



**POLITECNICO**  
MILANO 1863

RoadFix project by Haakon Flaten - Paal  
Kjekstad

# **Requirement Analysis and Specification Document**

---

**Deliverable:** RASD

**Title:** Requirement Analysis and Verification Document

**Authors:** Haakon Flaten and Paal Kjekstad

**Version:** 1.0

**Date:** 20-April-2021

**Download page:** [LINK TO YOUR REPOSITORY](#)

**Copyright:** Copyright © 2021, H.Flaten and P.Kjekstad – All rights reserved

---

## Contents

<b>Table of Contents</b>	3
<b>List of Figures</b>	5
<b>List of Tables</b>	5
<b>1 Introduction</b>	6
1.1 Purpose	6
1.2 Scope	6
1.3 Definitions, acronyms and abbreviations	6
1.3.1 Definitions	6
1.3.2 Acronyms	6
1.3.3 Abbreviations	6
1.4 Revision history	6
1.5 Reference Document	6
<b>2 Overall Description</b>	7
2.1 Product description	7
2.2 Functions	8
2.3 User Characteristics	8
2.4 Goals	8
2.5 Domain Assumptions	8
<b>3 Requirements</b>	9
3.1 Functional Requirements	9
3.2 Nonfunctional Requirements	9
3.3 Interacting software/hardware	9
3.3.1 Hardware Interface	9
3.3.2 Software Interface	9
<b>4 Relevant Phenomena</b>	10
4.1 The World	10
4.2 The Machine	10
4.3 Shared Phenomena	10
<b>5 Use cases</b>	11
5.1 Scenario	11
5.1.1 Use case	11
5.1.2 Actors	11
5.1.3 Entry condition	11
5.1.4 Flow of events	11
5.1.5 Exit condition	11
5.1.6 Exceptions	11
5.1.7 Special requirements	11
5.2 Scenario	12
5.2.1 Use case name: RoadRepair	12
5.2.2 Actors	12
5.2.3 Entry condition	12
5.2.4 Flow of events	12
5.2.5 Exit condition	12

5.2.6 Exceptions . . . . .	12
5.2.7 Special requirements . . . . .	12
5.2.8 Special requirements . . . . .	12
<b>6 Appendix . . . . .</b>	<b>13</b>
<b>7 Software used . . . . .</b>	<b>13</b>
<b>8 Effort . . . . .</b>	<b>13</b>
<b>References . . . . .</b>	<b>14</b>

## List of Figures

1	concept drawing of the map . . . . .	7
---	--------------------------------------	---

## List of Tables

## 1 Introduction

### 1.1 Purpose

The aim for the Requirement Analysis and Specification Document (RASD) is to make it easier to understand the interactions between the user and software.

Description of the requirements, both functional and non-functional, all scenarios and actions between the program and user will be discussed.

The persons who will go through the document is the software developers, asphalt contractors and road owners.

### 1.2 Scope

RoadFix will be a web application that makes the roads better by letting the road owners know where to allocate the resources in the most efficient way. In Epicollect5, data about potholes is stored and that information will be available for the road owners.

The idea is that damaged roads used by the most people, will get more reported damages than less used roads. Furthermore, this will increase the decision base for the road owners and help them allocate the resources where it is most needed. This in turn, will be socio-economically favourable, and that is the main goal of the program.

### 1.3 Definitions, acronyms and abbreviations

Clarifications of essential definitions, acronyms and abbreviations

#### 1.3.1 Definitions

- Asphalt constructor: The contractor who fixes the road
- Road Owners: The responsible part of the road, for example private shareholders, government, etc.
- Program: The RoadFix program is referred to as program
- Pothole: Damage on the road related to vehicles
- App: Application

#### 1.3.2 Acronyms

- RASD: Requirement Analysis and Specification Document
- GPS: Global Positioning System

#### 1.3.3 Abbreviations

No abbreviations were used other than acronyms.

## 1.4 Revision history

14.04.2021 - First writings in Latex

## 1.5 Reference Document

The assignment document: AY20-21 Project.pdf

## 2 Overall Description

### 2.1 Product description

The program, in development, is aimed to help the road owners to get a more efficient flow of information and then a bigger bases of information to make decisions.

