# System and Software Architecture Description (SSAD)

**LADOT Scanning**

**Team 08**

| Name | Primary Role | Secondary Role |
|---|---|---|
| Anirudh Govil | Project Manager | Life Cycle Planner |
| Jeffrey Colvin | Prototyper | Systems and Software Architect |
| Aditya Kumar | Feasibility Analyst | Project Manager |
| Nisheeth Joshi | Systems and Software Architect | Life Cycle Planner |
| Niraj Brahmkhatri | Operational Concept Engineer | Requirements Engineer |
| Corey Painter | IIV&V | Shaper |

# Version History

| Date | Author | Version | Changes made | Rationale |
|---|---|---|---|---|
| 10/09/11 | JC | 1.0 | • Original copy using with Instructional ICM-Sw v1.0 template | • Initial Draft |
| 10/10/11 | Nisheeth Joshi | 1.1 | • Updated section 2.0 template | • Initial Draft |
| 10/16/11 | NJ | 1.2 | • Updated incorrect text associated with diagrams | • TA Grading |
| 10/10/11 | JC | 2.0 | • Updated diagrams | • Diagrams misrepresented system and were not detailed enough |
| 10/23/11 | JC | 3.0 | • Modified diagrams | • Diagrams needed corrections due to rescoping/ARB session comments |
| 10/24/11 | JC | 3.1 | • Added future considerations<br>• Fixed some typos | • Needed to meet requirements |
| 10/27/11 | NJ | 3.2 | • Fixed some common error | • Changes as per the bugs marked by den student |
| 11/21/11 | JC | 3.3 | • Merged Models with <<evolutionary>> tags<br>• Updated Use Case descriptions with more in depth information<br>• Modified use case descriptions | • Feedback from DCP<br>• Requirement changes |
| 12/05/11 | JC | 3.4 | • Added xml tag specification | • Feedback from ARB |

# Table of Contents

# Table of Tables

# Table of Figures

# 1. Introduction

## 1.1 Purpose of the SSAD

The SSAD will outline the overall object-oriented analysis and design (referred to as OOA&D) of the system being developed. The system, once developed, should be faithful to the architecture defined in the SSAD and the builders (programmers) should use this document as a reference to the system architecture. Additionally, the SSAD may be used by the maintainer and clients to understand the structure of the system after the system has been delivered.

## 1.2 Status of the SSAD

This version of the SSAD specified the XML tags that are associated with the prototype data files as well as value specifications.

# 2. System Analysis

## 2.1 System Analysis Overview

The purpose of the new LADOT Scanning System is to eliminate the current document scanning process and to create a new electronic system to help maintain both time-keeping and maintenance reports for LADOT field workers. The LADOT Electronic Field System will be stored on LADOT trucks and will log job status (including time stamps) as well as notes about the job such as what the problem was and what equipment was needed to fix it. The software must only keep track of maintenance reports.

Additionally, an evolutionary web system will support helping with task assignment by showing maps with tasks and trucks. There will also be pages to review maintenance reports by specific queries.

