Project:	Auto-Learn
Team No.:	Team 2
Class:	CSE 3310.004 - Spring 2020
Module:	Assignment 4
Deliverable:	Final Binder

Version: [1.6] **Date: [04/29/2020]**

Contributors:

Edrik Aguilera
William Anderson
Ryan Laurents

Revision History

Version number	*Date*	*Originator*	*Reason for change*	*High level description of changes*
1.0	03/05/2020	Edrik Aguilera William Anderson Ryan Laurents	Initial draft	
1.1	03/09/2020	Ryan Laurents	Section 4	Added System requirements
1.2	03/23/2020	Edrik Aguilera	Section 5	Conceptual data model added
1.3	03/25/2020	Edrik Aguilera William Anderson	Section 5 and 3	CDM updated, context diagram added
1.4	03/27/2020	Edrik Aguilera William Anderson	Section 2 updated, Section 3 updated	Context diagram updated, additional business and system requirements added
1.5	03/28/2020	William Anderson	Convert to docx	Move SRA from Google Docs to Microsoft Word
1.6	04/28/2020	Ryan Laurents	Update changes	Updating changes to prep for the final binder.

TABLE OF CONTENTS

1. Introduction and Project Overview

Team 2 has been employed to design and implement a software application for vehicle recognition. This Android application should allow for our customers to upload their own photo or allow for a photo to be taken using the camera. The system will be up and operational during the last week of April 2020 just in time for new vehicles coming out for the 2021 model year. In addition to the minimum set of requirements listed below, Auto-Learn is open to any suggestions that end-users would like to see implemented in the app.

2. Objectives

2.1 BUSINESS OBJECTIVES

The following is a list of business objectives:

Objective 1: Member Registration: All members must provide the following information prior to using the system:

- First Name, Last Name
- UTA ID
- Email address
- Password
- Student or Teacher role

Objective 2: Login functionality: All members must login to the system with a user/password that was established during Member registration stage.

Objective 3: "Account Maintenance" functionality must be supported that allows user to close account and edit personal information and includes the following customer data:

- Email address
- Password
- Delete Account

Objective 4: The user can request classification of up to six different types of Vehicles:

- SUV
- Truck
- Coupe
- Convertible
- Sedan
- Van

Objective 5: The user must be able to view data and statistics of each classification.

Objective 6: The user must be able to upload their own photo of a vehicle or take a photo using the device's camera.

Objective 7: The user must be able to recover their password or username in the event of unauthorized access or loss of credentials.

2.2 SYSTEM OBJECTIVES

The following is a list of system objectives:

Objective 1: System will be an Android application

Objective 2: Tensorflow / Python will be used to implement the machine learning.

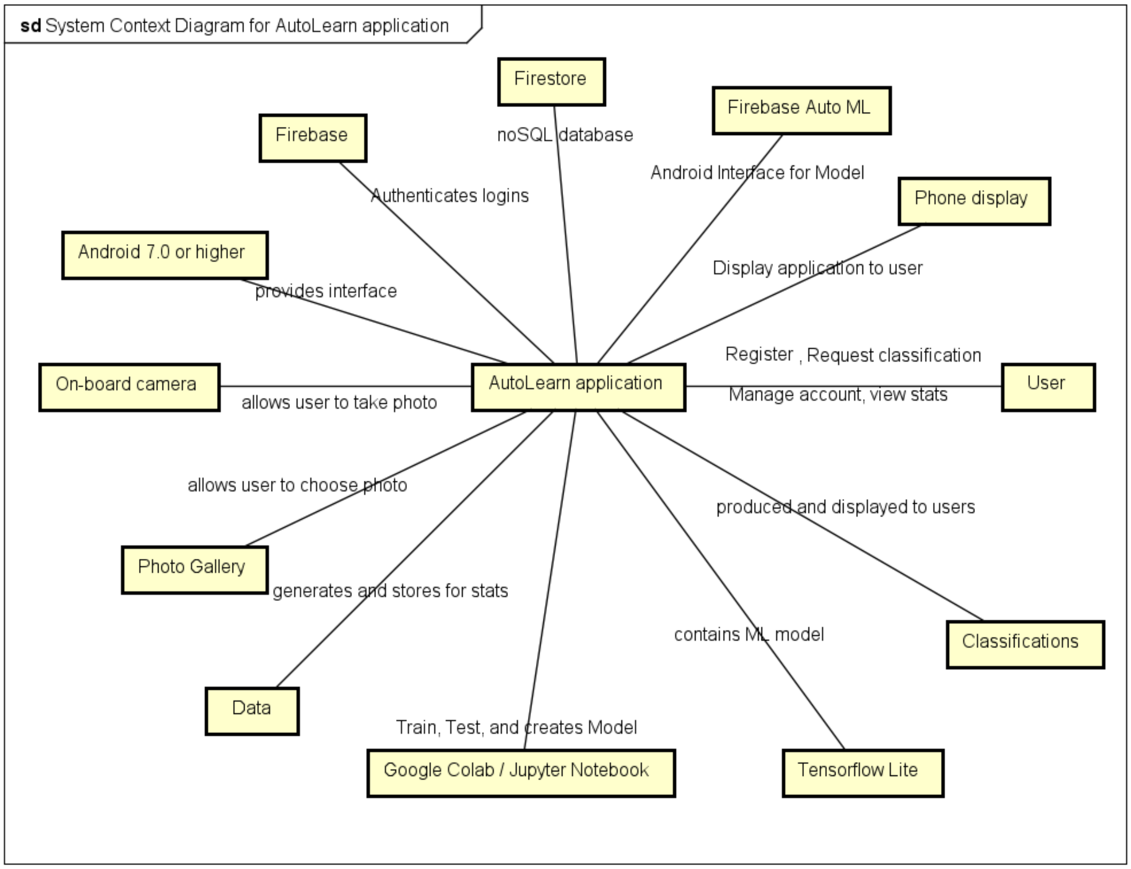
Objective 3: Firebase will be used to manage account credentials.

Objective 4: Java will be used in conjunction with Android Studio in order to develop the UI of the app.

Objective 5: Google's Material Design will be used as the UI theme of the app

Objective 6: App will save results for 6 different classifications to keep data persistent only until the app closes

3. Project Context Diagram


sd System Context Diagram for AutoLearn application
Firestore
noSQL database
Firebase Auto ML
Android Interface for Model
Firebase
Authenticates logins
Phone display
Display application to user
Android 7.0 or higher
provides interface
On-board camera
allows user to take photo
AutoLearn application
Register , Request classification
Manage account, view stats
User
allows user to choose photo
produced and displayed to users
Photo Gallery
generates and stores for stats
contains ML model
Classifications
Data
Train, Test, and creates Model
Google Colab / Jupyter Notebook
Tensorflow Lite

4. Systems Requirements

4.1 “Login” Requirements

Requirement Title:	Sign Up
Sequence No:	01
Short description:	Allow a new user to sign up for an account.
Description:	The “Sign Up” button will be located near the bottom of the login screen. Clicking the button will take the user to an Account Creation page. The user will be prompted for the following information: - Email Address - UTA ID - Password - Confirm Password - Student/Professor
Pre-Conditions:	The user has successfully downloaded the Auto-Learn app from the Google Play store and has launched the app.
Post Conditions:	Logs in the user and navigates them to the home page. The system will automatically send the user a confirmation email to the email address they provided upon signing up.
Other attributes:	The login system will error check the password and confirm password fields to ensure that they match. The system will also check to make sure the email address entered is a valid email address.

Requirement Title:	Login
Sequence No:	02
Short description:	Allow a user to log into their account.
Description:	The login page will display an “Email” entry field and a “Password” entry field. The user can type in their email address and password they provided upon signing up to login to their account.
Pre-Conditions:	The user has successfully downloaded the Auto-Learn app from the Google Play store and has launched the app.
Post Conditions:	Logs in the user and navigates them to the home page.
Other attributes:	The login system will check the email and password to make sure they match a corresponding profile in the database.

Requirement Title:	Forgot Password
Sequence No:	03
Short description:	Allow a user to request a password reset.
Description:	There will be a “Forgot Password?” button at the bottom of the login screen. Clicking this button will prompt the user for their email address. The system will then reset the password for the account associated with that address and send an email to the user with the new password.
Pre-Conditions:	The user has successfully downloaded the Auto-Learn app from the Google Play store and has launched the app. The user must already have an account with Auto-Learn.
Post Conditions:	The user will stay on the login screen and can enter their new password to gain access to their account.
Other attributes:	The system will send the confirmation email for the password reset to the user. The email will contain a link for a password reset.

Requirement Title:	AutoLearn Logo
Sequence No:	04
Short description:	Display the AutoLearn logo on the splash screen before the login screen shows.
Description:	The splash screen will display the name AutoLearn and the logo before displaying the login screen.
Pre-Conditions:	The user has successfully downloaded the AutoLearn app from the Google Play store and has launched the app.
Post Conditions:	N/A
Other attributes:	N/A

4.2 “Home Screen” Requirements

Requirement Title:	Home Screen
Sequence No:	01
Short description:	The home screen is the central hub for all the actions a user can take inside AutoLearn.
Description:	The home screen will display buttons that navigate to all of the subsections of the app. If the user does not have any classifications yet, the home screen will display “To get started upload a photo or take a photo” After completing one classification, the home screen will display a “View Results” button. The subsections shown as buttons are the following: - Settings (In the top left) - Statistics (In the top right) - Take Photo/Choose from Gallery (In the bottom right)
Pre-Conditions:	The user must successfully login to their user account.
Post Conditions:	N/A
Other attributes:	N/A

Requirement Title:	Statistics
Sequence No:	02
Short description:	Statistics button will take the user to a stats page
Description:	When the user first launches the app and creates a profile, there will be no statistics to show. If the user presses the statistics button at this point, there will be a message that pops-up for the user that asks them to classify something first. Once the user has classifications for the current session, they can click the Statistics button. Once clicked, an overlay will show all the percentages predicted for the current session in a bar graph. The percentages will be averaged out with how many classifications were performed.
Pre-Conditions:	The user must be logged in to their account and have performed at least one classification.
Post Conditions:	N/A
Other attributes:	N/A

4.3 “Menu” Requirements

Requirement Title:	Menu
Sequence No:	01
Short description:	The settings tab will show information about the app as well as some app-based settings options.
Description:	Pressing the settings button (3 horizontal lines) on the top left of the home screen will expand the settings menu. Within the settings menu there will be several options: - Account Settings - Model Information - Log out
Pre-Conditions:	The user must be logged in to their account and have pressed the menu button on the home screen.
Post Conditions:	The menu will expand out from the left side of the screen.
Other attributes:	N/A

Requirement Title:	Profile Picture
Sequence No:	02
Short description:	The user can change their profile picture by clicking the icon in the top left of the menu.
Description:	Pressing the profile picture icon will prompt the user with several options: - Take Photo - Choose from Gallery - Cancel The user may choose the appropriate option to update their profile picture to their liking.
Pre-Conditions:	The user must be logged in to their account and have pressed the menu button on the home screen.
Post Conditions:	The menu will expand out from the left side of the screen.
Other attributes:	N/A

Requirement Title:	Account Settings
Sequence No:	03
Short description:	Allow the user to change personal information related to their account.
Description:	Pressing the account settings button will take the user to a new screen that will allow them to change their email, change their password, or delete their account.
Pre-Conditions:	The user must be logged into their account. The user must have also pressed the menu button as well as the account settings button within that menu.
Post Conditions:	The user will be back at the home screen after updating their information.
Other attributes:	N/A

Requirement Title:	Model Information
Sequence No:	04
Short description:	The “Model Information” button will show info on the creation of the model used for classification.
Description:	Pressing the model information button will bring the user to an info screen. The information will include: - Average Precision - Recall Percentage - 6 classes of vehicles - Number of images used to train and test the model
Pre-Conditions:	The user must be logged into their account. They must also press the settings menu button as well as the model information button.
Post Conditions:	The user will be back on the home screen when done.
Other attributes:	N/A

Requirement Title:	Log Out
Sequence No:	05
Short description:	Press the log out button to log out of your account.
Description:	Pressing this button will log out of the users account and return them to the launch screen.
Pre-Conditions:	The user must be logged into their account. They must also press the menu button and hit “Log Out” within that menu.
Post Conditions:	The user will be navigated to the launch screen.
Other attributes:	N/A

4.4 “Account Settings” Requirements

Requirement Title:	Change Password
Sequence No:	01
Short description:	This function allows the user to change their password.
Description:	Due to the sensitive nature of a password, it will not be visible on the profile page. To change or update your password you must hit the “Change Password” button located on the account settings page. When pressed, the user will be navigated to a Change Password page. The page will ask you for your current password and to confirm that password. If both entry fields match each other as well as the database information and the “Enter” button is pressed, the password will be updated.
Pre-Conditions:	The user must be logged in to their account and have pressed the menu button on the home screen. They must also press the account settings page, then the “Change Password” button.
Post Conditions:	When the user has completed the prompts for changing password, they will be back on the home screen.
Other attributes:	

Requirement Title:	Change Email
Sequence No:	02
Short description:	This function allows the user to change their email address.
Description:	To change or update your email you must hit the “Change Email” button located on the Account Settings menu. When pressed, the user will be taken to a “Change Email” page. The user will then be prompted to enter their desired email address in the input box. Once completed, they may hit the “Verify Email” button on screen to send a verification email to their new email address.
Pre-Conditions:	The user must be logged in to their account and have pressed the menu button on the home screen. They must also press the account settings page, then the “Change Email” button.
Post Conditions:	When the user has completed the prompts, the will be back on the home screen.
Other attributes:	The system will send a verification email to the new email address to confirm that the email change has taken effect.

Requirement Title:	Delete Account
Sequence No:	03
Short description:	This function allows the user to delete their account.
Description:	Pressing the “Delete Account” button will prompt the user with “Are you sure?” There will be a “Cancel” and “DELETE” button on the bottom of the prompt. Pressing the cancel button will return them to the home screen. Pressing the delete button will return them to the login page.
Pre-Conditions:	The user must be logged in to their account and have pressed the menu button on the home screen. They must also press the account settings page, then the “Delete Account” button.
Post Conditions:	The user will be navigated back to the launch screen or the home screen depending on their decision.
Other attributes:	The users account information will be deleted.

4.5 "TAKE PHOTO" REQUIREMENTS

Requirement Title:	Take Photo
Sequence No:	01
Short description:	Pressing the "Take Photo" button will open the users camera to take a new photo for classification.
Description:	The "Take Photo" button will have a picture of a camera on it and will be located in the bottom left of the home screen. Pressing this button for the first time will ask the user for permission/access to the camera. If permission is given, the camera application will open. Once a photo is taken the classification process will start automatically.
Pre-Conditions:	The user must be logged in to their account.
Post Conditions:	The classification process starts immediately after taking a picture.
Other attributes:	Must prompt the user for permissions to access the camera.

4.6 “Choose from Gallery” Requirements

Requirement Title:	Choose from Gallery
Sequence No:	01
Short description:	The “Choose from Gallery” button will open the user’s gallery to select a photo to upload.
Description:	Pressing this button for the first time will ask the user for permission/access to the gallery. If permission is given, the user will be taken to the gallery. They will be able to navigate their folders to select one (1) image to submit for classification. Once they select an image, the classification will automatically begin.
Pre-Conditions:	The user must be logged in to their account.
Post Conditions:	The classification process starts immediately after uploading a photo.
Other attributes:	Must prompt the user for permissions to access the gallery.

