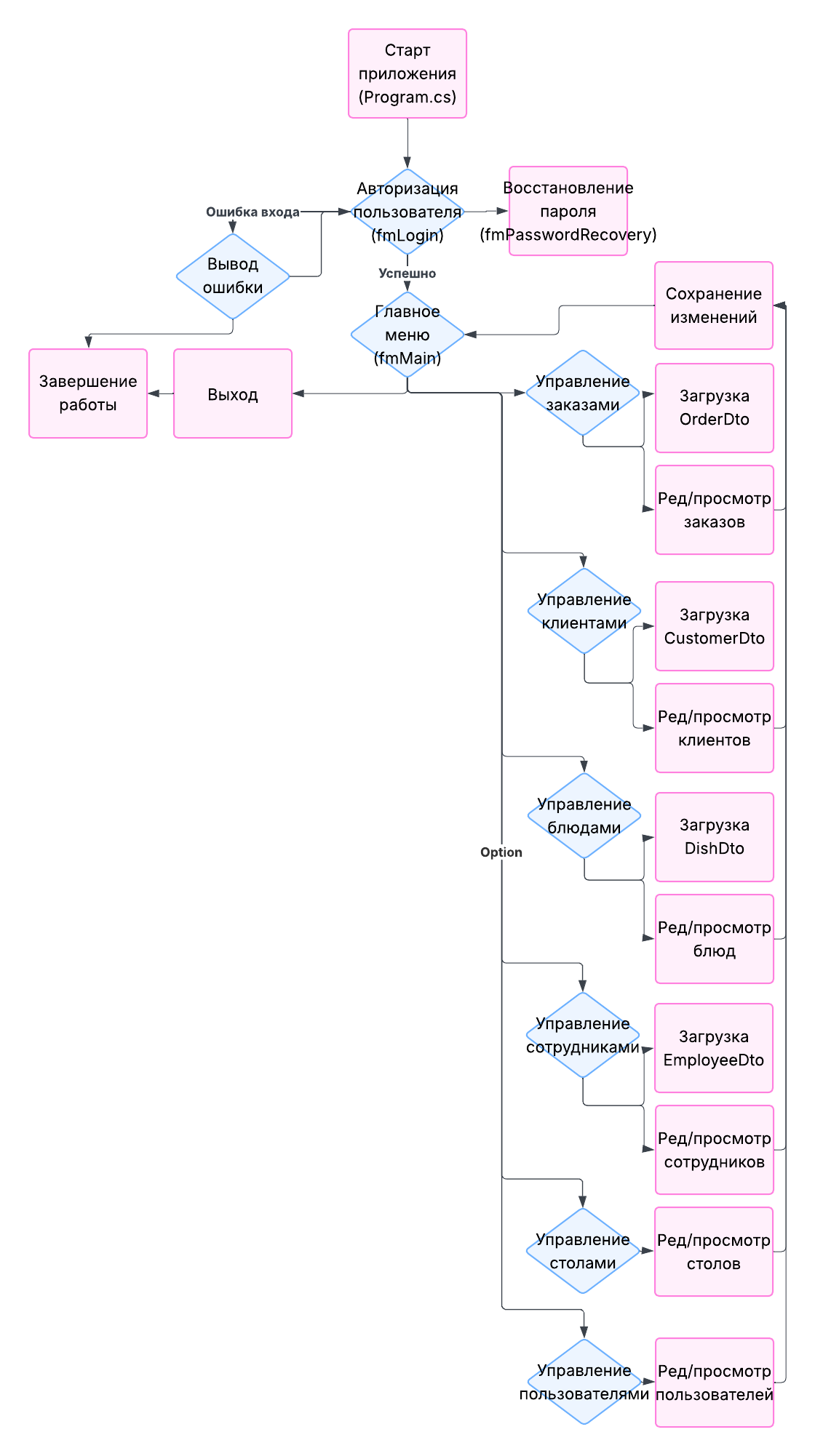
## 5 Перечень графических материалов

Для данной разработки следует подготовить следующие графические материалы:

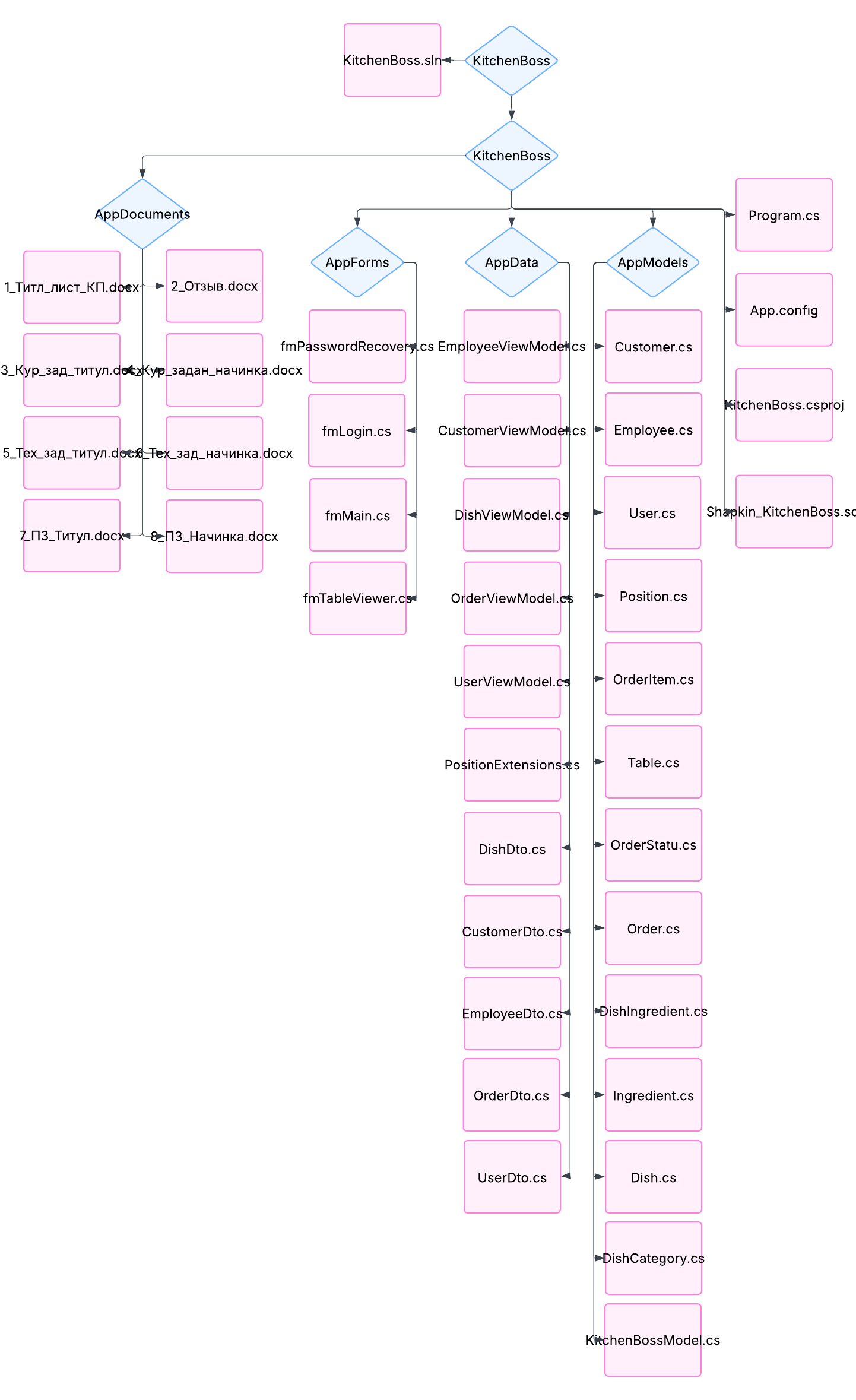
* блок-схема работы приложения;
* структура программы;
* функциональную схему программы;
* диаграмма деятельности;
* диаграмма вариантов использования;
* диаграмма переходов состояний;
* диаграмма классов;
* диаграмма базы данных.