### 2.1.1 System Context

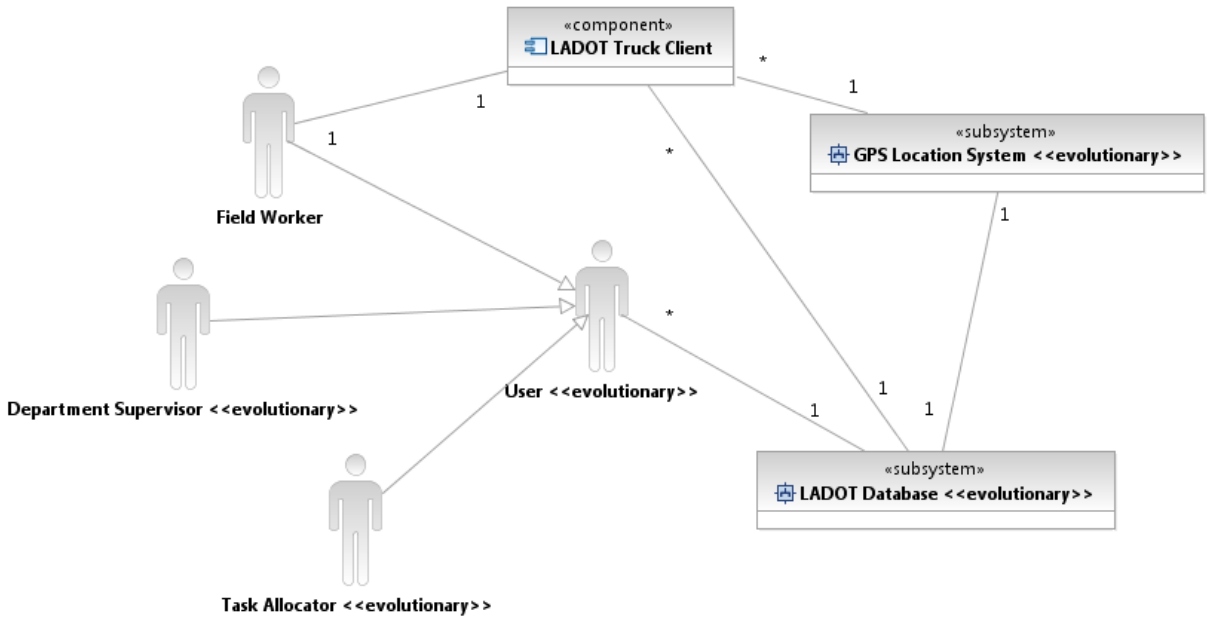


**Figure 1 - System Context Diagram**

**Table 1 - System Context Diagram Details**

| Actor | Description | Responsibilities |
|---|---|---|
| User | A super-type | A generalized actor for log-in on the |

| | | |
|---|---|---|
| | | computer system |
| Task Allocator | Person who will be choosing which tasks to allocate to certain trucks | Allocating tasks to trucks |
| Department Supervisor | A supervisor needing to create maintenance report logs | Query the database for logs |
| Field Worker | Employees who go to the field and do the work | Log job reports<br>View old logs |

