

**University Institute of Information Technology,**

**PMAS-Arid Agriculture University,**

**Rawalpindi Pakistan**

**Personality Prediction Using Deep Learning**

***By***

**Hammad Rehman Saad 17-ARID-1316**

**Imran Khan 17-ARID-1325**

**Tanzeel Ahmad 17-ARID-1425**

***Supervisor* Mr. Zeeshan Javed**

***Bachelor of Science in Computer Science (2017-2021)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software documentation and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Student Name 1 Student Name 2 Student Name 3

Hammad Rehman Saad Imran Khan Tanzeel Ahmad

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS) “Personality Prediction Using Deep Learning” was developed by “Hammad Rehman Saad**, 17-ARID-1316”**, “Imran Khan**, 17-ARID-1325”** and “**Tanzeel Ahmad, 17-ARID-1425”** under the supervision of

“Mr. Zeeshan Javed” and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Science.

---------------------------------------

**Supervisor**

---------------------------------------

**External Examiner (If any)**

---------------------------------------

**Administrator UIIT**

**Executive Summary**

In public places, there is often a need for monitoring people and different activities going on, which can be referred later for many reasons including security. Appointing humans for this task involves many problems such as increased employee hiring, accuracy problem, trust, no proof for later use, and also the fact that a human can remember things till a certain time limit. Talking about the current security system, they use dumb still cameras with a continuous recording facility ir-respective of the fact that any event may happen or not. Moreover they are usually pointing at a specific user defined locations so more than one cameras are required to cover the entire region.

To prevent all these problems from prevailing, the CSCS is developed. It is a surveillance system, which provides solution to many of these problems. It is a stand-alone application which doesn’t require any computer to operate. It monitors different situations using a camera which is able to rotate intelligently based on sensor messages and captures the scene in the form of video or photos later reference as well.

**C**ustomizable **S**urveillance **C**ontrol **S**ystem **(CSCS)** is a surveillance system that can be assigned a sensor type as in our case a heat sensor is used, it works accordingly, rotates the camera upon event detection and perform user defined actions like capturing video and stores them, for the future use.

It is an embedded system consisting of Linux fox kit with embedded a running server application also a camera, USB storage device and a sensor node base station is attached with fox kit. LAN communication is used by user to download the videos and to operate the system manually.

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Dr. Kashif Sattar” and our Co-Supervisor “Dr. Tariq Ali” for personal supervision, advice, valuable guidance and completion of this project. We are deeply indebted to him/her/them for encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Student Name 1 Student Name 2 Student Name 3

Hammad Rehman Saad Imran Khan Tanzeel Ahmad

**Abbreviations**

|  |  |
| --- | --- |
| **SRS** | Software Requirement Specification |
| **PC** | Personal Computer |
|  |  |
|  |  |
|  |  |

**Table of Contents**

**Introduction 1**

[1.1 Brief](#_Toc268523777) 2

[1.2 Relevance to Course Modules](#_Toc268523779) 2

[1.3 Project Background](#_Toc268523780) 3

[1.4 Literature Review](#_Toc268523780) 3

[1.5 Methodology and Software Life Cycle](#_Toc268523782) 5

**Problem Definition** 6

[2.1 Problem Statement](#_Toc268523795) 7

[2.2 Deliverables and Development Requirements](#_Toc268523796) 7

[2.3 Proposed Architecture](#_Toc268523804) 7

[2.4 Operating Environment](#_Toc268523807) 8

[2.5 Assumptions and Dependencies](#_Toc268523808) 8

**Requirement Analysis** 9

[3.1 Functional Requirments](#_Toc268523823) 10

[3.2 Non – Functional Requirments](#_Toc268523825) 11

[3.2.1 Usability](#_Toc268523787) 11

[3.2.2 Reliability](#_Toc268523787) 11

[3.2.3 Performance](#_Toc268523787) 11

[3.2.4 Supportability](#_Toc268523787) 11

[3.2.5 Design Constraints](#_Toc268523787) 11

[3.2.6 Licensing Requirements](#_Toc268523787) 11

[3.3 Use case Model](#_Toc268523823) 12

[3.3.1 Use Case Diagarm](#_Toc268523787) 12

[3.3.2 Actors Discription](#_Toc268523787) 16

[3.3.3 Use Case Discription](#_Toc268523787) 17

**The Design** 21

[4.1 UML Structural Diagrams](#_Toc268523830) 22

[4.1.1 Component Diagram](#_Toc268523787) 22

[4.1.2 System Component Diagram](#_Toc268523787) 24

[4.1.3 Package Diagram](#_Toc268523787) 25

[4.1.4 Deployment Diagram](#_Toc268523787) 26

[4.2 UML Behavioral Diagrams](#_Toc268523830) 27

[4.2.1 Activity Diagrams](#_Toc268523787) 27

[4.2.2 State Machine Diagrams](#_Toc268523787) 29

[4.3 UML Interaction Diagrams](#_Toc268523830) 30

[4.3.1 Sequence Diagrams](#_Toc268523787) 30

[4.4 Node Structure](#_Toc268523830) 31

[4.5 Communication Design Protocol](#_Toc268523830) 32

**Implementation** 33

[5.1 Communication Protocol Implementation](#_Toc268523830) 34

[5.2 PC Application Implementation](#_Toc268523830) 36

[5.3 Embedded Application Implementation](#_Toc268523830) 38

[5.4 Wireless Sensor Application Implementation](#_Toc268523830) 51

**Testing and Evaluation** 52

[6.1 Verification](#_Toc268523830) 53

[6.1.1 Functional Testing](#_Toc268523787) 53

[6.1.2 Static Testing](#_Toc268523787) 57

[6.2 Validation](#_Toc268523830) 57

[6.3 Usability Testing](#_Toc268523830) 57

[6.4 Unit Testing](#_Toc268523830) 57

* 1. [Integration Testing](#_Toc268523830) 57

6.6 [System Testing](#_Toc268523830) 57

**GUI** 58

**Future Work** 70

**References** 72

**List of Figures**

Fig 1.1 Block Diagram 8

Fig 2.1 Use Case Diagram 9

# Chapter 1: Introduction

This chapter provides the overview of the project. The first paragraph of every chapter should provide the chapter summary.

# Brief

This system predicts the personalities of different people. Particularly, adopting methods of predicting personality traits based on core concepts and algorithms of deep learning are the main objectives. Moreover, face detection and recognition along with expressions prediction are also included.

This system is based upon Agile methodology and the technologies that are used in this system are Python, OpenCV, RCNN, NN, LSTM, RNN and Anaconda.

# Relevance to Course Modules

Our project is related to various courses we have studied in our degree which are mentioned below:

* Database systems
* Distributed Database system

Above mentioned courses helped us in uploading of datasets into our database and maintain them.

* Software Engineering-I
* Software Engineering-II

Above mentioned courses not only helped us in developing the UML and Class diagrams as well as Use Cases of the model.

# Project Background

Many methods are used to assess a person's character and personality. Some are through the body movements, body language, handwriting and signatures. Assessing a person's character and personality will be the main objective in order to produce a better system to predict personalities. Some popular measures of personality are not useful because they are unreliable or invalid.

The proposed system will enable a more effective way to predicate the personality of a person by using the machine learning algorithms with the help of images and their textual data with the help of OpenCV, Dlib from a large number of dataset providing a consistent and fair information about them. This can be legally justified.

# Literature Review

The personality prediction through data mining uses classification algorithms performed for predicting the personality of the user. This Personality Prediction System will help the organization by saving their resources and also the applicants by telling them their strong and weak areas. Naïve Bayes Algorithm has the best accuracy in two methods tested with an average accuracy of around 60%. Support Vector Machine method performance was a little worse than Naïve Bayes due to the difficulties of separating a class of a word as dataset was not quite accurate. [1]

Sentiment Analysis and Personality Prediction through Social Media using Myers–Briggs Type Indicator (MBTI) personality assessment test to analyze textual data with the use of different classification algorithms. The data are collected from Kaggle with approximately 8600 rows of Twitter data. The system is tested using 25% of the dataset and the remaining 75% is for the training set. The results show an average accuracy rate of 78.2% with the use of different classification algorithms, and a 100% accuracy rate using the Random Forest (RF) and Decision Tree classifiers. [2]

The Live human–robot interactive public demonstrations for personality automatic emotion prediction is a system where social robots are used. A social robot is defined as a (semi-)autonomous robot that is able to communicate with humans or other robots, and engage in social interactions by following social behaviors and norms but has a few limitations i.e. the cognitive and social capabilities of the humanoid robots are still very limited. There is a genuine need for scientists working in the fields of robotics and artificial intelligence to demonstrate their work and engage the public. HRI is known to have lower repeatability, and tools and metrics developed in human–computer interaction do not directly transfer to HRI. [3]

# Analysis from Literature Review

This critical analysis aims to position the different approaches mentioned in Literature Review for personality prediction by highlighting the important limitations of those models. These are the limitations (a) In case of data mining approach the dataset wasn’t quite accurate due to which it resulted with low accuracy following both the methods of Naïve Bayes and SVM (b) The MBTI method has been failed to predict personalities. Some research suggests the MBTI is unreliable because the same person can get different results when retaking the test (c) In Live-Human interactive demonstration humanoid robot the cognitive and social capabilities of the humanoid robots are still very limited due to which HRI is known to have lower repeatability.

