

Smart Gate System

Odelia Hochman | odeliamos0@gmail.com

Efrat Cohen | cohenefratoni747@gmail.com

Instructor : Dr. Amos Azaria

Software Requirements Specification Document

Version: 1.0

Date: 13/12/2020

Table of Contents

1. Introduction

<i>1.1 Purpose</i>	4
<i>1.2 Scope</i>	4
<i>1.3 Definitions, Acronyms, and Abbreviations</i>	4
<i>1.4 References</i>	5
<i>1.5 Overview</i>	5

2. The Overall Description

<i>2.1 Product Perspective</i>	5-6
2.1.1 System Interfaces	6
2.1.2 Interfaces	6
2.1.3 Hardware Interfaces	7
2.1.4 Software Interfaces	7
2.1.5 Communications Interfaces	7
2.1.6 Memory Constraints	7
2.1.7 Operations	7
2.1.8 Site Adaptation Requirements	8
<i>2.2 Product Functions</i>	8
<i>2.3 User Characteristics</i>	8
<i>2.4 Constraints</i>	8
<i>2.5 Assumptions and Dependencies</i>	9
<i>2.6 Apportioning of Requirements</i>	9

3. Specific Requirements 10

<i>3.1 External interfaces</i>	10
<i>3.2 Functions</i>	10
<i>3.3 Performance Requirements</i>	10
<i>3.4 Logical Database Requirements</i>	11
<i>3.5 Design Constraints</i>	11
3.5.1 Standards Compliance	11
<i>3.6 Software System Attributes</i>	12

3.6.1 Reliability	12
3.6.2 Availability	12
3.6.3 Security	12
3.6.4 Maintainability	12
3.6.5 Portability	12
3.7 <i>Organizing the Specific Requirements</i>	12
3.7.1 System Mode	12
3.7.2 User Class	12-13
3.7.3 Objects	13
3.7.4 Feature	13
3.7.5 Stimulus	13
3.7.6 Response	14
3.7.7 Functional Hierarchy	14
3.8 <i>Additional Comments</i>	14
4. Change Management Process	14
5. Document Approvals	14
6. Supporting Information	14

1. Introduction

1.1 Purpose

This document designed to give a complete description of the requirements for the Smart Gate System 1.0 software. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 Scope

The system is called "Smart Gate".

The system is designed for parking lots where management wants to allow access only to authorized persons.

The system is suitable for all types of parking lots: private parking lots, employee parking lots, etc.

Our system will be made up of two parts: the first part is a system that recognizes the driver's face and license plate of his vehicle using algorithms, and the second part is an application designed for the parking lot guard who will connect to the system of face recognition and license plate so that the guard can know in real-time who enters the parking lot.

If the system fails to identify the face and license plate, the guard will have to verify details with the driver (ID card, Phone number, etc) and open the parking lot gate for him through the app.

The system will also save security costs by having the guard supervise several parking lots at the same time.

1.3 Definitions, Acronyms, and Abbreviations.

- **GUI** - Graphical User Interface
- **DB** - DataBase
- **JSON** - JavaScript Object Notation. An open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value).
- **Raspberry Pi** - Raspberry Pi is a series of small single-board computers.
- **LCD** - Liquid Crystal Display. electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers.
- **Owner** - Owns the system.
- **Client** - The people who enter the parking lot.
- **User** - The guard who uses the app to authenticate the people entering the parking lot.

1.4 References

- **Face recognition algorithm** -
<https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>
- **License plate recognition algorithm** -
<https://medium.com/programming-fever/license-plate-recognition-using-opencv-python-7611f85cdd6c>

1.5 Overview

The SRS document contains detailed information about the product and the system, both for the client and the programmer.

Section 2 is written for the customer. Gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

Section 3 is written primarily for programmers. It lists all the functional and non-functional requirements of the product and explains the system setup.

2. The Overall Description

2.1 Product Perspective

Smart Gate is a new system designed for parking lots.

The system prevents unauthorized people from entering the parking lot by identifying the driver's face and license plate.

The system allows tracking of the people entering the parking lot in real-time.

The context diagram in Figure 1 illustrates the external entities and system interfaces.

The system is expected to evolve.

