Faculty of Technology and Engineering
The Maharaja Sayajirao University of Baroda

SOFTWARE REQUIREMENT SPECIFICATIONS

FOR

FACE IDENTIFICATION SYSTEM

Version 1.0

(Approved)

Prepared By:

Jainam Shah(539)
Sameer Amesara(535)
Sohamkumar Patel(534)

Submitted To:

Ms. Bela Shah / Ms. Bhumi

INDEX

INTRODUCTION TO FACE IDENTIFICATION SYSTEM

1-1) Aim & Objective:

This project is intended to identify a person using the images previously taken. The identifications can be done on the basis of images stored in the form of slices of head, eyes, lips, DNA, nose etc.

1-2) Problem Statement:

The development of a reliable program to perform Face Identification of Criminal is still a challenge today. Many had underestimated the complexity of this problem as identification of face is a process that can be easily done by Human eyes. However, for a computer system to model and imitate the face of criminal, there are many challenges involved. One of the challenges faced in Face Identification is poor resolution of the images of the criminals and are hard to identify. Another thing which can hamper the identification of criminal is poor knowledge of eyewitness and the ability of operator to imitate the face of criminal.

Criminal record generally contains personal information about particular person along with photograph. To identify any Criminal we need some identification regarding person, which are given by eyewitness. Identification can be done in many ways like finger print, eyes, DNA etc. One of the applications is face identification. The face is our primary focus of attention in social inter course playing a major role in conveying identifications and emotion. Although the ability to infer intelligence or character from facial appearance is suspect, the human ability to recognize face is remarkable.

1-3) Project Scope:

The scope of the project is confined to store the image and store in the database. When a person has to be identified on the basis of the images stored in the database which are compared with the existing details.

1-4) Overview of the Project:

This project is aimed to identify the criminals in any investigation department. Here the technique is we already store some images of the criminals in our database along with his details and those images are segmented into many slices say eyes, hairs, lips, nose, etc. These images are again stored in another database record so to identify any criminals; eyewitnesses will see the images or slices that appear on the screen by using it we develop the face, which may or may not be matched with our images. If any image is matched up to 99% then we predict that he is only the criminal. Thus using this project it provides a very friendly environment for both operator and eyewitness to easily design any face can identify criminals very easy.

1-5) Existing System:

This system is manual system only. Here, have a facility to store the criminal images. If you want to compare the criminal images with the existing images it is manual process. This process is very slow to give the result. It is very critical to find the criminal images.

1-6) Proposed System:

To overcome the drawbacks that were in the existing system we develop a system that will be very useful for any investigation department. Here the program keeps track of the record number of each slice during the construction of identifiable human face and calculate maximum number of slices of the similar record number. Based on this record number the program retrieves the personal record of the suspect (whose slice constituted the major parts of the constructed human face) on exercising the “locate” option.

1-7) Advantages:

- Very fast and accurate & easy to find the criminals.
- No need of any extra manual effort.
- No fear of data loss & doesn’t require any extra hardware device.
- Just need a little knowledge to operate the system.

2-1) Functional Requirements:

- Insert the image into database.
  - **Input**: Image of the criminal will be inserted with the required details.
  - **Output**: Image will be shown in our database
  - **Processing**: Image will be converted to a data entry and will be stored in database.
- Split the image into no of parts.
  - **Input**: One Image will be taken out from the database and would be divided into 4 parts comprising of head, eyes, nose, and lips.
  - **Output**: We would get 4 different images of one criminal which will be stored in the database.
  - **Processing**: Images would be cropped into 4 parts such that the lips, eyes, head and nose of the criminal are separated.
- Merge the parts.
  - **Input**: The parts of the criminal identified by the eyewitness would be merged in our software.
  - **Output**: Final image of the criminal will be created by our software and would be shown in the database under Witnessed Criminals.
  - **Processing**: Images of head, eyes, nose and lips as told by the witness would be merged.
- Identify the image.
  - **Input**: The final image stored under the Witnessed Criminal will be shown to the eyewitness.
  - **Output**: The eyewitness would identify the criminal and would tell if the image resembles the actual criminal.
  - **Processing:** The eyewitness would see the final image and tell the admin if the image resembles the actual criminal.
- Draw image manually.
  - **Input:** The image finalized by the eyewitness would be given to sketch artist to make the final image of the criminal.