4.7 “CLASSIFICATION” REQUIREMENTS

Requirement Title:	Classification
Sequence No:	01
Short description:	The classification process will use the machine learning model to determine the make/model of the vehicle in question.
Description:	Immediately after the user either takes a new photo with their camera or uploads a photo from their gallery, the classification process will begin. The photo will be input into the machine learning model. When the model is done with the classification process, the results will be shown to the user. The results will be shown as six percentages (one for each class) and the percentage of each that the model guesses for the corresponding picture.
Pre-Conditions:	The user must be logged in to their account and upload a photo via “Take Photo” or “Choose from Gallery.”
Post Conditions:	The user will end on the home screen.
Other attributes:	N/A

5. Software Processes and Infrastructure

5.1 HARDWARE AND INFRASTRUCTURE

Android 7.0 Nougat and higher

Android SDK

Java Development Kit

Android Studio 3.6

Android Emulator / Galaxy S7 edge / Galaxy S8 Active / Galaxy Tab A 8.0

Anaconda 2019.10 / Python 3.7

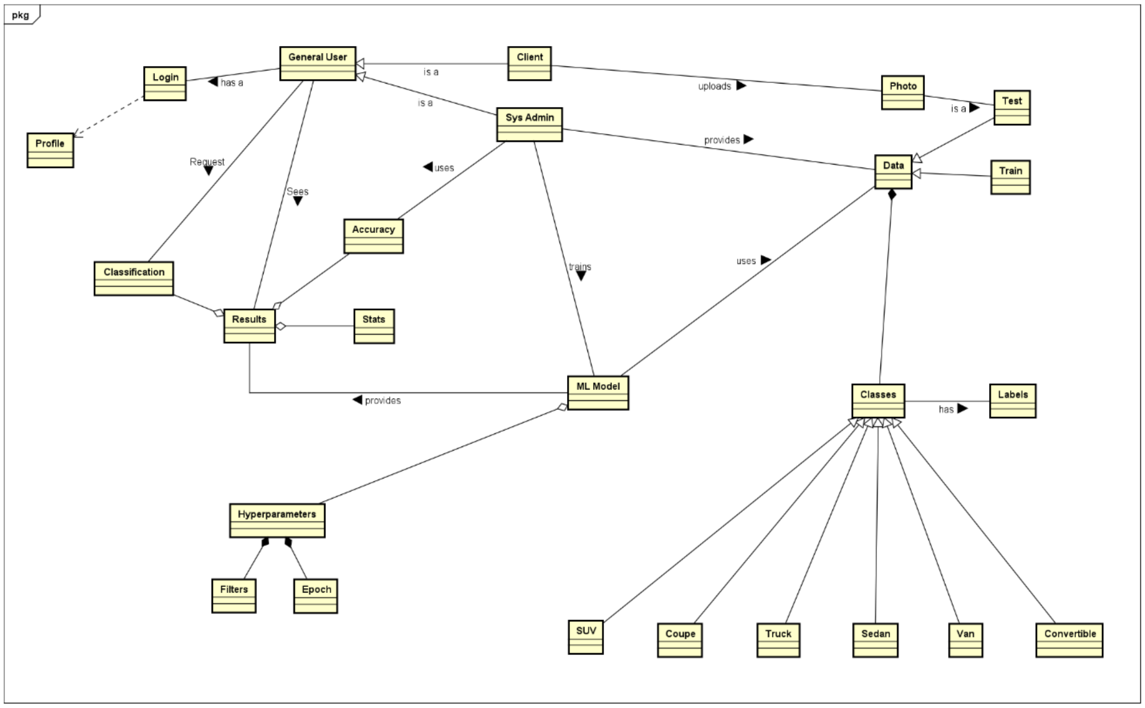
Firebase AutoML API

Jupyter Notebook/Google Colab

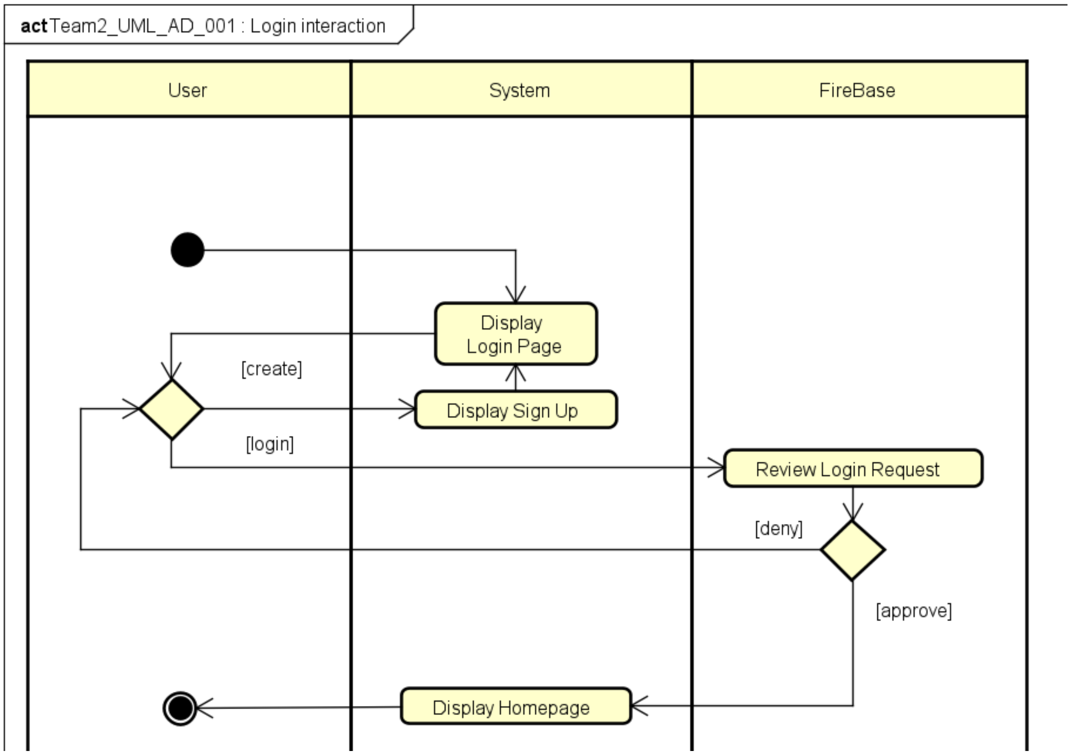
Tensorflow 2.0

Google Firebase

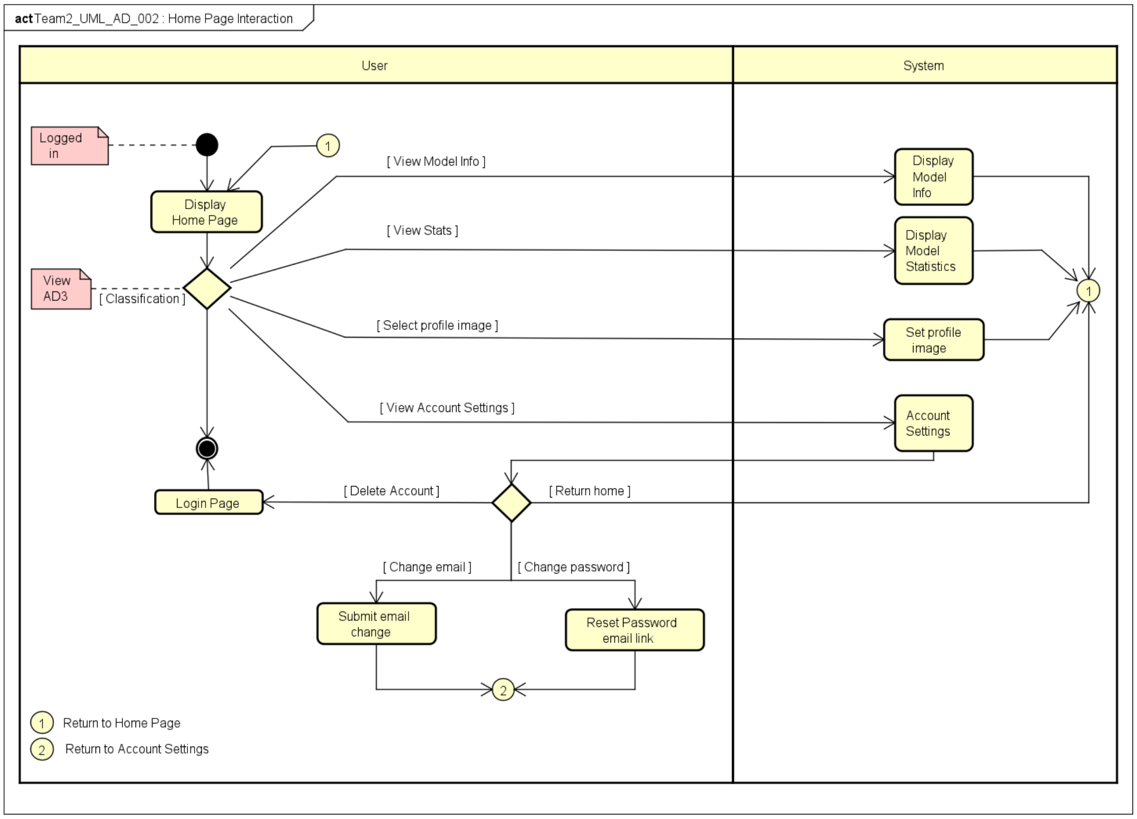
5.2 UML DIAGRAMS



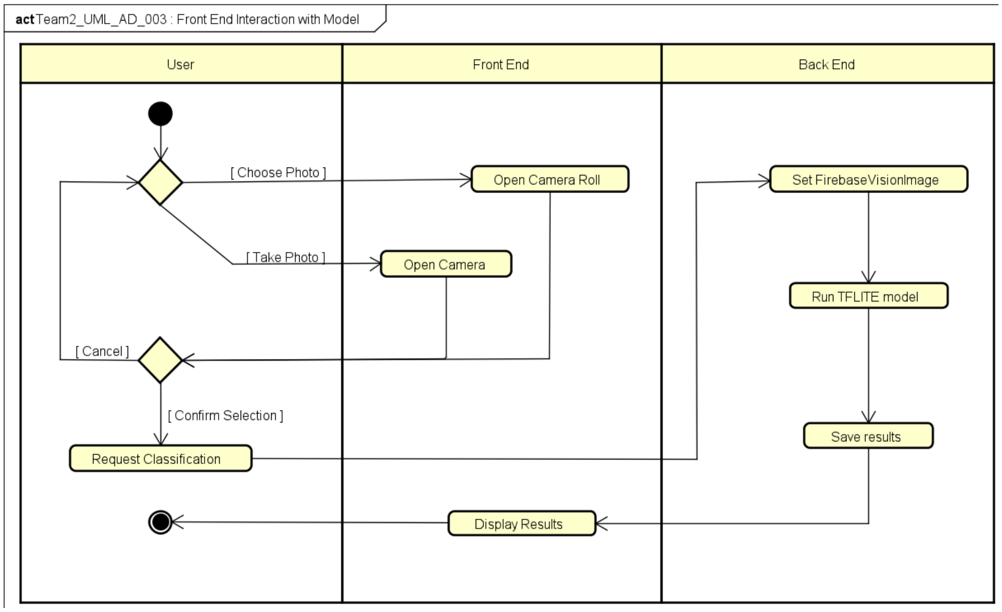
CLASS DIAGRAM



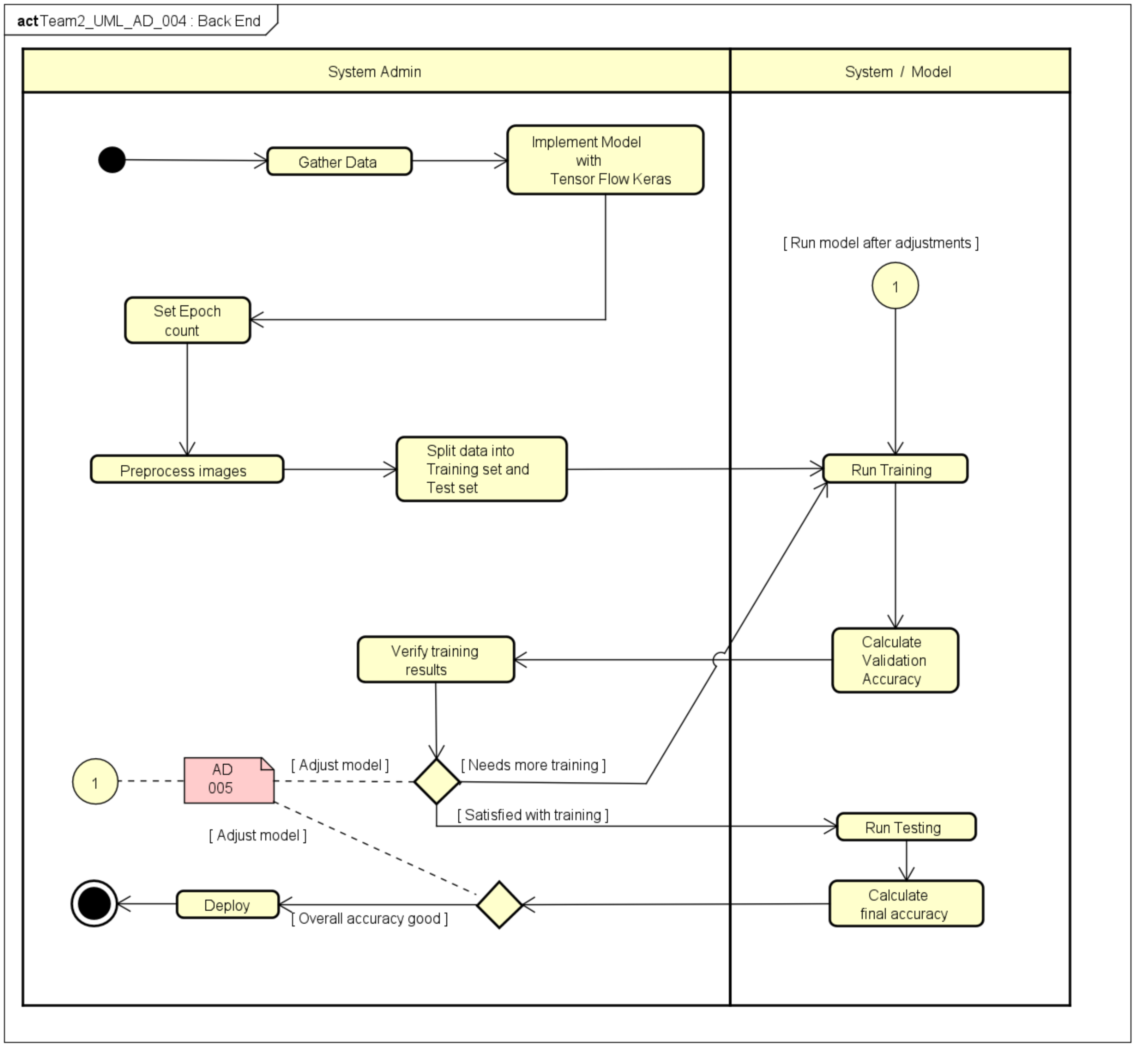
ACTIVITY DIAGRAM – Login Interaction



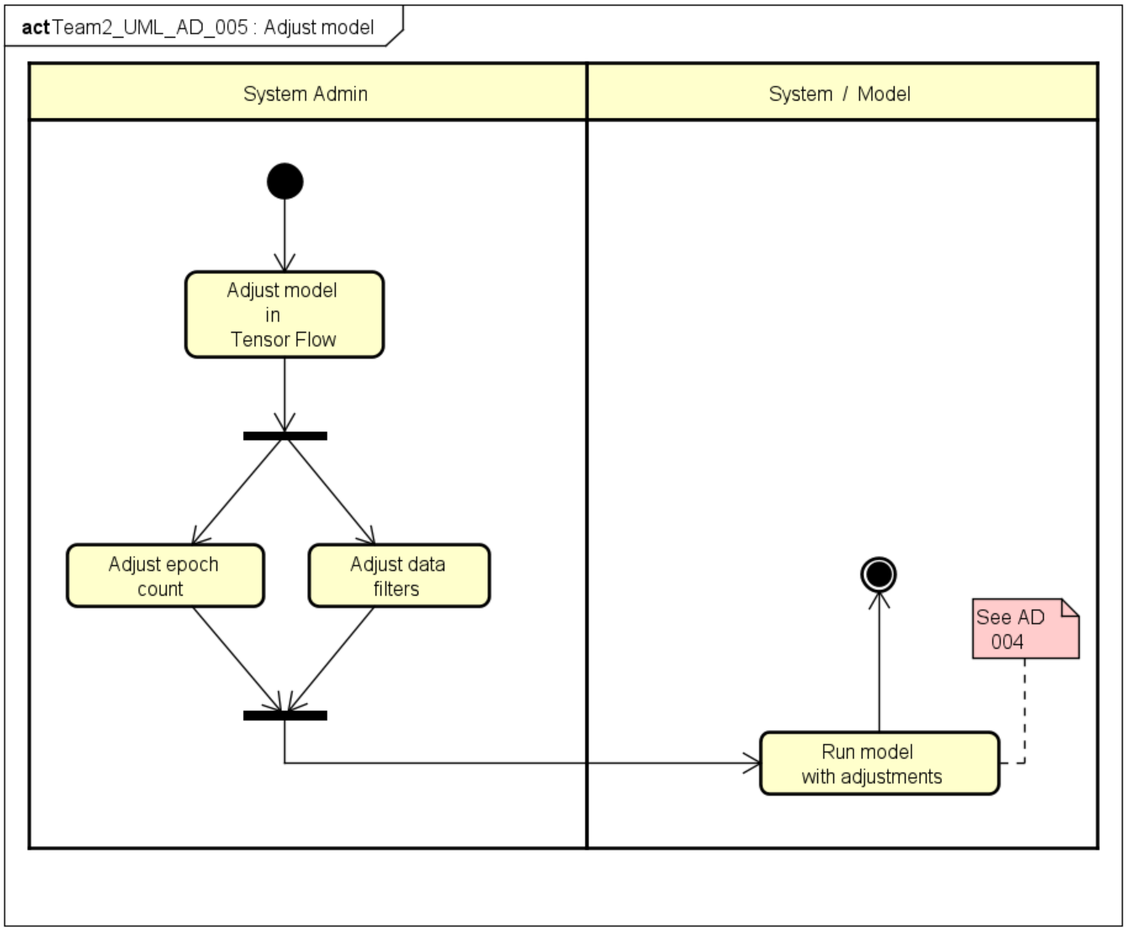
ACTIVITY DIAGRAM – Home Page Interaction