## 2.1.2 Artifacts & Information

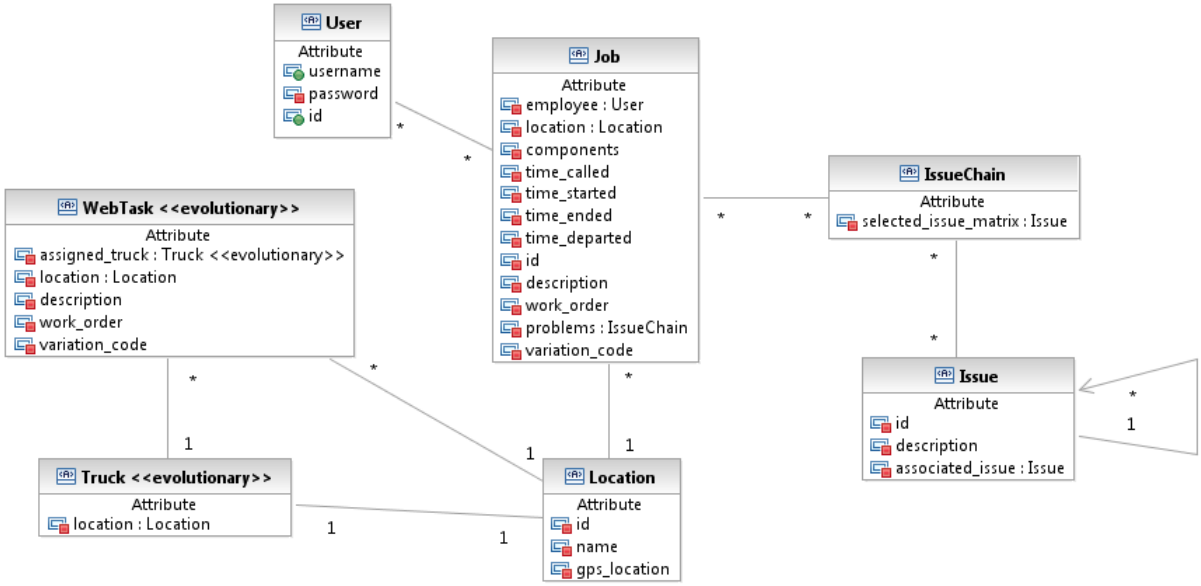


**Figure 2 - Artifacts and Information Diagram**

**Table 2 - Artifacts and Information Diagram Details**

| Artifact | Purpose |
|---|---|
| Issue | This keep a note of one of the issues that were resolved during a job (ex: broken red light on east-facing lamp) |
| Job | A representation of a job being performed by a field worker after it has been assigned |
| Truck | A representation of a truck in the field, including it's location |
| User | A user of any of the systems (either truck of computer system) |

| | |
|---|---|
| | depending on their *permission* attribute |
| IssueChain | This stores a complete set of issues in a list of lists, and is a representation of the depth of sub issues in relation to each other |
| Location | Either a job location (28th and Figuroa) or the location of a truck, depending on the owner |
| Web Task | A representation of a task that needs assigning (note: this is not linked to a Job as trucks are not guaranteed to have internet connectivity. Field Worker's must manually input task data into the computer system Job artifact) |

#### 2.1.2.1 Artifacts XML Summary

This section will outline the tags used in the XML files read and generated by the application. They link to the columns a database should contain.

Fields marked as "(unique)" must be unique for the program to run correctly. When a new value is added, it should be a non-existing number otherwise old data will no longer be correct.

##### 2.1.2.1.1 IssueCodes.xml File Format

| Tag | Description | Contains |
|---|---|---|
| *DocumentElement* | An overall container tag | *T_ISSUE_CODES* |
| *T_ISSUE_CODES* | A row definition for an issue | *id*, *description*, *checkbox* |
| *id* | The issue codes id | Integer (unique) |
| *description* | The issue codes description | String |
| *checkbox* | Whether or not this issue is part of a list of issues that allow multiple selections | Boolean (0 or 1) for true false |

##### 2.1.2.1.2 IssueRelations.xml File Format

| Tag | Description | Contains |
|---|---|---|
| *DocumentElement* | An overall container tag | *T_ISSUE_RELATIONS* |
| *T_ISSUE_RELATIONS* | A row definition for a relationship | *parent_id*, *child_id* |
| *parent_id* | The id of the issue code that acts as the parent | Integer (should be identical to an id in the Issue Codes table) |
| *child_id* | The id of the issue code that acts as the child | Integer (should be identical to an id in the Issue Codes table) |

##### 2.1.2.1.3 Locations.xml File Format

| Tag | Description | Contains |
|---|---|---|
| *DocumentElement* | An overall container tag | *T_LOCATIONS* |
| *T_LOCATIONS* | A row definition for a location | *location_code*, *hi_axis*, *low_axis* |

| | | |
|---|---|---|
| *location_code* | The location code | Integer (unique) |
| *hi_axis* | The first cross-street | String |
| *low_axis* | The second cross-street | String |

#### 2.1.2.1.4 UserTable.xml File Format

| Tag | Description | Contains |
|---|---|---|
| *DocumentElement* | An overall container tag | *T_USER* |
| *T_USER* | A row definition for an user | *user_id*, *user_password*, *user_name* |
| *user_id* | The user's id | Integer (unique) |
| *user_password* | The user's password | String |
| *user_name* | The user's display name | String |

