THE UNIVERSITY of ADELAIDE

School of Mathematical and Computer Sciences
COMP SCI 3006 - Software Engineering & Project

Software Requirements Specification

Lunar Rover Mapping Robot

Revision 1.0

28th October 2017

Prepared by UG06

Philip Fai-Yuen Sung
a1211805

Brent Aaron Poland
a1647000

Christopher Luke Lyndon
a1694039

Zhongfan Zhang
a1670764

Charles Jonathon Zyzniewski
a1609619

Sioli Tiafau O'Connell
a1690418

Document Revision Table

Version	**Date**	**Comment**
0.1	21/8/17	Initial draft release
0.2	23/8/17	Added detailed section headings. Added dependency/source/complexity sections to system requirements. Moved general system requirements to the nonfunctional requirements section. Added revision table. Incited a naming convention tidy up.
0.3	24/8/17	Added a source section for R0017
0.4	26/10/17	Finalised prototype section to reflect closing state for the system.
1.0	28/10/17	All sections updated for timeframe and system capability for final revision.

Contents

1 Introduction

1.1 Purpose

The Software Requirements Specification (SRS) details the current requirements of the prototype Lunar Rover Mapping Robot to be developed for Space Explorations. The specified requirements apply to the initial release of the system which will be demonstrated by Monday 30th October 2017.

The SRS describes the requirements for a prototype system which will be developed to compete in the Google Lunar X-Prize competition. The prototype system includes the robot (constructed from the provided Lego Mindstorm EV3 kit) and the related software for control and mapping in a test environment.

1.2 Intended Audience and Reading Suggestions

This document is intended for Space Explorations and the Lunar Rover prototype development team.

For Space Explorations, the intention of the document is to define the system requirements which have been identified, and come to an agreement on these requirements. Space Explorations should pay special attention to the requirements in Sections 3, 4 and 5,

For developers, the intention of this document is to provide detailed specifications that can be used to develop the software. Developers should read the whole document, with particular focus on the requirements sections and Operating Environment.

1.3 Document Conventions

User requirements are given a unique identifier in the form of Rxxxx. Where 'x' is a number from 0-9. Non-functional requirements are given a unique identifier in the form of NRxxxx. Requirements are numbered by the order in which they were added to the document. Requirements that are modified will be marked "Requirement changed on date dd/mm/yy". Requirements that are deleted will be marked "Requirement Deleted". Requirements may not appear in numerical order on the document as they are first sorted categorically, then by number.

Each requirement is assigned a priority of either High, Medium, or Low. A high priority requirement relates to the safety of the Lunar Rover when running autonomously. A medium priority requirement relates to the operation of the Lunar Rover under manual or semi-automatic control. A low priority requirement relates to user interaction with the graphical user interface representation of the map and the completion time of the exploration. These priorities will be reviewed as changes occur and may be reassigned.

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 [1].

1.4 Project Scope

The software specified in this document is for a prototype of the Lunar Rover to take part in the Google Lunar X-Prize competition. The main purposes of the software controlling the Lunar Rover are to map an area on the moon within a given boundary, and detect any radiation which could indicate remnants of Apollo 17.

The motivation for the development of this software is to have a working prototype to map the crash site and discover any remnants of the Apollo 17 with precise measurements of the tracks leading to/from them.

If the mission is successful this will enable research teams to determine how any remnants of Apollo 17 has aged over 4 decades.

1.5 Future Changes To Requirements

We understand that the requirements outlined in this document may need to be altered or removed in the future. To ensure that this process is formalised, Space Exploration have been provided with a Requirements Change Request Form (see Appendix B). This form must be used to request any change to the requirements set out in this document. Any request submitted must allow for a one week response time and an implementation time that will be agreed on by both parties at the time of the response.

2 Overall Description

2.1 Product Perspective

The product described in this SRS is for a prototype Lunar Rover which will be represented by a Lego Mindstorm EV3 kit. The full size prototype will travel 500 metres and transmit high-definition video and images back to earth.