## 6 Графические материалы

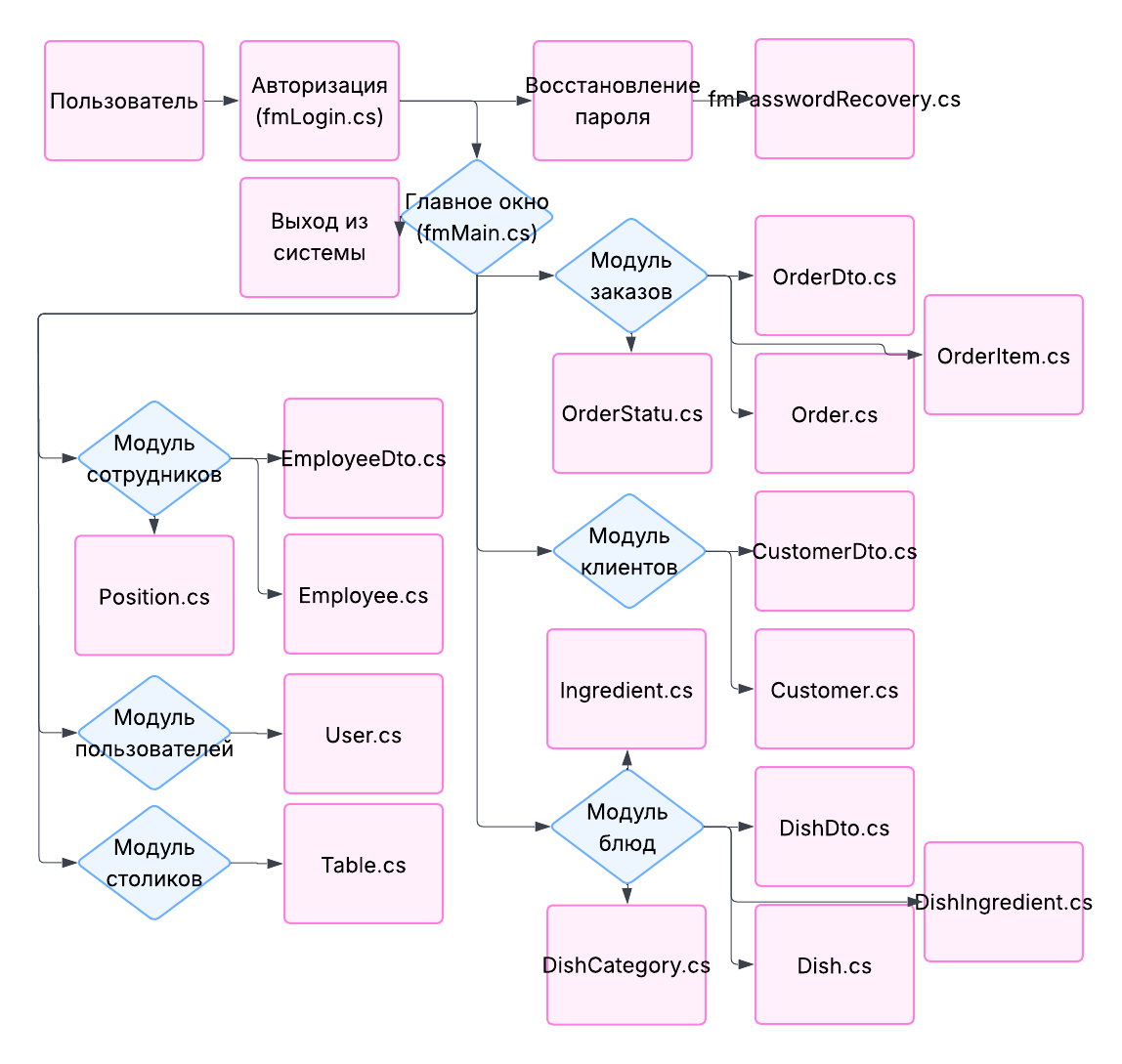
### 6.1 Блок-схема работы приложения Блок начала/конца = овал + фикс диаграммы в целом (в блок условия вход сверху выход снизу)