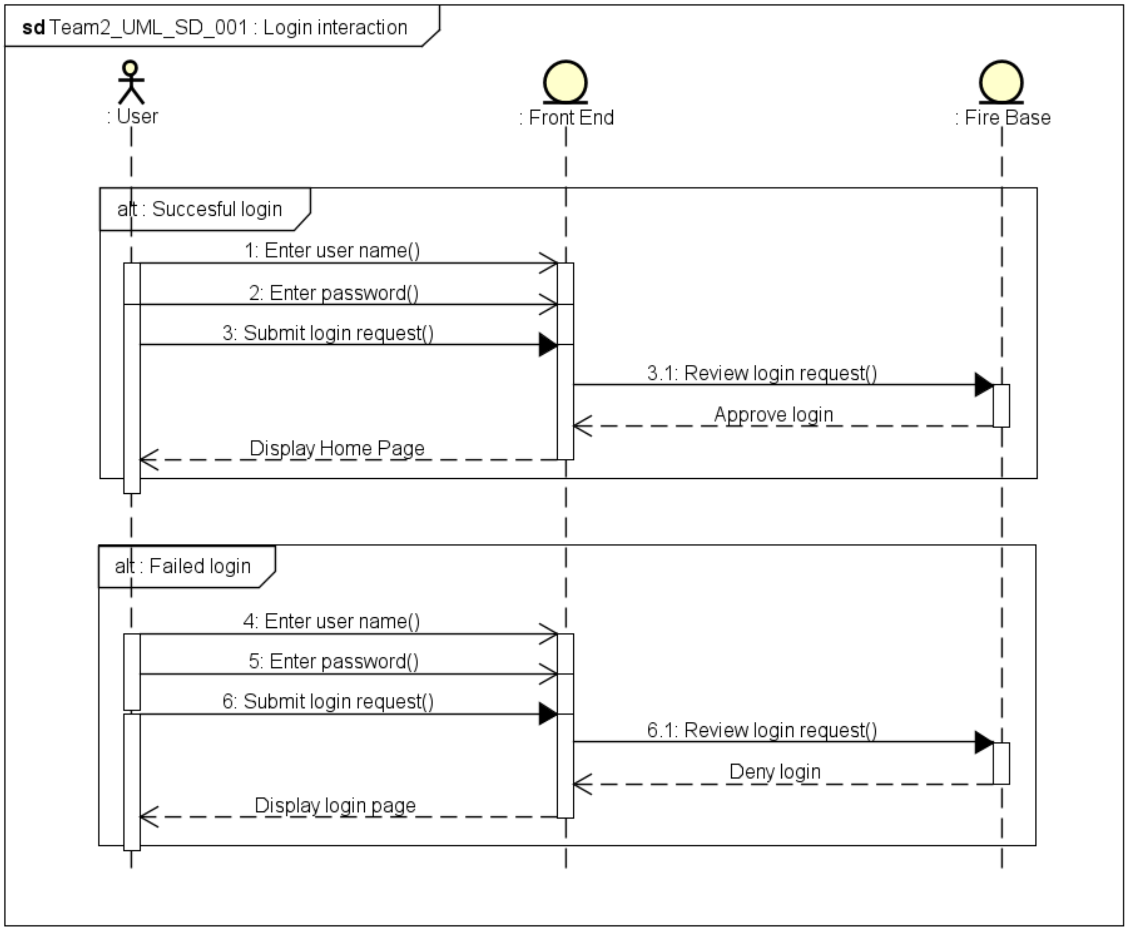
ACTIVITY DIAGRAM – Front End Interaction with Model


act Team2_UML_AD_004 : Back End
System Admin
System / Model
Gather Data
Implement Model with Tensor Flow Keras
Set Epoch count
Preprocess images
Split data into Training set and Test set
[Run model after adjustments]
1
Run Training
Calculate Validation Accuracy
Verify training results
1
AD 005
[Adjust model]
[Needs more training]
[Satisfied with training]
Run Testing
Calculate final accuracy
[Adjust model]
Deploy
[Overall accuracy good]

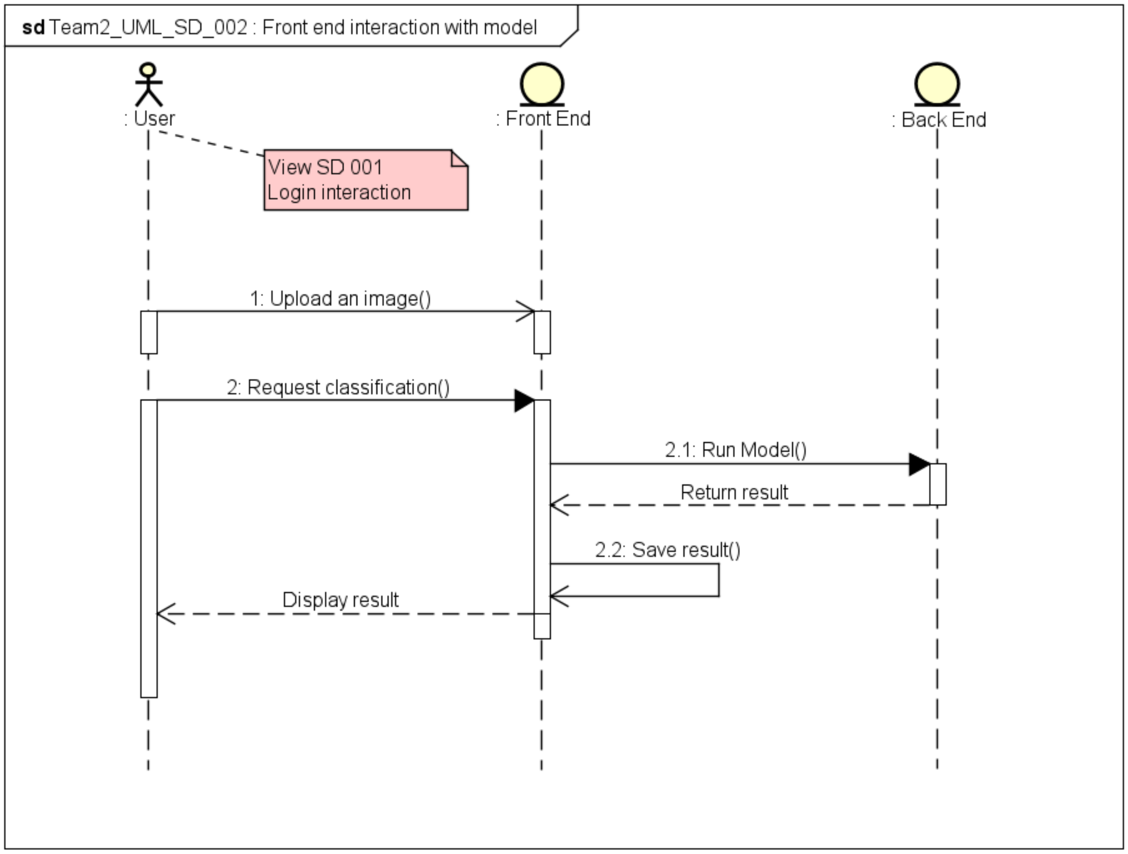
ACTIVITY DIAGRAM – Back End


act Team2_UML_AD_005 : Adjust model
System Admin
System / Model
Adjust model in Tensor Flow
Adjust epoch count
Adjust data filters
Run model with adjustments
See AD 004

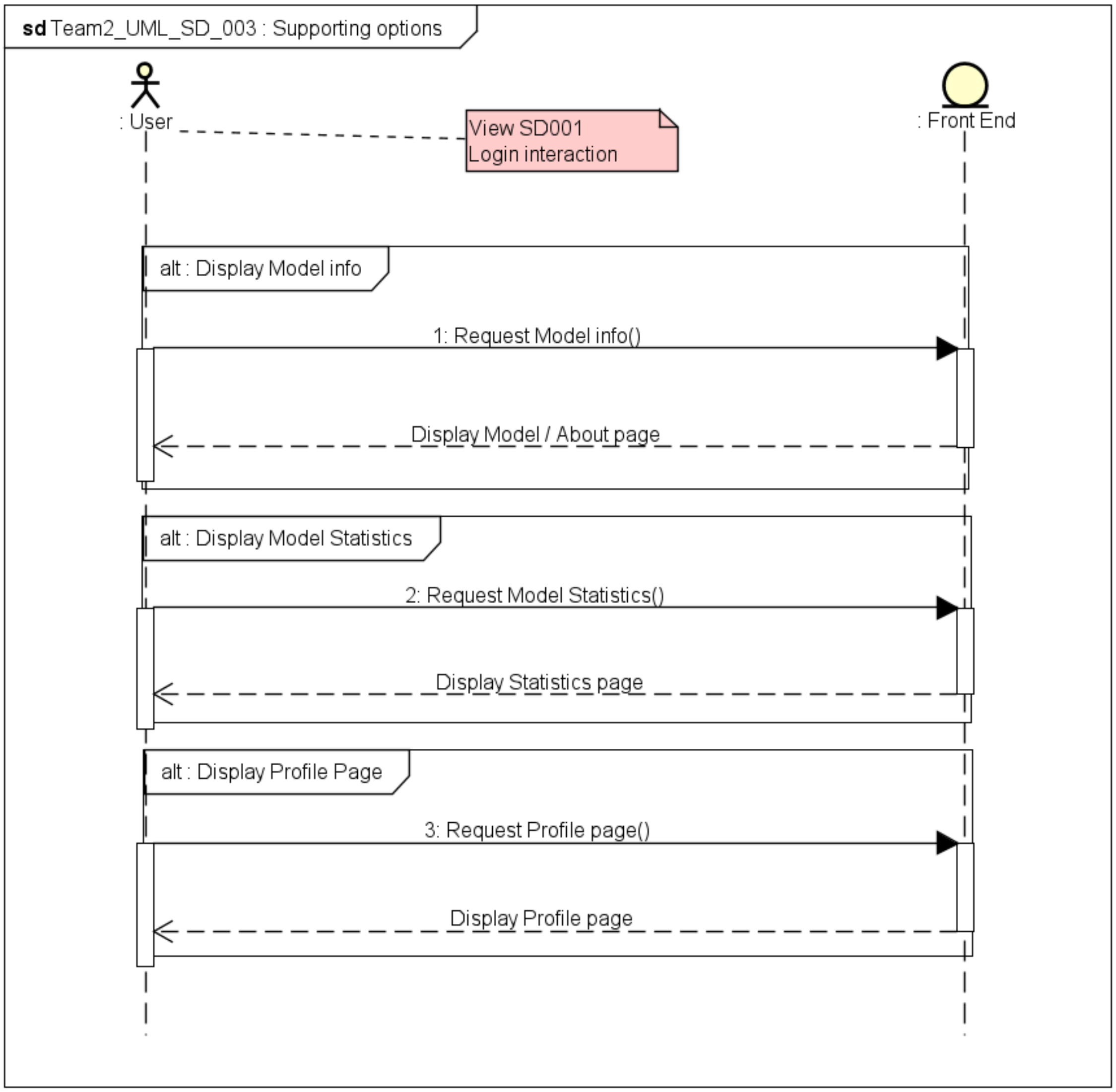
ACTIVITY DIAGRAM – Adjust Model



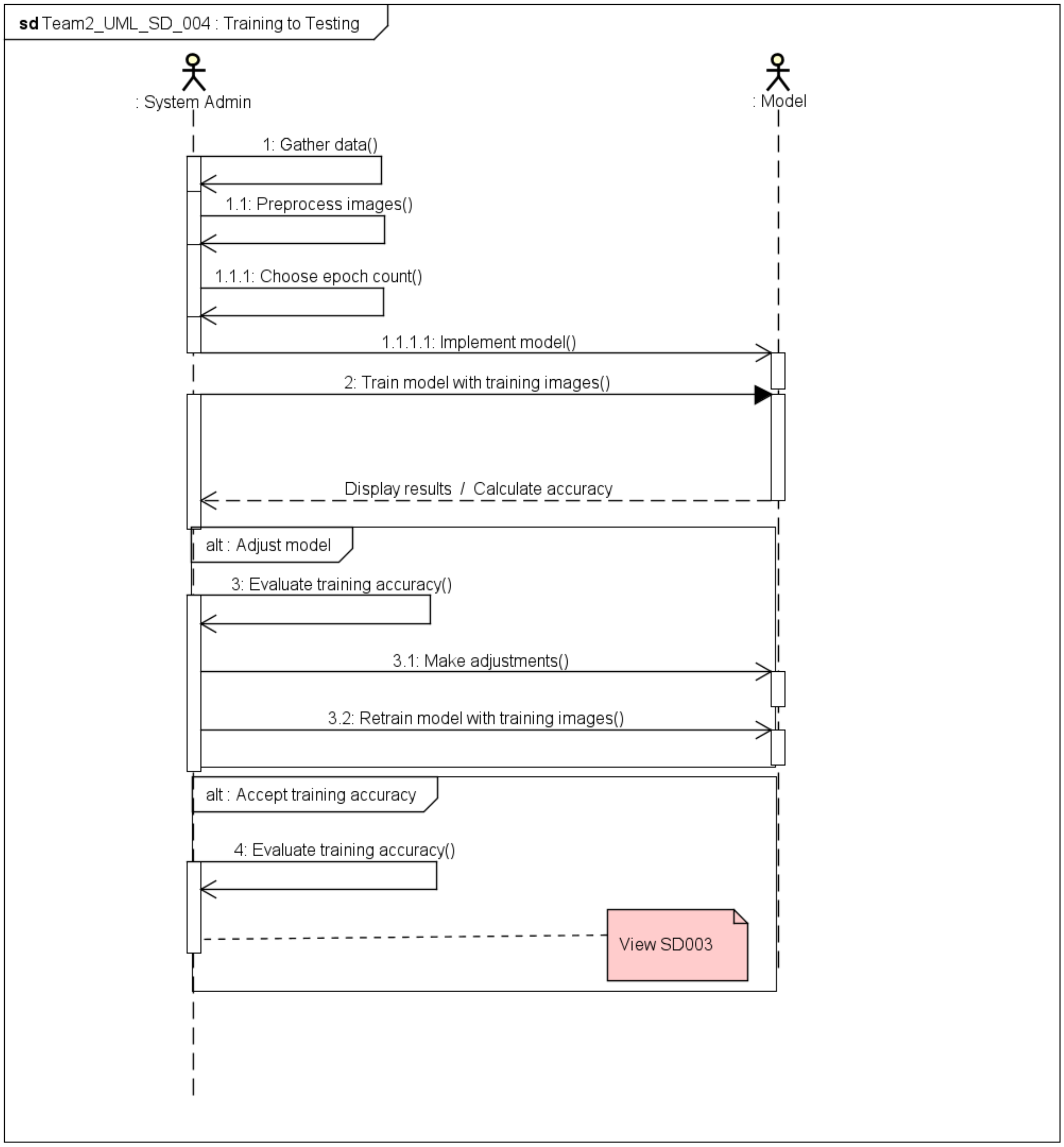
SEQUENCE DIAGRAM – Login Interaction



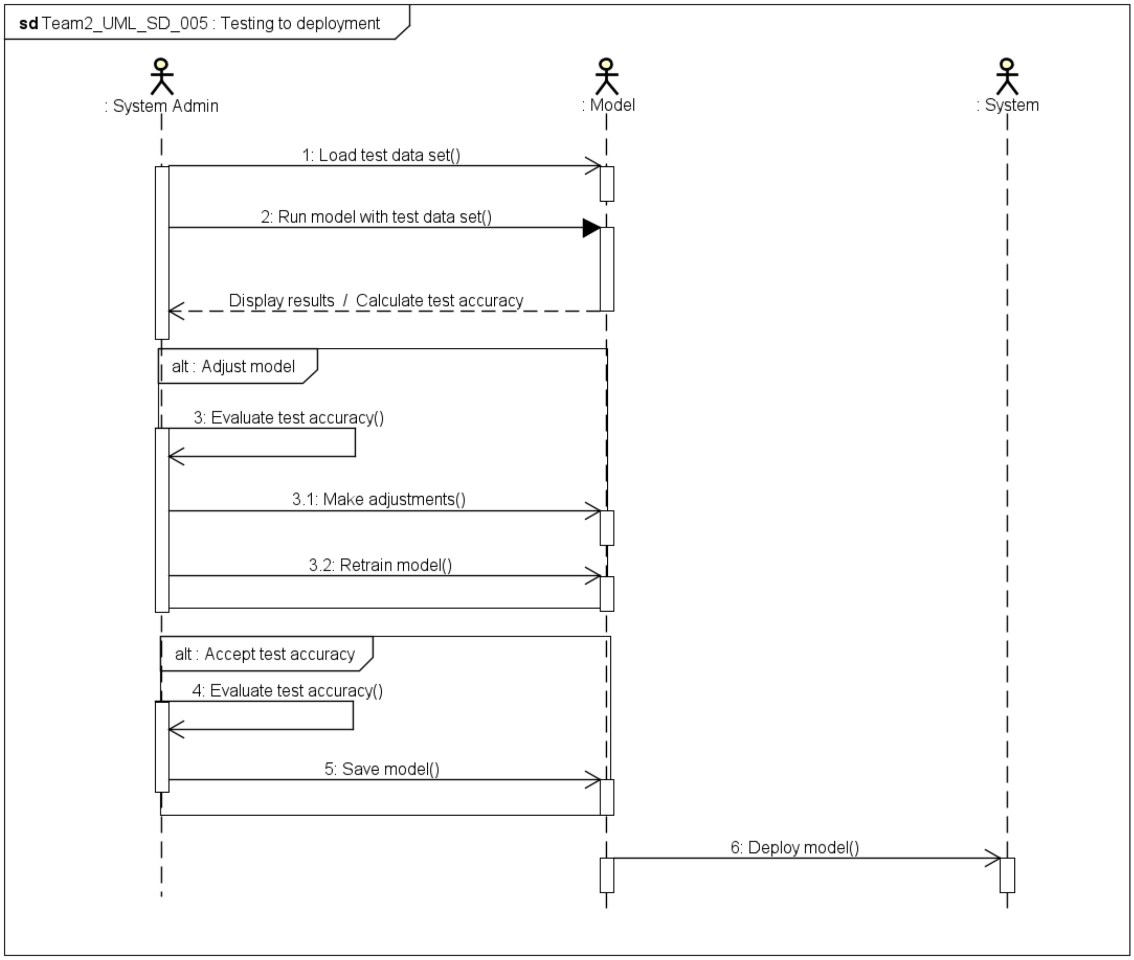
SEQUENCE DIAGRAM – Front end interaction with model


sd Team2_UML_SD_003 : Supporting options
: User
View SD001
Login interaction
: Front End
alt : Display Model info
1: Request Model info()
Display Model / About page
alt : Display Model Statistics
2: Request Model Statistics()
Display Statistics page
alt : Display Profile Page
3: Request Profile page()
Display Profile page