The data used in this application will be taken from Epicollect5. The data set in focus is a 190 element document with three attributes including geographical coordinates. The Epicollect5 project has its origin in the streets of London. All the 190 elements will be processed and added to the map. By visualising the damages on the map like small circles, the circles are drawn around the coordinates given in the data. Every registered damage that are in close proximity to each other will be marked as a group. When a group of damages are joined together and that group surpass a specific quantity, they will change colors on the map. The parts of the roads that are worst will get the color red, moving down to the color yellow for the least bad parts of the roads. This visualises the condition of the roads in a good and easy way. The picture below 1 is a concept drawing of how the design of the map screen in the program could look like.

This web application can also be programmed to take in data from a user of the road. More of a reporting application, where the user of the app does the work and the road owners can take out the users experience of the road, and use that to improve the right roads. In case of making this application it is needed a way for the user to add damages to the data base in a easy way. This feature can be added in later in the project, if time allows. It is not the first prioritization of the group and it will not be apart of any other sections written about in this document.

The software in this project will be created as a web application. A development of a mobile application could be sensible in the long run, if the feature of adding information on the road rapidly as the user reports potholes and damages. Hence mobile applications are more used than web applications on mobile devices, as Yoram Wurmser touches in his article [1].



Figure 1: concept drawing of the map

The program will have user profiles to be used by the road owners and asphalt contractors. Furthermore, this makes it easier to see who have made the changes done in the program. The profile will be able to change the status of all the registered potholes. So after a part of the road have been fixed the damages can be removed in the program.

## 2.2 Functions

The functions of the program will be as few and easy as possible to obtain a simple and user friendly interface. The functions the program needs are:

- Take in the information from the Epicollect5 data
- Possibility for the road owner to rate the damages
- Calculate and visualise the damages in regard to the rating system and GPS-coordinates
- Updated map with the potholes registered and rated by road owners
- Be able to take out lists of the registered damages
- Have the possibility to change the status of the damages

## 2.3 User Characteristics

The users have to possess a basic level of knowledge regarding computer use. It requires that the user can navigate to a web site and be able to download the files needed. They also need to have knowledge of how to rate the damages on the road.

For every picture the road owner rate, the program should calculate how much damage the road has considering other registered road damages in the same area.

## 2.4 Goals

It is desirable that RoadFix meets certain user needs, or goals.

- For every picture the road owner rate, the program should calculate how much damage the road has considering other registered road damages in the same area.
- It must be possible to take out lists of the visualised groups and the worst damages.
- After the road has been fixed the asphalt contractor must be able to change the status of the road to fixed.

## 2.5 Domain Assumptions

In order to let the program work, some assumptions are made:

- Every picture has a GPS-coordinate
- The GPS-coordinate have an error that is small
- The user have internet connection when using the application
- All the information from every picture needs to be interpreted the right way by the user

The user of the program is the road owners and the asphalt contractors. They will be able to work with information on the app and use it to keep day-to-day track of the work that's been done. That makes it easier for both parts to communicate during the progress.

## 3 Requirements

is needed to describe the interactions between the system and its environment independent from implementation.

### 3.1 Functional Requirements

For the program to work, as planned, it is needed to describe the interactions between the system and its environment independent from implementation. The user:

- Should be able to rate every picture
- Have to be able to access every picture after it has been evaluated
- Be able to update damages consecutively
- Be able to remove registered damages

### 3.2 Nonfunctional Requirements

Which are requirements you put on the program and its infrastructure. This is important in order to guarantee a seamless user experience, therefore some criteria needs to be met:

- No storage limit in the program
- The time from a update from the user to the updating of the map needs to be within 30 seconds
- The system should be available 24/7

## 3.3 Interacting software/hardware

### 3.3.1 Hardware Interface

Operating the program is easiest through a personal computer with a web browser that is connected to the Internet.

### 3.3.2 Software Interface

Web browsers that should be supported are Google Chrome, Safari, Internet Explorer, Opera, Firefox.