  - **Output:** The final image drawn by the sketch artist would be stored under the Criminals section in the database.
  - **Processing:** Final image would be stored by the admin in the database.
- Maintain separate login for admin and operator.
  - **Input**: Separate login for admin and the eyewitness would be done.
  - **Output**: Separate account would be created for admin and eyewitness.
  - **Processing**: The admin and eyewitness would have to create their respective accounts.
- Maintain information about each criminal
  - **Input**: The admin would maintain the database for the criminals, eye-witnesses, and Actual criminals.
  - **Output**: The complete database of all the users, suspected criminals and actual criminals.
  - **Processing**: The admin will enter the data into the database.

1-2) Non-functional Requirements:

- Reliability
- Security
- Testing
- Usability
- Performance Requirements
- Human-Computer Interface
- Maintainability & Reusability

1-3) Constraints:

- This system would not work on twin brothers/sisters.
- The system would not work properly if there are very less criminal records.
- The efficiency of this system would also depend upon the eyewitness' memory power.
- The efficiency would be limited by the ability of sketch artist to draw the image of actual criminal.

1-4) Module Description:

Well-structured designs improve the maintainability of a system. A structured system is one that is developed from the top down and modular, that is, broken down into manageable components. In this project we modularized the system so that they have minimal effect on each other.

This application is designed into five independent modules which take care of different tasks efficiently.

1. User Interface Module:

Actually every application has one user interface for accessing the entire application. In this application also we are providing one user interface for accessing this application. The user interface designed completely based on the end users. It is provide friendly accessing to the users. This user interface has attractive look and feel. Technically I am using the swings in core java for preparing this user interface.

2. Admin Module:

User requirements	Elaboration	Further Elaboration
Create	Assign new user id & password for an employee.	
Delete	Administrator can delete the user id & password of unwanted employee.	
Update	First the details of employees are to be obtained by using user id & password.	After obtaining the original details the updated details are submitted.

3. Client Module:

User requirements	Elaboration	Further Elaboration
Login	Employee login to home page by entering id & password.	
Adding details	Personal details of criminal store in to data base	Images are cropped and saved in database.
Update process	Enter criminal id and obtain his details	Update the details and images of existing criminal
Delete process	Enter criminal id	Delete the details and image of unwanted criminal
Logout	Logout in to the home page	

4. Splitting and Merging Module:

Requirements	Elaboration	Further Elaboration
View clippings	View all clips and select the clip shown by eyewitness	Compare the clippings with images of criminals
Construction	Construct the face of criminal by clubbing all freeze clippings	

5. Database Operations Module:

ADD MODULE: The add module is helpful in adding the details of the criminals along with the details of the criminal photo. While adding the details of the criminal, we crop the image of the criminal and store those cropped parts in a separate database.

DELETE MODULE : This module deletes the criminal details along with the photo. The operator first submits the criminal id and searches for the availability of the id in the database. If that id is available in the database, then the operator may delete the record of that particular r criminal.

UPDATE MODULE : The operator first enters the criminal id and searches for the availability of that id .If that id is available in the database , then the details of that criminal are retrieved and the operator can update the details of that criminal and that updated details of the criminal are stored in the database again for future retrieval.

6. Identify Module:

The cropped parts of the criminals, along with the criminal Id are viewed by the eyewitness .The eyewitness selects particular cropped part of the criminal and it is freeze by the operator., then complete face of the criminal is constructed and the details of the criminal is retrieved.

