

---

# **Software Requirements Specification**

**for**  
**RouteRunner**

**Version 1.0 approved**

**Prepared by:**

**Gay Ming Kai**  
**Low Kan Yui**  
**Khoo An Xian**  
**Andria Tan Hseun Yin**  
**Aung Aung Pyae Phyo**

**Nanyang Technology University**

**30 October 2024**

# Table of Contents

<b>Introduction</b>	<b>3</b>
Purpose	3
Product Scope	3
References	3
Overall Description	5
Product Perspective	5
Product Functions	5
User Classes and Characteristics	5
Operating Environment	6
External Interface Requirements	7
Software Interfaces	7
System Features	9
Login	9
Manage Runners (for Operator)	9
Manage Jobs (for Operator)	10
View Active Runners (for Operator)	11
View Route (for Runner)	11
View Carpark Availability (for Runner)	12
View Job (for Runner)	12
Job Allocation	12
<b>Non-Functional Requirements</b>	<b>14</b>
Performance Requirements	14
Security Requirements	14
Software Quality Attributes	14
<b>Appendix A: Data Dictionary</b>	<b>16</b>
<b>Appendix B: Use Case Diagram and Descriptions</b>	<b>17</b>
Login (for All)	18
For Operators	19
For Runners	27
<b>Appendix C: User Interfaces</b>	<b>29</b>
<b>Appendix D: Dialog Map</b>	<b>34</b>
<b>Appendix E: Sequential Diagrams</b>	<b>35</b>

# Introduction

## ○ Purpose

This SRS document details the requirements for the "RouteRunner" application, designed to help small and medium-sized enterprises (SMEs) efficiently track their runners on-the-ground and assign jobs. This document outlines the functional and non-functional requirements of the system, specifying how operators and runners will manage job assignments and location tracking. This SRS covers version 1.0 of the application, encompassing the core features required for effective job management, route optimization, and real-time location tracking of runners.

## ○ Product Scope

RouteRunner is a web application designed to streamline the job allocation process for SMEs by enabling operators to assign jobs to runners effectively. The system optimizes job allocation based on priority levels and route proximity, reducing idle time and improving efficiency. It provides real-time location tracking of runners, ensuring operators have up-to-date information on job statuses and runner locations. The primary objective of RouteRunner is to enhance the coordination of task management, reduce operational inefficiencies, and support businesses in managing their field operations more effectively. The application aligns with business strategies focused on improving productivity and customer satisfaction.

## ○ References

Our team referenced the following documentation.

### Google Maps API Documentation

- **Author:** Google LLC
- **Version Number:** Latest stable version
- **Source/Location:** <https://developers.google.com/maps/documentation>
- **Description:** This documentation provides detailed information on integrating Google Maps services, including geolocation, route optimization, map visualization, and traffic updates, which are essential for RouteRunner's route planning and real-time navigation features.

### GovTech Carpark Availability API Documentation

- **Author:** Government Technology Agency of Singapore (GovTech)
- **Version Number:** Latest version as available on Data.gov.sg
- **Source/Location:** <https://data.gov.sg/collections/85/view>

- **Description:** This documentation outlines the use of the Carpark Availability API, which provides real-time data on carpark availability across various locations in Singapore. It is integrated into RouteRunner to assist runners in finding available parking near their job locations.

### **OneMap API**

- **Author:** Singapore Land Authority (SLA)
- **Version Number:** 3.0
- **Source/Location:** <https://www.onemap.gov.sg/apidocs/apidocs>
- **Description:** The OneMap API is Singapore's government geospatial data API, providing access to various map-based services, including geolocation, reverse geocoding, route calculation, and property searches. This API is primarily used to access detailed Singapore map data, which can be incorporated into web or mobile applications to enhance spatial awareness and services tailored to specific locations within Singapore.

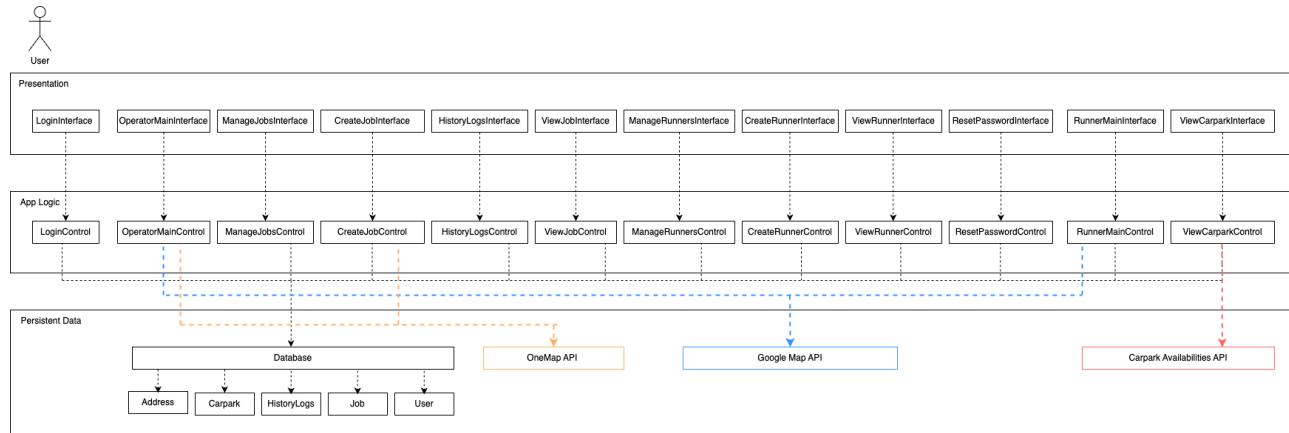
### **HDB Carpark Info**

- **Author:** HDB (Housing and Development Board)
- **Version Number:** 1st October (Updated start of every month)
- **Source/Location:** [https://data.gov.sg/datasets/d\\_23f946fa557947f93a8043bbef41dd09/view](https://data.gov.sg/datasets/d_23f946fa557947f93a8043bbef41dd09/view)
- **Description:** Information about HDB car parks such as operating hours, car park location (in SVY21), type of parking system, etc.

# Overall Description

- **Product Perspective**

RouteRunner is a self-contained product belonging to the on-demand service family.

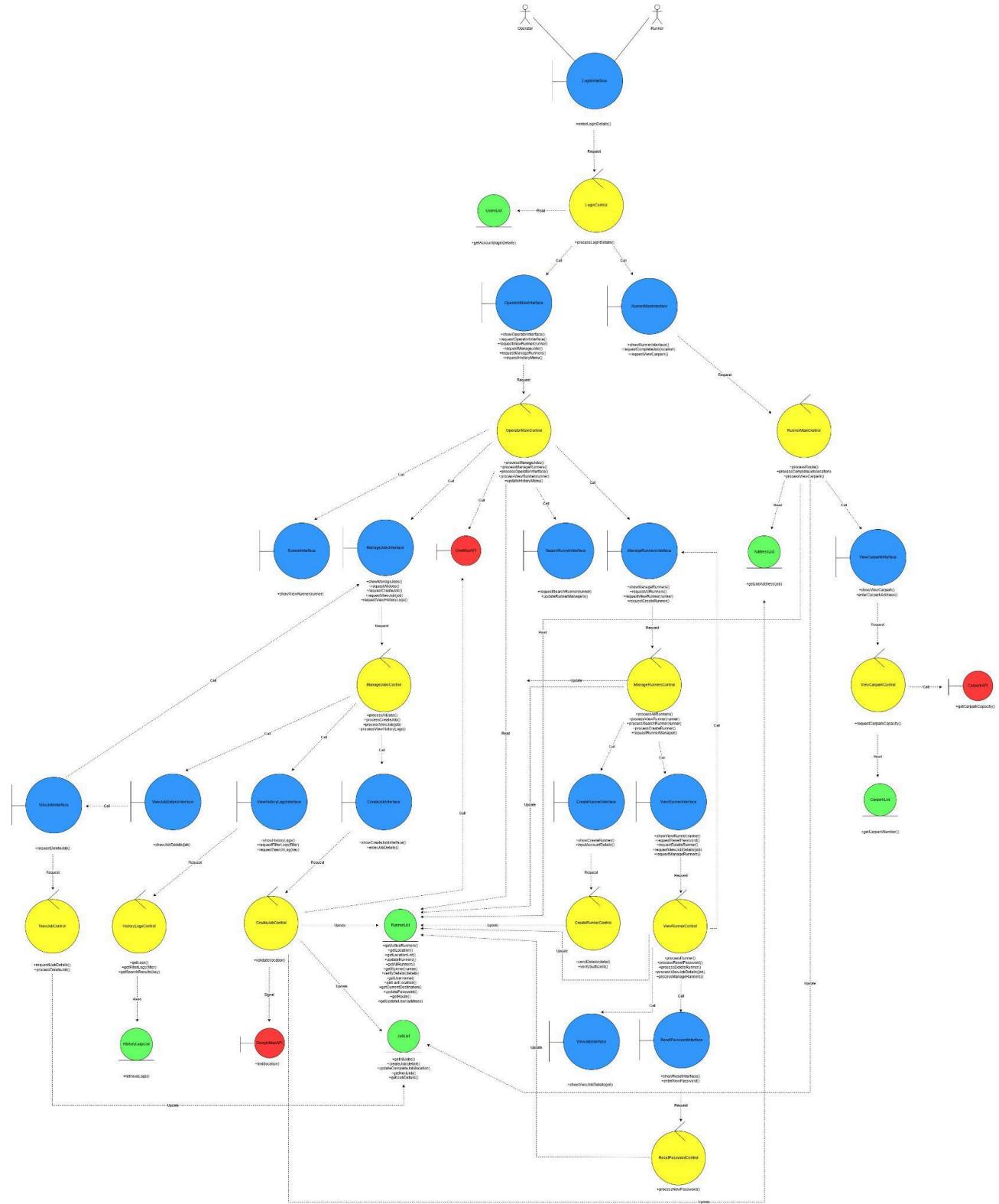


3-Layered Architecture Diagram

- **Product Functions**

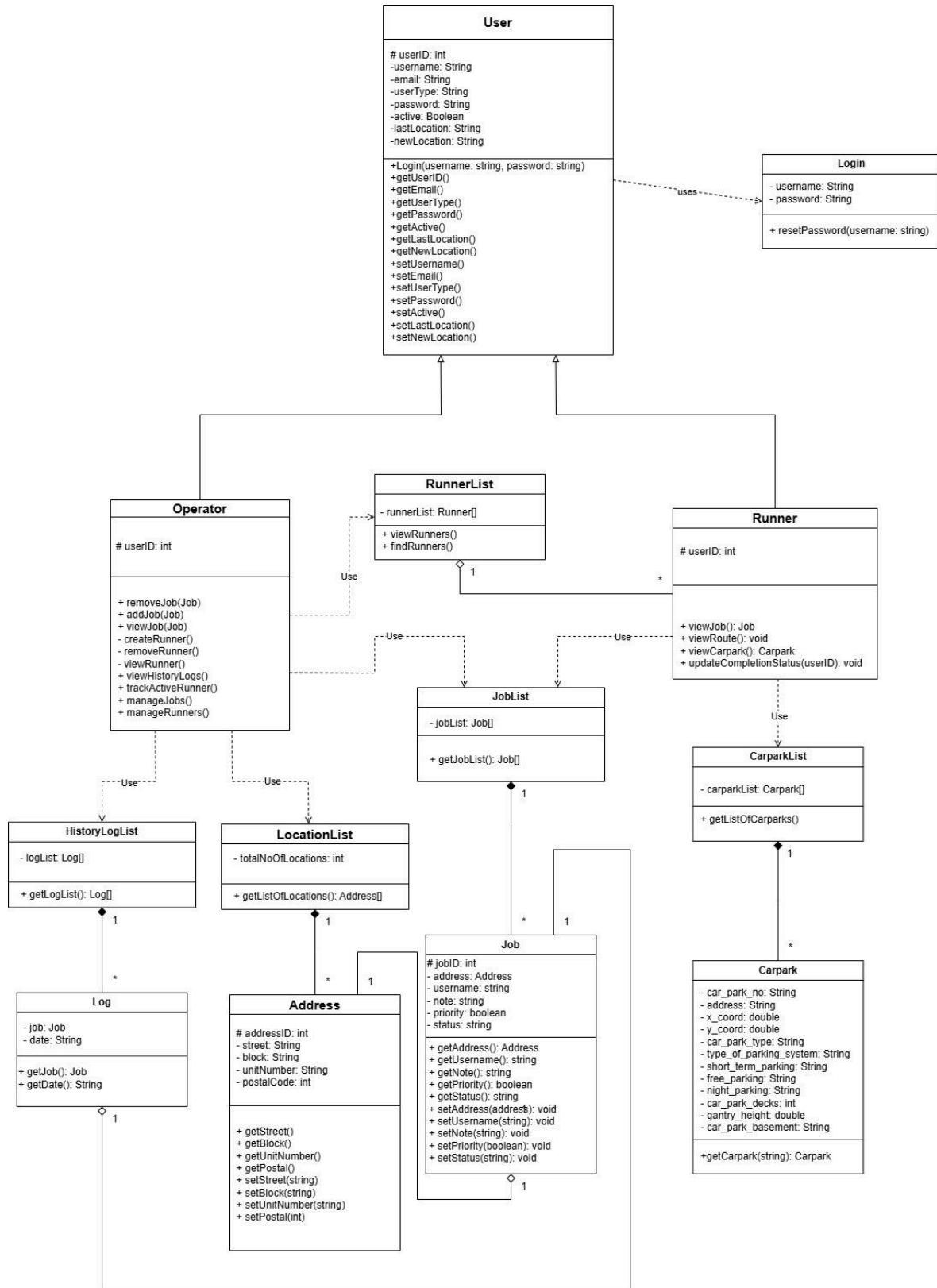
- Boundary Class Diagram

## Software Requirements Specification for RouteRunner



## Software Requirements Specification for RouteRunner

### Entity Class Diagram



For All	For Operators	For Runners	System
<ul style="list-style-type: none"><li>• Login</li><li>• View Job</li></ul>	<ul style="list-style-type: none"><li>• Manage Runners</li><li>• Manage Jobs</li><li>• View Active Runners</li></ul>	<ul style="list-style-type: none"><li>• View Route</li><li>• View Carpark Availability</li></ul>	<ul style="list-style-type: none"><li>• Job Allocation</li></ul>

## ○ **User Classes and Characteristics**

RouteRunner has 2 main user classes: Operator and Runner.

Operator:

- **Frequency of Use:** High; operators are expected to use the application throughout the day to manage job assignments and monitor runner activity.
- **Functions Used:** Operators will have access to functionalities related to job creation, allocation, route management, and runner tracking. They can also view and modify job statuses, manage runner accounts, and set priorities for tasks.
- **Privilege Levels:** High; operators have access to sensitive data, including runner information, job details, and real-time location tracking. Role-based access control ensures that only authorized personnel can modify job allocations and view certain data.

Runner:

- **Frequency of Use:** Moderate; runners interact with the system primarily to receive job assignments, update job statuses, and view their planned routes.
- **Functions Used:** Runners can log in to view their daily job list, check the priority and details of each task, update job completion statuses, and access route maps. They will also receive notifications about new assignments and changes to their routes.
- **Privilege Levels:** Low; runners have limited access to the system, primarily restricted to viewing and updating their own job-related information. They cannot modify job allocations or access the data of other runners.

## ○ **Operating Environment**

The application is built on the MERN stack (MongoDB, Express.js, React, Node.js), which provides a full-stack JavaScript environment.

# External Interface Requirements

## ○ Software Interfaces

RouteRunner integrates with several external software components to enhance its functionality and provide real-time, accurate information for users. The key software interfaces are described below:

### Google Maps API

- **Name and Version:** Google Maps Platform (latest stable version)
- **Purpose:** The application uses the Google Maps API for route optimization, geolocation services, and map visualization. This interface helps operators and runners view the most efficient routes for job assignments and display maps for navigating between job locations
- **Data In/Out:**
  - **Input:** Job locations (addresses), runner current locations, and destination addresses.
  - **Output:** Optimized routes, estimated travel times, and real-time map visualizations.
- **Nature of Communication:** RouteRunner sends location data and receives routing information through RESTful API calls. It uses secure HTTPS connections to ensure data privacy and integrity.

### GovTech Carpark Availability API

- **Name and Version:** GovTech Carpark Availability API (latest version as available on <https://data.gov.sg/>)
- **Purpose:** This interface provides real-time carpark availability data to assist runners in finding parking spots near job locations, reducing the time spent searching for parking. This feature is especially useful in busy urban areas where parking can be a challenge.
- **Data In/Out:**
  - **Input:** Location coordinates or addresses of nearby carparks.
  - **Output:** Real-time availability of parking spaces at carparks near the job location.
- **Nature of Communication:** RouteRunner makes HTTPS requests to the GovTech Carpark Availability API endpoint, retrieving real-time data. The application processes this information to display available carparks near job destinations on the runner's route map.

### Database Management System

- **Name and Version:** MongoDB (version 7.0.14, hosted at AWS Singapore)

- **Purpose:** The database stores all essential data, including user credentials, job details, runner information, and historical job logs. It also manages real-time updates on job statuses and route information for operators and runners.
- **Data In/Out:**
  - **Input:** User account information, job creation data, runner location updates, and status changes.
  - **Output:** Job lists for runners, route updates, job history records, and operator dashboard information.
- **Nature of Communication:** The application communicates with the database via NoSQL queries for MongoDB. Data transactions are secured with encryption protocols to ensure data integrity and security.

# System Features

## Login

### A. Description and Priority

The Login feature allows the System to authenticate the actor (either an Operator or Runner) and bring them to their respective homepages. It is of high priority.

### B. Stimulus/Response Sequences

Users must visit the RouteRunner web application.

### C. Functional Requirements

1. Users shall be able to log in by inputting a username and password.
  - 1.1. System must verify the username and password inputted with the accounts database.
  - 1.2. If the username is invalid, System must display an error message and allow User to input it again.
  - 1.3. If the password is invalid, System must display an error message and allow User to input it again.

## Manage Runners (*for Operator*)

### A. Description and Priority

The Manage Runners feature allows the Operator to view all Runners and their activity status, location and assigned jobs. It also allows the Operator to create or delete Runners. It is of high priority.

### B. Stimulus/Response Sequences

The Operator should log in, click on the burger menu at the top right corner, then click the Manage Runners button to stimulate this feature.

### C. Functional Requirements

1. Operator shall be able to manage Runners
  - 1.1. Operator shall be able to view the list of all Runners and their respective activity status
  - 1.2. Operator shall be able to search for a Runner by username
  - 1.3. Operator shall be able to **view individual Runner's details**
    - 1.3.1. Operator shall be able to view his last and next destination
    - 1.3.2. Operator shall be able to delete the Runner
    - 1.3.3. Operator shall be able to reset password for the Runner
    - 1.3.3.1. System must ensure that the password conforms to character restrictions\*

- 1.3.3.1.1. If the password is not valid, System must prompt Operator until a valid password is inputted
- 1.4. Operator shall be able to **create a new Runner**
  - 1.4.1. Operator shall be able register a new Runner by inserting his username, password and email
  - 1.4.2. System must ensure that the username is unique (ie. does not currently exist in User database)
    - 1.4.2.1. If the username is not unique, System must prompt Operator until a unique username is inputted
  - 1.4.3. System must ensure that the password conforms to character restrictions\*
    - 1.4.3.1. If the password is not valid, System must prompt Operator until a valid password is inputted
  - 1.4.4. System must ensure that the email is unique (i.e. does not exist in user User database)
    - 1.4.4.1. If the email is not unique, System must prompt Operator until a unique email is inputted
  - 1.4.5. System must ensure the email address is valid
    - 1.4.5.1. If the email is not valid, System must prompt Operator until a valid email is inputted

\*Password character restrictions: at least 8 characters long, contain at least a number, upper and lower case letters, and a special character

## **Manage Jobs (*for Operator*)**

### **A. Description and Priority**

The Manage Jobs feature allows the Operator to perform Create, Read, Update and Delete (CRUD) operations on the Job database. It is of high priority.

### **B. Stimulus/Response Sequences**

The Operator should log in, click on the burger menu at the top right corner, then click the Manage Jobs button to stimulate this feature.

### **C. Functional Requirements**

1. Operator shall be able to manage jobs
  - 1.1. Operator shall be able to view a list of all ongoing and pending jobs
  - 1.2. Operator shall be able to search for a job by Job ID
  - 1.3. Operator shall be able to **view individual job's details**
    - 1.3.1. System shall display the job ID, Runner assigned and job status
    - 1.3.2. Operator shall be able to delete the job
  - 1.4. Operator shall be able to **view a history log** of completed jobs
    - 1.4.1. System shall display the job ID, address, Runner assigned, job status, and date of each job

- 1.4.2. System shall be able to filter by job ID, address, runner username, and the completed date.
- 1.4.3. System shall be able to search for a specific archived job using the job ID
- 1.5. Operator shall be able to **create a new job**
  - 1.5.1. Operator shall be able to input the postal code of the job location
    - 1.5.1.1. Operator shall be able to verify if the postal code is mappable
      - 1.5.1.1.1. If the postal code is unmappable, System must prompt Operator until a mappable postal code is inputted
      - 1.5.1.1.2. If postal code is valid, System autofills the remaining location details (street name and block number)
  - 1.5.2. Operator shall be able to input the additional notes for the job
  - 1.5.3. Operator shall be able to input the priority level for the job
  - 1.5.4. System must add this new job into the Job database with its status marked as ‘pending’

## **View Active Runners (*for Operator*)**

### **A. Description and Priority**

The View Active Runners feature allows the Operator to view all active Runners and their location, both in a list and on a map. It is of medium priority.

### **B. Stimulus/Response Sequences**

The Operator should log in to stimulate this feature (on the Home Page)

### **C. Functional Requirements**

- 1. Operator shall be able to track active Runners
  - 1.1. Operator shall be able to view all active Runners and their locations
  - 1.2. Operator shall be able to view individual Runner’s details

## **View Route (*for Runner*)**

### **A. Description and Priority**

The View Route feature allows the Runner to view his route on a map and indicate completion of jobs to generate a new route. It is of high priority.

### **B. Stimulus/Response Sequences**

The Runner should log in to stimulate this feature (on the Home Page)

### **C. Functional Requirements**

- 1. Runner shall be able to view the route from his previous job to his next job on a map

2. Runner shall be able to indicate completion of a job
  - 2.1. System shall update the status of the job from ‘ongoing’ to ‘completed’ on the Job database
  - 2.2. System shall allocate a new job to the Runner
  - 2.3. System shall update the route on the map

## **View Carpark Availability (*for Runner*)**

### **A. Description and Priority**

The View Carpark Availability feature allows the Runner to check for vacancies in carparks nearest to the Runner’s next job destination. It is of low priority.

### **B. Stimulus/Response Sequences**

The Runner should log in, then click ‘View Carpark’ on the Homepage to stimulate this feature

### **C. Functional Requirements**

1. Runner shall be able to view the carpark number, address, free capacity and last updated timing for the car park nearest to his job location.

## **View Job (*for Runner*)**

### **A. Description and Priority**

The View Job feature allows the Runner to view details for the current job he is assigned. It is of high priority.

### **B. Stimulus/Response Sequences**

The Runner should log in, then click the burger menu and ‘View Job’ to stimulate this feature.

### **C. Functional Requirements**

1. Runner shall be able to view the job ID, address, notes and priority level for each job

## **Job Allocation**

### **A. Description and Priority**

The Job Allocation feature allocates newly created jobs to the most suitable Runner. Jobs are of high or low priority.

### **B. Stimulus/Response Sequences**

The Operator should create a new job to stimulate this feature.

### **C. Functional Requirements**

1. System must allocate a ‘pending’ job based on its priority level
  - 1.1. If job is high priority, System must allocate the job to the next available Runner
  - 1.2. If job is low priority, System must allocate the job to the Runner whose next destination is closest to the job
  - 1.3. System must update the status of the job from ‘pending’ to ‘ongoing’ in the Job database once it is allocated
  - 1.4. System must update the assigned Runner’s route and locations once the job is allocated

# Non-Functional Requirements

## ○ Performance Requirements

- System must allocate jobs to Runners within 3 seconds of the operator submitting the job.
- Location tracking updates must have a latency of no more than 2 seconds.

## ○ Security Requirements

- Users must have an account in order to use the application.
- Operators and Runners must log into the system with a password consisting of at least 8 characters, and at least a number, a lowercase letter, an uppercase letter and a special character.
- Sensitive data, such as the user's account password , must be encrypted in transit and at rest.

## ○ Software Quality Attributes

- Scalability
  - System must support up to 500 concurrent users without significant performance degradation to response time and service availability
  - The real-time locating system shall be able to scale from managing up to IoT devices without loss of data fidelity or monitoring capabilities.
- Reliability
  - System must ensure 99.9% uptime, ensuring availability during peak business hours.
  - After a failed operation (e.g. job allocation or notification delivery), System must retry operation within 2 seconds for up to 3 times, before reporting an error.
  - After a system reboot, the full system functionality must be restored within 5 minutes.
- Usability
  - User interface must have a clean and consistent design, with easy-to-understand icons and clear labels.
  - System must be displayed in the local language according to the user's locale.
  - System must display date, time, and address formats according to the user's locale settings.
- Compliance
  - System must comply with relevant data protection regulations, such as the general data protection regulation (GDPR).
  - System must log all user actions related to job allocation and location tracking for audit purposes.
- Maintainability
  - The codebase shall adhere to best practices for clean code, including modular design, well-documented functions, and consistent naming conventions.
  - System must be designed with ease of maintenance in mind, allowing for quick updates and bug fixes within 1 hour of downtime.

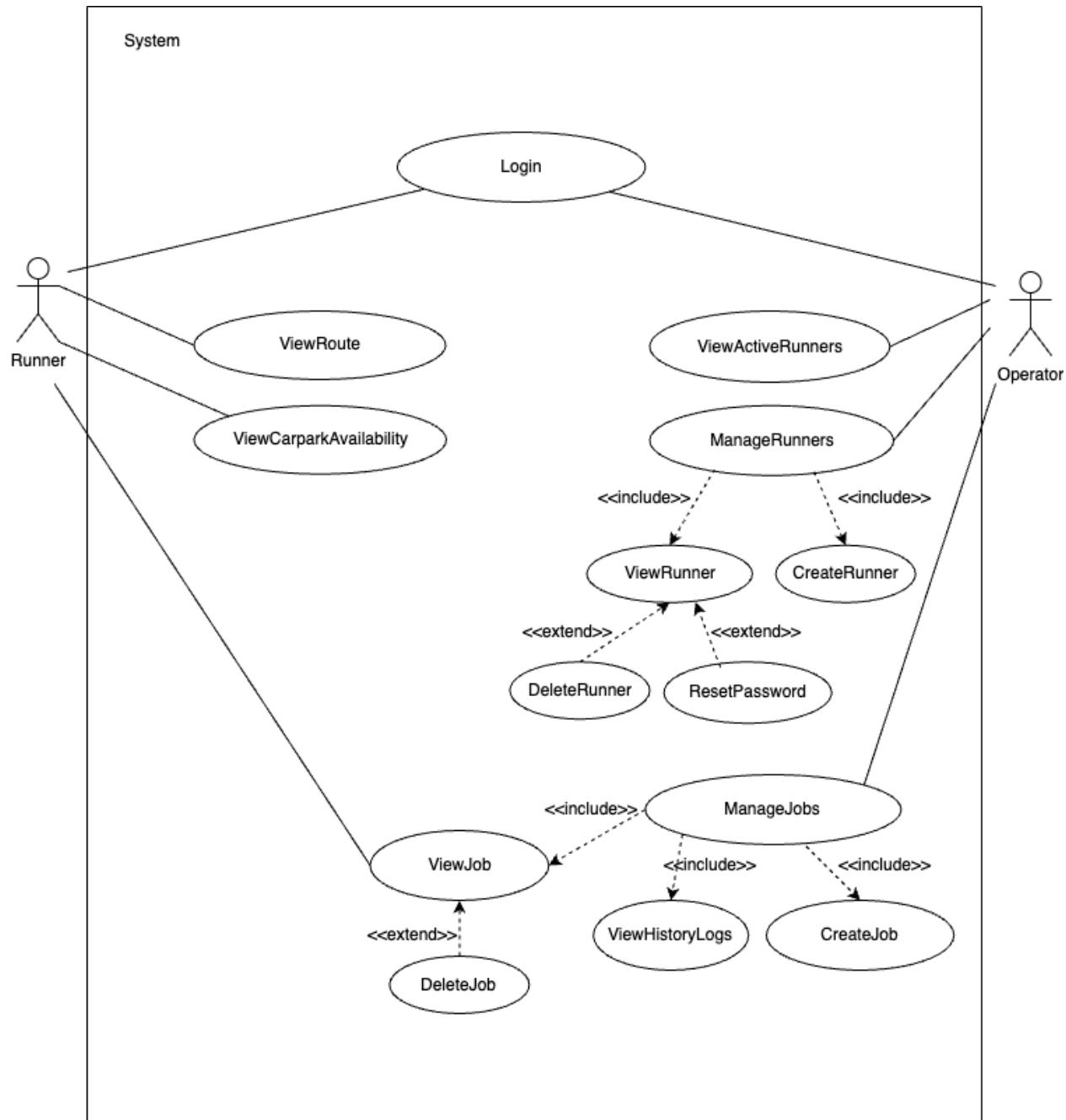
***Software Requirements Specification for RouteRunner***

- Automated testing shall cover at least 80% of the codebase to ensure reliability during updates.
- Portability & Compatibility
  - System must be compatible with both iOS and Android devices.
  - System shall be able to operate and maintain a consistent user experience across various mobile devices, including different screen sizes and resolutions.
- Interoperability
  - System shall be able to integrate seamlessly with external APIs, such as mapping services (e.g. Google Maps) for route optimization.

## Appendix A: Data Dictionary

Term	Definition
Operator	Human user of the system that is responsible for entering location and managing Runners.
Runner	Person that moves between assigned locations to complete tasks (eg. repair worker). Runner follows routes created by the system based on the allocated location. A Runner is either active or non-active.
Active Runner	A Runner that is logged into the system, and can be assigned a job.
Job	A task assigned to a Runner, consisting of a delivery or action at a specific location, and can be marked as high or low priority.
Completion status	Indicates whether the job is “waiting”, “ongoing” or “completed”.

## Appendix B: Use Case Diagram and Descriptions



## Login (for All)

<b>Use Case ID:</b>	SYS.ALL.1		
<b>Use Case Name:</b>	Login		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	8/9/24	<b>Date Last Updated:</b>	28/10/24

<b>Actor:</b>	Runner / Operator
<b>Description:</b>	This use case allows System to authenticate the actor and bring them to their respective homepages.
<b>Preconditions:</b>	1. The users have their accounts with username and password.
<b>Postconditions:</b>	1. Actor has been authenticated. 2. Actors gain access to the respective System's functions.
<b>Priority:</b>	High
<b>Frequency of Use:</b>	High
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>User keys in username and password</li> <li>System validates username and password against User database</li> <li>System returns a successful login if username and password are valid</li> <li>System routes the actor to their respective homepage</li> </ol>
<b>Alternative Flows:</b>	<p>SYS.ALL.1.AC.1: If the actor's username does not exist in the database,</p> <ol style="list-style-type: none"> <li>System displays an error message and allows the actor to key in again.</li> </ol> <p>SYS.ALL.1.AC.2: If the actor's username exists but the password is wrong,</p> <ol style="list-style-type: none"> <li>System displays an error message and allows the actor to key in again.</li> </ol>
<b>Exceptions:</b>	<p>SYS.ALL.1.EX.1: Database storing the actor credentials is not available.</p> <ol style="list-style-type: none"> <li>System displays message: "System is down temporarily. Please contact service desk at +65 XXXXXXXX"</li> </ol>
<b>Includes:</b>	NIL
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>Sensitive data such as users' passwords must be encrypted in transit and at rest.</li> </ol>
<b>Assumptions:</b>	<ol style="list-style-type: none"> <li>Operators' accounts must be created by developers of RouteRunner</li> <li>Runners' account must be created by Operators (see CreateRunner use case for Operators)</li> <li>If Operators forget their password, they must contact developers of RouteRunner to reset their password</li> <li>If Runners forget their password, they must contact their Operators to reset their password (see ViewRunner use case for Operators)</li> </ol>
<b>Notes and Issues:</b>	NIL

## For Operators

<b>Use Case ID:</b>	SYS.OP.1		
<b>Use Case Name:</b>	ManageJobs		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	10/9/24	<b>Date Last Updated:</b>	28/10/24

<b>Actor:</b>	Operator
<b>Description:</b>	This use case allows the Operator to perform Create, Read, Update and Delete (CRUD) operations on the jobs database.
<b>Preconditions:</b>	1. Operator must be authenticated.
<b>Postconditions:</b>	1. Jobs database is accessed by the Operator.
<b>Priority:</b>	High
<b>Frequency of Use:</b>	High
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on the burger menu on the Home page</li> <li>2. Operator selects Manage Jobs</li> <li>3. System displays a list of all pending and ongoing jobs, a search bar, and options to Create Job and View History Log</li> <li>4. <b>If Operator enters a Job ID in search bar</b>, System displays the selected job</li> <li>5. <b>If Operator clicks a specific job</b>, System routes Operator to View Job interface (see included ViewJob use case)</li> <li>6. <b>If Operator clicks 'Create Job'</b>, System routes Operator to Create Job interface (see included CreateJob use case)</li> <li>7. <b>If Operator clicks 'History Log'</b>, System routes Operator to History Log interface (see included HistoryLog use case)</li> </ol> <p>*pending job = yet to be assigned, ongoing job = assigned and in progress</p>
<b>Alternative Flows:</b>	SYS.OP.1.AC.1: If Operator enters invalid Job ID in search bar, <ol style="list-style-type: none"> <li>1. System displays blank list of jobs</li> </ol>
<b>Exceptions:</b>	SYS.OP.1.EX.1. Database storing the jobs is not available <ol style="list-style-type: none"> <li>1. System displays message: "System is down temporarily. Please contact service desk at +65 XXXXXXXX"</li> </ol>
<b>Includes:</b>	<ol style="list-style-type: none"> <li>1. CreateJob</li> <li>2. ViewJob</li> <li>3. HistoryLogs</li> </ol>
<b>Special Requirements:</b>	1. System must ensure 99.9% uptime, ensuring availability during peak business hours.
<b>Assumptions:</b>	<ol style="list-style-type: none"> <li>1. Alterations to the database are transactional only.</li> </ol>
<b>Notes and Issues:</b>	NIL

<b>Use Case ID:</b>	SYS.OP.1.1		
<b>Use Case Name:</b>	CreateJob		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	8/9/24	<b>Date Last Updated:</b>	28/10/24

<b>Actor:</b>	Operator
<b>Description:</b>	This use case allows the operator to create new jobs which will later be allocated by the System.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The Operator must be authenticated.</li> <li>2. The database must be accessible to add jobs.</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. A new job is created in the Job database.</li> <li>2. A new job is allocated to a suitable Runner.</li> </ol>
<b>Priority:</b>	High
<b>Frequency of Use:</b>	High
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on Create Job on the Manage Job interface</li> <li>2. Operator enters the postal code of the job location</li> <li>3. System verifies if postal code is mappable</li> <li>4. System autfills the remaining job location details (street name, block number) if postal code is mappable</li> <li>5. Operator adds additional notes and indicates priority level of the job</li> <li>6. Operator clicks Create Job</li> <li>7. System adds this new job into the Job database with its status marked as ‘pending’</li> <li>8. System allocates the job to the most suitable Runner</li> <li>9. System updates the status of the job from ‘pending’ to ‘ongoing’ when the job is allocated</li> </ol>
<b>Alternative Flows:</b>	<p>SYS.OP.1.1.AC.1: If postal code is unmappable,</p> <ol style="list-style-type: none"> <li>1. System displays error message and allows Operator to input again</li> </ol>
<b>Exceptions:</b>	<p>SYS.OP.1.1.EX.1. Database storing the jobs is not available</p> <ol style="list-style-type: none"> <li>2. System displays message: “System is down temporarily. Please contact service desk at +65 XXXXXXXX”</li> </ol>
<b>Includes:</b>	NIL
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. System must ensure 99.9% uptime, ensuring availability during peak business hours.</li> <li>2. System must log all user actions related to job allocation and location tracking for audit purposes.</li> </ol>
<b>Assumptions:</b>	NIL
<b>Notes and Issues:</b>	When new jobs are added, an ID should be assigned for easy reference.

<b>Use Case ID:</b>	SYS.OP.1.2		
<b>Use Case Name:</b>	HistoryLogs		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	8/9/24	<b>Date Last Updated:</b>	28/10/24

<b>Actor:</b>	Operator
<b>Description:</b>	This use case allows the operator to view history logs of all completed jobs.
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. The Operator must be authenticated.</li> <li>2. The Jobs database must be accessible and updated</li> </ul>
<b>Postconditions:</b>	1. System displays history logs
<b>Priority:</b>	Low
<b>Frequency of Use:</b>	Mid
<b>Flow of Events:</b>	<ul style="list-style-type: none"> <li>1. Operator clicks on History Logs on the Manage Job interface</li> <li>2. System displays list of all completed jobs and their details (including job ID, address, Runner assigned, job status, and date)</li> </ul>
<b>Alternative Flows:</b>	NIL
<b>Exceptions:</b>	<p>SYS.OP.1.2.EX.1. Database storing the jobs is not available</p> <p>3. System displays message: "System is down temporarily. Please contact service desk at +65 XXXXXXXX"</p>
<b>Includes:</b>	NIL
<b>Special Requirements:</b>	NIL
<b>Assumptions:</b>	NIL
<b>Notes and Issues:</b>	This history logs is for audit purposes

<b>Use Case ID:</b>	SYS.ALL.2		
<b>Use Case Name:</b>	ViewJob		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	8/9/24	<b>Date Last Updated:</b>	28/10/24

<b>Actor:</b>	Operator / Runner
<b>Description:</b>	This use case allows the Operator or Runner to browse details of specific jobs.
<b>Preconditions:</b>	1. The Operator/Runner must be authenticated
<b>Postconditions:</b>	1. System displays details of selected job 2. For Operators, System displays an option to Delete Job
<b>Priority:</b>	High
<b>Frequency of Use:</b>	High
<b>Flow of Events:</b>	<p>If Operator</p> <ol style="list-style-type: none"> <li>1. Operator clicks on a specific Job on the Manage Job interface</li> <li>2. System displays the details of the job and an option to Delete Job.</li> <li>3. If Operator clicks ‘Delete Job’, System deletes the job from the Job database</li> </ol> <p>If Runner</p> <ol style="list-style-type: none"> <li>1. Runner clicks on the burger menu in his home page</li> <li>2. Runner clicks on View Job</li> <li>3. System displays the details of the job</li> </ol>
<b>Alternative Flows:</b>	NIL
<b>Exceptions:</b>	<p>SYS.ALL.2.EX.1. If Operator tries to Delete Job for that is ongoing,</p> <ol style="list-style-type: none"> <li>1. The Delete Job button is disabled and Operator is unable to click it</li> </ol> <p>SYS.ALL.2.EX.2. Database storing the jobs is not available</p> <ol style="list-style-type: none"> <li>4. System displays message: “System is down temporarily. Please contact service desk at +65 XXXXXXXX”</li> </ol>
<b>Includes:</b>	NIL
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. System must enforce role-based access control (RBAC) to limit access to sensitive functionalities based on user roles. The Delete Job option should only be accessible for Operators</li> <li>2. System must support up to 500 concurrent users without significant performance degradation to response time and service availability</li> <li>1. System must ensure 99.9% uptime, ensuring availability during peak business hours.</li> <li>2. System must be compatible with both iOS and Android devices.</li> </ol>
<b>Assumptions:</b>	NIL
<b>Notes and Issues:</b>	

<b>Use Case ID:</b>	SYS.OP.2		
<b>Use Case Name:</b>	ManageRunners		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	8/9/24	<b>Date Last Updated:</b>	28/10/24

<b>Actor:</b>	Operator
<b>Description:</b>	This use case allows the Operator to perform Create, Read, Update and Delete (CRUD) operations on the Runners database.
<b>Preconditions:</b>	1. Operator must be authenticated
<b>Postconditions:</b>	1. Runners database is accessed by the Operator.
<b>Priority:</b>	Medium
<b>Frequency of Use:</b>	Low
<b>Flow of Events:</b>	<p>If Operator wants to view list of all Runners,</p> <ol style="list-style-type: none"> <li>1. Operator clicks on the burger menu on the Home page</li> <li>2. Operator selects Manage Runners</li> <li>3. System displays a list of all Runners and their activity status, a search bar, and an option to ‘Create New Runner’</li> <li>4. <b>If Operator enters a name in search bar</b>, System displays the respective Runner and his activity status</li> <li>5. <b>If Operator clicks on a Runner</b>, System routes Operator to View Runner interface (see included ViewRunner use case)</li> <li>6. <b>If Operator clicks ‘Create New Runner’</b>, System routes Operator to Create Runner interface (see included CreateRunner use case)</li> </ol>
<b>Alternative Flows:</b>	<p>SYS.OP.2.AC.1: If Operator enters invalid Job ID in search bar,</p> <ol style="list-style-type: none"> <li>2. System displays blank list of Runners</li> </ol>
<b>Exceptions:</b>	<p>SYS.OP.2.EX.1. Database storing the Runners’ information is not available.</p> <ol style="list-style-type: none"> <li>1. System displays message: “System is down temporarily. Please contact service desk at +65 XXXXXXXX”</li> </ol>
<b>Includes:</b>	<ol style="list-style-type: none"> <li>1. CreateRunner</li> <li>2. ViewRunner</li> </ol>
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. System must ensure 99.9% uptime, ensuring availability during peak business hours.</li> <li>2. System must be compatible with both iOS and Android devices.</li> </ol>
<b>Assumptions:</b>	NIL
<b>Notes and Issues:</b>	NIL

<b>Use Case ID:</b>	SYS.OP.2.1		
<b>Use Case Name:</b>	CreateRunner		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	10/9/24	<b>Date Last Updated:</b>	28/10/24

<b>Actor:</b>	Operator
<b>Description:</b>	This use case allows the Operator to create new Runner profiles to be inserted into the Runner database.
<b>Preconditions:</b>	1. Operator must be authenticated.
<b>Postconditions:</b>	1. A new Runner record is created in the Runner database.
<b>Priority:</b>	Medium
<b>Frequency of Use:</b>	Low
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks Create New Runner on Manage Runners Interface</li> <li>2. Operator fills in the username, password and email for the Runner</li> <li>3. Operator clicks Register</li> <li>4. System validates the inputs</li> <li>5. System creates a new Runner account in the User database</li> </ol>
<b>Alternative Flows:</b>	<p>SYS.OP.2.1.AC.1: If password does not conform to character restrictions,</p> <ol style="list-style-type: none"> <li>1. System displays error message and allows Operator to input new password</li> </ol> <p>SYS.OP.2.1.AC.2: If email is invalid,</p> <ol style="list-style-type: none"> <li>1. System displays error message and allows Operator to input email again</li> </ol> <p>SYS.OP.2.1.AC.3: If username already exists,</p> <ol style="list-style-type: none"> <li>1. System displays error message and allows Operator to input username again</li> </ol> <p>SYS.OP.2.1.AC.4: If email already exists,</p> <ol style="list-style-type: none"> <li>1. System displays error message and allows Operator to input email again</li> </ol>
<b>Exceptions:</b>	<p>SYS.OP.2.1.EX.1: An input field is left blank.</p> <ol style="list-style-type: none"> <li>1. System does not allow Operator to click Register</li> </ol>
<b>Includes:</b>	NIL
<b>Special Requirements:</b>	<p>New account passwords should be fulfill the following character restrictions:</p> <ul style="list-style-type: none"> <li>• at least 8 characters long,</li> <li>• a mixture of upper and lower case letters,</li> <li>• a special case character</li> <li>• a number</li> </ul>
<b>Assumptions:</b>	NIL
<b>Notes and Issues:</b>	NIL

<b>Use Case ID:</b>	SYS.OP.2.2		
<b>Use Case Name:</b>	ViewRunner		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	10/9/24	<b>Date Last Updated:</b>	28/10/24

<b>Actor:</b>	Operator
<b>Description:</b>	This use case allows the Operator to view details of a specific Runner, as well as delete the Runner or reset his password.
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. The Operator must be authenticated</li> <li>2. The Runner account exists in the database</li> </ul>
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. System displays details of selected Runner (including his last and next destination)</li> </ul>
<b>Priority:</b>	Mid
<b>Frequency of Use:</b>	Low
<b>Flow of Events:</b>	<ul style="list-style-type: none"> <li>1. Operator clicks on a specific Runner</li> <li>2. System displays the details of the Runner and an option to Delete Runner or Reset Password for the Runner</li> <li>3. <b>If Operator clicks ‘Delete Runner’</b>, System deletes the Runner from the User database.</li> <li>4. <b>If Operator clicks ‘Reset Password’</b>, Operator can input a new password for the runner and System resets the Runner’s password in the User database.</li> </ul>
<b>Alternative Flows:</b>	NIL
<b>Exceptions:</b>	NIL
<b>Includes:</b>	NIL
<b>Special Requirements:</b>	<ul style="list-style-type: none"> <li>1. System must ensure 99.9% uptime, ensuring availability during peak business hours.</li> <li>2. System must be compatible with both iOS and Android devices.</li> </ul>
<b>Assumptions:</b>	NIL
<b>Notes and Issues:</b>	

<b>Use Case ID:</b>	SYS.OP.3		
<b>Use Case Name:</b>	ViewActiveRunners		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	10/9/24	<b>Date Last Updated:</b>	28/10/24

<b>Actor:</b>	Operator
<b>Description:</b>	This use case allows the Operator and Runner to view locations for active Runners.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The Operator must be authenticated</li> <li>2. There are active Runners (Runners who are logged into their accounts)</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. System displays a list of the active Runners' names and their locations</li> <li>2. System displays the location of active Runners on a map</li> </ol>
<b>Priority:</b>	Mid
<b>Frequency of Use:</b>	High
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. Operator logs into his account</li> <li>2. System displays a list of the active Runners' names and their locations, and visualizes all their locations on a map</li> <li>3. <b>If Operator clicks on a Runner in the list</b>, System routes Operator to View Runner interface (see included ViewRunner use case)</li> </ol>
<b>Alternative Flows:</b>	NIL
<b>Exceptions:</b>	<p>SYS.OP.2.2.EX.1.. Database storing the Runners is not available</p> <ol style="list-style-type: none"> <li>1. System displays message: "System is down temporarily. Please contact service desk at +65 XXXXXXXX"</li> </ol>
<b>Includes:</b>	NIL
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. System shall be able to integrate seamlessly with Google Maps API for location visualization.</li> <li>2. System must ensure 99.9% uptime, ensuring availability during peak business hours.</li> <li>3. System must be compatible with both iOS and Android devices.</li> </ol>
<b>Assumptions:</b>	NIL
<b>Notes and Issues:</b>	NIL

## For Runners

<b>Use Case ID:</b>	SYS.RN.1		
<b>Use Case Name:</b>	ViewRoute		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	10/9/24	<b>Date Last Updated:</b>	28/10/24

<b>Actor:</b>	Runner
<b>Description:</b>	This use case allows the Runner to view the route to the next job and indicate completion when the job is done.
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. The Runner must be authenticated.</li> <li>2. The Runner must be active.</li> <li>3. The Runner must have a job allocated.</li> </ul>
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. System displays the route to the Runner's next job location (the route originates from his last job location)</li> </ul>
<b>Priority:</b>	High
<b>Frequency of Use:</b>	High
<b>Flow of Events:</b>	<ul style="list-style-type: none"> <li>1. Runner logs into his account</li> <li>2. System displays the route to his next allocated job on a map in the Home page</li> <li>3. Runner clicks Complete Job when he has completed the job</li> <li>4. System changes the job status from 'ongoing' to 'completed' in the Job database</li> <li>5. System allocates next Job to Runner</li> <li>6. System displays the updated route to his next allocated job on the map</li> </ul>
<b>Alternative Flows:</b>	NIL
<b>Exceptions:</b>	<p>SYS.RN.1.EX.1.. No job is available for allocation</p> <ul style="list-style-type: none"> <li>1. System does not update the route displayed on the map after Runner clicks Complete Job</li> </ul> <p>SYS.RN.1.EX.2. Route cannot be generated after job is allocated due to network issues</p> <ul style="list-style-type: none"> <li>1. System displays 'poor network connection' message</li> </ul>
<b>Includes:</b>	NIL
<b>Special Requirements:</b>	<ul style="list-style-type: none"> <li>1. System shall be able to integrate seamlessly with Google Maps API for route visualization.</li> <li>2. System must support up to 500 concurrent users without significant performance degradation to response time and service availability</li> <li>3. System must ensure 99.9% uptime, ensuring availability during peak business hours.</li> <li>4. System must be compatible with both iOS and Android devices.</li> </ul>
<b>Assumptions:</b>	NIL
<b>Notes and Issues:</b>	NIL

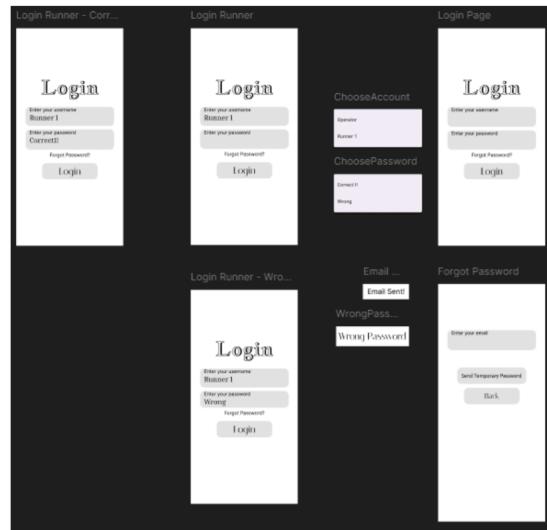
<b>Use Case ID:</b>	SYS.RN.2
---------------------	----------

<b>Use Case Name:</b>	ViewCarparkAvailability		
<b>Created By:</b>	Alvin & Kan Yui	<b>Last Updated By:</b>	An Xian
<b>Date Created:</b>	10/9/24	<b>Date Last Updated:</b>	28/10/24

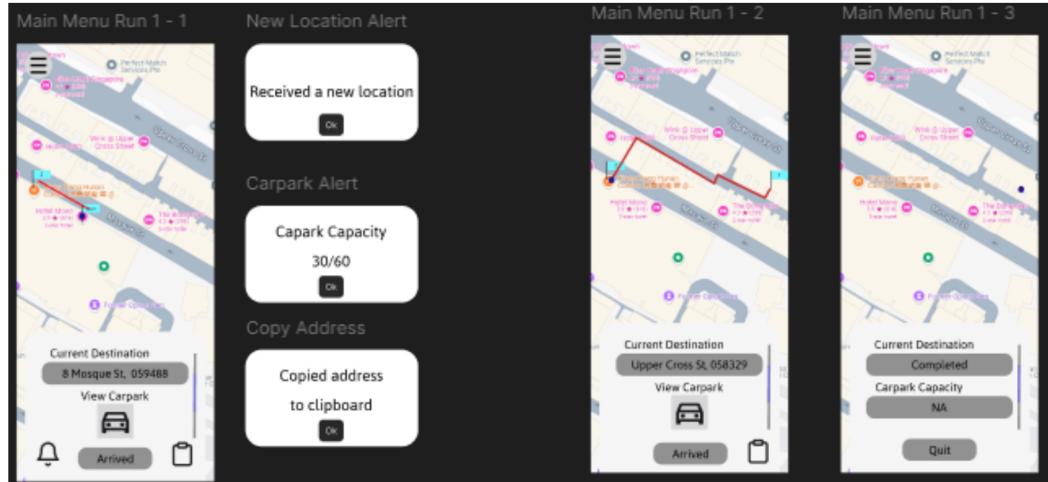
<b>Actor:</b>	Runner
<b>Description:</b>	This use case allows the Runner to view the capacity of the carpark nearest to their next job location.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The Runner must be authenticated.</li> <li>2. The Runner must be active.</li> <li>3. The Runner must have a job allocated.</li> </ol>
<b>Postconditions:</b>	1. System displays the relevant information of the carpark nearest to the Runner's next job location
<b>Priority:</b>	Medium
<b>Frequency of Use:</b>	Medium
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. Runner clicks View Carpark on the Runner Main Interface page</li> <li>2. System displays the carpark number, address, free capacity, and last updated timing of the carpark nearest to the Runner's destination</li> </ol>
<b>Alternative Flows:</b>	NIL
<b>Exceptions:</b>	<p>SYS.RN.2.EX.2 Carpark cannot be found due to issues with API connection</p> <ol style="list-style-type: none"> <li>1. System displays message: "System is temporarily down. Please try again in 5 minutes"</li> </ol>
<b>Includes:</b>	NIL
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. System must support up to 500 concurrent users without significant performance degradation to response time and service availability</li> <li>2. System must ensure 99.9% uptime, ensuring availability during peak business hours.</li> <li>3. System must be compatible with both iOS and Android devices.</li> </ol>
<b>Assumptions:</b>	NIL
<b>Notes and Issues:</b>	NIL

## Appendix C: User Interfaces

### Login



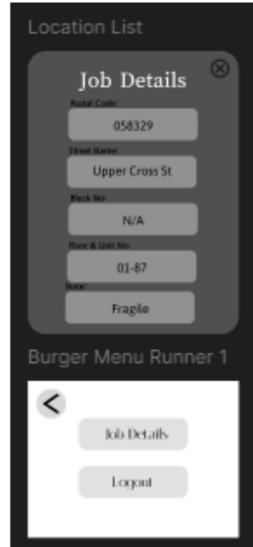
## Runner Main Menu Scenario 1



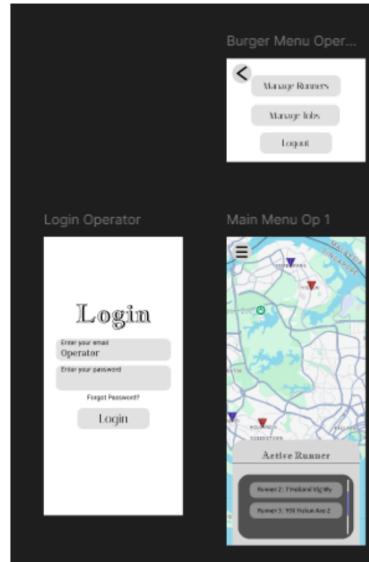
## Runner Main Menu Scenario 2



## Runner Main Menu Misc



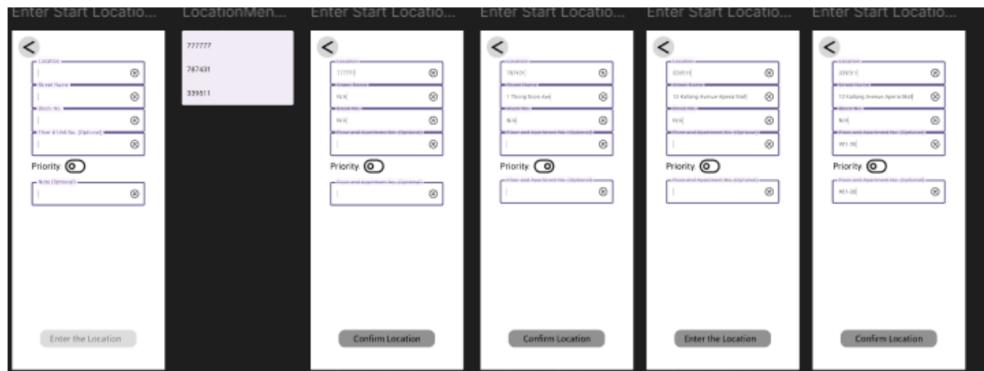
## Operator Main Menu



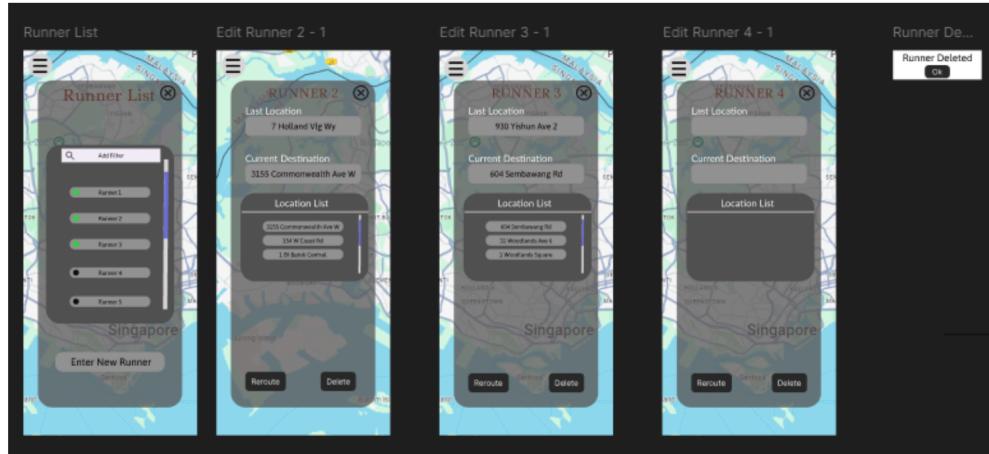
## ManageJobs



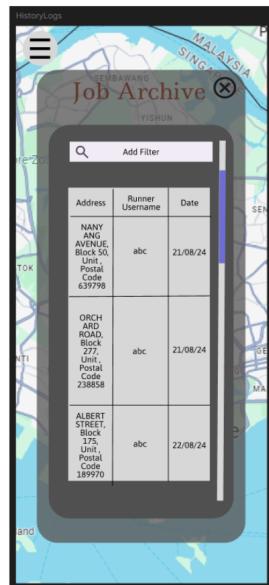
## CreateJobs



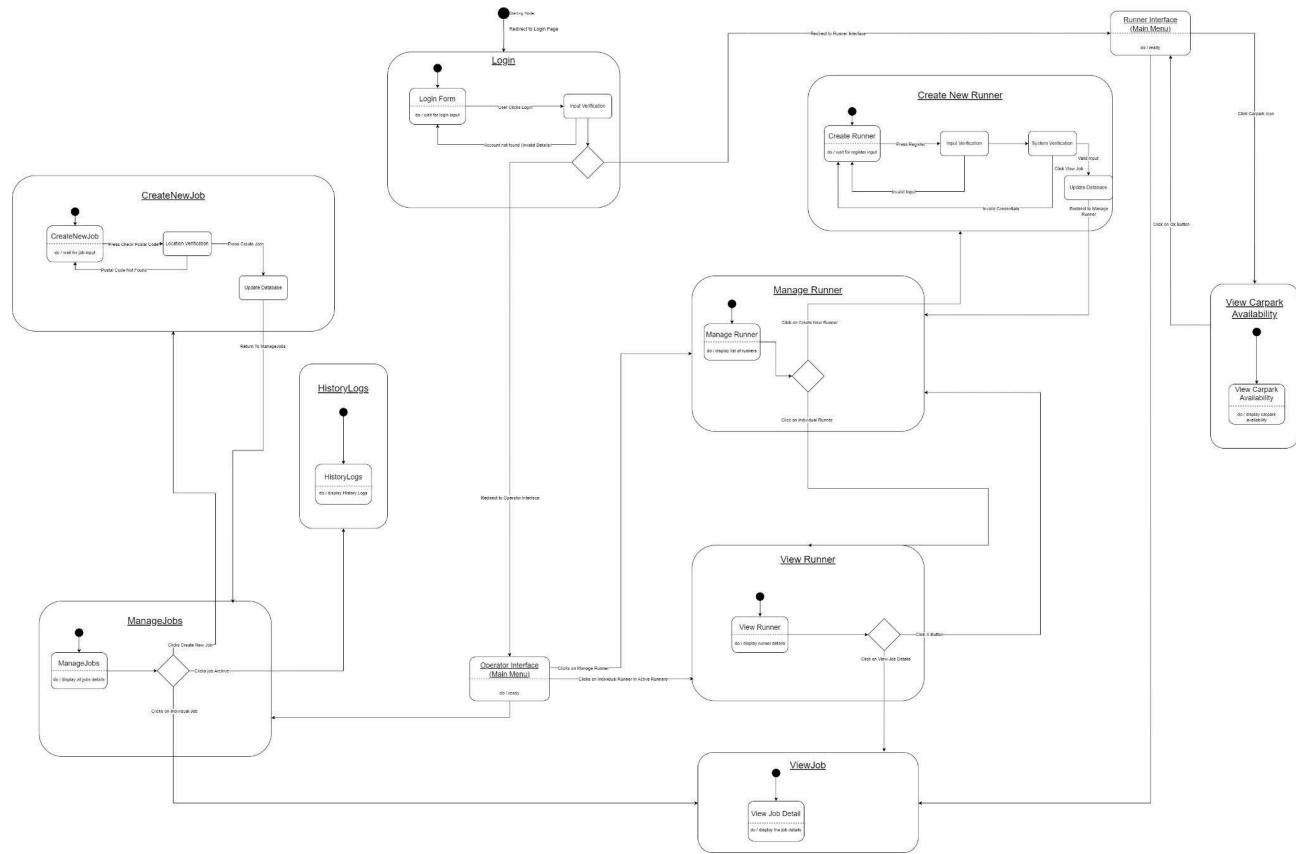
## ManageRunners



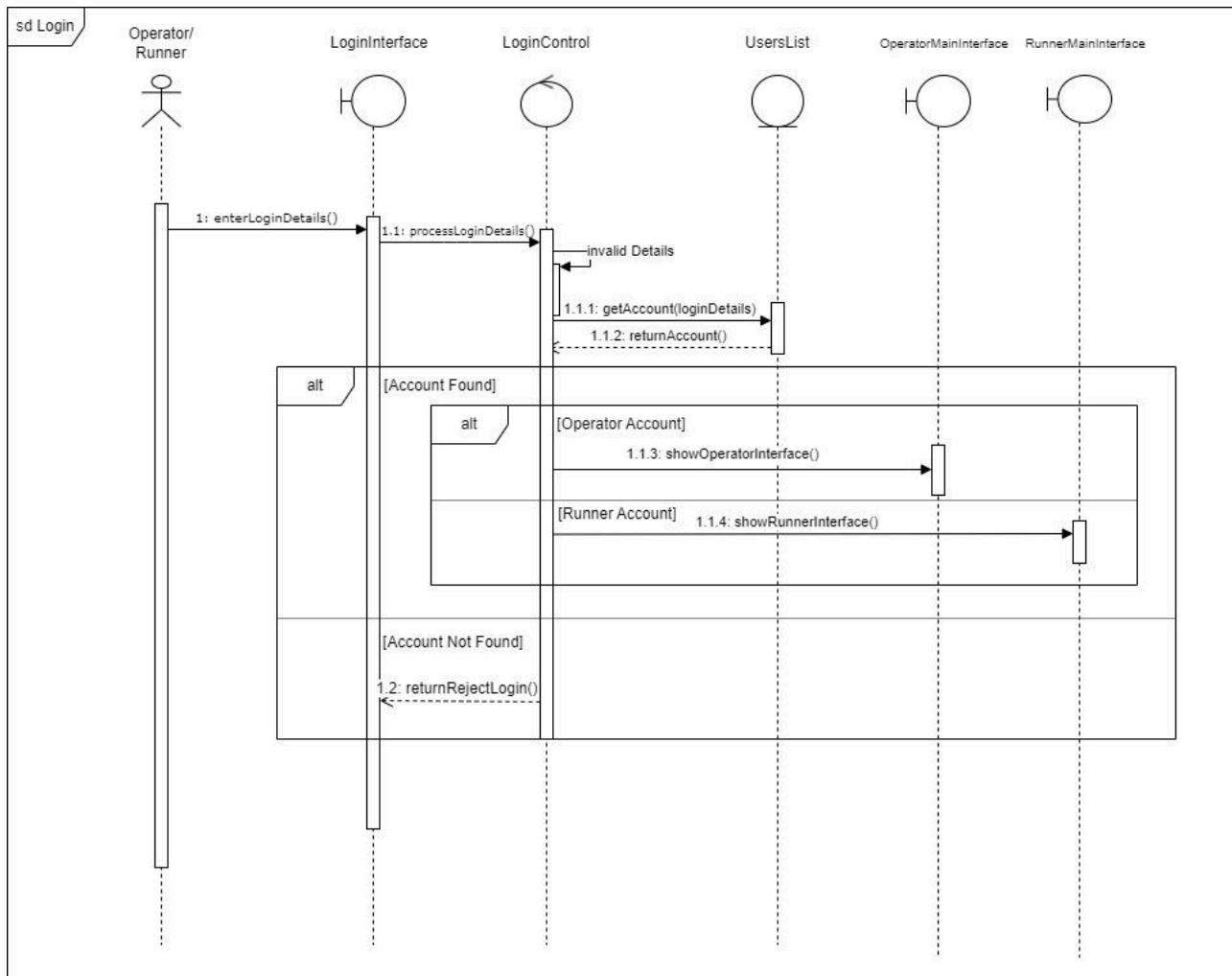
## HistoryLogs



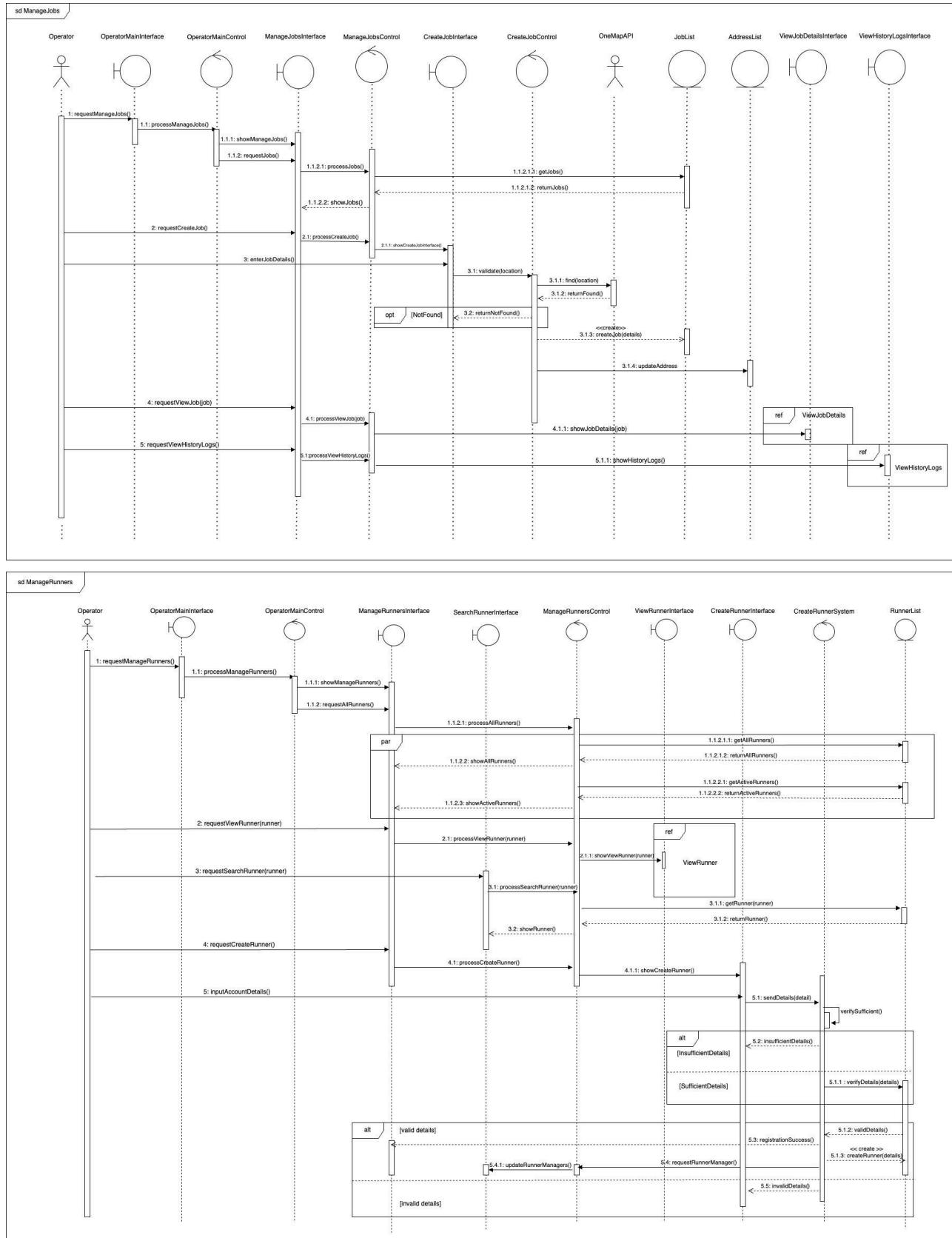
## Appendix D: Dialog Map



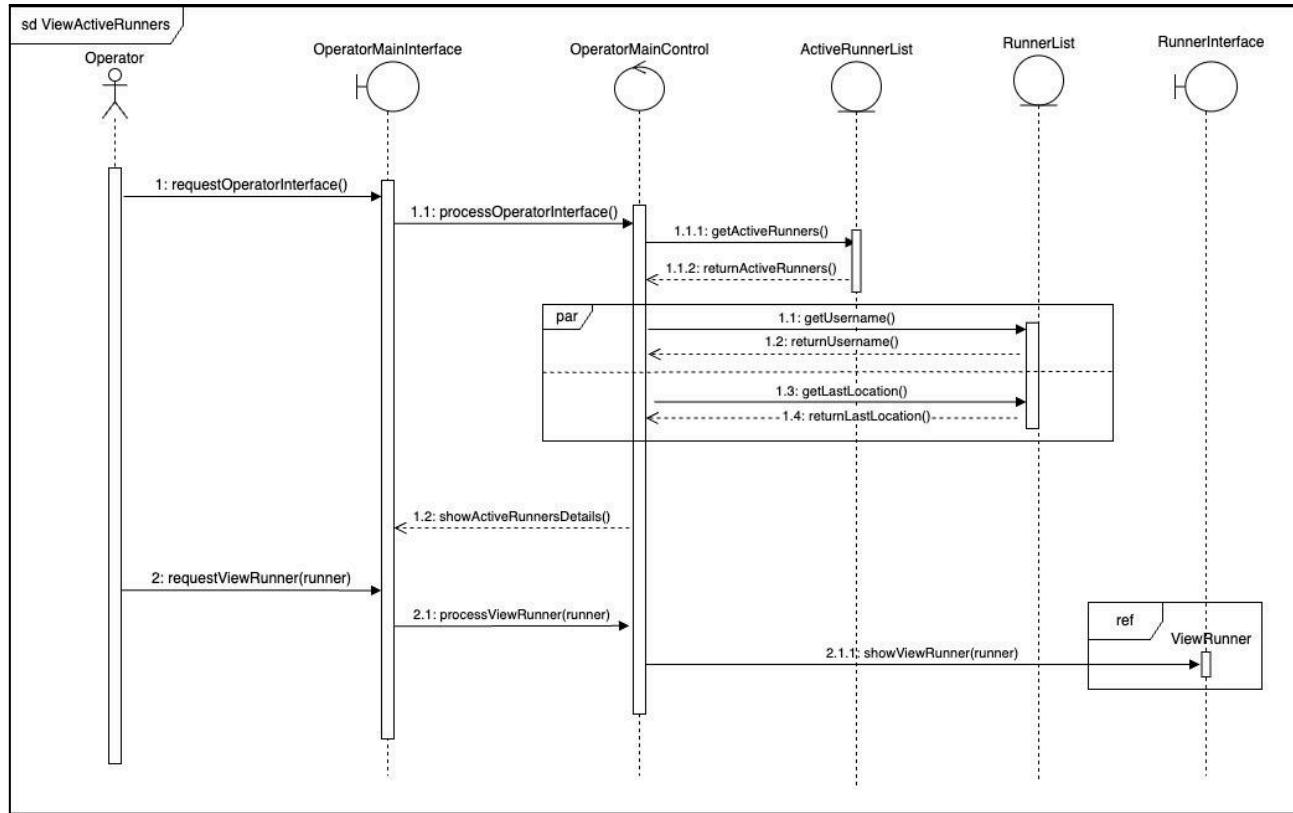
## Appendix E: Sequential Diagrams



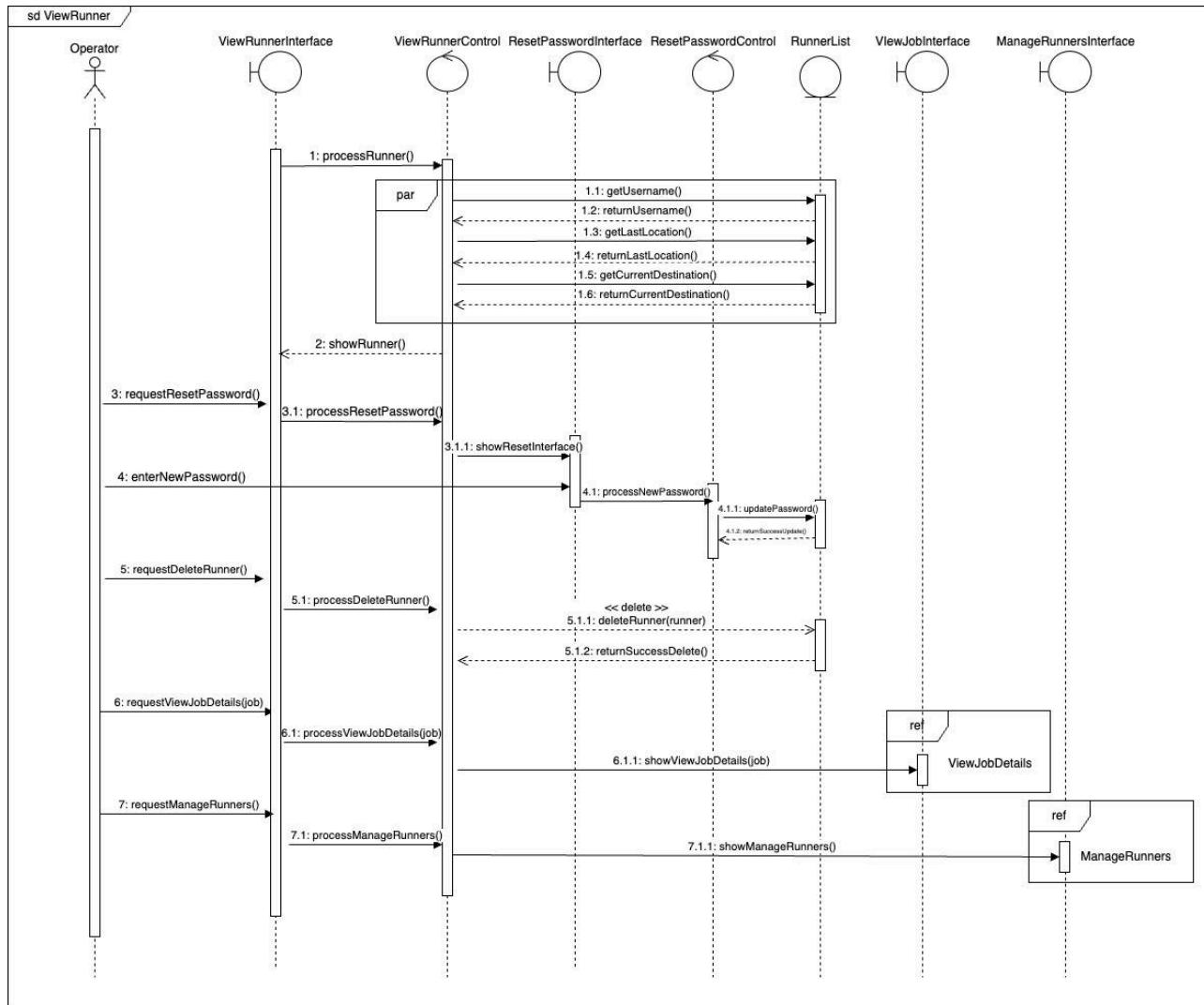
## Software Requirements Specification for RouteRunner



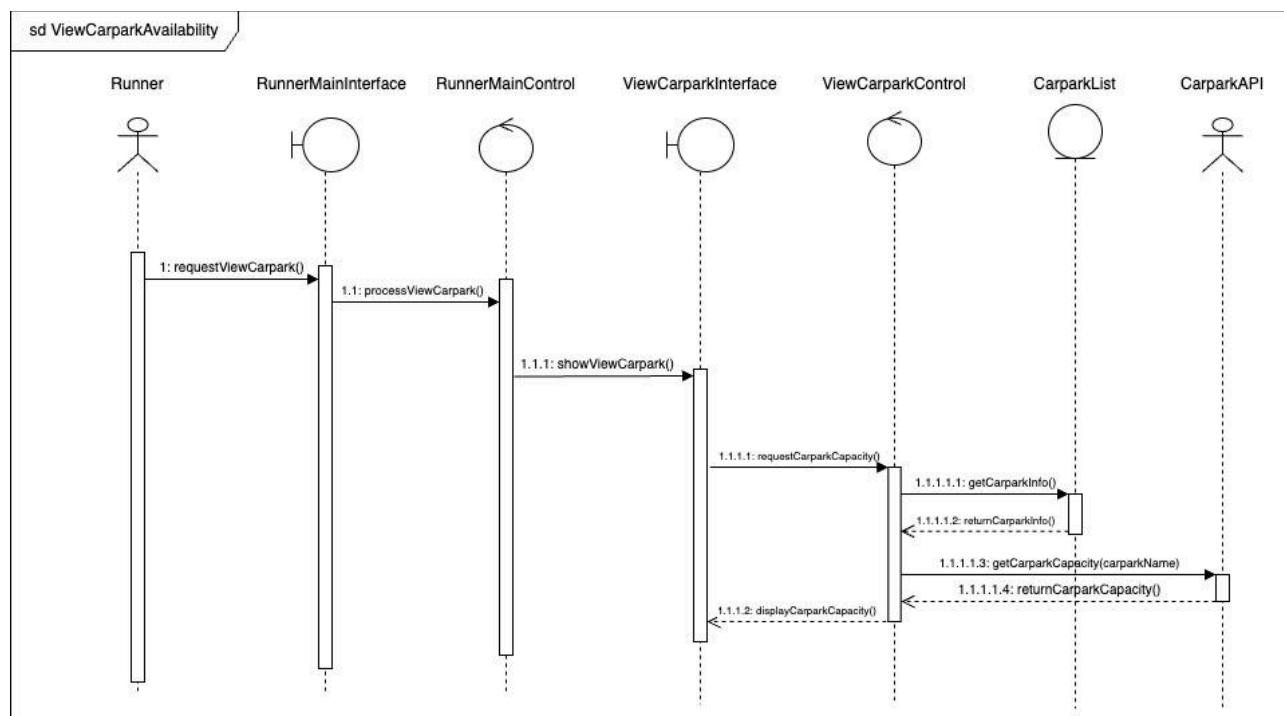
## Software Requirements Specification for RouteRunner



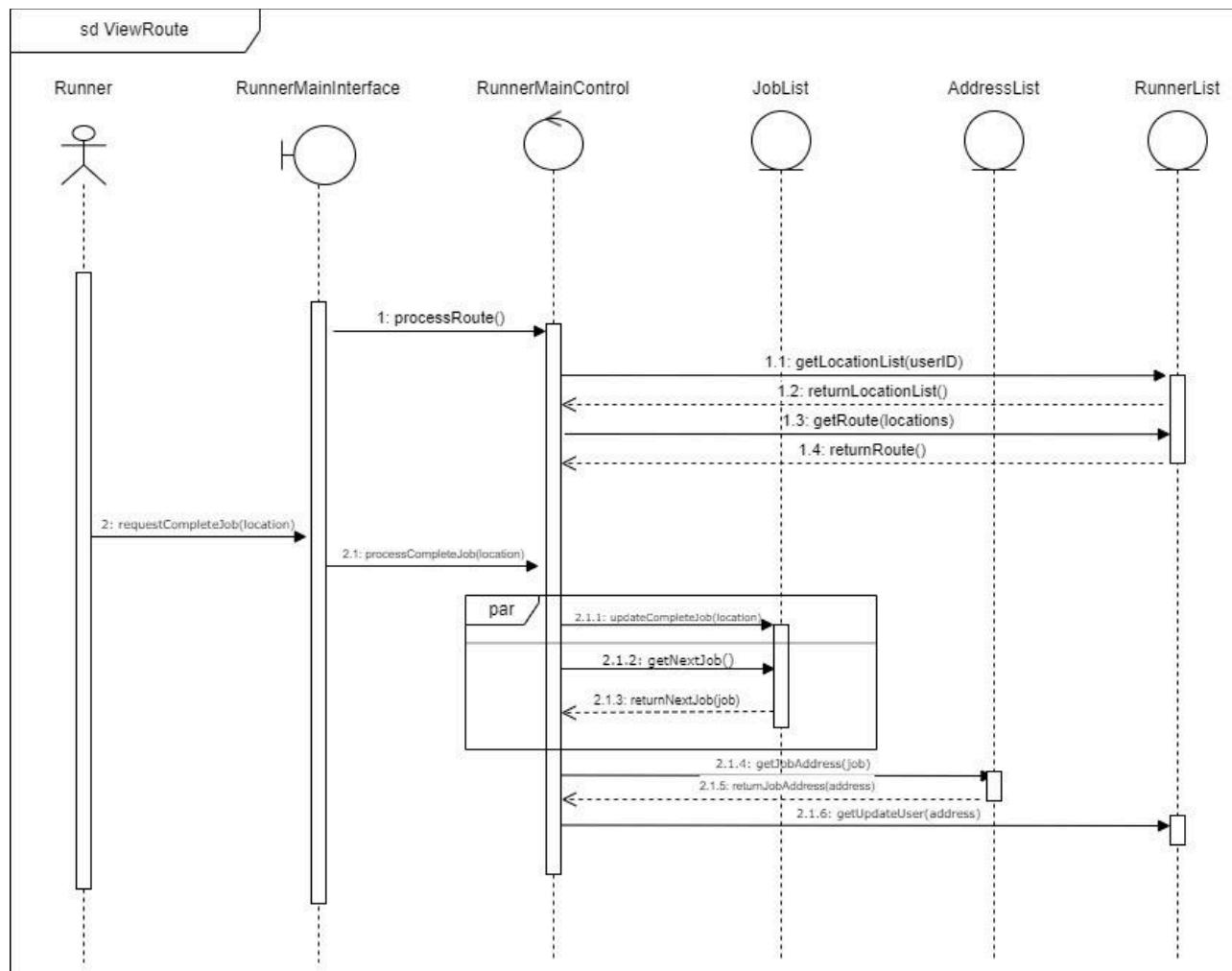
## Software Requirements Specification for RouteRunner



## Software Requirements Specification for RouteRunner



## Software Requirements Specification for RouteRunner



## Software Requirements Specification for RouteRunner

