Specification of software requirements

Subjec: Recognition of Plates

Date	Review	Description	Author
12/01/2020			Cortez Jonathan Coyago Eunice Cují Roberto Garcés Christian Guerra Luciana

Document validated by the parties on date: [Fecha]

For the client	By the supplying company
Universidad Internacional del Ecuador	SigmaProgrammers
Signed. Mr. / Mrs.[Nombre]	Signed. Mr / Mrs[Nombre]

Content

1 Introduction

Our vision is to carry out a program which allows us to recognize license plate numbers, based on a photograph, through a txt database that will inform us if the known license plate is accepted for entry or not.

.

1.1 Purpose

1.1.1 Automatically identify alphanumeric characters from an image.

1.1.2. Convert the text that appears in an image into a text file, generating an input confirmation.

1.1.3. Digitize the information.

1.2 Scope

The purpose of this document is to define the functional and non-functional specifications for the development of a license plate recognition system that will allow vehicle access to a parking place, this system will be used by students, teachers and managers.

1.3 Involved personnel

Name	Cortez Tamayo Jonathan Andrés
Role	Designer and programmer
Professional category	IT engineering student
Responsibilities	Desing and programming of the PRP
Contact information	jacortez3@espe.edu.ec

Name	Coyago Arce Eunice Pamela
Role	Designer and programmer
Professional category	IT engineering student
Responsibilities	Desing and programming of the PRP
Contact information	epcoyago@espe.edu.ec

Name	Cují Revelo Milton Roberto
Role	Designer and programmer
Professional category	IT engineering student
Responsibilities	Desing and programming of the PRP
Contact information	mrcuji@espe.edu.ec

Name	Garcés Mosquera Chistian Marce
Role	Designer and programmer
Professional category	IT engineering student

Responsibilities	Desing and programming of the PRP
Contact information	Cmgarces2@espe.edu.ec

Name	Guerra López Luciana Noemi
Role	Designer and programmer
Professional category	IT engineering student
Responsibilities	Desing and programming of the PRP
Contact information	lnguerra@espe.edu.ec

1.4   Definitions, acronyms and abbreviations

Name	Description
PRP:	Plate Recognition Program
User:	Person who will use the system to manage processes

1.5   References

Reference	Title	Route	Date	Author

[Ref.]IEEE	Standard IEEE 830 - 1998			

1.4   Summary

This project wishes to establish the use of technology through image recognition, to solve certain problems that are established in people's daily lives, the same that helps to facilitate certain actions and avoid wasting time, as well as the security issue Of the same. Optical character recognition is a text recognition process in most cases it is focused on the digitization of texts through an image, for this it is necessary that they are within the alphabet that is going to be used. From an image, a program will be implemented that allows recognizing characters, this information will be processed and stored in a text file to later be used or implemented in the automation of data entry in a text file, which implies a saving of human resource required for this action in order to improve or implement a service.

This project is limited to a parking lot which will help to recognize the license plates of each car when entering or leaving a parking lot and in turn recognize it in its database.

2 General description

2.1   Product perspective

The system establishes an image treatment which can be treated and dimensioned, in order to understand the text that it is composing, taking as an example a car license plate, which we can determine which license plate is and this text can be treated and the programming language I can understand it and it will pass to a string.