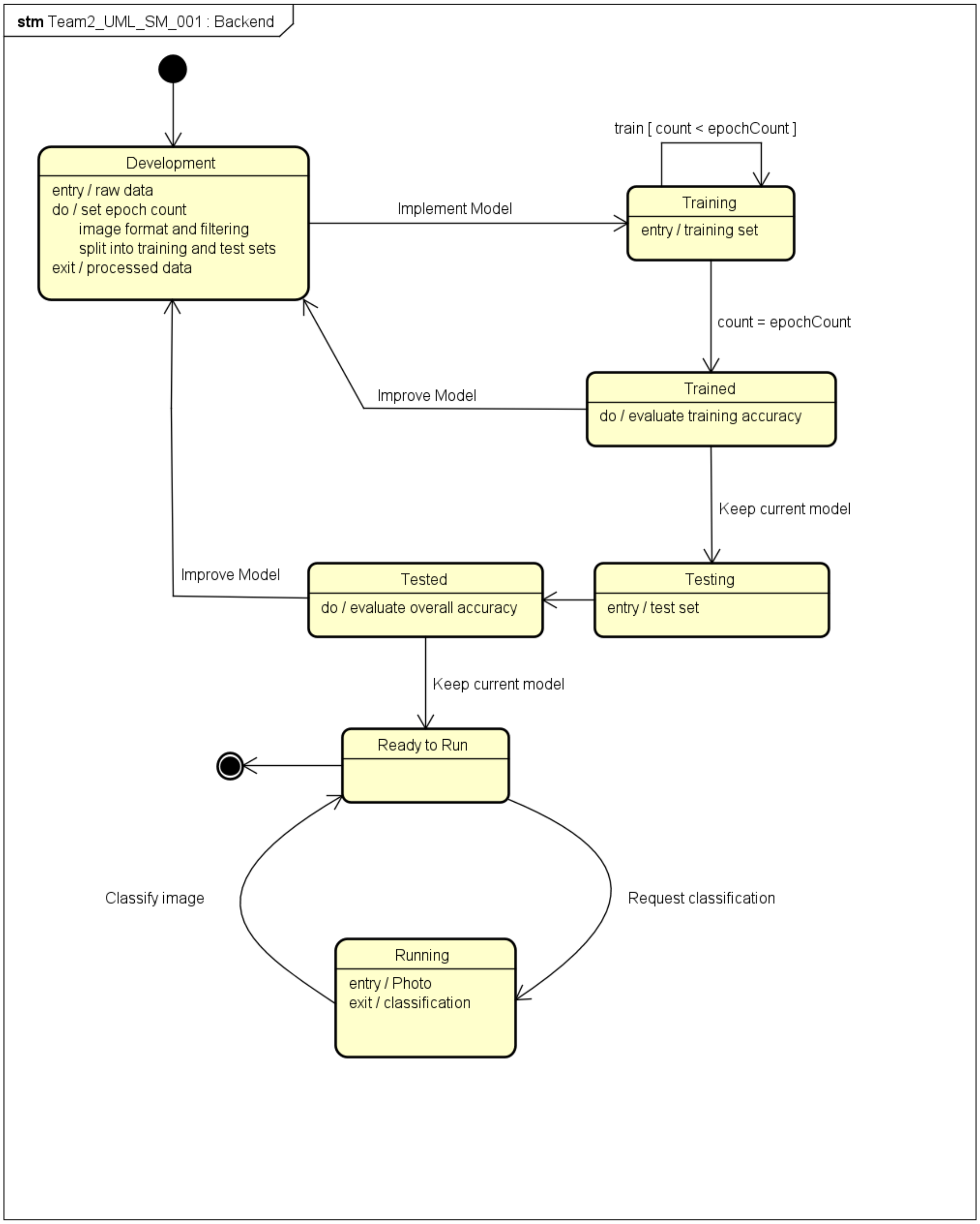
SEQUENCE DIAGRAM – Supporting options


sd Team2_UML_SD_004 : Training to Testing
: System Admin
: Model
1: Gather data()
1.1: Preprocess images()
1.1.1: Choose epoch count()
1.1.1.1: Implement model()
2: Train model with training images()
Display results / Calculate accuracy
alt : Adjust model
3: Evaluate training accuracy()
3.1: Make adjustments()
3.2: Retrain model with training images()
alt : Accept training accuracy
4: Evaluate training accuracy()
View SD003

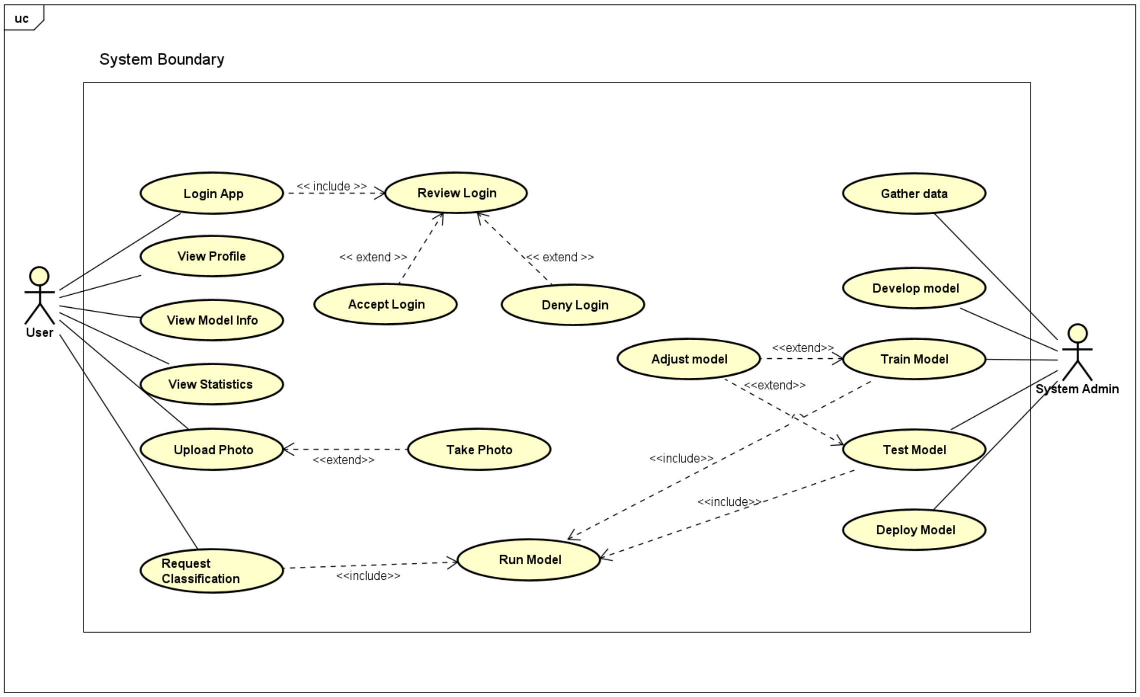
SEQUENCE DIAGRAM – Training to Testing


sd Team2_UML_SD_005 : Testing to deployment
: System Admin
: Model
: System
1: Load test data set()
2: Run model with test data set()
Display results / Calculate test accuracy
alt : Adjust model
3: Evaluate test accuracy()
3.1: Make adjustments()
3.2: Retrain model()
alt : Accept test accuracy
4: Evaluate test accuracy()
5: Save model()
6: Deploy model()

SEQUENCE DIAGRAM – Testing to deployment

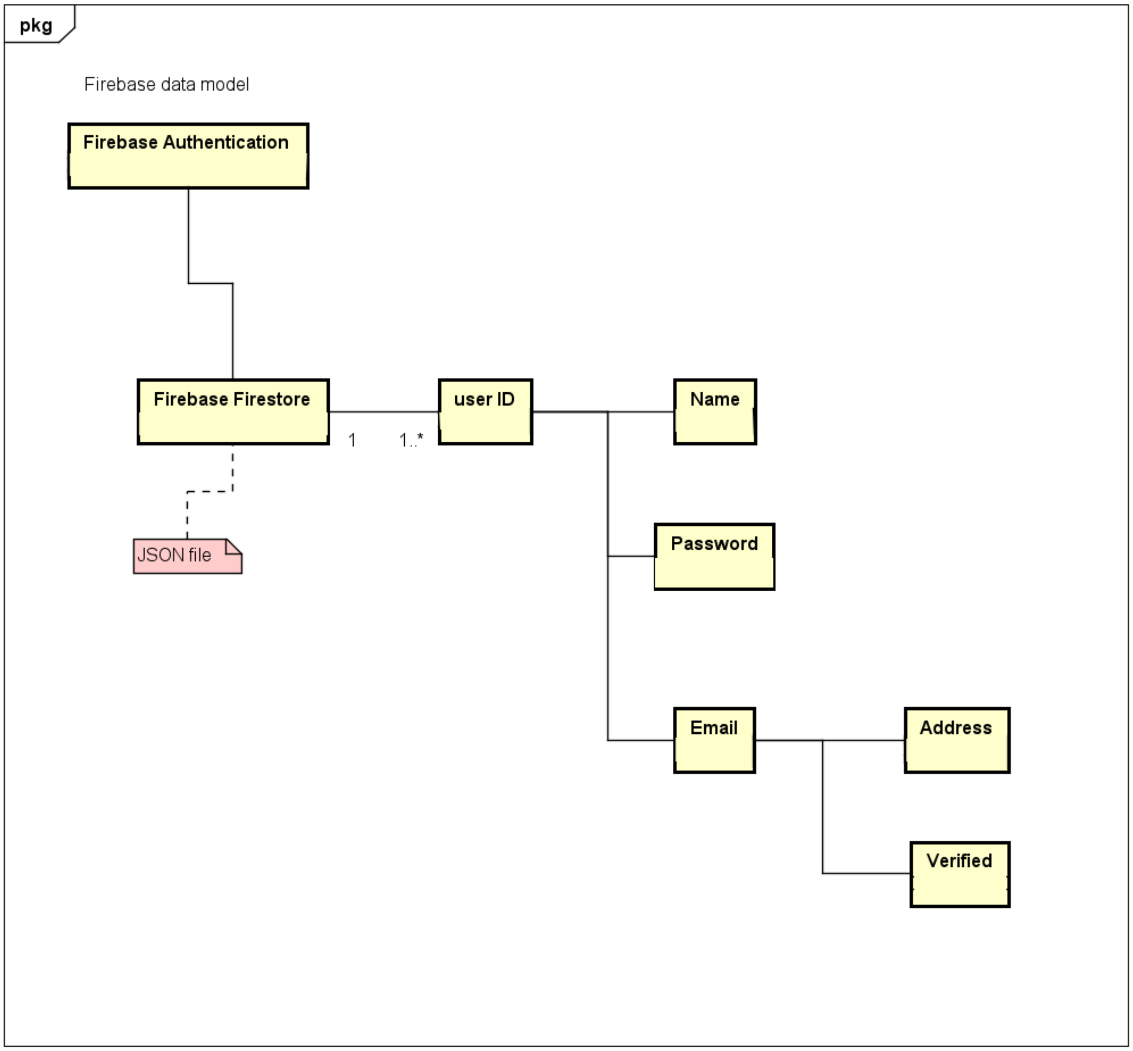


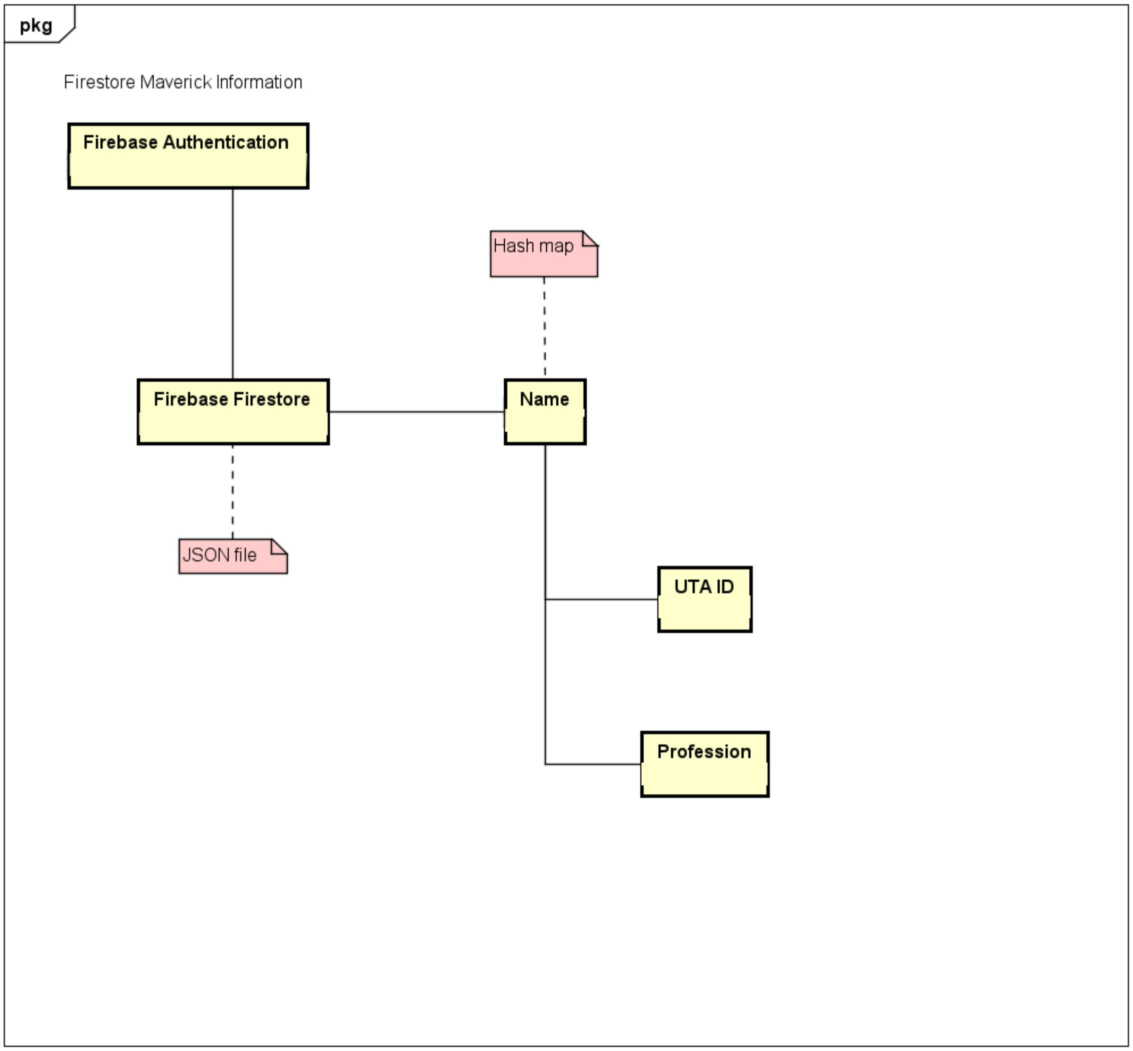
STATE MACHINE DIAGRAM – Back End



USE CASE DIAGRAM

5.3 Conceptual Data Model – Database




pkg
Firestore Maverick Information
Firebase Authentication
Hash map
Firebase Firestore
Name
JSON file
UTA ID
Profession

6. Test Plan

6.1 Introduction and Plan of Approach

Summary

Auto-Learn is an android application supporting version 7.0 (Nougat) of Android. The app should be able to allow users to upload pictures, and classify the picture if it has a vehicle or not. The app should use a machine learning model to classify the vehicles. The picture should be classified as one of the following categories: Coupe, SUV, Sedan, Convertible, Van, Truck. After the user has taken a picture or chosen one to classify the model should classify the picture and display the results to the user. The user can then choose to make another classification by uploading another picture, or the user can choose to view the statistics of the classification they just performed. The user should also be able to customize their account by choosing another profile picture, changing their password, changing their college/academic information, and the changes should be saved to a database. Previous classification percentages should be seen in the user's account statistics.

