Course:

Software Engineering - COMP-6905-001

Project:

Skier-Routing app

Iteration # 03

Group # 12

Group Members:

Name	Student Number
1) Jinhao Luo	202388555
2) Nakash	202381669
3) Osama Ahmed Masood	202389195
4) Rahul Subramanian	202382981

Website Deployed: https://mun-comp-6905-group-12-ski-routing-app.vercel.app

GitHub Repository: https://github.com/MunSoftwareGroup12/software_project

Table of Contents

1. Cover Page……...01
2. Table of contents ……02
3. Three Main Use cases ...……03
4. Additional use cases……07
5. Use case Diagram……...08
6. Domain Model ...……09
7. Sequence Diagram……10
8. Class Diagram……11
9. Iteration 02 Checklist ……..12
10. Links to Github……12
11. Link to the Deployed Web page …..……..12

Section 01:

Three Main Use Cases

Use Case 01: Search Route.

Use-case Name	Search Route
Participating Actors	Skier
Flow of events	1. Skier selects the option “Calculate Route”. 2. Ski-Area-App provides Skier with an overview of the ski area. 3. Ski-Area-App requests Skier to provide a start location. 4. Skier selects start location. 5. Ski-Area-App requests Skier to provide end location 6. Skier selects the end location. 7. Ski-Area-App requests Skier to provide info about acceptable slope difficulty level. 8. Skier enters slope restrictions. 9. Ski-Area-App calculates possible routes to the requested destination (based on restrictions) and informs Skier about the options. 10. Ski-Area-App requests Skier to select a route based on indicated criteria (e.g., fastest, longest, easiest, lifts only) 11. Skier selects one of the options. 12. Ski-Area-App displays the desired route (based on the selected criteria) and provides detailed navigation instructions. 13. Skier hovers over his desired route to see its details. 14. Skier uses the route to navigate to the desired location.
Alternative Flows	4a. Skier requests Ski-Area-App to determine Skier’s position. 4b. Ski-Area-App calculates Skier’s position and selects it as the start location. 4b.1 Ski-Area-App is unable to determine Skier’s position. 4b.2 Ski-Area-App informs Skier about failure to determine current location. (Use Case resumes at step 3.) 8a. Skier requests Ski-Area-App to use Skier’s stored profile. 8b. Ski-Area-App retrieves Skier’s profile and uses it to determine acceptable difficulty level. 8b.1 No profile is available. 8b.2 Ski-Area-App informs Skier about missing profile. (Use Case resumes at step 7.)

	13a. Skier chooses to see the result for another criteria (e.g., fastest, longest, easiest, lifts only) 13b. Ski-Area-App displays alternative search criterias. (Use Case resumes at step 12.)
Entry condition	● The skier needs to be in the ski resort map and there needs to be at least two selectable locations(including ski locations and special locations).
Exit condition	● The skier selects a satisfactory route, or the skier chooses to exit route planning.
Quality Requirements	● The system needs to display the route planning results within 5 seconds after the skier initiates the search. ● The route planning results need to be clear and easy to understand. ● The system needs to be able to handle various complex terrains and route combinations.

Use Case 02: Filter Map.

Use-case Name	Filter map
Participating Actors	Skier
Flow of events	1. The skier opens the filtering menu on the map. 2. The system shows a list of available filters to the skier which include options such as: 'Restaurants', 'Restrooms', 'Chairlifts', 'Slopes and Ski Routes', 'Ski Buses', and other such available filters. 3. The skier selects the option that he wants to filter the map view by. 4. The system updates the map according to the skier's criteria and presents the updated version to the skier. 5. The skier selects additional filters while the first filter is active. 6. The system filters the map by all the selected filters and updates the view respectively to show all the filtered results combined. 7. The skier uses the filtered map to navigate the resort. 8. The skier unselects one or more filters that were previously selected. 9. The system updates the map by removing the results of the removed filter from the view. 10. The Skier navigates the map with any or no filters active.
Alternative Flows	4a. The result after the filter is empty 1. The system shows the skier an empty map. 6a. If there is no data for any of the selected filters there are 1. no changes in the map view.
Entry condition	● The skier has successfully accessed the Skier-Routing app on a mobile device. ● The app has up-to-date and accurate map data of the ski resort.
Exit condition	● The map displays only the filtered information according to the skier's preferences.
Quality Requirements	● Filtering actions should be intuitive and require minimal clicks/taps. ● The map should update quickly after applying filters to provide real-time information.

Use Case 03: Check Location Details.

Use-case Name	Check Location Details
Participating Actors	Skier
Flow of events	1. The skier opens the ski resort navigation web app. 2. The system loads the map of the ski area, displaying slopes, restaurants, restrooms, and chairlifts. 3. The skier selects a specific location on the map (slope, restaurant, restroom, chairlift). 4. The skier selects a slope on the map. 5. The system retrieves and displays information about the slope's difficulty and the trail length.
Alternative Flows	4a. The skier selects a restaurant. 1. The system retrieves and displays information about the restaurant's cuisine and operating hours. 4b. The skier selects a chairlift. 1. The system retrieves and displays information about the chairlift's capacity. 4c. The skier selects a restroom. 1. The system retrieves and displays information about the restroom's operating hours.
Entry condition	● The skier has successfully accessed the Skier-Routing app on their mobile device. ● The skier interacts with the map by selecting a location.
Exit condition	● A pop-up panel is displayed with the relevant information based on the type of location clicked.
Quality Requirements	● The pop-up panel displaying information should appear quickly after the skier selects a location on the map. ● The system should be able to handle an increased number of skiers during peak times without a significant degradation in performance.

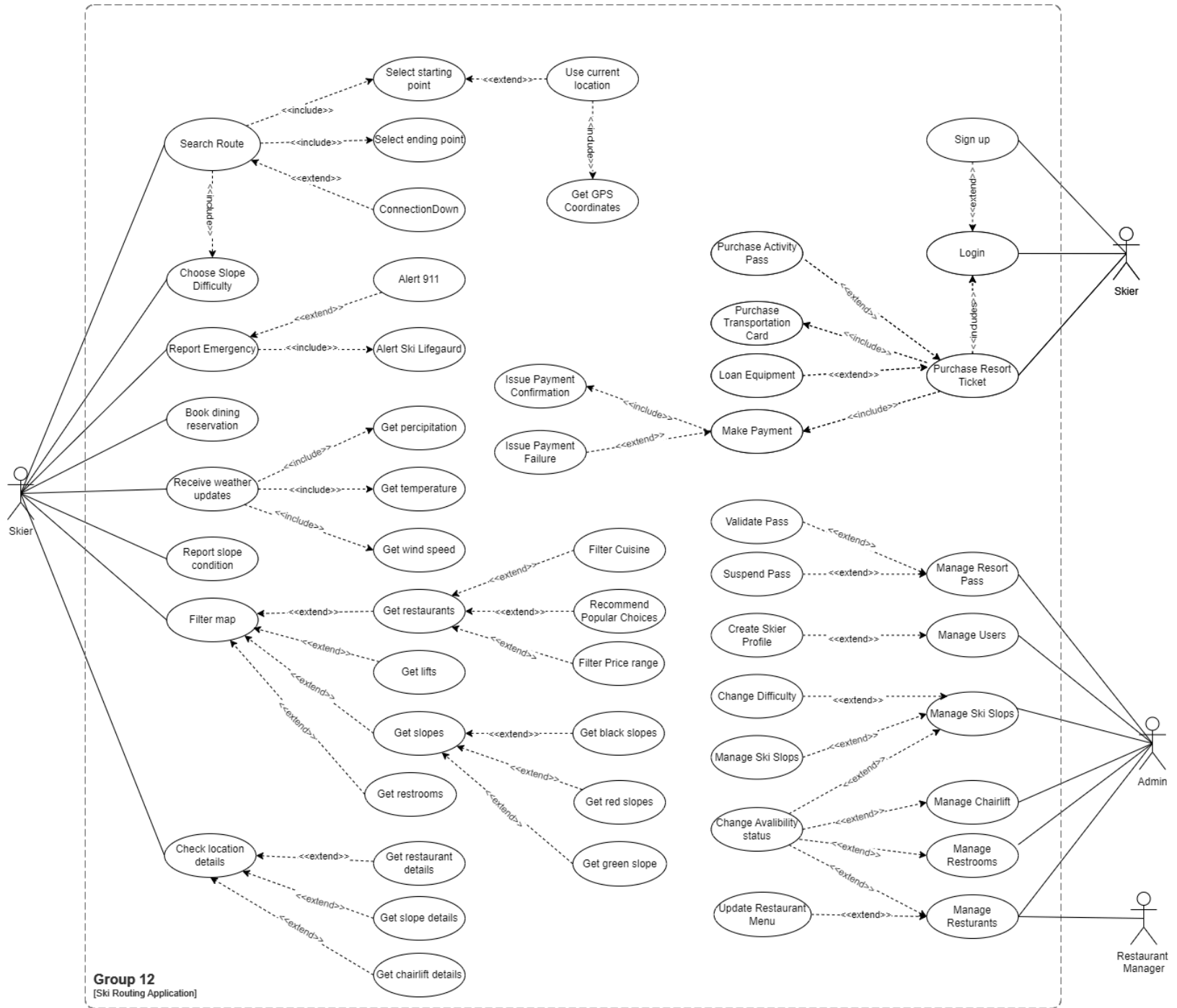
Section 02:

Additional use cases

1. **Identify Current Location**: Establish the skier's current position using the mobile device's GPS.
2. **Choose Slope difficulty**: Enable skiers to set their desired level of slope difficulty (blue, red, black).
3. **Find Public Restrooms**: Show locations of public restrooms within the ski area.
4. **Purchase Resort Ticket**: The skier goes online, selects everything he wants, and pays for it to get a pass for the resort.
5. **Check Lift Status**: Display the operating status of nearby lifts (open, closed, wait times).
6. **Book Restaurant Reservation**: Enable skiers to reserve bookings in restaurants.
7. **Report Emergency**: Provide a feature to request help or medical assistance.
8. **Rent Equipment**: Allow skiers to rent skiing equipment through the app.
9. **Receive weather update**s: Skiers can check the weather for the slopes in the resort.
10. **Display Route**: Show the calculated route on the map.

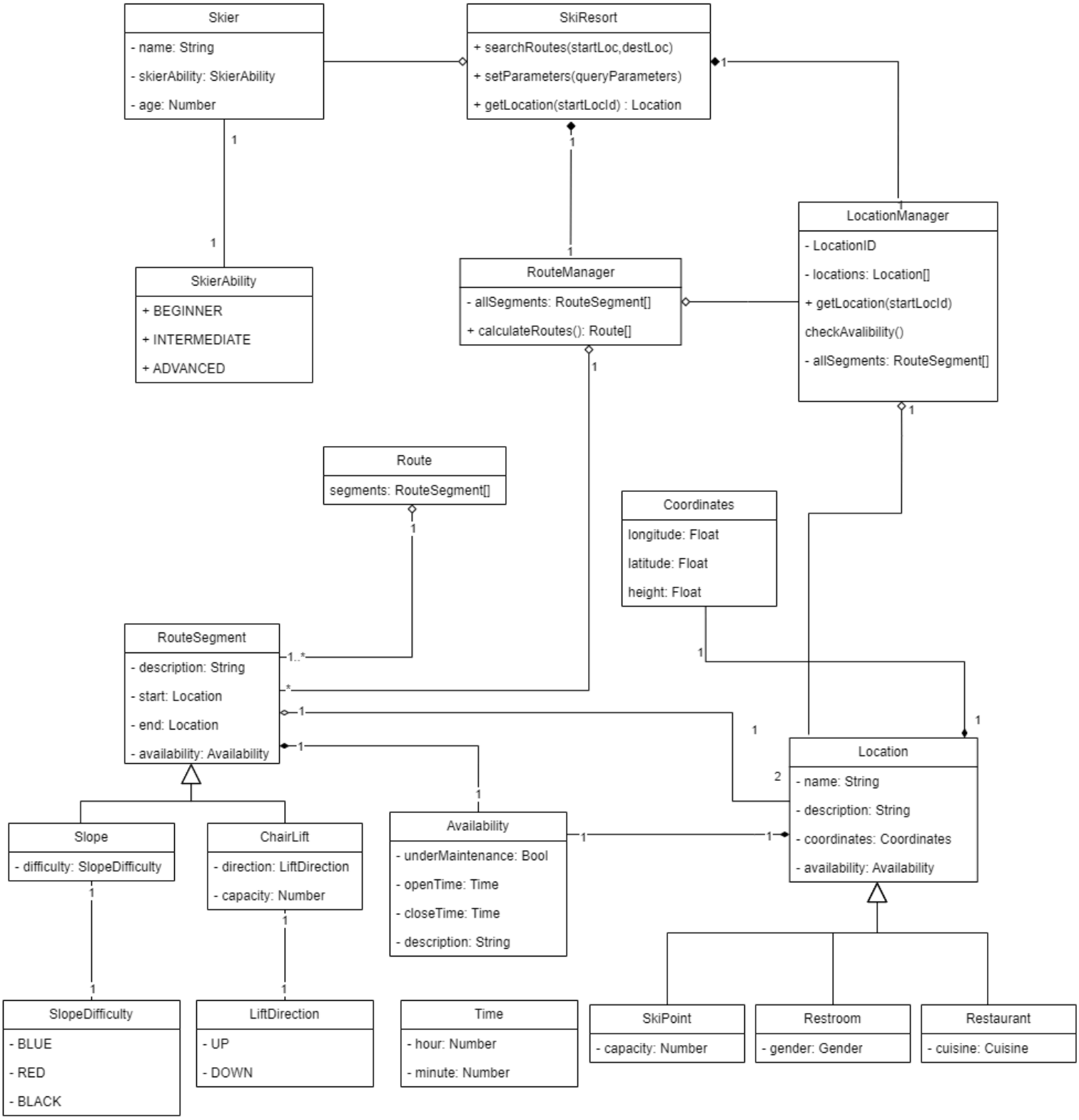
Section 03:

Use case Diagram



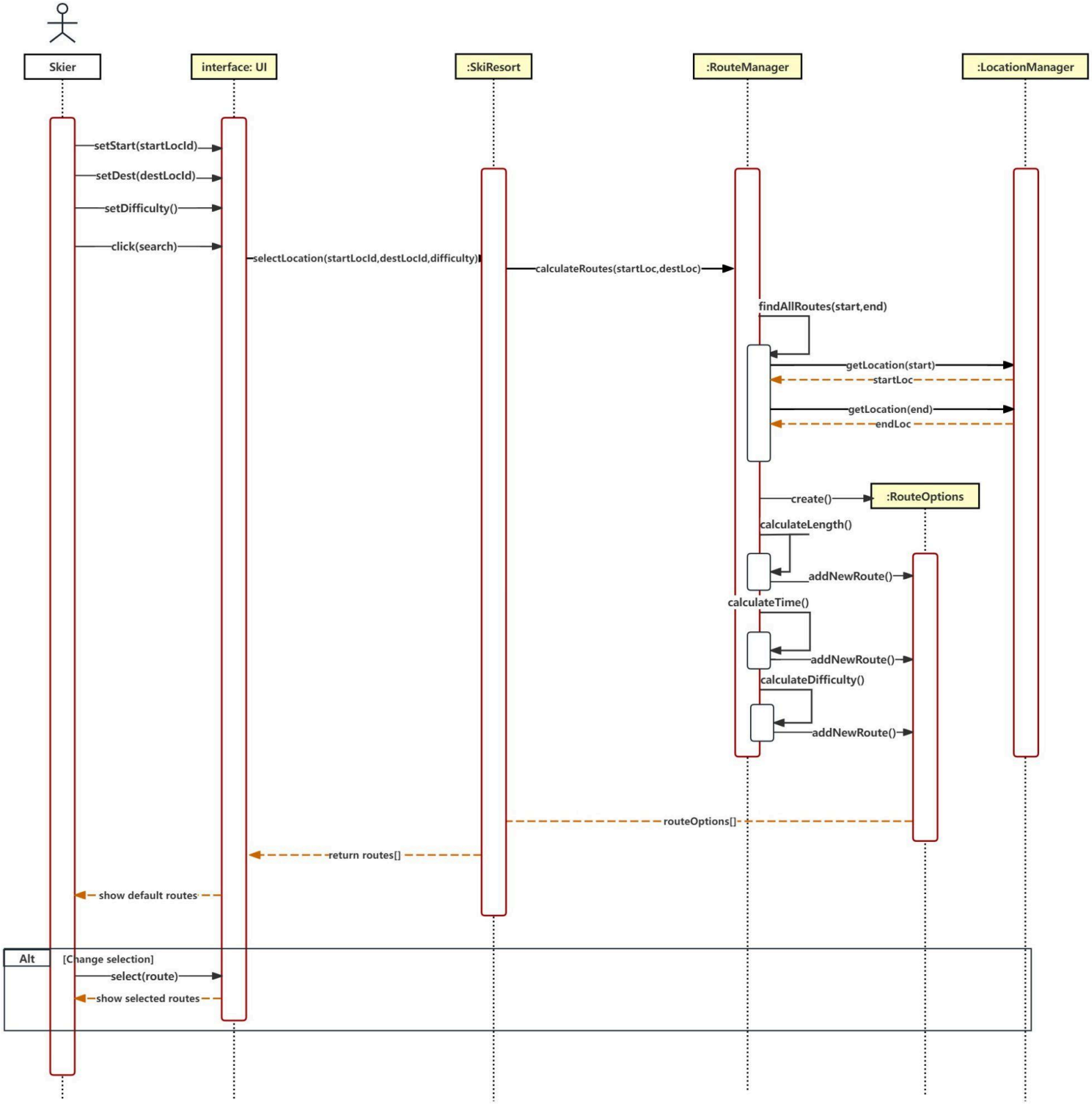
Section 04:

Domain Model



Section 05:

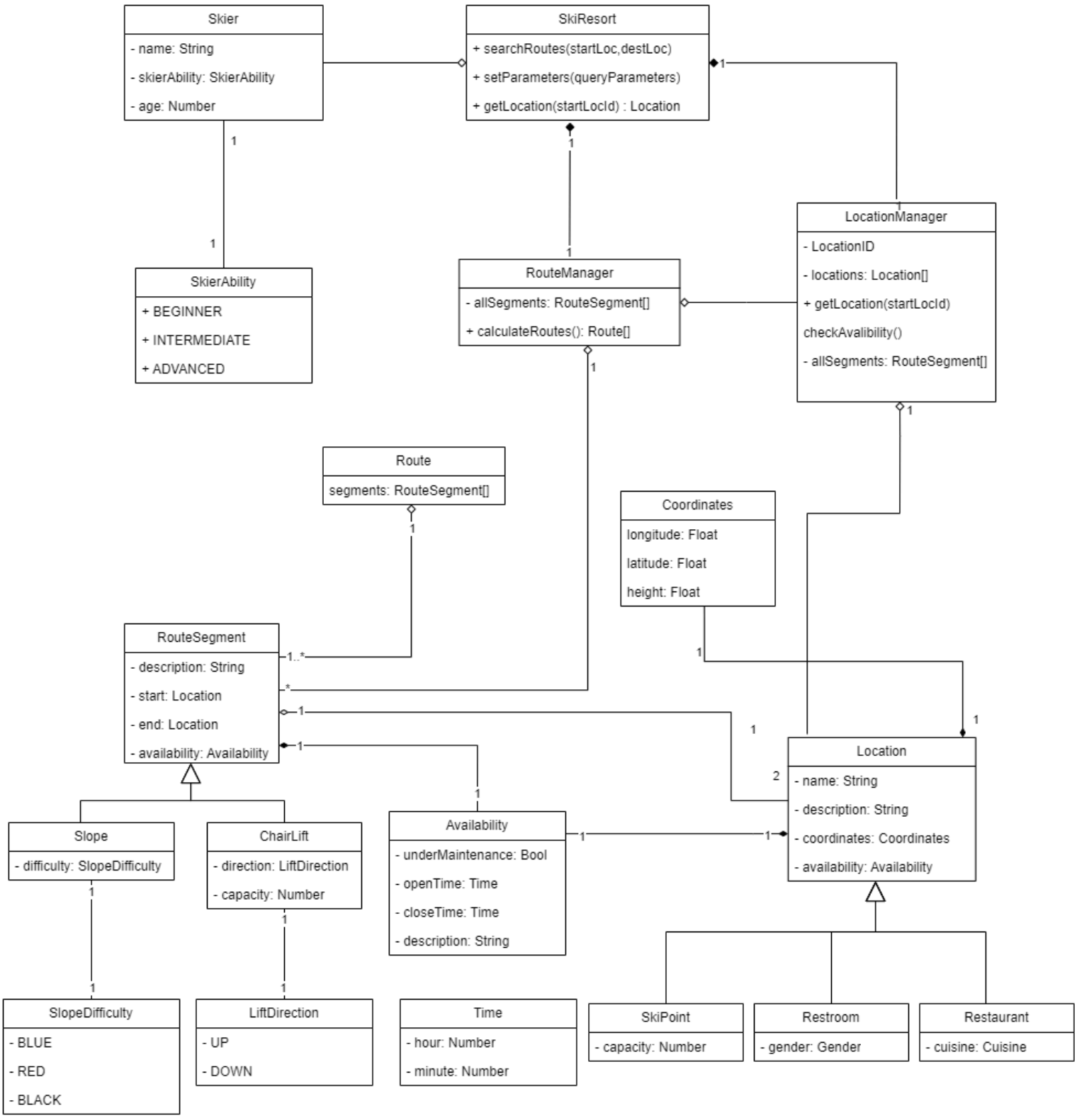
Sequence Diagram



Section 06:

Class Diagram

Classes implemented: SkiResort, RouteManager, LocationManager, Location (Mongo Entity), Route (Mongo Entity)



Checklist Iteration 03 for completed tasks:

- ☑ **Search Algorithm implementation.**
- ☑ **Show route as per criterias (e.g., fastest, longest, easiest, lifts only)**
- ☑ **Update Diagrams and Documentation.**
- ☑ **Complete Backend, Frontend and Deployment.**
- ☑ **Complete Project.**

(Completed 5 of 5)

Links:

Website Deployed:
https://mun-comp-6905-group-12-ski-routing-app.vercel.app

GitHub Repository:
https://github.com/MunSoftwareGroup12/software_project

(END)