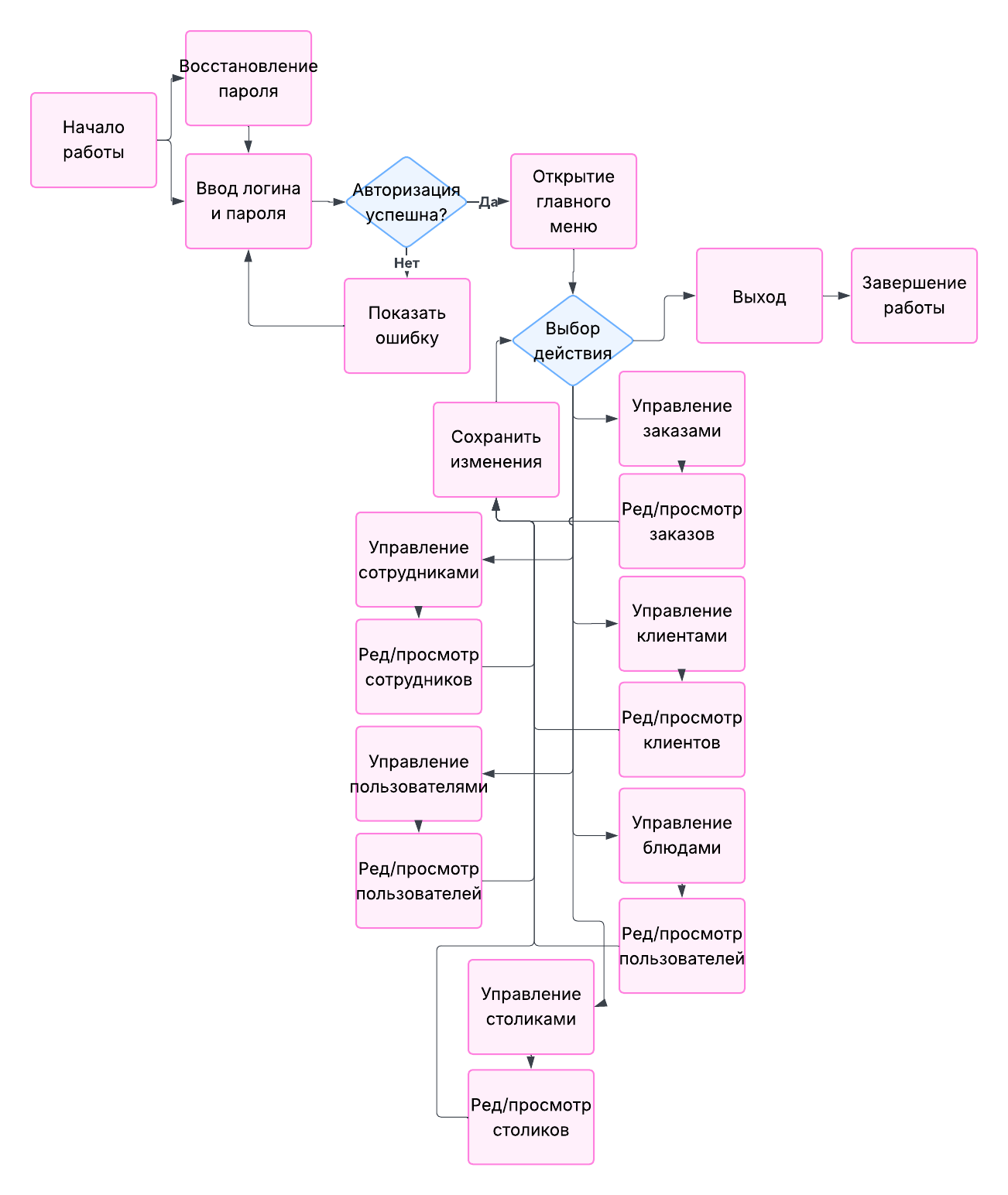
### 6.2 Структура программы Фикс диаграммы в целом (в блок условия вход сверху выход снизу)



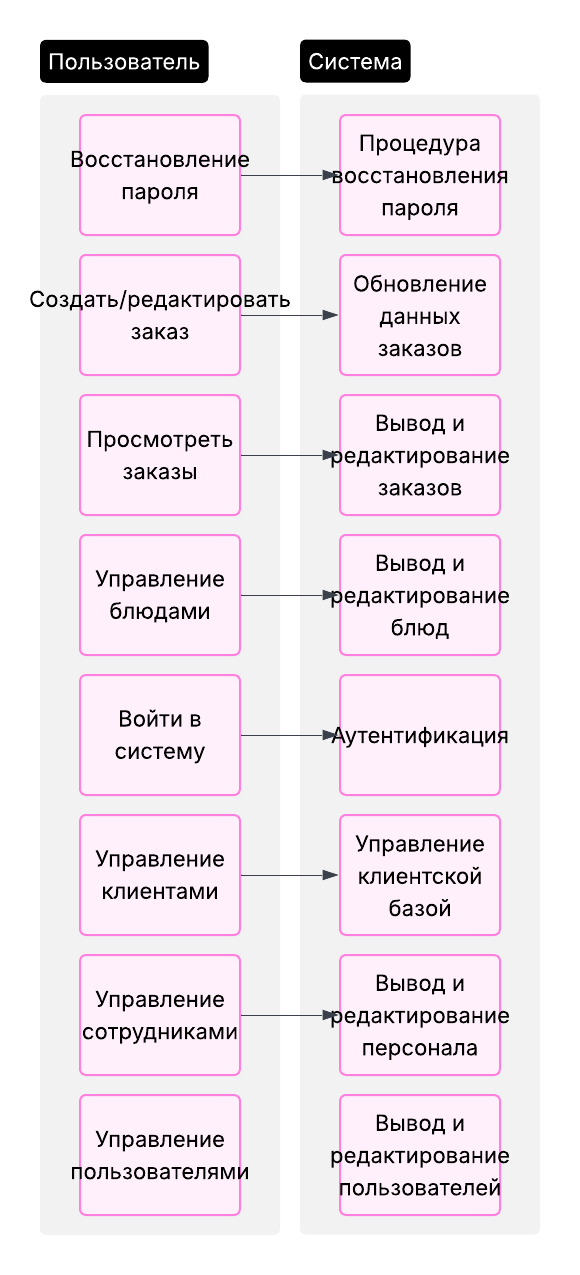
### 6.3 Функциональная схема программы Фикс диаграммы в целом (в блок условия вход сверху выход снизу)



