Auto Car Classifier
Outline of Requirements and Design Documentation

Edited by:

Fiwa Lekhulani
Abhinav Thakur
Vincent Soweto
Andrew Jordaan
Keorapetse Shiko

Contents

1 Introduction

In today's day and age of automating tasks, our system's aim is to automate the process of classification of cars using convolutional neural network. The main purpose of this document is to specify the requirements and capabilities of the AI Auto Car Classifier. This document will provide the domain model, user characteristics which contains the intended users of the system and their use of the system, functional requirements which contains the use cases that needs to be satisfied by our system, quality requirements which specifies each of the quality requirements relevant to our system and trace-ability matrix which contains a matrix with functional and quality requirements as rows and the subsystems as columns.

2 Domain Model

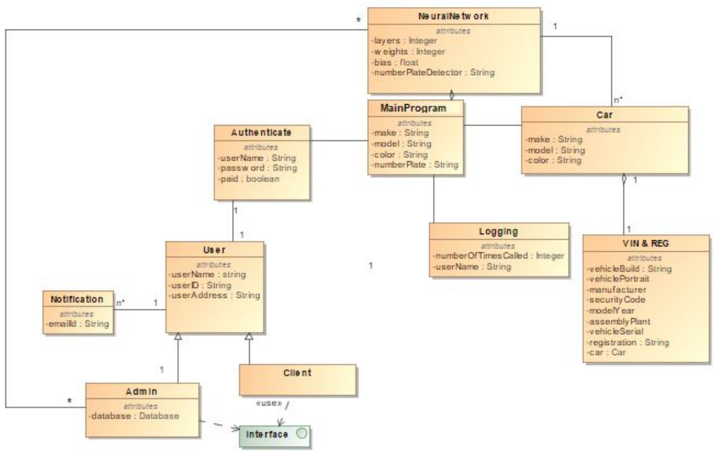

Figure 1: Domain Model

3 User Characteristics

The intended users of are car dealearship consultants that keep an inventory of all the cars in their organization. The system can can however be used by any individual looking to classify an unknown vehicle.

There are two types of users in the system. The first is the Administrator, who will keep track of the systems behaviour and logged events, while also investigating reasons behind some incorrectly classified vehicles. The second type of user is the general user, who will be the main user of the system. This user will be able to classify vehicles according to color, make, model, year and and other distinguishable car characteristics.

4 Functional Requirements

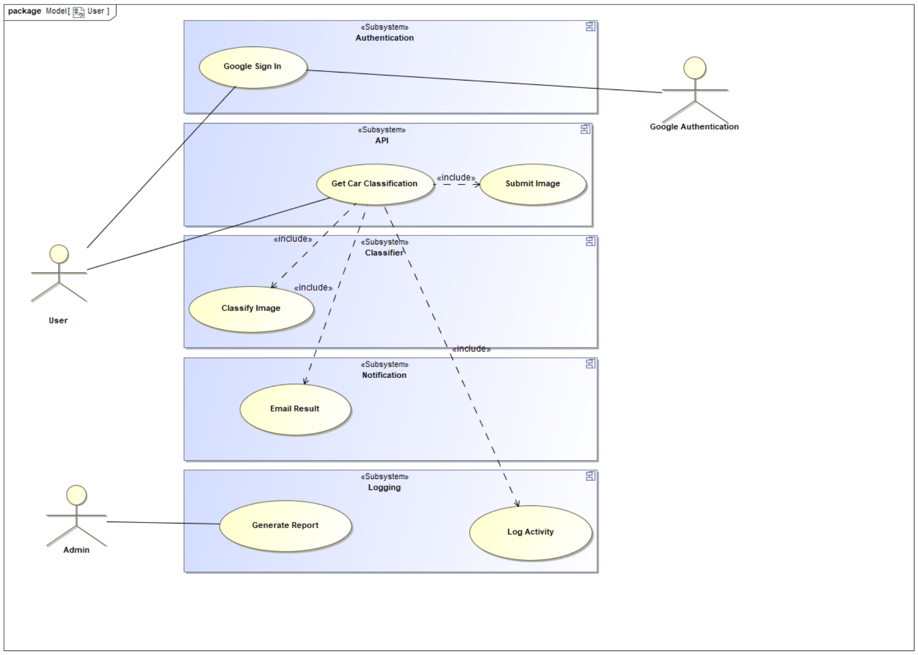

Figure 2: Use Case Diagram

4.1 Classification

The system must be able to capture an image.

The system must be able to accept a picture from the front, standard boom-gate height.

The system must be able to identify a vehicle.

The system must be able to identify a vehicles color.

The system must be able to identify a vehicles manufacturer based on the vehicle badge

The system must be able to identify a vehicles make, model and year.

4.2 Notification

The system must log all incidents and classify them according to severity

The system must notify the user if the server has queued your request due to the traffic rate.

The system must notify the user through email when results gathered by the classifier take longer to process.

4.3 Authentication

The system must allow users to register using their Google accounts.

4.4 Constraints

The system must use a database which is optimal for more frequent reads than writes

The system must be a web based UI which functions on all modern day browsers

5 Quality Requirements

5.0.1 Performance

The system must efficiently make use of bandwidth to ensure performance on slow connections to the server.

The system must queue the users if more than 10 pictures are being classified.

The system must limit the photo size up to 1MB in order to ensure efficiency.

The system must use of thread pooling in order to reuse our resources and decrease processing time.

The system must use server driven load balancing in order to allocate resources to processes that has not classified the image in the first 15 seconds.

5.0.2 Reliability

The system must be continuously tested and upgraded to improve the accuracy of the predicting

The system must backup all logs.

The system must make use of resource locking in the processing phase of our classification.

The system must make use of deadlock detection mechanisms if a picture is not classified in 30 seconds.

5.0.3 Security

The system must use Google sign-in as a means of a secure authentication.

The system must protect the contents of the website and application from DDos.

The system must use HTTP for requests.

The must have no ways of entering an unwanted state due to unintended operations.

5.0.4 Monitorability

The system must be remotely monitorable.

The system must report its status and usage to Admin Users

The system must report errors or problems

The system must log all successful and failed transaction of a user

5.0.5 Cost

The system must use existing technologies and libraries to keep costs to a minimum

The system must use existing hardware available to user

5.0.6 Usability

The system must be responsive

The system must show the user how busy the server is

5.0.7 Maintainability

The system must be maintained by admin users only

The system must notify the admin on whether classification failed.

The system must be modularised in order for classification sub-system can be trained to latest vehicles launched.

5.0.8 Flexibility

The system must be cross platform and available to all browsers.

6 Architectural design

Our system uses Client-server architecture style where in the our web based UI would be our access layer where the user uploads the picture of a car to our server. Our server application handles the back-end which makes use of the main programs and subroutines architectural style in order to call the classifier subroutines. The classifier subroutines depends on the output of whether the image uploaded is a car or not. If the uploaded image is a car, the main program calls various subroutines to obtain the color, make, model and number plate. After the classification process is done, the results are then displayed on the Client application which is our web-based UI. Our systems main objective is to analyze and process the picture of a car uploaded by a user and transform it into a page of information about the car, i.e. color, make and model. Our system has no interaction between the system and the user whilst the system is processing.Our server side consists of a network of information processing activities which is our neural network with multiple layers which is each responsible for processing the color, make, model and number plate.

Our design goals will be satisfied by us focusing on the following design principles: Information hiding: Our system satisfies information hiding by not displaying the process of how the user input is transformed into an output by the calling of subroutines in our main program. Design for change: Our system satisfies this by ensuring that our system is modular and is fulfills the design objective of ease of change and maintenance. Separation of concerns: Designing the modules so that each is assigned a distinct functionality in order to promote modularity and to ensure faster error detection in case, for unwanted output. Low coupling: Designing the modules to minimize control and data dependencies among modules ensure that the data inputted is computed in the most optimal way possible. High cohesion: Designing the modules to ensure each accomplishes a core functionality in order to have minimum modules to reduce inter-module communication within a system.

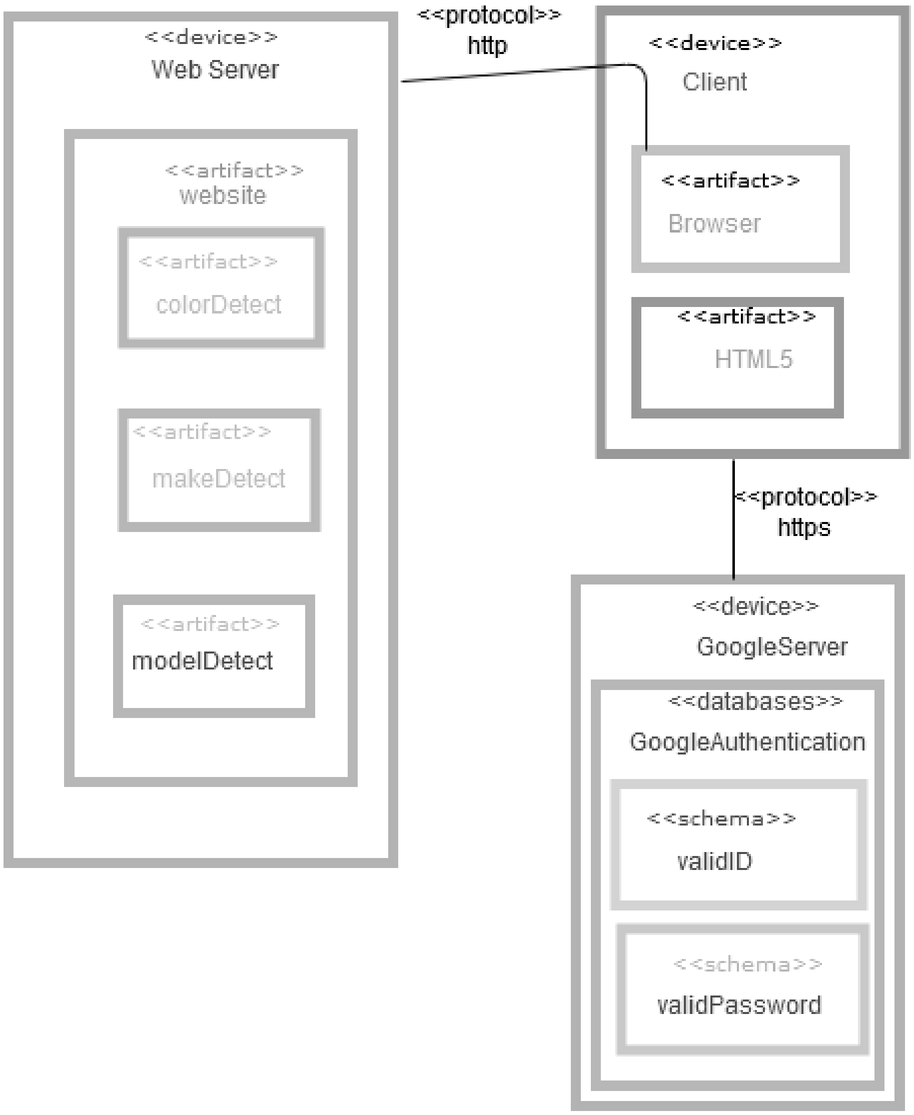

Figure 3: Deployment Diagram

7 Deployment Diagram

8 Traceability Matrix

	Classification	Authentication	Notification
R1	X		
R2	X		
R3	X		
R4	X	X	
R5	X	X	
R6			X
R7			X
R8			X
R9			X
R10			X
R11			X
R12		X	