

# System and Software Architecture Description (SSAD)

## Image Processing Platform

### Team 4

Name	First Role	Second Role
Hao Wu	Requirements Engineer	Software Architect
Junran Liu	Operational Concept Engineer	Software Architect
Meiyi Yang	Project Manager	Life Cycle Plan
Vinny DeGenova	IIV & V	Quality Focal Point
Xiangchen Zhao	Life Cycle Plan	Prototyper
Xinhui Liu	Feasibility Analyst	Operational Concept Engineer
Yifan Liu	Prototyper	Requirements Engineer

12/01/2016

# Version History

Date	Author	Version	Changes made	Rationale
10/04/16	HAO WU	1.0	<ul style="list-style-type: none"><li>• Original template for use with Instructional ICM-Sw v1.0</li></ul>	<ul style="list-style-type: none"><li>• Initial draft for use with Instructional ICM-Sw v1.0</li></ul>
10/14/16	HAO WU	1.1	<ul style="list-style-type: none"><li>• Modify the architecture</li></ul>	<ul style="list-style-type: none"><li>• For DC package</li></ul>
12/01/16	HAO WU	2.0	<ul style="list-style-type: none"><li>• Complete Technology-Independent Model</li><li>• Complete Technology-Specific System Design</li><li>• Complete Architectural Styles, Patterns and Frameworks</li></ul>	<ul style="list-style-type: none"><li>• For As Built package</li></ul>

# Table of Contents

<b>System and Software Architecture Description (SSAD)</b> .....	<b>i</b>
<b>Version History</b> .....	<b>i</b>
<b>Table of Contents</b> .....	<b>ii</b>
<b>Table of Tables</b> .....	<b>iii</b>
<b>Table of Figures</b> .....	<b>iv</b>
<b>1. Introduction</b> .....	<b>1</b>
<b>1.1 Purpose of the SSAD</b> .....	<b>1</b>
<b>1.2 Status of the SSAD</b> .....	<b>1</b>
<b>2. System Analysis</b> .....	<b>2</b>
<b>2.1 System Analysis Overview</b> .....	<b>2</b>
<b>2.2 System Analysis Rationale</b> .....	<b>7</b>
<b>3. Technology-Independent Model</b> .....	<b>8</b>
<b>3.1 Design Overview</b> .....	<b>8</b>
<b>3.2 Design Rationale</b> .....	<b>12</b>
<b>4. Technology-Specific System Design</b> .....	<b>13</b>
<b>4.1 Design Overview</b> .....	<b>13</b>
<b>4.2 Design Rationale</b> .....	<b>17</b>
<b>5. Architectural Styles, Patterns and Frameworks</b> .....	<b>18</b>

# Table of Tables

<i>Table 1: Actors Summary.....</i>	<i>2</i>
<i>Table 2: Artifacts and Information Summary .....</i>	<i>3</i>
<i>Table 3: Process Description.....</i>	<i>4</i>
<i>Table 4: Typical Course of Action.....</i>	<i>5</i>
<i>Table 5: Alternate Course of Action .....</i>	<i>5</i>
<i>Table 6: Exceptional Course of Action.....</i>	<i>5</i>
<i>Table 7: Process Description.....</i>	<i>5</i>
<i>Table 8: Typical Course of Action.....</i>	<i>5</i>
<i>Table 9: Alternate Course of Action .....</i>	<i>6</i>
<i>Table 10: Exceptional Course of Action.....</i>	<i>6</i>
<i>Table 11: Process Description.....</i>	<i>6</i>
<i>Table 12: Typical Course of Action .....</i>	<i>6</i>
<i>Table 13: Alternate Course of Action .....</i>	<i>6</i>
<i>Table 14: Exceptional Course of Action.....</i>	<i>7</i>
<i>Table 19: Hardware Component Description .....</i>	<i>9</i>
<i>Table 20: Software Component Description.....</i>	<i>10</i>
<i>Table 22: Design Class Description.....</i>	<i>10</i>
<i>Table 23: Hardware Component Description .....</i>	<i>14</i>
<i>Table 24: Software Component Description.....</i>	<i>14</i>
<i>Table 26: Design Class Description.....</i>	<i>14</i>
<i>Table 27: Architectural Styles, Patterns, and Frameworks.....</i>	<i>18</i>