2.2 Product Features

The features of the Lunar Rover and its software are to:

- Find the remnants of the Apollo 17 landing site.
- Survey and map the surrounding Apollo 17 area.
- Provide manual control to Lunar Rover operators through a GUI.
- Provide situational awareness to a Lunar Rover operator through a GUI.

2.3 User Classes and Characteristics

There is only a single user class - Rover Operators. Rover Operators are users who directly interact with the Lunar Rover. Rover operators have access to all functions of the Lunar Rover at any time through the GUI.

2.4 Operating Environment

The software will be run on a Lego Mindstorm EV3. The operator will be able to access the EV3 remotely via a wi-fi connection. The EV3 must always be connected to the wi-fi while running in order to transfer the scanned map data.

2.5 Design and Implementation Constraints

The map data will be stored in an XML format provided by a third-party on an undetermined date. Only the current map needs to be stored on the device at any time.

The Lunar Rover is represented by the contents of the Lego Mindstorm EV3 kit. Only sensors that are present in the kit by default can be used for the design of the Lunar Rover. This is limited to three interactive servo motors, a gyro sensor, touch sensors, colour sensors, and an ultrasonic sensor.

The software will be written using Java (version 1.7) with the use of the Eclipse Integrated Development Environment.

2.6 User Documentation

A tutorial will be provided to any operators of the Lunar Rover. The tutorial will describe the elements of the GUI and the use of any buttons or interactive elements.

2.7 Assumptions and Dependencies

We trust that the specified requirements are accurate and comprehensive, and will not be modified before the initial release, except throught the use of the Requirements Change Request Form. We assume that we will have access to the Lego Mindstorm EV3 kit to construct and test the system's operation prior to the demonstration deadline. Furthermore, we assume that the DTD will be provided to us with adequate time for implementation prior to the demonstration deadline. We also assume that during the demonstration we will have access to hardware with the appropriate software as outlined in Section 2.5.

3 System Requirements

System requirements refers to the control and movement of the Lunar Rover, mapping of the bounding area, object detection and the GUI. These are outlined in this section with each broken down into their own sub-requirements. These are subject to change on request using the Requirements Change Request Form found in appendix B.

3.1 Rover Control and Movement

R0001: Automatic Control

Description

At its core the system must be able to search within the boundary area for the radiation signal that aligns with the Apollo 17 remnants. The Lunar Rover shall be able to map the area completely without user input. This would mean that the Lunar Rover will start at a given starting point within the bounding area and then map features (outlined in R0004: Feature Cataloging) independently until the goal is reached or the entire bounding area has been mapped.

Rationale

In order to save the most time for the client and as the layout of the crash site is unknown an automated searching route is needed.

Acceptance criteria

Automatic control must allow the Lunar Rover to begin navigating the area until the target is found or all accessible areas have been mapped. On completion the Lunar Rover should return to the start location.

Source

Client Group Project Specification.

Status

Scheduled for implementation in the final prototype.

Dependencies

R0004: Feature Cataloging
R0007: No Go Zone Detection
R0008: Object Collision Detection
R0009: Operation Mode Selection
R0012: Examine Mapped Data

Complexity

This requirement is high in complexity and also in implementation time due to the number of other dependencies and their respective complexities.

Priority

High.

R0002: Semi-Automatic Control

Description

The Lunar Rover will be able to take in a new coordinate within the bounding area and will traverse a path from it's current location to the new coordinate whilst moving around objects that may block the path. If the coordinate is not within the bounding area the Lunar Rover will halt once the boundary of the bounding area has been found and await further instruction.

Rationale

The client may have an idea of where they would like the Lunar Rover to traverse, however they may still not have enough information to warrant manual control.

Acceptance criteria

The Lunar Rover will either have made it to the given coordinate or shall be waiting at the boundary that is in between it's original position and the goal coordinate position.

Source

Client Meeting One.

Status

Scheduled for implementation in the second prototype.