In contrary to above techniques mentioned, this project will result in better personality prediction results and better accuracy.

# Methodology and Software Lifecycle for This Project

We will use agile methodology. The agile approach is focused on giving high priority from the very beginning of the development cycle to customer involvement. Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly. This strategy saves the customer money and time because at each point of production the customer checks and approves the product. If there are problems, then modification can be made during production cycles to fix the issues.

* + 1. Rationale behind Selected Methodology

We selected agile methodology because:

### Easily and Quickly Adapt to Change

* Predictable Delivery Dates

**Chapter 2: Problem Definition**

# Problem Statement

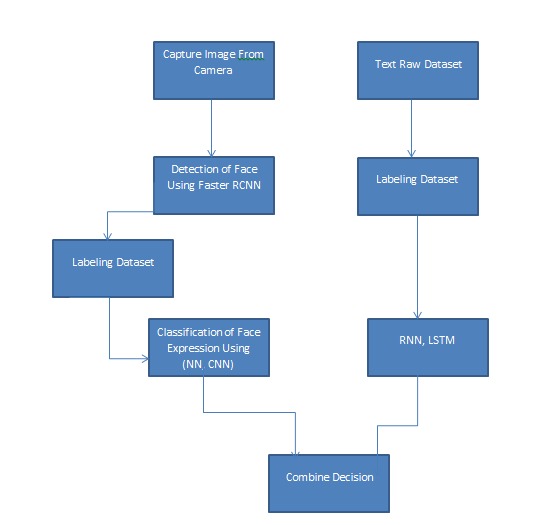
As with intelligence tests, the utility of self-report measures of personality depends on their reliability and construct validity. Some popular measures of personality are not useful because they are unreliable or invalid. In order to facilitate different departments, we are aiming to develop a system that will eradicate such limitations.

# Deliverables and Development Requirements

* Documentation
* User Panel
* Loading Data Panel
* Web System
* Classification System

# Proposed Architecture

Different methods are used to assess a person's character and personality. Some are through the body movements, body language, handwriting and signatures. In contrast to that assessing a person's character and personality using different techniques will be the main objective in order to produce a better system to predict personalities. In OpenCV we'll try face detection which will help in getting threshold of abstracting facial features from an image by introducing DLib. Using the libraries of machine learning and essential OpenCV functions we’d be able to render this vision to life. Computer vision and machine learning form the cornerstones of our approach to modeling and predicting human behavior. Machine learning is a technique in which computers are programmed to modify and improve their performance based on unique data input. In this way, the machines learn the process of learning, autonomously adapting to external change in the environment (i.e., new input factors) and reducing the need for constant re-design.



# 2.4 Operating Environment

* Windows 7 or higher
* Core i3 3rd generation
* 4Gb Ram

**2.5 Assumptions and Dependencies**

No assumptions and dependencies.