UML Diagram
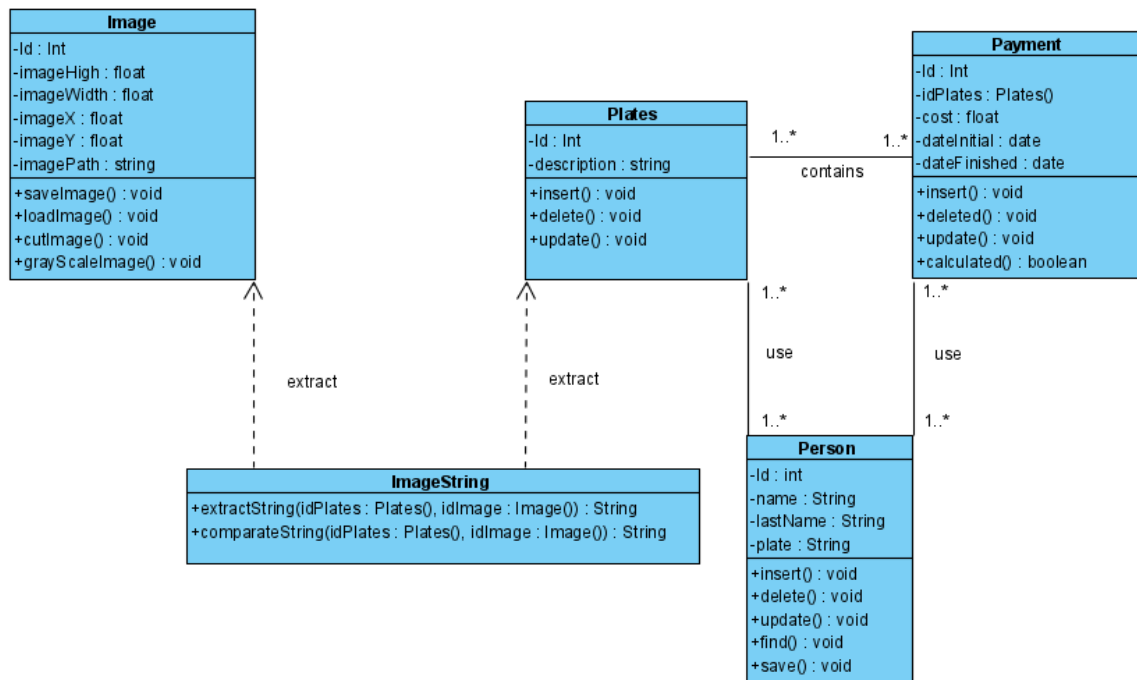
UML Diagram Description

Image
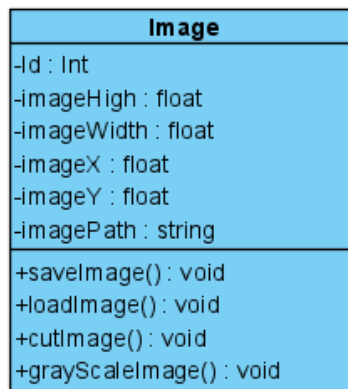
Attributes
ID: Unique identifier of the entire type image.
imageHigh: height of the float image.
imageWidth: width of the float image
imageX: position on the x-axis of the start of the image to be cut as a float
imageY: y-axis position of the start of the image to be cut as a float
imagePath: location of the string image.

Methods
saveImage(): Void method to save the image.
loadImage(): Void method to load the image to be processed.
cutImage(): Void method to cut the image.
grayScaleImage(): Void method to change the grayscale of the image.
Payment

```
                 Payment
-Id : Int
-idPlates : Plates()
-cost : float
-dateInitial : date
-dateFinished : date

+insert() : void
+deleted() : void
+update() : void
+calculated() : boolean
```

Attributes
Id: unique identifier of the int.
idPlates: identifier belonging to the plates type int class.
cost: variable that represents the payment made type float.
dateIntial: variable that determines the date of entry of the float payment.
dateFinished: variable that determines the end date of the float payment.

Methods
insert(): Void method in which it allows the entry of a payment.
deleted(): Void method in which it allows the removal of a payment.
update(): Void method in which you can update a payment.
calculated(): Method that returns a boolean in which it allows to generate and calculate the payment made and determines whether or not it was made.

Plates

```
            Plates
-Id : Int
-description : string

+insert() : void
+delete() : void
+update() : void
```

Attributes
Id: unique identifier of the int.
description: variable that allows to describe or store the string type plate.

Methods
insert(): Void method in which it allows the entry of a board.
deleted(): Void method in which you can delete a board.
update(): Void method to update a board.

ImageString

```
                        ImageString
+extractString(idPlates : Plates(), idImage : Image()) : String
+comparateString(idPlates : Plates(), idImage : Image()) : String
```
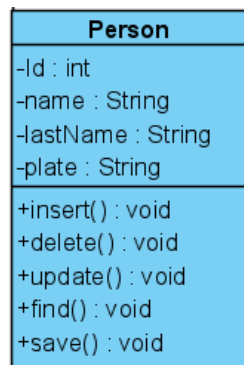
Methods
extractString(idPlates,idImage): Method that has as input the image and plate parameters, which return a string of the treated image.
ComprateString(idPlates,idImage): Method that has as input the parameters of plates and image, which return a string of the comparison of the string or its existence with the plates entered and paid.

Person

Attributes
Id: unique identifier of the int.
name: variable that allows to save the name of a person of type string.
lastName: variable that allows to save the surname of a person as a string.
plate: variable that allows to enter the series of a string plate.