# Table of Figures

<i>Figure 1: System Context Diagram .....</i>	<i>2</i>
<i>Figure 2: Artifacts and Information Diagram .....</i>	<i>3</i>
<i>Figure 3a: Process Diagram .....</i>	<i>4</i>
<i>Figure 4: Conceptual Domain Model.....</i>	<i>8</i>
<i>Figure 5: Hardware Component Class Diagram .....</i>	<i>8</i>
<i>Figure 6: Software Component Class Diagram .....</i>	<i>9</i>
<i>Figure 7: Deployment Diagram.....</i>	<i>9</i>
<i>Figure 9: Design Class Diagram.....</i>	<i>10</i>
<i>Figure 10: Robustness Diagram.....</i>	<i>12</i>
<i>Figure 11: Sequence Diagram.....</i>	<i>12</i>
<i>Figure 12: Hardware Component Class Diagram .....</i>	<i>13</i>
<i>Figure 13: Software Component Class Diagram .....</i>	<i>13</i>
<i>Figure 14: Deployment Diagram.....</i>	<i>13</i>
<i>Figure 16: Design Class Diagram.....</i>	<i>14</i>
<i>Figure 17a: Process Realization Diagram .....</i>	<i>16</i>

# **1. Introduction**

## **1.1 Purpose of the SSAD**

This document defines comprehensive architecture for the Image Processing Platform(IPP) system. It explicates different aspects of the system by using charts and diagram. The aim of the document is to characterize the details of the whole system, include software components and their functions.

## **1.2 Status of the SSAD**

This is the final version of the SSAD document and we already completed the system in details. This document helps others to understand the basic frame of our system and give an overview of the system. This document also explains why we design the system like this and how the idea came up.

## 2. System Analysis

### 2.1 System Analysis Overview

The IPP can help users to mine specific images from various content elements without acquiring specific knowledge. It identifies same topic images which are similar with the images user uploaded. This gives the possibility to users to monitor image data transferring on the internet and detect the violent elements from the data. Meanwhile, as more images are fetched by system, the accuracy of detection will be improved.

#### 2.1.1 System Context

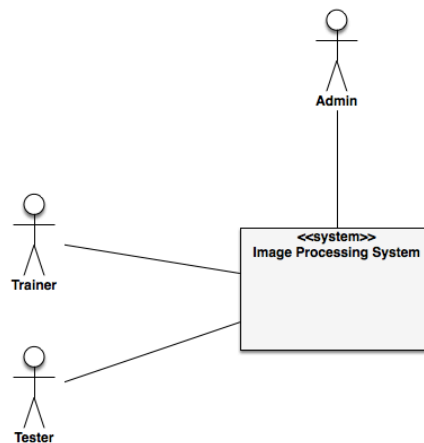
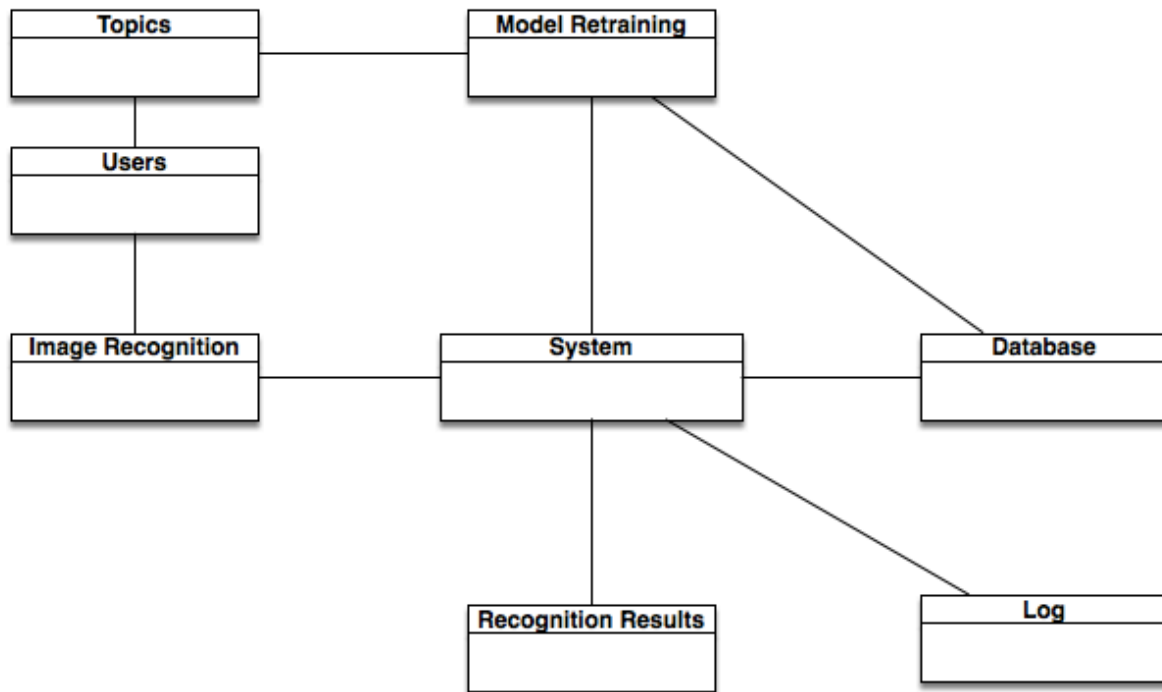