#### 2.1.2.1.5 VariationCodes.xml File Format

| Tag | Description | Contains |
|---|---|---|
| *DocumentElement* | An overall container tag | *T_VARIATION_CODES* |
| *T_VARIATION_CODES* | A row definition for a variation code | *code* |
| *code* | The variation code | Integer (unique) |

#### 2.1.2.1.6 WorkOrderCodes.xml File Format

| Tag | Description | Contains |
|---|---|---|
| *DocumentElement* | An overall container tag | *T_WORK_ORDERS* |
| *T_WORK_ORDERS* | A row definition for a work order | *code* |
| *code* | The work order code | Integer (unique) |

#### 2.1.2.1.7 <Jobs>.xml File Format

Multiple files contain job definitions. This file should never be hand-generated, it is created by the application.

| Tag | Description | Contains |
|---|---|---|
| *DocumentElement* | An overall container tag | *T_JOB* |
| *T_JOB* | A row definition for a job | *user_id*, *work_order_id*, *time_called*, *time_arrived*, *time_completed*, *time_departed*, *variation_code*, *unique_id*, *location_code* |
| *user_id* | The performer's user id | Integer (links to user table) |
| *work_order_id* | The job's work order id | Integer (links to work order table, *-1 if unassigned*) |
| *time_called* | The time call was received | DateTime |
| *time_arrived* | The time arrived at job location | DateTime (*DateTime.Min if* |

| | | |
|---|---|---|
| | | *unassigned)* |
| *time_completed* | The time job was completed | DateTime (*DateTime.Min if unassigned)* |
| *time_departed* | The time job was departed | DateTime (*DateTime.Min if unassigned)* |
| *variation_code* | The variation code | Integer (links to work order table, *-1 if unassigned*) |
| *unique_id* | A unique id for the job | GUID (unique) |
| *location_code* | The location of the job | Integer (links to location table, *-1 if unassigned*) |

#### 2.1.2.1.8 <LoggedIssue>.xml File Format

Multiple files contain logged issue definitions. This file should never be hand-generated, it is created by the application.

| Tag | Description | Contains |
|---|---|---|
| *DocumentElement* | An overall container tag | *T_LOGGED_ISSUE* |
| *T_LOGGED_ISSUE* | A row definition for a logged issue | *issue_user_id*, *issue_job_id*, *issue_depth_int*, *issue_id*, *issue_chain_id* |
| *issue_user_id* | The user who logged the issue | Integer (links to user table) |
| *issue_job_id* | The job the issue is associated with | GUID (links to job table) |
| *issue_depth_int* | The depth of the issue in relation to the chain | Integer |
| *issue_id* | The issue represented | Integer (links to issue table) |
| *issue_chain_id* | The id of the chain | GUID (shared with other issues in same chain) |

#### 2.1.2.1.9 <LoggedIssueComments>.xml File Format

Multiple files contain logged issue comment definitions. This file should never be hand-generated, it is created by the application.

| Tag | Description | Contains |
|---|---|---|
| *DocumentElement* | An overall container tag | *T_LOGGED_ISSUE_ COMMENTS* |
| *T_LOGGED_ISSUE_ COMMENTS* | A row definition for a issue chain comment | *job_code, comment, issue_chain_id* |
| *job_code* | The job the comment is associated with | GUID (links with job table) |
| *comment* | The comment | String (*empty string if unassigned*) |
| *issue_chain_id* | The issue chain this comment is associated with | GUID (links with logged issue table) |