# Chapter 3: Requirement Analysis

# Use Cases

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.1** |
| **Use Case Name:** | **Sign up** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User enters all required information in order to get signed up. |
| **Trigger:** | User is intended to Sign-up |
| **Preconditions:** | User must be on the Sign-up page. |
| **Postconditions:** | The user goes to the Login page after the completion of this requirement. |
| **Normal Flow:** | **User Actions:**   1. User enters First Name 2. User enters Last Name 3. User enters Email 4. User enters Password 5. User enters Confirm-Password 6. User clicks the Sign-up Button   **System Responses:**   1. System validates Email 2. System validates Password 3. System validates Confirm-Password 4. System enters data into the Database 5. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **3a.** In step 3 of the normal flow, if the user enters Email  System validates the Email and accepts the format  User moves to the next field  Use Case resumes on step 4 of the normal flow  **3b.** In step 3 of the normal flow, if the User enters Email  System validates Email, rejects the format. Prompts the user to  re-enter the Email  Use Case stays on step 3 of normal flow |
|  | **4a.** In step 4 of the normal flow, if the user enters Password  System validates Password and accepts the format  User moves to the next field  Use Case resumes on step 5 of the normal flow  **4b.** In step 4 of the normal flow, if the User enters Password  System validates Password, rejects the format. Prompts the  user to re-enter the Password  Use Case stays on step 4 of normal flow |
|  | **5a.** In step 5 of the normal flow, if the user enters Confirm-Password  System validates Confirm-Password and matches it with  previously entered Password  User clicks the Sign-up Button  Use Case resumes on step 6 of the normal flow  **5b.** In step 5 of the normal flow, if the user enters Confirm-Password  System validates Confirm-Password and finds it different from  the previously entered Password  Use Case stays on step 5 of the normal flow |
|  | **10a.** In step 10 of the normal flow, if the System stores data  in Database  System verifies the query and entities in the Database  Use Case resumes on step 11 of the normal flow  **10b.** In step 10 of the normal flow, if the System stores data  in Database  System finds ambiguity and is not able to store the data  Use Case stays on step 10 of normal flow |
|  | **11a.** In step 11 of the normal flow, if the System shows Success  Message  User is Signed up  And heads over to the Login Page  **11b.** In step 11 of the normal flow, System shows failure  message it will stay on the Sign-up page again. |
| **Exceptions:** | **3a.** In step 3 of the normal flow, if a user enters an invalid Email  Sign-up is disapproved  Message to user to re-enter a valid Email  Customer enters correct Email  Use Case resumes on step 4 of normal flow |
|  | **4a.** In step 4 of the normal flow, if a user enters an invalid Password  Sign-up is disapproved  Message to user to re-enter a valid Password  User enters correct Password  Use Case resumes on step 5 of normal flow |
|  | **5a.** In step 5 of the normal flow, if a user enters an invalid Password  in the Confirm-Password field  Sign-up is disapproved  Message to user to re-enter a valid Password  User enters correct Password in the Confirm Password field  Use Case resumes on step 6 of normal flow |
| **Includes:** | Login |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User knows the personal information in order to Sign up |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.2** |
| **Use Case Name:** | **Login** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User enters all required information in order to get Logged up. |
| **Trigger:** | User is intended to Login |
| **Preconditions:** | User must be on the Login page and already have signed up |
| **Postconditions:** | The user goes to the main page of application after the completion of this requirement. |
| **Normal Flow:** | **User Actions:**   1. User enters Email 2. User enters Password 3. User clicks the Login Button   **System Responses:**   1. System validates Email 2. System validates Password 3. System verifies data from the Database 4. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **1a.** In step 1 of the normal flow, if the user enters Email  System validates the Email and accepts the format  User moves to the next field  Use Case resumes on step 2 of the normal flow  **1b.** In step 1 of the normal flow, if the User enters Email  System validates Email, rejects the format. Prompts the user to  re-enter the Email  Use Case stays on step 1 of normal flow |
|  | **2a.** In step 2 of the normal flow, if the user enters Password  System validates Password and accepts the format  User moves to the next field  Use Case resumes on step 3 of the normal flow  **2b.** In step 2 of the normal flow, if the User enters Password  System validates Password, rejects the format. Prompts the  user to re-enter the Password  Use Case stays on step 2 of normal flow |
|  | **6a.** In step 6 of the normal flow, if the System verifies data  from Database  System verifies the query and entities in the Database  Use Case resumes on step 7 of the normal flow  **6b.**  In step 6 of the normal flow, if the System verifies data  from Database  System finds ambiguity and is not able to match the data  Use Case stays on step 6 of normal flow |
|  | **7a.** In step 7 of the normal flow, if the System shows Success  Message  User is Logged in  And heads over to the Main Page  **7b.** In step 7 of the normal flow, System shows failure  message it will stay on the Login-up page again. |
| **Exceptions:** | **1a.** In step 1 of the normal flow, if a user enters an invalid Email  Login is disapproved  Message to user to re-enter a valid Email  Customer enters correct Email  Use Case resumes on step 2 of normal flow |
|  | **2a.** In step 2 of the normal flow, if a user enters an invalid Password  Login is disapproved  Message to user to re-enter a valid Password  User enters a valid Password  Use Case resumes on step 3 of normal flow |
| **Includes:** |  |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User knows the Login credentials |
| **Notes and Issues:** | User has to put email in appropriate format. The length of Password must be greater than 4. Password has to be comprised of Lower case letters, Upper case letters and numeric |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.3** |
| **Use Case Name:** | **Logout** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User clicks the logout button |
| **Trigger:** | User is intended to Logout |
| **Preconditions:** | User must be already Logged in |
| **Postconditions:** | Successful logging out leads to desktop application’s Login page and failed logging out remains the user onto the main page. |
| **Normal Flow:** | **User Actions:**   1. User clicks the Logout button   **System Responses:**   1. Message is shown, “Success” or “Failure” |
|  | **2a.** In step 2 of the normal flow, if the System shows Success  Message  User is Logged out  And returns back to the Login page of application  **2b.** In step 2 of the normal flow, System shows failure  message, it will stay on the main page again. |
| **Includes:** |  |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User is aware of Logging out through the Logout button |
| **Notes and Issues:** | None |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.4** |
| **Use Case Name:** | **Loading of Textual Dataset** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User uploads a file into the dataset uploading section |
| **Trigger:** | User is intended to load the textual dataset file |
| **Preconditions:** | User must upload the appropriate file into the required field |
| **Postconditions:** | The dataset must have loaded |
| **Normal Flow:** | **User Actions:**   1. User clicks the browse button to find the file 2. User clicks the upload button to store the file into the   database  **System Responses:**   1. System stores the file into the Database 2. Message is shown, “Success” or “Failure” |
|  | **3a.** In step 3 of the normal flow, if the System stores file  into the database  System verifies the query and entities in the Database  Use Case resumes on step 4 of the normal flow  **3b.** In step 3 of the normal flow, if the System stores file  into the database  System finds ambiguity and is not able to store the file  Use Case stays on step 3 of normal flow |
|  | **4a.** In step 4 of the normal flow, if the System shows Success  Message  Dataset file is loaded  And stays onto the same page  **4b.** In step 4 of the normal flow, System shows failure  Message, System prompts the User to reload the file  it will stay on the same page again. |
| **Exceptions:** | **2a.** In step 2 of the normal flow, if a user uploads an invalid format  Loading of dataset is disapproved  Message to user to re-upload a valid file  User uploads the correct format  Use Case resumes on step 4 of normal flow |
| **Includes:** | None |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User is aware of uploading the dataset file through the upload button |
| **Notes and Issues:** | User must have to upload the appropriate dataset file format |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.5** |
| **Use Case Name:** | **Training of Textual Dataset** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User trains the uploaded dataset file |
| **Trigger:** | User is intended to train the textual dataset file |
| **Preconditions:** | User must have uploaded the textual dataset file |
| **Postconditions:** | The dataset must have been trained |
| **Normal Flow:** | **User Actions:**   1. User clicks the train dataset button to train the dataset file   **System Responses:**   1. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **2a.** In step 2 of the normal flow, if the System shows Success  Message  Model is trained on the Textual dataset  And stays onto the same page  **2b.** In step 2 of the normal flow, System shows failure  Message, System prompts the User to retrain the model  it will stay on the same page again. |
| **Includes:** | None |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User is aware of training the model through the Train model button |
| **Notes and Issues:** | User must have to upload appropriate dataset file format |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.6** |
| **Use Case Name:** | **Feature Extraction using Textual dataset** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User extracts the features from an uploaded dataset file |
| **Trigger:** | User is intended to extract features from the dataset file |
| **Preconditions:** | User must have uploaded the textual dataset file and it must have been trained |
| **Postconditions:** | Features must have been extracted |
| **Normal Flow:** | **User Actions:**   1. User clicks the Extract Features button to extract features from the dataset file   **System Responses:**   1. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **2a.** In step 2 of the normal flow, if the System shows Success  Message  Features are extracted from the Textual dataset file  And stays onto the same page  **2b.** In step 2 of the normal flow, System shows failure  message, System prompts the User to re-extract the features  it will stay on the same page again. |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User is aware of extracting the features using textual datasets by clicking extract features button. |
| **Notes and Issues:** | No. |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.7** |
| **Use Case Name:** | **Loading of Image type Dataset** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User uploads a file into the dataset uploading section |
