**USE CASE DESCRIPTION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.ALL.1 | | |
| **Use Case Name:** | Login | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 8/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Runner / Operator |
| **Description:** | This use case allows System to authenticate the actor and bring them to their respective homepages. |
| **Preconditions:** | 1. The users have their accounts with username and password. |
| **Postconditions:** | 1. Actor has been authenticated. 2. Actors gain access to the respective System’s functions. |
| **Priority:** | High |
| **Frequency of Use:** | Medium |
| **Flow of Events:** | 1. Actor key in username and password 2. System cycles through database to validate actor’s credentials 3. If the actor's profile exists in the database, System returns a successful login. 4. System routes the actor to their respective homepage. |
| **Alternative Flows:** | **Incorrect credentials**   1. If the actor’s profile does not exist in the database, System displays error login and allows the actor to key in again**.**   **Incorrect password**   1. If the actor’s profile exist but the password is wrong, System displays incorrect password and only allows actor to key in for up to 3 more attempts    1. After 3 more incorrect attempts, System will lock the actor’s profile and send an email to the corresponding actor to reset the password. |
| **Exceptions:** | 1. Database storing the actor credentials is not available. |
| **Includes:** | NIL |
| **Special Requirements:** | 1. Operators and Runners must log into the system with a password consisting of at least 8 characters, and at least a lowercase letter, a capital letter and a special character. 2. All sensitive data, including user credentials and location data, must be encrypted in transit and at rest. 3. System must be compatible with both iOS and Android devices. |
| **Assumptions:** | 1. Operator’s email and credentials are provided to developers to add them into the system manually. |
| **Notes and Issues:** | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.ALL.1.1 | | |
| **Use Case Name:** | ResetPassword | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 8/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Runner / Operator |
| **Description:** | This use case allows the users to change their password. |
| **Preconditions:** | 1. The actor forgets the password. 2. The actor wants to change password for security reasons. |
| **Postconditions:** | 1. The password is updated on the user’s account. |
| **Priority:** | Low |
| **Frequency of Use:** | Low |
| **Flow of Events:** | 1. The actor wants to reset his password. 2. System sends the password reset email to the actor’s email. 3. The actor resets the password from the link sent in the email. |
| **Alternative Flows:** | NIL |
| **Exceptions:** | 1. The username does not exist. |
| **Includes:** | NIL |
| **Special Requirements:** | 1. Operators and Runners must log into the system with a password consisting of at least 8 characters, and at least a lowercase letter, a capital letter and a special character. 2. All sensitive data, including user credentials and location data, must be encrypted in transit and at rest. |
| **Assumptions:** | 1. Alteration to the actor profile database is transactional only. |
| **Notes and Issues:** | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.RN.1 | | |
| **Use Case Name:** | ViewJobList | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 8/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Runner |
| **Description:** | This use case allows the runner to see the locations of the current and subsequent assigned jobs. |
| **Preconditions:** | 1. Runner must be authenticated. 2. Runner must be active. |
| **Postconditions:** | 1. Runners will be able to view the next location address aside from the current location or empty page if there is no next location. |
| **Priority:** | High |
| **Frequency of Use:** | High |
| **Flow of Events:** | 1. Runner clicks on the address in the job list. 2. System displays the route of the requested address on the map. |
| **Alternative Flows:** | NIL |
| **Exceptions:** | 1. Google server is not available. |
| **Includes:** | NIL |
| **Special Requirements:** | 1. System shall be able to integrate seamlessly with external APIs, such as mapping services (e.g. Google Maps) for route optimization. |
| **Assumptions:** | NIL |
| **Notes and Issues:** | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.RN.2 | | |
| **Use Case Name:** | ViewRoute | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 8/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Runner |
| **Description:** | This use case allows the actor to see the route from the last location to the next location displayed by the System. |
| **Preconditions:** | 1. The Runner must be authenticated. 2. The Runner must be active. 3. The Runner must have a job allocated. |
| **Postconditions:** | 1. The Runner arrives to the location and updates completion status |
| **Priority:** | High |
| **Frequency of Use:** | High |
| **Flow of Events:** | 1. The Runner is assigned a location. 2. The System displays the target location and route on a map. 3. The Runner travels to the assigned location. 4. The Runner updates completion status. 5. The System assigns a new location to the Runner with a new route from the last location to the next location. |
| **Alternative Flows:** | **Higher priority job allocated**   1. If a new job of higher priority is allocated to this Runner, the System assigns a new location with higher priority to this Runner. |