Dependencies

R0004: Feature Cataloging
R0007: No Go Zone Detection
R0008: Object Collision Detection
R0009: Operation Mode Selection
R0012: Examine Mapped Data

Complexity

This requirement is high in complexity and also in implementation time due to the number of other dependencies and their respective complexities.

Priority

Medium.

R0003: Manual Control

Description

Using a provided GUI, the client shall be able to move the Lunar Rover independently of any previously inputted route traversal instructions. It will be able to move using the GUI which will display three (4) buttons each representing one of the movement functions (engines forward, engines backward, rotate right and rotate left).

Rationale

The client may want to inspect a particular area in a more detailed fashion to solve a particular curiosity. This requirement is also in place because the Lunar Rover may need to be aligned to an already known location to start semi-automatic control.

Acceptance criteria

Interaction via a GUI moves the Lunar Rover forward and backwards and allows for rotation to align with a new heading.

Source

Client Group Project Specification.

Status

Scheduled for implementation in the first prototype.

Dependencies

R0007: No Go Zone Detection
R0008: Object Collision Detection
R0009: Operation Mode Selection
R0010: Manual Operation Controls

Complexity

This requirement is low in complexity but at a medium rate of implementation time as the GUI requirements depend on multiple other dependencies.

Priority

Medium.

3.2 Mapping

R0004: Feature Cataloging

Description

The Lunar Rover shall store information about the environment that it is traversing in the form of a graphical map generated from XML formatted data. Depending on what is encountered between the start and when the Lunar Rover reaches the goal/finishes operation, the features included could include at most, different types of radiation, the boundary marking the edge of the boundary area, miscellaneous obstructions, the Apollo 17 crash site remnants.

Rationale

Cataloging map features that the Lunar Rover is encountering will allow the Lunar Rover to find the goal and avoid collision with objects as well as letting it know where it has already traversed.

Acceptance criteria

Whilst traversing the Lunar Rover will store information internally about it's surroundings in the expected XML format. At the end of the traversal the Lunar Rover shall update the GUI with the features found.

Source

Client Group Project Specification.

Status

Scheduled for implementation in the second prototype.

Dependencies

R0001: Automatic Control
R0002: Semi-Automatic Control
R0003: Manual Control
R0007: No Go Zone Detection
R0008: Object Collision Detection

Complexity

This requirement is at a medium rate in complexity but is high in implementation time due to the number of other dependencies and their respective complexities.

Priority

High.

R0005: Import Map

Description

The Lunar Rover shall be able to take in a previously generated map file in the XML file format provided by the client. This should then update any data stored in the GUI and allow the Lunar Rover to know in advance where it will encounter any radiation, boundaries or miscellaneous objects from the map file.

Rationale

To make sure that the Lunar Rover has as much information as possible before a traversal.

Acceptance criteria

Before traversal begins, a map in the XML file format provided will be able to be uploaded to the Lunar Rover via the GUI and be displayed in the map area of the GUI.

Source

Client Group Project Specification.

Status

Scheduled for implementation in the final prototype.

Dependencies

R0004: Feature Cataloging
R0012: Examine Mapped Data
R0013: Map Data I/O

Complexity

This requirement is low in complexity but at a medium rate of implementation time as the GUI requirements depend on multiple other dependencies.

Priority

Low.

R0006: Exportable Map Data

Description

Once the Lunar Rover has finished a traversal and has created the map data by feature cataloging (R0004) the Lunar Rover shall be able to export this data.

Rationale

The exported map data may be beneficial to the client for analysis. It also provides data that the Lunar Rover can then use as input before another traversal.

Acceptance criteria

The Lunar Rover will save the map data showing the cataloged features in the XML format.

Source

Client Group Project Specification.

Status

Scheduled for implementation in the final prototype.

