**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

Факультет физико-математических и естественных наук

Кафедра прикладная информатика

**РЕФЕРАТ**

**на тему**

Разработка модели информационной системы касс железнодорожного вокзала на основе UML

09.03.03 — прикладная информатика

Выполнил Студент группы НПИбд-01-21

Студенческий билет № 1032215163

Абд эль хай Мохамад

«  27 »      Февраля 2024г.

Москва 2024

**Оглавление**

[Глава 1. Какие UML-диаграммы? 3](#_Toc1)

[Глава 2. Диаграмма Class UML 3](#_Toc2)

[**Классы железнодорожной системы бронирования. Диаграмма классов:** 3](#_Toc3)

[**Классы и их атрибуты диаграммы классов системы бронирования железных дорог:** 4](#_Toc4)

[**Классы и их методы Диаграммы классов системы бронирования железнодорожных билетов:** 4](#_Toc5)

# Глава 1. Какие UML-диаграммы?

Я решил реализовать диаграмму вариантов использования и диаграмму классов для создания модели системы.

Диаграммы Use case помогают идентифицировать различных участников (таких как кассир, клиенты, гости и администраторы) и их взаимодействие с системой. Это поможет понять общий функционал и требования к информационной системе кассы.

Диаграмма Class помогает идентифицировать различные классы (такие как «Заявка», «Клиент», «План», «Администратор» и т. д.), а также их отношения, атрибуты и методы. Это может предоставить подробное представление о структуре системы и помочь в понимании различных объектов и их взаимодействий внутри системы.

# Глава 2. Диаграмма Class UML

Я решил внедрить систему, похожую на ту, которой я пользуюсь ежедневно, например, информационную систему касс московского метрополитена. Обычно у меня может быть карта (тройка, социальная карта), и в этом случае я могу либо подписаться на план, либо пополнить карту.

Оба действия поддерживают самообслуживание, что означает, что я могу либо сделать это самостоятельно, либо пойти к кассиру и позволить ему/ей сделать это за меня. Другой вариант: если у меня нет карты, я могу купить разовый билет на метро. И именно поэтому я создал гостя и пользователя-клиента.

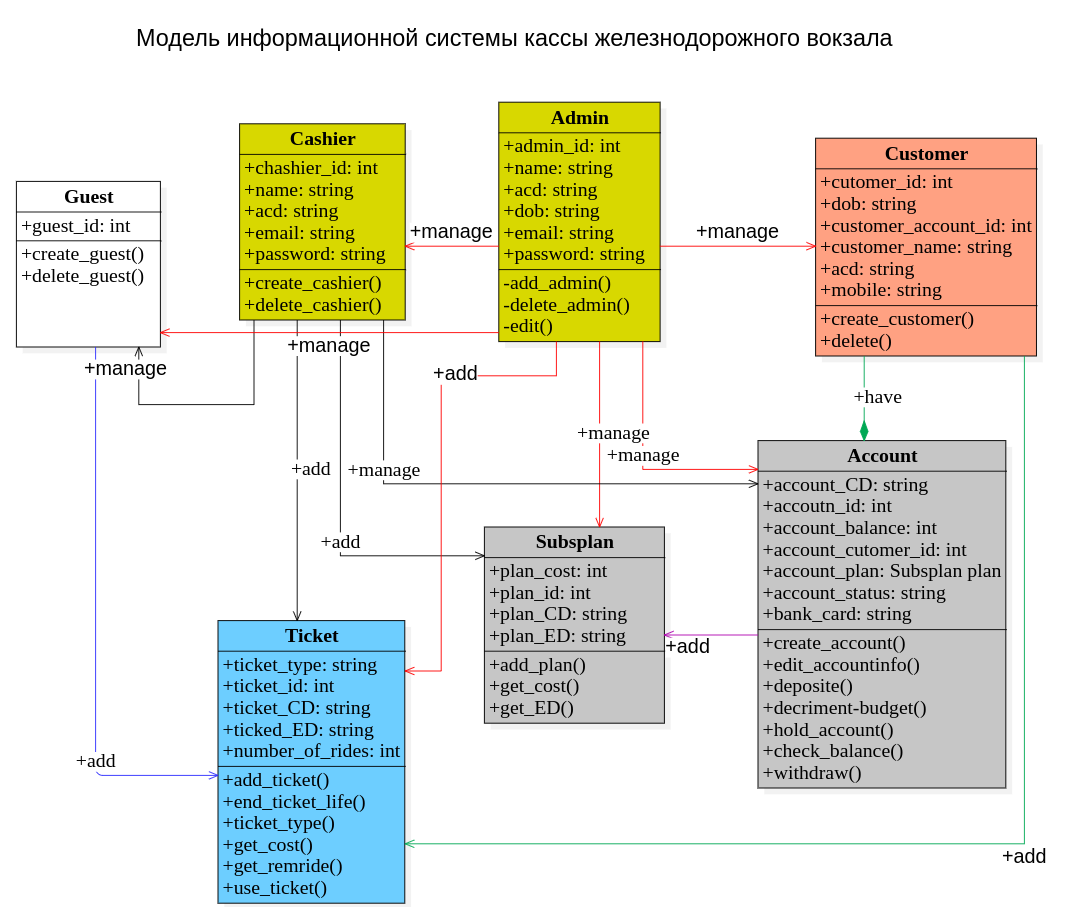
Гостю придется один раз зайти в систему, запросить билет и оплатить его. Клиент может это сделать, но, кроме того, он может подписаться на план.