Methods
**insert():** Void method in which it allows the entry of a plate or data of the person.
deleted(): Void method in which it allows the removal of a plate or data of the person.
update(): Void method in which it allows to update a plate or person's data.
find(): Void method in which it allows the entry of a plate or data of the person.
save(): Void method that allows to save data of a plate or a person.
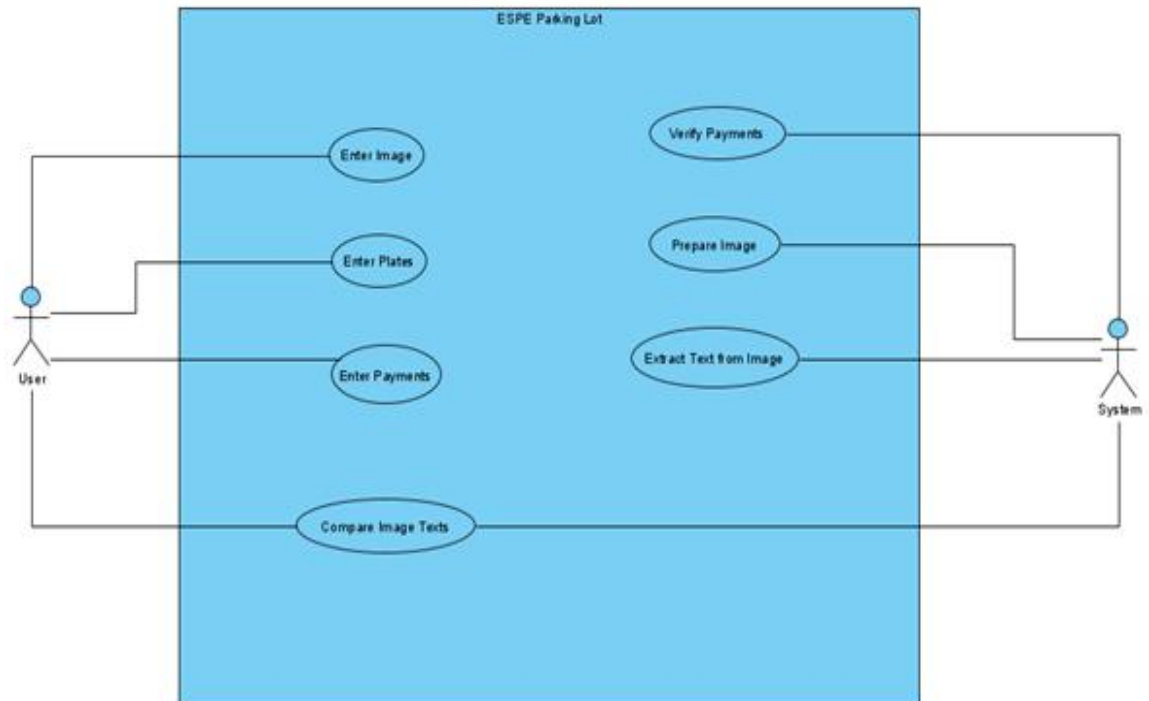
DEPENDENCE
- - - - - - - ->
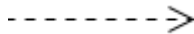
Dependency which momentarily uses the attributes of the other class.

Generalization to determine that it requires elements of the other class.

.

2.2 Product functionality

2.3 User characteristics

Type of user	Average person
Training	Basic technological knowledge
Activities	Enter information into the system

Type of user	Student

Training	Academic
Activities	School activities
Type of user	Educator
Training	Professional
Activities	Teaches

2.4 Restrictions

- The must be a camera connected for the program to work
- The program was designed for an environment with cars
- Language and use of technologies: Java
- The system must have a small database

2.5 Assumptions and dependencias

- The requirements described here are assumed to be stable
- The computers on which the system will be run must meet the requirements indicated above to guarantee a correct execution of the same.

3 Specific requirements

Requirement number	1
Requirement name	User authentication
Characteristic	Users must identify themselves to access the system

Description of the requirement	The system can be consulted by any user depending on their level of accessibility
Requeriment priority	High

3.1 Common interface requirements

3.1.1 User interfaces

The user interface will consist of a set of text fields. This must be built specifically for the proposed system and will be viewed from a screen.

3.1.2 Hardware interfaces

- camera
- mouse
- keyboard
- Computer
  .

3.1.3 Software interfaces

- Operating System: Windows XP or higher
- Explorer: Mozilla or Chrome.

3.1.4 Communication interfaces

Servers, clients and will communicate with each other, using standard internet protocols, whenever possible.

.

3.2   Functional requirements

3.2.1 Functional requirement 1

User authentication: users must be identified to save them in the system.The user must provide data such as:, Name, Surname, User and Password, plates

3.3   Non-functional requirements

3.3.1 Performance requirements

Ensuring that the design of the queries or plate scan or other process does not affect the performance of the database.

Error! Bookmark not defined.

3.3.2 Security

- Guarantee the reliability, security and performance of the computer system to the different users.
- Guarantee the security of the system with respect to the information and data that are handled such as documents, files.
- Facilities and controls to allow access to information to authorized personnel, with the intention of consulting and uploading pertinent information for each of them

3.3.3 Reliability

- The system must have an intuitive and simple user interface
- The user interface must conform to the characteristics of the institution, within which the
- The system will be incorporated.

.

3.4   Other requirements

4 Appendices