Dependencies

R0001: Automatic Control
R0002: Semi-Automatic Control
R0004: Feature Cataloging
R0007: No Go Zone Detection
R0008: Object Collision Detection
R0013: Map Data I/O

Complexity

This requirement is high in complexity and also in implementation time due to the number of dependencies and their respective complexities.

Priority

Low.

3.3 Object Detection

R0007: No Go Zone Detection

Description

Whilst mapping, the Lunar Rover shall be able to be halted and the client will be able to set a No Go Zone onto the map (R0008). The system shall allow the client to create a No Go Zone using the GUI by inputting a coordinate representing the center of the No Go Zone as well as a radius to dictate the size of the No Go Zone.

Rationale

There may be treacherous areas of the map which the client may deem that the Lunar Rover would see no benefit moving through. In this case a No Go Zone will keep the Lunar Rover safe from these areas.

Acceptance criteria

By interfacing with the GUI using an ”Create No Go Zone” feature the user will be able to input a coordinate representing the center of the No Go Zone as well as a radius to dictate the size of the No Go Zone. This will then be shown on the GUI and in the mapping data. If the Lunar Rover encounters the border of a No Go Zone it will simply avoid it on this and any further mapping traversals where the Lunar Rover has this same data.

Source

Client Group Project Specification.

Status

Scheduled for implementation in the final prototype.

Dependencies

R0001: Automatic Control
R0002: Semi-Automatic Control
R0003: Manual Control
R0004: Feature Cataloging
R0014: No Go Zone Creation Interface

Complexity

This requirement aligns for a medium rate of complexity and also in implementation time as it is a requirement that is built off of other components which would be assumed to be implemented before achieving this requirement.

Priority

High.

R0008: Object Collision Detection

Description

The Lunar Rover may encounter multiple unknown objects whilst traversing the boundary area and shall be able to avoid them.

Rationale

Being able to move around unknown objects serves as a method of damage prevention for the Lunar Rover.

Acceptance criteria

When the Lunar Rover encounters these objects in automatic control it shall be able to change to a different heading and continue on a new path that stops the Lunar Rover from colliding with the object. When the Lunar Rover encounters these objects in semi-automatic control it shall halt to stop the Lunar Rover from colliding with the object. When the Lunar Rover encounters these objects in manual control it shall be able to warn the client before colliding with the object.

Source

Client Meeting Two.

Status

Scheduled for implementation in the second prototype.

Dependencies

R0001: Automatic Control
R0002: Semi-Automatic Control
R0003: Manual Control

Complexity

This requirement is high in complexity and implementation time as it spans across all control implementations and takes on different forms depending on the control method that is in use.

Priority

High.

3.4 GUI

R0009: Operation Mode Selection

Description

The Lunar Rover shall be able to be swapped between a set of three (3) possible operation modes via the GUI.

Rationale

Depending on the situation and what particular information the client may have; The client may choose to have the Lunar Rover swap between operation of different modes and would need to have a way to interact with the Lunar Rover to change between these modes.

Acceptance criteria

The GUI will provide a way for the Lunar Rover to be set into one of three possible operation modes (Automatic, Semi-Automatic and Manual) at the click of a button.

Source

Client Meeting One.

Status

Scheduled for implementation in the second prototype.

Dependencies

N/A

Complexity

This requirement is low in complexity and implementation time.

Priority

Medium.

R0010: Manual Operation Controls

Description

The GUI shall provide an interface that is not physically on the Lunar Rover so that it can be remotely controlled.

Rationale

So that the client is able to direct the Lunar Rover in the way that they choose they must have an interface that will remotely move it.

Acceptance criteria

The GUI must have controls that allow for manually controlling the Lunar Rover in the form of three (4) buttons to either power engines forward, engines backward, rotate left or rotate right. When the Lunar Rover is set to manual control these buttons will then move the Lunar Rover remotely.

Source

Client Group Project Specification.

Status

Scheduled for implementation in the first prototype.

Dependencies

N/A

Complexity

This requirement is low in complexity and implementation time.

Priority

Low.