## 2.1.3 Behavior

### 2.1.3.1 Use-Case Model

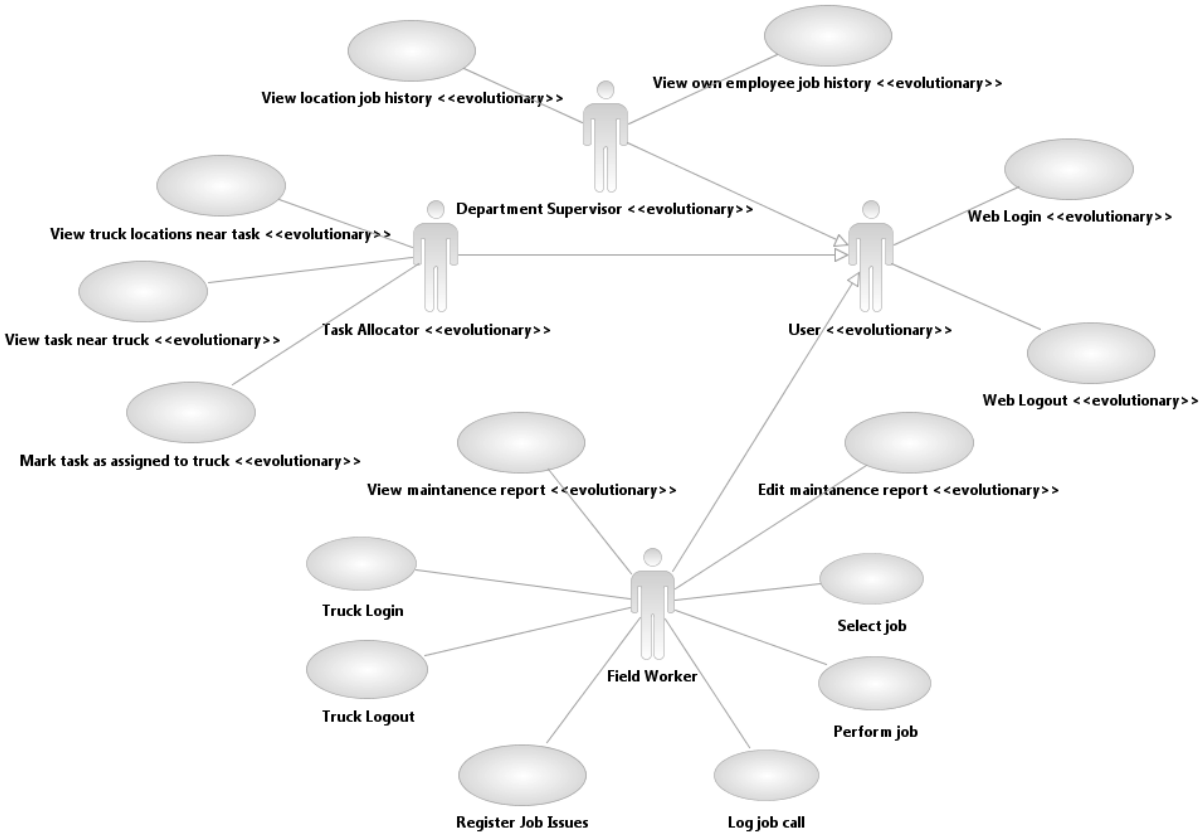


**Figure 3 - Truck System Use-Case Model**

#### 2.1.3.1.1 Log job call

| | |
|---|---|
| **Identifier** | UC-01: Log job calls |
| **Purpose** | Field worker needs to log that they were called with a job |
| **Requirements** | WC_523 Time stamp maintenance events |
| **Pre-conditions** | Must be using the truck system |
| **Post-conditions** | New call is added to the unregistered job queue |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Select "Log Call" button | |
| 2 | | Generate new empty "job" and adds to un-registered job list |
| 3 | | Mark "call received" time as current time |

#### 2.1.3.1.2 Perform Job

| | |
|---|---|
| **Identifier** | UC-02: Perform job |
| **Purpose** | Field worker needs to be able to log that they have performed a job |
| **Requirements** | WC_536 Standardize tasks to be entered; WC_523 Time stamp maintenance events |
| **Pre-conditions** | Must be logged into the truck system; must have job registered; must not be performing another job; must have been selected from UC-03 (Select job) |
| **Post-conditions** | Job is completed (has departure time logged) with issues and is stored to completed job list |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Select "Log Arrival" upon arrival | |
| 2 | | Time-stamp arrival in job |
| 3 | Select "Log Completion" upon completing | |
| 4 | | Time-stamp completion |
| 5 | Ensure "Register Job Issues" has been performed at least once | |
| 6 | Select "Log Departure" when departing | |
| 7 | | Time-stamp departure and close job |