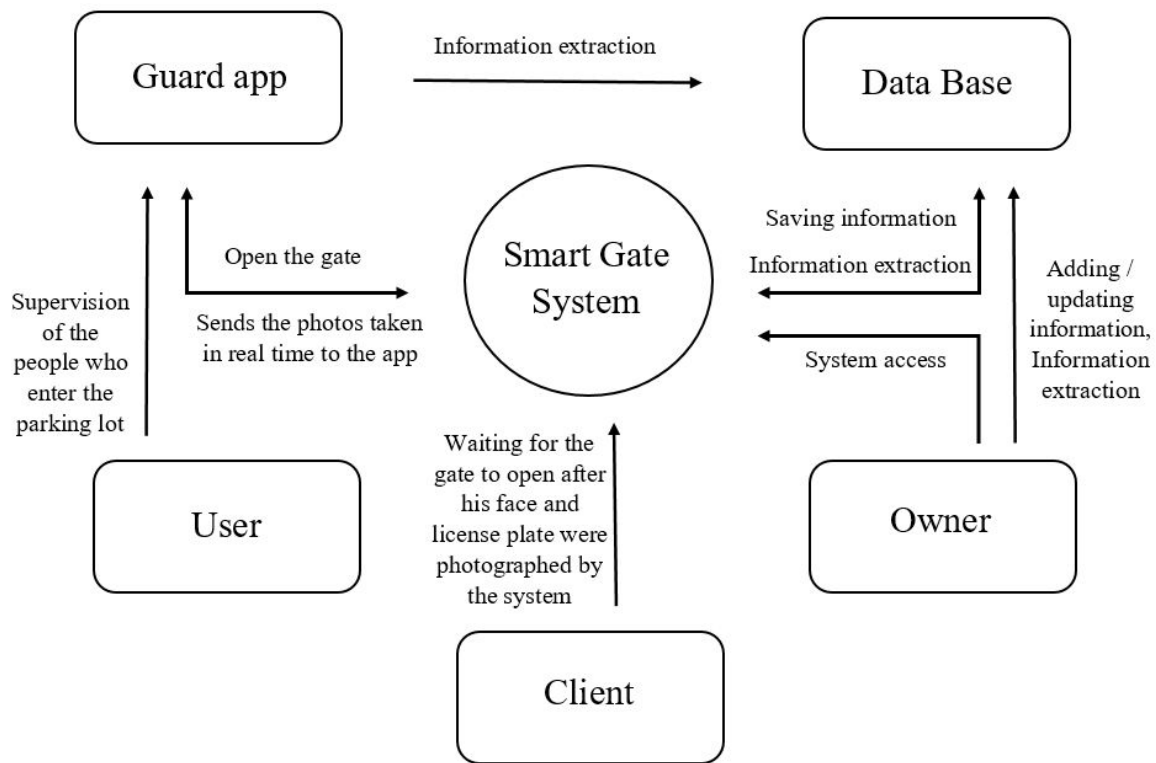


Fig.1

2.1.1 System Interfaces

- The system will perform face recognition and license plate recognition whenever it receives images from the camera. After analyzing the images using the algorithms, the values obtained will be checked against the values in the DB.
- The Android app will be activated by entering the username and password of the guard. The role of this system is to monitor those entering the parking lot and, if necessary, verify the driver's details and open the parking lot gate. The app must be linked to the DB so that you can retrieve the driver's information, and update in real-time who is entering the parking lot.

2.1.2 Interfaces

- The system will be used on Windows or Linux platforms.
- Internet connection required.
- Development and illustration will be supported by Python and Java technologies.
- The application is written in Java language and only supports the Android operating system.

2.1.3 Hardware Interfaces

Our system also consists of hardware:

- There are 2 cameras connected to some port, these cameras are an entrance to the microprocessor. (example: By getPicture function we will get the image from the camera).
- In addition, there is an LCD SCREEN - a monitor that is an output entrance from the microprocessor connected to an output port and displays messages to the user (by the SetText function).

2.1.4 Software Interfaces

- Android Studio - is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.
- Firebase - is a platform developed by Google for creating mobile and network applications. This is a key / value-type NoSQL database type but also similar to a DB document.

2.1.5 Communications Interfaces

There are no external communications interface requirements.

2.1.6 Memory Constraints

For optimal use of the program, the microprocessor should support up to 32 GB of operating system memory and the software it wanted should take up no more than 15 gigabytes so that memory for multiple images remains so that each HD image obtained is at least 1 MB.

2.1.7 Operations

- The system works 24/7
- When the client arrives at the parking lot, he must press the red button that will activate the 2 cameras installed in the parking lot.
The cameras will photograph the license plate of the car and the face of the driver.
The system will transfer the images to the user (guard) app.
At the same time, image processing algorithms will work on the images and try to find a match with the data in DB.

2.1.8 Site Adaptation Requirements

- Internet connection required.
- Android operating system for the app.
- The owner must enter all employee information into the DB.
- The owners need to purchase from the company "Smart Gate System": two cameras that will be installed in the parking lot and photograph the license plate of the car and the face of the driver, a parking lot entrance stand that includes a red button to activate the cameras and a monitor that will display the verification process carried out in the system in real-time.
The system will be connected to the parking lot gate for it to open automatically when authentication is verified successfully.

2.2 Product Functions

- Face recognition Function - in this function, an algorithm for face recognition of the driver entering the parking lot will be activated.
- License plate identification function - in this function, an algorithm for identifying the license plate of the vehicle entering the parking lot will be activated.
- Function for opening the parking gate - a function that opens the parking gate after verifying the driver's details.
- Function that connects the system to the application - this function will connect the system to the application and transmit information to the guard in real-time.
-

2.3 User Characteristics

- Client - owns a driver's license.
- User - a person authorized to engage in security. A person who passed certification or is licensed in the field.
- Owner - familiar with the basic concepts of using a personal computer.

2.4 Constraints

The system will have a set time to identify the driver's face and the license plate of the car and verify them with the details in the DB in order to open the gate in the shortest time and prevent traffic jams at the entrance to the parking lot.

2.5 Assumptions and Dependencies

- The assumption that the details of the persons authorized in the system are correct.
- The assumption that each person whose authorized details are in the system and a person who is not authorized whose details are not in the system.
- The assumption that the parking lot gate will remain open until the vehicle enters the parking lot safely.

2.6 Apportioning of Requirements.

Version 1.0 - Development of algorithms that will identify the driver's face and license plate images and verify them against the images in the DB.

Version 1.1 - Running the algorithms on images in real-time, developing the guard app, and linking the system to it to receive the images in real-time.

Version 1.2 - Link the guard app to DB and the parking lot gate.

Version 1.3 - GUI development for the owners through which the details of authorized persons will be updated.

3. Specific Requirements

3.1 External Interfaces

[SKIP THIS PART]

3.2 Functions

- The system will have a button that will activate the 2 cameras in the parking lot
- The system will make it possible to save the license plate numbers of vehicles entering the parking lot in real-time.
- The system may show the driver the identification process in real-time through a monitor.
- The system will recognize a person's facial features using face recognition algorithms.
- The system will recognize numbers and letters that appear on a license plate using license plate recognition algorithms.
- The system will have limited time to complete the identification process at the entrance to the parking lot.
- The system will allow the storage of the guard and driver data in the DB.
- The system will allow the information of the authorized persons in the system to be updated.
- The system will be linked to the application to transfer data to it.
- The system will allow driver data to be verified manually through the app.
- The system will allow the gate to be opened automatically after verifying the details in the system.
- The system will allow the gate to be opened manually through the app if the verification of the details fails

3.3 Performance Requirements

[SKIP THIS PART]