In general here are the functions users should be able to do with Auto-Learn in no specific order:

- Upload a picture
- Create an account
- Classify a picture with a vehicle
- View statistics of the previous classification performed
- View session classification averages
- Customize their account
- Change profile picture, personal information, and password

Assumptions

- User's phone has wifi connection
- User's phone has a camera
- User's phone has a touch screen
- User's phone has sufficient system requirements
- User's phone has the accurate operating system

Components covered

1. Login
2. Sign up
3. Home Screen
4. Menu
5. Account Settings
6. Take Photo
7. Upload Photo
8. Classification

6.2 TEST CASES: "LOGIN"

Project Name: Auto-Learn
Test Case Name: Login
Test Case Id: CSE3310/Spring 2020/Team2/ Login

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1 - Sign Up	Click the "Sign up" text in the bottom right of the login page.	The user will be navigated to the sign up page.	**Pass:** New activity is presented
TC2 - Login	Tab into the email and password fields and enter a valid user ID/password.	System should allow the user to login and navigate to the home screen.	**Pass:** Firebase successfully retrieves the account information.
TC3 - Invalid Login	Tab into the email and password fields and enter an invalid username and password.	System should not accept the invalid credentials and prevent you from entering the system.	**Pass:** Firebase successfully checks for valid login information.
TC4 - Forgot Password	Enter a valid username and press "Forgot Password".	System should prompt you with a security question and send a temporary password to your email.	**Pass:** The app will send a password reset link to your email
TC6 - Logo	Auto-Learn logo.	The Auto-Learn logo should display on the splash page.	**Pass**

6.3 TEST CASES: “SIGN UP”

Project Name:	Auto-Learn
Test Case Name:	Sign up
Test Case Id:	CSE3310/Spring 2020/Team2/Sign_Up

Test Case No.	**Test Case Description**	**Expected results**	**Outcome** **Pass, Fail, Other (comments)**
TC1 - Enter Profile Info	Enter the below information and hit “Next” ● Name ● Profile picture (optional) ● Email - Valid email address. ● Password ● Confirm password	Application will create a firebase account and navigate to Maverick Information If the user provided a picture, store it in the newly created firebase user.	**Pass:** Firebase successfully stores the new account information.
TC2 - Maverick Information	From the Maverick Information page select a profession: (Student/Professor) and enter UTA ID, then press next	System will store additional information in the Firebase firestore no-SQL database. Application navigates to Email verification	**Pass:** Firebase successfully retrieves the account information.
TC3 - Email verification	From email verification page press Send Email button	Users will receive an email verification request. Application will log the user in and display the Home Page.	**Pass:** Firebase successfully sends an email request

6.4 TEST CASES: "HOME SCREEN"

Project Name: Auto-Learn
Test Case Name: Home Screen
Test Case Id: CSE3310/Spring 2020/Team2/ Home_Screen

Test Case No.	**Test Case Description**	**Expected results**	**Outcome** **Pass, Fail, Other (comments)**
TC1 - Menu	Click the "Menu" button in the top left of the home screen.	Navigation view with additional menu items, and profile information displayed opens from the left side of the screen.	**Pass**
TC2 - Model Statistics	Click the "Graph" button in the top right of the home screen.	Model Statistics popup will display with previous classifications.	**Pass**
TC4 - Select an image	Click the "Gallery" button in the bottom right of the home screen.	Pop up menu will display with options: ● Take photo ● Choose from Gallery ● Cancel	**Pass**
TC5 - View Results	With an image loaded into the ImageView by the user, click the View Results button	Machine learning model will be run on image and results will be displayed in Results Dialog	**Pass**

6.5 Test Cases: "Menu"

Project Name:	Auto-Learn
Test Case Name:	Menu
Test Case Id:	CSE3310/Spring 2020/Team2/Menu

Test Case No.	**Test Case Description**	**Expected results**	**Outcome** **Pass, Fail, Other (comments)**
TC1 - Back Home	From menu press the right side of the screen or swipe menu away	Settings menu will disappear and Home page will be displayed	**Pass**
TC2 - Account Settings	From menu click the Account Settings button	Application navigates to the Account Settings page where user may: ● Change password ● Change email ● Delete account	**Pass**
TC3 - Profile Info displayed	From menu, navigation view header will display information: ● Name ● UTA ID ● Profession ● Profile Picture	Application extracts information from Google Firebase	**Pass**
TC4 - Change Profile Picture	Click the Profile Picture Bubble located in the navigation view header.	Pop up menu will display with options: ● Take photo ● Choose from Gallery ● Cancel Upon completion firebase user profile picture will be updated	**Pass**
TC5 - Model Info	From settings menu press "Model Information"	Application displays an Information Dialog where data gathered by Google Firebase ML Kit will be displayed along with links to data sets.	**Pass**

TC6 - Logout	From settings menu press “Logout”	User is logged out and application navigates to “Login Page”	**Pass**

6.6 TEST CASES: “ACCOUNT SETTINGS”

Project Name: Auto-Learn
Test Case Name: Account Settings
Test Case Id: CSE3310/Spring 2020/Team2/Account_Settings

Test Case No.	**Test Case Description**	**Expected results**	**Outcome** **Pass, Fail, Other (comments)**
TC1 - Back Home	Click the back arrow in the top left of the “Profile” page.	The user will be navigated back to the home screen.	**Pass**
TC2 - Change Email	From Account settings menu click the “Change Email” button	Change email page will be displayed where user can enter the new email	**Pass**
TC3 - Change Password	From Account settings menu, click the “Change Password” button	Change password page will be displayed where the user enters a new password and confirms the new password.	**Pass**
TC4 - Delete Account	From Account settings menu click the “Delete Account” button	Pop up prompt will ask user to confirm password, before deleting the account	**Pass**

6.7 TEST CASES: "TAKE PHOTO"

Project Name: Auto-Learn
Test Case Name: Take Photo
Test Case Id: CSE3310/Spring 2020/Team2/Take_Photo

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1 - Permission	First time users will be prompted to grant the application permission/access to the device's native camera	Permissions pop up should appear with two options: Allow / Deny	**Pass**
TC2 - Deny Permission	From permissions pop up user clicks "Deny"	Camera will not be accessed and notification will be displayed to inform user to go to App Settings to allow permission	**Pass**
TC3 - Allow Permission	From permissions pop up user clicks "Allow"	The device's camera will be accessed	**Pass**
TC4 - Take Photo	From the Camera display press take a photo	Once the photo is taken the classification process will start automatically and results will be displayed in Results Dialog	**Pass**
TC6 - Back Home	From the Camera display press the back arrow	Application will navigate to "Home Page	**Pass**

6.8 Test Cases: “Choose from Gallery”

Project Name: Auto-Learn
Test Case Name: Upload Photo
Test Case Id: CSE3310/Spring 2020/Team2/Upload_Photo

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1 - Permission	First time users will be prompted to grant the application permission/access to the device’s photo gallery	Permissions pop up should appear with two options: Allow / Deny	**Pass**
TC2 - Deny Permission	From permissions pop up user clicks “Deny”	Photo gallery will not be accessed and notification will be displayed to inform user to go to App Settings to allow permission	**Pass**
TC3 - Allow Permission	From permissions pop up user clicks “Allow”	The device’s photo gallery will be accessed, with a greyed out submit button	**Pass**
TC4 - Select Image	Select one (1) image	After selecting an image, run the model and display Results Dialog	**Pass**
TC6 - Back Home	From the photo gallery press the back arrow	Application will navigate to “Home Page”	**Pass**

6.9 TEST CASES: “CLASSIFICATION”

Project Name: Auto-Learn
Test Case Name: Classification
Test Case Id: CSE3310/Spring 2020/Team2/Classification

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1 - SUV	The user takes/uploads a photo of an SUV (Sports Utility Vehicle). [Ex: Toyota RAV4]	The classification will return a set of percentages with the largest percentage in the SUV category.	**Pass**
TC2 - Sedan	The user takes/uploads a photo of a Sedan. [Ex: Toyota Camry]	The classification will return a set of percentages with the largest percentage in the Sedan category.	**Pass**
TC3 - Truck	The user takes/uploads a photo of a Pickup Truck. [Ex: Chevrolet Silverado]	The classification will return a set of percentages with the largest percentage in the Truck category.	**Pass**
TC4 - Convertible	The user takes/uploads a photo of a Convertible [Ex: Mercedes SL]	The classification will return a set of percentages with the largest percentage in the Convertible category.	**Pass**
TC5 - Coupe	The user takes/uploads a photo of a Coupe. [Ex: Porsche 911]	The classification will return a set of percentages with the largest percentage in the Coupe category.	**Pass**
TC6 - Van	The user takes/uploads a photo of a Van. [Ex: Honda Odyssey]	The classification will return a set of percentages with the largest percentage in the Van category.	**Pass**
TC7 - Other	The user takes/uploads a photo of a vehicle that doesn’t fall into any of these categories.	The classification will return a message with the category that it determines is a best fit for the uploaded photo.	**Pass**

7. Assumptions and Constraints

7.1 ASSUMPTIONS

The following is a list of assumptions:

- Ignore collecting money from external advertisement and general accounting
- Ignore compliance issues
- Ignore market conditions and demands
- Ignore future system additions

7.2 CONSTRAINTS

The following is a list of constraints:

- Team lacks Android experience
- Schedule very aggressive
- Team lacks TensorFlow experience
- Team lacks Python experience
- Team has very limited Java experience

7.3 OUT OF SCOPE MATERIAL

The following is a list of “out of scope” material:

- Post Project maintenance is not covered

8. Delivery and Schedule

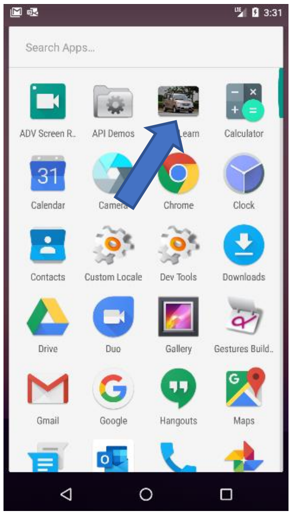
Task/Milestone Description	Anticipated Start Date	Anticipated End Date	Status	Comments
Prepare Requirements and UML diagram	01/31/2020	02/21/2020	Complete	Deliverable will be UL document. Increment 1 Deliverable
SRA document (Includes project objectives, Requirements and UML diagrams)	03/05/2020	03/31/2020	Complete	Deliverable will be the SRA document. All stakeholders agree on the content of the SRA by signing in section 8. Increment 2 Deliverable
Home screen design and implementation	3/31/2020	04/02/2020	Complete	
Login and registration design and implementation	03/31/2020	04/02/2020	Complete	
Set up Jupyter notebook for ML model	03/31/2020	04/02/2020	Complete	
Import all necessary imports into Jupyter	03/31/2020	04/02/2020	Complete	
Test case design	03/31/2020	04/16/2020	Complete	Increment 3 Deliverable
External Documentation (i.e. User Manual)	04/20/2020	04/20/2020	Complete	
Project presentation	04/28/2020	04/28/2020	Complete	
Final Milestone: project delivery		4/30/2020	Complete	Increment 4 Deliverable

9. Stakeholder Approval Form

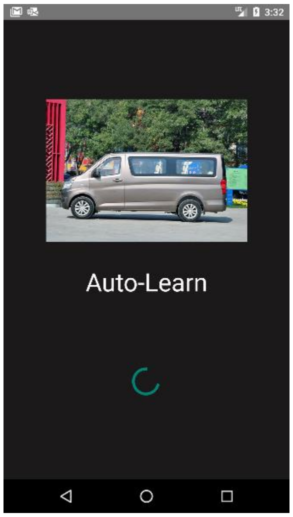
Stakeholder Name	Stakeholder Role	Stakeholder Comments	Stakeholder Approval Signature and Date
Rodrigo Augusto	Development Mgr.		
Edrik Aguilera	Developer		Edrik Aguilera 4/30/2020
William Anderson	Developer		4/30/2020
Ryan Laurents	Developer		Ryan Laurents 4/30/2020

Account Setup and Login

Press “Auto-Learn” in your Android application menu after installing the application.

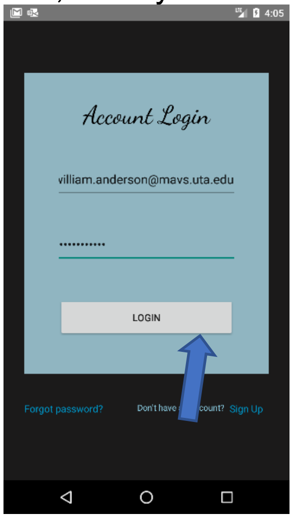


You are greeted by the Splash Screen for 2 seconds followed by the Login screen.



Login

If you already have an account, enter your account details and press “Login”.

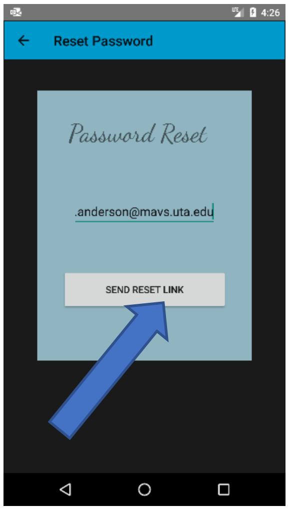


villiam.anderson@mavs.uta.edu

••••••••••

Forgot Password

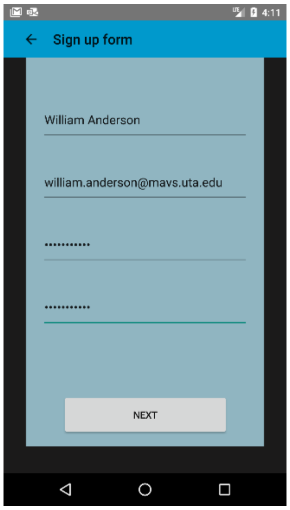
If you have forgotten your password, click on “Forgot Password” and enter your email to reset your password. You will receive a link to change your password in your email.



.anderson@mavs.uta.edu

Account Setup and Email Verification

If you do not have an account, click on the “Sign Up” button and enter your name, email, password, role and UTA ID. Click “Send Verification Email”. You will be logged in regardless of whether you verified your email or not. You will also receive a link in your email to verify it.

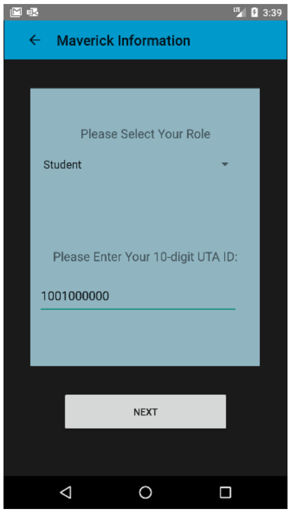


William Anderson

william.anderson@mavs.uta.edu

•••••••••••

•••••••••••

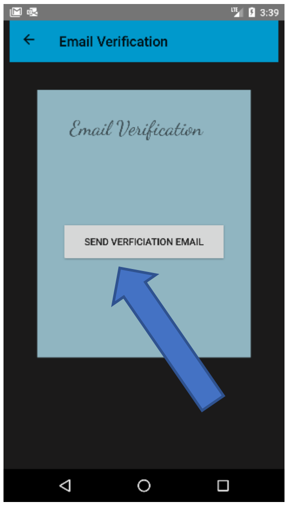


Please Select Your Role

Student

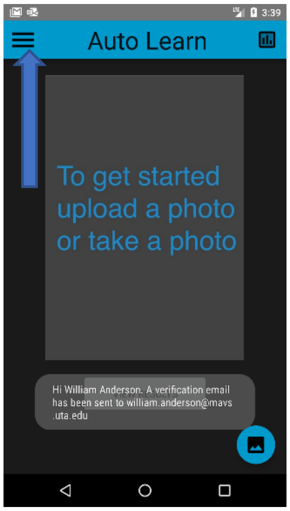
Please Enter Your 10-digit UTA ID:

1001000000



Account Management

Click on the menu bar to access your profile information.