| **Exceptions:** | 1. When the runner does not have the next location |
| **Includes:** | UpdateCompletionStatus |
| **Special Requirements:** | 1. The real-time locating system shall be able to scale from managing up to IoT devices without loss of data fidelity or monitoring capabilities. |
| **Assumptions:** | NIL |
| **Notes and Issues:** | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.RN.2.1 | | |
| **Use Case Name:** | UpdateCompletionStatus | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 10/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Runner |
| **Description:** | This use case allows the Runner to mark the current job as completed and remove the location from the job list. |
| **Preconditions:** | 1. The Runner must be authenticated. 2. The Runner must be active. 3. The Runner must have a job allocated. 4. The Runner must reach the assigned current location. |
| **Postconditions:** | 1. Next location will be moved up the queue and become the current location. |
| **Priority:** | High |
| **Frequency of Use:** | High |
| **Flow of Events:** | 1. Runner reaches the assigned location. 2. Runner sends an update request to the System. 3. The System shifts the next location to the current location. 4. The routes get updated from the completed location to the new location. |
| **Alternative Flows:** | **No new location available**   1. The System waits for the next location to be assigned to the Runner. |
| **Exceptions:** | 1. When the runner does not have the next location 2. The server encountered a bottleneck, temporarily preventing update requests from being processed |
| **Includes:** | NIL |
| **Special Requirements:** | 1. System must ensure 99.9% uptime, ensuring availability during peak business hours. 2. 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. |
| **Assumptions:** | NIL |
| **Notes and Issues:** | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.OP.1 | | |
| **Use Case Name:** | ManageJobs | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 10/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Operator |
| **Description:** | This use case allows the Operator to perform Create, Read, Update and Delete (CRUD) operations on the jobs database. |
| **Preconditions:** | 1. Operator must be authenticated. |
| **Postconditions:** | 1. Jobs database is accessed by the Operator. |
| **Priority:** | High |
| **Frequency of Use:** | High |
| **Flow of Events:** | 1. Operator wants to access with/without modification to the database. 2. System allows the Operator to view jobs, create jobs or remove jobs. 3. System displays currently assigned jobs in the list by default. |
| **Alternative Flows:** | **Operator wants to add job**   1. Operator choose to create jobs to be assigned to the Runners.   **Operator wants to delete job**   1. Operator choose to delete jobs that are no longer relevant or invalid. |
| **Exceptions:** | 1. Database storing the jobs is not available |
| **Includes:** | 1. RemoveJob 2. CreateJobs 3. ViewJobs |
| **Special Requirements:** | 1. System must allocate jobs to Runners within 3 seconds of the operator submitting the job. 2. System must ensure 99.9% uptime, ensuring availability during peak business hours. |
| **Assumptions:** | 1. Alterations to the database are transactional only. |
| **Notes and Issues:** | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.OP.1.1 | | |
| **Use Case Name:** | ViewJobs | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 8/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Operator |
| **Description:** | This use case allows the Operator to browse all jobs allocated to Runners and all jobs that the Runners are currently working on. |
| **Preconditions:** | 1. The Operator must be authenticated |
| **Postconditions:** | 1. System user interface displays all jobs allocated to Runners. |
| **Priority:** | Medium |
| **Frequency of Use:** | Medium |
| **Flow of Events:** | 1. The Operator clicks on the managed jobs. 2. System displays a list of jobs with a search filter if needed by the Operator to search for a specific job. 3. Operator performs a search about a specific job. 4. The System finds and displays the location details. |
| **Alternative Flows:** | **Operator searches for jobs**   1. Operator performs a search on the job based on the other attributes such as address etc. 2. System finds and displays the search result based on those other attributes. |
| **Exceptions:** | 1. No match found 2. Jobs do not exist in the Database |
| **Includes:** | NIL |
| **Special Requirements:** | 1. System must support up to 500 concurrent users without significant performance degradation to response time and service availability 2. System must ensure 99.9% uptime, ensuring availability during peak business hours. 3. All sensitive data, including user credentials and location data, must be encrypted in transit and at rest. 4. System must be compatible with both iOS and Android devices. |
| **Assumptions:** | NIL |
| **Notes and Issues:** | 1. There is a search bar and filter options to see the jobs keyed into the system. 2. Sorting function in a lexicographical order (optional) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.OP.1.2 | | |
| **Use Case Name:** | CreateJob | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 8/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Operator |
| **Description:** | This use case allows the operator to add new jobs to the list which will be allocated by the System. |