Figure 1: System Context Diagram

Table 1: Actors Summary

Actor	Description	Responsibilities
Trainer	User of the system	Upload images from local and start the model training process
Tester	User of the system	Upload image from local and execute the image recognition process.
Administrator	Administrator of the system	Control and monitor the system

## 2.1.2 Artifacts & Information



**Figure 2: Artifacts and Information Diagram**

**Table 2: Artifacts and Information Summary**

Artifact	Purpose
Image Recognition	To recognize image's category.
Model Training	The system update model with uploaded images.
Topics	Topic is necessary for Image Recognition
Database	To save model and Image to database
System	To process and analysis images from user
Users	To provide user a function to upload images
Recognition Results	To recognize images
Log	To inform administrator system condition.



### 2.1.3 Behavior

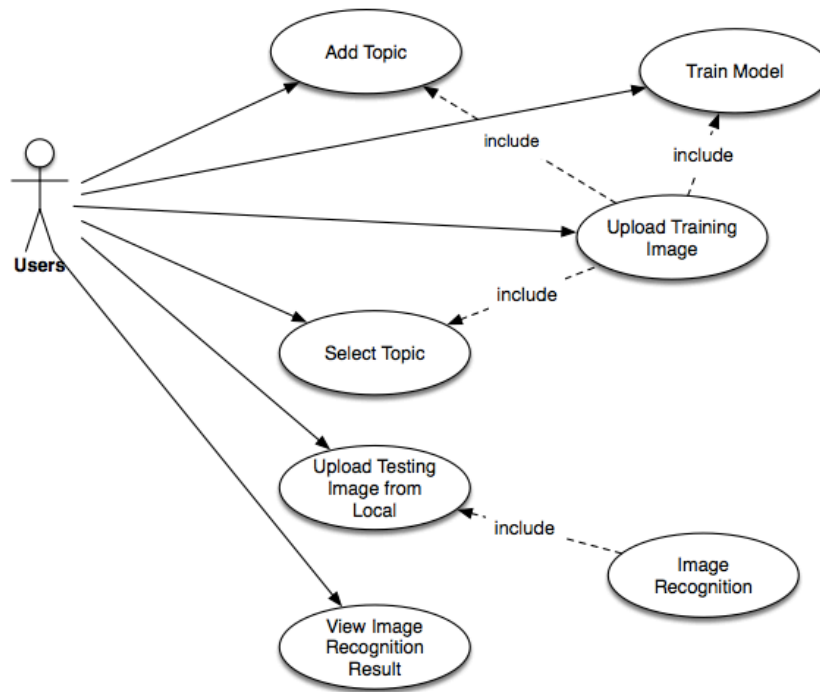


Figure 3a: Process Diagram

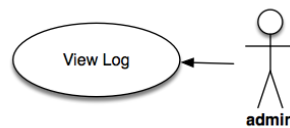


Figure 3b: Process Diagram

1. Table 3 is the process description of uploading training image. Table 4, Table 5, Table 6 talk about the uploading process in detail, the different ways to upload training images and exceptional course of action.

Table 3: Process Description

<b>Identifier</b>	Upload Training Images
<b>Purpose</b>	Upload a set of image to server
<b>Requirements</b>	WC_4147,WC_4109,WC_4040

<b>Development Risks</b>	The system need to identify the format of each images and the system can handle multiple images uploading at same time.
<b>Pre-conditions</b>	Users need to upload proper format images and have good network conditions.
<b>Post-conditions</b>	User trained model

**Table 4: Typical Course of Action**

Seq#	Actor's Action	System's Response
1	Trainer clicks upload button	System provides interface to user to upload images
2	Trainer clicks submit button	System starts to receive images

**Table 5: Alternate Course of Action**

Seq#	Actor's Action	System's Response
1		
2		

**Table 6: Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	Trainer clicks submit button	if there is no image uploaded or the image is improper, the system will inform user and redirect to the uploading page and show the error to user

2. Table 7 gives the description of selecting a topic. Table 8, Table 9, Table 10 talk about the selecting process in detail, the different ways to select topic and exceptional course of action.

