RASD + Tiberio - Saeid Rezaei

RASD - Requirement Analysis and Specification Document

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Tiberio - Saeid Rezaei
Version:	1.0
Date:	18-October-2019
Download page:	https://github.com/TiberioG/GalbiatiRezaei.git
Copyright:	Copyright © 2019, Tiberio - Saeid Rezaei – All rights reserved

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

general description ...

1.2 Scope

1.2.1 Description of the given problem

1.2.2 Goals

[G1] Allow users to notify authorities about traffic violations

[G2] Allow users to send pictures of violations

[G3] Be sure every information uploaded is never altered

[G4] Automatically add metadata to the reported pictures

[G5] allow users to mine information recorded

[G6] have at least two different priviledge for mining data

[G7] generate traffic tickets

[G8] authorities can see the the licence plates of violators, regular users cannot

1.3 Definitions, acronyms, abbreviations

1.3.1 Definitions

1.3.2 acronyms

1.3.3 abbreviations

1.4 Revision history

1.5 Reference Documents

1.6 Document Structure

2 Overall Description

2.1 Product perspective

add here class diagram + verbal description

2.2 Product functions

2.2.1 login

2.2.2 sending pics

2.2.3 mining info

2.2.4 issue a ticket

2.2.5 generate statistics

2.3 User characteristics

2.4 Assumptions, dependencies and constraints

D the device should acquire position with an accuracy of enouth meters in order to univocally determine the road (e.g. 5 meters)

D the device should take pictures with enouth resolution to be able to read the licence plate using the external software

D in every picture the licence plate should be visible and the kind of violation

D the number and kind of violation should be finite (defined by the law)

D every authority account is verified and it's not possible to be created using the frontend

The app will be dependent on a third-party service to read the licence plate of the cars. (For example http://www.openalpr.com)

The app will be dependent on a smartphone, which has to provide the following features:

1. Internet connection, possibily using 2G/3G/4G in order to be available where there is no WiFi, considering the use case "on the road"
2. GPS sensor

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2 Functional Requirements

think about use cases @both

3.2.1 User

3.2.2 Third party

3.2.3 Requirements

Requirements in order to satisfy the goals

[R1] test

3.3 Performance Requirements

3.4 Design Constraints

3.4.1 Standards compliance

The app shoudld be available for the two main operating systems of smartphones: Android Os and Apple iOS.
The traffic violations which can be reported should be compliant to the local traffic code where the app will be used.
For an use in Italy the app should be compliant to the "Codice della Strada", in particular parking violations are reported in Art. 157.

3.4.2 Hardware limitations

The app will have a server side and a client side (smartphone). On server side limitations can be the size of available storage and the bandwidth. On smartphone side we have the network connectivity and GPS limitations in some areas.

3.4.3 Any other constraint

Application should be compliant to European GDPR and don't track users.

3.5 Software System Attributes

3.5.1 Reliability

3.5.2 Availability

3.5.3 Security

3.5.4 Maintainability

3.5.5 Portability

4 Formal Analysis Using Alloy

5 Effort Spent

References