#### 2.1.3.1.3 Select Job

| | |
|---|---|
| **Identifier** | UC-03: Select job |
| **Purpose** | Field worker needs to be able to choose which job to do |
| **Requirements** | WC_1048 and WC_536 Retrieval of codes and Standardize tasks to be entered |
| **Pre-conditions** | Must be logged in; must not be performing another job; must have registered jobs in queue |
| **Post-conditions** | Job is registered as being performed |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | | Load list of registered job in queue |
| 2 | Select job from queue | |
| 3 | Select "begin job" button | |
| 4 | | Register job as started, transfer to Perform Job page |

#### 2.1.3.1.4 Truck Login

| | |
|---|---|
| **Identifier** | UC-04: Truck Login |
| **Purpose** | Field worker needs to be able to log-in to perform jobs |
| **Requirements** | WC_521 Log in |
| **Post-conditions** | User is logged into the system |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Enter username and password | |
| 2 | Submit data | |
| 3 | | Validate username/password |
| | | Time-stamp beginning of user session |
| 4 | | Store user data in code until logout |

#### 2.1.3.1.5 Truck Logout

| | |
|---|---|
| **Identifier** | UC-05: Truck Logout |
| **Purpose** | Need to be able to log-out to have other users sign on and to export data |
| **Requirements** | WC_521 Log out |
| **Pre-conditions** | Must be logged in to the truck system |
| **Post-conditions** | User is logged out of the system |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Click logout button | |
| 2 | | Time-stamp end of user session |
| 3 | | Save user session until exported |

#### 2.1.3.1.6 Register Job Issues

| | |
|---|---|
| **Identifier** | UC-06: Register Job Issues |
| **Purpose** | Field worker must be able to register what issues arised at the job |
| **Requirements** | WC_536 Standardize tasks to be entered |
| **Pre-conditions** | Must be logged in; must be performing a job |
| **Post-conditions** | New issue chain is added to the current job |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Select "Log Job Issue" from the job performance page | |
| 2 | | Load the log issue popup |
| 3 | | Load the entry set of issues into the popup |
| 4 | Select an issue(s) from either the dropdown or list | |
| 5 | | Load set of child issues (if exists) |
| 6 | Repeat 4&5 until entire chain has been created | |
| 7 | Select "Save Issue" | |
| 8 | | Create issue chain and save to current job |
| 9 | | Return to perform job page |

#### 2.1.3.1.7 Evolutionary Use Case Information

The evolutionary use cases are not tied to specific requirements and as such contain no "requirements" information. These evolutionary clauses are aspects that were taken into account during the development of the project, but must be re-assessed when continued with future development.

##### 2.1.3.1.7.1 View location job history

| | |
|---|---|
| **Identifier** | UC-07: View location job history |
| **Purpose** | Review the jobs that have been performed at a specific location |
| **Pre-Conditions** | Must be logged into the web system with Department Superisor permissions |
| **Post-Conditions** | User is allowed to view the history of jobs performed at the location |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Enter location id | |
| 2 | Click "view" | |
| 3 | | Validate id existence |
| 4 | | Query database for jobs with location id |
| 5 | | Return table of jobs performed on location chronologically |

##### 2.1.3.1.7.2 View Own Employee Job History

| | |
|---|---|
| **Identifier** | UC-08: View Own Employee Job History |
| **Purpose** | Review job performance by a certain individual |
| **Pre-Conditions** | Must be logged into the system with Department Superisor permissions |
| **Post-Conditions** | User is allowed to view the history of jobs performed by the |

| | |
|---|---|
| | employee |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Enter employee id | |
| 2 | Click "view" | |
| 3 | | Validate id existence |
| 4 | | Validate user permission for id |
| 5 | | Query database for jobs completed by employee id |
| 6 | | Return table of jobs performed by employee chronologically |

#### 2.1.3.1.7.3 Web Login

| | |
|---|---|
| **Identifier** | UC-09: Web Login |
| **Purpose** | To log in to the online system |
| **Post-Conditions** | User is logged into the online system |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Enter username and password | |
| 2 | Submit data | |
| 3 | | Validate username/password |
| 4 | | Store user data as encrypted cookie until log out |

#### 2.1.3.1.7.4 Web Logout

| | |
|---|---|
| **Identifier** | UC-10: Web Logout |
| **Purpose** | To log out of the online system |
| **Pre-conditions** | Must be logged into the online system |
| **Pre-conditions** | User is logged out of the online system |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Click logout button | |
| 2 | | Remove user cookie |

#### 2.1.3.1.7.5 View Maintenance Report

| | |
|---|---|
| **Identifier** | UC-11: View maintenance report |
| **Purpose** | Field worker can review their maintenance reports |
| **Pre-conditions** | Must be logged in to the online system with field worker permissions |
| **Post-conditions** | User can review a specific maintenance report |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Select report to view from list | |
| 2 | Click "View" button | |
| 3 | | Displays read-only job summary until "Back" or "Edit" button is selected |

#### 2.1.3.1.7.6 Edit Maintenance Report

| | |
|---|---|
| **Identifier** | UC-12: Edit maintenance report |
| **Purpose** | Employees must be able to edit reports in case of mistakes |
| **Pre-conditions** | Must be logged in to the online system with field worker permissions |
| **Post-conditions** | Maintenance report is updated and stored to database |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Select "Edit" button on view report page | |
| 2 | | Display job page with editable fields |
| 3 | Edit fields in need of changing | |
| 4 | Click "Submit" button | |
| 5 | | Update changed fields in job database |
| 6 | | Change page to "view report" page |

#### 2.1.3.1.7.7 View Truck Map Near Task

| | |
|---|---|
| **Identifier** | UC-13: View Truck Map Near Task |
| **Purpose** | Allocator needs to be view where trucks are in relation to a task |
| **Pre-conditions** | Must be logged in to the online system with task allocator permissions |
| **Post-conditions** | User can see a map and list of trucks in relation to the task |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Select an task from the map or task list | |
| 2 | | Hide all other tasks from map |
| 3 | | Replace task list with a full list of trucks, sorted by distance from task |

#### 2.1.3.1.7.8 View Task Map Near Truck

| | |
|---|---|
| **Identifier** | UC-14: View Task Map Near Truck |
| **Purpose** | Allocator needs to be view where tasks are in relation to a truck |
| **Pre-conditions** | Must be logged in to the online system with task allocator permissions |
| **Post-conditions** | User can see a map and list of tasks in relation to the truck |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Select an truck from the map or truck list | |
| 2 | | Hide all other trucks from map |
| 3 | | Replace truck list with a full list of tasks, sorted by distance from truck |

#### 2.1.3.1.7.9 Mark Task as Assigned to Truck

| | |
|---|---|
| **Identifier** | UC-15: Mark Task as Assigned to Truck |
| **Purpose** | Allocator needs to be able to "assign" a task (note: this is not linked to the truck, as you cannot guarantee the trucks internet connectivity. The allocator still must place a call to the field worker, where they will input the call) |
| **Pre-conditions** | Must be logged in to the online system with task allocator permissions |
| **Post-conditions** | The task is removed from the list of current unassigned tasks |

| Seq# | Actor's Action | System's Response |
|---|---|---|
| 1 | Select a task from the truck map or a truck from the task map | |
| 2 | | Change the page to the confirmation screen |
| 3 | Select "confirm allocation" | |
| 4 | | Remove task from task list |