**Table 7: Process Description**

<b>Identifier</b>	Add Topic
<b>Purpose</b>	denominate a classifier
<b>Requirements</b>	WC_4147
<b>Development Risks</b>	This process needs to insure that users input appropriate characters and inform users the error information when they failure to meet the input requirement
<b>Pre-conditions</b>	Users need to users input appropriate characters
<b>Post-conditions</b>	User train model

**Table 8: Typical Course of Action**

Seq#	Actor's Action	System's Response
------	----------------	-------------------

1	Trainer clicks add topic button	System provides interface to user to input the topic name.
2	Trainer clicks save button	System adds a topic and saves it in the database

**Table 9: Alternate Course of Action**

Seq#	Actor's Action	System's Response
1		

**Table 10: Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	Trainer clicks save button	if the string user typed in is not match the requirement, system will inform the error information to users.

3. Table 11 gives the description of selecting a topic. Table 12, Table 13, Table 14 talk about the training process in detail, the different ways to start process and exceptional course of action.

**Table 11: Process Description**

<b>Identifier</b>	Model Training
<b>Purpose</b>	Activate the training process to generate a specific classifier and update model.
<b>Requirements</b>	WC_4107
<b>Development Risks</b>	This process needs to apply machine learning algorithm to generate a corresponding classifier on model.
<b>Pre-conditions</b>	Users need to upload a set of images in advance and already assigned a topic to the images.
<b>Post-conditions</b>	Users train model

**Table 12: Typical Course of Action**

Seq#	Actor's Action	System's Response
1	Trainer clicks training button	System checks the images conditions and preprocess the images for smaller size.
2	Trainer view progress button	System returns a feedback (eg. a progress bar)

**Table 13: Alternate Course of Action**

Seq#	Actor's Action	System's Response
1		

**Table 14: Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	Trainer clicks training button	Users don't upload the images or assign a topic.

4. Table 11 gives the description of selecting a topic. Table 12, Table 13, Table 14 talk about the training process in detail, the different ways to start process and exceptional course of action.

## 2.1.4 Modes of Operation

The IPP has two main modes of operation:

1. Training Model mode: This mode is generated by machine learning algorithm and images uploaded by trainer.
2. Image Recognition mode: This mode is used by testers. These users can just use the ready-made model instead of uploading images to generate a new model.

## 2.2 System Analysis Rationale

Most of our users are non-technical persons so that we need design a very simple interface and noticeable entrance. Given that trainers may don't have enough images to generate classifier, we provide extra function which is uploading image from Instagram to let trainer upload same topic images as many as possible. We also allow trainers to use other trainer's topic to update the topic. That's will help them to improve their classifier's accuracy.