| **Trigger:** | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] |
| **Preconditions:** | User must upload the appropriate file into the required field |
| **Postconditions:** | The dataset must have loaded |
| **Normal Flow:** | **User Actions:**   1. User clicks the browse button to find the file 2. User clicks the upload button to store the file into the   database  **System Responses:**   1. System stores the file into the Database 2. Message is shown, “Success” or “Failure” |
| **Alternate Flow:** | **3a.** In step 3 of the normal flow, if the System stores file  into the database  System verifies the query and entities in the Database  Use Case resumes on step 4 of the normal flow  **3b.** In step 3 of the normal flow, if the System stores file  into the database  System finds ambiguity and is not able to store the file  Use Case stays on step 3 of normal flow |
|  | **4a.** In step 4 of the normal flow, if the System shows Success  Message  Dataset file is loaded  And stays onto the same page  **4b.** In step 4 of the normal flow, System shows failure  Message, System prompts the User to reload the file  it will stay on the same page again. |
| **Exceptions:** | **2a.** In step 2 of the normal flow, if a user uploads an invalid format  Loading of dataset is disapproved  Message to user to re-upload a valid file  User uploads the correct format  Use Case resumes on step 4 of normal flow |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User is aware of loading image type datasets using browse button. |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.8** |
| **Use Case Name:** | **Training of Image type Dataset** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User trains the uploaded dataset file |
| **Trigger:** | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] |
| **Preconditions:** | User must have uploaded the Image dataset file |
| **Postconditions:** | The dataset must have been trained |
| **Normal Flow:** | **User Actions:**   1. User clicks the train model button to train the dataset file   **System Responses:**   1. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **2a.** In step 2 of the normal flow, if the System shows Success  Message  Model is trained on the Textual dataset  And stays onto the same page  **2b.** In step 2 of the normal flow, System shows failure  Message, System prompts the User to retrain the model  it will stay on the same page again. |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User is aware of training the image datasets using train dataset button. |
| **Notes and Issues:** | The dataset must have been loaded. |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.9** |
| **Use Case Name:** | **Feature Extraction using Image type dataset** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User extracts the features from an uploaded dataset file |
| **Trigger:** | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] |
| **Preconditions:** | User must have uploaded the image type dataset file and it must have been trained |
| **Postconditions:** | Features must have been extracted |
| **Normal Flow:** | **User Actions:**   1. User clicks the Extract Features button to extract features from the dataset file   **System Responses:**   1. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **2a.** In step 2 of the normal flow, if the System shows Success  Message  Features are extracted from the image dataset file  And stays onto the same page  **2b.** In step 2 of the normal flow, System shows failure  message, System prompts the User to re-extract the features  it will stay on the same page again. |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User is aware of extracting the feature using extract feature button. |
| **Notes and Issues:** | The dataset must have been loaded and trained. |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.10** |
| **Use Case Name:** | **Personality Classification using Textual dataset** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | Classifying personalities into different classes based on uploaded dataset file |
| **Trigger:** | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] |
| **Preconditions:** | User must have uploaded the textual dataset file, trained it and extracted features from it |
| **Postconditions:** | Personality type must have been predicted |
| **Normal Flow:** | **User Actions:**   1. User clicks the Prediction button   **System Responses:**   1. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **2a.** In step 2 of the normal flow, if the System shows Success  Message  Personality has been classified from the textual dataset file  And stays onto the same page  **2b.** In step 2 of the normal flow, System shows failure  message, System prompts the User to repeat the same process  it will stay on the same page again. |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User is aware of classifying personality using prediction button. |
| **Notes and Issues:** | The textual dataset must have been loaded, trained and features should have been extracted. |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.11** |
| **Use Case Name:** | **Personality Classification using Image type dataset** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | Classifying personalities into different classes based on uploaded dataset file |
| **Trigger:** | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] |
| **Preconditions:** | User must have uploaded the image dataset file, trained it and extracted features from it |
| **Postconditions:** | Personality type must have been predicted |
| **Normal Flow:** | **User Actions:**   1. User clicks the Prediction button   **System Responses:**   1. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **2a.** In step 2 of the normal flow, if the System shows Success  Message  Personality has been classified from the image dataset file  And stays onto the same page  **2b.** In step 2 of the normal flow, System shows failure  message, System prompts the User to repeat the same process  it will stay on the same page again. |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | User is aware of classifying personality using prediction button. |
| **Notes and Issues:** | The image dataset must have been loaded, trained and features should have been extracted. |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.12** |
| **Use Case Name:** | **Upload Images** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | The user provides the input image for face detection. |
| **Trigger:** | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] |
| **Preconditions:** | The user must be on the Face detection tab |
| **Postconditions:** | The image has been saved successfully or an error occurred. |
| **Normal Flow:** | **User Actions:**   1. User clicks the browse button to choose an image 2. User clicks on Upload button   **System Responses:**   1. System stores the file into the Database 2. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **3a.** In step 3 of the normal flow, if the System stores file  into the database  System verifies the query and entities in the Database  Use Case resumes on step 4 of the normal flow  **3b.** In step 3 of the normal flow, if the System stores file  into the database  System finds ambiguity and is not able to store the file  Use Case stays on step 3 of normal flow |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | [List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.  e.g. For the Withdraw Cash Use Case, an assumption could be:  The Bank Customer understands either English or Spanish language.] |
| **Notes and Issues:** | [List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. e.g.  What is the maximum size of the PIN that a use can have?] |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.13** |
| **Use Case Name:** | **Face Recognition** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | This feature recognizes the faces of the people. |
| **Trigger:** | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] |
| **Preconditions:** | The face should have been detected already. |
| **Postconditions:** | The record is saved. |
| **Normal Flow:** | **User Actions:**   1. User clicks the Open Camera button 2. User faces the camera, and it recognizes the user   **System Responses:**   1. System stores the file into the Database 2. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **3a.** In step 3 of the normal flow, if the System stores file  into the database  System verifies the query and entities in the Database  Use Case resumes on step 4 of the normal flow  **3b.** In step 3 of the normal flow, if the System stores file  into the database  System finds ambiguity and is not able to store the file  Use Case stays on step 3 of normal flow |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | [List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.  e.g. For the Withdraw Cash Use Case, an assumption could be:  The Bank Customer understands either English or Spanish language.] |
| **Notes and Issues:** | [List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. e.g.  What is the maximum size of the PIN that a use can have?] |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.14** |
| **Use Case Name:** | **Training of Images** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User trains the uploaded image file |
| **Trigger:** | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] |
| **Preconditions:** | User must have uploaded the image file |
| **Postconditions:** | The image must have been trained |
| **Normal Flow:** | **User Actions:**   1. User clicks the train model button to train the uploaded images   **System Responses:**   1. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **2a.** In step 2 of the normal flow, if the System shows Success  Message  Model is trained on the Uploaded images  And stays onto the same page  **2b.** In step 2 of the normal flow, System shows failure  Message, System prompts the User to retrain the model  it will stay on the same page again. |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | [List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.  e.g. For the Withdraw Cash Use Case, an assumption could be:  The Bank Customer understands either English or Spanish language.] |
| **Notes and Issues:** | [List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. e.g.  What is the maximum size of the PIN that a use can have?] |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.15** |
| **Use Case Name:** | **Feature Extraction using Uploaded Images** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | User extracts the features from uploaded images |
| **Trigger:** | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] |
| **Preconditions:** | User must have uploaded the image type dataset file and it must have been trained |
| **Postconditions:** | Features must have been extracted |
| **Normal Flow:** | **User Actions:**   1. User clicks the Extract Features button to extract features from the uploaded image(s)   **System Responses:**   1. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **2a.** In step 2 of the normal flow, if the System shows Success  Message  Features are extracted from the image file  And stays onto the same page  **2b.** In step 2 of the normal flow, System shows failure  message, System prompts the User to re-extract the features  it will stay on the same page again. |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | [List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.  e.g. For the Withdraw Cash Use Case, an assumption could be:  The Bank Customer understands either English or Spanish language.] |
| **Notes and Issues:** | [List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. e.g.  What is the maximum size of the PIN that a use can have?] |