## 4 Relevant Phenomena

### 4.1 The World

Requirement engineering is concerned with phenomena occurring in the World. The World is defined as the portion of the real world that the Machine can influence. For the RoadFix program, this is some World phenomena:

- Occurrence of damaged roads
- Taking picture of the road
- Reduction of asphalt supply
- Asphalt contractors repairing the road

### 4.2 The Machine

On the other hand, some phenomena occurs inside the Machine. This is the portion of the system to be developed, and this are some examples:

- Storage of new information
- Computation of roads that should be fixed

### 4.3 Shared Phenomena

Some phenomena are shared between the Machine and the World, and this is how they are connected. Here are some examples of shared phenomena:

- A phenomena that are controlled by the Machine and observed by the World is the allocation of asphalt contractor to a damaged road. The Machine gives information when a damaged road needs reparation, and the world observes as it happens.
- The interpretation of a picture and how severe the damage are will be controlled by the World and observed by the Machine.
- Updating road condition status is a shared phenomena. It is controlled by the World and observed by the Machine

## 5 Use cases

### 5.1 Scenario

"Tiffany, an employee in the road owner company wants to check how the conditions on the road are. She logs into RoadFix and there she finds pictures of different roads. She then starts to rate the damage to the roads based on the picture from Epicollect5. After rating a couple of pictures the program indicates that a zone has turned yellow. She then contacts Axl, an employee at the asphalt contractor company, and tells him that one area has turned yellow. He confirms that he has received the message."

#### 5.1.1 Use case

The RoadRating use case

#### 5.1.2 Actors

- Road owner: Responsible for the road, Tiffany in the scenario
- Project engineer: Resource allocator in the asphalt construction company, Axl in the scenario

#### 5.1.3 Entry condition

True, a picture is always available for review.

#### 5.1.4 Flow of events

- The Owner starts working and logs into RoadFix
- The owner opens the first picture
- RoadFix answer with giving the Owner a form
- The Owner rates the picture from 1-10, and submit
- RoadFix then calculate if the total rating in a specific area is exceeding a certain level
- After a while one area is exceeding the level for the yellow limit
- The program marks all the pictures that is in the area with a yellow color
- The owner follows up with contacting the project engineer and tells that one area has turned yellow

#### 5.1.5 Exit condition

The project engineer has received and confirmed the message.

#### 5.1.6 Exceptions

- If there are no more pictures from Epicollect5 and all the damages has been fixed, then it will not be more work to do for the road owner or the asphalt contractor.
- If the color turns red before yellow, the same procedure are followed by the road owner

#### 5.1.7 Special requirements

- The storage limit is sufficient
- The calculation time of the program is less than 30 seconds

## 5.2 Scenario

Axl, gets notified from Tiffany that a zone just turned yellow. He then review the picture from Epicollect5 and rating from Tiffany, and acknowledge the report. He looks at the weekly schedule plan and notice that a team of constructors are finished in 2 hours. He notify Karol, the boss of the constructors, and inform him about the new task they have in 2 hours. When the road is eventually fixed, Axl changes the status of the damages in RoadFix to "repaired damage".

### 5.2.1 Use case name: RoadRepair

#### 5.2.2 Actors

- Road owner: Responsible for the road, Tiffany in the scenario
- Project engineer: Resource allocator in the asphalt construction company, Axl in the scenario
- Chief constructor: The leader for the asphalt constructor team, Karol in the scenario

#### 5.2.3 Entry condition

True, a asphalt constructor team is always available within a reasonable time.

#### 5.2.4 Flow of events

- The project engineer is notified from the owner that a zone has turned yellow
- The project engineer reviews the report at RoadFix and start to allocate resources
- The first constructor chief is available in 2 hours
- The project engineer give the location to the constructor chief and tells them to be there in 2 hours
- When finished, the project engineer change the report in RoadFix

#### 5.2.5 Exit condition

The project engineer has changed the report in RoadFix.

#### 5.2.6 Exceptions

#### 5.2.7 Special requirements

- A work team is not available in reasonable time, the project engineer calls the road owner
- If the report from the road owner says that it is red, the project engineer needs to allocate resources in less than 30 minutes

#### 5.2.8 Special requirements

No special requirements

## 6 Appendix

## 7 Software used

- Overleaf as an editor for Latex

## 8 Effort

Both have used the same amount of work and have helped each other with all sections. There is no point to specify hours spent on each activity since our work overlaps.

- Paal Kjekstad: 20 hours
- Haakon Flaten: 20 hours

## References

- [1] Yoram Wurmser. The Majority of Americans' Mobile Time Spent Takes Place in Apps. *E-Marketer*, page 2021, 2020.