3.4 Logical Database Requirements

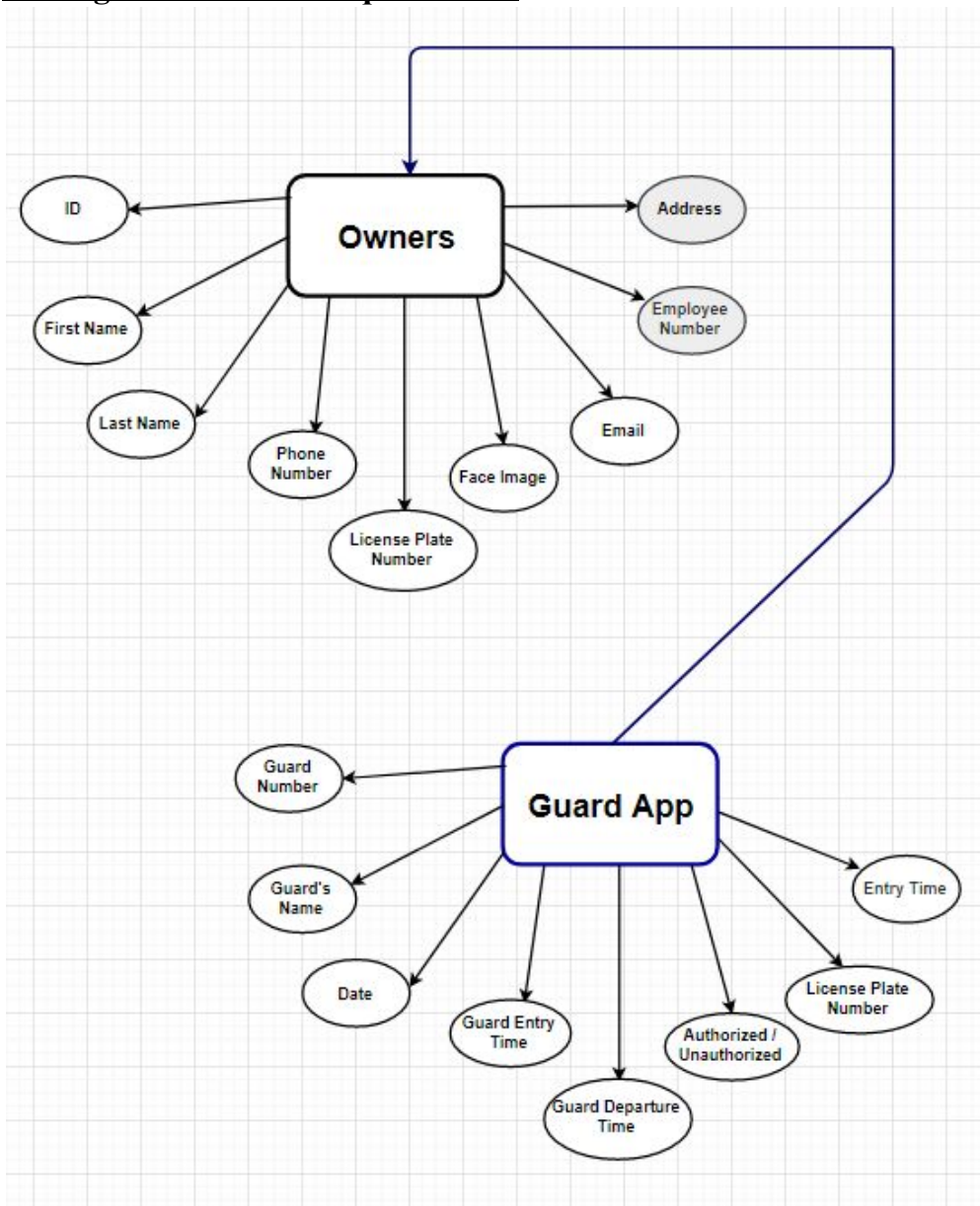


Fig. 2

3.5 Design Constraints

- We will use a free DB that is limited in space and time of use.
- Lack of memory in Raspberry Pi .

3.5.1 Standards Compliance

[SKIP THIS PART]

3.6 Software System Attributes

[SKIP THIS PART – FILL ONLY SECTIONS “3.6.2 Availability”, “3.6.3 Security”]

3.6.1 Reliability

3.6.2 Availability

- The system will allow the guard to verify the driver's details in case the identification fails.
- The system may allow the driver to press the button again after a few minutes to re-identify if the identification fails and there is no response from the guard

3.6.3 Security

- Only the owners will have access to the system with the details of the people authorized to enter the parking lot. The login information must be entered at the entrance to the system and thus the privacy of the employees will be preserved.
- To use the app, the guard must enter the login information.

3.6.4 Maintainability

3.6.5 Portability

3.7 Organizing the Specific Requirements

3.7.1 System Mode

The system will have only one mode.

The system will identify the driver's face and the license plate of his vehicle and open the gate in the parking lot for him.

3.7.2 User Class

The system contains three types of users:

1. Owner- this user has access to the entire system without permission restriction. He can update the details of the authorized persons in the system and also he has the option to see the license plates that have entered the parking lot in real-time.
2. Parking lot guard - this user has access to the app only. It has the option to verify the driver information in the system manually and in addition to open the parking lot gate through the app.

3. Driver - this user has no access to the system other than pressing the button in the parking lot entry position to open the cameras.

3.7.3 Objects

Smart Gate System:

- Functions:
 - Perform training on images using face recognition and license plate algorithms.
 - Run the model on real-time photos.
 - Send real-time photos to the app.
 - Set a time for the identification process
 - Write to DB
 - Read from DB

Application:

- Functions:
 - Receive the photos in real-time from the system.
 - Write to DB
 - Read from DB
 - Opening of the parking lot gate

License plate:

- Functions:
 - Get license plate in string (Letters and Numbers)
 - Write to DB

Face:

- Functions:
 - Get face photo
 - Write to DB

3.7.4 Feature

[SKIP THIS PART]

3.7.5 Stimulus

Pressing the button at the entrance to the parking lot causes the parking lot cameras to turn on and photograph the driver's face and license plate of his vehicle. The system receives the images and as a result, the identification algorithms start working.

3.7.6 Response

The system receives the data results after activating the identification algorithms and its response is to open the parking lot gate.

3.7.7 Functional Hierarchy

We have already organized our requirements as above.

3.8 Additional Comments

[SKIP THIS PART]

4. Change Management Process

[SKIP THIS PART]

5. Document Approvals

[YOUR SUPERVISOR - Dr. Amos Azaria]

Identify the approvers of the SRS document. Approver name, signature, and date should be used.

6. Supporting Information

[SKIP THIS PART]