## 3. Technology-Independent Model

### 3.1 Design Overview

#### 3.1.1 System Structure

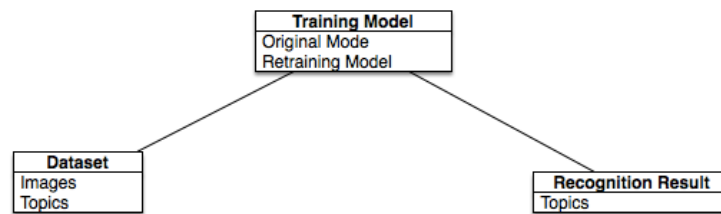


Figure 4: Conceptual Domain Model

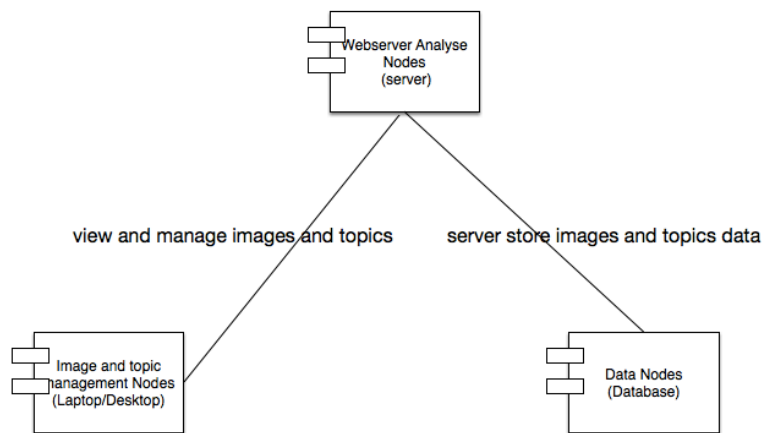


Figure 5: Hardware Component Class Diagram

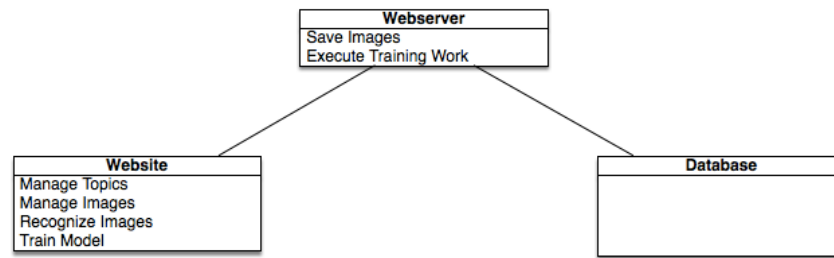


Figure 6: Software Component Class Diagram

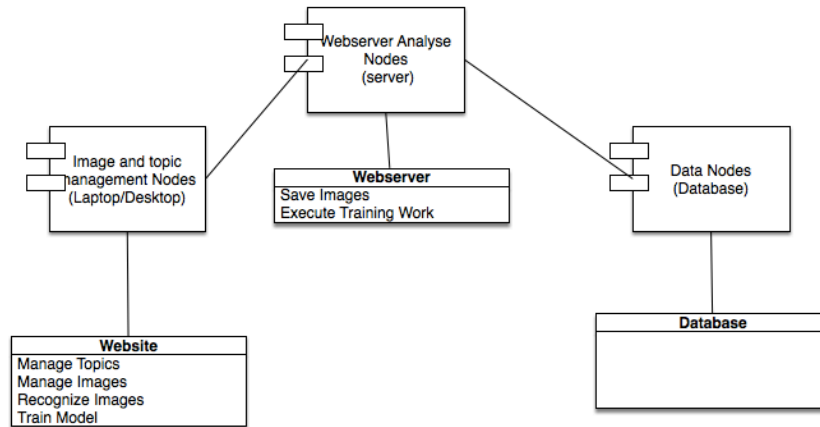


Figure 7: Deployment Diagram

Table 15: Hardware Component Description

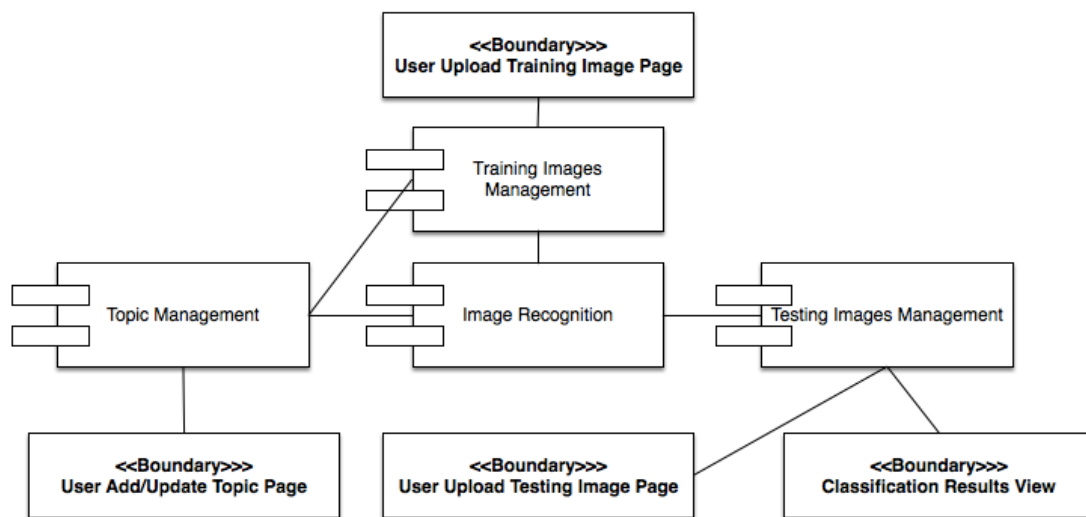
Hardware Component	Description
Image and Topic Management Nodes	Image and Topic Management Nodes are the computers/laptops that the users use to upload images and add or modify topics in the system.
Webserver Analyse Nodes	Webserver Analyse Nodes are the server which run the webserver. They process all the training tasks and testing tasks
Data Nodes	Data Nodes is database service run on the computer/server

**Table 16: Software Component Description**

Software Component	Description
Website	Website software component is the interface provided by us. It gives the view to user to let them view the content and upload their dataset and testing Image.
Webserver	Webserver software component is used to process the requests and receive the images uploaded by users. The core image classification algorithm is contained in the webserver.
Database	Database software component stores some useful data for reusing them.