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1.2.16** |
| **Use Case Name:** | **Personality Classification using Uploaded Images** |
| **Actors:** | **Primary Actor:** Administrator  **Secondary Actor:** None |
| **Description:** | Classifying personalities into different classes based on uploaded Images |
| **Trigger:** | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] |
| **Preconditions:** | User must have uploaded the image file, trained it and extracted features from it |
| **Postconditions:** | Personality type must have been predicted |
| **Normal Flow:** | **User Actions:**   1. User clicks the Prediction button   **System Responses:**   1. Message is shown, “Success” or “Failure” |
| **Alternative Flows:** | **2a.** In step 2 of the normal flow, if the System shows Success  Message  Personality has been classified from the Uploaded image file  And stays onto the same page  **2b.** In step 2 of the normal flow, System shows failure  message, System prompts the User to repeat the same process  it will stay on the same page again. |
| **Includes:** | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] |
| **Special Requirements:** | There are no special requirements associated with this use case. |
| **Assumptions:** | [List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.  e.g. For the Withdraw Cash Use Case, an assumption could be:  The Bank Customer understands either English or Spanish language.] |
| **Notes and Issues:** | [List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. e.g.  What is the maximum size of the PIN that a use can have?] |

**3.2. FUNCTIONAL REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **1** | **Title:** | **Sign up** |
| Description: | | User enters all required information in order to get signed up. |
| Input: | | User enters First Name, Last Name, Email, Password, Confirm Password. |
| Output: | | Display Success or Failed message. |
| Pre-condition: | | User must be on the Signup page. |
| Post-condition: | | The user goes to the Login page after the completion of this requirement. |
| Action: | | Input all the information in the required fields and clicks the “Sign up” button. |
| Destination: | | Login page |
| Source: | | Keyboard, Mouse |

|  |  |  |
| --- | --- | --- |
| **2** | **Title:** | **Login** |
| Description: | | User enters the required credentials in order to get Logged in. |
| Input: | | Email, Password |
| Output: | | Display Success or Failed message |
| Pre-condition: | | User must have already signed up |
| Post-condition: | | Successful logging in leads to desktop application’s main page and failed logging in turns the user back to the login page. |
| Action: | | User enters the required information in the fields and clicks the “Login up” button |
| Destination: | | Desktop Application |
| Source: | | Keyboard, Mouse |

|  |  |  |
| --- | --- | --- |
| **3** | **Title:** | **Logout** |
| Description: | | User must Logout from the system |
| Input: | | User clicks on Logout Button |
| Output: | | Display Success or Failed message |
| Pre-condition: | | User must be already Logged in |
| Post-condition: | | Successful logging out leads to desktop application’s Login page and failed logging out remains the user onto the main page. |
| Action: | | Log out |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **4** | **Title:** | **Loading of Textual dataset** |
| Description: | | User uploads a file into the dataset uploading section |
| Input: | | Textual dataset file |
| Output: | | Display Success or Failed message |
| Pre-condition: | | User must upload the appropriate file into the required field |
| Post-condition: | | The dataset must have loaded |
| Action: | | User chooses the textual dataset file and clicks the “Upload” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **5** | **Title:** | **Training of Textual dataset** |
| Description: | | User trains the uploaded dataset file |
| Input: | | User clicks the train button |
| Output: | | Display Success or Failed message |
| Pre-condition: | | User must have uploaded the textual dataset file |
| Post-condition: | | The dataset must have been trained |
| Action: | | User clicks the “Train model” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **6** | **Title:** | **Loading of Image dataset** |
| Description: | | User uploads a file into the dataset uploading section |
| Input: | | Image type dataset file |
| Output: | | Display Success or Failed message |
| Pre-condition: | | User must upload the appropriate file into the required field |
| Post-condition: | | The dataset must have loaded |
| Action: | | User chooses the Image dataset file and clicks the “Upload” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **7** | **Title:** | **Training of Image type dataset** |
| Description: | | User trains the uploaded dataset file |
| Input: | | User clicks the train button |
| Output: | | Display Success or Failed message |
| Pre-condition: | | User must have uploaded the image dataset file |
| Post-condition: | | The dataset must have trained |
| Action: | | User clicks the “Train dataset” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **8** | **Title:** | **Feature Extraction using Textual dataset** |
| Description: | | User extracts the features from an uploaded dataset file |
| Input: | | User clicks the extract features button |
| Output: | | Display Success or Failed message |
| Pre-condition: | | User must have uploaded the textual dataset file and it must have been trained |
| Post-condition: | | Features must have been extracted |
| Action: | | User clicks the “Features Extraction” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **9** | **Title:** | **Feature Extraction using Image dataset** |
| Description: | | User extracts the features from the dataset file |
| Input: | | User clicks the extract features button |
| Output: | | Display Success or Failed message |
| Pre-condition: | | User must have uploaded the Image dataset and it must have been trained |
| Post-condition: | | Features must have been extracted |
| Action: | | User clicks the “Features Extraction” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **10** | **Title:** | **Personality Classification using Textual dataset** |
| Description: | | Classifying personalities into different classes based on uploaded dataset file |
| Input: | | User clicks the prediction button |
| Output: | | Display the accuracy of trained model. |
| Pre-condition: | | User must have uploaded the textual dataset file, trained it and extracted features from it |
| Post-condition: | | Personality type must have been predicted |
| Action: | | User clicks the “Prediction” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **11** | **Title:** | **Upload images** |
| Description: | | The user provides the input image for face detection. |
| Input: | | Input images. |
| Output: | | The images successfully got stored into the database. |
| Pre-condition: | | The user must be on the Face detection tab |
| Post-condition: | | The image has been saved successfully or an error occurred. |
| Action: | | Click the “Upload image**”** button. |
| Destination: | | Desktop application screen. |
| Source | | Keyboard, Mouse |

