# **3. СТАДИЯ «ТЕХНИЧЕСКИЙ ПРОЕКТ»**

На стадии технического проекта проводится объектная декомпозиция системы, результаты которой представляются UML диаграммами классов ее компонентов.

## *3.1. Разработка диаграмм концептуальных классов*

На этапе разработки диаграмм концептуальных классов проведем объектную декомпозиция подсистемы хранения данных системы Gallery. Диаграмма концептуальных классов (рис. 3.1) подсистемы хранения данных представляет структурную модель, описывающую управление пользователями, отделами, категориями и транзакциями.

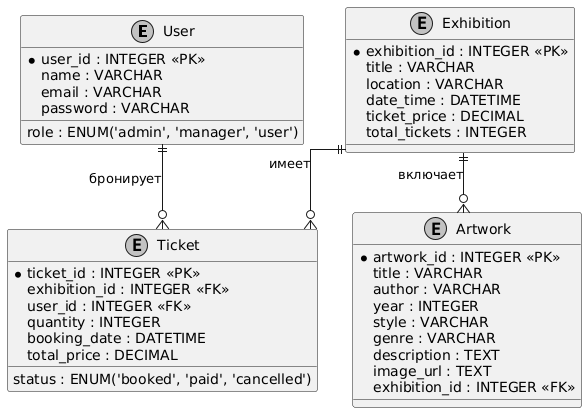


Рисунок 3.1. Диаграмма концептуальных классов подсистемы хранения данных

Объекты класса «User» связаны с объектами класса *«Ticket»* через атрибут *user\_id* в классе *«Ticket»*. Это позволяет определить, какие билеты забронировал конкретный пользователь. Связь один-ко-многим означает, что один пользователь может забронировать несколько билетов.

Объекты класса *«Exhibition»* связаны с объектами класса *«Ticket»* через атрибут *exhibition\_id* в классе *«Ticket»*. Это указывает, на какую выставку забронированы билеты. Связь один-ко-многим означает, что одна выставка может быть связана с множеством билетов.

Объекты класса *«Exhibition»* связаны с объектами класса *«Artwork»* через атрибут *exhibition\_id* в классе *«Artwork»*. Это позволяет определить, какие картины представлены на выставке. Связь один-ко-многим означает, что одна выставка может включать множество картин, но каждая картина принадлежит только одной выставке (или не принадлежит ни одной, если хранится в фондах).

Модель поддерживает управление пользователями (администраторы, менеджеры, пользователи) через атрибут *role*. Менеджеры формируют выставки, привязывая к ним картины (каждая картина — только на одной выставке). Пользователи бронируют билеты, а система автоматически рассчитывает стоимость (*total\_price*) и отслеживает статус (*booked, paid, cancelled*).

Данные о бронированиях позволяют анализировать доходность выставок, контролировать доступные билеты (*total\_tickets*) и предоставлять пользователям информацию о картинах и выставках.

## 3.2*. Разработка диаграмм программных классов*

На рисунке 8 представлена диаграмма классов системы, разработанная в соответствии с диаграммой пакетов (рис. 2) и множеством диаграмм вариантов использования (рис. 3 - 6). В отличие от рассмотренных ранее диаграмм концептуальных классов, эта диаграмма содержит так называемые программные классы, в которых определены не только атрибуты, но и методы — операции классов.

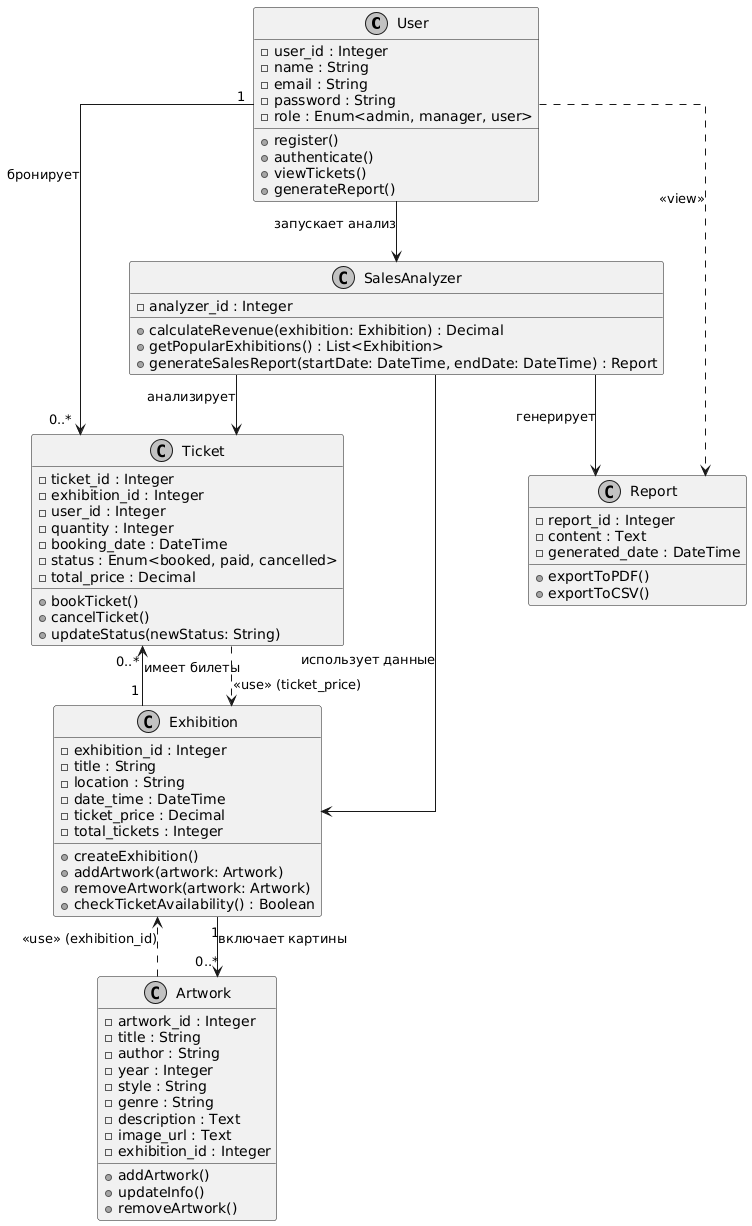


Рисунок 8. Диаграмма классов системы

Класс User управляет учетными записями пользователей (администраторы, менеджеры, посетители) и предоставляет методы для регистрации, аутентификации и просмотра билетов. Администраторы и менеджеры также могут генерировать отчеты через метод generateReport().

Класс Artwork хранит данные о картинах: название, автор, год создания, стиль, жанр, описание и ссылку на изображение. Каждая картина может быть привязана только к одной выставке через атрибут exhibition\_id.

Класс Exhibition содержит информацию о выставках: название, место, дату, стоимость билетов и общее количество билетов. Методы класса позволяют создавать выставки, добавлять/удалять картины и проверять доступность билетов.

Класс Ticket отвечает за бронирование билетов. Он автоматически рассчитывает общую стоимость (total\_price) на основе цены билета и количества, а также отслеживает статус бронирования (booked, paid, cancelled).

Классы SalesAnalyzer и Report обеспечивают анализ продаж. SalesAnalyzer вычисляет выручку, определяет популярные выставки и формирует отчеты, которые сохраняются в классе Report и могут экспортироваться в PDF или CSV.

Пользователи бронируют билеты через класс Ticket, который связан с выставкой (Exhibition) и пользователем (User). Выставки включают картины (Artwork), гарантируя, что оригинал представлен только на одной выставке. Анализ продаж выполняется классом SalesAnalyzer, использующим данные о билетах и выставках для генерации отчетов.

## 3.3*. Разработка диаграмм состояний*

Рассмотренные выше диаграммы классов представляют статические модели компонентов проектируемой системы и отражают результаты её объектной декомпозиции. Перейдем к реализации динамических моделей логического уровня. Так как система ориентирована на три различные группы пользователей (каждая из групп имеет свои возможности), рассмотрим каждую из них отдельно.

Первое из целевых состояний (рис. 9) «Authentication» завершается переходом альтернативным переходом в случае, если аутентификация пользователя завершена успешно, он получает доступ к системе переходит в следующее целевое состояние «Main\_Menu», в противном случае — сразу переходит в конечное состояние.

