Auto Car Classifier
Outline of Requirements and Design Documentation

Edited by:

Fiwa Lekhulani
Abhinav Thakur
Vincent Soweto
Andrew Jordaan
Keorapetse Shiko

Contents

1 Introduction

AI-Auto-Car-Classifier is a web application aimed at automating the process of adding cars to a salespersons inventory of cars. This means given a cars image, the system should be able to identify the cars make, model and color whilst also extracting the vehicles number plate if it exists. It should also allow the salesperson to edit incorrect classifications and add additional fields The main purpose of this document is to specify the requirements and capabilities of the AI Auto Car Classifier. This document will provide a domain model, functional requirements of the system, a deployment diagram and the type of users it is intended for. It will also include the subsystem breakdown of the system, quality requirements and a traceability matrix that maps requirements to subsystems

web

2 Domain Model

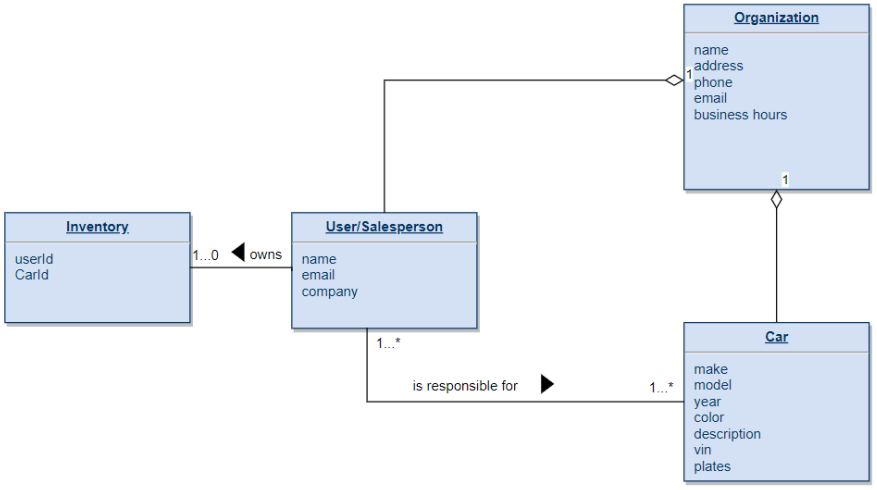

Figure 1: Domain Model

3 User Characteristics

The intended users of are car sales 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 responsible for the retraining and monitoring of the system. 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 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 detect whether an image has a vehicle. system must be able to detect when an image does not contain a car

The system must be able to identify a vehicles color.

The system must be able to identify a vehicles make (Manufacturer)

The system must be able to identify a vehicles model

4.2 Authentication

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

4.3 Constraints

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

The system must must mainly be focused on serving mobile devices

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

The system must ensure that processing of a single image occurs within a reasonable amount of time,

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.

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 denial of service attacks.

The system must use a secure transfer protocol for requests.

The system 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

There should be no cost associated with classifying an image

5.0.6 Usability

The system must be mobile responsive

The system must have a conspicuous, yet non-confusing way of notifying the user if something goes wrong

The system should notify the user if a process is going to take long, or is taking longer than expected.

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

The AI-Auto-Car-Classifier system is built using a custom architectural style which is a combination of the client-server and Main method and Subroutine architectural styles. In this case, our user interface along with its interaction with the user, will represent the client side of client-server architecture; while the server part is customized into an Application Programming Interface (API) that acts as the main method. One of the subroutines the Main program calls is the one for the object persisting model which saves all the classification the user has saved. Each of the Convolutional Neural Network models for learning the detection of a car, make and colour of the vehicle will represent our subsystems in the API respectively. A subroutine to a database management system would also be used to store the details the user wants to store.

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.

7 Deployment Diagram

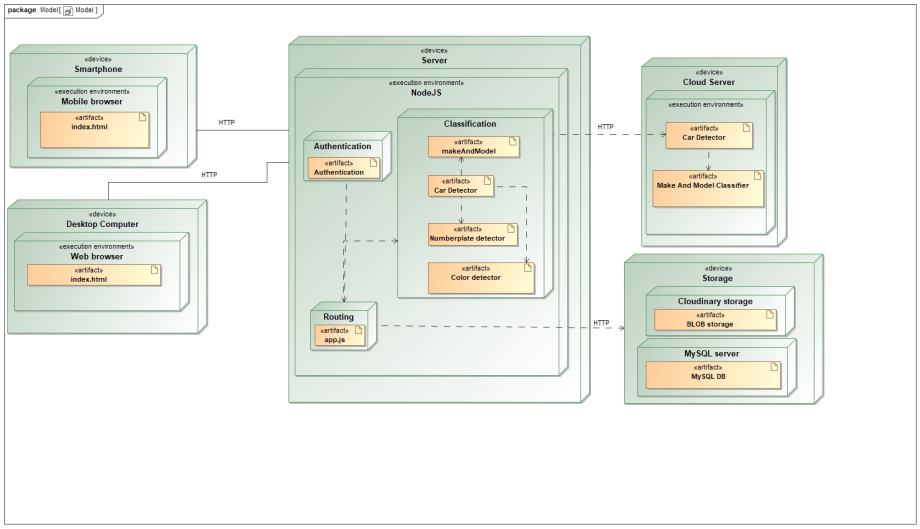

Figure 3: Deployment Diagram

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	
R13		X	
R14		X	
R15		X	
R16		X	
R17		X	
R18		X	
R19		X	
R20		X	
R21	X		
R22	X		
R23	X		
R24		X	
R25		X	
R26	X		
R27	X		
R28	X		
R29	X		
R30	X		
R31	X		
R32	X		
R33	X		
R34	X		