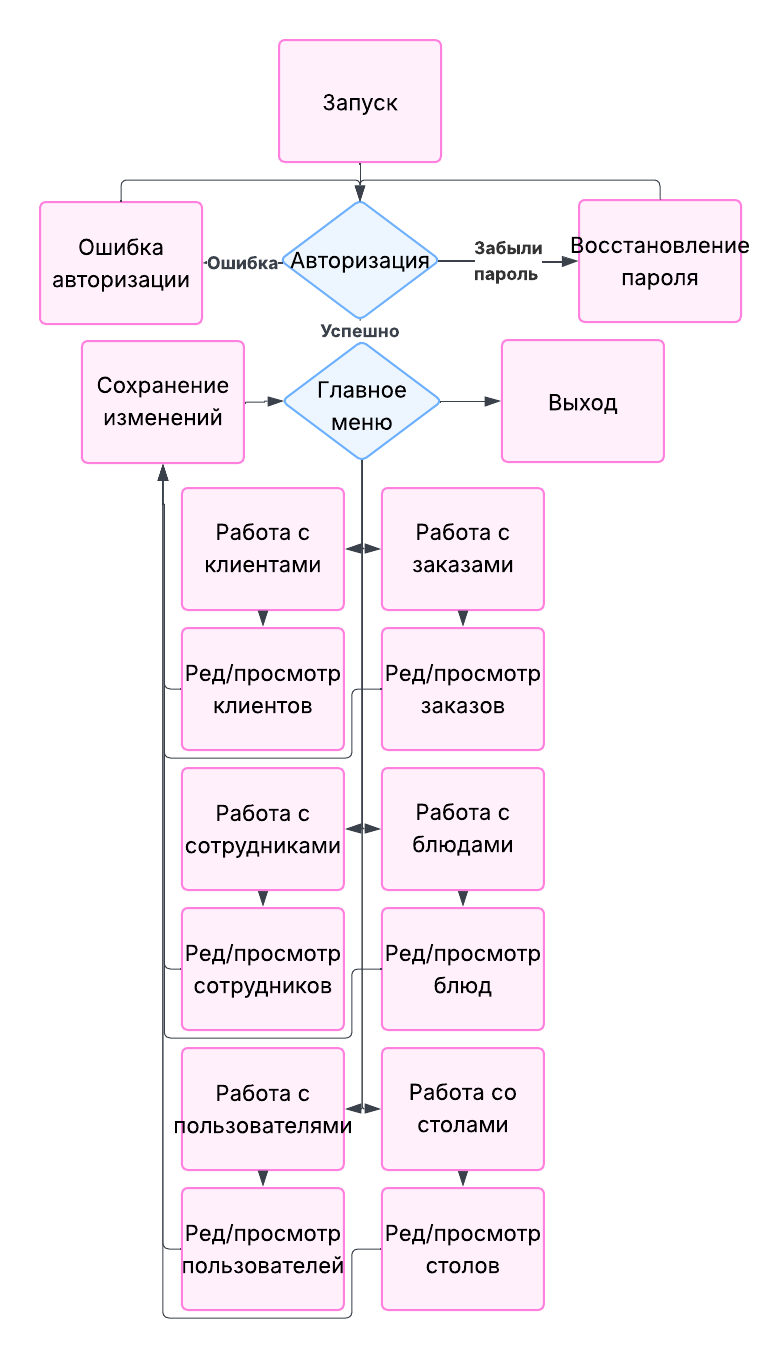
### 6.4 Диаграмма деятельности Фикс диаграммы в целом (в блок условия вход сверху выход снизу + блок начала/конца должен быть закрашенной точкой + должны быть “шины”)



