Team 4: Hunter

Shalin Patel, Sandeep Pabolu, Akhila Katta, Jackie Batson

Our team attempted 3 different projects; this is the documentation for all

Source code and documentation for projects 1 and 2

Source code and documentation for final project

YouTube video of final project

Project 1 Proposal: Smart Home using IoT

Motivation

Our motivation for this project is to gain experience working with the Internet of Things (IoT) by creating a smart home using relatively cheap and readily available materials. The Internet of Things involves the interconnection of everyday objects, equipped with sensors and transmitters, through an online network in order to exchange data and automatically notify users of specific changes. For the smart home, household items such as lights, doors, mailboxes, and appliances can be connected to the internet to enable a homeowner to monitor and detect changes. This frequent monitoring can potentially help reduce energy costs by making the home run more efficiently.

Significance/Uniqueness

Many homeowners do not currently utilize the IoT in their homes, but the IoT is considered the future of the way we interact with devices in the world. In 2015, 10 billion devices were connected to the internet, a number projected to grow to 34 billion by 2020, with 24 billion IoT devices [1]. Consumers, however, have been hesitant to embrace IoT technology, though simple smart home sensors can make the IoT more accessible to the average homeowner. We consider our project to be pushing the edge of this internet revolution.

Objective

Our objective is to connect regular objects in the home using Arduino and Raspberry Pi sensors that can collect data and be monitored by the user via an Android application. Some of the available sensors include motion and sound detection, temperature/humidity, smoke and gas detection, distance detection, and GPS monitoring. We hope to combine these in ways that would be useful to a homeowner to observe. Since this is also a software project, we also plan to focus on application aspects such as security, usability, and data analysis.

System Features

Some examples of sensor monitoring and notification include:

- When user is not home, notify user of door opening or motion detection inside the home (security)
- If door is opened, turn on lights
- Notify when mail is delivered

- Detect smoke, gas, and increased temperature; notify user when not home of possible fire or gas leak
- Place GPS tracker on dog collar and monitor dog location at all times
- Detect number of cat visits to the cat box since last cleaning

Since the sensor system is made of components, we can add more or reduce these components as we see fit during the course of the project. Additional system features will include measures to make the system secure; a usable and intuitive interface; and data collection, processing, and reporting tools (for example, which rooms are coldest or warmest for HVAC adjustment, where does the dog spend most of his time outside, etc.).

Similar projects

Raspberry Pi and Arduino have been used by many other tech hobbyists for home automation. Some examples:

- Uber Automation w/Arduino and Pi [2]
- Building a Home Automation System with OpenHAB to Control LEDs Wirelessly [3]
- Many additional examples on the Raspberry Pi website [4]

Alternative projects

Smartwatch translator

Charging slot locator

Android and web monitoring of worker time and location and dispatch to jobs

References

1. “Here are the IoT trends that will change the way businesses, governments, and consumers interact with the world”. Greenhough, J. and Camhi, J. Business Insider. August 29, 2016. Accessed January 29, 2016 at http://www.businessinsider.com/top-internet-of-things-trends-2016-1
2. Uber Home Automation with Arduino and Pi. Accessed January 29, 2016 at http://www.instructables.com/id/Uber-Home-Automation-w-Arduino-Pi/
3. Building a Home Automation System with OpenHab to Control LEDs Wirelessly. Bunker, J. Accessed January 29, 2016 at http://makezine.com/projects/building-a-home-automation-system-with-openhab-to-control-leds-wirelessly/
4. Accessed January 29, 2016 at https://www.raspberrypi.org/blog/tag/home-automation/

Project 2: Second Iteration and Project Work

Introduction

Our project deals with the cardboard application and majorly focuses on the voice or speech control of the application which allows the user a lot many features in addition to the current cardboard application. This application can be easily installed as the other applications making it more handy for the cardboard users.

Project Goal and Objectives (revised)

Our objective is to make the next level of the cardboard application with more sophisticated implementation of the voice control feature. In the regular cardboard application any act needs the use of the touch. So why not go hands free? Using the voice control we can browse through the playing video front and back, play the next video, skip the video, pause the video, increase the volume levels in the video and the list can be extended for a large set of operations. Our future scope will be extending the features of security to this voice like the google services have provided in order to restrict the intersections of the different voices.

Significance/Uniqueness

None of the cardboard applications render the feature of speech control in it. So to make man’s life more sophisticated and interesting the combination of the features of the ok google in the cardboard will become a new area of interest. This opens a new style of using the application’s feature like by simply connecting the android device in the cardboard we can start streaming through pages, become hands free and that too within the small pricing range. I think these features make our project to be pushing the edge of this internet revolution.

System Features:

The main features include:

- Browsing through the video with the voice over.
- Opening the new video or selecting the next video.
- Skip the video for specific time.
- Moving to the next video.
- Playing the previous video.
- Change in the levels of the volume during the video.

Second Iteration Report

The user uses mobile phone and its camera to take pictures. When user wants to do any activity like comparison of image, getting face attributes, the application will pass the data to web API. Then as per retrieved result, there will be application logic which manipulates the response and displays results. For google home authentication, is the user's validation response is true, then it will allow google home to operate. There is the provision of Face DB which stores all images.

5. Project Management

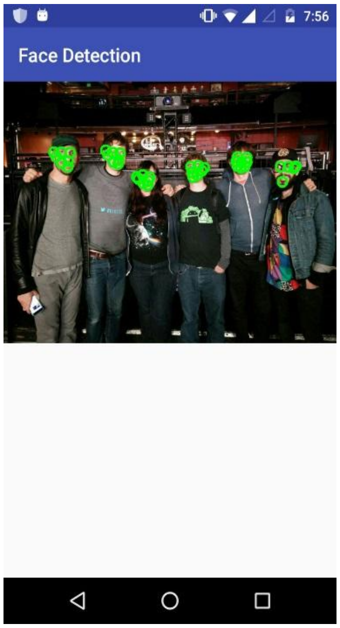
For iteration 2, We completed the following tasks:

1. Research cardboard and face recognition apis
2. Created the login and the register page.
3. Developed the face recognition application for the login.
4. Working application of the cardboard on an android phone.

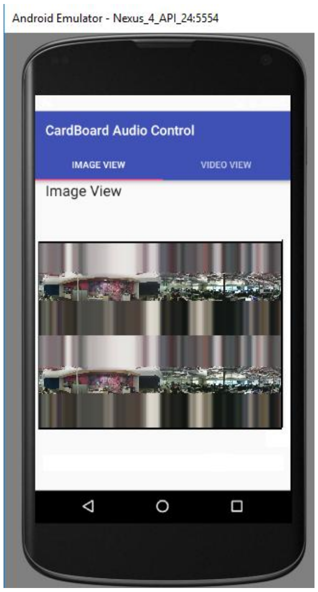
   The relative application images are presented in the following:

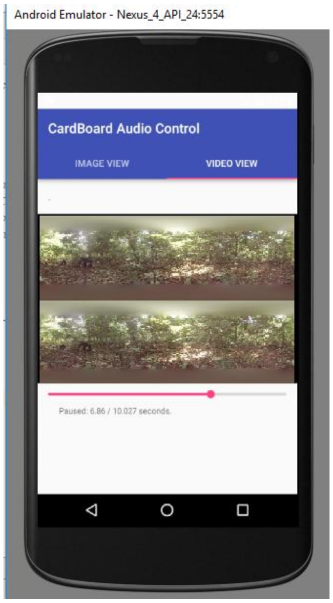
Test results:

Face recognition as the sign in option:



Working cardboard application:





Paused: 6.86 / 10.027 seconds.

References:

1. Innovation spot light http://ieeexplore-spotlight.ieee.org/article/virtual-presence-avatars-and-vr-applications/
2. http://through-the-interface.typepad.com/through_the_interface/2014/11/adding-more-speech-recognition-to-our-stereoscopic-google-cardboard-viewer.html
3. http://ieeexplore-spotlight.ieee.org/article/virtual-presence-avatars-and-vr-applications/
4. https://www.wired.com/2015/12/google-cardboard-camera-app/
5. https://uploadvr.com/vr-101-whats-the-difference-between-rift-vive-playstation-vr-cardboard-and-gear-vr/

Project 3: Deep Learning with Clarifai API

I. INTRODUCTION

This project will be to create a Native/Web hybrid mobile application in which a user can take a picture, and get an identification and information about what the image is. The project will involve training and testing of images using the Clarifai open API for deep learning purposes.

II. PROJECT GOAL AND OBJECTIVES

1) The overall goal of the project is to explore deep learning and hybrid mobile app development.
2) Problem statement: create a native/web hybrid mobile application in which a user can take a picture and get an identification and information about the image using the Clarifai open API.
3) Specific features of the app will be an interface with the device's camera so the user can take a picture. The picture will then query the Clarifai API, which will return an identification of the picture. Using that identification, the app will return additional information about the picture to the user from its own database. The Clarifai API will be pre-trained using specific categories of images for the application.
4) Significance: machine learning and deep learning are considered the next wave in technological advancement. The combination of expanding storage and processing power, along with the use of the internet, has greatly increased the amount of data available for computer learning. A computer can be trained using available data to create a model that can be used to predict the output from new input. Our app will be significant because it will train a model for a specific image recognition task, detailed for our specific needs.

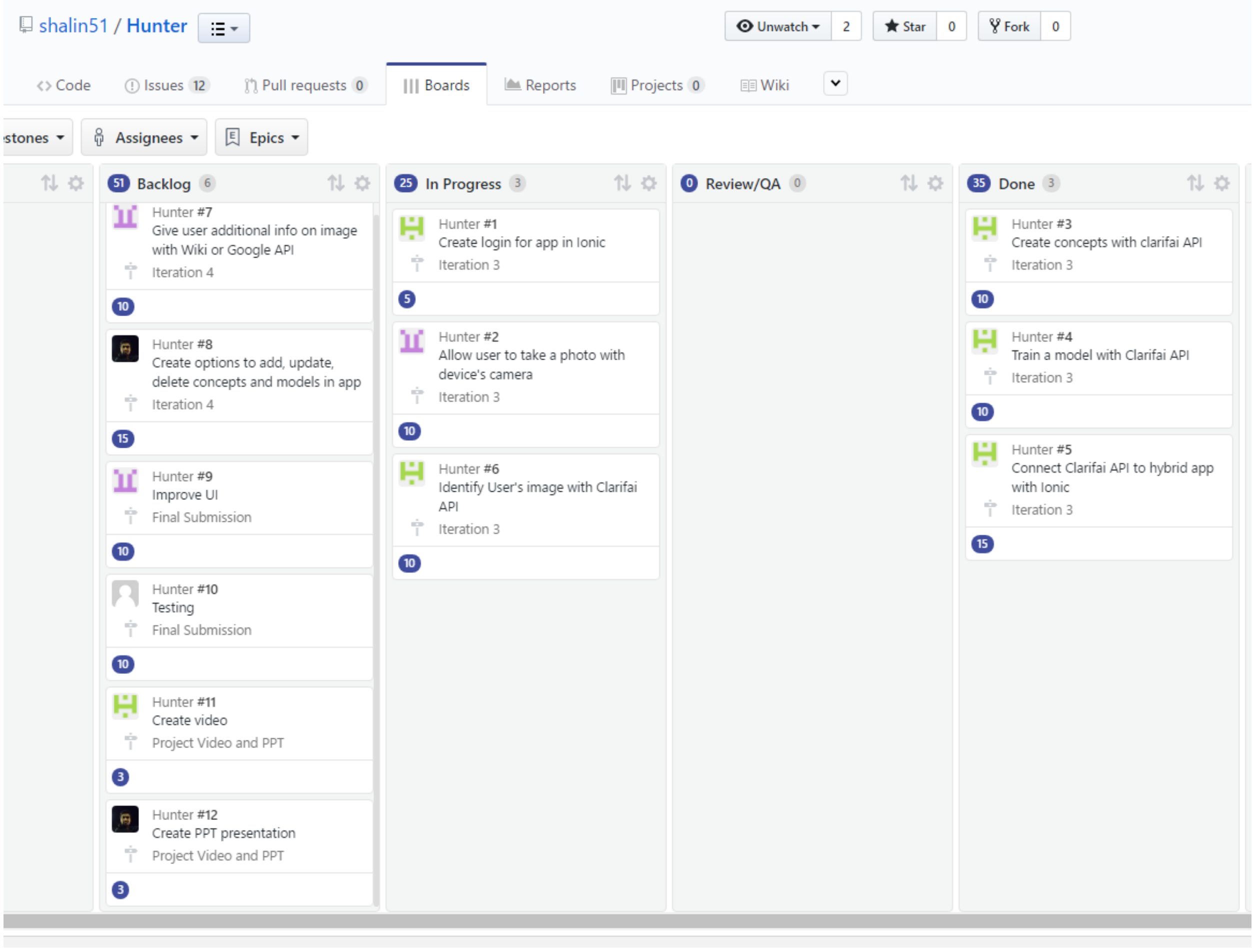
III. PROJECT PLAN

1) Scheduled use cases (Product Backlog):

PRODUCT BACKLOG			
Story	Description	Story Points	Priority
1	As a user, I would like to be able to log in to the system via my phone	5	3
2	As a user, I would like to be able to take a photo with my phone from the app	10	2
3	As a user, I would like to have my captured image identified.	10	1
4	As a user, I would like to have additional information provided about the captured image.	15	8
5	As an administrator, I'd like to use the app to add categories of images (concepts) to train	15	4
6	As an administrator, I'd like to use the app to create models for image recognition	20	5
7	As an administrator, I'd like to use the app to update models	15	6

8	As an administrator, I’d like to use the app to delete models	5	7

2) Timelines and responsibility:



51 Backlog 6

Hunter #7
Give user additional info on image with Wiki or Google API
Iteration 4
10

Hunter #8
Create options to add, update, delete concepts and models in app
Iteration 4
15

Hunter #9
Improve UI
Final Submission
10

Hunter #10
Testing
Final Submission
10

Hunter #11
Create video
Project Video and PPT
3

Hunter #12
Create PPT presentation
Project Video and PPT
3

25 In Progress 3

Hunter #1
Create login for app in Ionic
Iteration 3
5

Hunter #2
Allow user to take a photo with device's camera
Iteration 3
10

Hunter #6
Identify User's image with Clarifai API
Iteration 3
10

0 Review/QA 0

35 Done 3

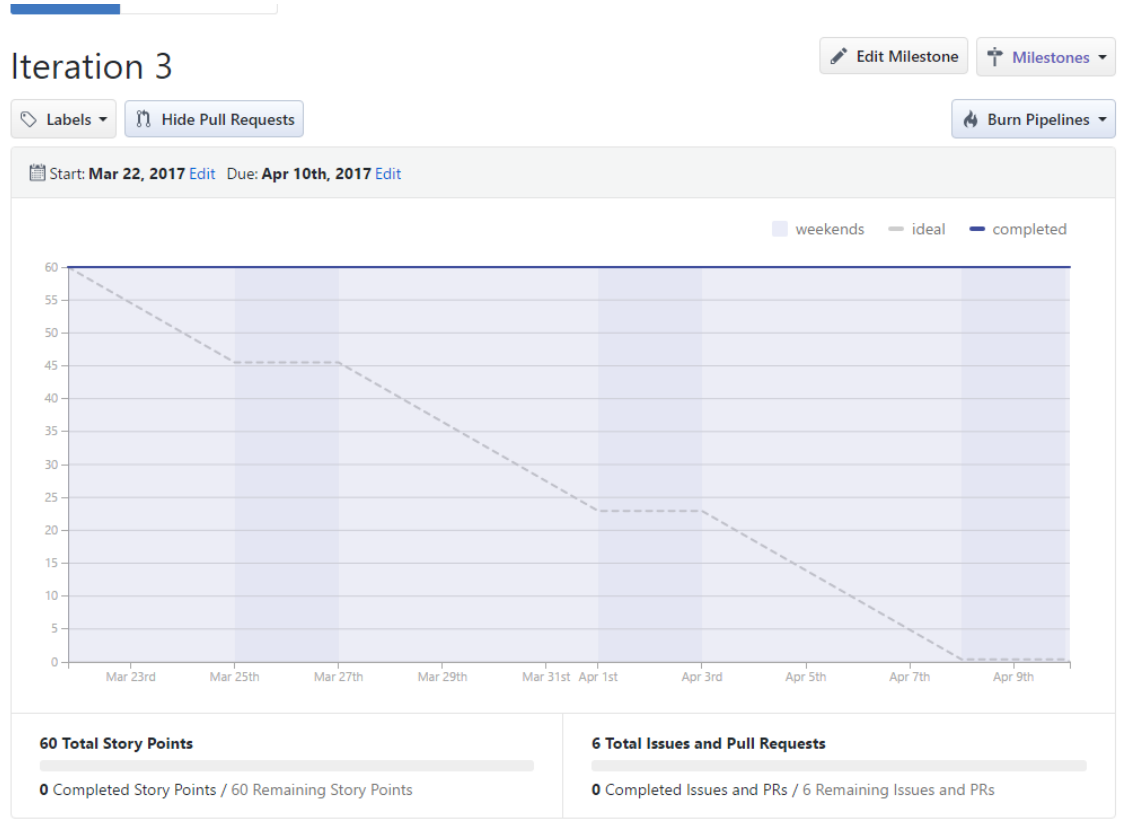
Hunter #3
Create concepts with clarifai API
Iteration 3
10

Hunter #4
Train a model with Clarifai API
Iteration 3
10

Hunter #5
Connect Clarifai API to hybrid app with Ionic
Iteration 3
15

3) Burndown chart:

i. Iteration 3:



Iteration 3

Start: **Mar 22, 2017** Edit Due: **Apr 10th, 2017** Edit

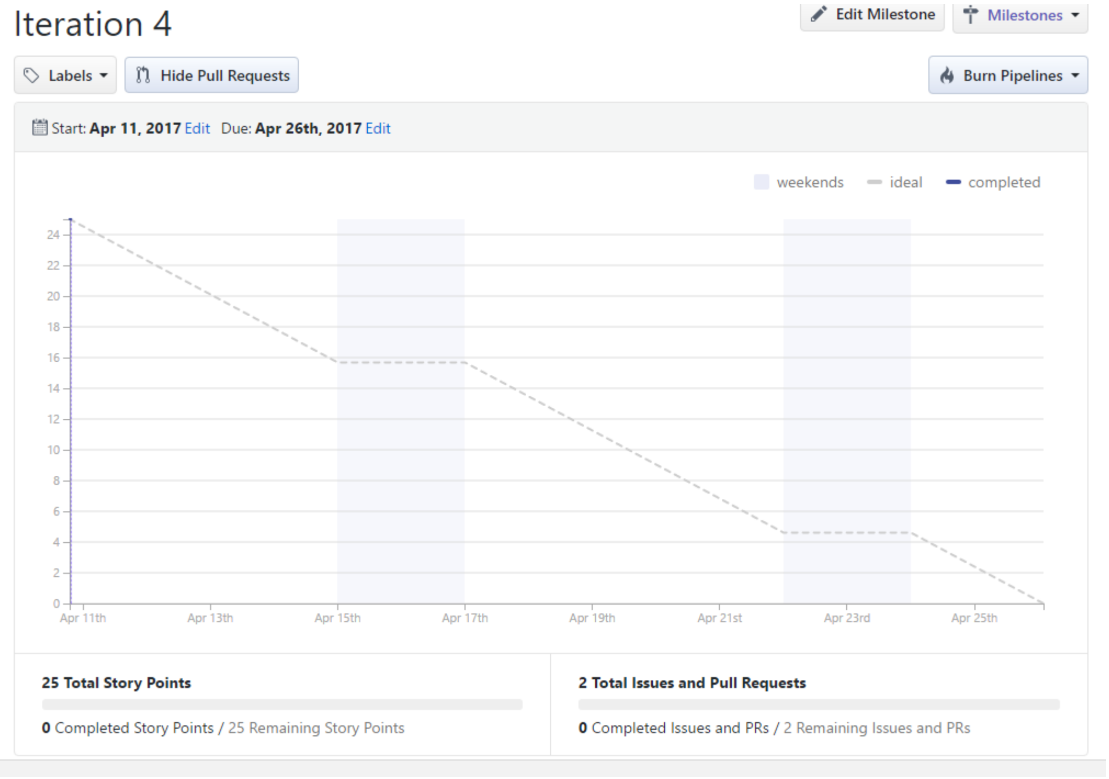
60 Total Story Points

0 Completed Story Points / 60 Remaining Story Points

6 Total Issues and Pull Requests

0 Completed Issues and PRs / 6 Remaining Issues and PRs

ii. Iteration 4:



Iteration 4

Start: **Apr 11, 2017** Edit Due: **Apr 26th, 2017** Edit

25 Total Story Points

0 Completed Story Points / 25 Remaining Story Points

2 Total Issues and Pull Requests

0 Completed Issues and PRs / 2 Remaining Issues and PRs

IV. ITERATION 3 REPORT

1) Services/APIs used:

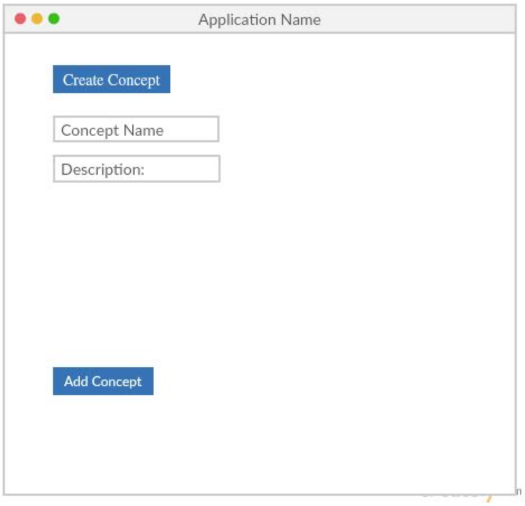
   We used the Clarifai API (https://clarifai.com) . With Clarifai, we could upload our own categories of training images (called concepts by Clarifai). Using those images, Clarifai creates a model specific to our unique case for image recognition. It returns a tag for the image along with a percent confidence in that label.

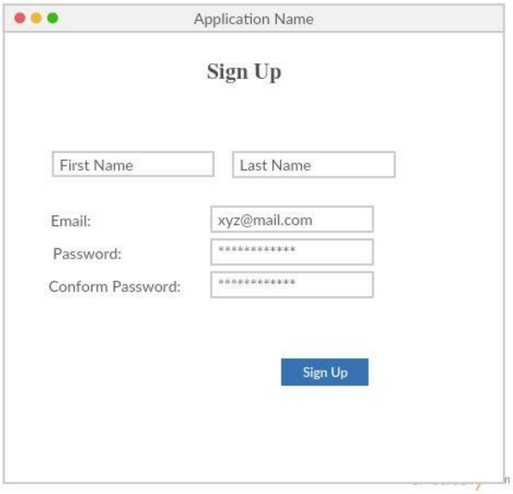
   Other technologies:

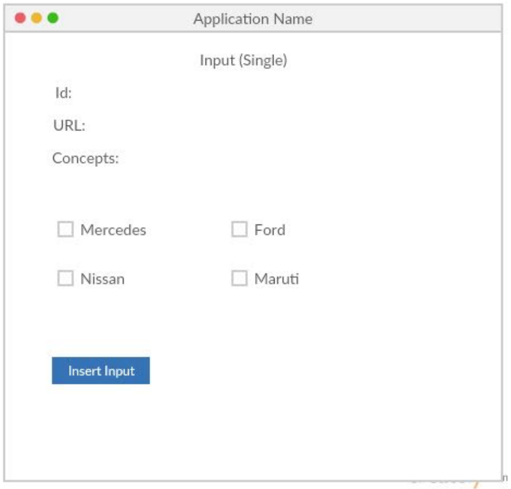
   We also will use the Ionic framework for creating a hybrid web/Android application. It will be written in Java with Angular.js, HTML, CSS, Bootstrap.

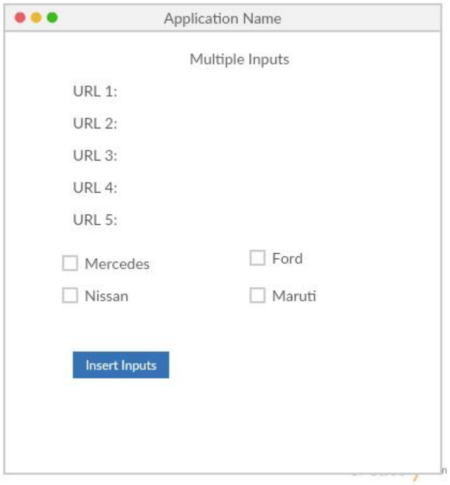
2) Detailed design

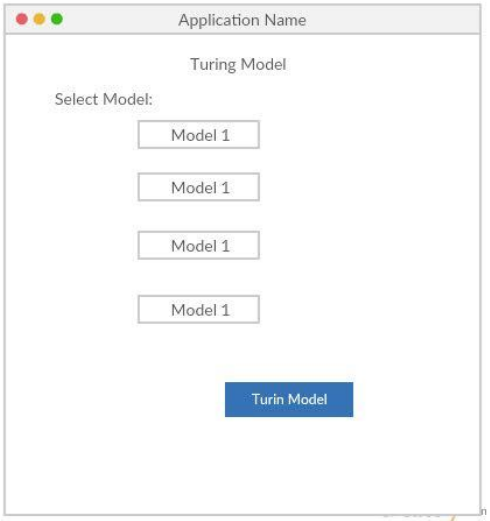
   a. Wireframes

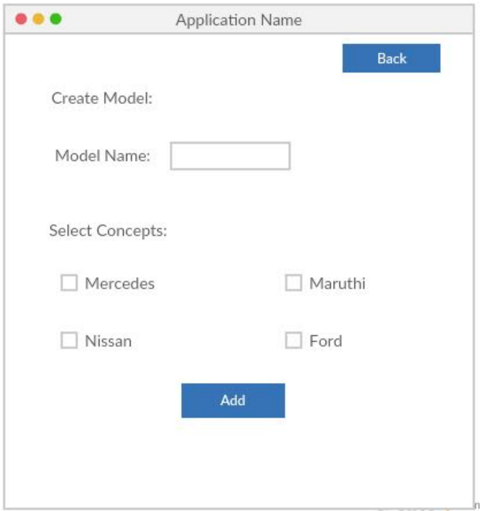










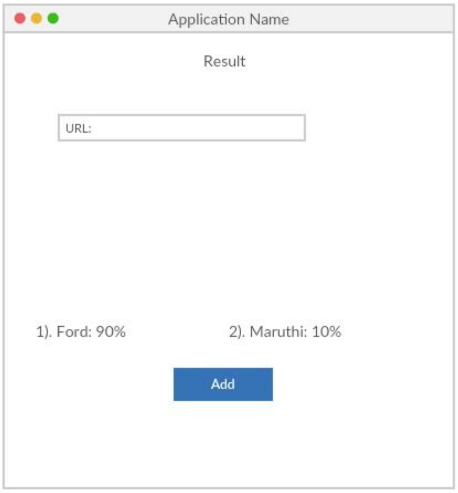


Model Name:

Select Concepts:

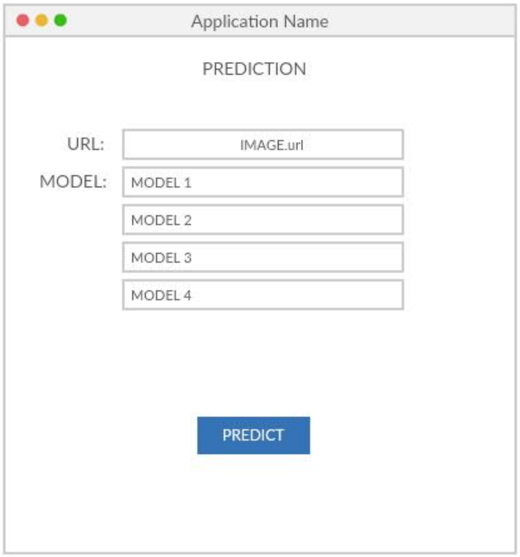
□ Mercedes □ Maruthi

□ Nissan □ Ford



URL:

1). Ford: 90% 2). Maruthi: 10%



URL: IMAGE.url

MODEL: MODEL 1

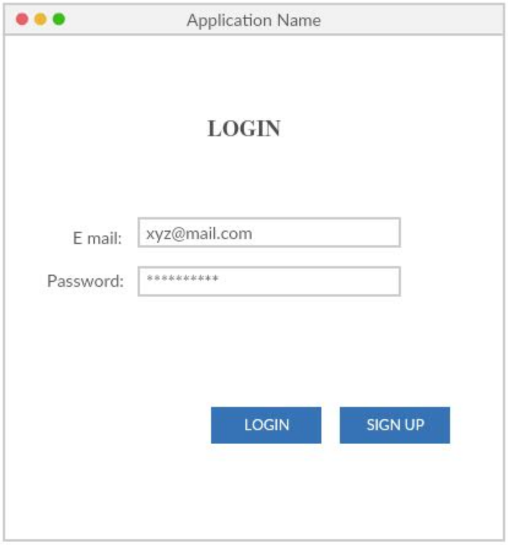
MODEL 2

MODEL 3

MODEL 4

1). Ford

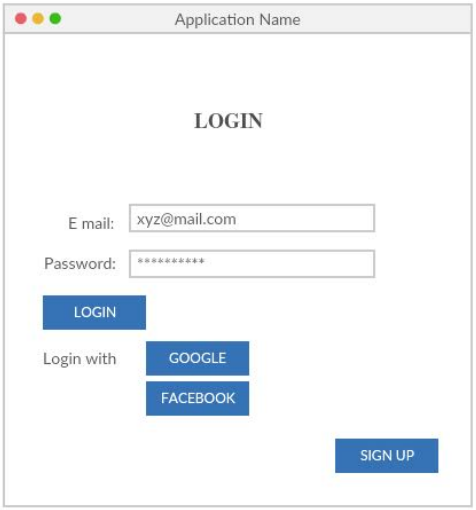
1). Ford



LOGIN

E mail: xyz@mail.com

Password: **********



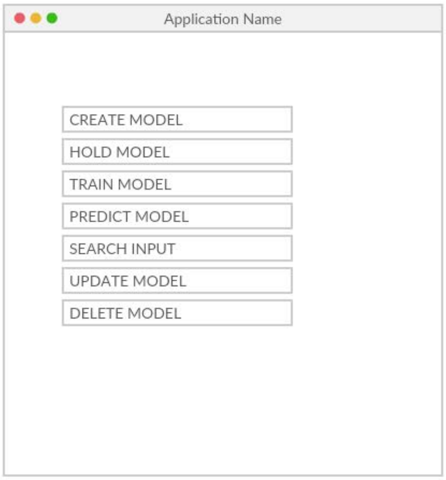
LOGIN

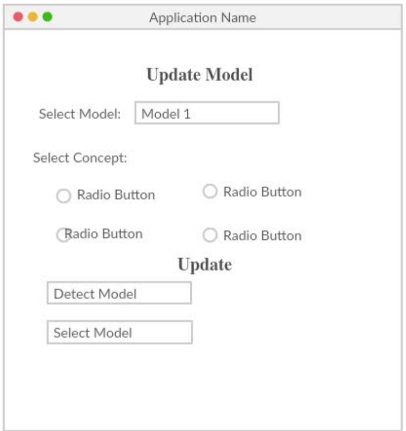
E mail: xyz@mail.com

Password: **********

1). Ford

1). Ford





Update Model

Select Model: Model 1

Select Concept:

○ Radio Button ○ Radio Button

○ Radio Button ○ Radio Button

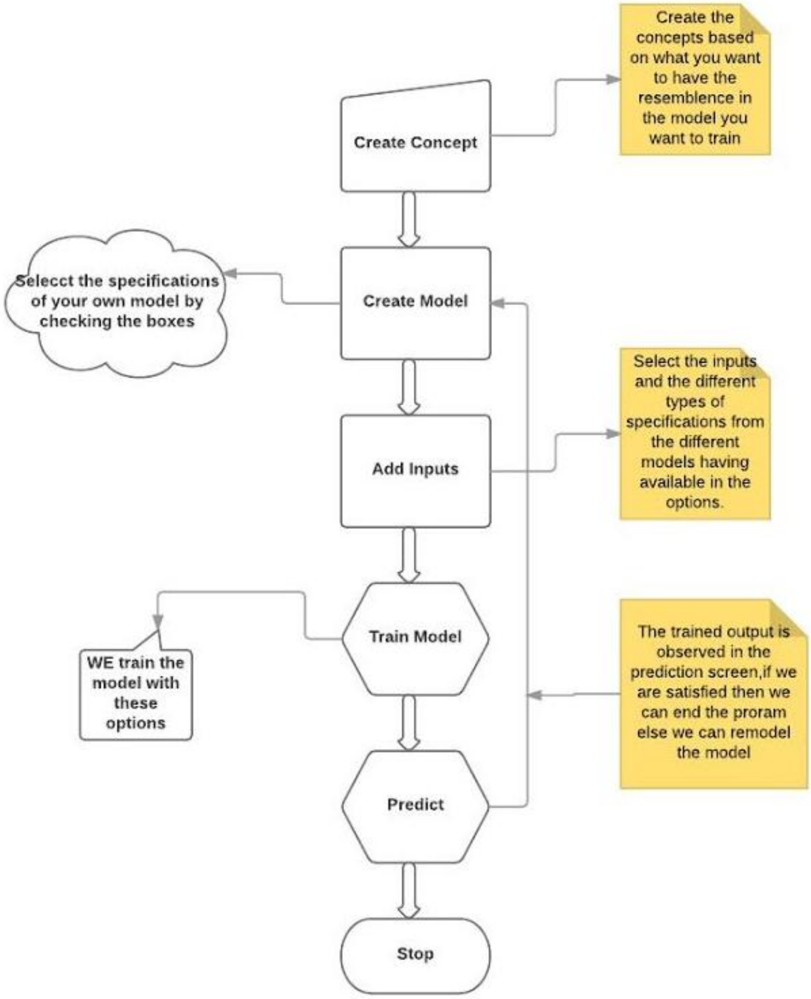
Update

1). Ford

1). Ford

b. UML diagrams

i. Flow chart:


Create Concept
Create the concepts based on what you want to have the resemblence in the model you want to train
Create Model
Seleсct the specifications of your own model by checking the boxes
Add Inputs
Select the inputs and the different types of specifications from the different models having available in the options.
Train Model
WE train the model with these options
Predict
The trained output is observed in the prediction screen,if we are satisfied then we can end the proram else we can remodel the model
Stop

c. User stories

S1. Authentication

The user logs into the system

Cost: low

Risk: high

Value: high

Actors: Clarifai app user

Precondition: the user has not yet entered the system

Description

1. This use case begins when a user opens up the application and is not logged in.
2. The user is prompted to enter a user id and password.
3. If the user's credentials are authenticated, the use case ends in success with the user reaching the application's functions.

Alternative: an invalid user id and password are entered at step 2. An error message is displayed and step 2 is repeated until the use case succeeds to step 3.

Open issues: how many times should they be allowed to enter invalid credentials before being denied access? How would access be reinstated? Logout when the app is closed or after a certain amount of time?

S2. The user takes a photo with their device's camera
Cost: low
Risk: high
Value: high
Actors: Clarifai app user
Precondition: The user has successfully logged in to the app.
Description:

1. This use case begins when a user has successfully logged in to the app.
2. The user is given an option to take a photo with their device's camera.
3. The camera opens up and the user takes a picture.
4. The use case ends when the image is successfully captured.

Alternative: The user could potentially skip taking a photo and instead select an item from the device's image gallery.
Open issues: Add access to camera, storage, and gallery?

S3. The user's image is identified
Cost: high
Risk: high
Value: high
Actors: Clarifai app user
Precondition: The user has chosen an image to identify, either from the camera or gallery
Description:

1. This use case begins when a user has selected an image to identify
2. The image is sent to the Clarifai API and a prediction is returned.
3. The use case ends when the prediction is returned

Open issues: What happens if the image is identified incorrectly? Does the user have the option to give feedback?

S4. The user is given additional info about the image
Cost: high
Risk: high
Value: low
Actors: Clarifai app user
Precondition: the user's image has been identified
Description:

1. This use case begins when a user's image has been identified with a tag.
2. The user is then given additional information with the tag, e.g. a description of the item from Wikipedia or Google Knowledge Graph.
3. The use case ends when the information has been returned. The user is then asked if they would like to try another image.

Open issues: What if the image was identified incorrectly? Do we get this far or is the user asked for a different image?

S5: An administrator adds a new concept to train.
Cost: high
Risk: medium

Value: high
Actors: an administrative user
Precondition: The administrator has selected an option to add a concept
Description:

1. The use case begins when an admin selects an option to add a new concept to train
2. The admin enters a name or label for this concept
3. The admin enters multiple URLs for different images that match the label
4. The use case ends when the images have been used for training and the model is complete

Open issues: is this option available to mobile app users? For any user or just an admin? Or is it only available in a web app?

S6: An administrator creates a model for image recognition
Cost: high
Risk: medium
Value: medium
Actors: an administrative user
Precondition: The admin has selected an option to create a model. The admin has already chosen the concepts for the model.
Description:

1. The use case begins when an admin selects an option to add a model.
2. The admin selects any concepts to be used for the model.
3. The admin selects any configuration options for the model.
4. The use case ends in success when the API develops a model based on the concepts provided.

Open issues: is this option available to mobile app users? For any user or just an admin? Or is it only available in a web app?

S7. An administrator updates a model
Cost: medium
Risk: medium
Value: medium
Actors: an administrative user
Precondition: the admin has selected an option to update an existing model. The admin has a new concept to add or configuration parameters to modify.
Description:

1. The use case begins when an admin selects an option to update a model.
2. The admin selects the existing model to modify.
3. The admin selects to add a concept to the model or alter configurations
4. The use case ends in success when the API updates the model based on the selections.

Open issues: is this option available to mobile app users? For any user or just an admin? Or is it only available in a web app?

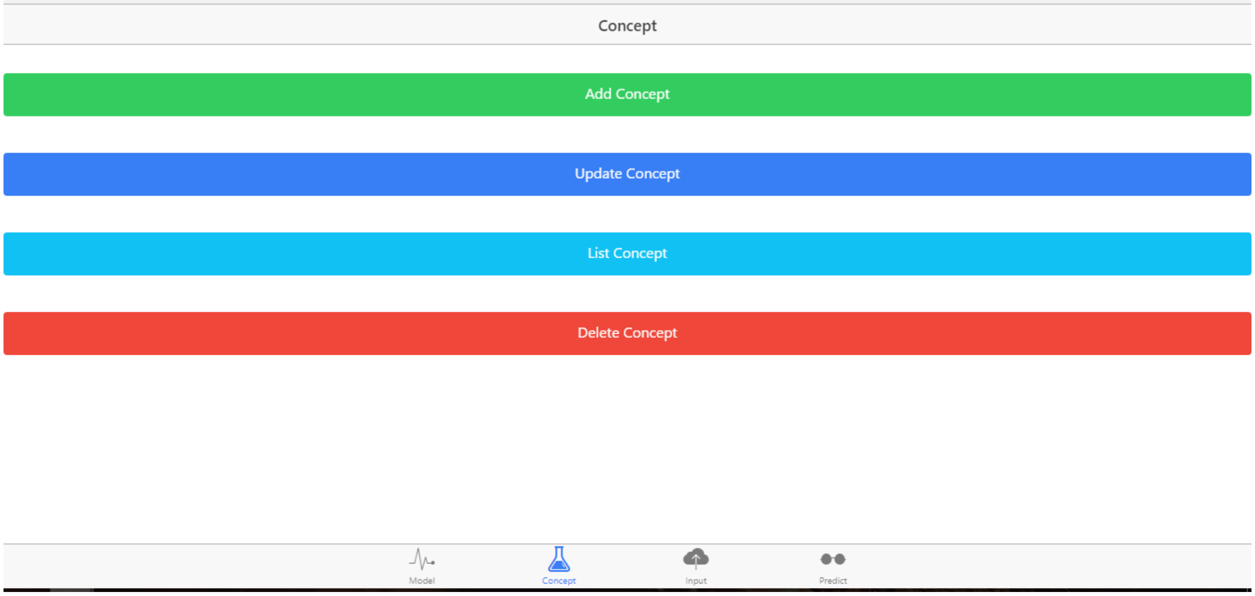
S8. An administrator deletes a model
Cost: low
Risk: low
Value: low
Actors: an administrative user
Precondition: a model exists that the admin wishes to delete.
Description:

1. The use case begins when an admin selects an option to delete a model.
2. The admin selects the existing model to delete.

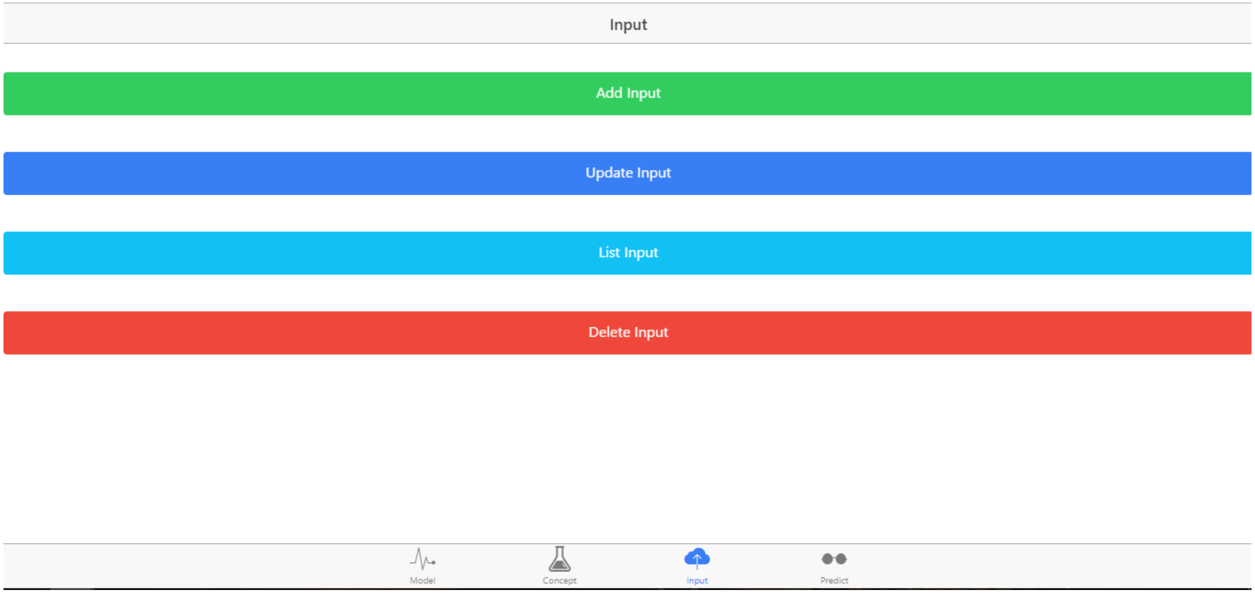
3. The use case ends when the model is deleted from the app. The model can no longer be used to identify user images.

Open issues: is this option available to mobile app users? For any user or just an admin? Or is it only available in a web app?

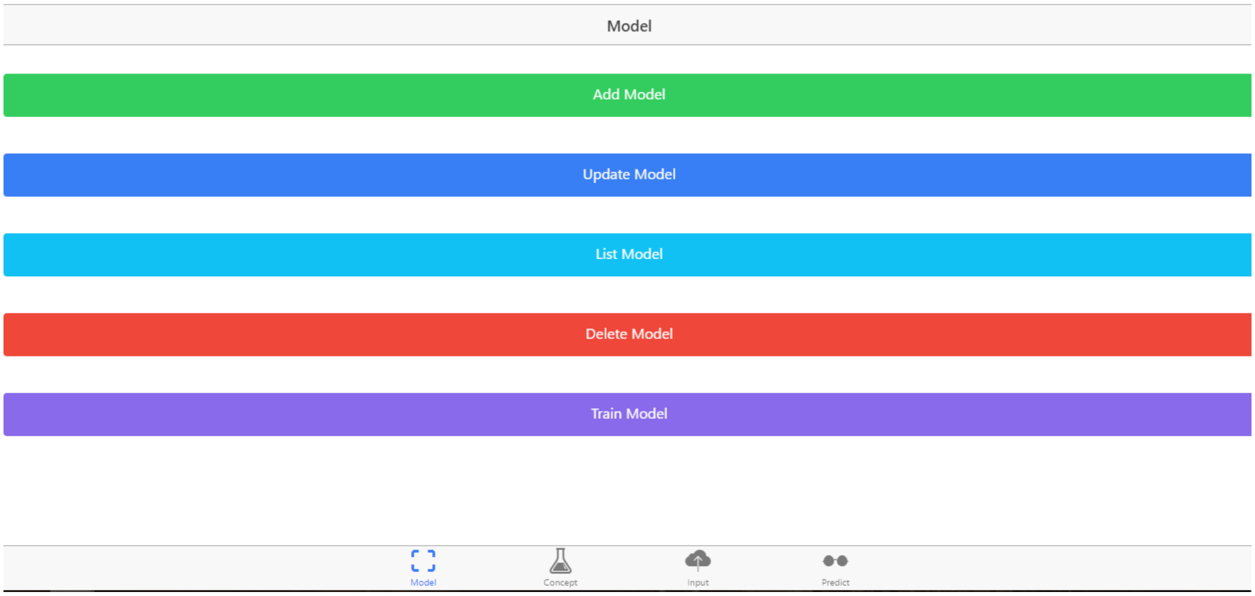
3) Testing: TBD
4) Implementation and Deployment
    i. Concept Screen



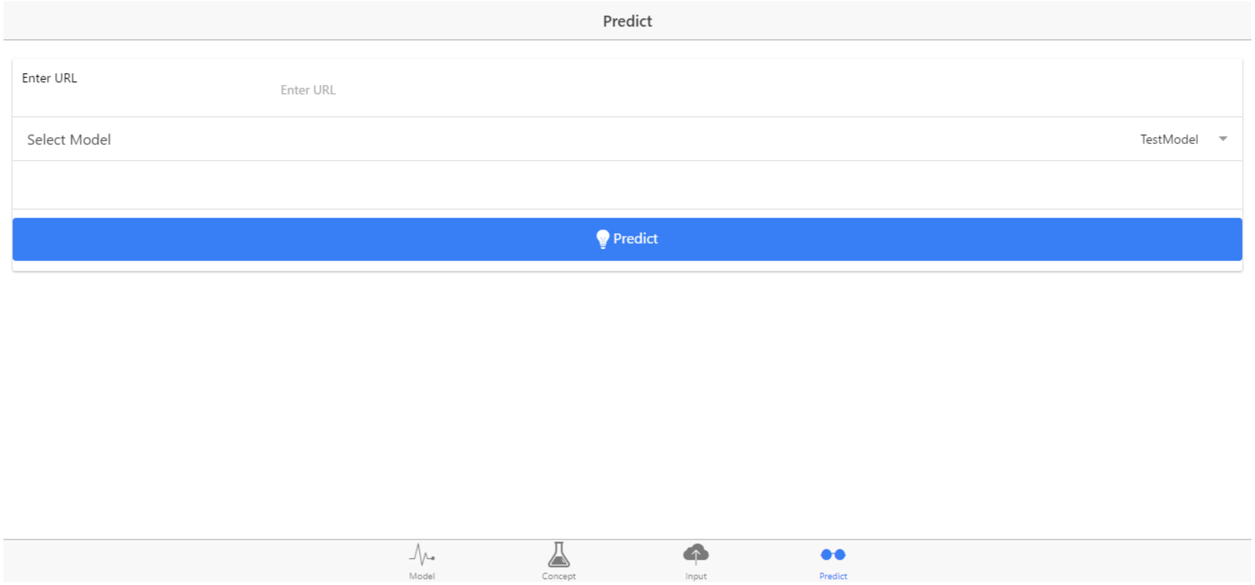
    ii. Input Screen



    iii. Model Screen



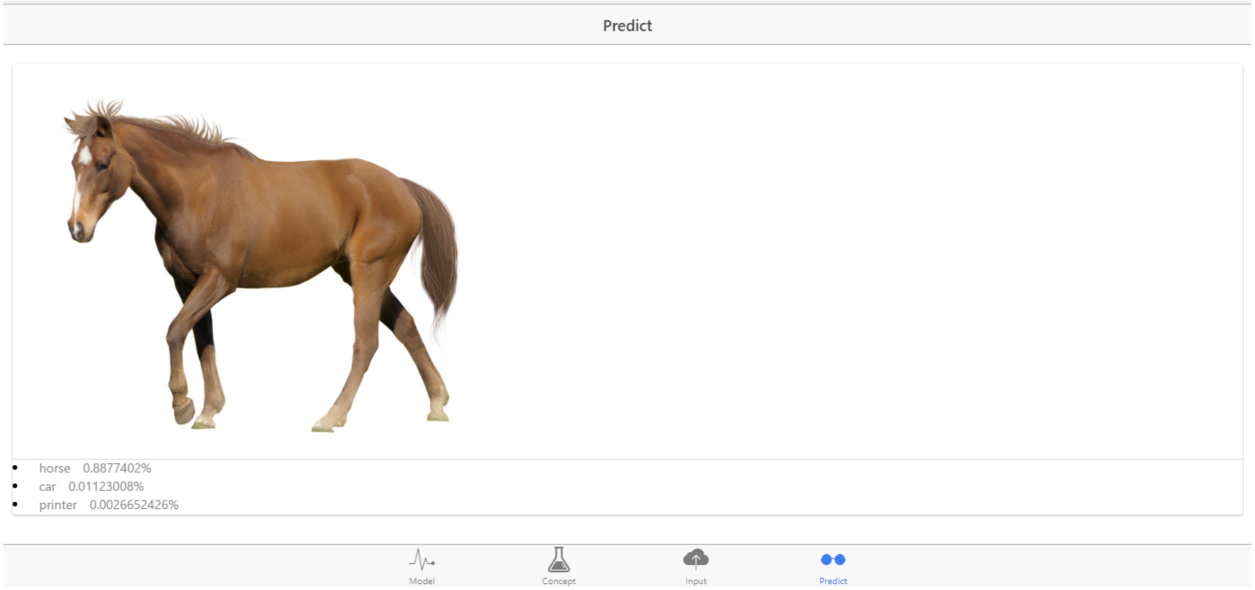
iv. Prediction screen



Enter URL: Enter URL

Select Model: TestModel

v. Final Results



- horse 0.8877402%
- car 0.01123008%
- printer 0.0026652426%

5) Project Management
   a. Work completed:
      i. The hybrid app was developed using Ionic framework and Clarifai API
      ii. Three concepts were trained: horse, car, and printer
      iii. A model was created with Clarifai API
      iv. Results are returned in the app for the entered image
   b. Work to be completed:
      i. Add login to app
      ii. Let users submit own photos through camera or gallery
      iii. Add new concepts to match user theme
      iv. Improve overall UI
      v. Testing
   c. Concerns: getting the app finished in time after starting project from scratch each iteration.

V. ITERATION 4 REPORT

1) Services/APIs used

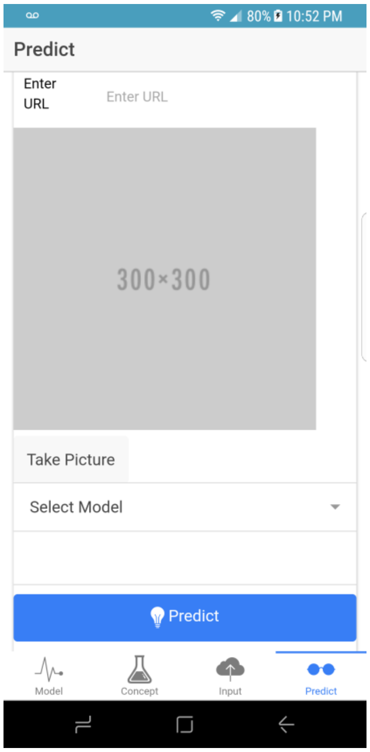
   We added functionality that uses the camera for the device to send an image to the Clarifai API. We also utilized the Google Charts API to generate a bar chart showing the confidence level of the results.

2) Implementation: uses Ionic

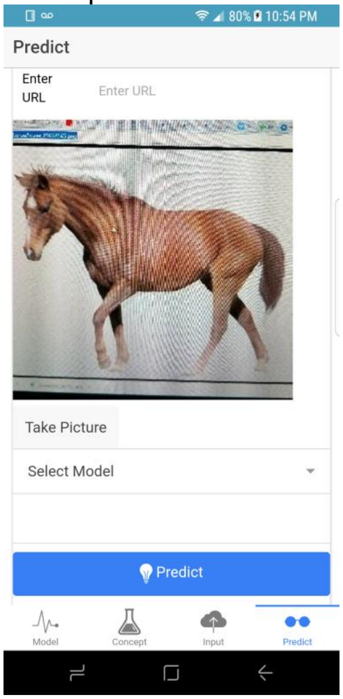
   Source code: https://github.com/shalin51/Hunter/tree/master/Source

3) Deployment: Updated screenshots of deployment to mobile device

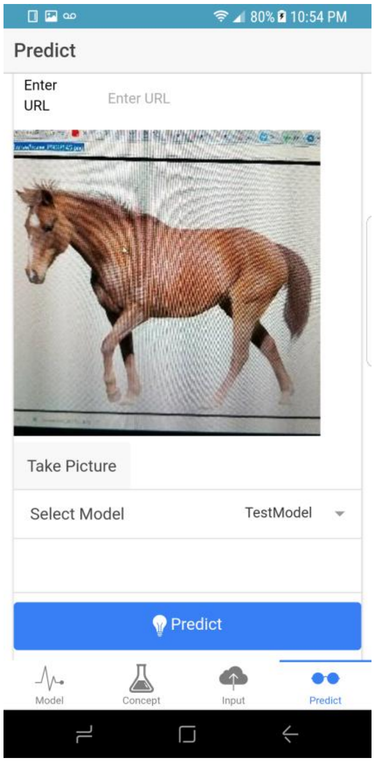
   There is now an option to either enter a URL or take a photo under the prediction tab:



After the Take Picture button is pressed, the devices camera will open. On my device, I took a photo of a horse on my computer screen. It is then displayed in the box:

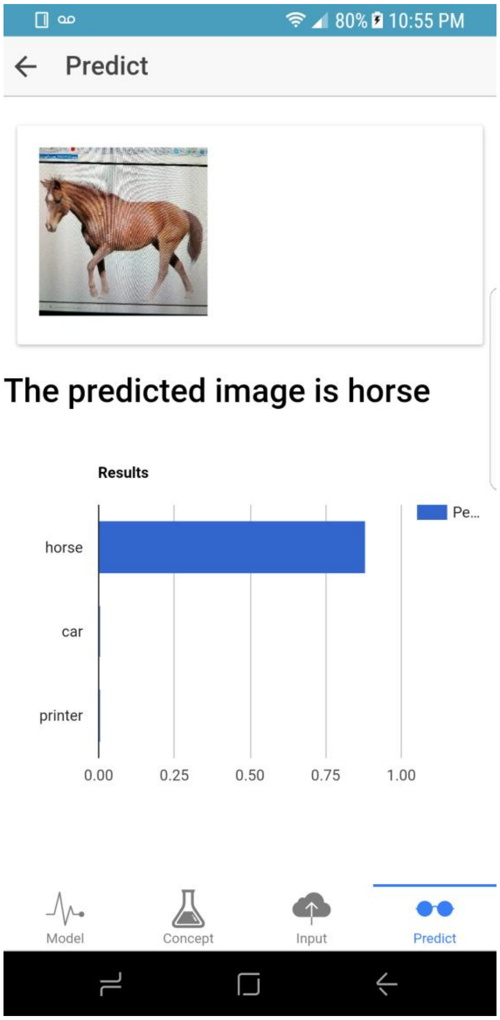


The model is then selected:



Select Model TestModel

After the predict button is pressed, we see the image followed by a prediction of what the image is. Then we see a graph showing the confidence level of the prediction relative to other options:



Predict

The predicted image is horse

4) Project Management
   a. Work completed:
      i. Users can enter their own photo for prediction
      ii. The top prediction is shown
      iii. A chart showing confidence levels is shown
      iv. Troubleshooting with the model/concept functionality
   b. Work to be completed:
      i. Finish bug fixes
      ii. Improve UI
      iii. Make video
      iv. Make ppt presentation

USER MANUAL

Table of Contents

Introduction

This project will be to create a Native/Web hybrid mobile application in which a user can take a picture and get an identification and information about what the image is. The project will involve training and testing of images using the Clarifai open API for deep learning purposes. The user will be able to create their own models for image recognition to match their own needs.

How to use this application

1. When the user opens the application, they will see four possible tabs at the bottom of the screen, labeled Model, Concept, Input, and Predict. They are defined as follows:
   a. Model: a general model consists of different categories for image recognition. The model generally captures a theme of images. For example, a model might be called foods that contains categories like apple, broccoli, sushi, pizza or a model called animals might contain categories like cat, dog, horse, and cow.
   b. Concept: these are the categories of images that make up the model. For example, a concept might be called Cat. This concept might be added to the model called Animals. It could also be added to other models, like Pets or a General model that includes many concepts.
   c. Input: an input is an image associated with a concept that is later used to train a model. For example, a user can upload and input of an image of a cat, and associate this with the Cat concept that is later used to train the animal model.
   d. Predict: this tab is used to make final predictions for an unlabeled image that the user wants the model to identify.
2. The user has two options: they can either immediately start predicting images based on existing models, or they can train add concepts and inputs and train their own models for use. Beginning with the latter, the user first should add some concepts and thus clicks on the Concept tab.
3. The Concept tab includes buttons to Add a Concept, Update a Concept, List Concepts, and Delete a Concept. These options enable CRUD operations on the Concepts in the database. First, the user should create some concepts by clicking the Add a Concept button.
4. The user is then given a box to enter in the name of the Concept and press Submit. After pressing Submit, the user is given a confirmation dialog followed by an alert that the concept was added.

The user can continue entering as many concept names as they'd like. The next step is to either add inputs or create a model (order doesn't matter). First, we will add inputs.

5. After selecting the Inputs tab, the user should select to add Inputs (they can also view and delete inputs here). To add an input, the user is given options to either upload an image file from their computer or enter a URL.
6. After the input is entered via file upload or URL, the user can select the concept that this input is associated with. The list of concepts includes the concepts that were added in step 4. For example, if we added an image of a cat, we would choose the Cat Concept and select submit to add the input.
7. After all inputs are added for the Concepts (a minimum of 10 images per concept is recommended for the Clarifai API), the next step is to create the model, so the user selects the Model tab.
8. The Model tab includes buttons to Add a model, Update a Model, List Models, Delete a Model, and Train a Model. These options enable CRUD operations on the models in the database and the training of a new model. First, the user may want to add a model.
9. After clicking on the Add a Model button, the user can enter in a Model name. On pressing Submit, the user is given a popup to confirm and then affirm that the model has been created.
10. The next step is to train the model. The user selects this button and selects the model to train from the available list. The user should be sure that all concepts and inputs desired have been entered for this model before training it. The user is then alerted of success in training the model.
11. The model is now ready for prediction! The user selects the Predict tab. From here, the user can either enter an image URL that they would like to identify, or they can elect to take a picture with their device. Since this is a hybrid app, if the user is using it on the web, they will not be able to take a picture. If on a mobile device, the user can press the button to take a picture. After taking a picture and confirming it, they can see the image that will be predicted.
12. The user must select a model from the list of available models and press the predict button. After a short time, the predicted label for the image is displayed, along with a chart displaying the confidence of the prediction relative to the other concepts available in that model.
13. The idea of the app is for the user to create their own models to use as they see fit. The user can create and delete models to use for teaching students or training employees, for example.

Error Recognition

If the app incorrectly identifies the image, there is no error reporting. A future feature might be to enable a user to give feedback on predicted images in order to improve the model.

Sample Interactions

For a clear step by step demo of the features of the application, please view our youtube video here. The demo shows a mobile application on an Android device, but this is a hybrid app that is also usable on the web (without picture taking capability) or on iOS or Windows phones.

Known Bugs

One known limitation of the app is that our free Clarifai subscription allows only 10 concepts and limited interactions. Model creation is not limited except by the operation limit.

PROJECT MANAGEMENT

Project Management

Link to our agile site on github
For screenshots on our agile development, see the section of this report starting on page 7.

Project Evaluation

1. In the end, the project came together well , though it was difficult for us to all work together on the same schedule with vastly different skill sets.
2. A major issue was that we started off too ambitious. We changed project topics three times and wasted a lot of time working on two different projects that did not come to fruition, detailed at the beginning of this report. We started with the prospect a Google Home application on the smartphone, but discarded this after Iteration 1 and the TA feedback. Our second app was to use Google Cardboard and voice controls to run an application, but discarded this again after the second iteration and TA meeting. Our third project stuck, but was largely selected by one group member. Since two iterations were already spent, we did not have as much time to work on this part of the app as we would have liked.
3. The design process was not very clear but followed the Agile process of incremental development and redevelopment.
4. The agile process worked well and kept us productive at each iteration. Our major issue was changing projects for each of the first three iterations. For next time, I think the TA meetings should be much sooner, because we often didn't receive feedback until well into the next iteration.
5. We did not stick to our original planned schedule because we changed projects several times. It was sometimes difficult to meet in a group or even meet with the TA. We used the WhatsApp application to keep in touch, though it wasn't always effective.
6. We were primarily managed by Shalin, who developed our final project idea. This structure worked fine, though it would have been better if everyone had been able to contribute equally. In a real world project, team members would have diverse skill sets that each enabled them to contribute to a project in an individual way, meetings and updates are more regular, and individuals are responsible for their own work. This is not usually the case with student team projects.
7. For next year, the major suggestion is to give students better ideas for projects and scope. We did not get great TA feedback and floundered on several projects before selecting one that worked.

FINAL PROJECT POWERPOINT

MACHINE LEARNING WITH CLARIFAI API

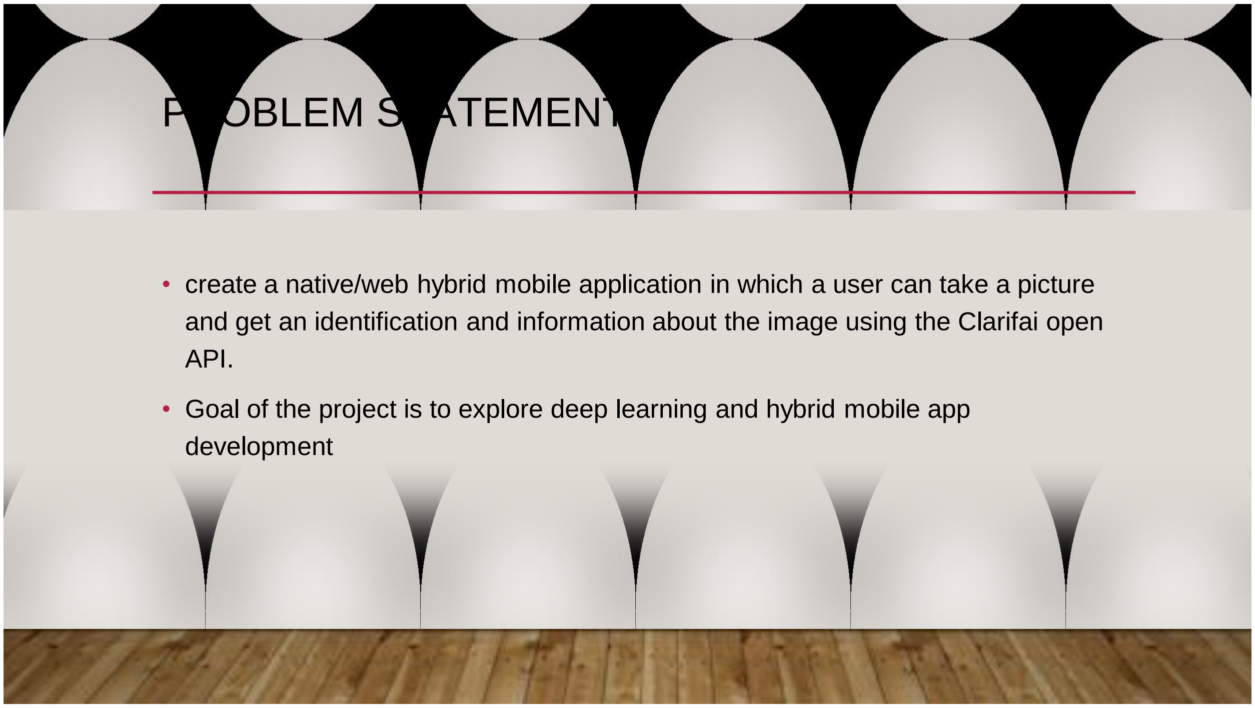
Team -4

Shalin Patel

Jackie Batson

Sandeep pabolu

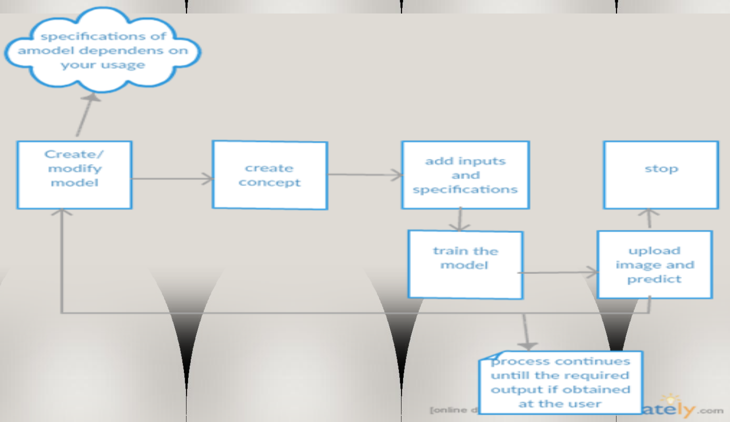
Akhila katta



PROBLEM STATEMENT

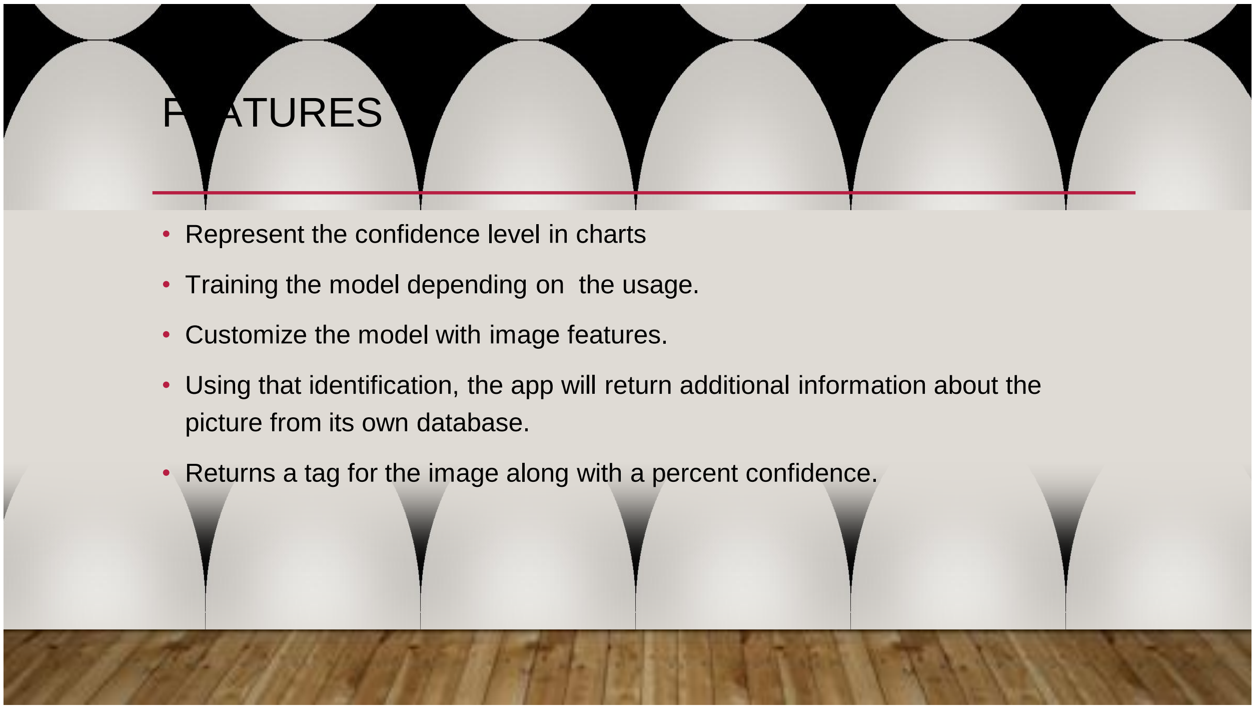
- create a native/web hybrid mobile application in which a user can take a picture and get an identification and information about the image using the Clarifai open API.
- Goal of the project is to explore deep learning and hybrid mobile app development

SOFTWARE WORK FLOW


specifications of amodel dependens on your usage
Create/ modify model
create concept
add inputs and specifications
stop
train the model
upload image and predict
process continues untill the required output if obtained at the user
[online d
ately.com

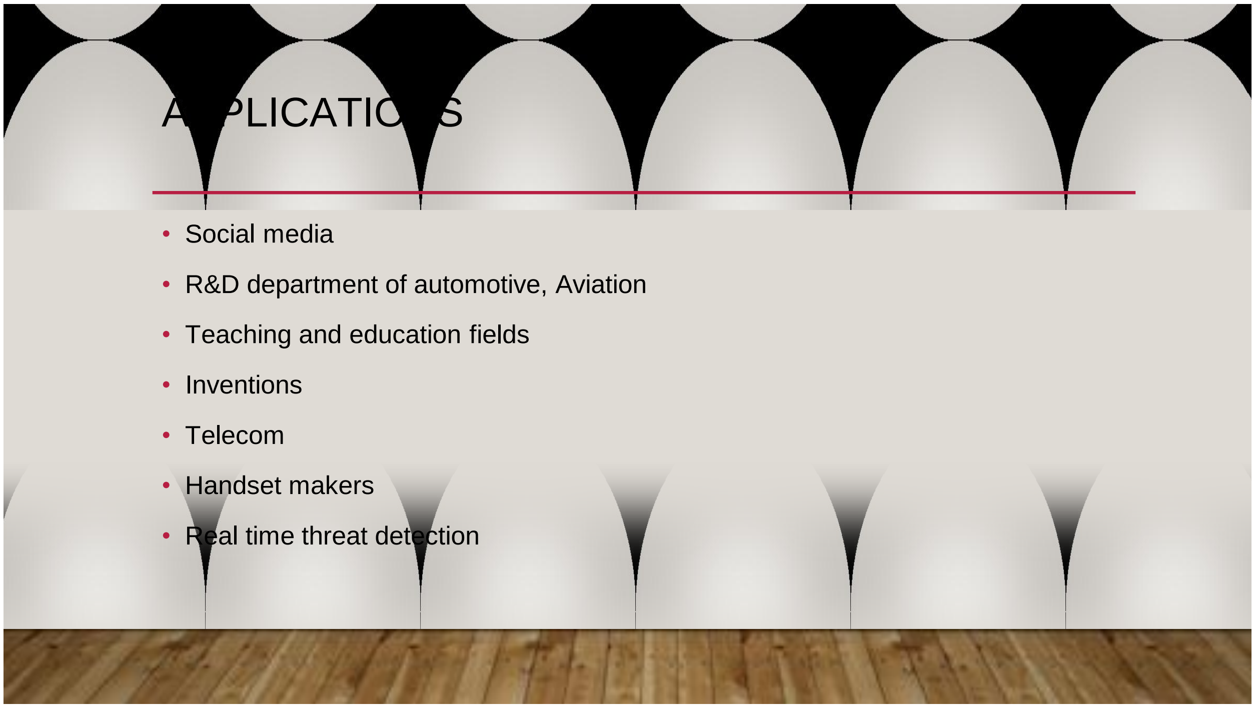
TECHNOLOGIES

- Angular js
- Firebase
- Visual recognition with Clarifai API
- HTML, CSS
- Bootstrap
- Ionic framework
- Google charts API



FEATURES

- Represent the confidence level in charts
- Training the model depending on the usage.
- Customize the model with image features.
- Using that identification, the app will return additional information about the picture from its own database.
- Returns a tag for the image along with a percent confidence.



APPLICATIONS

- Social media
- R&D department of automotive, Aviation
- Teaching and education fields
- Inventions
- Telecom
- Handset makers
- Real time threat detection

ALL LINKS

Source code and documentation for projects 1 and 2
Source code and documentation for final project
YouTube video of final project