| **Preconditions:** | 1. The Operator must be authenticated. 2. The database must be available to add jobs. 3. There are new jobs available to be placed in the System. |
| **Postconditions:** | 1. New jobs are created in the System’s database. 2. New jobs are placed in the priority queue. 3. New jobs are deployed to the Runners. |
| **Priority:** | High |
| **Frequency of Use:** | High |
| **Flow of Events:** | 1. The Operator adds the address and priority level of the new job. 2. The System validates the address. 3. If the address is valid, the System selects the available Runners who have a shorter destination queue. 4. The System assigns the location to the Runner and rearranges the next location to be routed based on the higher priority location first. 5. The System routes to the next location once it receives an update completion status of the last location from the Runner. |
| **Alternative Flows:** | **Incorrect address**   1. If the address is invalid, the System alerts the Operator by displaying an error message. 2. The system prompts the Operator to reenter the address to be validated. |
| **Exceptions:** | 1. The address is not a local address 2. Although Jobs are added, there is no runner in the system 3. System is unable to craft a possible route |
| **Includes:** | 1. AllocateJob |
| **Special Requirements:** | 1. System must allocate jobs to Runners within 3 seconds of the operator submitting the job. 2. Location tracking updates must have a latency of no more than 2 seconds. 3. System must ensure 99.9% uptime, ensuring availability during peak business hours. 4. All sensitive data, including user credentials and location data, must be encrypted in transit and at rest. 5. System must enforce role-based access control (RBAC) to limit access to sensitive functionalities based on user roles. 6. System must log all user actions related to job allocation and location tracking for audit purposes. |
| **Assumptions:** | NIL |
| **Notes and Issues:** | When new jobs are added, an ID should be assigned for easy reference. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.OP.1.3 | | |
| **Use Case Name:** | RemoveJob | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 10/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Operator |
| **Description:** | This use case allows the Operator to remove locations from the list. |
| **Preconditions:** | 1. Operator must be authenticated. 2. Location must be in the list. |
| **Postconditions:** | 1. One or more jobs removed from the jobs database. |
| **Priority:** | High |
| **Frequency of Use:** | Low |
| **Flow of Events:** | 1. Operator searches and requests for deletion of a specific job. 2. System acknowledges the deletion request. 3. System proceeds with the removal of a specific job. 4. System alerts the Runner and updates the location list of the Runner responsible for the removed job. |
| **Alternative Flows:** | NIL |
| **Exceptions:** | 1. Removing jobs that have already been completed by the Runner but yet to be updated in the database. (i.e conflict between update completion status request and remove request) |
| **Includes:** | NIL |
| **Special Requirements:** | 1. System must ensure 99.9% uptime, ensuring availability during peak business hours. |
| **Assumptions:** | 1. Alterations to the database is transactional only. |
| **Notes and Issues:** | Suppose a job has been deleted, the Runner responsible for the job, must be notified of the deletion. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.OP.1.2.1 | | |
| **Use Case Name:** | ManualAllocateJob | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 10/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Operator |
| **Description:** | This use case allows the Operator to manually allocate jobs to the Runners. This action overrules the System’s automatic allocation. |
| **Preconditions:** | 1. Operator must be authenticated. 2. Operator wants to hand over the location to a more suitable Runner. |
| **Postconditions:** | 1. System hands over the destination from the current Runner to another selected Runner. |
| **Priority:** | Medium |
| **Frequency of Use:** | Low |
| **Flow of Events:** | 1. System initially assigns the location to a Runner. 2. Operator selects another Runner from the list of active Runners. 3. System acknowledges the manual overriding of the Runner assignment. 4. System updates accordingly. |
| **Alternative Flows:** | **Selecting the same runner**   1. If the Operator selects the same Runner from the list of active Runners, the System does nothing. |
| **Exceptions:** | NIL |
| **Includes:** | NIL |
| **Special Requirements:** | NIL |
| **Assumptions:** | NIL |
| **Notes and Issues:** | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.OP.2 | | |
| **Use Case Name:** | TrackActiveRunners | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 10/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Operator |
| **Description:** | This use case allows the Operator to see the active Runner’s current destination. |
| **Preconditions:** | 1. Operator must be authenticated. |
| **Postconditions:** | 1. System user interface displays information about each active Runner. |
| **Priority:** | High |
| **Frequency of Use:** | High |
| **Flow of Events:** | 1. System displays a list of Runners currently active. 2. Operator clicks on a specific Runner for more details. 3. System displays the information of a Runner such as last location, current destination and Runner’s full itinerary. |
| **Alternative Flows:** | NIL |
| **Exceptions:** | NIL |
| **Includes:** | NIL |