|  |  |  |
| --- | --- | --- |
| **12** | **Title:** | **Face Recognition** |
| Description: | | This feature recognizes the faces of the people. |
| Input: | | Real-Time images of the people and input static images of the specific person. |
| Output: | | Recognize the face of the person. |
| Pre-condition: | | The face should have been detected already. |
| Post-condition: | | The record is saved. |
| Action: | | Click On “Face Login” button. |
| Destination: | | Screen, Face of the person. |
| Source: | | Keyboard, Mouse, GUI camera |

|  |  |  |
| --- | --- | --- |
| **13** | **Title:** | **Training of Images** |
| Description: | | User trains the uploaded image file |
| Input: | | User clicks the train button |
| Output: | | Display Success or Failed message |
| Pre-condition: | | User must have uploaded the image file |
| Post-condition: | | The image must have trained |
| Action: | | User clicks the “Train Image” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **14** | **Title:** | **Feature Extraction using Uploaded Images** |
| Description: | | User extracts the features from an uploaded Image file |
| Input: | | User clicks the extract features button |
| Output: | | Display Success or Failed message |
| Pre-condition: | | User must have uploaded the Image file and it must have been trained |
| Post-condition: | | Features must have been extracted |
| Action: | | User clicks the “Features Extraction” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **15** | **Title:** | **Personality Classification using Uploaded Image** |
| Description: | | Classifying personalities into different classes based on uploaded image |
| Input: | | User clicks the prediction button |
| Output: | | Display the accuracy of trained model. |
| Pre-condition: | | User must have uploaded the image, trained it and extracted features from it |
| Post-condition: | | Personality must have been classified |
| Action: | | User clicks the “Prediction” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

|  |  |  |
| --- | --- | --- |
| **16** | **Title:** | **Personality Classification using Image type dataset** |
| Description: | | Classifying personalities into different classes based on uploaded dataset file |
| Input: | | User clicks the compute results button |
| Output: | | Display the accuracy of trained model. |
| Pre-condition: | | User must have uploaded the textual dataset file, trained it and extracted features from it |
| Post-condition: | | Personality must have been classified |
| Action: | | User clicks the “Prediction” button |
| Destination: | | Desktop Application |
| Source: | | Mouse |

* 1. **Non-Functional Requirements**

### Usability

Our system basically defines a personality prediction system which is very user friendly and any user can understand its functionality in just 4 to 5 minutes.

### Reliability

Our system is reliable and accurately predicts the personality in 85% cases; however, it may need upgradation in future.

### Integrity

We use authentication and validation to the email and password of user account to make them secure. It can be tested, rendering error free results. It lacks security vulnerabilities which is the most important aspect of a system. It is easy to edit and upgrade without introducing new errors.

### 3.3.4 Performance

Our system performs well without any lag. It is highly responsive, has an efficient throughput, execution time is less than 5 minutes.

**3.4 EXTERNAL INTERFACE REQUIREMENTS**

* **Hardware:**

None

* **Database:**

MySQL

**3.5 PERFORMANCE REQUIREMENTS**

The system is very much interactive, and the delays involved are for sure less. So in every action-response of the system, there are no immediate delays. In case of opening windows forms, of popping error messages and saving the settings or sessions there is delay much below 2 seconds. It also will not take any longer for loading and training the datasets. The system is highly secure, safe and reliable.

**3.6 CHAPTER SUMMARY**

In this chapter we learnt what functionality our system will have once completed and how would it impact the performance. We discussed in detail the methods/functions our system will perform and how will they work in accordance with the above requirements. We also discussed non-functional requirement and what impact our proposed requirements will have on the overall project during run time.

**USE CASE MODEL**

Diagram

Description automatically generated

**INTERFACES:**

**Diagram, text

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface

Description automatically generated**

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

# 

# Diagram Description automatically generated with medium confidence

# Graphical user interface, text Description automatically generated

# Chapter 4: Design and Architecture

[**4.1 UML Structural Diagrams**](#_Toc268523830)

* + 1. [**System Component Diagram**](#_Toc268523787)

Diagram

Description automatically generated

[**4.1.2 Deployment Diagram**](#_Toc268523787)

Diagram

Description automatically generated

[**4.2 UML Behavioral Diagrams**](#_Toc268523830)

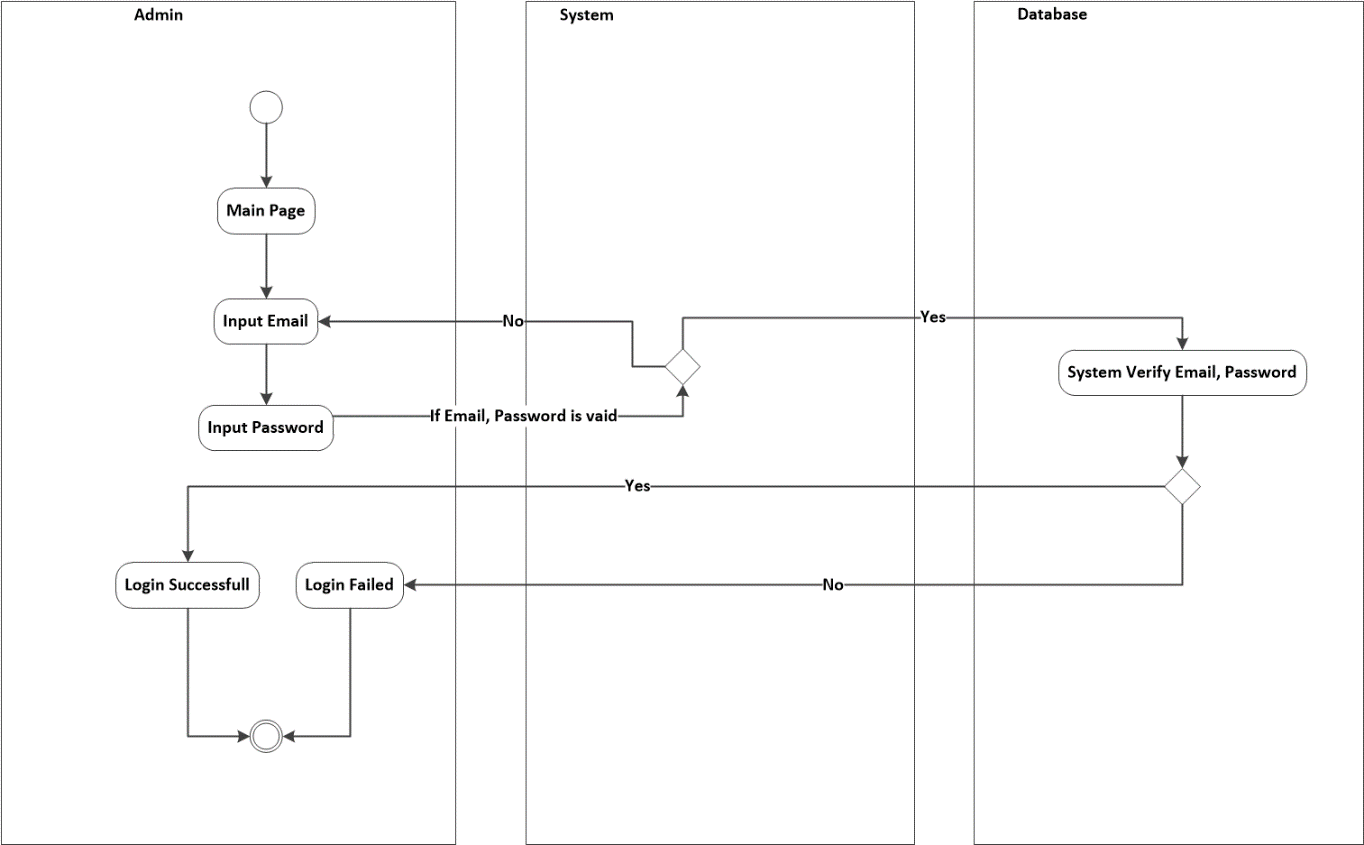
* + 1. [**Activity Diagrams**](#_Toc268523787)