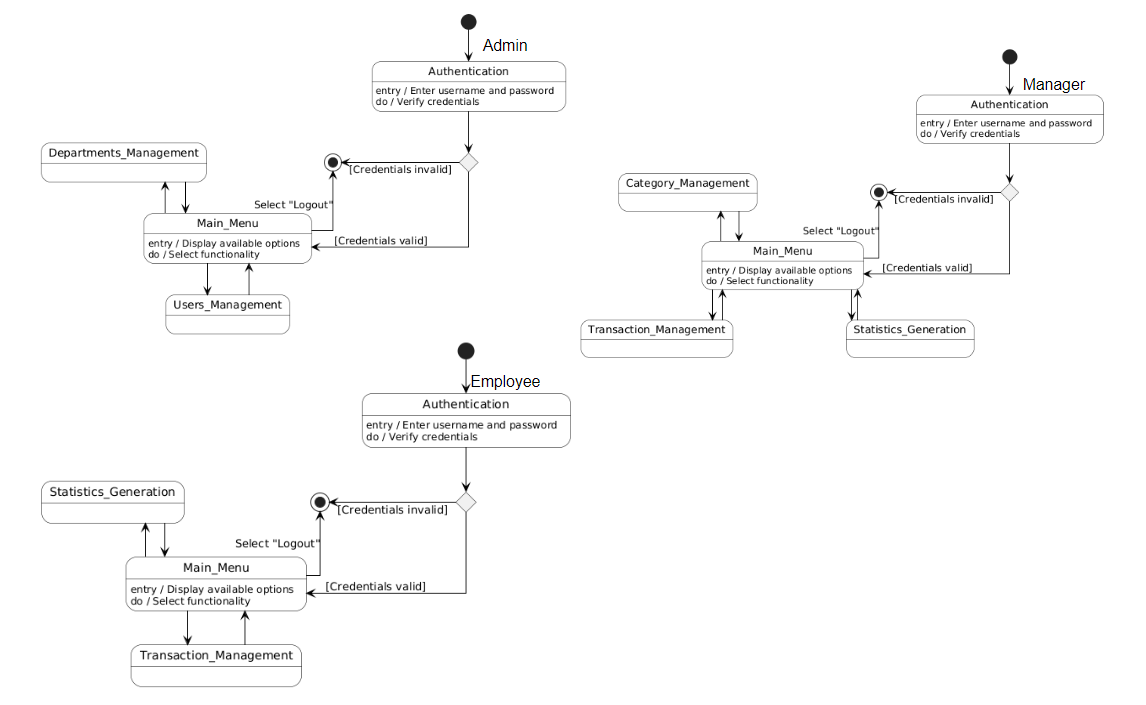


Рисунок 9. Обобщенная диаграмма состояний

В состоянии «Main\_Menu» выполняется входное действие – вывод на экран меню выбора доступных функций и последующее действие, связанное с ожиданием выбора и переходом к выбранной функции. В зависимости от роли пользователя доступен ограниченный выбор функций. Рассмотрим подробнее некоторые ключевые состояния.

Состояние «Statistics\_Generation» доступно сотрудникам и менеджерам. Рассмотрим каждое из состояний входящих в это состояние (рис. 10). Первое целевое состояние «Parameter\_Selection» представляет этап, на котором пользователь задаёт параметры для генерации статистического отчёта. Входное действие (entry): Система выполняет загрузку доступных фильтров, таких как временной диапазон, категории транзакций или типы операций, которые пользователь может выбрать для формирования отчёта. Текущее действие (do): Пользователь задаёт фильтры для отчёта, выбирая необходимые параметры из предложенных системой.

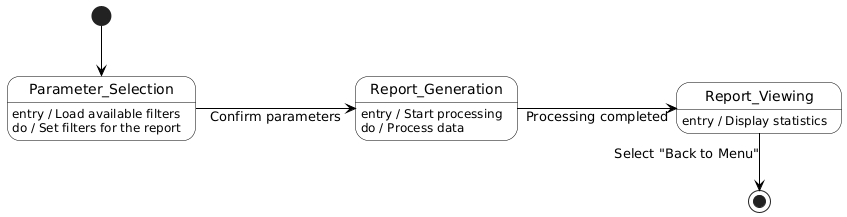


Рисунок 10. Диаграмма состояний блока «Main\_Menu»

После того как пользователь подтверждает выбранные параметры, система переходит в следующее состояние «Report\_Generation».

Состояние «Report\_Generation» описывает процесс обработки данных на основе заданных пользователем параметров. Система инициирует обработку данных, а после выполняется обработка данных, в ходе которой система агрегирует транзакции и формирует отчёт. После завершения обработки система переходит в состояние «Report\_Viewing».

Состояние «Report\_Viewing» представляет этап, на котором пользователь получает сгенерированный отчёт.

Рассмотрим переход из состояния «Main\_Menu» в состояние «Transaction\_Management» (рис. 11). Первое целевое состояние «Viewing\_Transactions» представляет этап, на котором пользователь просматривает список транзакций.

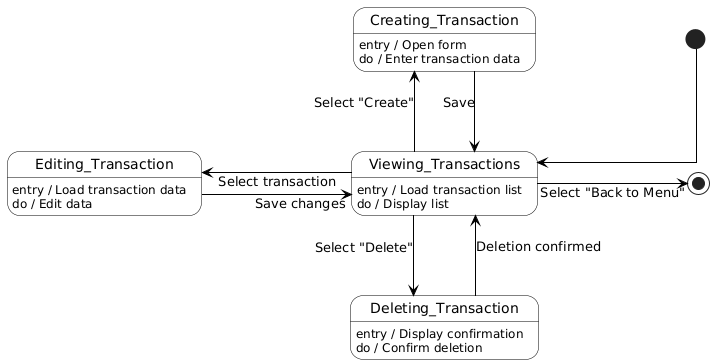


Рисунок 11. Диаграмма состояний блока «Transaction\_Management»

Входное действие: Система выполняет загрузку списка транзакций  
Текущее действие: Система отображает список транзакций для просмотра пользователем.

Из этого состояния возможны несколько переходов в зависимости от действий пользователя: создание новой транзакции, редактирование существующей, удаление транзакции или возврат в главное меню.

Состояние «Viewing\_Transactions» переходит в состояние «Creating\_Transaction», если пользователь выбирает создание новой транзакции. Система открывает форму для создания транзакции, пользователь вводит данные транзакции.После завершения ввода данных пользователь сохраняет транзакцию, и система возвращается в состояние «Viewing\_Transactions».

Состояние «Viewing\_Transactions» также переходит в состояние «Editing\_Transaction», если пользователь выбирает редактирование существующей транзакции. Система загружает данные выбранной транзакции, пользователь редактирует данные. После завершения редактирования пользователь сохраняет изменения, и система возвращается в состояние «Viewing\_Transactions».

Состояние «Viewing\_Transactions» переходит в состояние «Deleting\_Transaction», если пользователь выбирает удаление транзакции. Система отображает запрос на подтверждение удаления, пользователь подтверждает удаление (Confirm deletion).  
После подтверждения удаления система возвращается в состояние «Viewing\_Transactions».

Состояние «Viewing\_Transactions» завершает процесс, если пользователь выбирает возврат в главное меню, и система переходит в конечное состояние.

Рассмотрим переход из состояния «Main\_Menu» в состояние «Categories\_Management» (рис. 12), доступное только менеджерам. Включает в себя состояния аналогичные описанным выше состояниям работы с транзакциями.

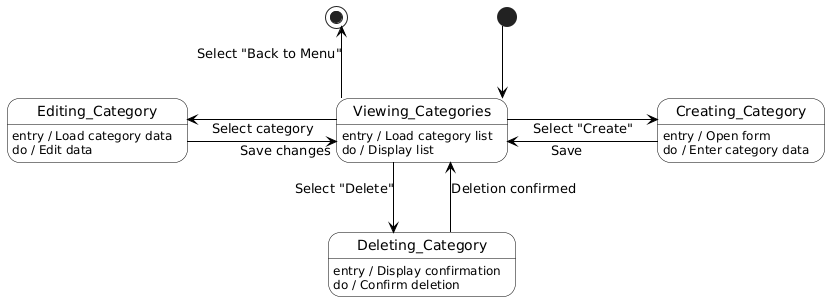


Рисунок 12. Диаграмма состояний блока «Categories \_Management»

Рассмотрим переход из состояния «Main\_Menu» в состояние «Departments\_Management» (рис. 13), доступное только администраторам системы. Включает в себя состояния аналогичные описанным выше состояниям работы с транзакциями.

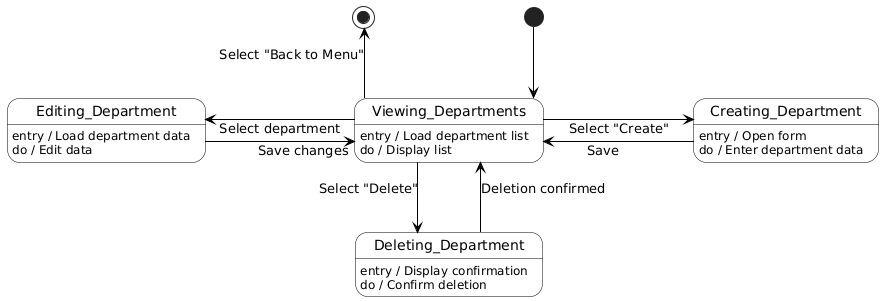


Рисунок 13. Диаграмма состояний блока «Departments \_Management»

Рассмотрим переход из состояния «Main\_Menu» в состояние «Users\_Management» (рис. 14), доступное только администраторам системы. Включает в себя состояния аналогичные описанным выше состояниям работы с транзакциями.

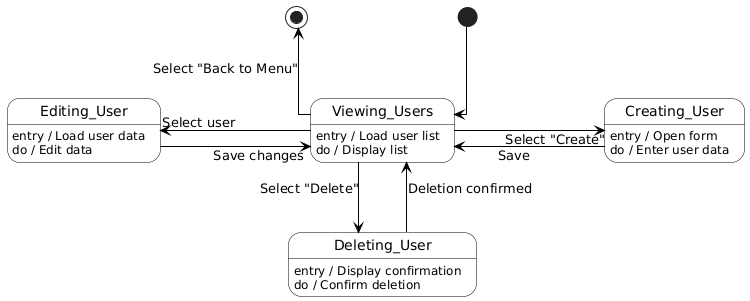


Рисунок 14. Диаграмма состояний блока «Users \_Management»