SOFTWARE DEVELOPMENT LIFE CYCLE (SLDC)

3-1) SDLC METHDOLOGIES:

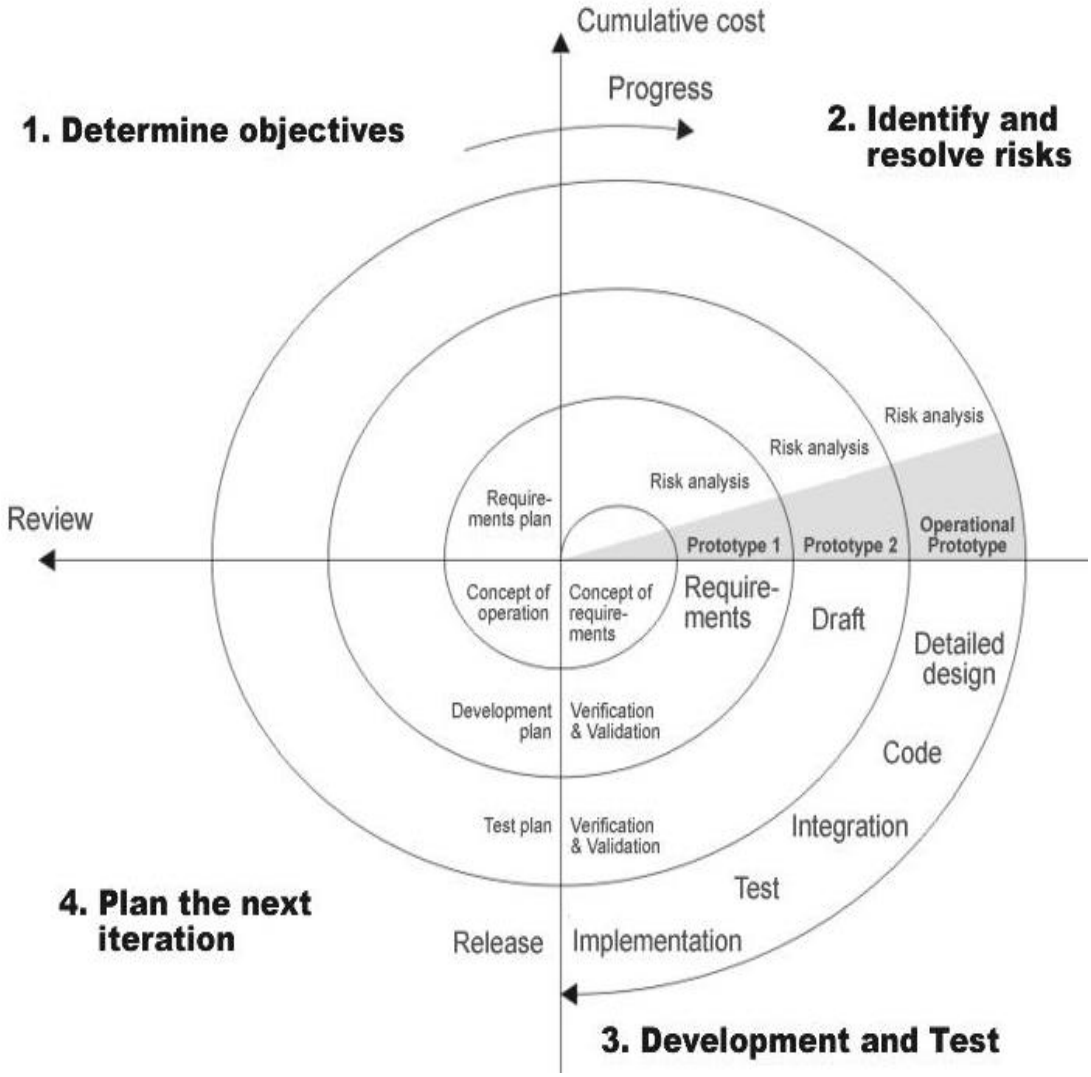
This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

Spiral Model is a combination of a waterfall model and iterative model. Each phase in spiral model begins with a design goal and ends with the client reviewing the progress. The spiral model was first mentioned by Barry Boehm in his 1986 paper.

The development team in Spiral-SDLC model starts with a small set of requirement and goes through each development phase for those set of requirements. The software engineering team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase.