| **Special Requirements:** | 1. All sensitive data, including user credentials and location data, must be encrypted in transit and at rest. |
| **Assumptions:** | NIL |
| **Notes and Issues:** | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.OP.3 | | |
| **Use Case Name:** | ManageRunners | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 8/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Operator |
| **Description:** | This use case allows the Operator to perform Create, Read, Update and Delete (CRUD) operations on the Runners database. |
| **Preconditions:** | 1. Operator must be authenticated |
| **Postconditions:** | 1. Runners database is accessed by the Operator. |
| **Priority:** | Medium |
| **Frequency of Use:** | Low |
| **Flow of Events:** | 1. Operator wants to access with/without modification to the Runner database. 2. System allows the Operator to view, create or remove Runners 3. System displays a list of Runners and whether each Runner is active. |
| **Alternative Flows:** | NIL |
| **Exceptions:** | 1. Database storing the Runners’ information is not available. |
| **Includes:** | 1. CreateRunner 2. RemoveRunners 3. ViewRunners |
| **Special Requirements:** | 1. All sensitive data, including user credentials and location data, must be encrypted in transit and at rest. 2. System must enforce role-based access control (RBAC) to limit access to sensitive functionalities based on user roles. |
| **Assumptions:** | NIL |
| **Notes and Issues:** | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.OP.3.1 | | |
| **Use Case Name:** | ViewRunners | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 10/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Operator |
| **Description:** | This use case allows the Operator to see the details of the runner which includes the activity and a list of locations assigned to the Runner. |
| **Preconditions:** | 1. Operator must be authenticated. |
| **Postconditions:** | 1. Operator have the knowledge of the current destination of the specific runner and his next few locations. |
| **Priority:** | Medium |
| **Frequency of Use:** | Low |
| **Flow of Events:** | 1. System displays a list of Runners. 2. Operator clicks on each Runner for more details. 3. System displays the last location, current destination and locations lists of the Runner. |
| **Alternative Flows:** | **Use of search filter**   1. If the Operator is unable to find Runner by scrolling, the Operator uses a search filter to find the specific Runner’s details. |
| **Exceptions:** | 1. Runner does not exist after the Operators attempts to use the search filter. |
| **Includes:** | NIL |
| **Special Requirements:** | NIL |
| **Assumptions:** | NIL |
| **Notes and Issues:** | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.OP.3.2 | | |
| **Use Case Name:** | RemoveRunners | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 10/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Operator |
| **Description:** | This use case allows the removal of Runner’s records in the Runner database. |
| **Preconditions:** | 1. Operator must be authenticated. |
| **Postconditions:** | 1. One or more Runner records are deleted from the Runner database. |
| **Priority:** | Medium |
| **Frequency of Use:** | Low |
| **Flow of Events:** | 1. Operator searches and requests for deletion of a specific Runner record. 2. System acknowledges the deletion request. 3. System proceeds with the removal of a specific Runner record. 4. System notifies the Operator that the deletion is successful. |
| **Alternative Flows:** | 1. For active Runners records, System deauthenticated the Runner and forces log off on the active Runner’s device. |
| **Exceptions:** | NIL |
| **Includes:** | NIL |
| **Special Requirements:** | NIL |
| **Assumptions:** | 1. Alterations to the database is transactional only |
| **Notes and Issues:** | Suppose an active Runner is deleted, the jobs currently assigned to the runner must be reallocated to other active Runner. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | SYS.OP.3.3 | | |
| **Use Case Name:** | CreateRunner | | |
| **Created By:** | Alvin & Kan Yui | **Last Updated By:** | Alvin & Kan Yui |
| **Date Created:** | 10/9/24 | **Date Last Updated:** | 10/9/24 |

|  |  |
| --- | --- |
| **Actor:** | Operator |
| **Description:** | This use case allows the Operator to create new Runner profiles to be inserted into the Runner database. |
| **Preconditions:** | 1. Operator must be authenticated. |
| **Postconditions:** | 1. One or more Runner records are created in the Runner database. |
| **Priority:** | Medium |
| **Frequency of Use:** | Medium |
| **Flow of Events:** | 1. Operator registers the new Runner’s credentials (i.e username, email and password. 2. Default password is entered by the operator which can be changed by the Runner in the event of usage. 3. System acknowledges entry by the Operator. 4. System updates the Runner database by inserting the newly created Runner record. |
| **Alternative Flows:** | **Weak password**   1. If the password is not secure enough, System prompts the Operator to re-enter the default password. |
| **Exceptions:** | 1. Any input field left blank. |
| **Includes:** | NIL |
| **Special Requirements:** | 1. New account passwords should be contain    1. at least 10 characters long,    2. a mixture of upper and lower case letters,    3. a special case character |
| **Assumptions:** | NIL |
| **Notes and Issues:** | Generation of Unique ID should be randomly generated and checked that it does not exist before the System assigns to new Runner. |