**Sign Up:**

Diagram

Description automatically generated

**Log In:**

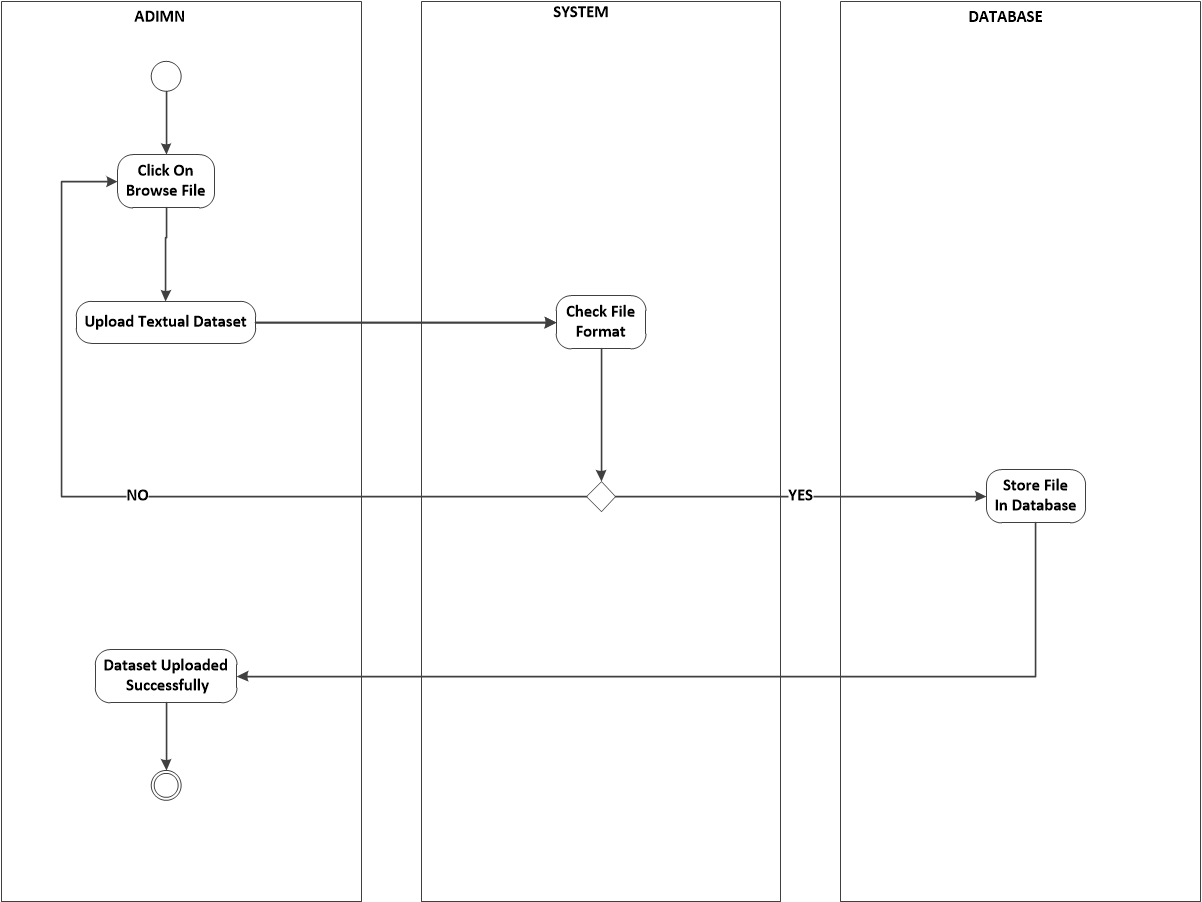


**Log Out:**

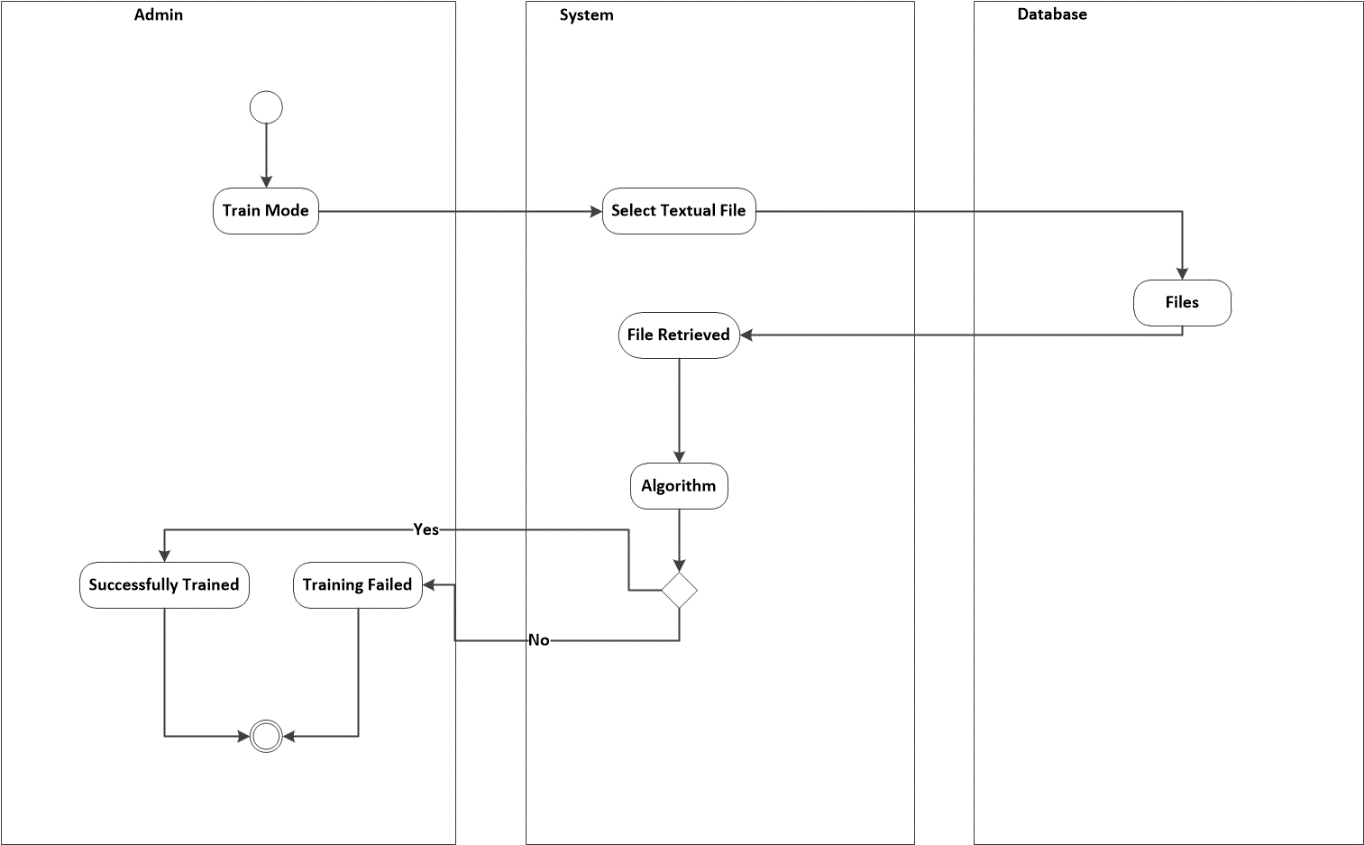
Diagram

Description automatically generated

**Loading of Textual Dataset**



**Training of Textual Dataset :**



**Loading of Image Dataset:**

**Diagram, schematic

Description automatically generated**

**Training of Image Type Dataset:**

Diagram, schematic

Description automatically generated

**Feature Extraction Using Textual Dataset:**

Diagram, schematic

Description automatically generated

**Feature Extraction Using Image Dataset:**

Diagram, schematic

Description automatically generated

**Personality Classification Using Textual Dataset:**

Diagram

Description automatically generated

**Upload Images:**

Diagram, schematic

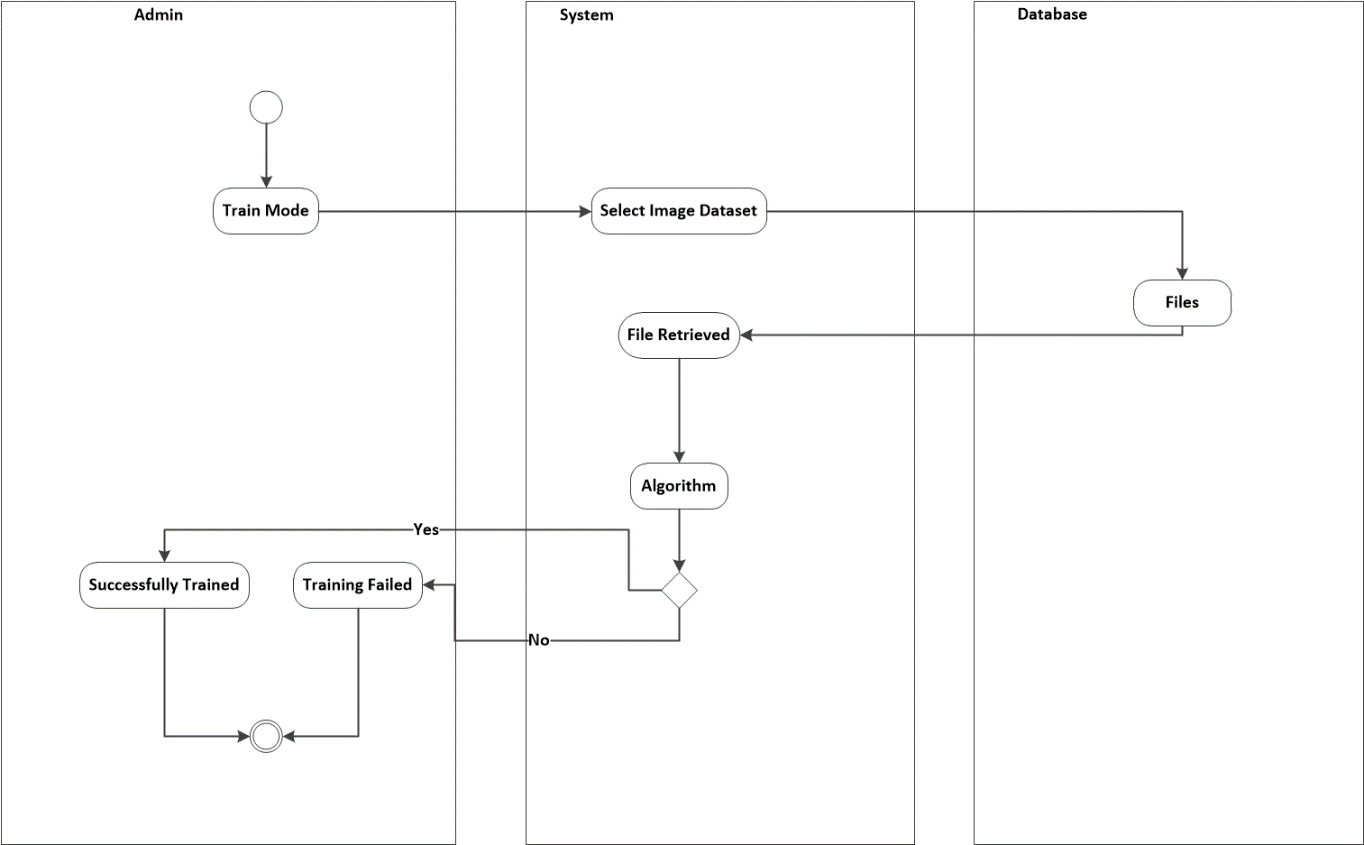
Description automatically generated

**Face Recognition:**

A picture containing diagram

Description automatically generated

**Training of Images**



**Feature Extraction using Uploaded Images**

Diagram, schematic

Description automatically generated

**Personality Classification using Uploaded Image:**

**Diagram, schematic

Description automatically generated**

**Personality Classification Using Image Type Dataset:**

**Diagram

Description automatically generated**

[**4.2.2 State Machine Diagrams**](#_Toc268523787)

[**4.3 UML Interaction Diagrams**](#_Toc268523830)

* + 1. [**Sequence Diagrams**](#_Toc268523787)

**Sign Up:**

Diagram

Description automatically generated

**Log In:**

**Graphical user interface

Description automatically generated**

**Log Out:**

A picture containing text

Description automatically generated

**Loading of Textual Dataset**

**Graphical user interface

Description automatically generated**

**Training of Textual Dataset:**

**Diagram

Description automatically generated**

**Loading of Image Dataset:**

**Graphical user interface

Description automatically generated**

**Training of Image Type Dataset:**

Diagram

Description automatically generated

**Feature Extraction Using Textual Dataset:**

Diagram

Description automatically generated

**Feature Extraction Using Image Dataset:**

Diagram

Description automatically generated

**Personality Classification Using Textual Dataset:**

A picture containing text

Description automatically generated

**Upload Images:**

Diagram

Description automatically generated

**Face Recognition:**

Diagram

Description automatically generated

**Training of Images**

Diagram

Description automatically generated

**Feature Extraction using Uploaded Images**

Diagram

Description automatically generated

**Personality Classification using Uploaded Image:**

**A picture containing text

Description automatically generated**

**Personality Classification Using Image Type Dataset:**

**A picture containing text

Description automatically generated**

# Chapter 5: Implementation

# Tools and Technologies

**Spyder(Anaconda)**

Spyder is an open-source cross-platform integrated development environment for scientific programming in the Python language.

We used Spyder because it uses Qt for its GUI and is designed to use either of the PyQt or PySide Python bindings.

And we also used Spyder because it is suitable for scientific programming in Python, data science and machine learning.

**QtDesigner**

Qt Designer is a Qt tool that provides us with a what-you-see-is-what-you-get user interface to create GUIs for our PyQt applications productively and efficiently. With this tool, we can create GUIs by dragging and dropping QWidget objects on an empty form.

And we can arrange them into a coherent GUI using different layout managers.

**Wamp Server**

It Stands for "Windows, Apache, MySQL, and PHP."

We used Wamp server to load datsets into database and also to store login information of Users and Admin.

It runs as a process in Apache and communicates with the MySQL database.

**MS WORD**

Microsoft Word is a graphical word processing program that we can type with. Its purpose is to allow us to type and save documents.

We used MS WORD to type the Documentation of our project.

It has advanced features which allow us to format and edit our files and documents in the best possible way.

**MS PowerPoint**

PowerPoint is used to create presentations. The presentations are comprised of slides, which may contain text, images, and other media.

The purpose of PowerPoint is to act as a visual aid as we go along presenting their option, ideas, etc.

We used PowerPoint to present our project idea and documentation in charts and visual form

While the evaluation of project.

**Python 3.8**

Python is a general-purpose programming language. Unlike HTML, CSS, and JavaScript, it can be used for other types of programming and software development besides web development..

We used Python because it is the best fit for machine learning, deep learning and AI-based projects.

We used it because of consistency and access to great libraries and frameworks for machine learning.

**PyQt5**

PyQt5 is one of the most used modules in building GUI apps in Python.

Our main reason of using PyQt5 is the PyQt5 designer, which makes it so easy to develop complex GUI apps in a short time. We just have to drag widgets to build our form.

**TensorFlow**

It is an open source artificial intelligence library, using data flow graphs to build models.

We used TensorFlow mainly for: Classification, Perception, Understanding, Discovering, Prediction and Creation

**OpenCV**

OpenCV is a cross-platform library using which we can develop real-time computer vision applications.

We used It because it mainly focuses on image processing and analysis including features like face detection and object detection.

**MySQL Connector**

We used mySQL Connector to enable python program to access mysql database. Because python needs a driver to access mysql database. This is necessary because each make of database server has its own specific protocol for transporting requests to, and results from, the server to application programs.