3-2) Spiral Model Phases:

Planning	• It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer
Risk Analysis	• Identification of potential risk is done while risk mitigation strategy is planned and finalized
Engineering	• It includes testing, coding and deploying software at the customer site
Evaluation	• Evaluation of software by the customer. Also, includes identifying and monitoring risks such as schedule slippage and cost overrun



3-3) Reasons to Use Spiral Model:

- High amount of risk analysis hence, avoidance of Risk is enhanced.
- Good for large and mission-critical projects.
- Strong approval and documentation control.
- Additional Functionality can be added at a later date.
- Software is produced early in the software life cycle.
- Project estimates in terms of schedule, cost etc become more and more realistic as the project moves forward and loops in spiral get completed.
- It is suitable for high risk projects, where business needs may be unstable. A highly customized product can be developed using this.

3-4) Software Requirements:

- Operating System : Windows
- Graphical User Interface : Java Swing, AWT.
- Application Logic : Java SE 11.0.2.
- Database : Oracle 10c.
- IDE/Workbench : My Eclipse 2017.

3-5) Hardware Requirements:

- Processor : Pentium III – 900 MHz
- Hard Disk : 20 GB
- RAM : 128 MB

3-6) Type of Design:

Object–Oriented Design (OOD) involves implementation of the conceptual model produced during object-oriented analysis. In OOD, concepts in the analysis model, which are technology–independent, are mapped onto implementing classes, constraints are identified and interfaces are designed, resulting in a model for the solution domain, a detailed description of how the system is to be built on concrete technologies.

The implementation details generally include:

- Restructuring the class data.
- Implementation of methods, internal data structures and algorithms.
- Implementation of control.
- Implementation of associations.

Grady Booch has defined object-oriented design as “a method of design encompassing the process of object-oriented decomposition and a notation for depicting both logical and physical as well as static and dynamic models of the system under design”.

A key goal of the object-oriented approach is to decrease the "semantic gap" between the system and the real world, and to have the system be constructed using terminology that is almost the same as the stakeholders use in everyday business. Object-oriented modeling is an essential tool to facilitate this.

Object: An object is a real-world element in an object–oriented environment that may have a physical or a conceptual existence. Each object has:

- Identity that distinguishes it from other objects in the system.
- State that determines the characteristic properties of an object as well as the values of the properties that the object holds.
- Behavior that represents externally visible activities performed by an object in terms of changes in its state.

Objects can be modeled according to the needs of the application. An object may have a physical existence, like a customer, a car, etc.; or an intangible conceptual existence, like a project, a process, etc.

Class: A class represents a collection of objects having same characteristic properties that exhibit common behavior. It gives the blueprint or description of the objects that can be created from it. Creation of an object as a member of a class is called instantiation. Thus, object is an instance of a class. The constituents of a class are:

- A set of attributes for the objects that are to be instantiated from the class.
- Generally, different objects of a class have some difference in the values of the attributes. Attributes are often referred as class data.
- A set of operations that portray the behavior of the objects of the class. Operations are also referred as functions or methods.

There are two Properties of Object orientation:

1) **Abstraction:** Abstraction allows us to represent complex real world in simplest manner. It is the process of identifying the relevant qualities and behaviors an object should possess; in other words, to represent the necessary feature without representing the background details.
2) **Encapsulation:** Encapsulation is the process of binding both attributes and methods together within a class. Through encapsulation, the internal details of a class can be hidden from outside. It permits the elements of the class to be accessed from outside only through the interface provided by the class.

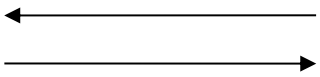
4-1) Data Flow Diagrams (DFD) Description:

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart.

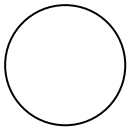
DFD's are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts.

The Basic Notation used to create a DFD's are as follows:

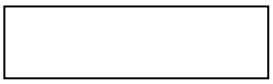
1. Data flow: Data move in a specific direction from an origin to a destination.



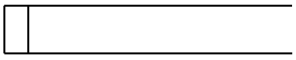
2. Process: People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.



