­­Cairo University

Faculty of Computers and Artificial Intelligence

Department of Computer Science

Phone Finder

Dr. Amr Badr

Dr. Eman Hosney

Implemented by



*1st 20160086 1st Gomaa Abied Mohamed*

*2nd 20160170 2nd Amr Mohamed Fahmy*

*3rd 20160146 3rd Ezz Asaad saad*

*4th 20160060 4th Alaa El-menshawy Abd ElFtah*

Graduation Project

Academic Year 2019-2020

Final Documentation

**Contents**

[Chapter 1: Introduction: 4](#_Toc48157953)

[1) Motivation & Problem definition: 4](#_Toc48157954)

[2) Objective: 6](#_Toc48157955)

[3) Gantt Chart: 7](#_Toc48157956)

[4) Development Methodology: 8](#_Toc48157957)

[5) Tools and SW: 10](#_Toc48157958)

[6) Organization Report: 12](#_Toc48157959)

[Chapter 2: Related work: 13](#_Toc48157960)

[**Chapter 3: System analysis:** 15](#_Toc48157961)

[3.1- Project Specification: 15](#_Toc48157962)

[3.2- use-case diagram:- 17](#_Toc48157963)

[3.3-Use-case Description: 17](#_Toc48157964)

[3.4 – System test cases: 24](#_Toc48157965)

[Chapter 4: System Design: 34](#_Toc48157966)

[1-System component: 34](#_Toc48157967)

[2 – System class Diagram: 35](#_Toc48157968)

[3- Sequence Diagram: 36](#_Toc48157969)

[4- Project ERD: 40](#_Toc48157970)

[5 - System GUI Design: 41](#_Toc48157971)

[Chapter 5: Implementation: 48](#_Toc48157972)

[References: 52](#_Toc48157973)

**Table of figures:**

[Figure 1 7](#_Toc48157974)

[Figure 2 7](#_Toc48157975)

[Figure 3 7](#_Toc48157976)

[Figure 4 7](#_Toc48157977)

[Figure 5 7](#_Toc48157978)

[Figure 6 8](#_Toc48157979)

[Figure 7 9](#_Toc48157980)

[Figure 8 17](#_Toc48157981)

[Figure 9 30](#_Toc48157982)

[Figure 10 30](#_Toc48157983)

[Figure 11 31](#_Toc48157984)

[Figure 12 31](#_Toc48157985)

[Figure 13 32](#_Toc48157986)

[Figure 14 32](#_Toc48157987)

[Figure 15 33](#_Toc48157988)

[Figure 16 34](#_Toc48157989)

[Figure 17 35](#_Toc48157990)

[Figure 18 36](#_Toc48157991)

[Figure 19 37](#_Toc48157992)

[Figure 20 38](#_Toc48157993)

[Figure 21 39](#_Toc48157994)

[Figure 22 40](#_Toc48157995)

[Figure 23 41](#_Toc48157996)

[Figure 24 42](#_Toc48157997)

[Figure 25 42](#_Toc48157998)

[Figure 26 43](#_Toc48157999)

[Figure 27 44](#_Toc48158000)

[Figure 28 44](#_Toc48158001)

[Figure 29 45](#_Toc48158002)

[Figure 30 45](#_Toc48158003)

[Figure 31 46](#_Toc48158004)

[Figure 32 46](#_Toc48158005)

[Figure 33 47](#_Toc48158006)

[Figure 34 48](#_Toc48158007)

[Figure 35 49](#_Toc48158008)

[Figure 36 50](#_Toc48158009)

[Figure 37 51](#_Toc48158010)

**Table of Abbreviations:**

|  |  |
| --- | --- |
| KNN | K nearest neighbor |
| NLP | Natural language processing |
| ML | Machine Learning |
| JS | Java script |
| DL | Deep Learning |

## Chapter 1: Introduction:

### 1) Motivation & Problem definition:

* To help the user or store owner to find mobile with his preference faster and its store which sold it , by giving our web application his preference features such as ( battery , camera , size ,…) then it will recommend him with lots of mobile that near to this features.
* When users need or prefer offers and instead of the user take more time to compare stores offers or even find stores which sell by offers our web application will get recommend shops for users, by showing them mobile store posts which contain the mobile photo, its price and offers if there is.
* To help user or store owner to predict price for his specific features which he wanted by giving our web application his features and it will give him the price.
* When some users need to buy a new mobile so they will have to compare among different mobiles so our web application get their features faster by take a photo for each mobile and then get all its features so the user won’t take time to take his decision.
* Some people like the mobile's exterior but they don’t know their features and names so they can’t search on internet if they didn’t know the name so our web application help them by taking a photo for this mobile then our application will get all their features .
* When the user wants to fix his mobile on trusted store and communicate with trusted people so we help him to find the trusted stores by giving rate to them by users, when the mobile store finishes fixing user’s mobile the user take him rate to help other people to Judge this store.
* To help mobile store owners to get clients which not on its local area but also in any other place by make posts of its mobiles and offers if there is.
* When mobile goods reached the store owner it will take much more time to get all features of each mobile so our web application help them to get report of goods and this report contains each mobile features.

### 2) Objective:

**Phone Recognition:**  
 - Building Deep Learning (DL) Model to Recognize the mobile brand and name.

**Recommendation System:**  
 - Designing Machine Learning (ML) model to recommend the best Store and mobile according to User Preferences.

**Software community:**  
 - Developing Software community to communicate between Mobile stores and customers.

**Cost:**

* Our application depending on using mobile or laptop or pc so it does not contain labor cost.
* Our application saves the customer the effort and time of getting to the store or looking for mobile store to buy new mobile.
* Our application saves the store owners from costing of the manual advertisement by making the posts on our.

**Performance:**

- Taking less response time for making any of the application’s functions.

**Usability:**

- Our application is very easy to use and learn.

### 3) Gantt Chart:

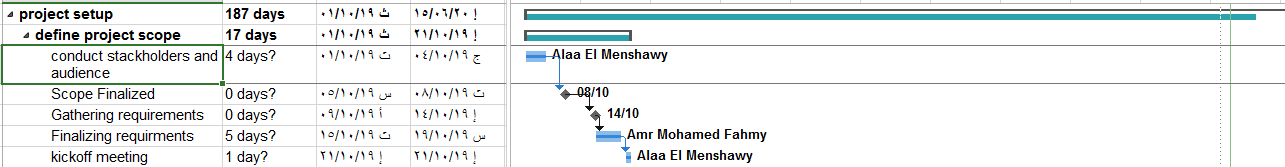
****

Figure 1

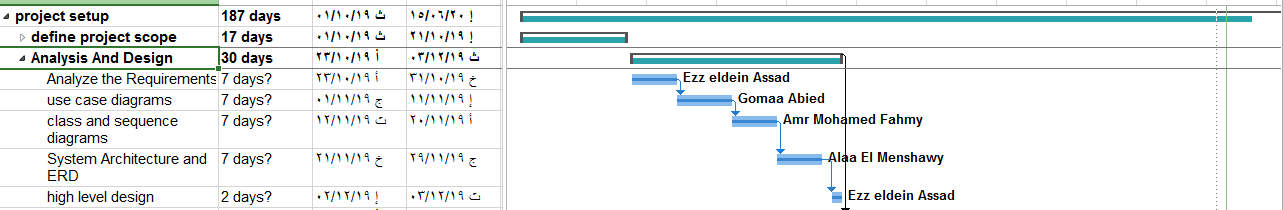
****

Figure 2

****

Figure 3

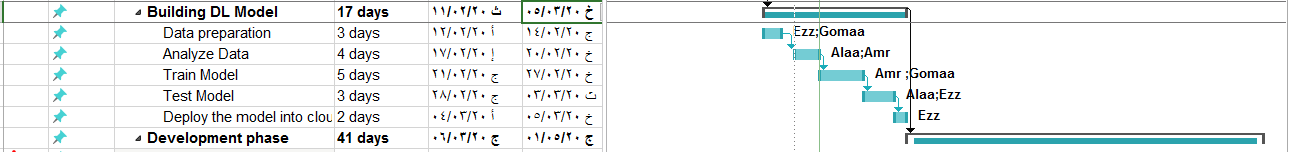
****

Figure 4

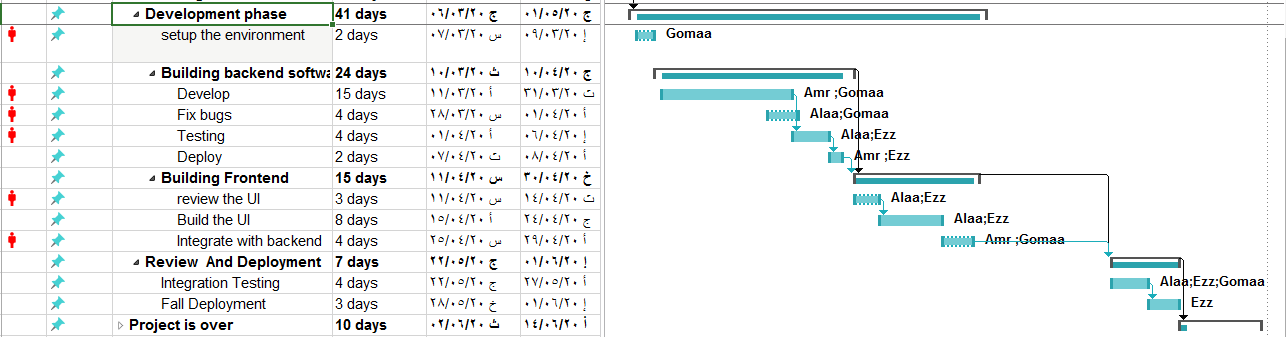
****

Figure 5

### 4) Development Methodology:

**Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

**System Design:** The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

**Implementation:** With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

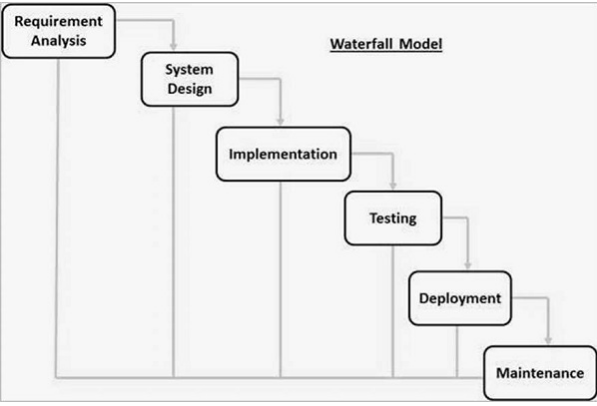


Figure 6

**Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

**Deployment of system:** Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

**Maintenance:** There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

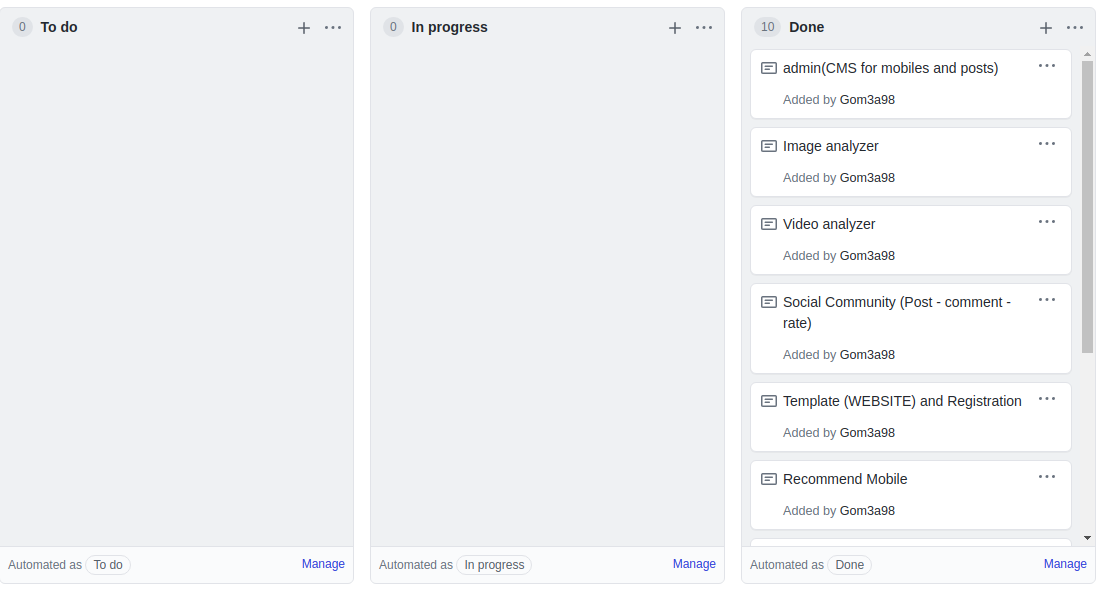
****

Figure 7

### 5) Tools and SW:

**Anaconda platform:**

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment

**- Keras:**

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs.

**- Tensor flow:**

An end-to-end open source machine learning platform for everyone. Discover Tensor Flow's flexible ecosystem of tools, libraries and community resources.

**- Yolo:**

Keras application for Object Detection.

**- Flask:**

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions

Python language.

**- Google maps API:**

Google Maps is a web mapping service developed by Google. It offers satellite imagery, aerial photography, street maps.

**- Google Drive:**

Google Drive is a file storage and synchronization service developed by Google

**- MYSQL:**

MySQL is an open-source relational database management system.

**- Google co-lab environment:**

Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML and more.

**- ReactJS:**

Declarative React makes it painless to create interactive UIs.

**- JS**

**- JQUERY:**

JQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation.

**- Axios:**

Axios is an easy to use HTTP client. Learn how to use it within your React applications.

**- Bootstrap:**

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.

**- UI Material:**

React components for faster and easier web development. Build your own design system, or start with Material Design.

### 6) Organization Report:

**Abstract and Introduction:**

The main idea is intelligent assistant that provides capability of recommending mobile with specific features, compare among mobiles and choose the best, provide mobile community to communicate users with store owners, getting mobiles features faster through taking a video for these mobiles.

**Motivation and Performance:**

* Try helping user and store owner getting the mobile which they prefer by recommending mobiles with user’s features then user can choose from them his prefer mobile.
* Helping users and store owner to compare among several phones and choose the best for them.
* Help users to fix its phones in trusted shop.
* To help the mobile store owners making advertisements by posting mobiles.
* The user can be recommended by stores with rate which help him to take decision of dealing with this shop or not.

**Conclusion:**

* Finally we wish to build safe community among shops.
* To trust user the store which deal with.
* Facilitating in recommending mobile to user by some features.
* Guiding users to get their prefer mobile.

## Chapter 2: Related work:

|  |  |  |
| --- | --- | --- |
| **Website** | **Its features** | **Our Improved features** |
| [https://www.gsmarena.com](https://www.gsmarena.com/samsung-phones-f-9-0-p2.php) | this website has the following features:   * Comparing among mobiles statically. * Searching for mobile information of all brands over the world. * Doesn’t have capability of estimating mobile price based on some features. | So our application use machine learning and deep learning to provide intelligent guide to select specific phone, recommend some phone based on selected feature, model to provide capability of estimating mobile price based on some features. And NLP model to compare among mobiles and determine which the best.  So we get the mobile dataset which contains mobile information. |
| <https://www.etsy.com/> | Etsy’s mobile site configures nicely to both smartphones and tablets.  It provides a prominent search bar and a button to open their App, should you wish to do your shopping there. They also have implemented a popular grid format with photos that many mobile sites use to ensure the display is clean and simple to use.  Can search for stores that sell mobiles with offers. | - Recommend mobiles easily with user’s features, by let the user put his features and our application use machine learning algorithm (cosine similarity).  - Recommend stores according to distance or rate or both, let the user choose the rate or choose his location to get stores by using machine learning algorithm (KNN).  - User can get all mobile features easily by send photo for the mobile to our web application then user get all its features. |
| <https://www.kimovil.com/en/> | - Compare price, specs, reviews and benchmarks for smartphones. Find the perfect mobile at the best price from thousands of web pages. | - User or store owner can know the mobile price by his features, let the user put his features and use machine learning algorithm (logistic regression).  - User or store owner can compare between two mobiles by getting user’s photo for each mobile then get all features for each mobile. |

# **Chapter 3: System analysis:**

### 3.1- Project Specification:

#### 3.1.1- Functional requirements:

**3.1.1.1 - Detects the mobile price:**

Taking user features such as (battery - HW - width - height) and detect its price according to these features.

**3.1.1.2 - Compares among mobiles according to specific features:**

Taking user features which wants to compare with and taking a photo for each mobile then users gets all features of each mobile to take decision of which is the best of them.

**3.1.1.3 - Detects of mobile features:**

Taking mobile photo from user and user will get its features and its global rate.

**3.1.1.4 - Detects mobiles characteristics of video of mobile goods:**

Mobile store owner will get list of mobiles names and these amount on goods by take video of goods.

**3.1.1.5 - Gives rate to mobile stores to repair or sell the mobiles:**

User will get list of mobile stores according to smallest distance or largest rate or both.

**3.1.1.6 - Comments on stores posts:**

User can comment on work of this mobile store either on selling mobile or repairing mobile to help another users.

**3.1.1.7 - Shares offer and discounts of mobile stores:**

Mobile store owner can make advertisement to help user to find suitable mobile, it is like a community between mobile stores and users.

#### 3.1.2- Nonfunctional Requirements:-

**3.1.2.1 - Usability:**

3.1.2.1.1 - Efficiency of use: the average time it takes to accomplish a user’s goals will be very small.

3.1.2.1.2 - Intuitiveness: its interface is very simple for user to deal with.

**3.1.2.2 - Security:**

Ensure that the software is protected from unauthorized access to the system and its stored data. It considers different levels of authorization and authentication across different users roles , the user should have account to login by it and the only person have a right to update this software or deal with data is the admin.

For instance, data privacy is a security characteristic that describes who can create, see, copy, change, or delete information.

**3.1.2.3 - Reliability:**

The database update process must roll back all related updates when any update fails.

**3.1.2.4. Scalability:**

The website attendance limit is scalable enough to support 200,000 users at a time.

**3.1.2.5. Performance:**

The front-page load time is no more than 2 seconds for users that access the website using an LTE mobile connection.

### 3.2- use-case diagram:-

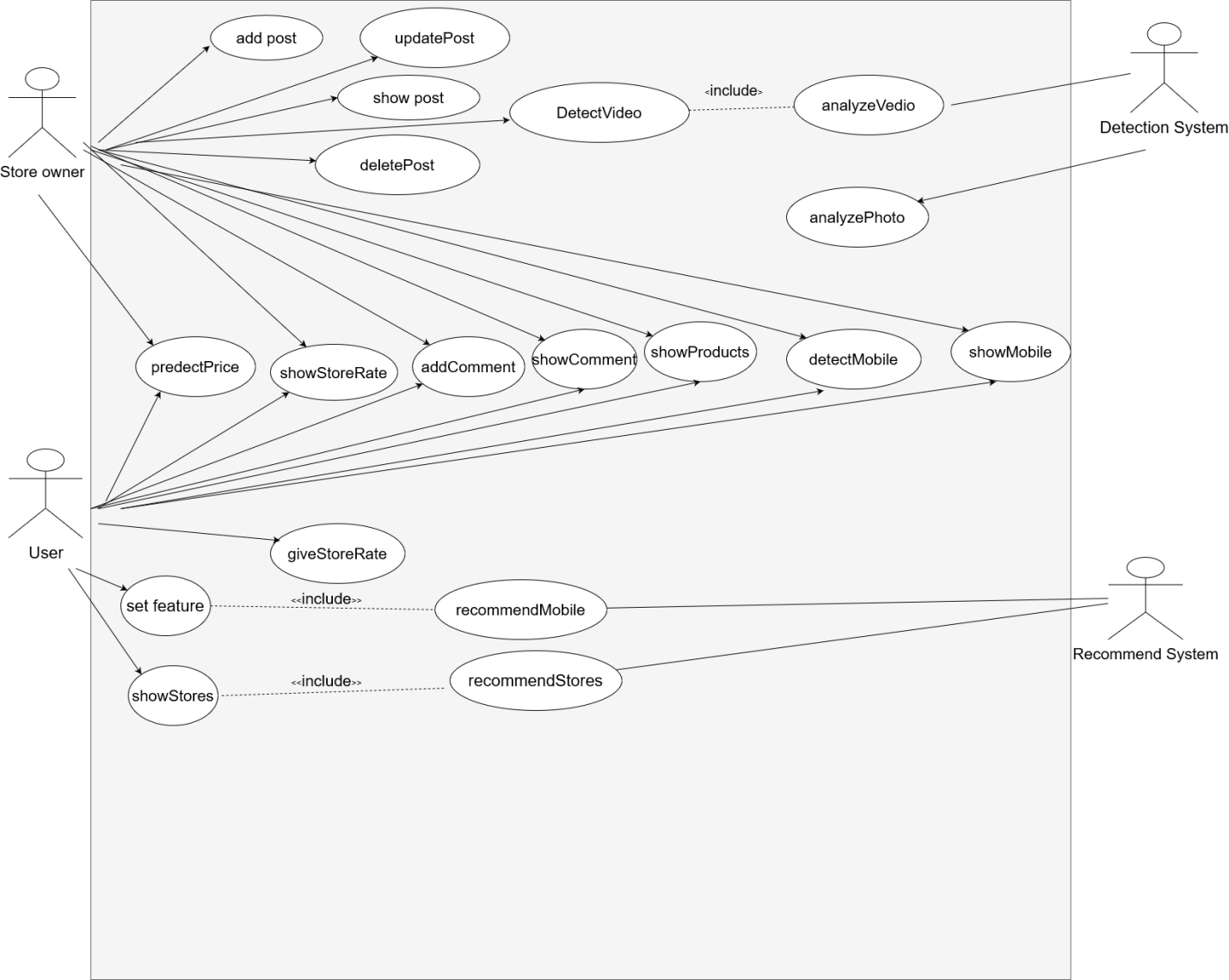


Figure 8

### 3.3-Use-case Description:

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Add post | |
| Actors: | Store Owner | |
| Pre-conditions: | Store owner will Register and have a session | |
| Post-conditions: | Have store profile | |
| Flow of events: | **User Action** | **System Action** |
| 1- Store owner will fill the fields and upload photo. |  |
|  | 2- System will check the data then store it in database. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Predict price | |
| Actors: | User , store owner | |
| Pre-conditions: | User or store owner will register on the system | |
| Post-conditions: | Have form to predict price | |
| Flow of events: | **User Action** | **System Action** |
| 1- The user fill the fields by his features to predict price. |  |
|  | 2- System will check the data then predict price. |
| Exceptions: | **User Action** | **System Action** |
| 1- There is invalid data. |  |
|  | System will send to user to enter valid data. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Recommend Mobile | |
| Actors: | User , store owner | |
| Pre-conditions: | User or store owner will register on the system | |
| Post-conditions: | Have form to recommend mobile | |
| Flow of events: | **User Action** | **System Action** |
| 1- The user fill the fields by his features to recommend mobile. |  |
|  | 2- System will check the data then recommend mobile. |
| Exceptions: | **User Action** | **System Action** |
| 1- There is invalid data. |  |
|  | System will send to user to enter valid data... |

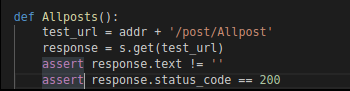
|  |  |  |
| --- | --- | --- |
|  |  | |
| Use Case ID: | 4 | |
| Use Case Name: | Recommend Shop | |
| Actors: | User , store owner | |
| Pre-conditions: | User or store owner will register on the system | |
| Post-conditions: | Have form to recommend shop | |
| Flow of events: | **User Action** | **System Action** |
| 1- The user will choose his the target location and rate for the store. |  |
|  | 2- System will check the data then recommend shop by rate, distance or both. |
| Exceptions: | **User Action** | **System Action** |
| 1- There is invalid data. |  |
|  | System will send to user to enter valid data. |

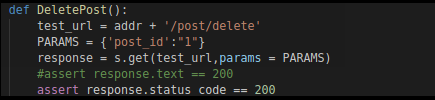
|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Detect mobile | |
| Actors: | User , store owner | |
| Pre-conditions: | User or store owner will register on the system | |
| Post-conditions: | Have form to upload photo to detect mobile | |
| Flow of events: | **User Action** | **System Action** |
| 1- The user will take photo for mobile to detect its features and upload this photo. |  |
|  | 2- System will know the mobile name and get its feature then return them to user. |
| Exceptions: | **User Action** | **System Action** |
|  | 1- The uploaded file not image. |  |
|  |  | System will send to user to put the right image not original file. |

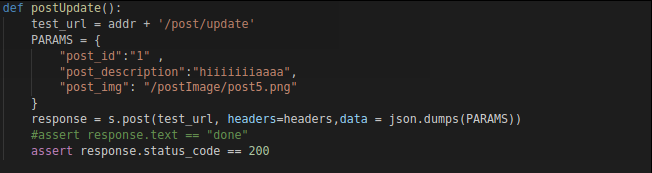
|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Detect video | |
| Actors: | User , store owner | |
| Pre-conditions: | User or store owner will register on the system | |
| Post-conditions: | Have form to upload video to detect mobiles features | |
| Flow of events: | **User Action** | **System Action** |
| 1- The user will take video for mobiles to detect their features and upload this video. |  |
|  | 2- System will make video analysis then retrieve their features. |
| Exceptions: | **User Action** | **System Action** |
|  | 1- The uploaded file not video. |  |
|  |  | System will send to user to upload the right video not original file. |

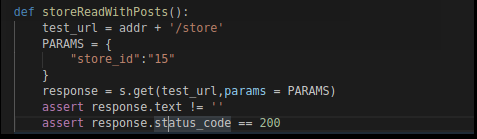
­

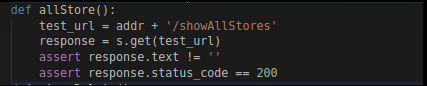
### 3.4 – System test cases:

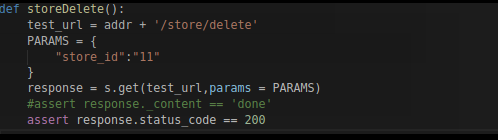


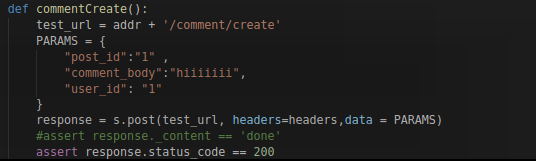


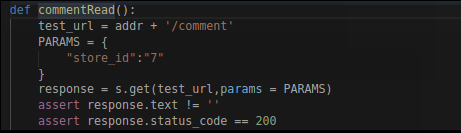


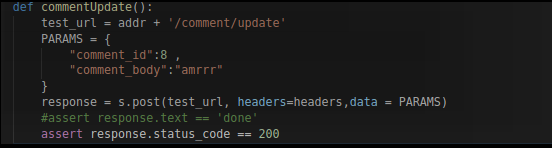


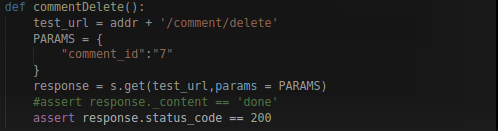


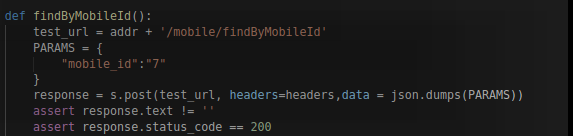


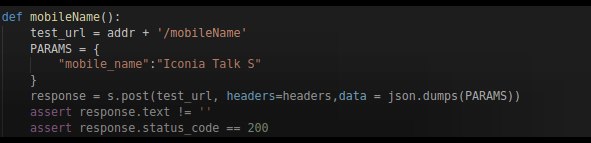


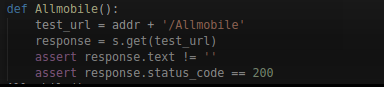


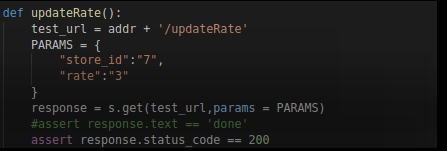




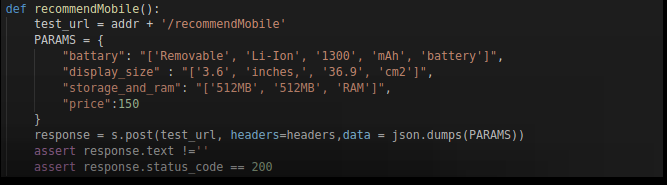


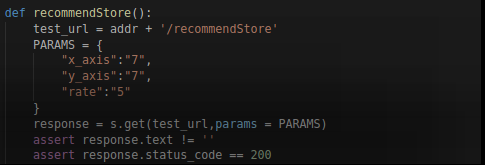


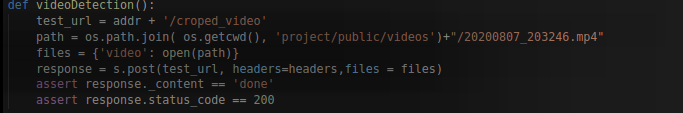


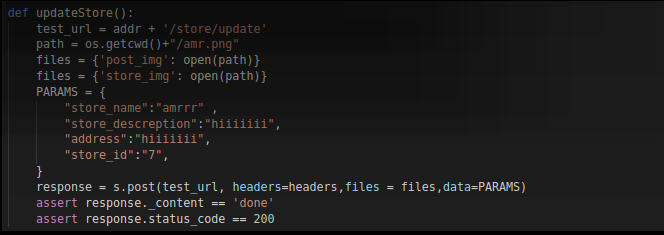


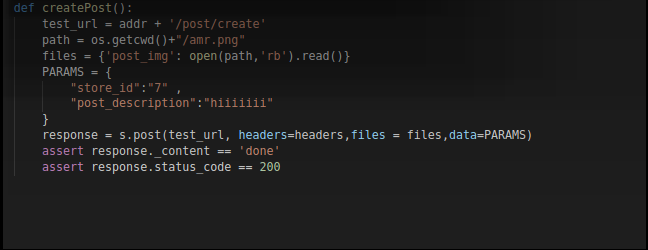


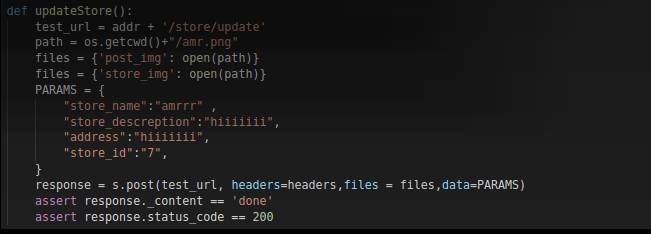


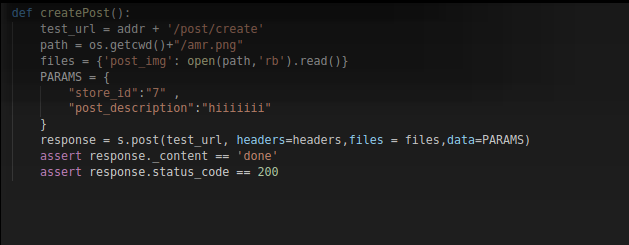












**Test case for login:**

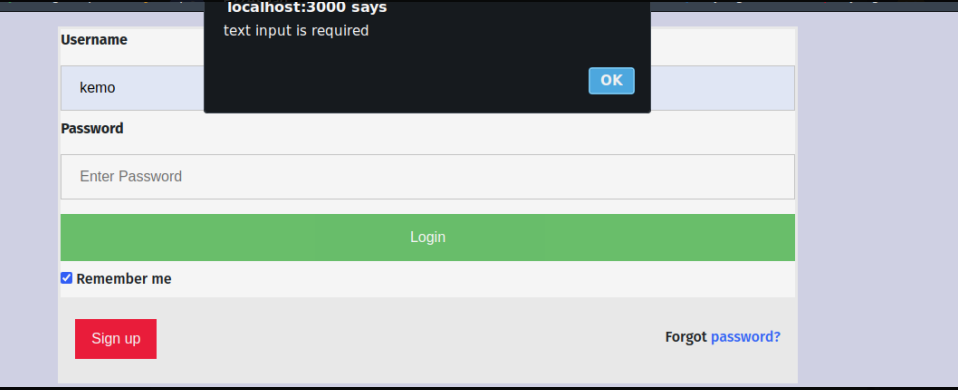


Figure 9

**Test case for recommend shop:**

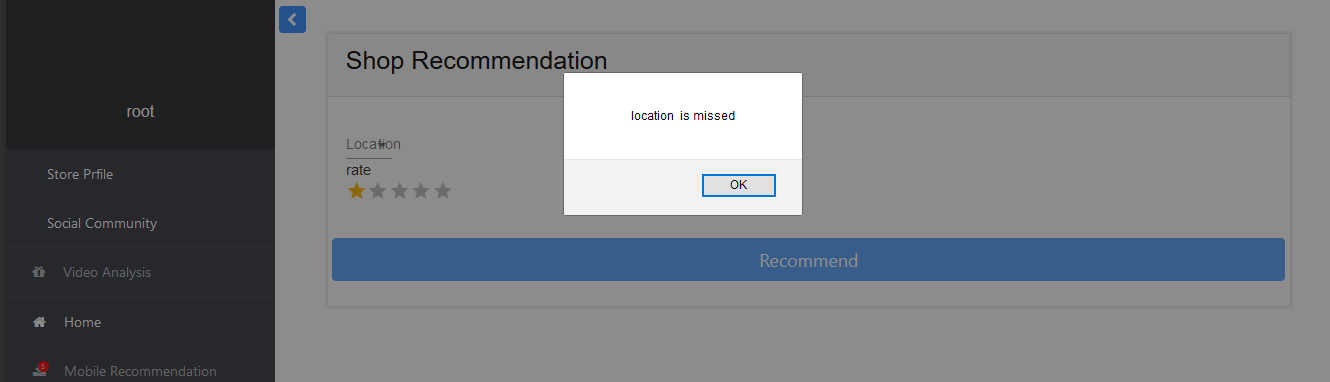


Figure 10

**Test case for recommend mobile:**

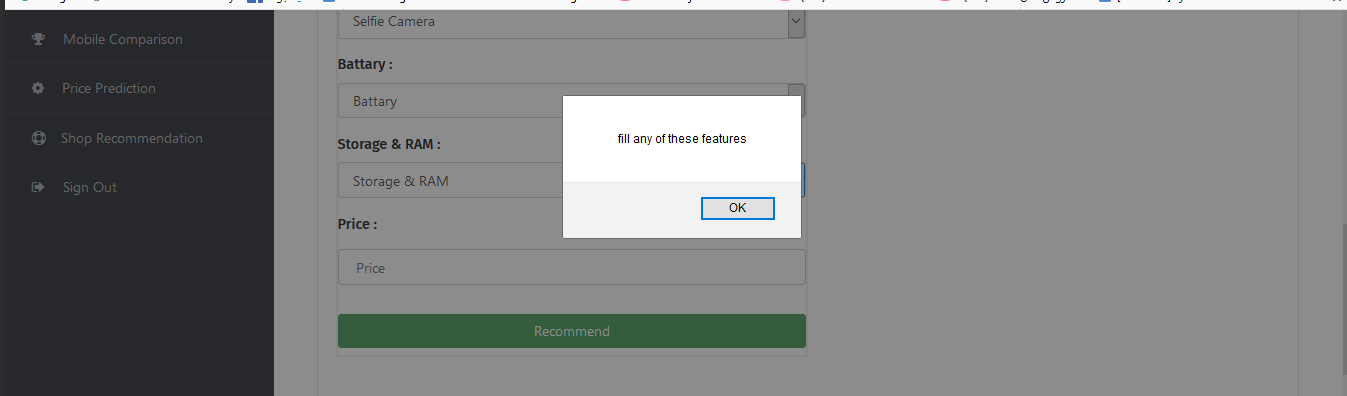


Figure 11

**Test case for recommend shop:**

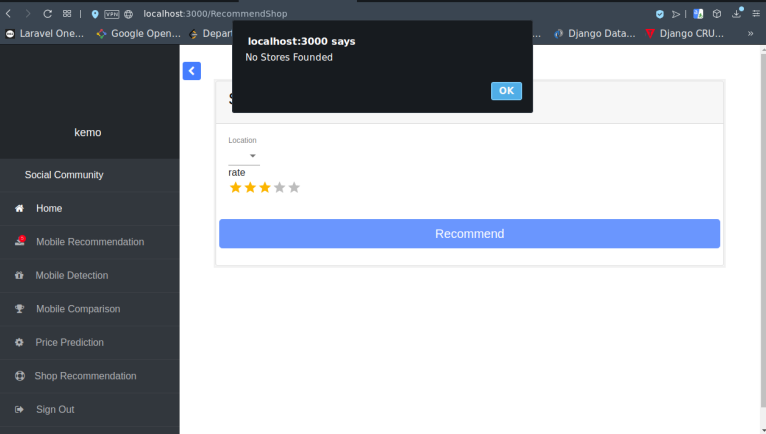


Figure 12

**Test case for update store profile:**

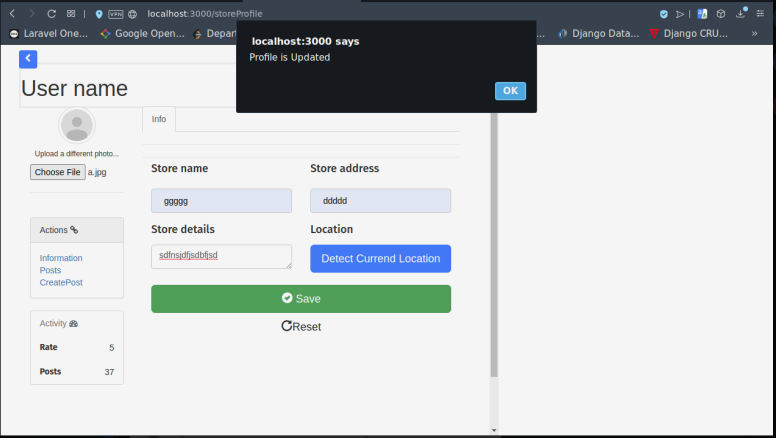


Figure 13

**Test case for create post:**

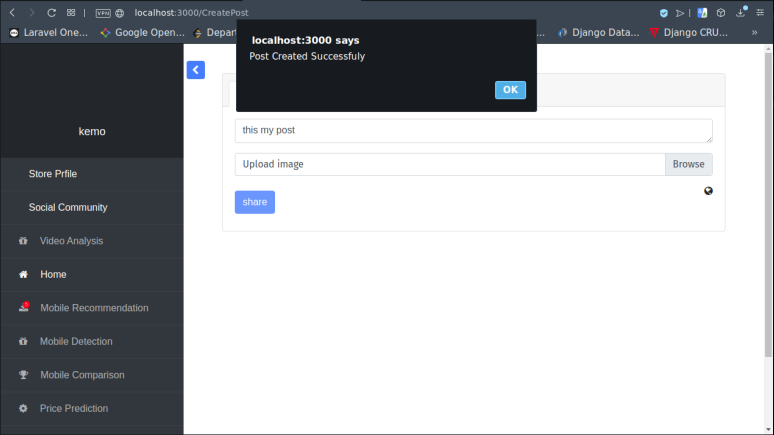


Figure 14

**Test case failed for create post:**

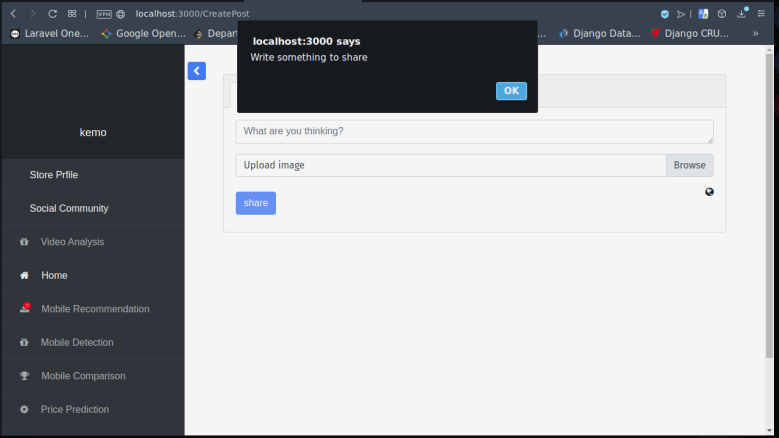


Figure 15

## Chapter 4: System Design:

### 1-System component:

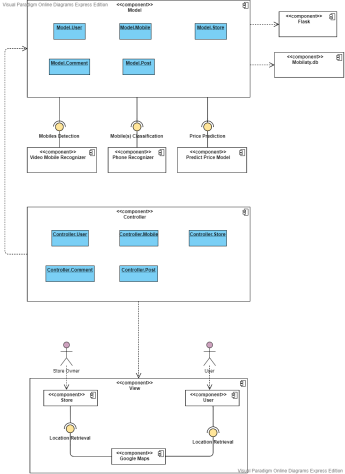


Figure 16

### 2 – System class Diagram:



Figure 17

### 3- Sequence Diagram:

###### **Store recommendation:-**

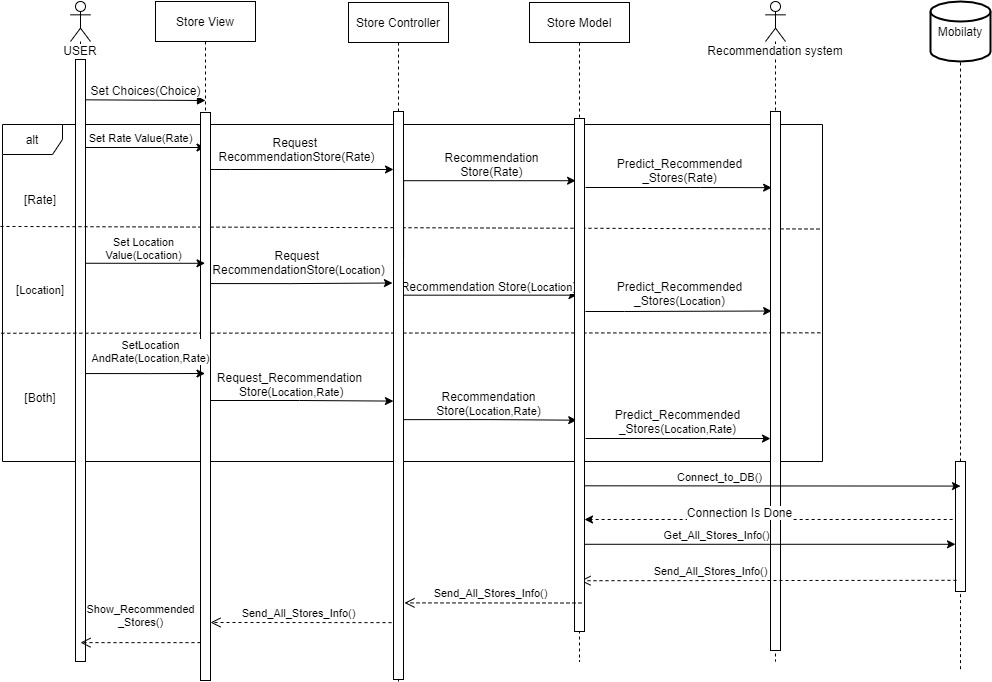


Figure 18

**Photo analyzer:-**

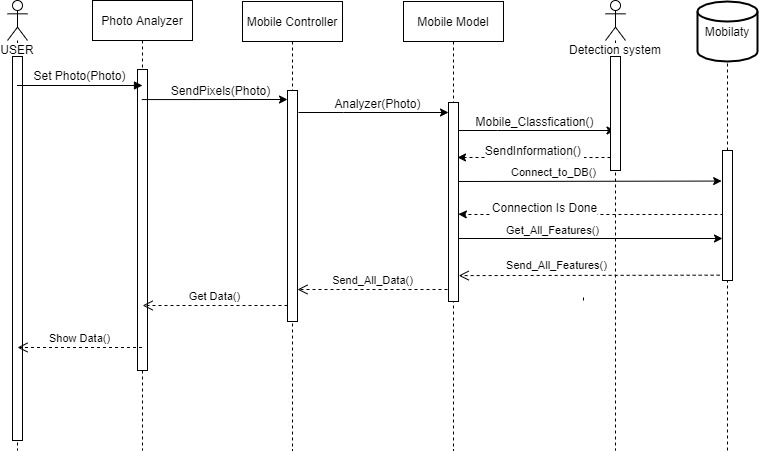


Figure 19

**Mobile recommendation:-**

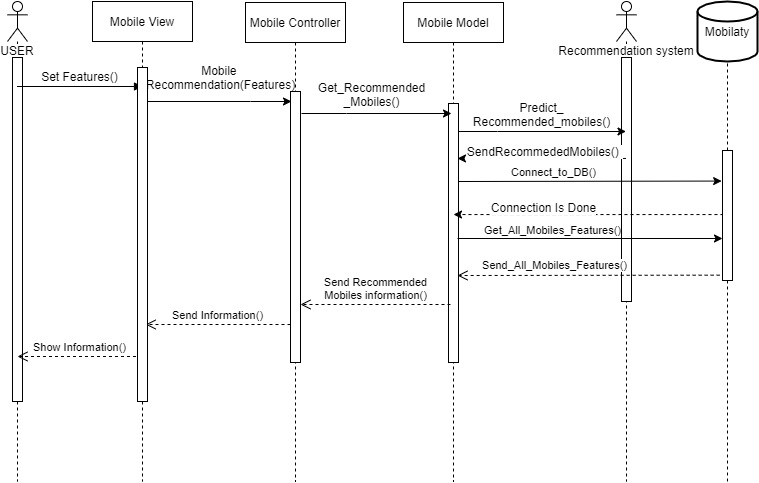


Figure 20

**Video analyzer function:-**

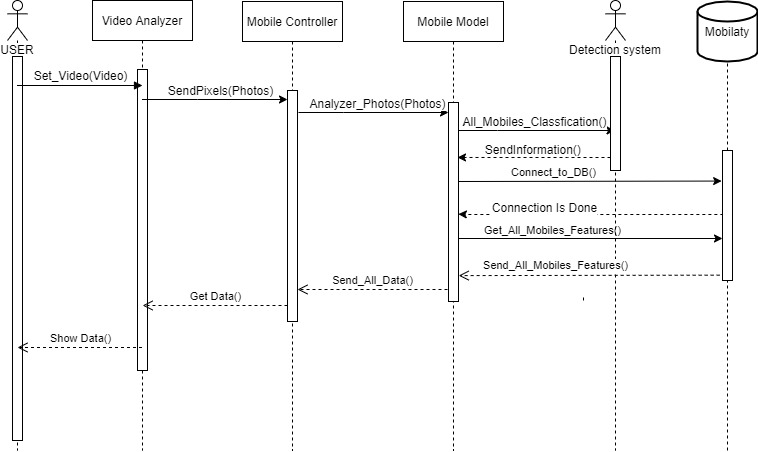


Figure 21

### 4- Project ERD:

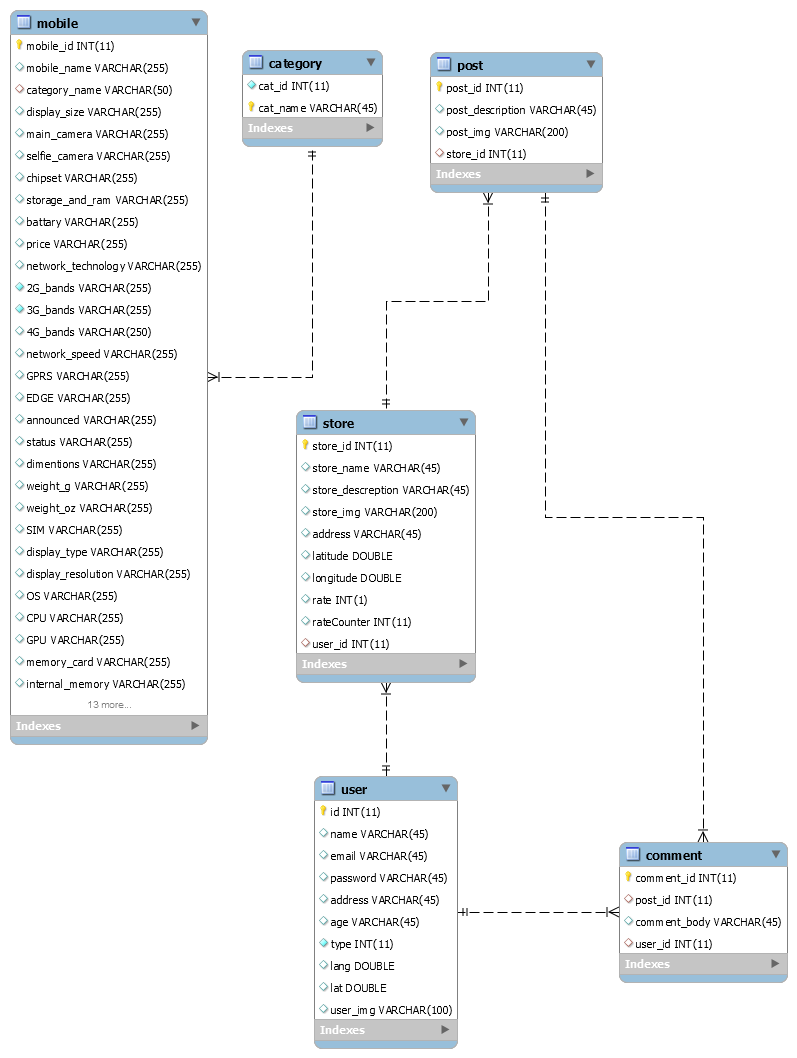


Figure 22

## 5 - System GUI Design:

**Home component:**

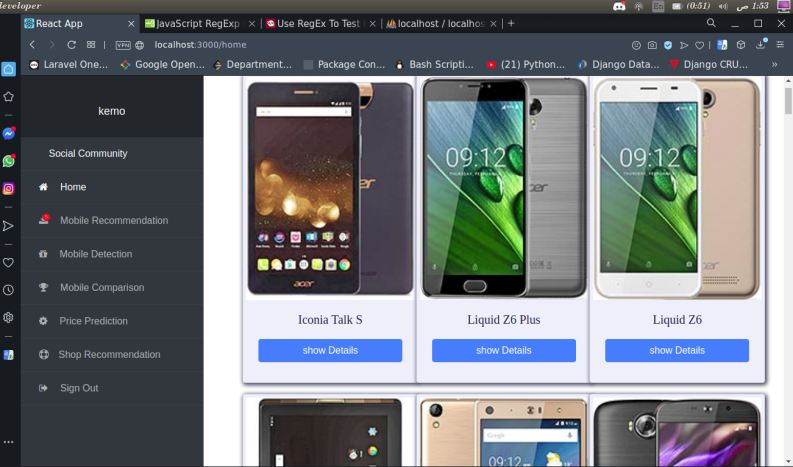
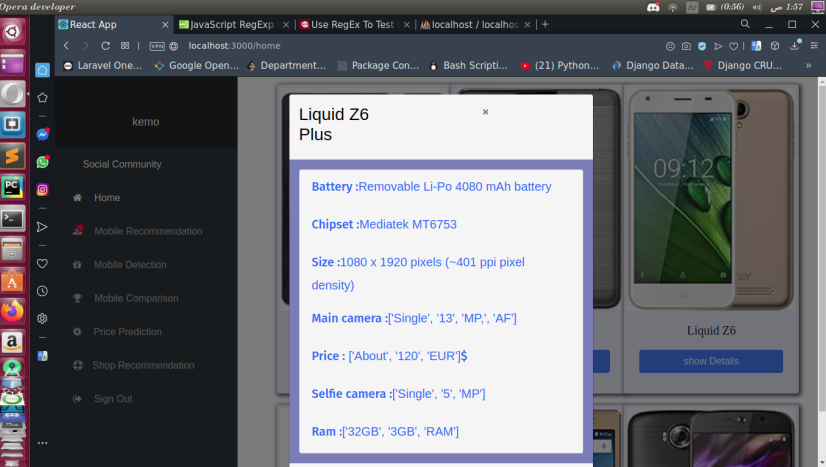


Figure 23

**Modal Component:**



**System community dashboard:**

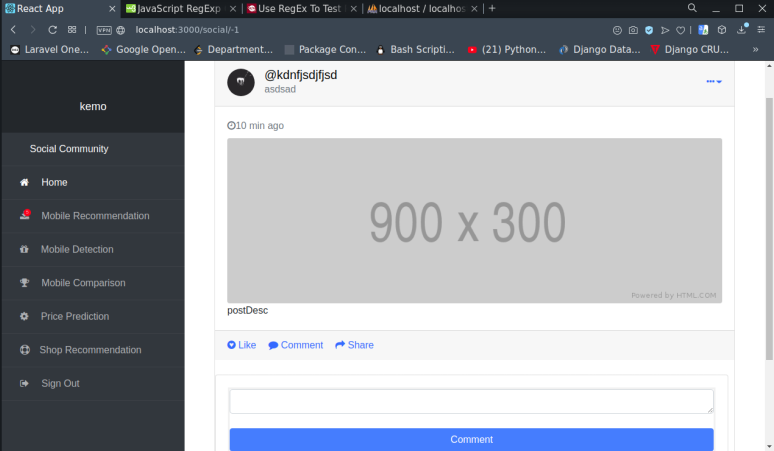


Figure 24

**Comments section:**

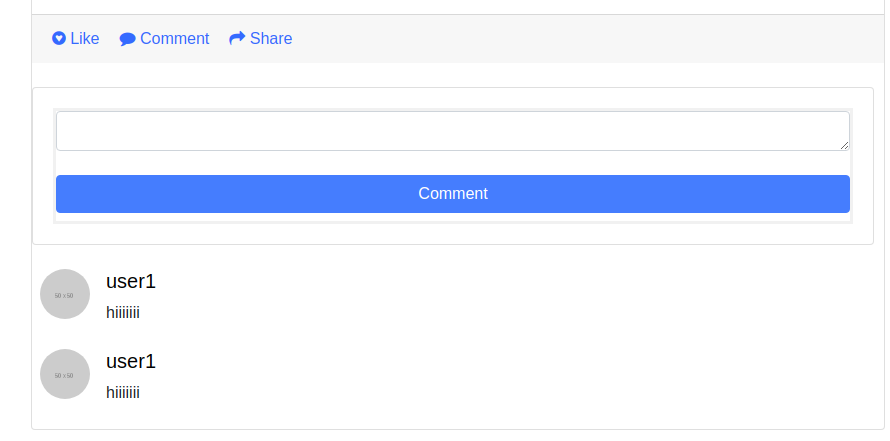


Figure 25

**Social community on smart phones:**

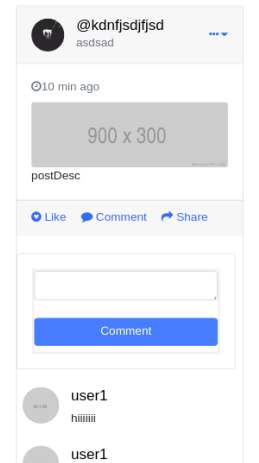


Figure 26

**Mobile Recommendation:**

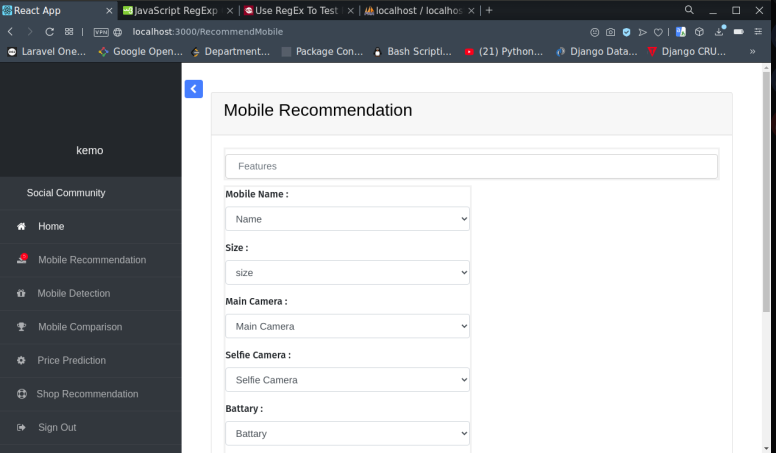


Figure 27

**Mobile Detection:**

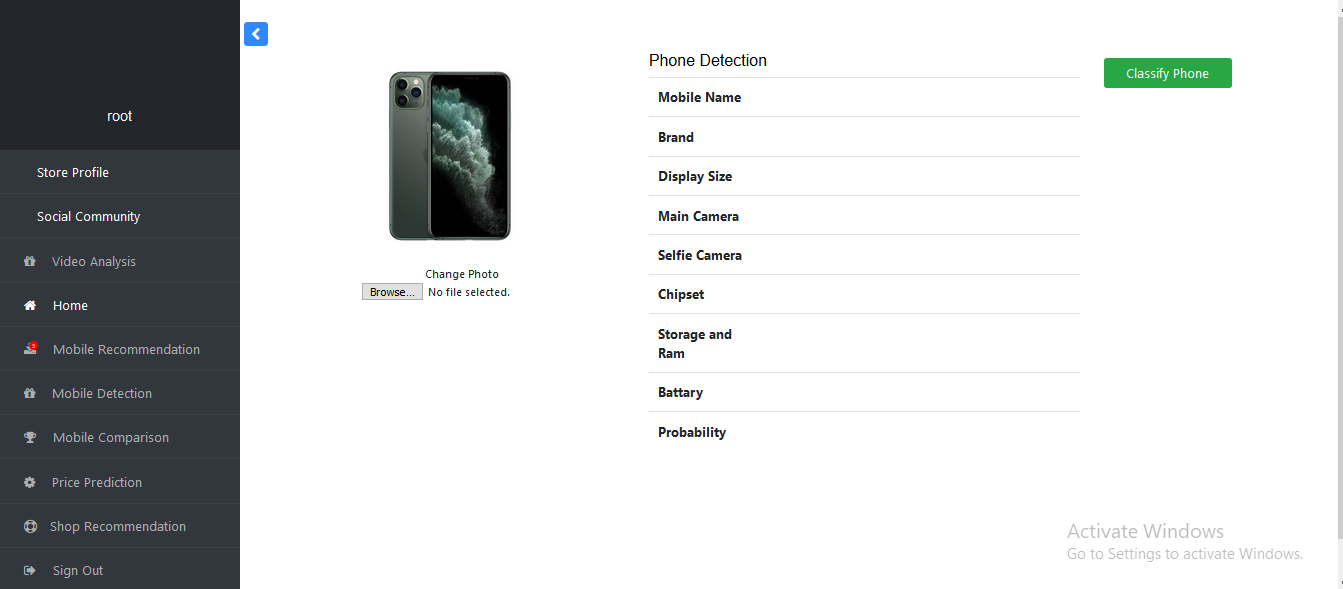


Figure 28

**Mobile comparison:**

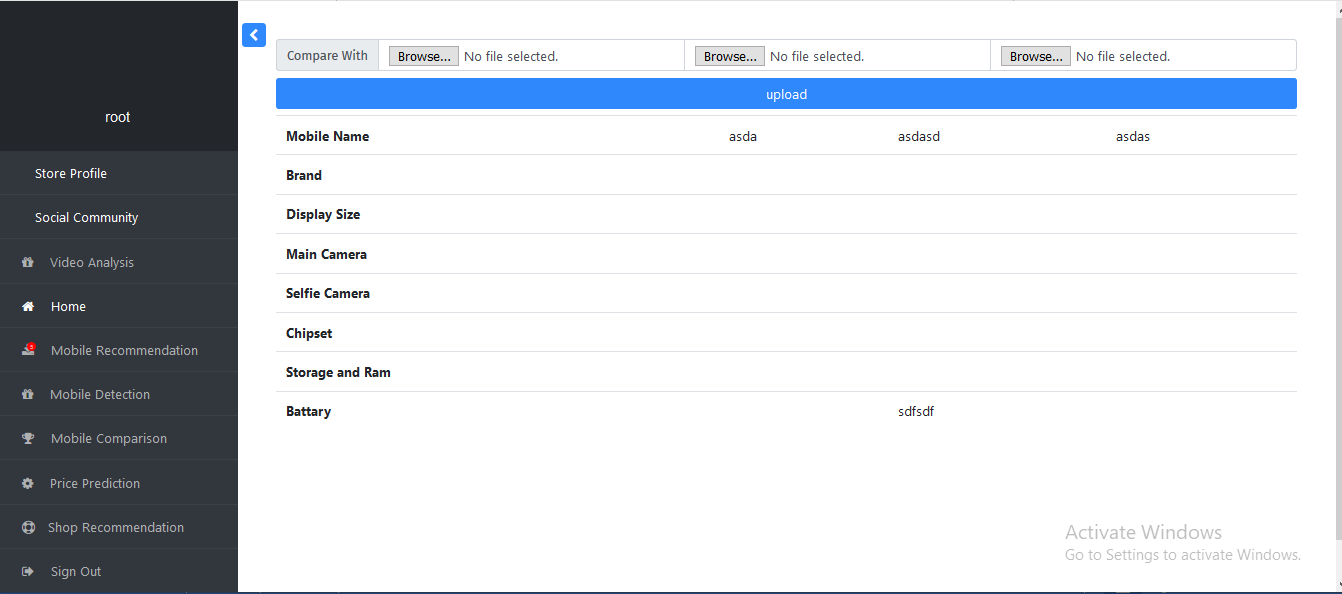


Figure 29

**Store profile:**

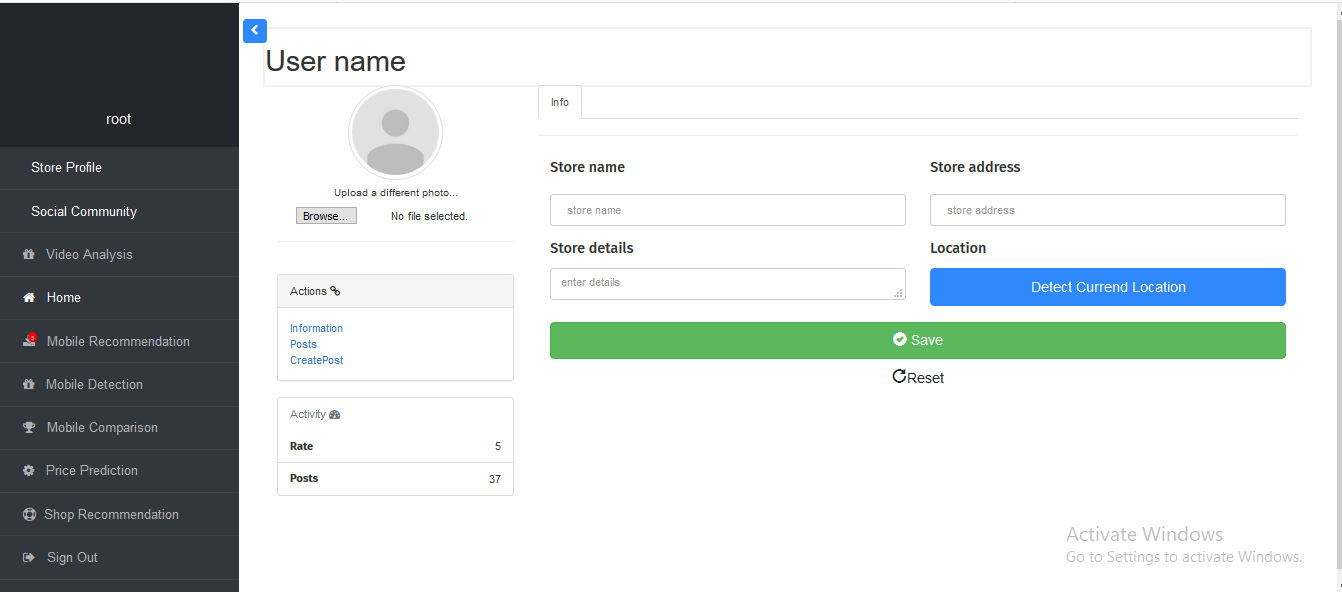


Figure 30

**Shop recommendation:**

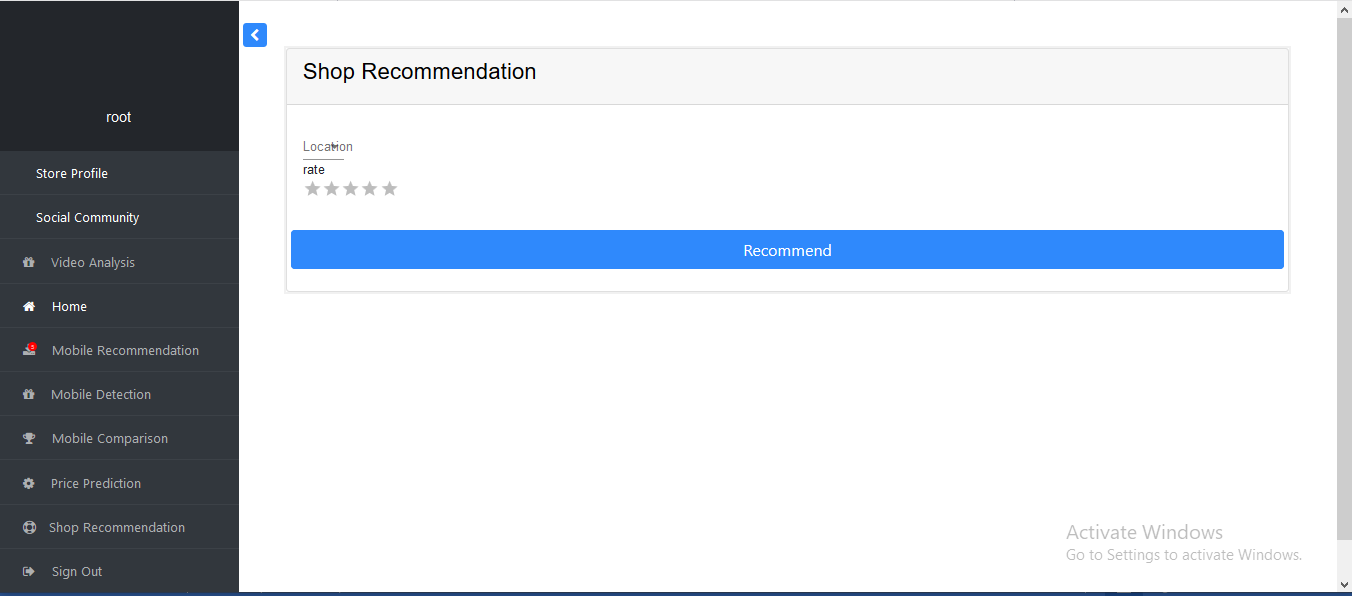


Figure 31

**Create post:**

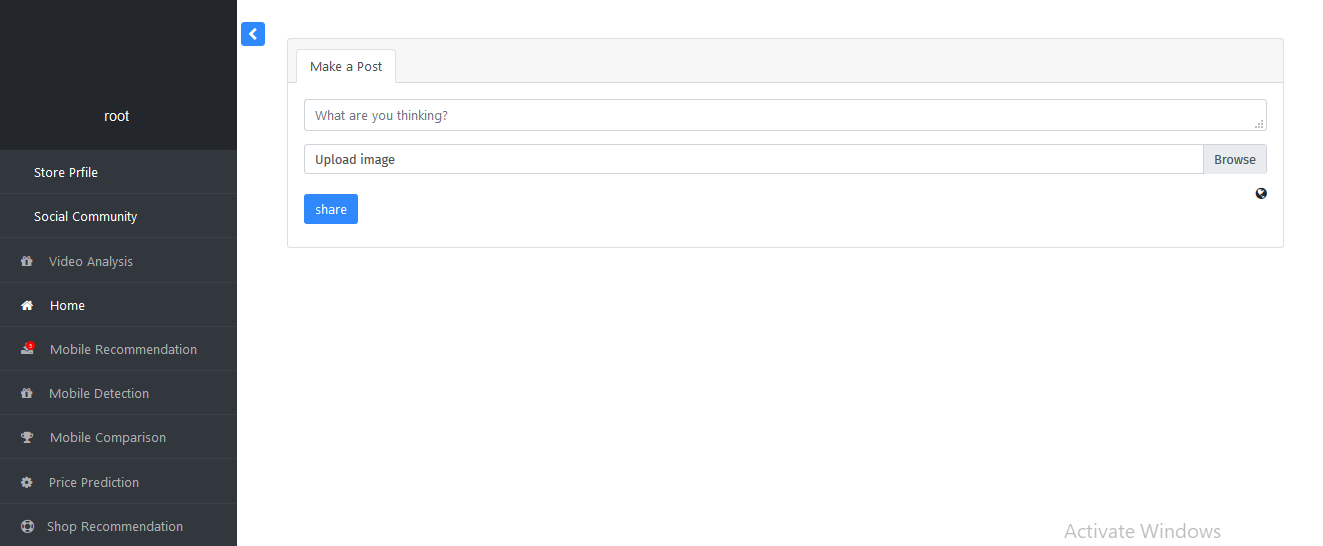


Figure 32

**Upload video:**

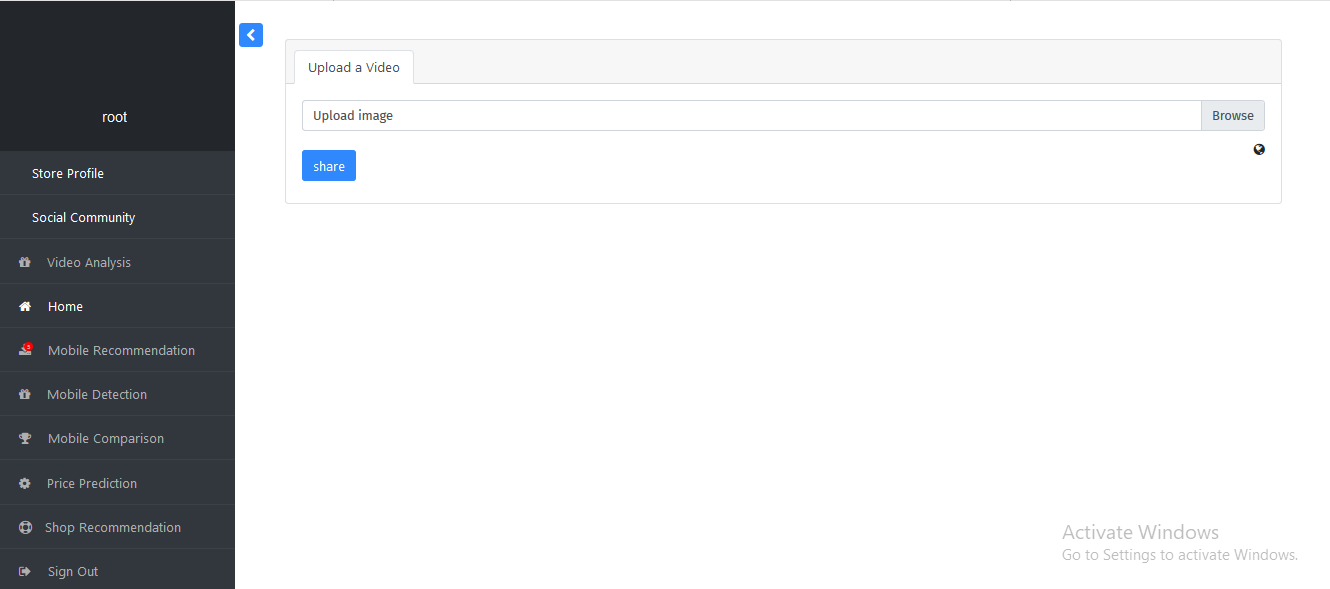


Figure 33

## Chapter 5: Implementation:

**Predict Price:**

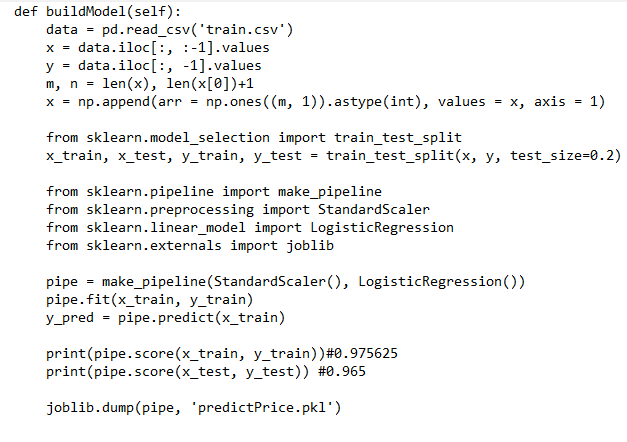


Figure 34

This function take the mobile features as input like (Battery – size – selfi-camera – main c00000amera - ….) then return if this mobile is very cheap or cheap or expensiv0e.

**Detect mobile:**



Figure 35

This function take photo and return the name of the mobile.

**Recommend Mobile:**

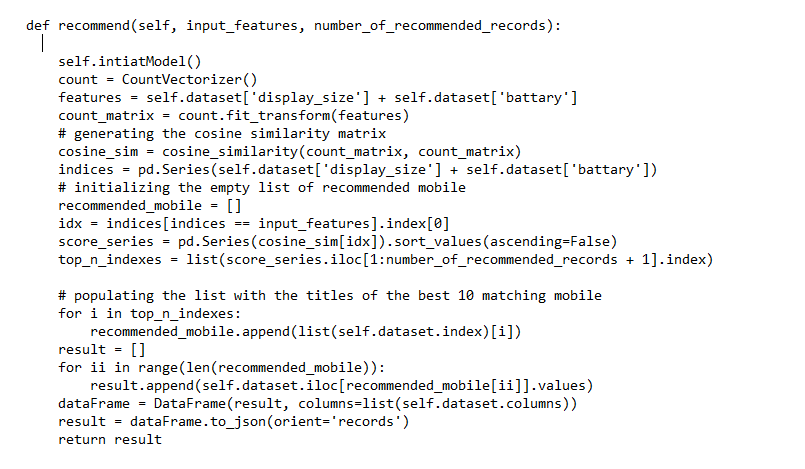


Figure 36

This function take the mobile features and then recommend mobiles.

**Recommend store:**

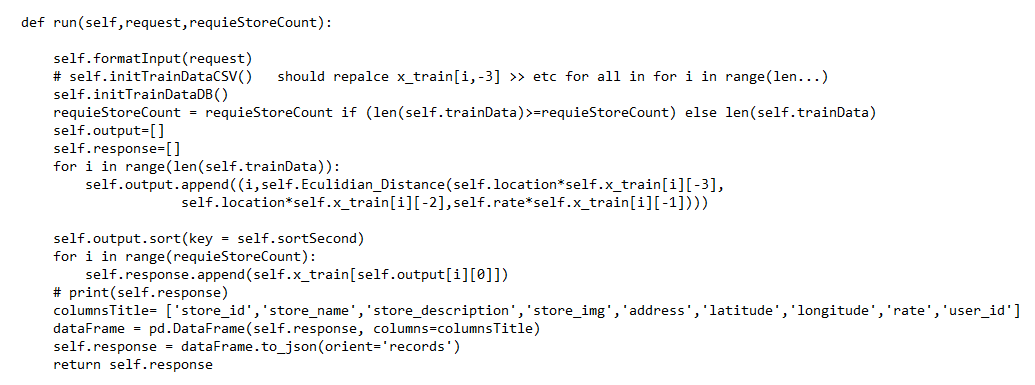


Figure 37

This function take the target store location and its rate to recommend stores.

## References:

- <https://www.atlassian.com/agile/kanban/cards>

- <http://d2l.ai/index.html>

- <https://keras.io/api/>

- <https://imageai.readthedocs.io/en/latest/detection/>

- <https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/>

- <https://www.themarketingtechnologist.co/a-recommendation-system-for-blogs-content-based-similarity-part-2/?fbclid=IwAR1tFiiv1wqbwFRt86Ldrzd5FJEWZxv8EEUXEY4dbM8bTGQxCPR0U6CdlQc>

- <https://datascienceplus.com/building-a-book-recommender-system-the-basics-knn-and-matrix-factorization/?fbclid=IwAR3r0VgR64a7PxMsJUTfvOxOVPdM-DoPg-n-k4mYrURwv5iNV6HbNPDQ30M>

- <https://www.geeksforgeeks.org/image-classifier-using-cnn/?fbclid=IwAR3TEA9ByMfWGk2BK2Oa1DrerOkh55hZggxPCMlKS3zaygEAIQNJD4Nl0Po>

- <https://heartbeat.fritz.ai/awesome-mobile-machine-learning-6dfee1bbaafd>

- <https://www.analyticsvidhya.com/blog/2018/06/understanding-building-object-detection-model-python/?fbclid=IwAR0S0C316Nfw9B2DD8Ntg56ZMxIucggpojns4s6i7J9dcjX96ryjbxCTlqg>

- <https://www.machinelearningplus.com/nlp/cosine-similarity/#:~:text=Cosine%20similarity%20is%20a%20metric,in%20a%20multi-dimensional%20space>.

<https://medium.com/acing-ai/what-is-cosine-similarity-matrix-f0819e674ad1>

- <https://towardsdatascience.com/cosine-similarity-for-movie-recommendation-system-e1852018cf76>

- <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>

- <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

- <https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e>

- <https://dev.to/amelia19/28-best-react-js-admin-dashboard-templates-46b1>

- <https://www.gsmarena.com/makers.php3>

- <https://www.edureka.co/blog/web-scraping-with-python/>

- Peter Flach, Machine Learning. The Art and Science of Algorithms that Make Sense of Data.

- T. Mitchell, Machine Learning, McGraw-Hill

- John D. Kelleher, Fundamental of Machine Learning for predictive Data Analytic.

- Dive into Deep Learning Book