LLM Automated Evaluation Results

**Functional Requirement Evaluation Table**

Based on the user's request, I conducted a detailed analysis of the requirements document, including identifying data entities, merging functional requirements, identifying hallucinated requirements, complete referencing, and testability. The analysis strictly followed the rules and standards provided by the user. The complete evaluation results are as follows.

**Step Descriptions:**

1. **Data Entity Identification**: Core data entities managed internally by the system were extracted from the functional requirements (excluding external interfaces). A data entity refers to a core object that is persistently stored or managed within the system. For example, "Route" is a data entity, whereas an external API like Map Service API is not.
2. **Functional Requirement Consolidation**:
   * Consolidation Rules: Merge identical functional points for the same data entity; exclude system-level general settings (e.g., user preference setting, multilingual support).
   * After consolidation, there are 11 functional requirements (from the original 13, 1.7 User Preference Setting and 1.8 Multilingual Support were excluded).
   * No consolidation cases: All requirements relate to different entities or different operations (e.g., Route Planning and Route History both involve "Route" but one creates a new route and the other retrieves past data, so they are not merged).
3. **Hallucinated Requirement Judgment**:
   * Criteria: A hallucinated requirement is one that introduces operations involving a data entity that cannot be connected to any existing entity in the system, causing the entity-relationship (E-R) diagram to lose weak connectivity.
   * System E-R Diagram Basis: All data entities are interconnected via core concepts such as Location, User, or Route. For instance, the "Location" entity connects to Route, Traffic Condition, POI, etc., while the "User Account" entity connects to Feedback and HistoryRoute. The overall E-R diagram remains weakly connected with no isolated nodes.
4. **Evaluation Table**: Each functional requirement was evaluated individually using short names derived from the functional descriptions.

**Data Entity Statistics**

* A total of **10 data entities** are involved in the document. These entities are extracted from the functional requirements and are core internal management objects:
  + **Location**: Used for origin, destination, current position (related to 1.1, 1.2, 1.4, 1.5, 1.6, 1.13)
  + **Route**: Includes path details, estimated time, transfer count (related to 1.1, 1.11)
  + **TrafficCondition**: Road congestion, incident information (related to 1.2)
  + **Schedule**: Departure and arrival times of public transport (related to 1.3)
  + **PointOfInterest (POI)**: Name, type, distance (related to 1.4)
  + **Trajectory**: User movement paths (related to 1.5)
  + **UserAccount**: Registration info, login status (related to 1.10)
  + **Feedback**: User-submitted issues or suggestions (related to 1.9)
  + **HistoryRoute**: Date, origin, destination (related to 1.11)
  + **WeatherInfo**: Real-time weather and forecast (related to 1.13)
* Note: Requirement 1.12 (Traffic Cost Estimation) produces “cost estimate” as an output, which is treated as a **computed attribute**, not an independent managed entity, and thus not counted separately. Similarly, 1.6 (Navigation Guidance) outputs “real-time navigation guidance,” which is considered a temporary service output, not a persistent entity.

**Functional Requirement Evaluation Table**

| Functional Requirement | Hallucination | Justification |

|------------------------|---------------|---------------|

1.1 Route Planning | No | Involves Location and Route entities; input/output logic is clear; depends on map and traffic APIs; function is feasible.  
1.2 Real-Time Traffic Update | No | Involves TrafficCondition and Location entities; acquires data based on current location; logic is well-defined.  
1.3 Timetable Query | No | Involves Schedule and Location entities; supported by public transport APIs; structure is reasonable.  
1.4 POI Search | No | Involves PointOfInterest and Location entities; supports keyword and distance filtering; logic is clear.  
1.5 Trajectory Recording | No | Involves Trajectory and Location entities; user location association is clear; process is complete.  
1.6 Navigation Guidance | No | Involves Route and Location entities; planning and dynamic guidance logic is sound.  
1.9 Community Feedback | No | Involves Feedback and UserAccount entities; submission confirmation is clear; function is valid.  
1.10 Account Management | No | Involves the core UserAccount entity; supports registration, login, and account operations; structure is valid.  
1.11 Route History | No | Involves HistoryRoute and Route entities; retrieval path is clear.  
1.12 Cost Estimation | No | Computes based on Route entity; estimation logic is valid; works with transport APIs.  
1.13 Weather Forecast | No | Involves WeatherInfo and Location entities; calls weather API for data; logic is realistic.

**Summary**

* A total of **10 data entities** are involved (Location, Route, TrafficCondition, Schedule, PointOfInterest, Trajectory, UserAccount, Feedback, HistoryRoute, WeatherInfo).
* After removing general-purpose settings (no actual merges), the number of functional requirements is **11**.
* Number of hallucinated requirements: **0** (all involved entities are interconnected via Location, User, or Route; the E-R diagram remains weakly connected).
* Number of non-hallucinated requirements: **11** (all are non-hallucinated).

Evaluation followed strict criteria, but since the requirement definitions are clear and the entities are well connected, no non-compliant items were found. All functional requirements meet the standards, indicating high documentation quality.