3. Source: External sources or destination of data, which may be People, programs, organizations or other entities.



4. Data Store: Here data are stored or referenced by a process in the System.



1. Use case Description:

In software engineering, a use case diagram in the Unified Modelling language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

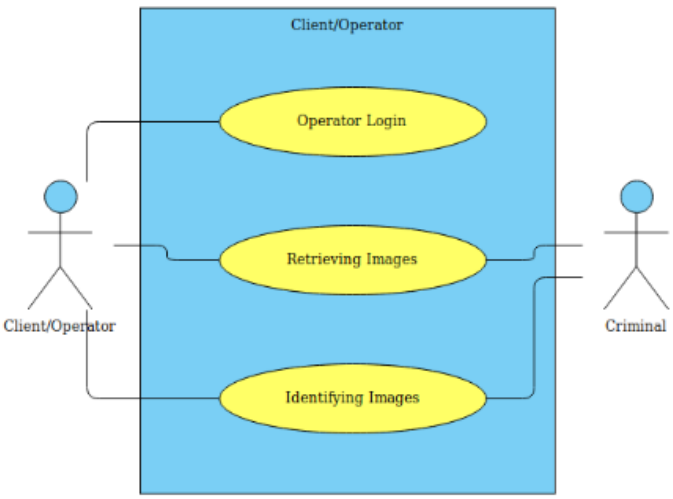
The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Use Case diagrams are formally included in two modelling languages defined by the OMG. Both the UML and SysML standards define a graphical notation for modelling use cases with diagrams. One complaint about the standards has been that they do not define a format for describing these use cases. Generally, both graphical notation and descriptions are important as they document the use case, showing the purpose for which an actor uses a system.

The use case diagram shows the position or context of the use case among other use cases. As an organizing mechanism, a set of consistent, coherent use cases promotes a useful picture of system behavior, a common understanding between the customer/owner/user and the development team.

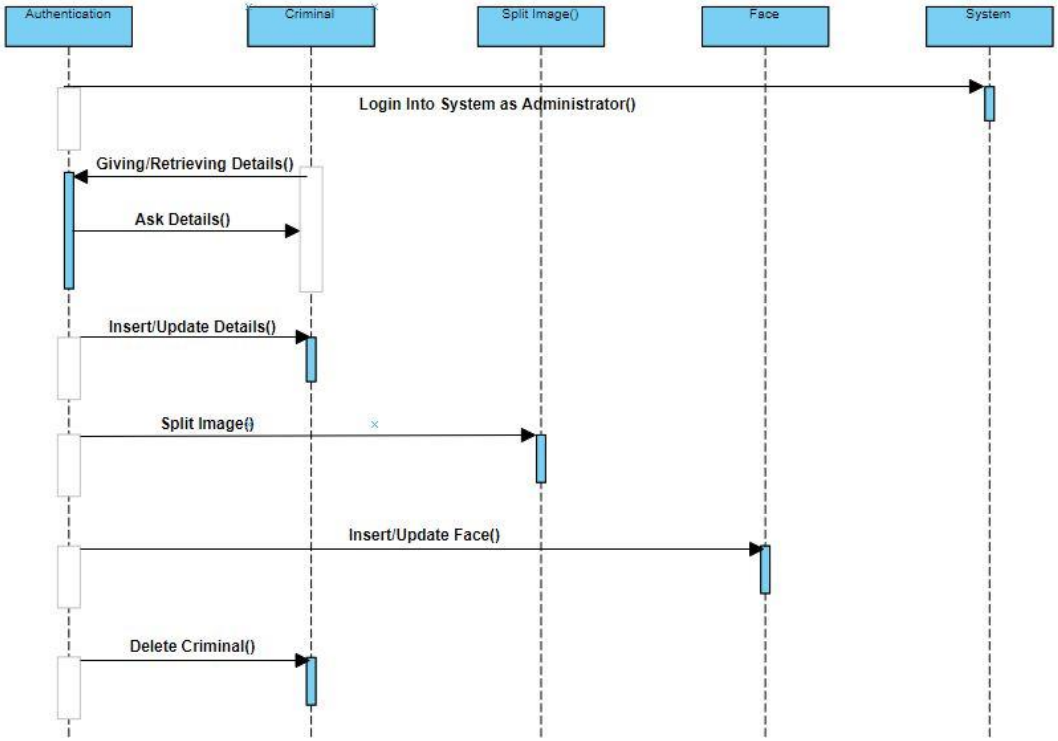
Use-Case Diagrams:

For Operator:

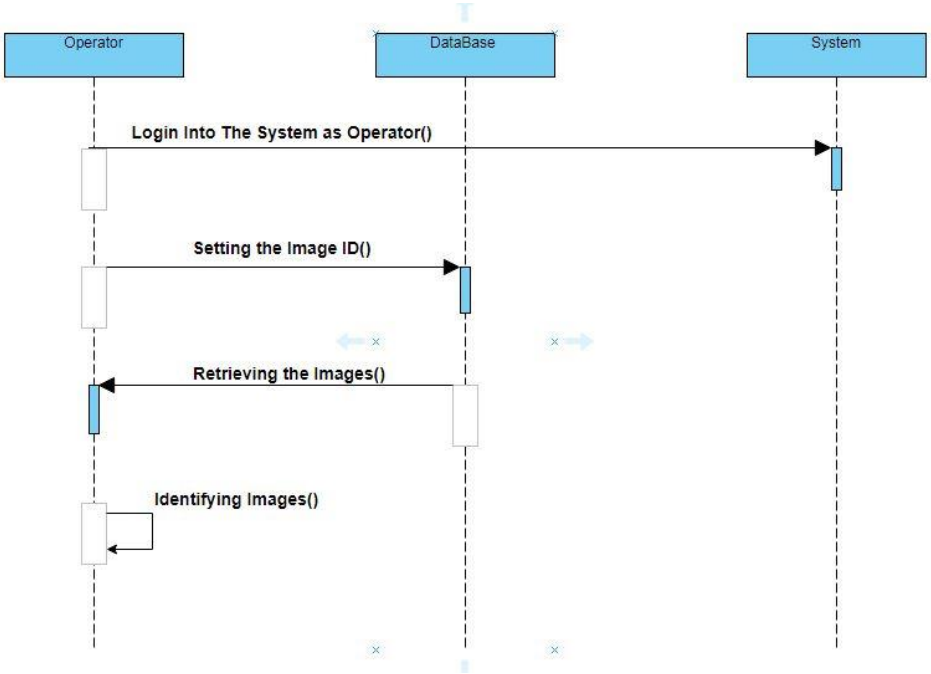


Sequence Diagram:

For Administrator:



For Operator:

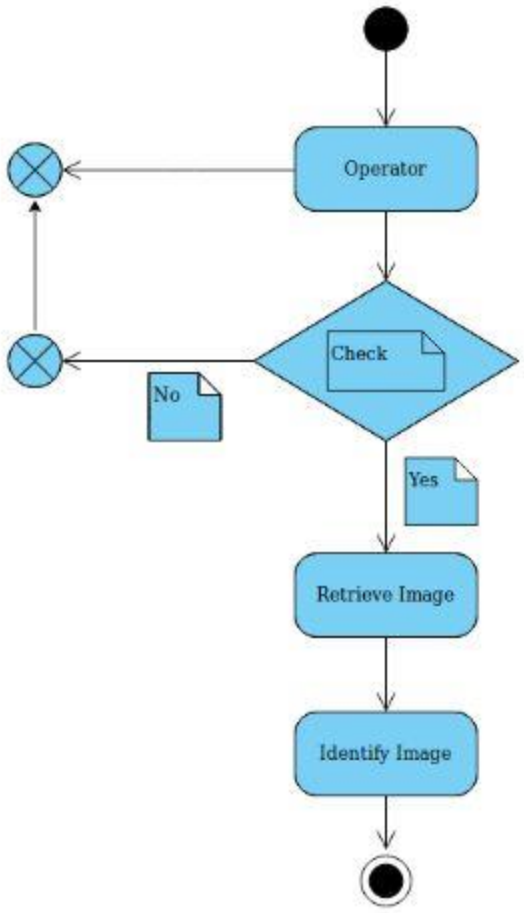


2. Activity Diagram Description:

Activity diagrams are a loosely defined diagram technique for showing workflows of stepwise activities and actions, with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. n SysML the activity diagram has been extended to indicate flows among steps that convey physical element (e.g., gasoline) or energy (e.g., torque, pressure).

