Sanjana Pavani

The goal is to enhance Husky Maps by integrating real-time public transportation data, user-generated hazards, and personalized location recommendations. This redesign aims to make Husky Maps more dynamic, user-centric, and responsive to real-time changes in the transit environment.

Function 1: addRealTimeTransitData

- **Inputs:** A list of real-time transit data, including bus/train IDs, stop IDs, arrival/departure times, and routes.
- **Outputs:** Updates the existing transit graph with real-time data.
- **Behavior:** This function updates the transit graph by incorporating real-time arrival and departure times, ensuring the graph reflects the current state of public transportation. For example, if a bus is delayed, this function updates the relevant nodes and edges in the graph to reflect the new arrival and departure times.

Function 2: addUserGeneratedHazard

- **Inputs:** Hazard location (stop ID), description, severity, and user ID.
- **Outputs:** Updates the graph with hazard information.
- **Behavior:** Allows users to report hazards such as construction, accidents, or weather conditions. The hazards are added to the graph and can affect route recommendations. For instance, if a user reports a construction site blocking a route, the graph will update to show the hazard, and the route planning will take this hazard into account, potentially suggesting alternative routes.

Function 3: recommendPersonalizedLocations

- **Inputs:** User ID and preferences (e.g., types of places liked, friends' visited locations).
- **Outputs:** A list of recommended locations.
- **Behavior:** Uses user preferences and social data to recommend locations. The function analyzes past user behavior and preferences to suggest places of interest. For example, if a user frequently visits coffee shops and parks, the function will recommend similar locations, factoring in the user's history and preferences.

Affordance Analysis:

- **Actions and Outcomes:** The ADT provides functionalities for updating the transit graph with real-time data, reporting and tracking hazards, and recommending personalized locations. These actions help improve user experience by providing up-to-date and relevant information, ensuring safer and more efficient travel, and offering personalized suggestions that enhance user satisfaction.

- **Value-Sensitive Design Principles:**
    - **Foreground Human Values:** The ADT prioritizes convenience, safety, and personalization, which are important for enhancing user experience and satisfaction. By providing real-time data and hazard information, it ensures that users can make informed decisions about their travel routes. Personalized recommendations enhance the user experience by making the system more relevant to individual needs and preferences.
    - **Pervasive Uptake:** Widespread usage of this ADT would lead to a more dynamic and user-centric transit system, improving overall efficiency and safety. As more users contribute real-time data and hazard reports, the system becomes more robust and accurate, benefiting all users.
    - **Stakeholders:** Direct stakeholders include public transit users and authorities. Indirect stakeholders include urban planners and local businesses benefiting from increased foot traffic due to personalized recommendations. For instance, businesses near frequently recommended locations may see increased patronage, and urban planners can use the data to improve infrastructure and transit planning.

Development Approach:

- **Choice of Redesign:** This redesign was chosen to address gaps in real-time data integration, hazard reporting, and personalized user experiences. These enhancements align with the increasing demand for smart city solutions and the need for adaptive and responsive transit systems.
- **Reflection and Notes:** The iterative design involved evaluating current limitations, brainstorming potential solutions, and refining the functions for optimal performance. Alternatives like static transit data and generic location recommendations were ruled out due to their lack of adaptability and personalization. The development process focused on creating a flexible and scalable solution that could handle real-time updates and user-generated data efficiently.

Implementation:

Most Efficient Approach: Graph Data Structure

- **Graph Representation:** Nodes represent bus stops, and edges represent routes with associated weights for travel times. Real-time data and hazards are stored as attributes of nodes and edges. The graph allows for efficient updates and queries, making it suitable for handling dynamic transit information.

Function Implementations:

1. addRealTimeTransitData

- **Implementation:**
  - **Steps:** Iterate through the list of real-time data. For each entry, update the corresponding nodes and edges in the graph with new arrival/departure times. This involves checking if the node (stop) exists in the graph and then updating its attributes with the real-time information.
  - **Asymptotic Analysis:** Updating nodes and edges is O(1) for each entry, leading to O(n) for n entries. This ensures that the function can handle large volumes of real-time data efficiently.

2. addUserGeneratedHazard

- **Implementation:**
  - **Steps:** Add a hazard entry to the graph at the specified stop, including the description and severity. Adjust route recommendations based on the hazard severity. This involves adding the hazard as an attribute to the node and updating the edge weights or disabling edges if necessary.
  - **Asymptotic Analysis:** Adding a hazard is O(1), but adjusting routes depends on the number of affected edges, leading to O(m) for m affected edges. This allows the system to quickly incorporate new hazards and adjust routing recommendations.

3. recommendPersonalizedLocations

- **Implementation:**
  - **Steps:** Analyze user preferences and past behavior. Use collaborative filtering to find similar users and their visited locations. Recommend top locations based on aggregated data. This involves querying the user preference database and comparing it with other users' data to generate recommendations.
  - **Asymptotic Analysis:** Collaborative filtering can be O(u + v) where u is the number of users and v is the number of locations, leading to efficient recommendations. This ensures that the system can provide personalized recommendations quickly, even for a large number of users.

Comparison to Alternatives:

Alternative Approach 1: Static Data Structures

- **Data Structure:** Arrays and lists for static data storage.
- **Reason for Ruling Out:** Lack of flexibility for real-time updates and inefficient handling of dynamic data. Static data structures would require complete reprocessing for each update, leading to high computational costs and slow performance.

Alternative Approach 2: Basic Linked Lists

- **Data Structure:** Linked lists for storing transit routes and hazards.
- **Reason for Ruling Out:** Inefficient access and update times, especially for real-time data and hazard adjustments. Linked lists would make it difficult to quickly access and update specific nodes and edges, leading to slower performance and a less responsive system.

Second Redesign:

Identified Assumption

The initial design assumes users can interpret visual maps and text descriptions easily. This assumption may not hold true for visually impaired users or those who prefer text-based navigation.

Second Redesign: Enhanced Accessibility Features

- **Redesign Description:** Integrate screen reading capabilities and hierarchical text descriptions for visually impaired users. This redesign aims to make Husky Maps more accessible by providing alternative ways to navigate the transit system.

Implementation:

1. addScreenReaderSupport

- **Inputs:** Map data and user preferences.
- **Outputs:** Text descriptions of map data.
- **Behavior:** Generates hierarchical text descriptions of the map, focusing on the most visible landmarks and popular places. This involves creating a text-based representation of the visual map data, which can be read by screen readers.

2. generateTextualNavigation

- **Inputs:** Start and destination points.
- **Outputs:** Step-by-step textual navigation.
- **Behavior:** Converts the visual route information into detailed textual instructions. This involves parsing the graph data to generate a sequence of text instructions that describe the route in detail, suitable for users who require text-based navigation.

3. recommendAccessibleRoutes

- **Inputs:** User accessibility needs.
- **Outputs:** List of accessible routes.
- **Behavior:** Recommends routes that prioritize accessibility features such as ramps and elevators. This involves querying the graph for nodes and edges that meet accessibility criteria and generating route recommendations based on this filtered data.

Affordance and Asymptotic Analysis:

- **Affordance Analysis:** Enhances accessibility, ensuring that visually impaired users can navigate the transit system effectively. By providing text descriptions and accessible route recommendations, the system becomes more inclusive and user-friendly.
- **Asymptotic Analysis:** Screen reading and text generation are O(n) for n nodes, ensuring efficient performance. This ensures that even large maps can be processed quickly, providing timely information to users.

Reflection and Alternatives:

- **Development Approach:** The redesign focuses on inclusivity, ensuring that all users can access and use the transit system effectively. This involved understanding the needs of visually impaired users and designing features that cater specifically to their requirements.
- **Alternatives:** Generic text descriptions and basic accessibility features were considered but ruled out due to their lack of depth and personalization. The chosen redesign provides a more comprehensive and user-centric solution, enhancing the overall accessibility of Husky Maps.

This iterative design process has led to significant enhancements in Husky Maps, making it more dynamic, user-friendly, and accessible. By integrating real-time data, hazard reporting, personalized recommendations, and accessibility features, the redesigned ADTs provide a strong solution for modern transit needs. This approach ensures that Husky Maps can cater to a diverse user base, offering personalized transit information.