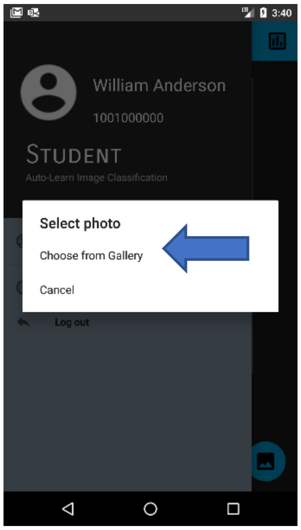


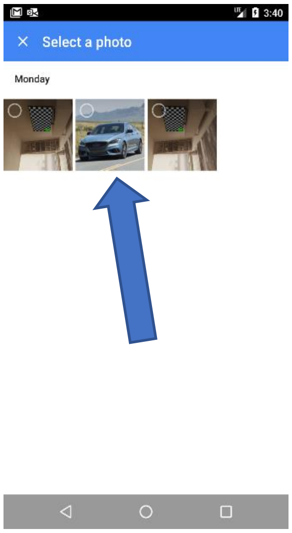
To get started upload a photo or take a photo

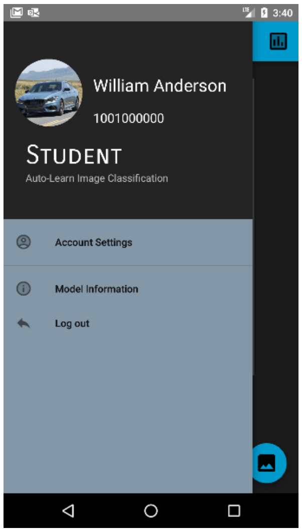
Hi William Anderson. A verification email has been sent to william.anderson@mavs .uta.edu

Setting a Profile Photo (optional)

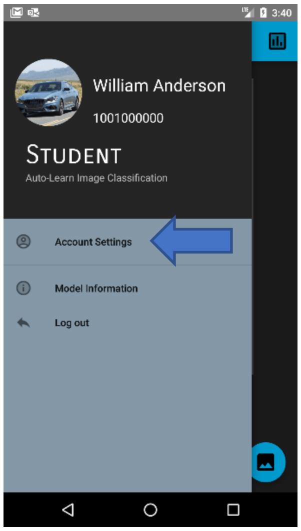
You can optionally upload a profile photo by clicking on the profile icon and click “Choose from Gallery”. Once a photo is selected, your profile photo will appear. To change your profile photo, the process is the same.

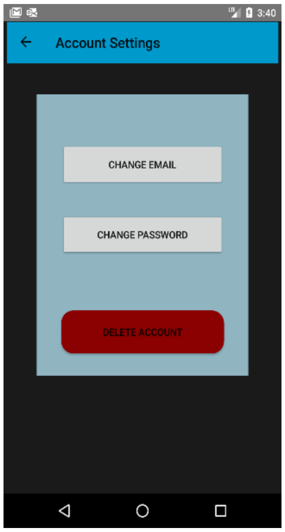






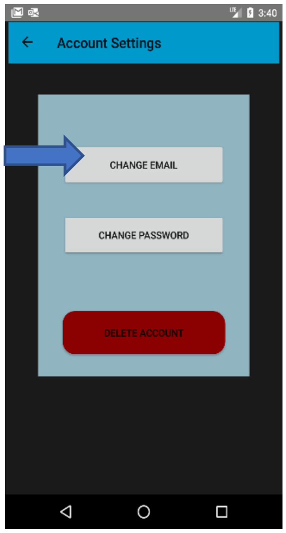
7. Clicking on “Account Settings” will give you options to change email, password, or delete account.

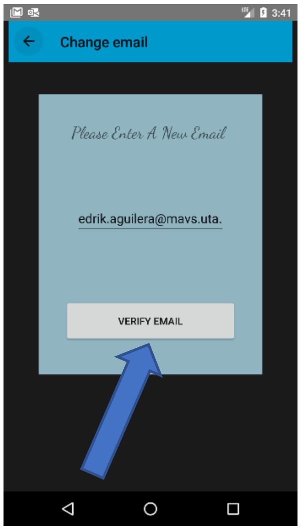




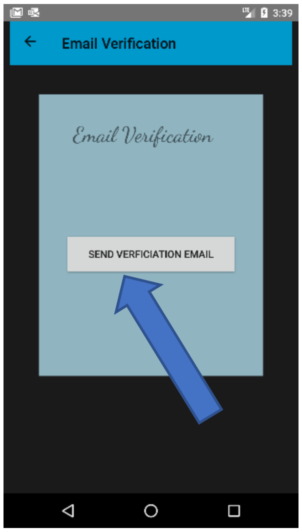
Change Email

Clicking on “Change Email” will show a entry where you can enter a new email that is not linked to an existing account. Upon completion, you will be redirected to verify your email. Upon completion, you will be taken back to the home screen.



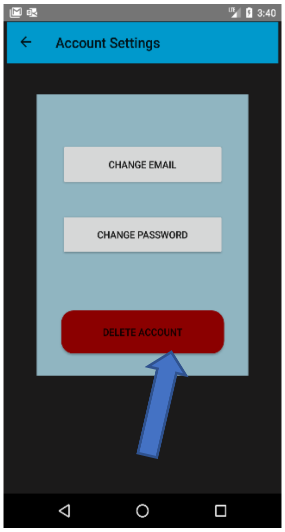


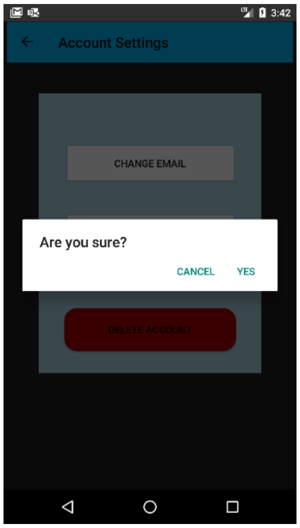
edrik.aguilera@mavs.uta.



Delete Account

Clicking on “Delete Account” will generate a pop-up dialog to confirm the deletion request. If the request is confirmed, you will be taken back to the login screen. If denied, you will be taken back to “Account Settings”

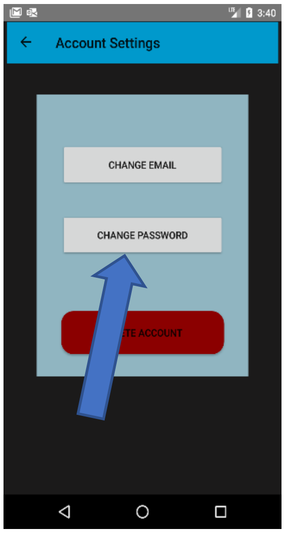


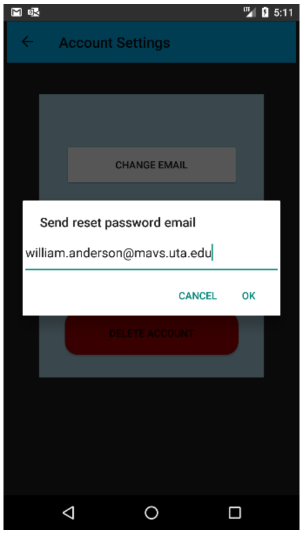


Are you sure?

Change Password

Clicking on “Change Password” will generate a pop-up dialog to enter your email. If the request is confirmed, you will be sent a link in your email to change your password. If denied, you will be taken back to “Account Settings”



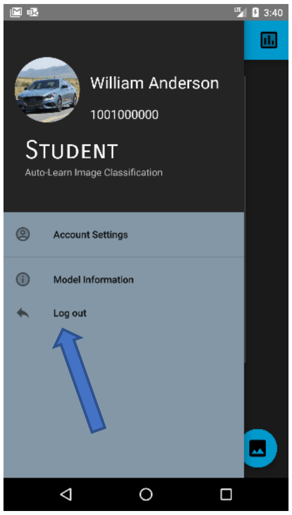


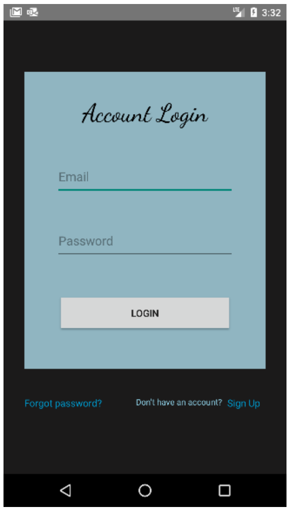
Send reset password email

william.anderson@mavs.uta.edu

Logging Out

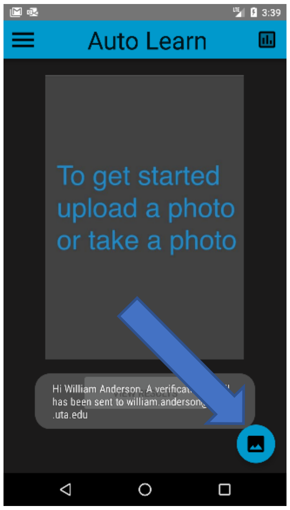
To log out of Auto-Learn, click on the Menu button and click the “Log Out” button. You will be redirected to the “Login Screen”. You may then switch accounts or sign in again.



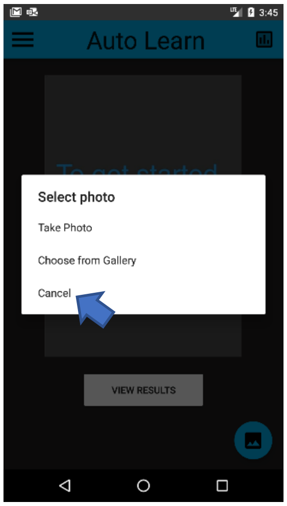


Running a Classification

To run a classification, click on the gallery icon in the bottom right corner. A pop-up will appear if you want to take a photo or upload an image. Clicking “Cancel” will take you back to the Home Screen



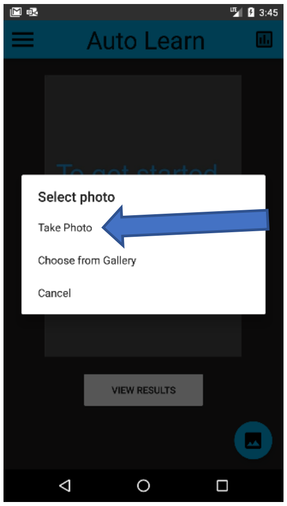
Hi William Anderson. A verificat... has been sent to william.anderson... .uta.edu

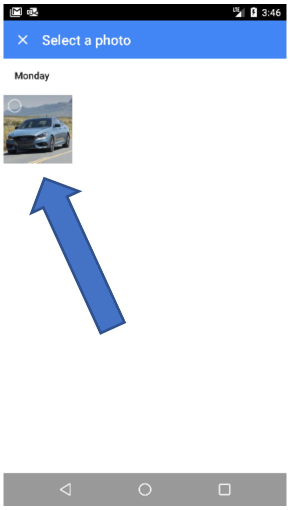


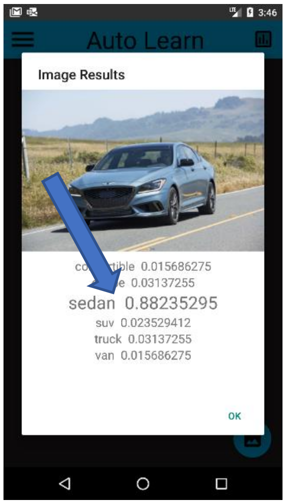
Uploading a Photo

To upload a photo, in the pop-up menu, click on “Choose from Gallery”. Select a photo. You will be redirected back to Auto-Learn where results will be displayed.

The results are of the model’s confidence level and scaled from 0 to 1. The classification with the highest confidence level will have a bold and large font compared to the five other categories. Clicking View Results will show the pop-up again from Home Screen.





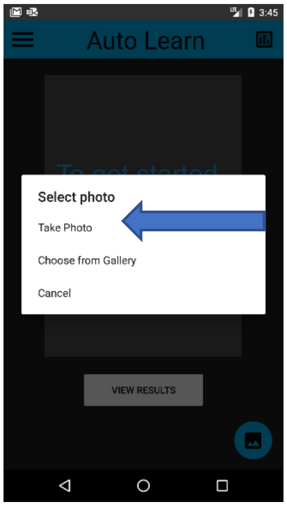


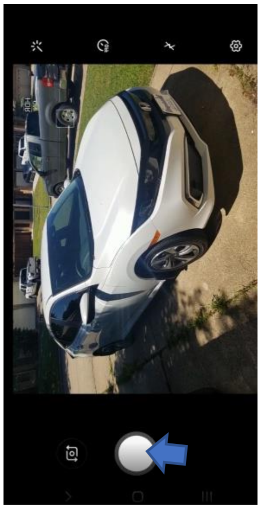
co...tible 0.015686275
...e 0.03137255
sedan 0.88235295
suv 0.023529412
truck 0.03137255
van 0.015686275

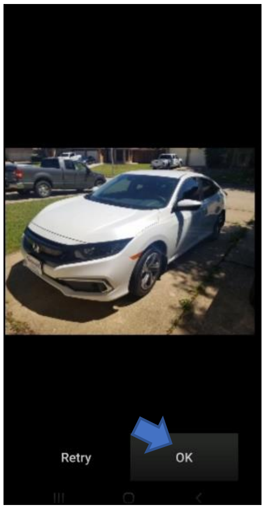
Taking a Photo

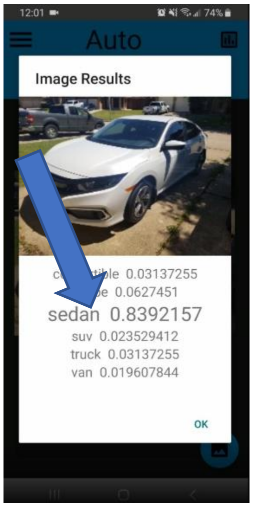
To upload a photo, in the pop-up menu, click on “Take Photo”. Take a phot and click the confirm button if you are satisfied with the photo or retake the photo. You will be then redirected back to Auto-Learn where results will be displayed.

The results are of the model’s confidence level and scaled from 0 to 1. The classification with the highest confidence level will have a bold and large font compared to the five other categories. Clicking View Results will show the pop-up again from Home Screen.







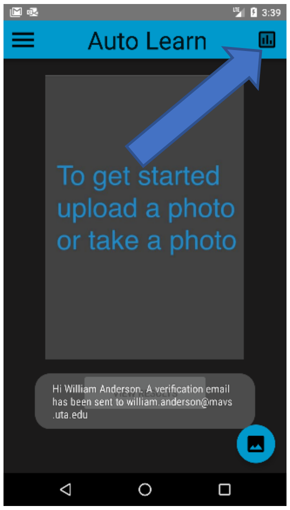


convertible 0.03137255
coupe 0.0627451
sedan 0.8392157
suv 0.023529412
truck 0.03137255
van 0.019607844

Image Statistics and Model Information

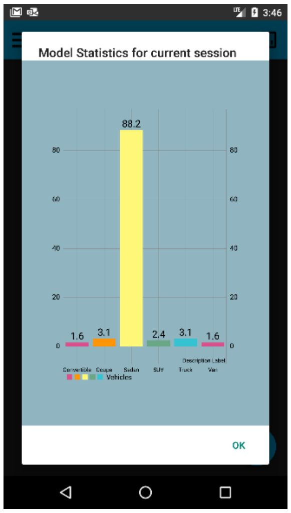
Image Statistics

To access the Image Statistics, click the “Statistics” icon in the upper right corner. A pop-up will appear showing a bar graph of the average confidence levels of each application. Adding all six classifications confidence levels will result in a total of 1. Clicking OK will cancel the dialog but you can click on the button at any time will reappear the dialog.



To get started upload a photo or take a photo

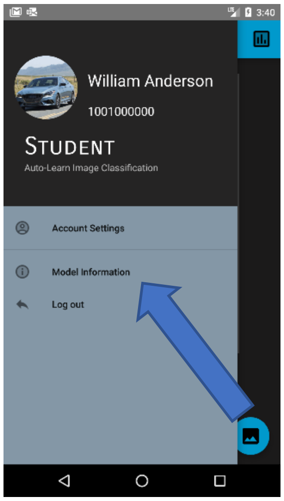
Hi William Anderson. A verification email has been sent to william.anderson@mavs.uta.edu

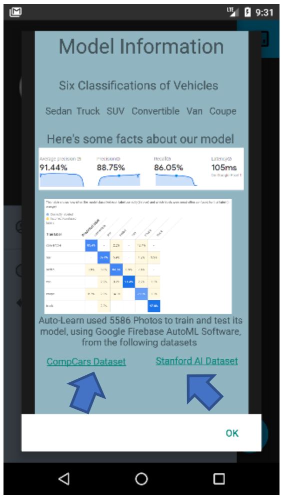


Model Statistics for current session

Model Information

Clicking on “Model Information” in the Menu will show a pop-up with the machine learning model information. It also has two hyperlinks to the datasets used for our model and clicking either one will redirect you to the website





Model Information

Six Classifications of Vehicles

Sedan Truck SUV Convertible Van Coupe

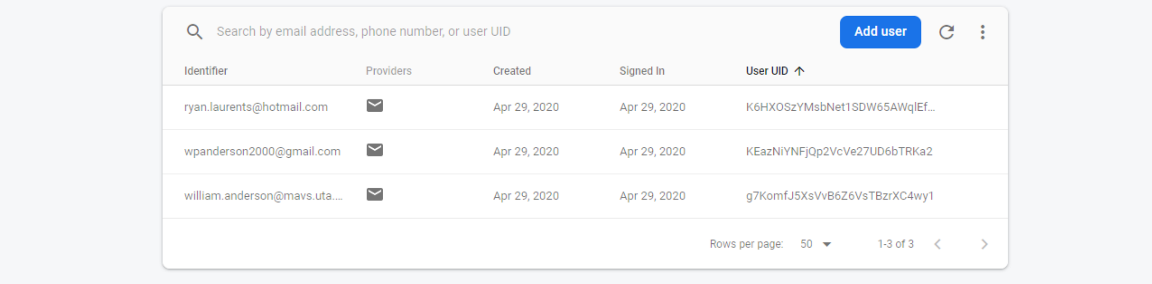
Here's some facts about our model

Average precision	Precision	Recall	Latency
91.44%	88.75%	86.05%	105ms

Auto-Learn used 5586 Photos to train and test its model, using Google Firebase AutoML Software, from the following datasets

11. Source Code

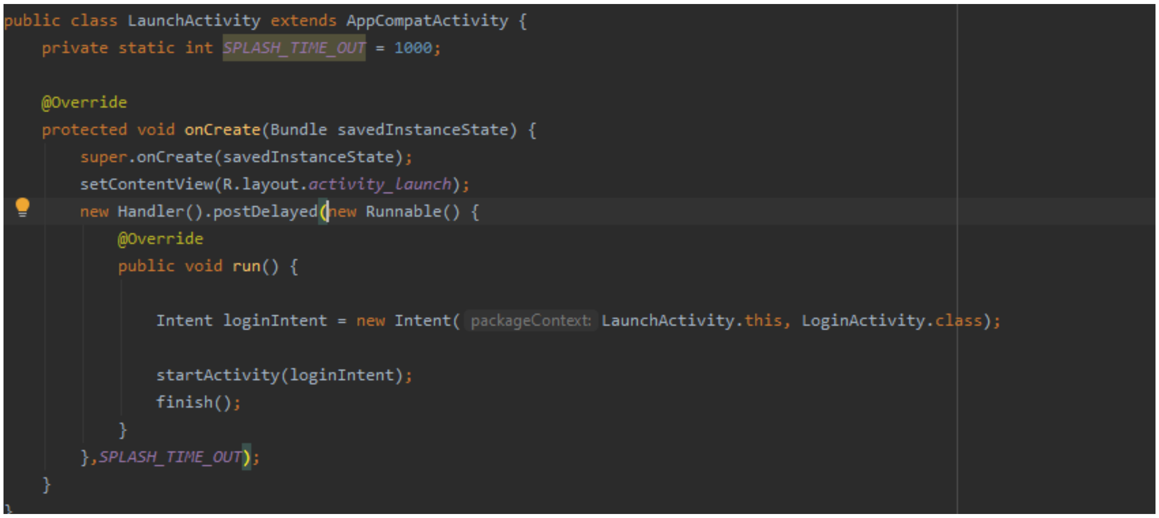
Following is a screenshot of our Firebase database for login:



Identifier	Providers	Created	Signed In	User UID ↑
ryan.laurents@hotmail.com	✉	Apr 29, 2020	Apr 29, 2020	K6HXOSzYMsbNet1SDW65AWqlEf...
wpanderson2000@gmail.com	✉	Apr 29, 2020	Apr 29, 2020	KEazNiYNFjQp2VcVe27UD6bTRKa2
william.anderson@mavs.uta....	✉	Apr 29, 2020	Apr 29, 2020	g7KomfJ5XsVvB6Z6VsTBzrXC4wy1

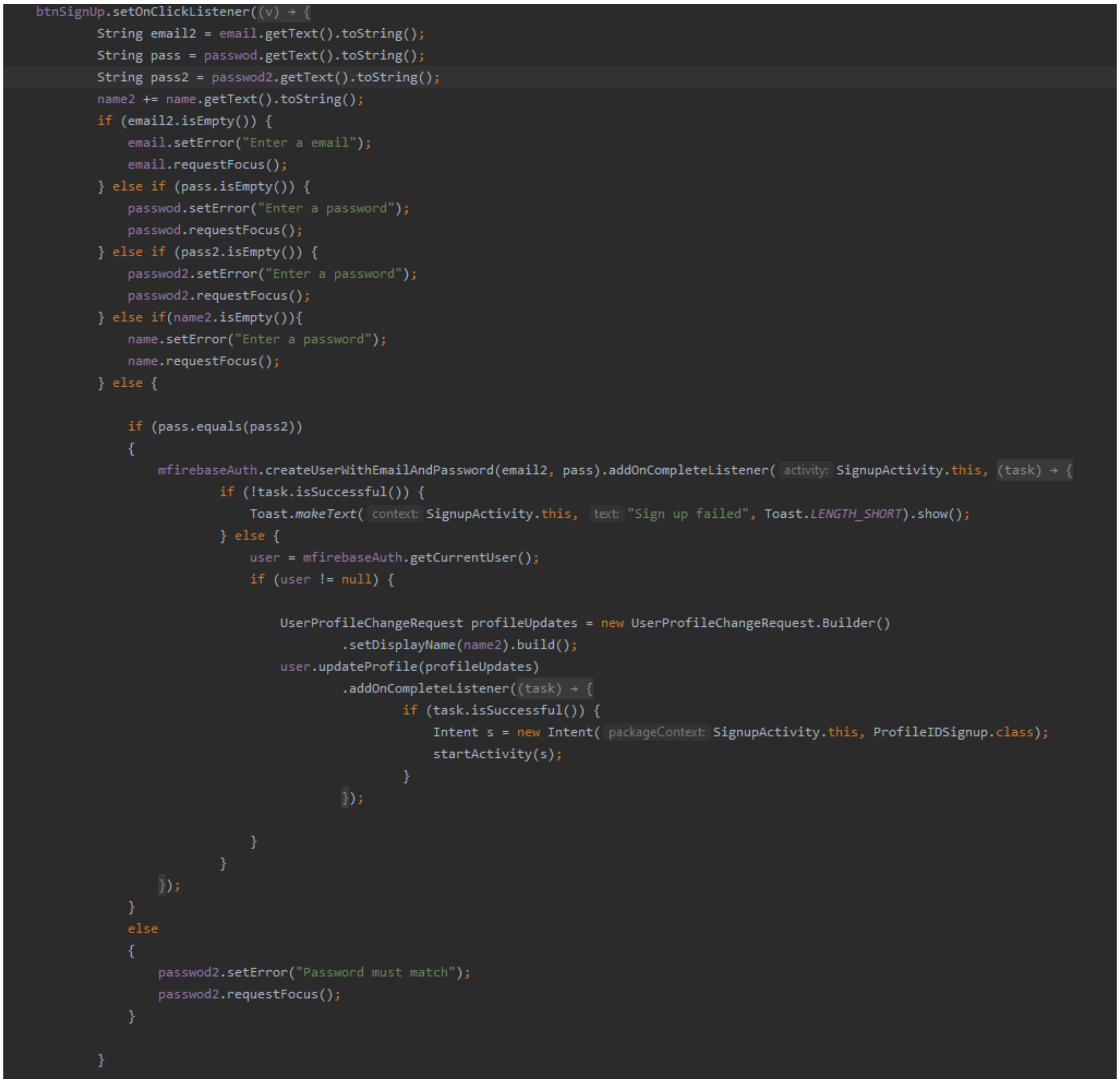
Following is our source code from the most significant Java files, ignoring the self-generated code by the IDE.

LaunchActivity.java



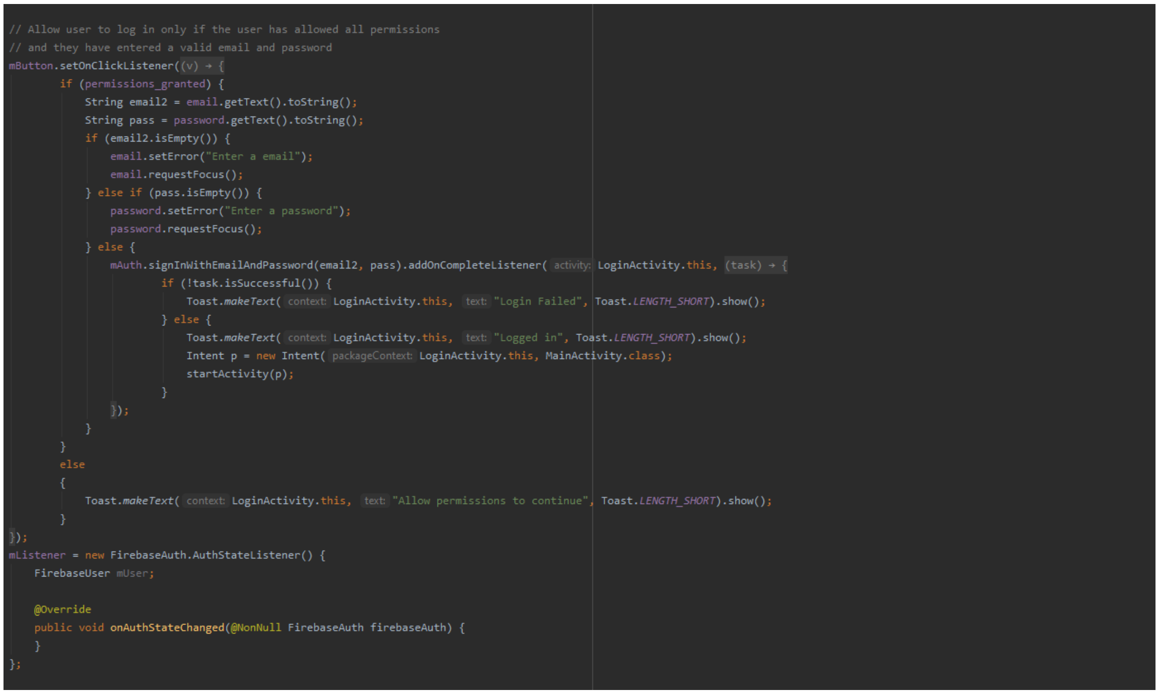
```java
public class LaunchActivity extends AppCompatActivity {
    private static int SPLASH_TIME_OUT = 1000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_launch);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {

                Intent loginIntent = new Intent( packageContext: LaunchActivity.this, LoginActivity.class);

                startActivity(loginIntent);
                finish();
            }
        },SPLASH_TIME_OUT);
    }
```

SignupActivity.java



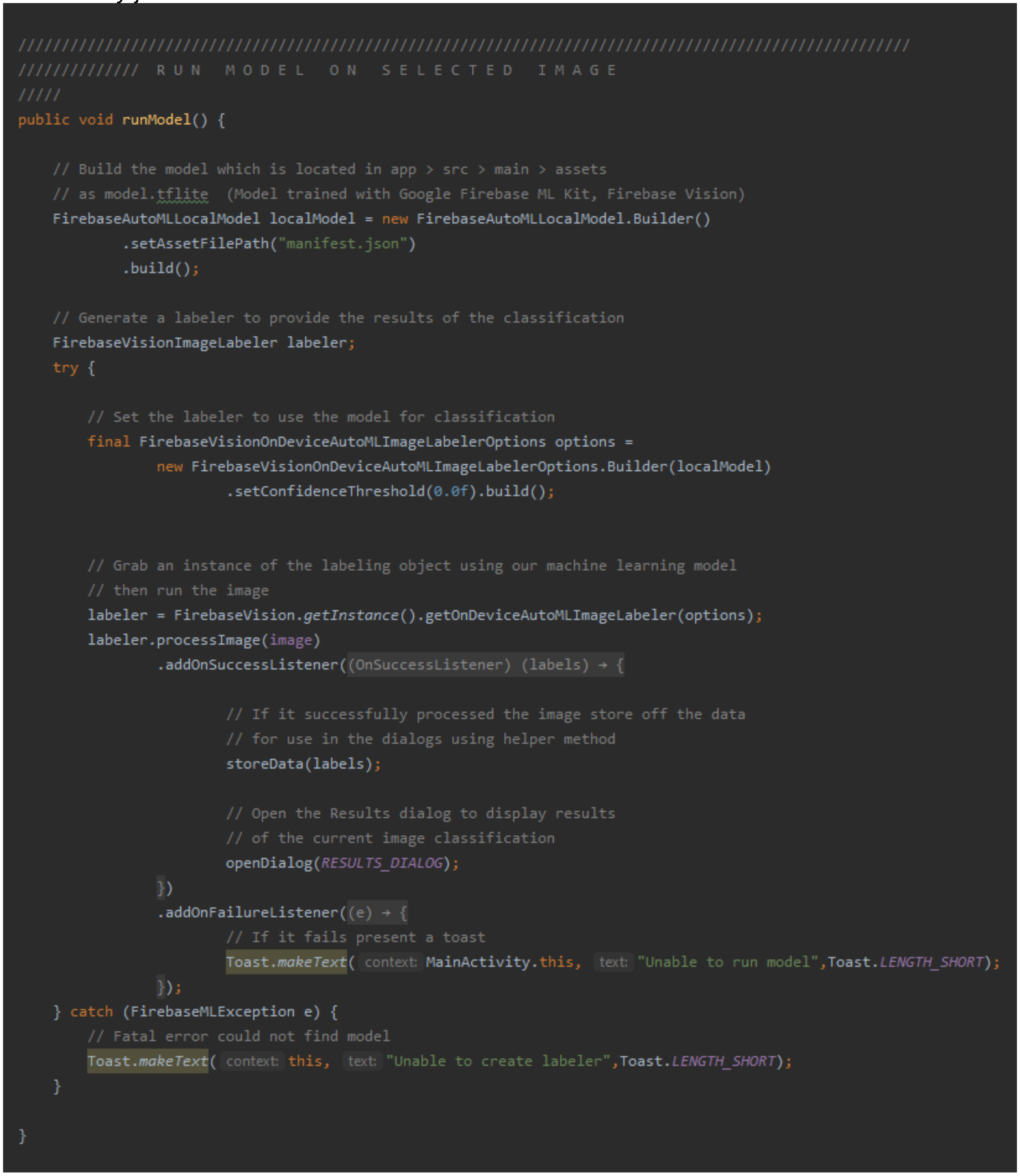
```java
btnSignUp.setOnClickListener((v) → {
        String email2 = email.getText().toString();
        String pass = passwod.getText().toString();
        String pass2 = passwod2.getText().toString();
        name2 += name.getText().toString();
        if (email2.isEmpty()) {
            email.setError("Enter a email");
            email.requestFocus();
        } else if (pass.isEmpty()) {
            passwod.setError("Enter a password");
            passwod.requestFocus();
        } else if (pass2.isEmpty()) {
            passwod2.setError("Enter a password");
            passwod2.requestFocus();
        } else if(name2.isEmpty()){
            name.setError("Enter a password");
            name.requestFocus();
        } else {

            if (pass.equals(pass2))
            {
                mfirebaseAuth.createUserWithEmailAndPassword(email2, pass).addOnCompleteListener( activity: SignupActivity.this, (task) → {
                        if (!task.isSuccessful()) {
                            Toast.makeText( context: SignupActivity.this,  text: "Sign up failed", Toast.LENGTH_SHORT).show();
                        } else {
                            user = mfirebaseAuth.getCurrentUser();
                            if (user != null) {

                                UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()
                                        .setDisplayName(name2).build();
                                user.updateProfile(profileUpdates)
                                        .addOnCompleteListener((task) → {
                                                if (task.isSuccessful()) {
                                                    Intent s = new Intent( packageContext: SignupActivity.this, ProfileIDSignup.class);
                                                    startActivity(s);
                                                }
                                        });

                            }
                        }
                });
            }
            else
            {
                passwod2.setError("Password must match");
                passwod2.requestFocus();
            }

        }
```