## 3.1.2 Design Classes

### 3.1.2.1 <Classes n>

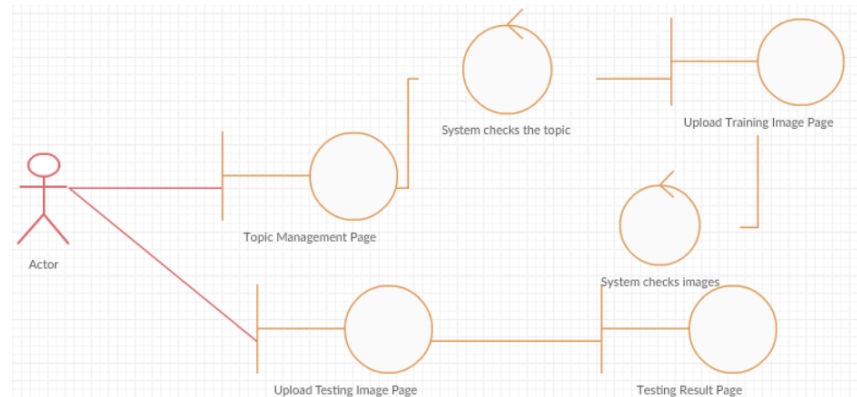
**Figure 8: Design Class Diagram****Table 17: Design Class Description**

Class	Type	Description
User Add/Update Topic Page	Boundary	The user can add a new topic or update the topic they select to scale up or update the new training model.
Topic Management	Component	This component directs user to the topic management page

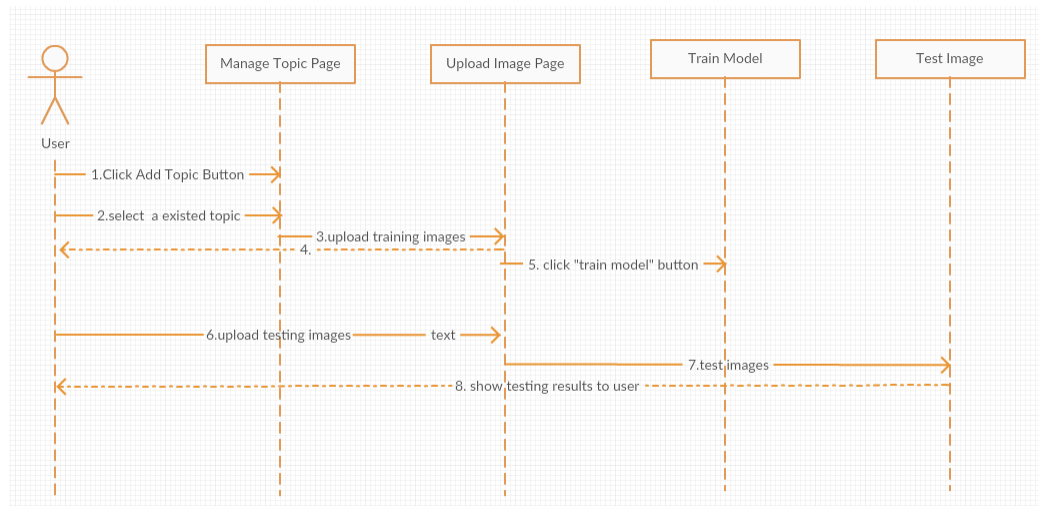
Image Recognition	Component	This component execute image recognition algorithm and create a new model after training process is finished
Training Images Management	Component	This component directs user to the a image management page basic on which topic user choose.
User Uploading Training Image Page	Boundary	This page gives the uploading image function to user.
User Upload Testing Image Page	Boundary	This page gives the uploading image and choosing model function to user.
Training Images Management	Component	This component execute image recognition algorithm to detect images.
Classification Result Page	Boundary	This pages show the testing result to user.



### 3.1.3 Process Realization



**Figure 9: Robustness Diagram**



**Figure 10: Sequence Diagram**

## 3.2 Design Rationale

For the sake of applying our image recognition algorithm into webserver, we need provide some interface to user to let them add new classification and upload correspond images to enlarge the model. Considering most of our users have only little experience computer skill, our process flow should be simple to operate. So we facilitate the process into two main step, one is adding or selecting a topic and the other on is uploading images. The system should be easy to every user. In testing part, we want user can access our image recognition algorithm easily. So what users need to do is just uploading several testing images. After click the testing button, we will give back the final result to users.

## 4. Technology-Specific System Design

### 4.1 Design Overview

#### 4.1.1 System Structure

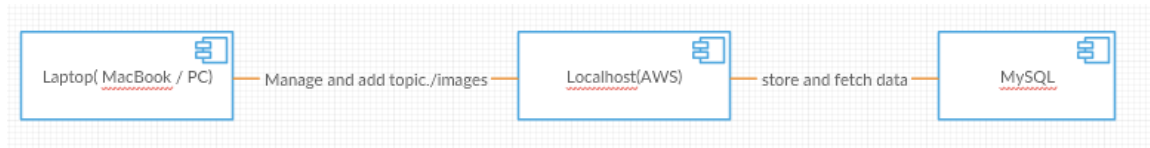


Figure 11: Hardware Component Class Diagram

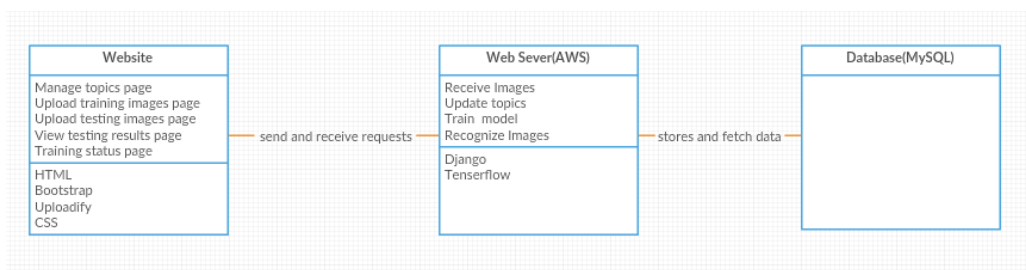


Figure 12: Software Component Class Diagram

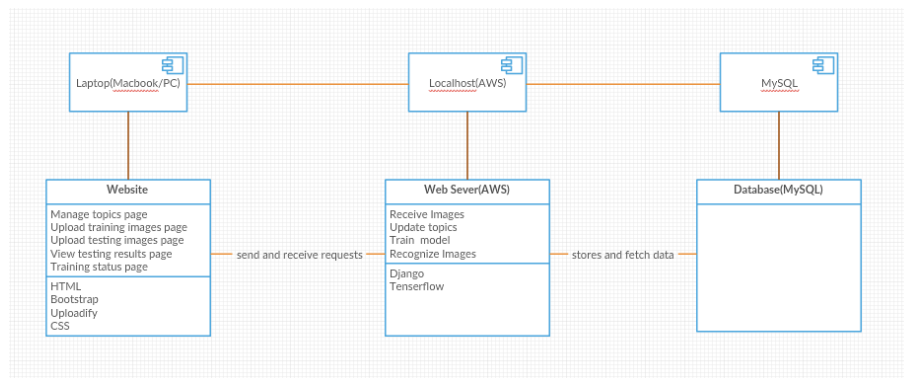


Figure 13: Deployment Diagram

**Table 18: Hardware Component Description**

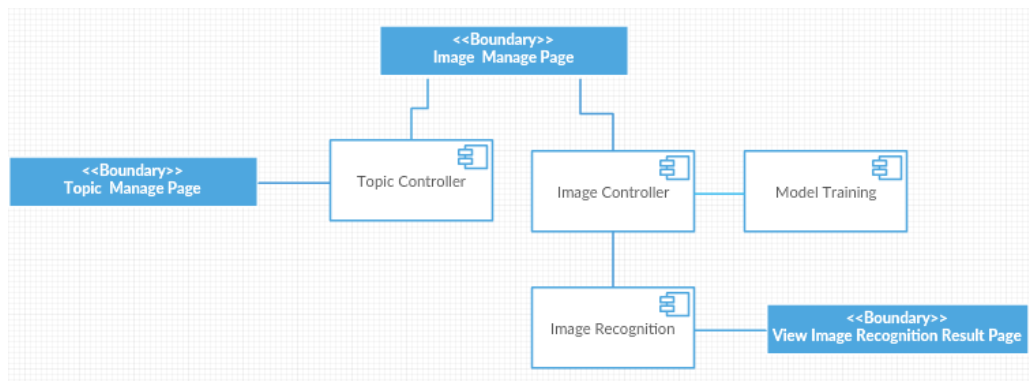
Hardware Component	Description
Laptop(MacBook/PC)	Users use any laptop can access our website and use our system.
LocalHost(AWS)	This component is a webserver which running on AWS. The webserver listen and process the requests from users
MySQL	We use MySQL database to store useful data.

**Table 19: Software Component Description**

Software Component	Description
Website	The Website component uses HTML, CSS, javascript, Uploadify and Bootstrap.
Web Server(AWS)	The Web Server component is deployed on the AWS and use Django web framework and python3 language.
Database(MySQL)	The Database component save the image files and their information for next retraining.

## 4.1.2 Design Classes

### 4.1.2.1 <Classes n>

**Figure 14: Design Class Diagram****Table 20: Design Class Description**

Class	Type	Description
Topic Manage Page	Boundary	Users use this page to add a new category or update the category

Image Manage Page	Boundary	Users use this page to upload training or testing images to server and train a new model or recognize images
View Image Recognition Result Page	Boundary	Users can view the recognition result through this page
Topic Controller	Component	Topic Controller component collect topic information input by users and redirects to Image Manage page.
Image Controller	Component	This controller component allows users uploading multiple images through Image Manage Page. In addition, the component will send the image files to server.
Model Training	Component	This component will run the learning algorithm, process the images and analyze them to build a new model.
Image Recognition	Component	This component will classify the images and pass the results to the View Image Recognition Result Page.

### 4.1.3 Process Realization

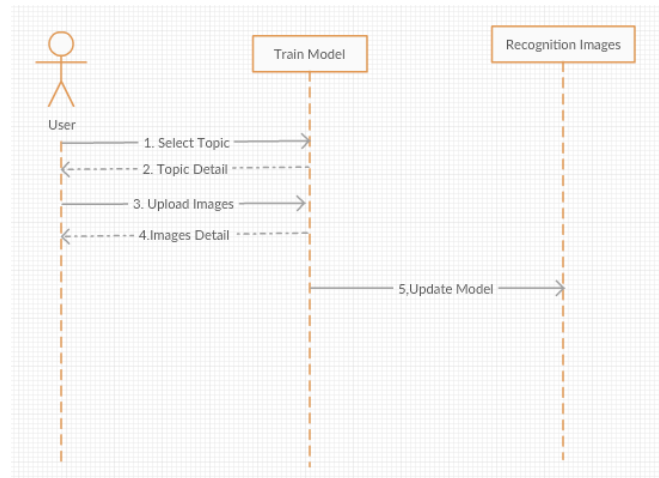


Figure 15a: Process Realization Diagram

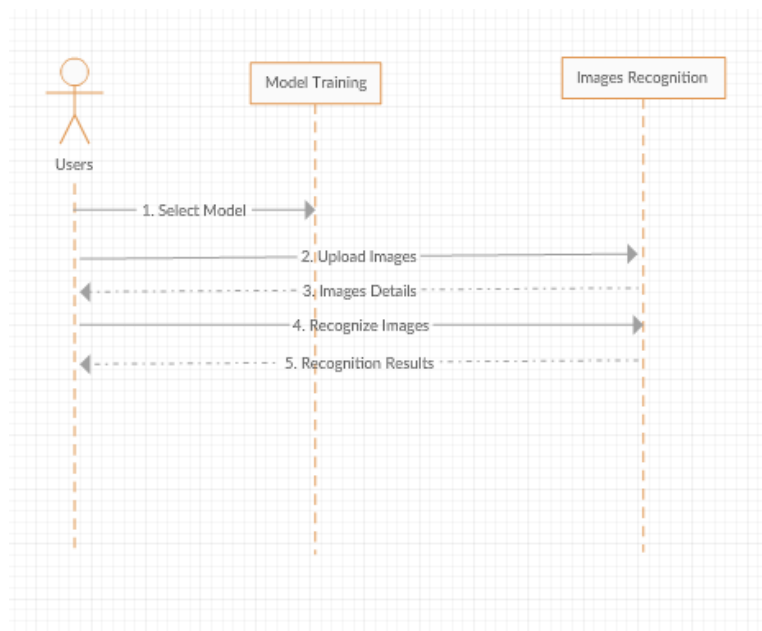


Figure 17b: Process Realization Diagram

## 4.2 Design Rationale

Our overall architecture is very simple, but very efficient. We use django as our entire web framework and connect the server to database. The reason why we choose django is because most of us have some experience on python language and we can develop our system quickly and efficiently. Our client provides suggests us to deploy our system onto AWS, so we choose to run our web server on the AWS. We use Tensorflow as our machine learning algorithm because it uses advanced algorithm to classify images efficiently. Tensorflow also has high accuracy rate on image recognition compared with other similar algorithm.

## 5. Architectural Styles, Patterns and Frameworks

**Table 21: Architectural Styles, Patterns, and Frameworks**

Name	Description	Benefits, Costs, and Limitations
Facade pattern	The learning algorithm integrated on the web server is very complicated, so what we choose façade pattern to help our development. The second reason we choose façade pattern is that we have many third library, such as bootstrap, uploadify, tensorflow, the façade pattern can reduce the dependencies of these third library on our system to make our system easier to modify.	The benefit of using façade pattern is reducing the difficulty of modifying the system when we need to substitute component in the system. The cost and limitation is that we need spend lots of time to code many interface-related code.
MVC	MVC(Model-View-Controller) is the most common software design pattern. We choose MVC design pattern because we are familiar with it and the we framework we choose is also completely support MCV.	The benefits of using MVC is that we can easily divide the entire development work into every team member instead of considering how to integrate everyone's work.