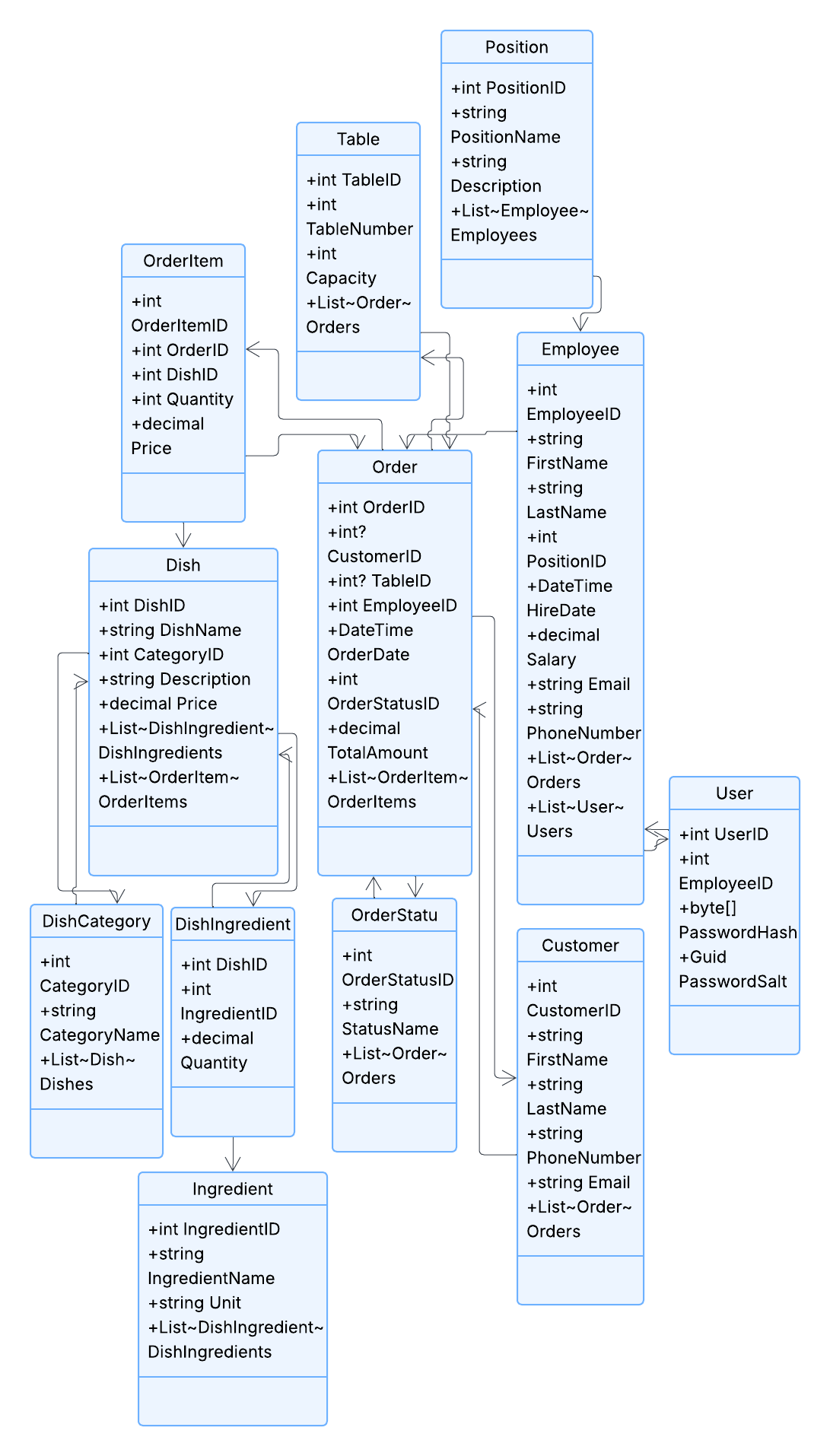
### 6.5 Диаграмма вариантов использования Фикс диаграммы в целом (должны быть человечки и овалы)



### 6.6 Диаграмма переходов состояний Фикс диаграммы в целом



### 6.7 Диаграмма классов Фикс диаграммы в целом (см. википедию)



### 6.8 Диаграмма базы данных Фикс диаграммы в целом (нет обозначения ключей + должны быть норм стрелочки, а не только выколотые)