LoginActivity.java



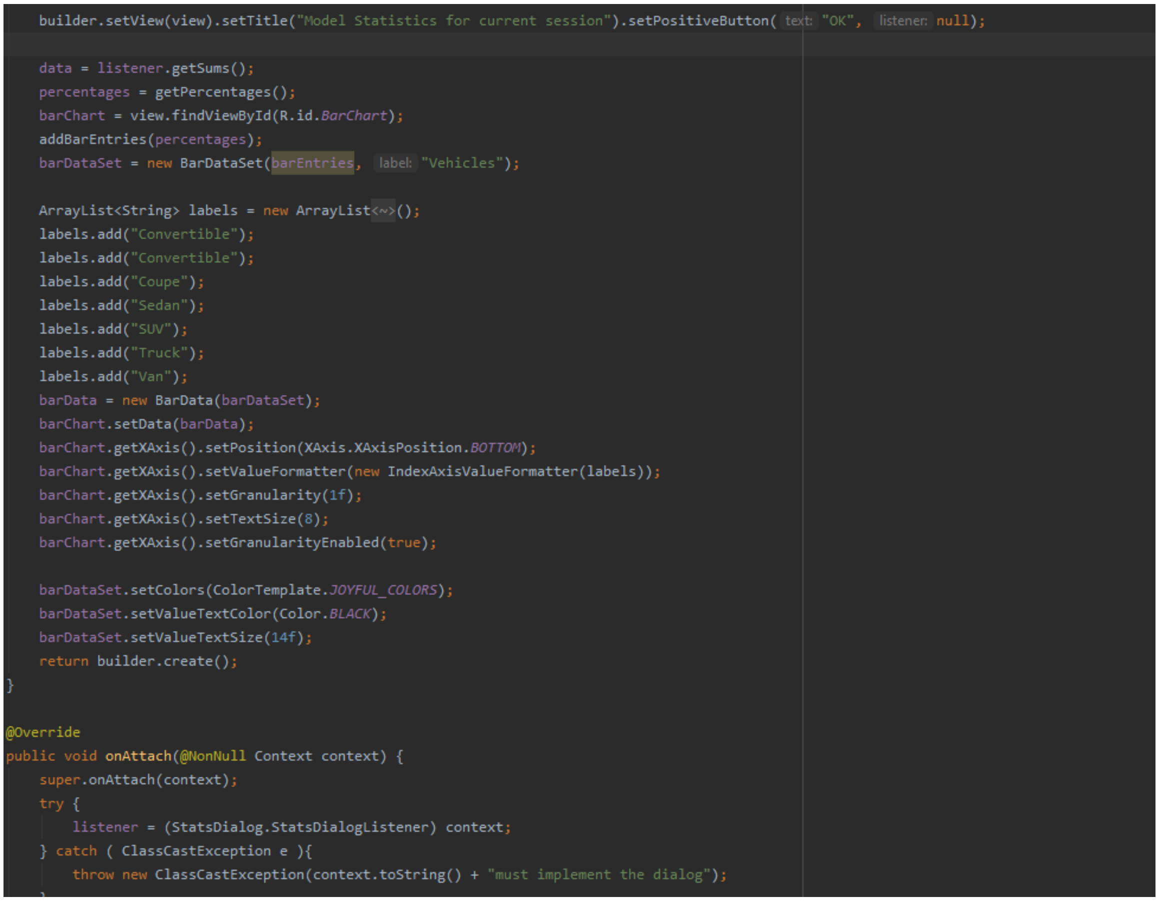
```java
// Allow user to log in only if the user has allowed all permissions
// and they have entered a valid email and password
mButton.setOnClickListener((v) → {
        if (permissions_granted) {
            String email2 = email.getText().toString();
            String pass = password.getText().toString();
            if (email2.isEmpty()) {
                email.setError("Enter a email");
                email.requestFocus();
            } else if (pass.isEmpty()) {
                password.setError("Enter a password");
                password.requestFocus();
            } else {
                mAuth.signInWithEmailAndPassword(email2, pass).addOnCompleteListener( activity: LoginActivity.this, (task) → {
                        if (!task.isSuccessful()) {
                            Toast.makeText( context: LoginActivity.this,  text: "Login Failed", Toast.LENGTH_SHORT).show();
                        } else {
                            Toast.makeText( context: LoginActivity.this,  text: "Logged in", Toast.LENGTH_SHORT).show();
                            Intent p = new Intent( packageContext: LoginActivity.this, MainActivity.class);
                            startActivity(p);
                        }
                });
            }
        }
        else
        {
            Toast.makeText( context: LoginActivity.this,  text: "Allow permissions to continue", Toast.LENGTH_SHORT).show();
        }
});
mListener = new FirebaseAuth.AuthStateListener() {
    FirebaseUser mUser;

    @Override
    public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {
    }
};
```

MainActivity.java:



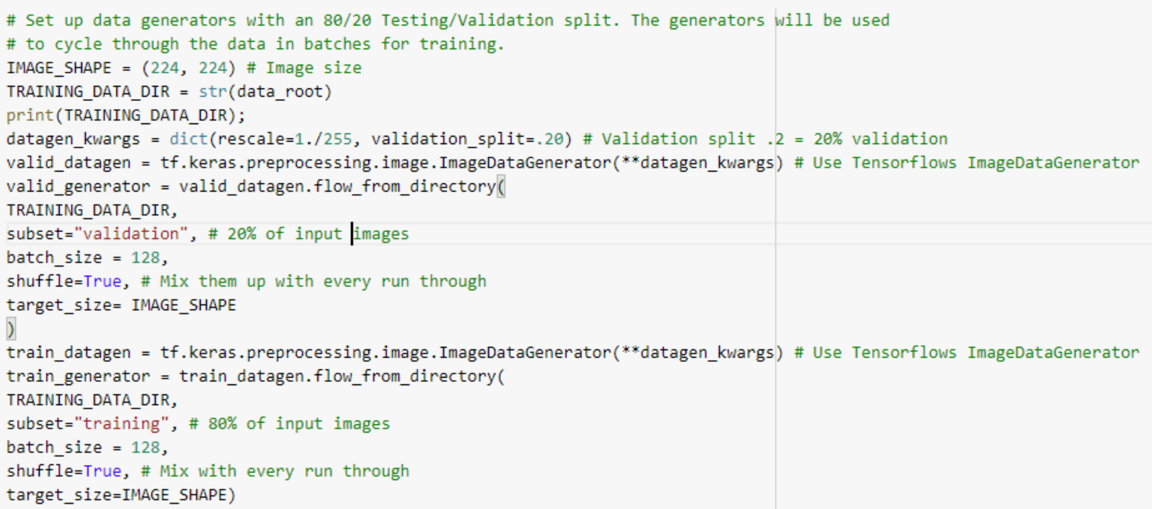
```
/////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////// RUN MODEL ON SELECTED IMAGE
/////
public void runModel() {

    // Build the model which is located in app > src > main > assets
    // as model.tflite  (Model trained with Google Firebase ML Kit, Firebase Vision)
    FirebaseAutoMLLocalModel localModel = new FirebaseAutoMLLocalModel.Builder()
            .setAssetFilePath("manifest.json")
            .build();

    // Generate a labeler to provide the results of the classification
    FirebaseVisionImageLabeler labeler;
    try {

        // Set the labeler to use the model for classification
        final FirebaseVisionOnDeviceAutoMLImageLabelerOptions options =
                new FirebaseVisionOnDeviceAutoMLImageLabelerOptions.Builder(localModel)
                        .setConfidenceThreshold(0.0f).build();


        // Grab an instance of the labeling object using our machine learning model
        // then run the image
        labeler = FirebaseVision.getInstance().getOnDeviceAutoMLImageLabeler(options);
        labeler.processImage(image)
                .addOnSuccessListener((OnSuccessListener) (labels) → {

                        // If it successfully processed the image store off the data
                        // for use in the dialogs using helper method
                        storeData(labels);

                        // Open the Results dialog to display results
                        // of the current image classification
                        openDialog(RESULTS_DIALOG);
                })
                .addOnFailureListener((e) → {
                        // If it fails present a toast
                        Toast.makeText( context: MainActivity.this,  text: "Unable to run model",Toast.LENGTH_SHORT);
                });
    } catch (FirebaseMLException e) {
        // Fatal error could not find model
        Toast.makeText( context: this,  text: "Unable to create labeler",Toast.LENGTH_SHORT);
    }

}
```

ResultsDalog.java

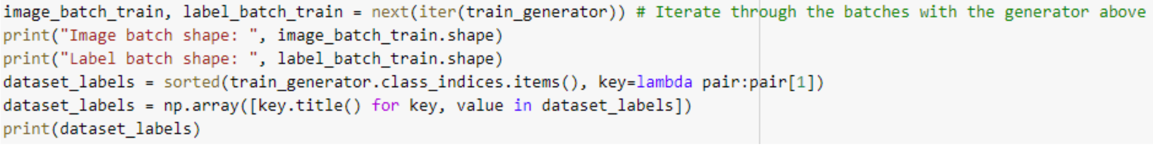


```
    builder.setView(view).setTitle("Model Statistics for current session").setPositiveButton( text: "OK",  listener: null);

    data = listener.getSums();
    percentages = getPercentages();
    barChart = view.findViewById(R.id.BarChart);
    addBarEntries(percentages);
    barDataSet = new BarDataSet(barEntries,  label: "Vehicles");

    ArrayList<String> labels = new ArrayList<~>();
    labels.add("Convertible");
    labels.add("Convertible");
    labels.add("Coupe");
    labels.add("Sedan");
    labels.add("SUV");
    labels.add("Truck");
    labels.add("Van");
    barData = new BarData(barDataSet);
    barChart.setData(barData);
    barChart.getXAxis().setPosition(XAxis.XAxisPosition.BOTTOM);
    barChart.getXAxis().setValueFormatter(new IndexAxisValueFormatter(labels));
    barChart.getXAxis().setGranularity(1f);
    barChart.getXAxis().setTextSize(8);
    barChart.getXAxis().setGranularityEnabled(true);

    barDataSet.setColors(ColorTemplate.JOYFUL_COLORS);
    barDataSet.setValueTextColor(Color.BLACK);
    barDataSet.setValueTextSize(14f);
    return builder.create();
}

@Override
public void onAttach(@NonNull Context context) {
    super.onAttach(context);
    try {
        listener = (StatsDialog.StatsDialogListener) context;
    } catch ( ClassCastException e ){
        throw new ClassCastException(context.toString() + "must implement the dialog");
    }
```

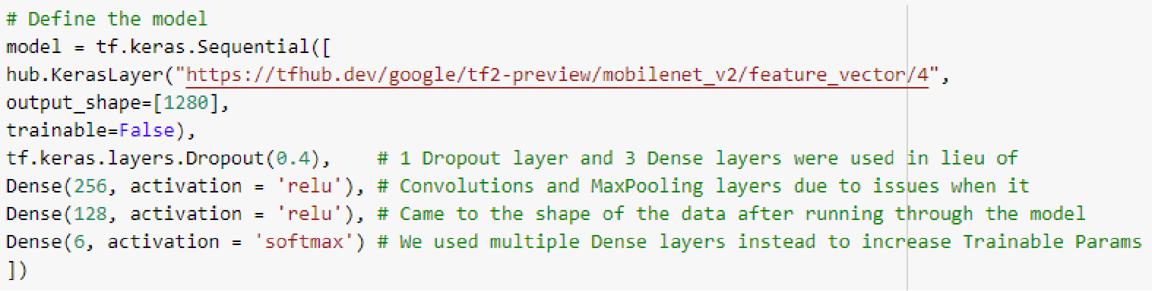
All following screenshots are from the machine learning aspect of the project. We made use of Google Collab and Jupyter, although we used more of Google Collab so we could access our data stored in a shared Google drive. These images are all from the same source code file.



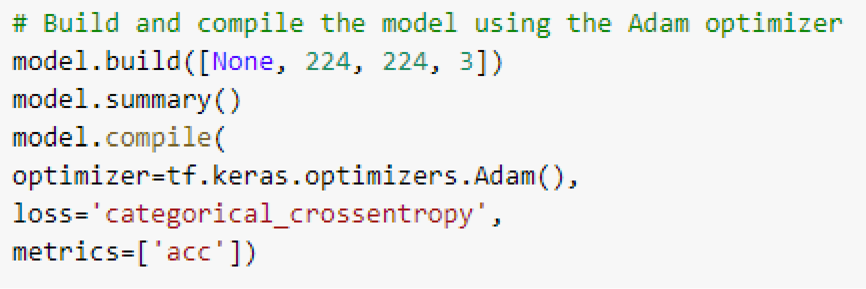
```
# Set up data generators with an 80/20 Testing/Validation split. The generators will be used
# to cycle through the data in batches for training.
IMAGE_SHAPE = (224, 224) # Image size
TRAINING_DATA_DIR = str(data_root)
print(TRAINING_DATA_DIR);
datagen_kwargs = dict(rescale=1./255, validation_split=.20) # Validation split .2 = 20% validation
valid_datagen = tf.keras.preprocessing.image.ImageDataGenerator(**datagen_kwargs) # Use Tensorflows ImageDataGenerator
valid_generator = valid_datagen.flow_from_directory(
TRAINING_DATA_DIR,
subset="validation", # 20% of input images
batch_size = 128,
shuffle=True, # Mix them up with every run through
target_size= IMAGE_SHAPE
)
train_datagen = tf.keras.preprocessing.image.ImageDataGenerator(**datagen_kwargs) # Use Tensorflows ImageDataGenerator
train_generator = train_datagen.flow_from_directory(
TRAINING_DATA_DIR,
subset="training", # 80% of input images
batch_size = 128,
shuffle=True, # Mix with every run through
target_size=IMAGE_SHAPE)
```



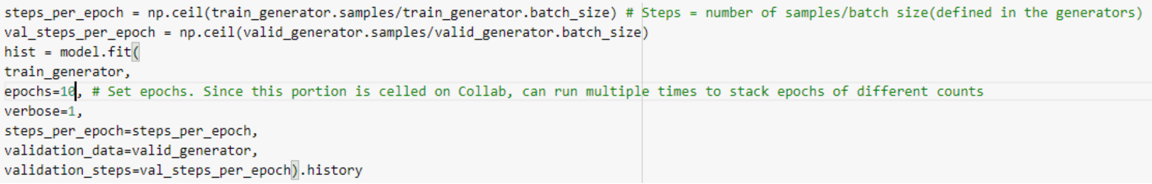
```
image_batch_train, label_batch_train = next(iter(train_generator)) # Iterate through the batches with the generator above
print("Image batch shape: ", image_batch_train.shape)
print("Label batch shape: ", label_batch_train.shape)
dataset_labels = sorted(train_generator.class_indices.items(), key=lambda pair:pair[1])
dataset_labels = np.array([key.title() for key, value in dataset_labels])
print(dataset_labels)
```



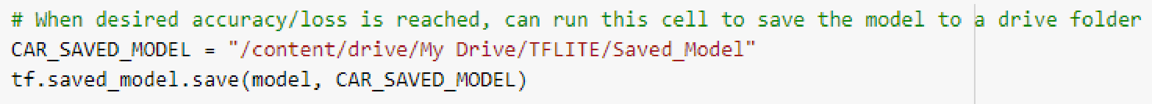
```
# Define the model
model = tf.keras.Sequential([
hub.KerasLayer("https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4",
output_shape=[1280],
trainable=False),
tf.keras.layers.Dropout(0.4),    # 1 Dropout layer and 3 Dense layers were used in lieu of
Dense(256, activation = 'relu'), # Convolutions and MaxPooling layers due to issues when it
Dense(128, activation = 'relu'), # Came to the shape of the data after running through the model
Dense(6, activation = 'softmax') # We used multiple Dense layers instead to increase Trainable Params
])
```



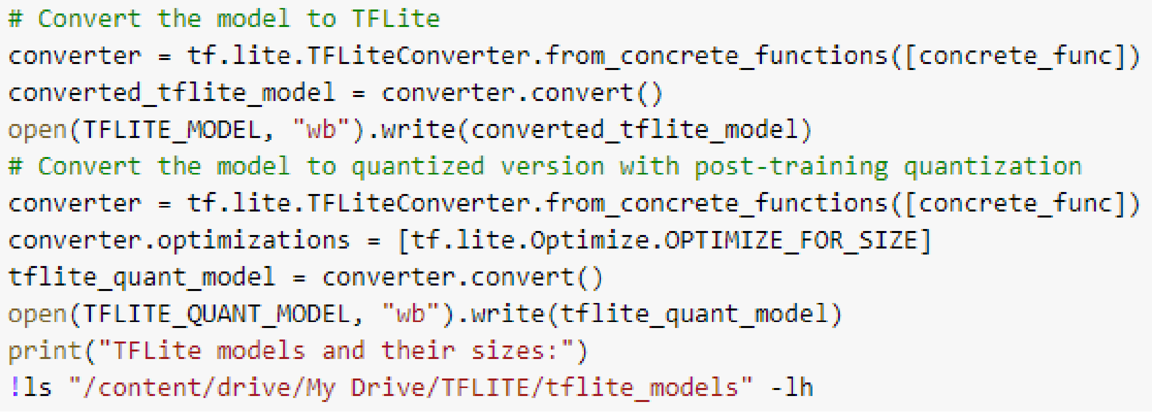
```
# Build and compile the model using the Adam optimizer
model.build([None, 224, 224, 3])
model.summary()
model.compile(
optimizer=tf.keras.optimizers.Adam(),
loss='categorical_crossentropy',
metrics=['acc'])
```



```
steps_per_epoch = np.ceil(train_generator.samples/train_generator.batch_size) # Steps = number of samples/batch size(defined in the generators)
val_steps_per_epoch = np.ceil(valid_generator.samples/valid_generator.batch_size)
hist = model.fit(
train_generator,
epochs=10, # Set epochs. Since this portion is celled on Collab, can run multiple times to stack epochs of different counts
verbose=1,
steps_per_epoch=steps_per_epoch,
validation_data=valid_generator,
validation_steps=val_steps_per_epoch).history
```



```
# When desired accuracy/loss is reached, can run this cell to save the model to a drive folder
CAR_SAVED_MODEL = "/content/drive/My Drive/TFLITE/Saved_Model"
tf.saved_model.save(model, CAR_SAVED_MODEL)
```



```
# Convert the model to TFLite
converter = tf.lite.TFLiteConverter.from_concrete_functions([concrete_func])
converted_tflite_model = converter.convert()
open(TFLITE_MODEL, "wb").write(converted_tflite_model)
# Convert the model to quantized version with post-training quantization
converter = tf.lite.TFLiteConverter.from_concrete_functions([concrete_func])
converter.optimizations = [tf.lite.Optimize.OPTIMIZE_FOR_SIZE]
tflite_quant_model = converter.convert()
open(TFLITE_QUANT_MODEL, "wb").write(tflite_quant_model)
print("TFLite models and their sizes:")
!ls "/content/drive/My Drive/TFLITE/tflite_models" -lh
```

Appendix:

None