R0011: Halt Lunar Rover Operation

Description

The client will be able to stop the Lunar Rover immediately from it's current traversal.

Rationale

The client may have just received information that dictates using a different control method for the Lunar Rover or to possibly provide the Lunar Rover with more information before continuing.

Acceptance criteria

The GUI will provide an interface to halt the Lunar Rover from its current operation in the form of a button. When this button is clicked the Lunar Rover will halt.

Source

Client Meeting Two.

Status

Scheduled for implementation in the first prototype.

Dependencies

N/A

Complexity

This requirement is at a medium rate of complexity and implementation time as the command of the requirement spans across multiple control methods.

Priority

High.

R0012: Examine Mapped Data

Description

The GUI will show any map data that the Lunar Rover has currently cataloged.

Rationale

To provide the latest information to the client in a graphical form at times where reading through the XML map data may not be the best solution.

Acceptance criteria

The GUI shows all the features of the mapped data as a graphical display.

Source

Client Group Project Specification.

Status

Scheduled for implementation in the second prototype.

Dependencies

R0004: Feature Cataloging
R0005: Import Map
R0007: No Go Zone Detection
R0008: Object Collision Detection
R0012: Examine Mapped Data
R0013: Map Data I/O
R0014: No Go Zone Creation Interface

Complexity

This requirement is high in complexity and also in implementation time due to the number of dependencies and their respective complexities.

Priority

Low.

R0013: Map Data I/O

Description

The GUI will allow for an interface to import / export any map data that has been collated by the Lunar Rover.

Rationale

The client may have previous map data to provide the Lunar Rover with more information to result in a more efficient operation or otherwise want to store current map data for future use.

Acceptance criteria

The client is able to import / export map files by interacting with the buttons of the GUI. If importing map data the graphical display of the map will then reflect the information that data represents.

Source

Client Group Project Specification.

Status

Scheduled for implementation in the second prototype.

Dependencies

R0004: Feature Cataloging
R0005: Import Map
R0006: Exportable Map Data

Complexity

This requirement is at a medium rate of complexity and implementation time.

Priority

Low.

R0014: No Go Zone Creation Interface

Description

The GUI shall provide a system for entering data that represents the boundary of a No Go Zone.

Rationale

If / when the client discovers No Go Zones they need an interface that allows them to enter the boundary of this No Go Zone so that the Lunar Rover can then act accordingly.

Acceptance criteria

The GUI will provide an interface for the client to be able to enter the coordinate and radius as needed to create a No Go Zone. Once the client has entered the coordinate and radius and accepted them, the GUI will then update the graphical display to represent the No Go Zone.

Source

Client Group Project Specification.

Status

Scheduled for implementation in the final prototype.

Dependencies

R0004: Feature Cataloging
R0005: Import Map
R0012: Examine Mapped Data

Complexity

This requirement is at a medium rate of complexity and implementation time.

Priority

Medium.

R0015: Click For Coordinate

Description

The GUI shall provide the coordinates of a particular area of the map when the client clicks on the graphical display and this function is active.

Rationale

The client may like to know at what location a feature on the graphically displayed map is located at.

Acceptance criteria

If the client clicks on an area on the graphical display the GUI will display the coordinate relating to the location that they clicked on if the click for coordinate toggle switch is enabled. There will subsequently be another GUI button to enable / disable the click for coordinate functionality.

Source

Client Meeting Three.

Status

Scheduled for implementation in the final prototype.

Dependencies

N/A.

Complexity

This requirement is at a low rate of complexity and implementation time.

Priority

Low.

R0016: Status Bar

Description

The GUI shall provide a message showing the current state of the system.

Rationale

In order to provide feedback to the client whilst the system is in process, having the current system state allows for peace of mind for the operation and to allow the client to know the stage of exploration / traversal of the Lunar Rover.

Acceptance criteria

During all control modes there will be messages displayed as the system moves to different processes on a status bar that will be a part of the GUI providing feedback.

Source

Client Meeting Three.

Status

Scheduled for implementation in the second prototype.

Dependencies

N/A.

Complexity

This requirement is at a low rate of complexity and implementation time.

Priority

High.

R0017: Warnings for Manual Control

Description

The GUI shall display warning messages when an object is detected whilst under manual control. In describing this note that the system will not influence the Lunar Rover in any way.

Rationale

In order to protect the safety of the Lunar Rover when under manual control the system needs to have a way to warn the user of any impending objects that it may collide with.

Acceptance criteria

Whilst the Lunar Rover is under traversal if there is an unknown object detected in the path of the Lunar Rover the status bar of the GUI will display a warning.

Source

Client Meeting Three.

Status

Scheduled for implementation in the second prototype.

Dependencies

R0016: Status Bar
R0010: Manual Operation Controls
R0003: Manual Control
R0008: Object Collision Detection

Complexity

This requirement is at a medium rate of complexity and implementation time due to the other dependencies and their respective complexities.

Priority

High.

4 External Interface Requirements

4.1 User Interfaces

Users will interface with the Lunar Rover via a graphical user interface on a laptop. The user interface will have a visual representation of the current map and interactive buttons for manual controls of the Lunar Rover and the map.

4.2 Software Interfaces

The software operator relevant software will run on a laptop running either Mac OSX, Windows, or Linux running Java 1.7. The Lunar Rover software will run on a Lego Mindstorm EV3 using Java 1.7.

5 Nonfunctional Requirements

Any nonfunctional requirements are defined in this section. These requirements are not essential to the operation of the Lunar Rover however they are still important to provide a satisfactory solution to the problem outlined. As with the functional requirements these are subject to change upon request via the Requirement Change Request Form found in appendix B.

5.1 Performance Requirements

NR0001: General command performance and availability

Description

When the Lunar Rover receives any non-safety critical command from the operation and management software it SHOULD process the command within one second. Non-safety critical command is defined as any command that does not have its own requirement under Section 5.2. This requirement SHOULD NOT include the time delay from long distance communication or from the time required to complete the execution of the command.

Rationale

To allow realtime interaction the Lunar Rover must respond to commands quickly. Failing to execute a non-safety critical command within the time requirements does not represent a safety risk but does impact realtime usability, therefore the requirement has been relaxed from MUST to SHOULD.

Priority

Medium

NR0002: Mission completion performance

Description

In automatic mode the Lunar Rover should either find the goal or as a worst case have mapped the entire boundary area within 20 minutes as outlined by the client. This time frame does not apply to semi-automatic / manual control as that time frame is then dictated by the user of the system.

Rationale

As to make the operation worthwhile the Lunar Rover must complete the task within a reasonable time period.

Status

Scheduled for implementation in the final prototype.

Priority

Low

5.2 Safety Requirements

NR0003: Stop command performance and availability

Description

When the Lunar Rover receives a stop command from the operation and management software, the Lunar Rover MUST NOT take longer than one second to process the command and come to a full stop. This requirement SHOULD NOT include the time delay from long distance communication.

Rationale

Given the Lunar Rover operates in a non-recoverable environment and that safety is a high priority, it is paramount that it MUST be able to stop at any moment to avoid endangering itself and its mission.

Priority

High

NR0007: Manual mode command performance and availability

Description

When the Lunar Rover receives a switch to manual mode command from the operation and management software, the Lunar Rover MUST NOT take longer than one second to process the command and come to a full stop. This requirement SHOULD NOT include the time delay from long distance communication.

Rationale

Given the Lunar Rover operates in a non-recoverable environment and that safety is a high priority, it is paramount that an operator MUST be able to enter manual mode and override any automatic commands to avoid endangering the Lunar Rover and its mission.

Priority

High

NR0004: Avoid high force collisions

Description

The Lunar Rover MUST NOT collide with an object with enough force that the object moves or causes its wheels to spin without the Lunar Rover moving, instead it SHOULD stop.

Rationale

High force collisions can cause undesirable damage to the Lunar Rover or surrounding objects.

Priority

High

NR0005: Avoid entering craters

Description

The Lunar Rover MUST NOT allow two or more wheels to enter the crater (this requirement is subject to change based on the Lunar Rover's configuration)

Rationale

Driving into a crater is potentially a non-recoverable situation ending the mission prematurely.

Priority

High

5.3 Software Quality Attributes

NR0006: GUI user friendly

Description

The GUI SHOULD be user friendly and with any operator being able to use it within 5 minutes.

Rationale

Ease of use of the GUI will minimize user error during operation and also have optimal control of the rover at all times.

Priority

Low

6 Prototype Outline

It is planned that three prototypes will be produced including the final prototype to be used for the final presentation. The requirements and the timeframe for each are outlined here.

6.1 First Prototype

Description

This is the first representation of the Lunar Rover and will be presented to the client with the basic operations of the Lunar Rover and the GUI including manual control for movement and the halt command.

Requirements

R0003 Manual Control
R0008: Object Collision Detection
R0010 Manual Operation Controls
R0011 Halt Lunar Rover Operation

Timeframe

Presented to the client in the 4th client meeting.

6.2 Second Prototype

Description

This will be the second representation of the system and will be presented to the client with more features that align with the outlined requirements in order to start testing familiarity with the system and the Lunar Rover.

Requirements

Current proposed requirements to be achieved are: R0002: Semi-Automatic Control
R0004: Feature Cataloging
R0009: Operation Mode Selection
R0012: Examine Mapped Data
R0013: Map Data I/O
R0016: Status Bar
R0017: Warnings for Manual Control
However this is to be refined and confirmed after feedback from the first prototype.

Requirement changed on date 26/10/17
After feedback from the client the team was instructed that automatic movement would be more preferable for the second prototype than semi-automatic control. The goal system for the second prototype was tweaked and resulted in the following requirements being met.

Closing Second Prototype Requirements

R0001: Automatic Control
R0004: Feature Cataloging
R0005: Import Map
R0009: Operation Mode Selection
R0012: Examine Mapped Data

R0013: Map Data I/O

Timeframe

Presented to the client in the 9th client meeting.

6.3 Final Prototype

Description

This will be the final representation of the system and will be presented to the client as their complete solution.

Requirements

Current proposed requirements to be achieved are:
R0001: Automatic Control
R0005: Import Map
R0006: Exportable Map Data
R0007: No Go Zone Detection
R0014: No Go Zone Creation Interface
R0015: Click For Coordinate
However this is to be refined and confirmed after feedback from the first and second prototype.

Requirement changed on date 26/10/17
After further feedback from the client as well as to adhere to the restraints on timeframe the closing final prototype requirements that were achieved are as follows.

Closing Final Prototype Requirements

R0002: Semi-Automatic Control
R0006: Exportable Map Data
R0007: No Go Zone Detection
R0014: No Go Zone Creation Interface
R0015: Click For Coordinate
R0016: Status Bar
R0017: Warnings for Manual Control

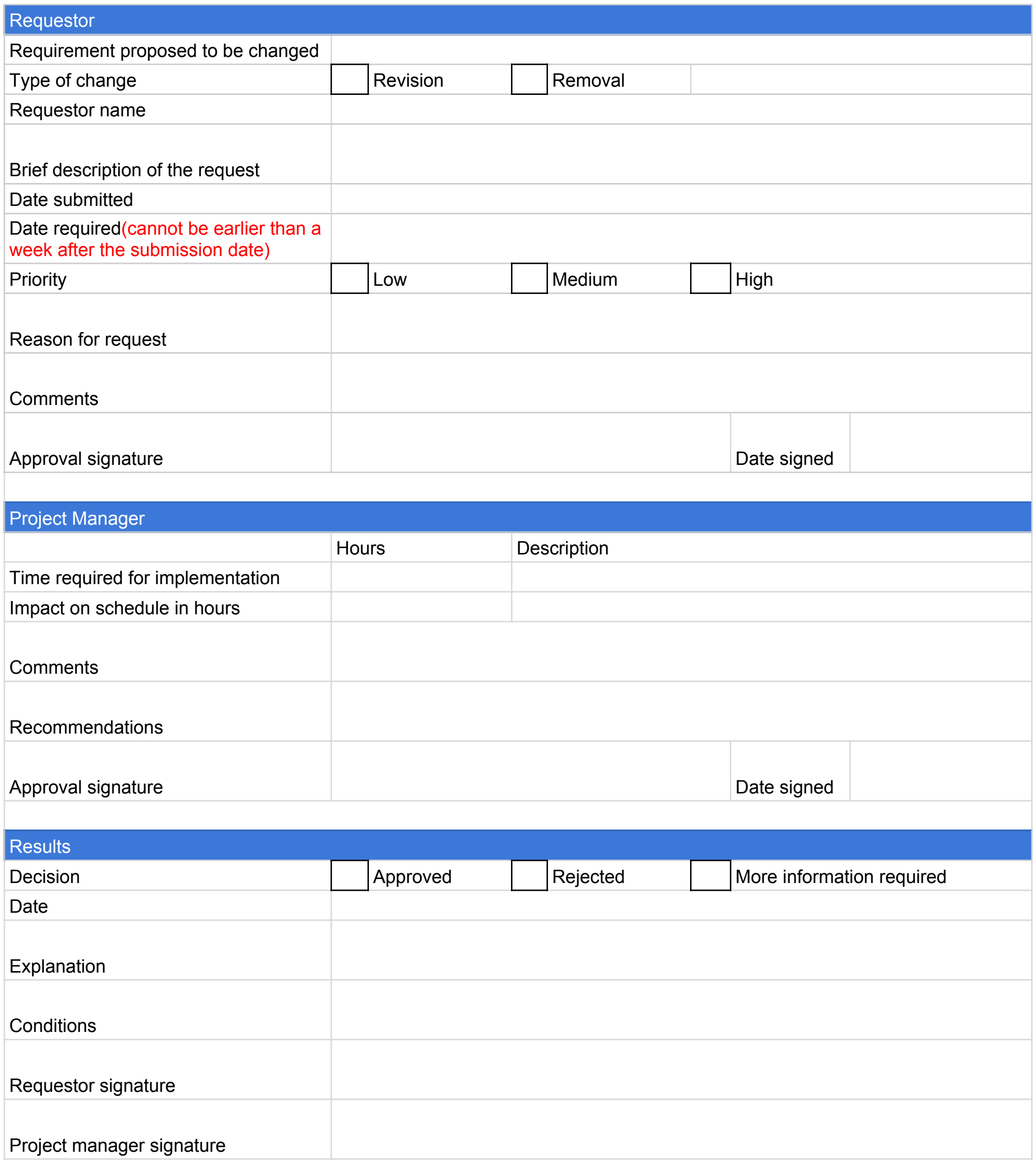
Timeframe

Presented to the client in the final presentation.

A Glossary

DTD	Document Type Definition
GUI	Graphical User Interface
I/O	Input/Output
SRS	Software Requirements Specification
XML	Extensible Markup Language

B Requirements Change Request Form



Requestor			
Requirement proposed to be changed			
Type of change	□ Revision	□ Removal	
Requestor name			
Brief description of the request			
Date submitted			
Date required(cannot be earlier than a week after the submission date)			
Priority	□ Low	□ Medium	□ High
Reason for request			
Comments			
Approval signature		Date signed	
Project Manager			
	Hours	Description	
Time required for implementation			
Impact on schedule in hours			
Comments			
Recommendations			
Approval signature		Date signed	
Results			
Decision	□ Approved	□ Rejected	□ More information required
Date			
Explanation			
Conditions			
Requestor signature			
Project manager signature			

C References

1. Bradner, S., ”Key Words for Use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997.