# Chapter 6: Testing and Evaluation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **Test Case 1** | | | | | | |
| **Test Case ID:** SPA\_01 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Login | | | |  | | |
| **Test Title:** Test Login | | | |  | | |
| **Description:** User enters all required information to get logged in | | | |  | | |
| **Pre-Condition:**  User must be on login page and must be signed up | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User enters email | Registered email | System validates the Email | Success | pass |  |
| **2** | User enters password | Registered password | System validates the password | Success | pass |  |
| **3** | User clicks login button |  | Login successful | Success | Pass |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **Test Case 2** | | | | | | |
| **Test Case ID:** SPA\_02 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Login | | | |  | | |
| **Test Title:** Test Login | | | |  | | |
| **Description:** User enters all required information to get logged in | | | |  | | |
| **Pre-Condition:**  User must be on login page and must be signed up | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User enters email | Registered email | System validates the Email | Success | pass |  |
| **2** | User enters password | Registered password | System validates the password | Success | Fail | Wrong Password |
| **3** | User clicks login button |  | Login successful | Login Failed | Fail | Re-enter the password |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **Test Case3** | | | | | | |
| **Test Case ID:** SPA\_03 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Sign Up | | | |  | | |
| **Test Title:** Test Sign Up | | | |  | | |
| **Description:** User enters all the required information to get Signed Up | | | |  | | |
| **Pre-Condition:** User must be on the Sign Up page | | | |  | | |
| **Dependencies:** Admin must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User enters the first name | Hammad | System stores the data | Success | Pass | None |
| **2** | User enters the last name | Imran | System stores the data | Success | Pass | None |
| **3** | User enters the email | hr@123.com | System Validates the email | Success | Pass | Must be according to given format |
| **4** | User enters the password | 123456 | System Validates the password | Success | Pass |  |
| **5** | User enters the confirm password | 123456 | System Validates the password | Success | Pass |  |
| **6** | User clicks the Sign Up button | System enters data into database | Record stored | Success | Pass |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **Test Case4** | | | | | | |
| **Test Case ID:** SPA\_04 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Sign Up | | | |  | | |
| **Test Title:** Test Sign Up | | | |  | | |
| **Description:** User enters all the required information to get Signed Up | | | |  | | |
| **Pre-Condition:** User must be on the Sign Up page | | | |  | | |
| **Dependencies:** Admin must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User enters the first name | Hammad | System stores the data | Success | Pass | None |
| **2** | User enters the last name | Imran | System stores the data | Success | Pass | None |
| **3** | User enters the email | hr@123.com | System Validates the email | Email is not valid | Fail | Must be according to given format |
| **4** | User enters the password | 123456 | System Validates the password | Success | Pass |  |
| **5** | User enters the confirm password | 123455 | System Validates the password | Password does not match | Fail | Confirm password should matches the original password |
| **6** | User clicks the Sign Up button | System enters data into database | Record stored | Fix the errors above | Fail |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **Test Case 5** | | | | | | |
| **Test Case ID:** SPA\_05 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Loading Of Textual Dataset | | | |  | | |
| **Test Title:** Test Load Textual Dataset | | | |  | | |
| **Description:** User uploads a file into Dataset Uploading section | | | |  | | |
| **Pre-Condition:**  User must upload a appropriate file into required field | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User clicks the browse button to find the file |  | Open Files in PC | Success | Pass |  |
| **2** | User clicks the upload button to store file into database | Textual dataset | System stores the file into database | Success | Pass | User must upload appropriate file |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **Test Case 6** | | | | | | |
| **Test Case ID:** SPA\_06 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Loading Of Textual Dataset | | | |  | | |
| **Test Title:** Test Load Textual Dataset | | | |  | | |
| **Description:** User uploads a file into Dataset Uploading section | | | |  | | |
| **Pre-Condition:**  User must upload a appropriate file into required field | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User clicks the browse button to find the file |  | Open files in PC | Success | Pass |  |
| **2** | User clicks the upload button to store file into database | Textual dataset | System stores the file into database | Inappropriate File Format | Fail | Invalid File Format |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **Test Case 7** | | | | | | |
| **Test Case ID:** SPA\_07 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Training Of Textual Dataset | | | |  | | |
| **Test Title:** Test Train Textual Dataset | | | |  | | |
| **Description:** User trains the uploaded dataset file | | | |  | | |
| **Pre-Condition:** User must have uploaded the textual dataset file | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User clicks the train dataset button to train the dataset file | Textual Dataset | Success | Success | Pass | Model is trained on textual dataset |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **Test Case 8** | | | | | | |
| **Test Case ID:** SPA\_08 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Training Of Textual Dataset | | | |  | | |
| **Test Title:** Test Train Textual Dataset | | | |  | | |
| **Description:** User trains the uploaded dataset file | | | |  | | |
| **Pre-Condition:** User must have uploaded the textual dataset file | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User clicks the train dataset button to train the dataset file | Textual Dataset | Success | Fail | Fail | Invalid Format/Retrain  Model |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **Test Case 9** | | | | | | |
| **Test Case ID:** SPA\_09 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Feature Extraction Using Textual Dataset | | | |  | | |
| **Test Title:** Test Extract Features of Textual Dataset | | | |  | | |
| **Description:** User extracts the features from uploaded dataset file | | | |  | | |
| **Pre-Condition:** User must have uploaded the textual dataset file and it must have been trained. | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User clicks the extract features button to extract the features from the dataset file | Textual Dataset | Success | Success | Pass | Features are extracted from the textual dataset |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **Test Case10** | | | | | | |
| **Test Case ID:** SPA\_07 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Feature Extraction Using Textual Dataset | | | |  | | |
| **Test Title:** Test Extract Features of Textual Dataset | | | |  | | |
| **Description:** User extracts the features from uploaded dataset file | | | |  | | |
| **Pre-Condition:** User must have uploaded the textual dataset file and it must have been trained. | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User clicks the extract features button to extract the features from the dataset file | Textual Dataset | Features Extracted | Fail | Fail | Re extract the features |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **TestCase11** | | | | | | |
| **Test Case ID:** SPA\_11 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Personality Classification Using Textual Dataset | | | |  | | |
| **Test Title:** Test Personality Classification | | | |  | | |
| **Description:** Classifying Personalities into different classes based on uploaded dataset file | | | |  | | |
| **Pre-Condition:**  User must have uploaded the textual dataset file, trained it and extracted features from it. | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User clicks the prediction Button | Textual Dataset | Personality type predicted | Successfully predicted the personality | Pass | Dataset must be trained and feature should be extracted |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **TestCase12** | | | | | | |
| **Test Case ID:** SPA\_12 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Personality Classification Using Textual Dataset | | | |  | | |
| **Test Title:** Test Personality Classification | | | |  | | |
| **Description:** Classifying Personalities into different classes based on uploaded dataset file | | | |  | | |
| **Pre-Condition:**  User must have uploaded the textual dataset file, trained it and extracted features from it. | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User clicks the prediction Button | Textual Dataset | Personality type predicted | Failed to predict the personality | Fail | User must retrain model and re extract features. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Personality Prediction System Using Deep Learning (Desktop App)** | | | | | | |
| **TestCase13** | | | | | | |
| **Test Case ID:** SPA\_13 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | |  | | |
| **Module Name:** Face Recognition | | | |  | | |
| **Test Title:** Test face recognition | | | |  | | |
| **Description:** This feature recognizes the faces of people | | | |  | | |
| **Pre-Condition:**  The face should have been detected already | | | |  | | |
| **Dependencies:** User must be Login first. | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User clicks the open camera button |  | Open camera | Success | Pass |  |
| **2** | User faces the camera it recognizes the user | Saved image | Image stored in database | Success | Pass | None |

# Chapter 7: Conclusion and Future Work

# Conclusion

This desktop application predicts personalities using both techniques natural language processing as well as image processing. The user signs up or logs in to go through this application either by putting text or images to generate results against the text or images provided.

# Future Work

There is always a need for improvements, adding more features and extending every project. This project also holds the readiness to be improved further in the future.

* Using different Datasets.
* Different feature extractors could be used to build more accurate vectors.
* Classifiers could be changed for comparison to generate more accurate and better results this way accuracies could also be improved.

# References

1. Katiyar, Sandhya, Himdweep Walia, and Sanjay Kumar. "Personality Classification System using Data Mining." 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO). IEEE, 2020.
2. Genina, Alàa, Mariam Gawich, and Abdelfatah Hegazy. "An Approach for Sentiment Analysis and Personality Prediction Using Myers Briggs Type Indicator." International Conference on Advanced Intelligent Systems and Informatics. Springer, Cham, 2020.
3. Gunes, Hatice, Oya Celiktutan, and Evangelos Sariyanidi. "Live human–robot interactive public demonstrations with automatic emotion and personality prediction." Philosophical Transactions of the Royal Society B 374.1771 (2019): 20180026.