Activity Diagram:

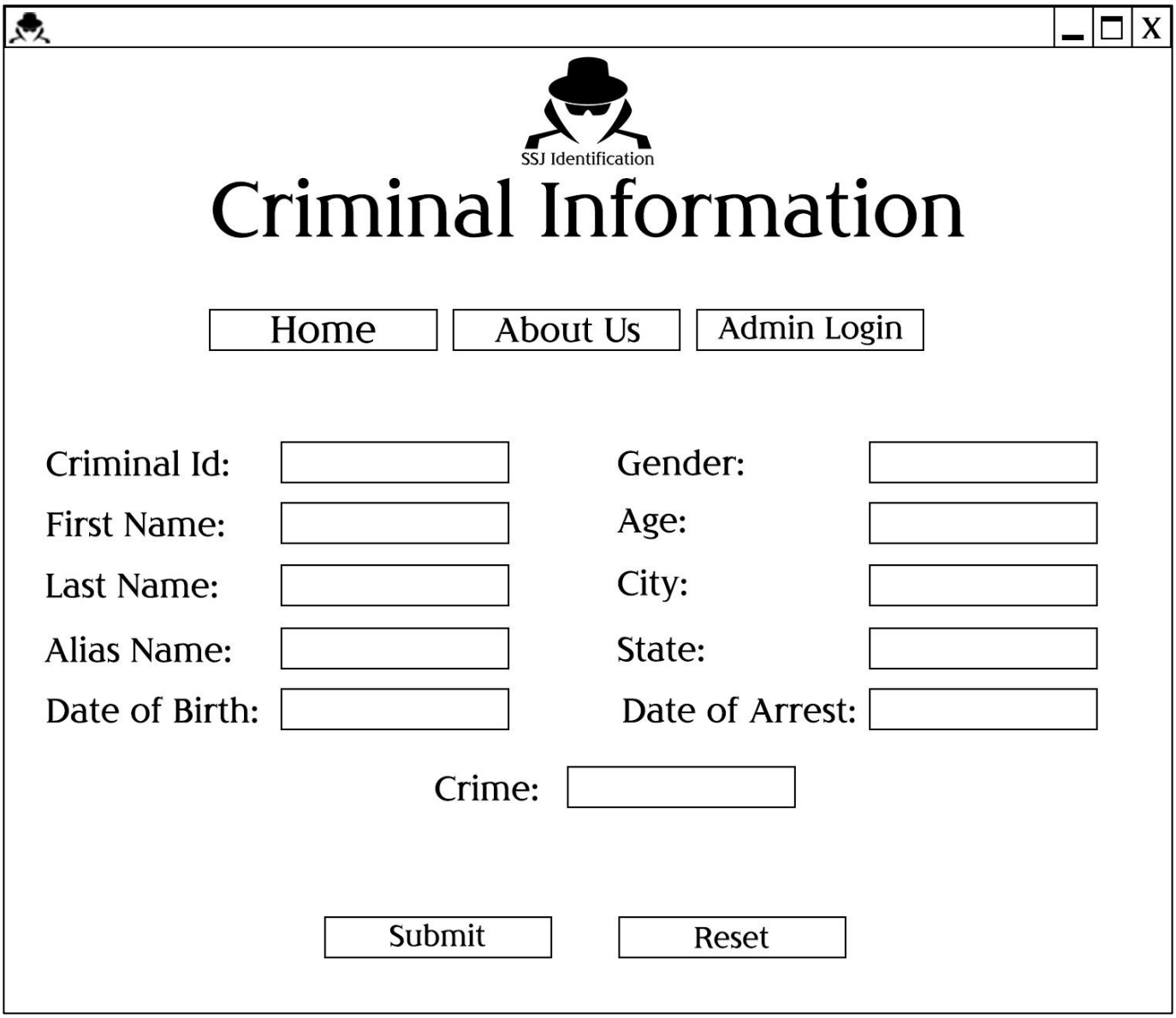
For Operator:



Login Page:


SSJ Identification
LOGIN PAGE
Home
About Us
Admin Login
User Name:
Password:
Type of User:
Admin
Submit
Reset

Criminal Information:



Insert Image:

Image Data:

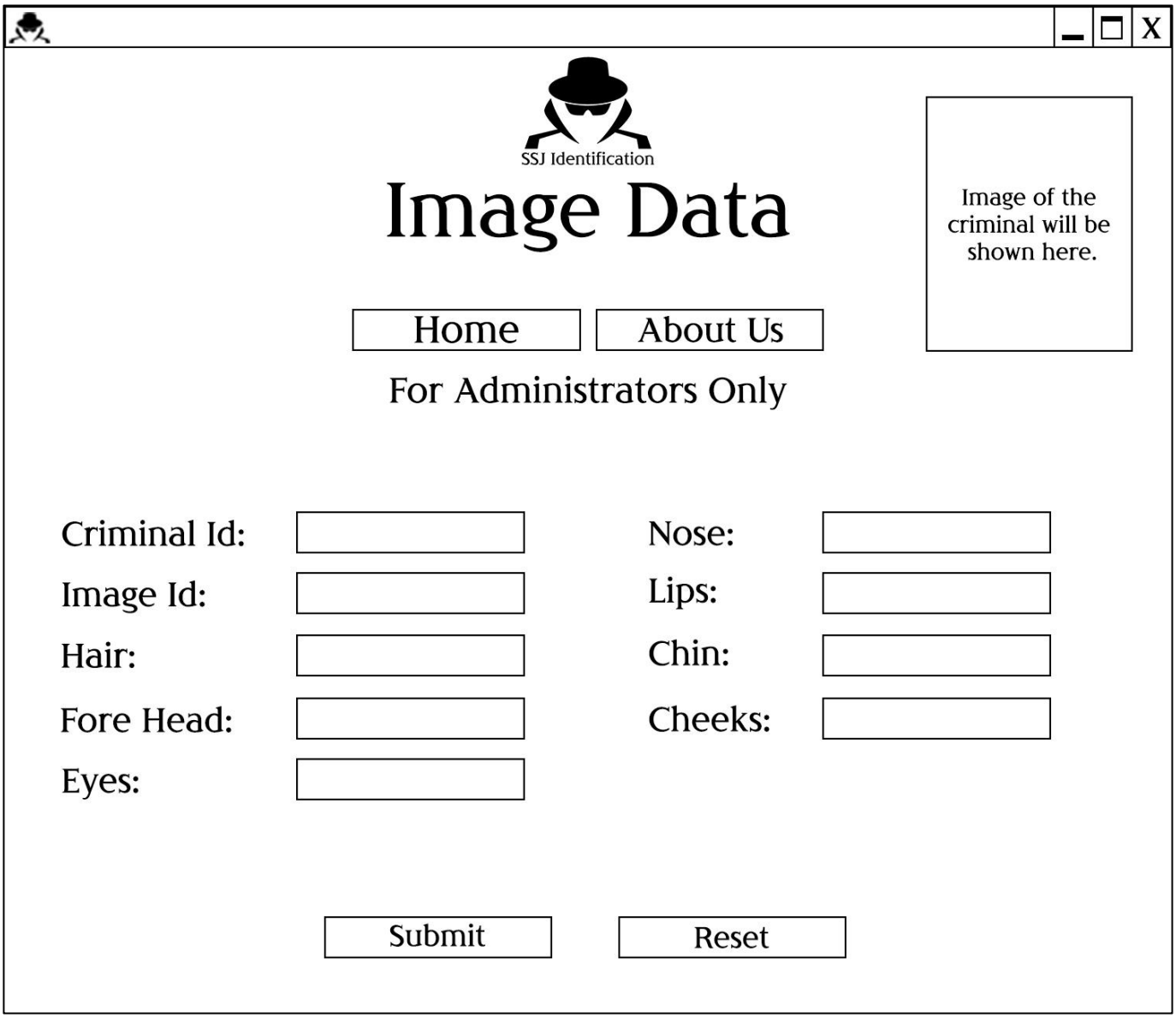

SSJ Identification
Image Data
Image of the criminal will be shown here.
Home
About Us
For Administrators Only
Criminal Id:
Image Id:
Hair:
Fore Head:
Eyes:
Nose:
Lips:
Chin:
Cheeks:
Submit
Reset

Image Deletion: