Subject: Object Oriented Programming

NRC: 2858

Members:

Cortez Tamayo Jonathan Andrés

Coyago Arce Eunice Pamela

Cují Revelo Milton Roberto

Garcés Mosquera Christian Marce

Guerra López Luciana Noemi

Subject: Recognition of Plates

Contenido

General objective

- Develop a machine vision system, using object oriented programming, to identify the text in an image.

Specific Objectives

- Define the process of an image, knowing its parameters and dimensions, to recognize and establish its measures and content.
- Distinguish the images entered, with the loading or uploading of images, to know the type of image processing
- Decompose the entered image, with the recognition using text recognition to know the text that the image entered or to be treated.

Theory

Glossary

- **Object Oriented Programming:** Manipulates input and output data through specific objects.
- **Library:** Implementation and coding of a programming language.
- **Class:** Set of variables.
- **Variable:** Values or operations to take or process.
- **Programmable Function:** Main algorithm that allows to solve a specific task.
- **Algorithm:** Set of instructions.
- **Object:** Information elements characterized by real data.
- **Method:** Actions that the object can do by itself.
- **Attribute:** Variables related to the state of an object.
- **Event or Behavior:** Realizable actions.
- **Messages:** Data transmission.
- **Instance:** Call of an object.
- **String:** Data type that interprets a text string.

Definition

The system establishes an image treatment which can be treated and sized, in order to understand the text that composes taking as an example a car plate, which we can determine that plate is and this text can treat and programming language can understand and go to a string.

Problem

An image is known which cannot be interpreted by the machine and its text is not perceptible by the programming language and this can be used or treated as a text string.

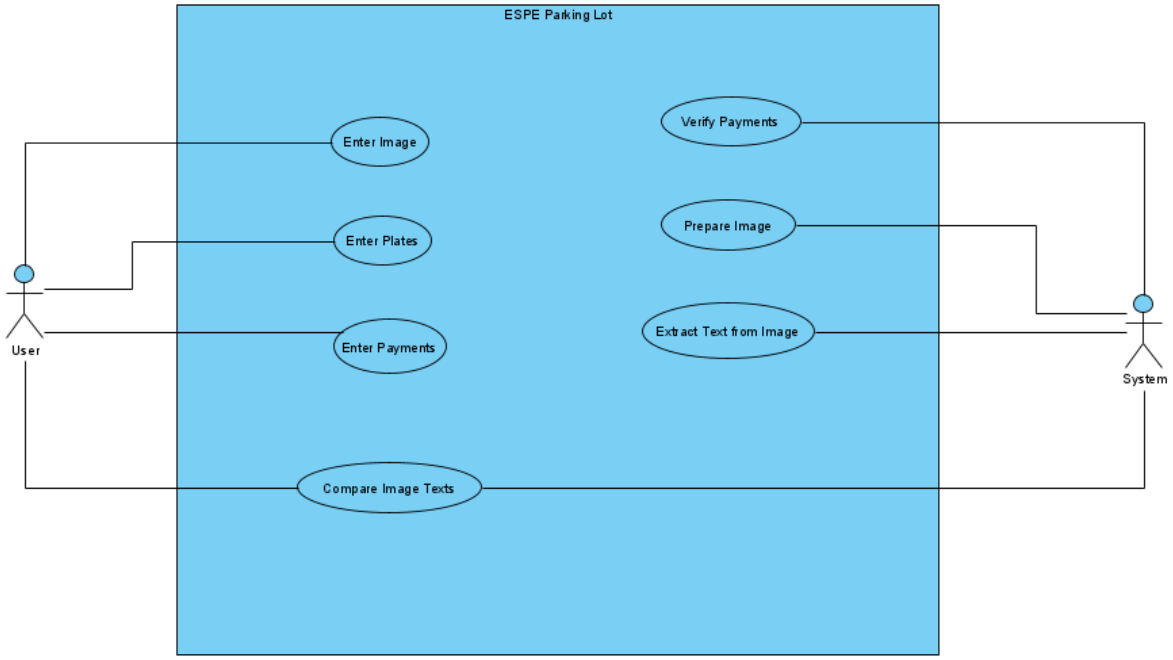
Solution

It is to establish a system in which it contemplates the use of programming oriented to objects, to be able to determine or to read the text that an image has in case of an image of the plates of a car this will determine the text to be able to be treated and to be interpreted by the machine.

Scope

The system will allow the entry of an image so that it can treat the image and recognize the text that it has delimiting that the image will be entered manually by the user.

Use case diagram



Use case description

Name:	Enter image
Autor:	Cortez Tamayo Jonathan Andrés Coyago Arce Eunice Pamela Cují Revelo Milton Roberto Garcés Mosquera Christian Marce Guerra López Luciana Noemi
Date:	19/01/2021
Description: Allows the user to enter an image into the system.	
Actors: User	
Preconditions: The actor must enter the system.	
Normal flow 1. The actor entered the system correctly. 2. The actor loads the image to be treated.	
Alternative flow The system checks if the image corresponds to the established format. If not, the system will issue an error message "IMAGE NOT CORRESPONDING".	
Postconditions: The actor has loaded the image correctly.	

Name:	Enter image
Autor:	Cortez Tamayo Jonathan Andrés Coyago Arce Eunice Pamela Cují Revelo Milton Roberto Garcés Mosquera Christian Marce Guerra López Luciana Noemi
Date:	19/01/2021
Description: Allows the user to enter data from the plates.	
Actors: User	
Preconditions: The actor must enter the system.	
Normal flow: 1. The actor entered the system correctly. 2. The actor loads the data from the plates.	
Alternative flow: The system checks if the data is correct, otherwise it will issue an error message "DATA INPUT IS INCORRECT".	
Postconditions: The actor has correctly loaded the plate data.	

Name:	Enter payments
Autor:	Cortez Tamayo Jonathan Andrés Coyago Arce Eunice Pamela Cují Revelo Milton Roberto Garcés Mosquera Christian Marce Guerra López Luciana Noemi
Date:	19/01/2021
Description: It allows the user to enter the payments corresponding to the plates.	
Actors: User	
Preconditions: The actor must enter the system.	
Normal flow: 1. The actor entered the system correctly. 2. The actor loads the data from the plates.	
Alternative flow: The system checks if the data is correct, otherwise it will issue an error message "DATA INPUT IS INCORRECT".	
Postconditions: The actor has correctly charged the plate payments.	

Name:	Verify payments
Autor:	Cortez Tamayo Jonathan Andrés Coyago Arce Eunice Pamela Cují Revelo Milton Roberto Garcés Mosquera Christian Marce Guerra López Luciana Noemi
Date:	19/01/2021
Description: Allows the system to check if payments are made.	
Actors: System	
Preconditions: Payment data must be entered.	
Normal flow: 1. The system will search for the existence of the plates. 2. The system will verify if you are with the right payments.	
Alternative flow: The system will verify if the payments and the plates are correct, otherwise it will issue an error message "PLACES IN EXISTING OR PAYMENTS ARE NOT UP TO DATE".	
Postconditions: The system gives proper passage to the payments on the plates.	

Name:	Prepare image
Autor:	Cortez Tamayo Jonathan Andrés Coyago Arce Eunice Pamela Cují Revelo Milton Roberto Garcés Mosquera Christian Marce Guerra López Luciana Noemi
Date:	19/01/2021

Description:

Allows the system to treat the image by setting the desired dimensions and colors.

Actors:

System

Preconditions:

The image must be previously loaded.

Normal flow:

1. The system cuts the image to the desired dimensions depending on the image loaded.
2. The system will change the color tone to gray scales.

Alternative flow:

The system analyses the image treatment and if the dimensions exceed the established standard it will issue an error of "INADEQUATE DIMENSIONS".

Postconditions:

The system will create the image in grayscales.

Name:	Extract text from image
Autor:	Cortez Tamayo Jonathan Andrés Coyago Arce Eunice Pamela Cují Revelo Milton Roberto Garcés Mosquera Christian Marce Guerra López Luciana Noemi
Date:	19/01/2021
Description: The system will extract the text from the image in a text string.	
Actors: User	
Preconditions: The system contains the image in grayscales for text extraction.	
Normal flow: 1. The system will correctly extract the text string that has the image	
Alternative flow: The system checks if the image is treated properly and in the opposite case, the text will not be adequate.	
Postconditions: The text string will be printed and may be visible to both the system and the user.	

Name:	Compare image text
Autor:	Cortez Tamayo Jonathan Andrés Coyago Arce Eunice Pamela Cují Revelo Milton Roberto Garcés Mosquera Christian Marce Guerra López Luciana Noemi
Date:	19/01/2021

Description:

It allows the actor to buy the text of the image processed the car data entered.

Actors:

User

Preconditions:

The system must have prepared the image.

Normal flow:

1. The actor will be able to see the text string of the entered image.
2. The system will buy if the text exists in the car data.

Alternative flow:

The actor will observe that the text string and this one if it does not exist will emit an error message "PLATE DOES NOT EXIST".

Postconditions:

The actor will be able to buy the chain extracted with the car data.

UML Diagram

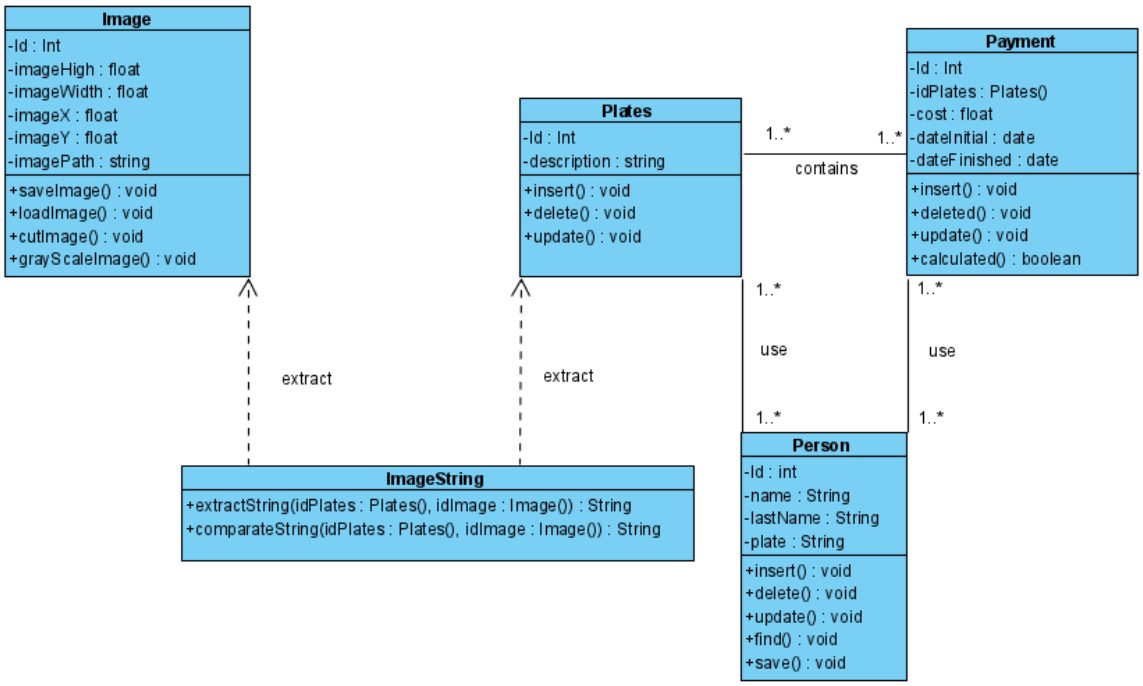
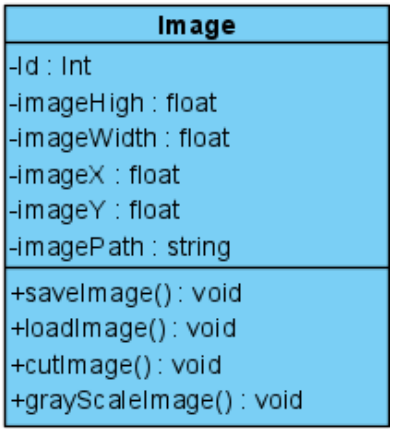


Image
-Id : Int
-imageHigh : float
-imageWidth : float
-imageX : float
-imageY : float
-imagePath : string
+saveImage() : void
+loadImage() : void
+cutImage() : void
+grayScaleImage() : void
Plates
-Id : Int
-description : string
+insert() : void
+delete() : void
+update() : void
Payment
-Id : Int
-idPlates : Plates()
-cost : float
-dateInitial : date
-dateFinished : date
+insert() : void
+deleted() : void
+update() : void
+calculated() : boolean
1..*
1..*
contains
extract
extract
1..*
use
1..*
1..*
use
1..*
ImageString
+extractString(idPlates : Plates(), idImage : Image()) : String
+comparateString(idPlates : Plates(), idImage : Image()) : String
Person
-Id : int
-name : String
-lastName : String
-plate : String
+insert() : void
+delete() : void
+update() : void
+find() : void
+save() : void

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

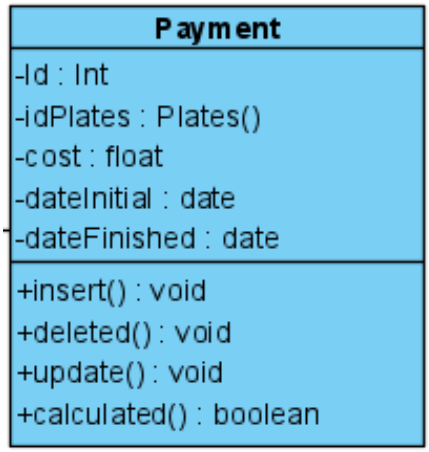
saveImage(): Void method to save the image.

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



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

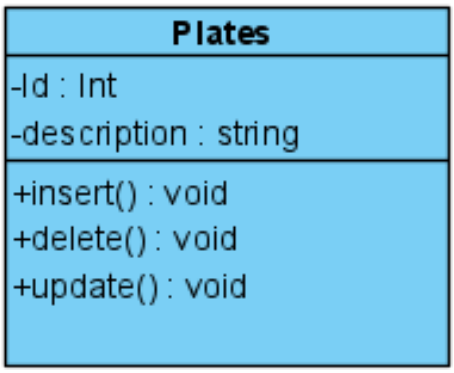
insert(): Void method in which it allows the entry of a payment.

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



Attributes

Id: unique identifier of the int.

description: variable that allows to describe or store the string type plate.

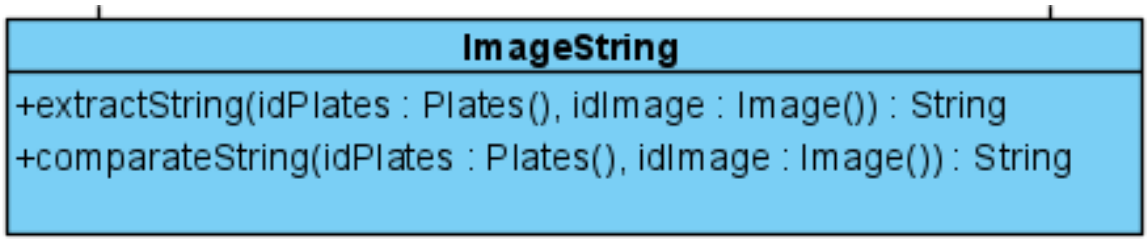
Methods

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

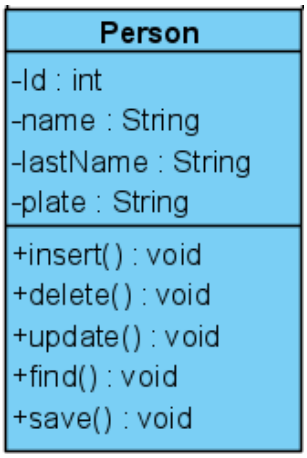


Methods

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.

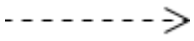
deleted(): Void method in which it allows the removal of a plate or data of the person.

update(): Void method in which it allows to update a plate or person's data.

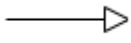
find(): Void method in which it allows the entry of a plate or data of the person.

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.