Потенційна ієрархія класів(може перероблюватись)

**Manager - Головний клас-менеджер.**

Методи:

planRoute(vehicle, destination)

startSimulation()

trackVehicleMovement()

endSimulation()

**Vehicle - Клас, що представляє окремй транспортний засіб.**

Методи:

planRoute(destination, stops)

Поля:

Тип транспортного засобу (легкове авто, вантажне авто, електричне авто тощо)

Поточна локація

Список планованих дій (поїхати до точки А, заправитися в точці Б тощо)

**Route - Клас, що представляє транспортний шлях.**

Методи:

setRestrictions(restrictions)

Поля:

Список точок на маршруті

Обмеження на шляху (дозволена швидкість, обмеження для вантажних авто тощо)

**Simulation - Клас для моделювання руху транспортних засобів.**

Методи:

startSimulation()

updateVehiclePosition()

handleEvents()

**Event - Клас, що представляє події в симуляції (заправка, посадка/висадка пасажирів).**

Методи:

triggerEvent(vehicle, event\_type, event\_location)

**Діаграма прецедентів (UML Use Case Diagram):**

**Користувач - Головний актор.**

Планування маршруту

Запуск симуляції руху

Слідкування за рухом транспортних засобів

Завершення симуляції

Система - Основний прецедентний контейнер.

**Транспортні засоби (Vehicles) - Прецедент для управління транспортними засобами.**

Додавання транспортного засобу

Видалення транспортного засобу

Планування маршруту для транспорту

**Транспортні шляхи (Routes) - Прецедент для управління транспортними шляхами.**

Додавання шляху

Видалення шляху

Задання обмежень на шляху

**Моделювання руху (Simulation) - Прецедент для моделювання руху транспортних засобів.**

Запуск симуляції

Опис руху транспортних засобів

Відображення руху транспортних засобів(після створення граф інтерфейса)

Стартовий код класа менеджера

#include <iostream>

#include <vector>

class Manager

{

public:

Manager() {}

// Метод для планування маршруту для транспортного засобу

void planRoute(Vehicle& vehicle, const Point& destination, const std::vector<Point>& stops)

{

// Логіка для планування маршруту

}

// Метод для запуску симуляції руху

void startSimulation()

{

// Логіка для запуску симуляції

}

// Метод для відстеження руху транспортних засобів

void trackVehicleMovement()

{

// Логіка для відстеження руху

}

// Метод для завершення симуляції

void endSimulation()

{

// Логіка для завершення симуляції

}

};

int main()

{

}