### 2.1 Классы и их атрибуты диаграммы классов системы бронирования железных дорог:

* **Ticket Class:** ticket\_type, ticket\_id, ticked\_ED,ticket\_CD
* **Plan Class:** plan\_cost: int, plan\_id: int, plan\_CD, plan\_ED: string
* **Admin Class:** admin\_id: int, name: string, acd: string, dob: string, email: string, password: string
* **Cashier Class:** chashier\_id: int, name: string, email: string, password: string
* **Customer Class:** cutomer\_id: int, dob: string, customer\_account\_id: int, customer\_name: string, mobile: string
* **Guest Class:** guest\_id: int
* **Account:** account\_CD: string, accoutn\_id: int, account\_balance: int, account\_cutomer\_id: int, account\_plan: Subsplan plan, bank\_card: string, account\_status: string

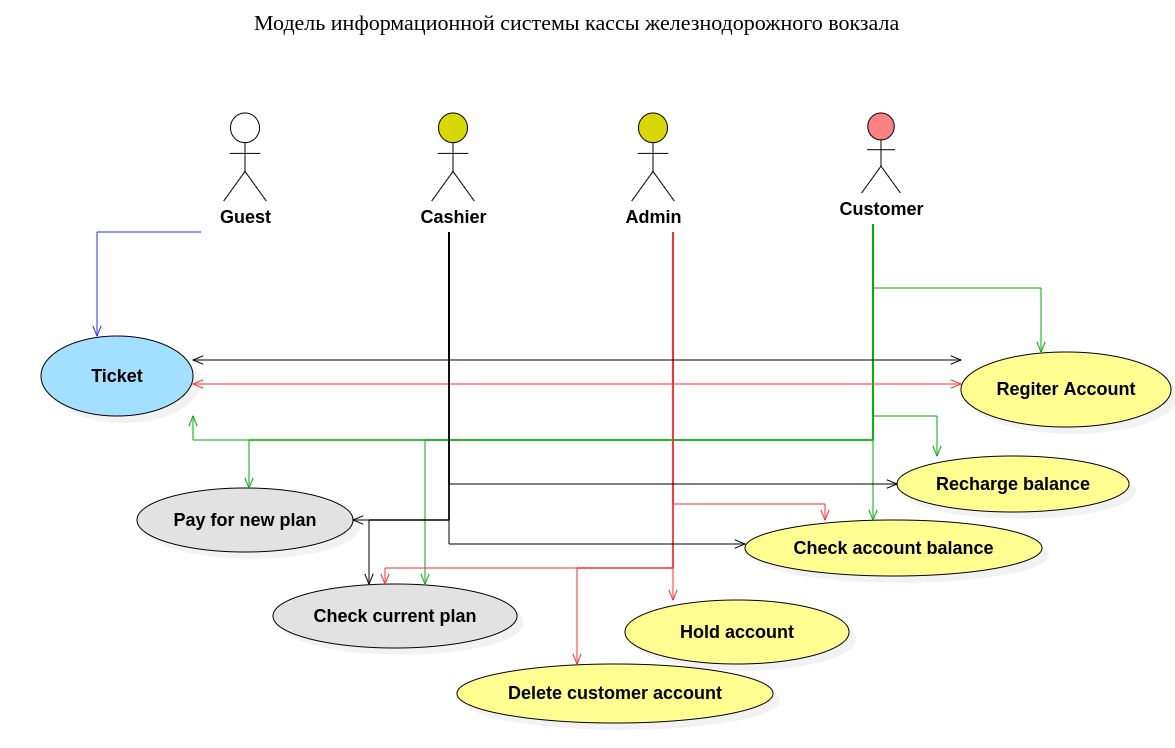
### 2.2 Классы и их методы Диаграммы классов системы бронирования железнодорожных билетов:

* **Ticket Class:** add\_ticket(), end\_ticket\_life(), ticket\_type(), get\_cost(), get\_remride(), use\_ticket()
* **Plan Class:** add\_plan(), get\_cost(), get\_ED()
* **Admin Class:** add\_admin(), delete\_admin(), edit()
* **Cashier Class:** create\_cashier(), delete\_cashier()
* **Customer Class:** create\_customer(), delete()
* **Guest Class:** create\_guest(), delete\_guest()
* **Account**: create\_account(), edit\_accountinfo(), deposite(), hold\_account(), check\_balance(), withdraw()



# Глава 3. Диаграмма Use case UML

Диаграмма вариантов использования включает в себя четырех участников системы (администратор, кассир, клиент, гость). он также содержит некоторые варианты использования, такие как создание учетной записи, подписка на план или заполнение карты.



Первая диаграмма User case:

