A Project Report On

MISSING CHILD

Submitted by

MUHAMMED MUSAMMIL K (PYAWBCA014

MOHAMMED NISHAD (PYAWBCA022)

SAMEEHANA V (PYAWBCA035)

FATHIMA SHAMLA (PYAWBCA001)

Guided By

Ms. VINEESHA VENUGOPALAN

Submitted in partial fulfilment of the requirements for Award

Of the degree of

BACHELOR OF COMPUTER APPLICATIONS

UNIVERSITY OF CALICUT

PRIYADARSHINI ARTS AND SCIENCE COLLEGE

MELMURI, MALAPPURAM

Affiliated to the University of Calicut

2022-2025

PRIYADARSHINI ARTS AND SCIENCE COLLEGE

MELMURI, MALAPPURAM

Affiliated to the University of Calicut

CERTIFICATE

This is to certify that project report entitled as "**MISSING CHILD**" is a bonafide record of the original work done by MUHAMMED MUSAMMIL K(PYAWBCA014), MOHAMMED NISHAD (PYAWBCA022), SAMEEHANA V (PYAWBCA035), FATHIMA SHAMLA (PYAWBCA001) toward the partial fulfillment of the requirements for the award of the degree of **BACHELOR OF COMPUTER APPLICATION** under **UNIVERSITY OF CALICUT**. We also certify that the work is original and has not been submitted to any other university or in a part of any degree or diploma.

Ms. VINEESHA VENUGOPALAN
INTERNAL GUIDE

Mr. MIDHUN M
HEAD OF THE DEPARTMENT

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

I express gratitude to the Almighty for providing me with the courage and perseverance to complete this project. Special thanks to our principal, Prof. M. Shanthakumar, for the Infrastructure and lab support. Heartfelt appreciation to Mr. Midhun M, Head of Computer Science, for continuous guidance. I am thankful to project coordinator Vineesha Vennugopalan, for sincere support. Deep thanks to my parents for moral support. Lastly, thanks to friends for their help in successfully completing this work.

MUHAMMED MUSAMMIL K(PYAWBCA014)

MOHAMMED NISHAD (PYAWBCA022)

SAMEEHANA V (PYAWBCA035)

FATHIMA SHAMLA (PYAWBCA001)

DECLARATION

I hereby declare that the project entitled "Missing Child" submitted to University of Calicut in a partial fulfilment of the requirements for the award of the Degree of Bachelor of Computer Application, UNIVERSITY OF CALICUT is a record of original project done by me during the project of study at PRIYADARSHINI ARTS AND SCIENCE COLLEGE, MELMURI under the supervision and guidance of Ms. Vineesha Vennugopalan , faculty of Priyadarshini Arts and Science College.

Place: Melmuri, Malappuram

Date:

MUHAMMED MUSAMMIL K (PYAWBACA014)

MOHAMMED NISHAD (PYAWBCA022)

SAMEEHANA V (PYAWBCA035)

FATHIMA SHAMLA (PYAWBCA001)

CONTENTS

Introduction

1. Introduction

Every year, a large number of children go missing in India, often due to reasons like abduction, trafficking, or running away from home. On average, around 174 children are reported missing every day, and sadly, many of these cases remain unresolved. The emotional impact on families is devastating, and the longer a child remains missing, the harder it becomes to find them.

Currently, tracking and identifying missing children is done manually through posters, social media, and community reports. While these methods help, they are slow and inefficient. Identifying a missing child from a large group of children is especially challenging.

To address this problem, this project proposes a new approach: a Missing Child Identification System that uses advanced technology to recognize missing children by their faces. This system allows people to upload photos of children they believe may be missing, which are then compared to a database of missing children's photos. Using deep learning and machine learning, the system analyzes these photos to accurately match them with the right child.

The system uses Convolutional Neural Networks (CNNs) to extract key facial features and a Support Vector Machine (SVM) to classify and identify the child. This method is highly effective in handling challenges such as different lighting conditions, noise in the images, and even age changes over time.

With this technology, the system can identify missing children more accurately and quickly than traditional methods. It has the potential to bring hope to families by speeding up the search process and improving the chances of finding missing children.

In this project, we aim to develop an easy-to-use platform that combines face recognition technology with machine learning to help identify missing children faster and more reliably, ultimately making a positive impact on child safety and well-being.

1.1 Abstract

In India, a large number of children go missing every year due to reasons like abduction, trafficking, or running away. Unfortunately, many of these children remain untraced, causing significant distress for their families. Current methods, like posters and social media, are slow and often not very effective. To address this problem, we propose a **Missing Child Identification System** that uses **Deep Learning** and **Multiclass Support Vector Machine (SVM)** for face recognition.

The system allows people to upload photos of suspected missing children to a central platform. The uploaded images are then automatically compared with a database of missing children's photos. The system uses a **Convolutional Neural Network (CNN)**, specifically the **VGG-Face** model, to extract important features from the images. These features are then analyzed and classified using an **SVM** to identify the missing child.

One of the biggest challenges in identifying missing children is dealing with different conditions like poor lighting, noisy images, age changes, or different poses. Our system handles these challenges effectively by using deep learning techniques to ensure accurate identification, even when the photos are not perfect.

The system was tested on **43 real missing child cases** and achieved an impressive **99.41% accuracy**. This shows that the system can accurately match images of missing children, even in real-world situations.

This system is faster, more reliable, and scalable compared to existing methods. By combining deep learning and machine learning, it offers a more effective solution to the problem of missing children.

The **Missing Child Identification System** can significantly improve how missing children are found, helping authorities and families locate missing children more quickly and accurately. This system provides a more efficient way to solve this critical issue, ultimately increasing the chances of reuniting families with their missing children.

System Study and Analysis

2.1 Preliminary Investigation

The problem of missing children in India is growing, with hundreds of children reported missing every day. Traditional methods like posters and social media campaigns are slow and often ineffective, leading to a significant number of cases remaining unsolved. These methods struggle to process large volumes of cases efficiently, and detecting missing children from crowds or photographs manually is challenging and time-consuming.

To address these challenges, this project proposes a **Missing Child Identification System** that utilizes **Deep Learning** and **Machine Learning** techniques. Specifically, it uses **Convolutional Neural Networks (CNNs)** for face recognition, which allows the public to upload images of suspected missing children. The system then compares these images to a database of missing children's photographs, accurately identifying matches by using a pre-trained **VGG-Face model** combined with a **Support Vector Machine (SVM)** classifier.

The system was tested on **43 missing child cases**, achieving an impressive **99.41% accuracy**. This result shows the system's potential to improve the process of identifying missing children by providing a faster, more accurate, and scalable solution compared to traditional methods. With this deep learning approach, the system offers a reliable way to address the issue of missing children in India more effectively.

2.2. Existing System

In India, a large number of children go missing every year due to various reasons such as abduction, trafficking, running away, or being lost. According to reports, approximately 174 children go missing daily, and half of them remain untraced. The current methods used to track missing children rely heavily on manual interventions, such as posting flyers, distributing information through social media platforms, and word-of-mouth. While these methods are widespread, they often lack efficiency and have limited success in reaching the right audience quickly.

The existing systems are largely centralized around the manual handling of missing child reports, where parents or guardians file complaints with local authorities. These systems use databases of photographs and descriptions, which are then manually compared or cross-checked by authorities. While some government initiatives have been implemented, such as the **TrackChild** portal, they still rely heavily on public awareness and active reporting, making the process slow and difficult to scale. Additionally, these systems struggle with issues such as low-quality images, lighting conditions, age progression, and pose variations, which can make accurate identification challenging.

Furthermore, existing systems are not equipped to automatically analyze and match images in real-time. They often depend on human effort to search through extensive databases of missing children. This process can be prone to errors, delays, and misidentifications, leaving many children untraced for long periods. Thus, there is a significant need for an automated system that can quickly and accurately match uploaded images with a database of missing children, reducing the time it takes to locate them and improving the chances of recovery.

Limitations of Existing Systems:

1. **Manual and Slow Identification**: Existing systems rely heavily on manual processes to match missing child reports and photographs, which leads to delays and inefficiencies in locating missing children.

2. **Inconsistent Image Quality**: The existing systems struggle to handle variations in image quality, lighting, and angles, which can affect the accuracy of child identification from photos.

3. **Lack of Automation and Scalability**: Current systems lack automation, making it difficult to process large databases of missing children quickly. This limits the ability to scale and handle a growing number of cases effectively.

2.3. Proposed System

Proposed System

The proposed **Missing Child Identification System** leverages advanced technologies, such as **Deep Learning** and **Machine Learning**, to automatically identify missing children based on face recognition. This system aims to improve the accuracy, speed, and scalability of identifying missing children compared to existing methods.

1. **Automated Image Matching**: The system uses **Convolutional Neural Networks (CNNs)**, specifically the pre-trained **VGG-Face model**, for face recognition. The public can upload images of children they believe might be missing, which will then be automatically compared to a database of missing children's photographs. This automated process eliminates the need for manual intervention, speeding up the identification process.
2. **Improved Accuracy**: By utilizing **CNNs** for facial feature extraction and a **Support Vector Machine (SVM)** for classification, the system ensures high recognition accuracy, even in challenging conditions like poor lighting, different poses, and varying image quality. The system is designed to handle image variations such as age progression and pose differences, increasing its effectiveness in real-world scenarios.
3. **Scalable and Accessible Solution**: The system is built to handle large volumes of data efficiently. It allows both authorities and the general public to upload and access information through a centralized portal. With its automated image matching and processing capabilities, the system can handle numerous missing child cases simultaneously, making it highly scalable. This ensures a faster response time, increasing the chances of successfully locating missing children.

In summary, the proposed system combines deep learning and machine learning to provide an efficient, scalable, and highly accurate solution to the problem of missing children, overcoming the limitations of current manual and slow identification processes.

Advantages of Proposed System

1. **High Accuracy and Reliability**: The system utilizes advanced **Deep Learning** techniques, such as **CNNs** and **SVM**, to ensure accurate identification of missing children, even under challenging conditions like poor lighting or different facial poses.

2. **Faster and Automated Process**: By automating the image matching and identification process, the system significantly speeds up the response time, eliminating delays caused by manual intervention and ensuring quicker identification of missing children.

3. **Scalability and Efficiency**: The system can efficiently process large volumes of data and handle numerous missing child cases simultaneously, making it scalable and capable of operating effectively even as the number of cases grows.

2.2.Feasibility Study

A feasibility study is a preliminary study undertaken to determine and document a project's viability. The results of this study are used to make a decision whether to proceed with the project. If it indeed leads to a project being approved, it will - before the real work of the proposed project starts - be used to ascertain the likelihood of the project's success. It is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative. It, for example, can decide whether an order processing be carried out by a new system more efficiently than the previous one. The feasibility study proposes one or more conceptual solutions to the problem set for the project. The conceptual solution gives an idea of what the new system will look like. They define what will be done on the computer and what will remain manual. It also indicates what input will be needed by the system and what outputs will be produced. These solutions should be proven feasible and a preferred solution is accepted.

The feasibility study environment enables all alternatives to be discussed and evaluated. This phase starts with an identification of the main characteristics of the required system. During this stage it is important to collect information as much as possible about the software package that might meet the specification from as many sources as possible.

Normally, the central endeavor of a feasibility study is a cost benefit analysis of various alternatives. It can be defined as a systematic comparison between the cost of carrying out a service or activity and the value of that service or activity. The main benefits are qualitative than

quantitative.

A feasibility study could be used to test a new working system, which could be used because:

- The current system may no longer suit its purpose,
- Technological advancement may have rendered the current system obsolete,
- The business is expanding, allowing it to cope with extra work load,
- Customers are complaining about the speed and quality of work the business provides.
- Competitors are now winning a big enough market share due to an effective integration of a computerized system.

When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. Facts considered in the feasibility analysis were

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

Technical Feasibility

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on an outline design of system requirements in terms of Input, Output, Fields, Programs, and Procedures. This can be qualified in terms of volumes of data, trends, frequency of updating etc. in order to give an introduction to the technical system.

The system requires normal configuration computer system that are commonly available. The software requirements are Python and Android, Windows 8 or higher versions of OS. Thus proposed system is technically feasible.

Operational Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and there for it will accept a broad audience.

The proposed system offers:

- Greater user friendliness
- Better output which can be easily interpreted.
- Higher speed.
- Meets the requirements of the organizations.

Economic Feasibility

This involves questions such as whether the firm can afford to build the system, whether its benefits should substantially exceed its costs, and whether the project has higher priority and profits than other projects that might use the same resources. This also includes whether the project is in the condition to fulfill all the eligibility criteria and the responsibility of both sides in case there are two parties involved in performing any project.

This study presents tangible and intangible benefits from the project by comparing the developments and operational costs. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve the quality of service.

Thus feasibility study should center along the following points:

- Improvement resulting over the existing method in terms of accuracy, timeliness.
- Cost comparison.
- Estimate on the life expectancy of the hardware.
- Overall objective.

Project Planning and Scheduling

3.1Project Planning

For the successful completion of the every project there must be detailed scheduling. The software development has different participating steps. First of all, I done the requirement analysis phase. For this I visit different sites that offer resume writing helps, visits different business websites, and I discuss with my friends and project guide.

After collecting the requirements a detailed study of preliminary investigation is done.
It includes six major questions.

1. What is being done?
2. How it is being done?
3. Does a problem exist?
4. If a problem exists how severe it is?
5. How frequently does it occur?
6. What is main reason for that problem?

After the analysis phase the requirement is divided into modules. The design document is divided into modules. The document is created, which includes data flow diagrams, ER diagrams etc.

As next step the actual development of the system takes place. The design representations are translated into codes. Documentation of codes are done by providing an explanations of how procedures are used. Documentation is essential to test the program and carry on maintenance once the application has been installed.

As next step testing is done. After a system has been developed it is very important to check if it fulfills the customer requirements.

Implementation of the system means putting up system on user's side. Like any system there is an aging process. Therefore the system requires periodic maintenance for software or hardware.

3.2Project Scheduling

	Week 1	Week 2	Week 3	Week 4	Week 5
System Study					
System Design					
Coding					
Testing					

System Requirement Specification

4. System Requirements

Hardware Specification

The selection of hardware is very important in the existence and proper working of any of the software. When selecting hardware, the size and capacity requirements are also important. The hardware must suit all application developments**.**

- Processor : i3 or above
- RAM : 4 GB or above
- System Bus : 32 Bit or 64 Bit
- Monitor : LED display
- Key Board : Windows compatible
- Mouse : Windows compatible
- Mobile : Android Supported Mobile Phone
- HDD : 500 Gb or Above

Software Specification

One of the most difficult tasks is selecting software, once the system requirement is find out then we have to determine whether a particular software package fits for those system requirements.

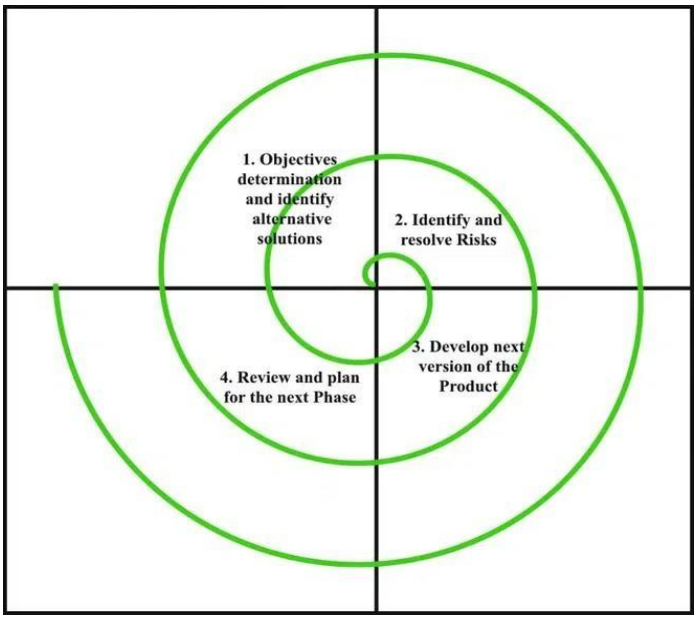
This section summarizes the application requirement.

- Database : MySQL
- Operating System : Windows 8 or above
- Front End : HTML, CSS, BOOTSTRAP, JAVASCRIPT
- Back End : Python-Django or Flutter/Java
- Technologies used : Android
- IDE : PyCharm community , Android Studio , PyCharm community , vs code
- Web browser : Chrome, Explorer, Edge…etc.

System Development And Life Cycle

5.1 System Development and Life Cycle

Spiral Model



The Spiral Model is a software development lifecycle (SDLC) approach that combines the **structured elements of the waterfall model** with the **iterative and incremental nature** of agile development. This unique blend prioritizes **risk management** throughout the entire development process.

Key characteristics:

- **Iterative:** Development progresses through **repeating cycles**, each building upon the previous one.
- **Risk-driven:** Each cycle emphasizes **identifying and mitigating potential risks** associated with the planned features.
- **Flexible and adaptable:** The project scope and approach can be **adjusted based on risk assessments** and ongoing evaluation.
- **Suited for complex projects:** Ideal for managing large, complex projects with **uncertain requirements**.

Benefits:

- **Early risk identification:** Proactive risk assessment helps avoid costly last-minute changes.
- **Adaptability:** Allows for adjustments based on evolving needs and feedback.
- **Continuous improvement:** Each iteration refines the product, leading to a more robust outcome.

Drawbacks:

- **Complexity:** Managing multiple iterations and risk assessments requires careful planning.
- **Uncertain cost and schedule:** Iterative nature makes it challenging to establish fixed costs and timelines upfront.
- **Need for experienced management:** Requires skilled professionals to manage risk and adapt the project effectively.

In essence, the Spiral Model offers a flexible and risk-mitigating approach to software development, particularly valuable for complex projects with evolving requirements and high-risk factors.

System Design

6.1 System Design

The most creative and challenging phase of the system life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specification that will be applied in implementing the candidate system. It also include the construction of the program and the program testing the key question involved here is “how the problem should be solved”.

System design is a solution for the question of how to the approach to the creation of a new system. This important phase is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended feasible study. Emphasis is on translating the performance requirements into design specifications. Design goes through logical and physical system; prepare input and output specification; make credit, security and control specification; details the implementation plan; prepare a logical design walk. Physical design maps out the physical system, plans the system implements, devices a test and implementation plan and specifies any new hardware and software.

The first most is to determine how the output is to be produced and in what format. Samples of output and input are presented. Second, input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing, including a list of programs needed to meet the system’s objectives and complete documentation. Finally, details related to justification of the system and estimate of the impact of the candidate system on the user and organization are documentation and evaluated by management as a step towards implementation. The final report prior to the implementation phases includes procedural flowcharts, record layouts and workable plan for implementing the candidate system.

6.2 Module Description

Admin:

- Login
- Add and manage child help line
- View missing case and detailes
- View emergency complaints
- View complaint and send reply
- View feedback
- Camera management

Child help line:

- Login
- View missing case details
- Verify and update missing case statuses
- Add and manage notifications
- View notification from cameras
- Respond to emergency complaints

Users:

- Registration
- Login
- Send feedback
- Send complaint and view reply
- Manage child missing case
- View notification
- View missing case update
- Send emergency complaint to child help line and view
- Chat

6.3Data Design (DFD)

A data flow diagram (DFD) or a bubble chart is a graphical tool for structured analysis. DFD models a system by using external entities from which data flow to a process, which transforms the data and creates output data flows which go other process or external entities or files. Data in files may also flow to processes as inputs.

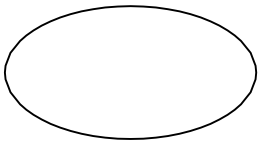
DFDs can be hierarchically organized, which help in partitioning and analyzing large systems. As a first step, one dataflow diagram can depict an entire system which gives the system overview. It is called context diagram of level0 DFD. The context diagram can be further expanded.

The successive expansion of a DFD from the context diagram to those giving more details is known as leveling of DFD. Thus a top down approach is used, starting with an overview and then working out the details. The main merit of the DFD is that it can provide an overview of what data a system would process, what transformation of data are done, what files are used, and where the results flow.

DFD Design Notation

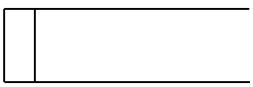
In DFD, there are four main symbols:

Source or Destination of Data



Process

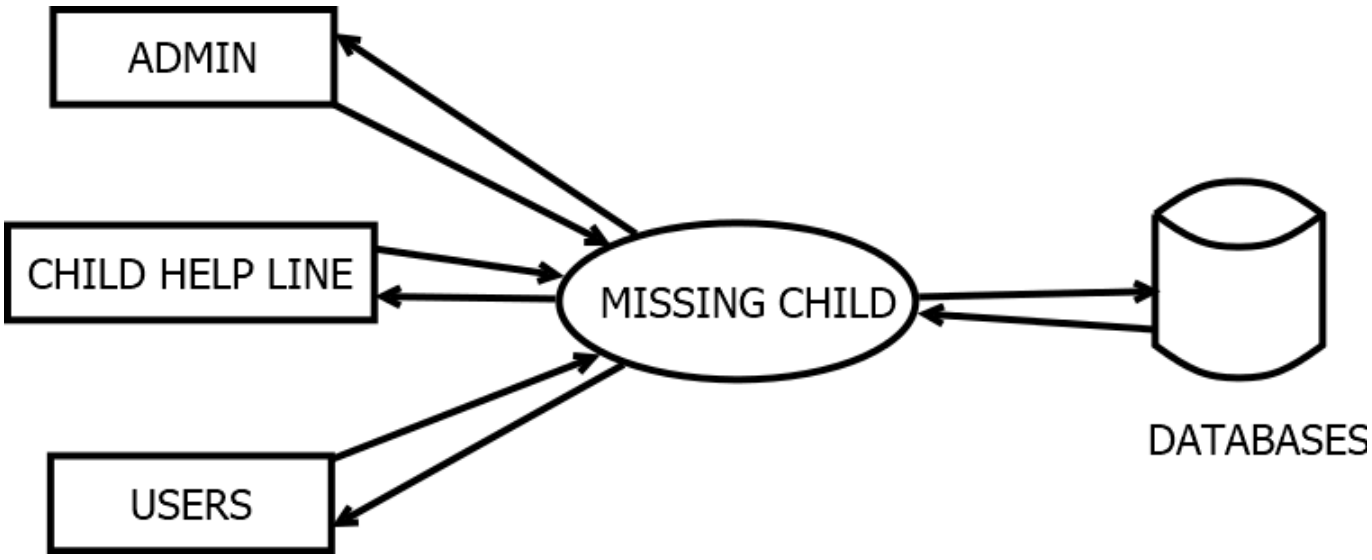
Flow of Data



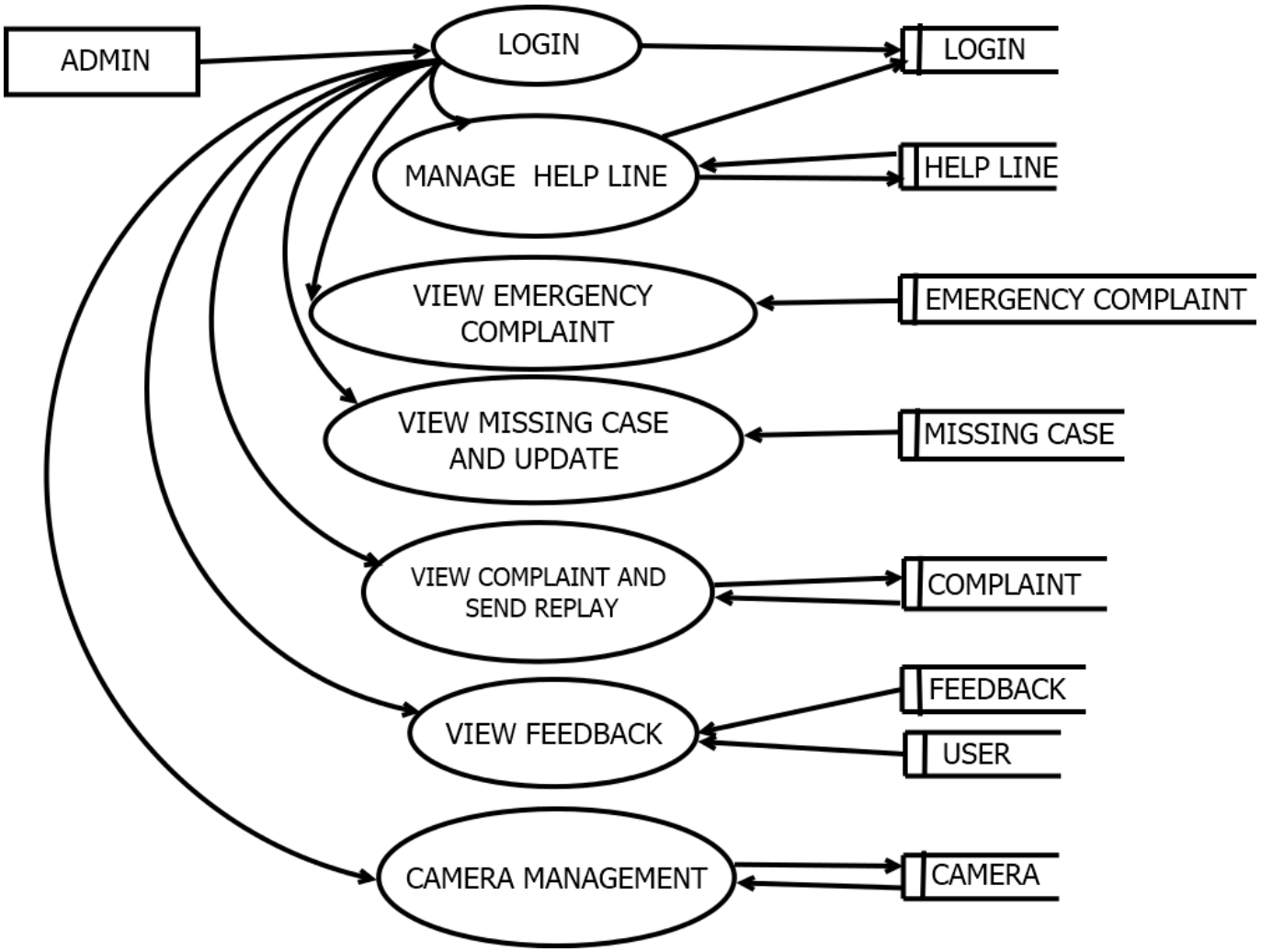
Temporary Repository of Data

LEVEL 0 DFD

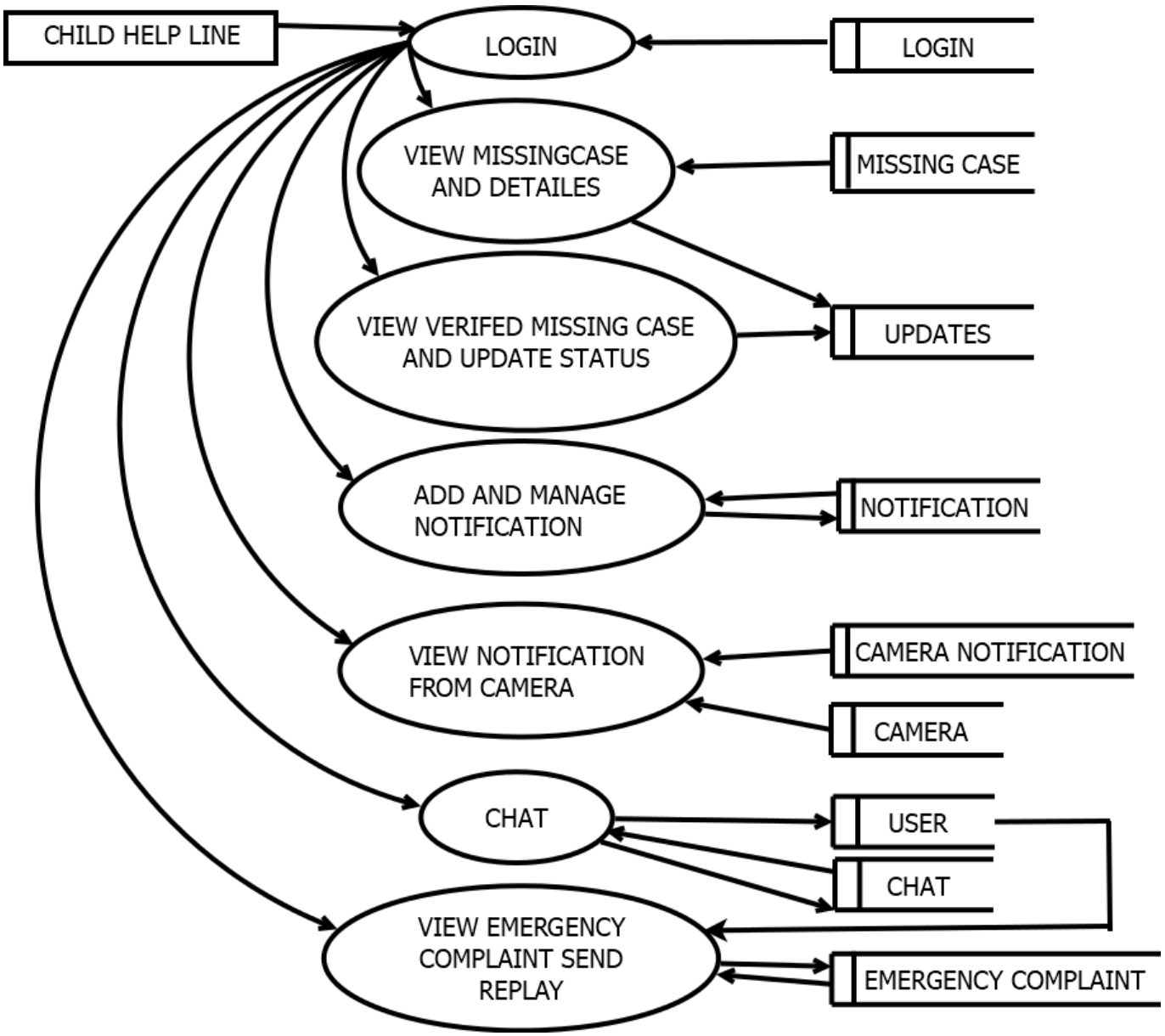
LEVEL 1 DFD



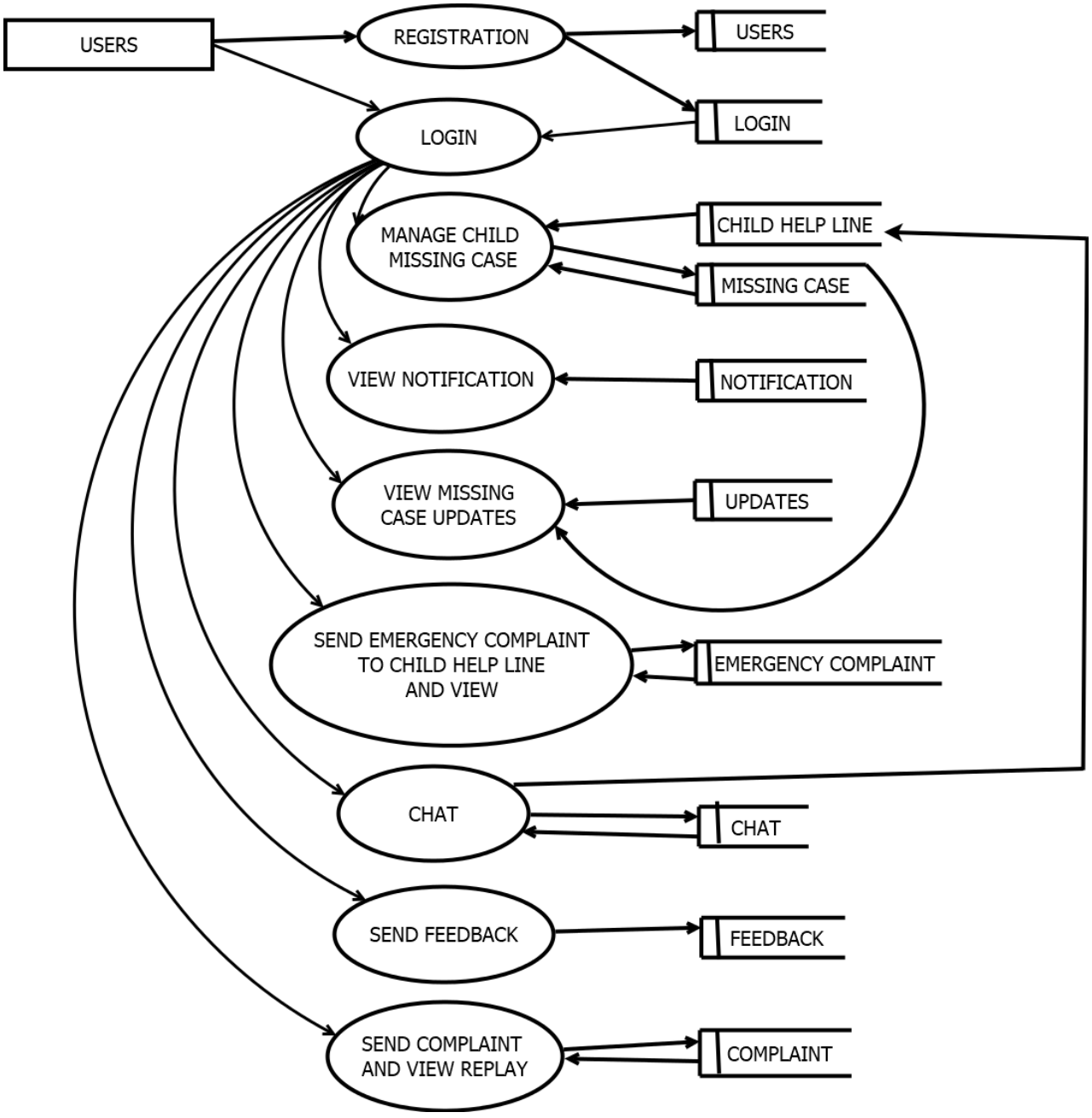
LEVEL 1.1 DFD


ADMIN
LOGIN
LOGIN
MANAGE HELP LINE
HELP LINE
VIEW EMERGENCY COMPLAINT
EMERGENCY COMPLAINT
VIEW MISSING CASE AND UPDATE
MISSING CASE
VIEW COMPLAINT AND SEND REPLAY
COMPLAINT
VIEW FEEDBACK
FEEDBACK
USER
CAMERA MANAGEMENT
CAMERA

LEVEL 1.2 DFD


CHILD HELP LINE
LOGIN
LOGIN
VIEW MISSINGCASE AND DETAILES
MISSING CASE
VIEW VERIFED MISSING CASE AND UPDATE STATUS
UPDATES
ADD AND MANAGE NOTIFICATION
NOTIFICATION
VIEW NOTIFICATION FROM CAMERA
CAMERA NOTIFICATION
CAMERA
CHAT
USER
CHAT
VIEW EMERGENCY COMPLAINT SEND REPLAY
EMERGENCY COMPLAINT

LEVEL 1.3 DFD


USERS
REGISTRATION
USERS
LOGIN
LOGIN
MANAGE CHILD MISSING CASE
CHILD HELP LINE
MISSING CASE
VIEW NOTIFICATION
NOTIFICATION
VIEW MISSING CASE UPDATES
UPDATES
SEND EMERGENCY COMPLAINT TO CHILD HELP LINE AND VIEW
EMERGENCY COMPLAINT
CHAT
CHAT
SEND FEEDBACK
FEEDBACK
SEND COMPLAINT AND VIEW REPLAY
COMPLAINT

6.4Table Design

Login Table

Column Name	Data Type	Constraints
id	int	primary key
username	varchar	not null
password	varchar	not null
Type	varchar	not null

Child Help Line Table

Column Name	Data Type	Constraints
id	int	Primary key
login_id	int	Foreign key
name	varchar	not null
latitude	varchar	not null
place	varchar	not null
post	varchar	not null
pin	int	not null
email	varchar	not null
phone	int	not null
longitude	varchar	not null

Emergency-complaint Table

Column Name	Data Type	Constraints
id	int	Primary key
Child help line_id	int	Foreign key
User_id	int	Foreign key
Message	varchar	not null
Date	date	not null
Status	varchar	not null

Missing case Table

Column Name	Data Type	Constraints
id	int	Primary key
User_id	int	Foreign key
Child name	varchar	not null
Child image	file	not null
Missing_date	date	not null
Description	varchar	not null
age	int	not null
Upload_date	date	not null

Complaint Table

Column Name	Data Type	Constraints
id	int	Primary key
User_id	int	Foreign key
Description	varchar	Not null
reply	varchar	Not null
Date	date	Not null
Age	int	Not null

Feedback Table

Column Name	Data Type	Constraints
id	int	Primary key
User_id	int	Foreign key
Description	Varchar	not null
Date	date	Not null

User Table

Column Name	Data Type	Constraints
Id	int	Primary key
Login_id	int	Foreign key
Name	varchar	not null
email	varchar	not null
phone	int	not null
place	varchar	not null
pin	int	not null
post	varchar	not null

Updates Table

Column Name	Data Type	Constraints
id	int	Primary key
Missing_id	int	Foreign key
Description	varchar	not null
date	date	not null

Notification Table

Column Name	Data Type	Constraints
id	int	Primary key
Heading	varchar	not null
date	Date	not null
Description	Varchar	not null

Camera notification Table

Column Name	Data Type	Constraints
id	int	Primary key
Missing_id	int	Foreign key
Latitude	varchar	not null
longitude	varchar	not null
date	date	not null
Child image	file	not null
Camera_number	int	Foreign key

Camera Table

Column Name	Data Type	Constraints
id	int	Primary key
Camera_id	int	Foreign key
latitude	varchar	not null
longitude	varchar	not null

Chat Table

Column Name	Data Type	Constraints
id	int	Primary key
From_id	int	Foreign key
To_id	int	not null
date	date	Not null
emergency	varchar	not null

6.5ER Diagram

An ER diagram can express the overall structure of the database graphically. ER diagrams are simple and clear. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

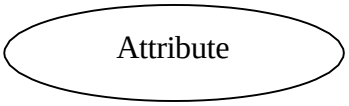
Entity Relationship Diagram Notations

Entity:

An entity is an object or concept about which you want to store information.

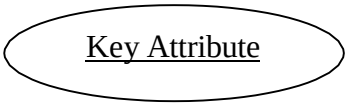
Attribute:

Each entity has attributes, or particular properties that describe the entity.



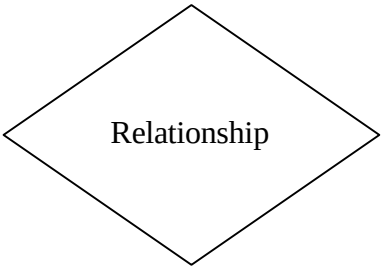
Key attribute:

A key attribute is the unique, distinguishing characteristic of the entity.

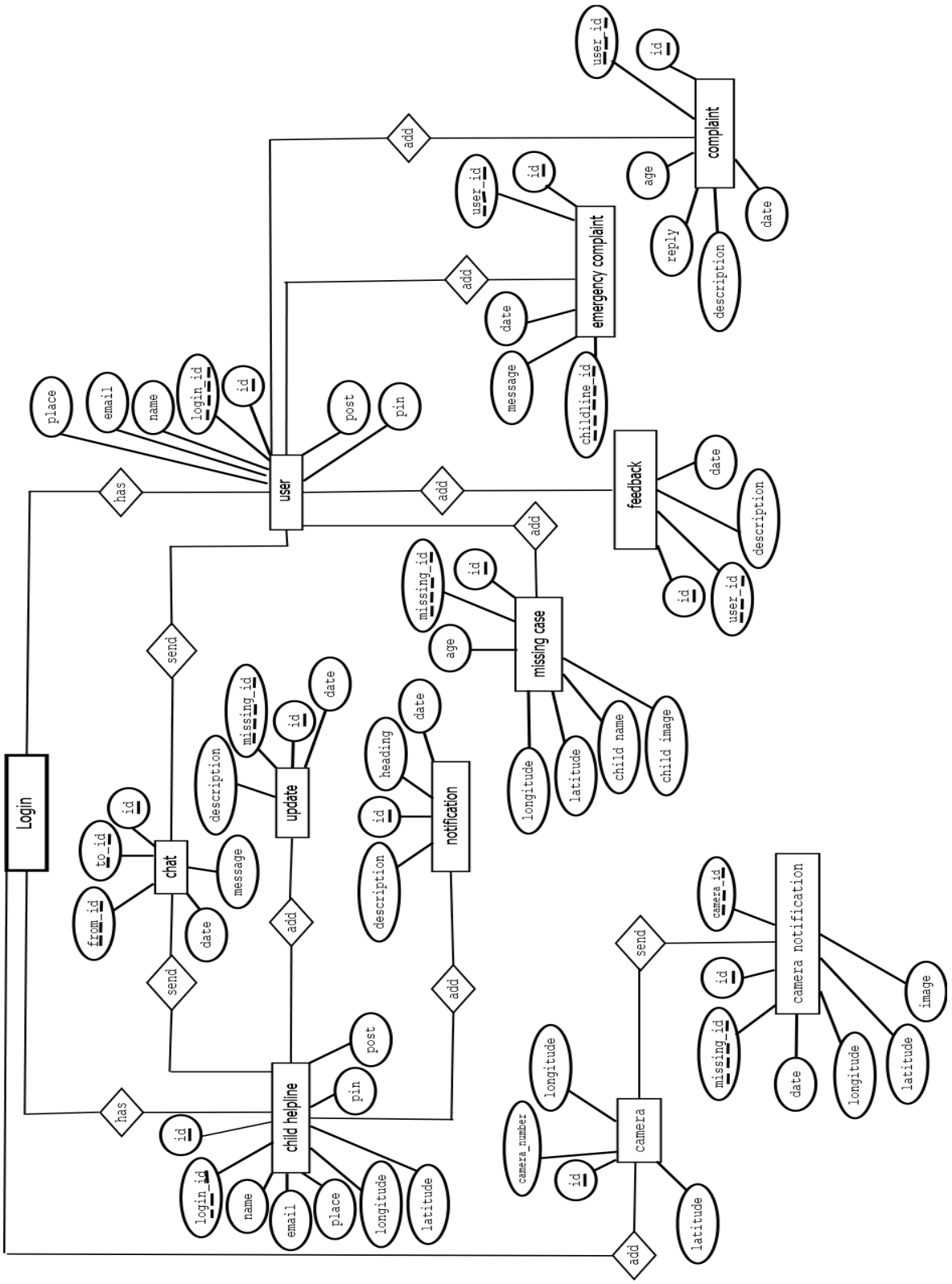


Relationships:

Relationships illustrate how two entities share information in the database structure.



ER Diagram


Login
has
user
place
email
name
login_id
id
post
pin
add
complaint
user_id
id
age
reply
description
date
emergency complaint
user_id
id
date
message
childline_id
feedback
date
description
id
user_id
missing case
missing_id
id
age
longitude
latitude
child name
child image
send
chat
id
to_id
from_id
message
date
update
missing_id
id
date
description
notification
date
heading
id
description
child helpline
post
pin
id
login_id
name
email
place
longitude
latitude
camera
camera_number
id
longitude
latitude
camera notification
camera_id
id
missing_id
date
longitude
latitude
image

6.6Normalization

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repeatation of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updation, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation. Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.

Insertion anomaly: Inability to add data to the database due to absence of other data

Deletion anomaly: Unintended loss of data due to deletion of other data.

Update anomaly: Data inconsistency resulting from data redundancy and partial update

Normal Forms: These are the rules for structuring relations that eliminate anomalies.

FIRST NORMAL FORM:

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

SECOND NORMAL FORM:

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

1 Primary key is a not a composite primary key
2 No non key attributes are present
3 Every non key attribute is fully functionally dependent on full set of primary key.

THIRD NORMAL FORM:

A relation is said to be in third normal form if their exits no transitive dependencies.

Transitive Dependency: If two non-key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

Coding and Implementation

7.1 Coding Environment

Front End

An Integrated Development Environment (IDE) (also known as Integrated Design Environment or Integrated Debugging Environment) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of:

- A source code editor
- A compiler and or an interpreter
- Build automation tools
- A debugger

Pycharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as Data Science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license.

Features

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask [professional edition only]
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development [professional edition only]
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge
- Support for scientific tools like matplotlib, numpy and scipy [professional edition only]

It competes mainly with a number of other Python-oriented IDEs, including Eclipse's PyDev, and the more broadly focused Komodo IDE.

Android Studio

Android Studio is the official Integrated Development Environment (IDE) specifically designed for building Android apps. Developed by Google in collaboration with JetBrains, it provides a comprehensive and user-friendly platform for developers of all skill levels.

As an IDE, Android Studio offers a powerful code editor based on IntelliJ IDEA. This editor provides features like code completion, syntax highlighting, and refactoring tools to streamline the development process. Additionally, Android Studio boasts built-in emulators and device testing capabilities. This allows developers to test their apps on a variety of virtual devices or directly on their physical devices, ensuring compatibility across different platforms.

Furthermore, Android Studio utilizes a flexible build system based on Gradle. This system allows for easy customization and management of the app's build process. This is crucial for developers who need to manage dependencies, configurations, and different build variants efficiently.

Finally, Android Studio comes equipped with performance profilers. These tools help developers analyze their app's memory and CPU usage. This information is essential for identifying and addressing performance bottlenecks, ensuring a smooth and efficient user experience.

In conclusion, Android Studio offers a comprehensive suite of features that cater to the needs of developers at all stages. With its powerful code editor, built-in testing tools, flexible build system, and performance profilers, Android Studio provides the ideal platform to bring your Android app ideas to life.

Back End

Database servers

A database server is used to store data in a database. Users can access the data and manipulate it. There are many types of databases. The most popular among them is the Relational Database Management System (RDBMS).

RDBMS

RDBMS is a type of database management system that stores data in the form of related tables. Relational database are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways. An important feature of relational systems is that a single database can be spread across several tables. This differs from flat-file database, in which each database is self- contained in a single table.

MySQL

MySQL is an open source relational database and it includes advanced data types. MySQL operates using client/server architecture in which the server runs on the machine containing the database and client connect to the server over the network. MySQL run on all platforms supported by MySQL and provides the most direct means of interacting with the server, so it's the logical client to begin with.

- You need to have the MySQL software installed.
- You need a MySQL account so that you can connect to the server.
- You need a database to work with

The required software includes the MySQL clients and a MySQL clients and a MySQL server. The client program must be located on the machine where you will working. The server can be located on our machine although that is not required. As long as you have permission to connect to it the server can be located anywhere. In addition to the MySQL software you will need a MySQL account so that the server will allow you to connect and create us sample database and its table.

Microsoft SQL Server 2008 is a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration. In this article, we'll cover six of the more frequently used tool: Enterprise Manager, Query analyzer, SQL Profiler, Service Manager, Data Transformation Services and Books Online. Let's take a brief look at each:

Enterprise Manager is the main administrative console for SQL Server installations. It provides you with a graphical "birds-eye" view of all of the SQL Server installation on your network. You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.

Query Analyzer offers a quick method for performing queries against any of your SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.

SQL Profiler provides a window into the inner workings of your database. You can monitor many different event types and observe database performance in real time. SQL Profiler allows you to capture and replay system "traces" that log various activities. It's a great tool for optimizing databases with performance issues or troubleshooting particular problems.

Service Manager is used to control the MS SQL Server (the main SQL Server process), MSDTC (Microsoft Distributed Transaction Coordinator) and SQLServer– Agent processes. An icon for this service Manager to start, stop or pause any one of these services.

Data Transformation Services (DTS) provide an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats. The most commonly used DTS application is the "Import and Export Data" wizard found in the SQL Server program group.

Programming Languages

Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently. There are two major Python versions- **Python 2 and Python 3**. Both are quite different.

HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document

7.2. Implementation

Implementation of the system refers to the final installing of the package in its real environment, to the satisfaction of the indeed users and the operation of the system. It is the process of converting a new or revised system design to operation. It is the key stage in achieving successful new system. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to new system. It must therefore be carefully planned and controlled. Proper guidance should be imparted to the users so that he is comfortable in using the application.

Implementation Plan

The transformation from theoretical designs to working system is done in this stage. Developed package of system is tested with simple data, accurate error identification and then through proposed change from the user etc. a dress rehearsal working of system is done, so as the system is scrutinized, for pointing out errors and modifications required if any keeping in mind the expectations and specifications from the system.

Education And Training

The expectations from the system are achived by the people who will be involved to be confident of their role in the new system. The complexity of the system is directly proportional to the amount of training and education given for the user .Education is different from the training, as the user through education can be a part of development of the system. Education has the capability to make training more interesting and important contributions in the system changes.

Training just means to give user specific skills in order to meet their new job requirements. The role of system analyst in training will make it more understandable and effective. Training provides a better overview of new system and its present objectives.

Training Of The Application Software

Awareness about the new system is made to the users through training, and with the underlying philosophy of the system (screen design, flow, error types during inputs, validation checks etc.) application use the system, as the users of the system may be of at different levels of hierarchy.

Post Implementation Review

System performance v/s expected requirements are evaluated. The implementation problems if any is taken seriously and taken care of along with admiring the achievements , failures etc. The works done here are used to improve the efficiency and user friendliness of the system.

Security

System security is a branch of technology known as information security as applied to computers and networks. The objective of system security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The term system security, means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tempering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. The technologies of system security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a program with security in mind often imposes restrictions on that program's behavior.

Maintenance

Maintenance is making adaptation of the software for external changes (requirements changes or enhancements) and internal changes (fixing bugs). When changes are made during the maintenance phase all preceding steps of the model must be revisited.

There are three types of maintenance:

1. Corrective(Fixing bugs/errors)
2. Adaptive(Updates due to environment changes)
3. Perfective(Enhancements, requirements changes)

Maintenance is enigma of the system development. The definition of the software maintenance can be given describing four activities that are undertaken after the program is released for use.

The maintenance activity occurs since it is unreasonable to assume that software testing will uncover all in a large system. The second activity that contributes the definition of maintenance occurs since rapid changes are encountered in every aspects of computing. The third activity involves recommendation for new capabilities, modification to the existing functions and general enhancements when the software is used. The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability.

Testing

8.1.Testing Objectives

Testing is an important step in the software engineering process that could view rather than constructive. Testing is the process of executing a program with the intent of finding an error a good test is that has probability to find an as yet undiscovered error.

- A good case is one that has a high probability of finding an unpredictable error.
- A successful case is one that has a high probability of finding an unpredictable error.
- A good test case is one that provides solution to that unpredictable error.
- A test plan entailed the following activities. We prepare list plan.
- We specified condition for users acceptance testing.
- We prepared list data for program testing.
- Also we prepared list data transaction plan testing.
- Then we planned user training.
- Our programs were compiled and assembled.
- Job performance aids were prepared.

8.2.Testing Strategy

Software testing determines the correctness, completeness, and quality of software being developed. Validation refers to the process of checking that the developed software meets the requirements specified by the user. The activities involved in the testing phase basically evaluate the capability of that system meets its requirements. The main objective of software testing is to detect errors in the software. Errors occur if some part of the developed system is found to be incorrect, incomplete or inconsistent. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).It involves the execution of a software component or system to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- is sufficiently usable
- can be installed and run in its intended environments, and
- achieves the general result its stake holders desire

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software bugs (errors or other defects).Software testing can provide objective, independent information about the quality of software and risk of its failure to users and/or sponsors. Software testing can be conducted as soon as executable software (even if partially complete) exists. The overall approach to software development often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then

implemented in testable programs. In contrast, under an Agile approach, requirements, programming, and testing are often done concurrently.

White-box Testing

Tests are performed to ensure that all internal operations of the software are performed according to the specifications of the client. This is called White box testing. White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system– level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements. The details entered by the administrator are saved and stored in the database, and testing is done to verify whether the control of each form or action is working in the exact way.

Black-box Testing

Tests are performed to ensure that each function is working properly. This is referred to as Black-box testing. Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output without any knowledge of the test object's internal structure. Testing is conducted in the system so that the functions namely Login, sending requests, payment etc... are done properly.

Condition Testing

Test cases are derived to determine whether the logic conditions and decision statements are free from errors. Condition testing strategy is used to check if the operators used are correct and to verify conditions such as if an error message is displayed if a non-registered user is signed in to the app, or a user is registered without providing his body mass index value.

Loop Testing

This testing is used to check the variety of loops present in programming. The working of the loops such as while, for and do while are checked for its proper execution. The statements inside the loop body are executed line by line for every condition that satisfies the loop.

Unit Testing

Unit testing focused verification efforts on the smallest unit of software design, the module. This is also known as "module testing". The modules are tested separately. This testing is carried out during programming stage itself. In this testing step each module is found to be working satisfactorily as regard to the expected output from the module.

Project Aspect: User interfaces are tested for data acceptance. Each of the modules such as Login, item add modules etc are tested individually and found error free.

Integration Testing

Data can be lost across the interfaces; one module can have an adverse effect on the other; sub functions when combined, may not produce the desired major functions. The integration testing is a systematic testing for constructing the programs structure, while at the same time conducting tests to uncover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combines and tested as a whole. Here correction is difficult because the vast expenses of the entire program complicate the isolation of causes. Thus in the integration testing step, all the errors uncovered are corrected for the next testing steps.

Project aspect: Using integrated test plans prepared in the design phase of the system developed as a guide, the integration test was carried out. The modules are integrated and tested and all the errors found in the system were corrected for the next testing steps.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in specific format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. The output format of the screen is found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, output comes out as the specified requirements by the user. Hence output testing does not result in any correction in the system. Various reports are generated in graphical output format and being pictorial representation it is found more convenient to understand by the users of the system.

User Acceptance Testing

User acceptance testing of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. This is done with regard to the following points.

Objectives

- A good case is one that has a high probability of finding an unpredictable error.
- A successful case is one that has a high probability of finding an unpredictable error.
- A good test case is one that provides solution to that unpredictable error.
- A test plan entailed the following activities. We prepare list plan.
- We specified condition for users acceptance testing.

- We prepared list data for program testing.
- Also we prepared list data transaction plan testing.
- Then we planned user training.
- Our programs were compiled and assembled.
- Job performance aids were prepared.

Test Results

Test results emphasize how the actual results differed from the expected results. This suggests the need for re-testing, and to discover the source of differences. The test phase of systems development process involves the defining of the criteria by which the system will be tested and measuring the criteria against the acceptable failure rate. Individual modules are tested during the development itself. Errors detected are corrected and re-tested, and the project leader has verified the compliance. Each input, output and processes are tested to verify that it performs as specified in the design. The units in the system are re-compiled and errors found are corrected as indicated by the compiler. The tests are repeated until all known errors are eliminated and the program matched the design specifications. Separate tests are performed to ensure that program units are properly interfaced with each other to form a complete system.

Future Enhancement

9.1 Future Enhancement

In the future, the Missing Child Identification System can be enhanced by integrating advanced artificial intelligence techniques and real-time surveillance. By incorporating facial recognition with live CCTV feeds in public places such as railway stations, bus terminals, and shopping malls, authorities can automatically scan and detect missing children. Implementing improved deep learning models like FaceNet or ArcFace can enhance accuracy, ensuring better identification even under varying lighting conditions, occlusions, or aging effects. Additionally, mobile applications can be developed to allow citizens to instantly report sightings of missing children, enabling faster response times from law enforcement.

Another significant enhancement could be the integration of blockchain technology to ensure secure and tamper-proof data storage. This would help maintain the privacy of missing child records while allowing authorized agencies to access and update information seamlessly. Moreover, predictive analytics powered by AI can be utilized to analyze trends in child abduction or missing cases, identifying high-risk areas and enabling proactive measures to prevent future incidents. Automated alert systems can also be implemented to notify law enforcement and nearby users when a potential match is detected, significantly improving the efficiency of rescue operations.

.

Conclusion

10. Conclusion

The Missing Child Identification System using Deep Learning and Multiclass SVM provides a powerful and efficient solution for addressing the critical issue of missing children. By leveraging deep learning techniques such as CNN-based facial feature extraction and an SVM classifier, the system enhances the accuracy and reliability of child identification. The integration of a public portal allows citizens to contribute by uploading images, aiding law enforcement in their search efforts. With an impressive classification accuracy of 99.41%, the system significantly improves the chances of reuniting missing children with their families.

Furthermore, the proposed system overcomes the limitations of traditional missing child search methods by incorporating AI-driven automation and real-time matching capabilities. As technology continues to evolve, future enhancements such as real-time surveillance integration, blockchain security, and predictive analytics will further strengthen its effectiveness. By adopting such advanced solutions, authorities, NGOs, and the public can work together to reduce the number of untraced missing children, ultimately creating a safer society for the most vulnerable members of the population.

.

APPENDIX

11.1 Sample code

Views.py

```
def addnotification(request):
    return render(request,'child help line/addnotification.html')


@login_required(login_url='/')
def addnotification_post(request):    Heading=request.POST['textfield']
    Description=request.POST['textfield2']
    ob=notification_table()
    ob.heading=Heading
    ob.description=Description
    ob.date=datetime.datetime.now().today().date()
    ob.save()
    return
HttpResponse('''<script>alert('Added');window.location='/viewnotification'</script>''')


def delete_viewnotification(request,id):
    ob=notification_table.objects.get(id=id)
    ob.delete()
    return redirect('/viewnotification')


@login_required(login_url='/')
def editviewnotification(request,id):
    request.session['id']=id
    ob=notification_table.objects.get(id=id)
    return render(request,'child help line/editviewnotification.html',{"data":ob})


@login_required(login_url='/')
def editviewnotification_post(request):
    Heading = request.POST['textfield']
    Description = request.POST['textfield2']
    ob = notification_table.objects.get(id=request.session['id'])
    ob.heading = Heading
    ob.description = Description
    ob.save()
    return
HttpResponse('''<script>alert('Added');window.location='/viewnotification'</script>''')
```

Login.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:missingchild/user_reg.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:http/http.dart' as http;

import 'forgot_password.dart';
import 'homes.dart';
import 'ippage.dart';

void main() => runApp(MissingChildApp());

class MissingChildApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Missing Child ',
      theme: ThemeData(
        primaryColor: Colors.blueAccent,
        hintColor: Colors.blueGrey,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: LoginPage(),
    );
  }
}

class LoginPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return WillPopScope(
      onWillPop: () async {
        Navigator.pushReplacement(
            context, MaterialPageRoute(builder: (context) => IpHomePage()));
        return false;
      },
      child: Scaffold(
        extendBodyBehindAppBar: true,
        appBar: AppBar(
          title: Text('MISSING CHILD ', style: TextStyle(color: Colors.white)),
          backgroundColor: Colors.transparent,
          elevation: 0,
        ),
        body: Stack(
          children: [
            Positioned.fill(
              child: Container(
                decoration: BoxDecoration(
                  image: DecorationImage(
                    image: AssetImage('assets/image10.jpg'),
                    fit: BoxFit.cover,
                  ),
                ),
```

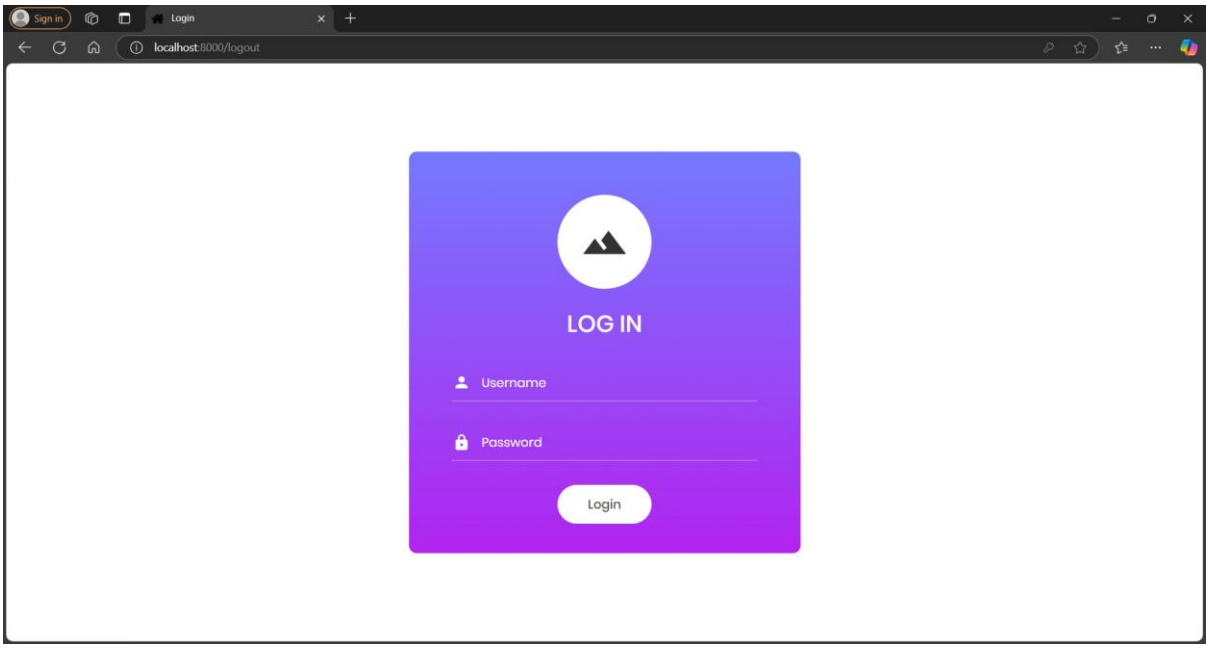
```
              ),
            ),
            Center(
              child: SingleChildScrollView(
                padding: EdgeInsets.all(20.0),
                child: LoginForm(),
              ),
            ),
          ],
        ),
      ),
    );
  }
}

class LoginForm extends StatefulWidget {
  @override
  _LoginFormState createState() => _LoginFormState();
}

class _LoginFormState extends State<LoginForm> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final _formKey = GlobalKey<FormState>();

  Object? get urlsl => null;

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: EdgeInsets.all(20.0),
      decoration: BoxDecoration(
        color: Colors.black.withOpacity(0.0), // Adjusted opacity for less
transparency
        borderRadius: BorderRadius.circular(20.0),
      ),
      child: Form(
        key: _formKey,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text(
              "Login to Missing Child",
              style: TextStyle(fontSize: 24.0, fontWeight: FontWeight.bold,
color: Colors.white),
            ),
            SizedBox(height: 30.0),
            _buildTextField(_emailController, "Username / Email", Icons.email),
            SizedBox(height: 20.0),
            _buildTextField(_passwordController, "Password", Icons.lock,
isPassword: true),
            SizedBox(height: 20.0),
            ElevatedButton(
              onPressed: () {
                if (_formKey.currentState!.validate()) {
                  sendData();
                }
              },
              style: ElevatedButton.styleFrom(
                padding: EdgeInsets.symmetric(vertical: 15.0),
                backgroundColor: Colors.teal,
                shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(10.0)),
```

```
              ),
              child: Text('Login', style: TextStyle(color: Colors.white,
fontSize: 18.0, fontWeight: FontWeight.bold)),
            ),
            SizedBox(height: 20.0),
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                TextButton(
                  onPressed: () => Navigator.push(context,
MaterialPageRoute(builder: (context) => SignUpForm())),
                  child: Text('Register', style: TextStyle(color:
Colors.white)),
                ),
                TextButton(
                  onPressed: () => Navigator.push(context,
MaterialPageRoute(builder: (context) => ForgetPassword())),
                  child: Text('Forgot Password?', style: TextStyle(color:
Colors.white)),
                ),
              ],
            ),
          ],
        ),
      ),
    );



  }

  Widget _buildTextField(
      TextEditingController controller, String label, IconData icon,
      {bool isPassword = false}) {
    return TextFormField(
      controller: controller,
      decoration: InputDecoration(
        labelText: label,
        labelStyle: TextStyle(color: Colors.white.withOpacity(0.9), fontSize:
16),
        filled: true,
        fillColor: Colors.white.withOpacity(0.2), // Transparent effect
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(15.0),
          borderSide: BorderSide.none,
        ),
        enabledBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(15.0),
          borderSide: BorderSide(color: Colors.white.withOpacity(0.5)),
        ),
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(15.0),
          borderSide: BorderSide(color: Colors.tealAccent, width: 2.0),
        ),
        prefixIcon: Icon(icon, color: Colors.white.withOpacity(0.9)),
        contentPadding: EdgeInsets.symmetric(vertical: 16, horizontal: 20),
      ),
      style: TextStyle(color: Colors.white, fontSize: 16),
      obscureText: isPassword,
      validator: (String? value) =>
      (value == null || value.isEmpty) ? "Please enter your $label." : null,
    );
  }
```

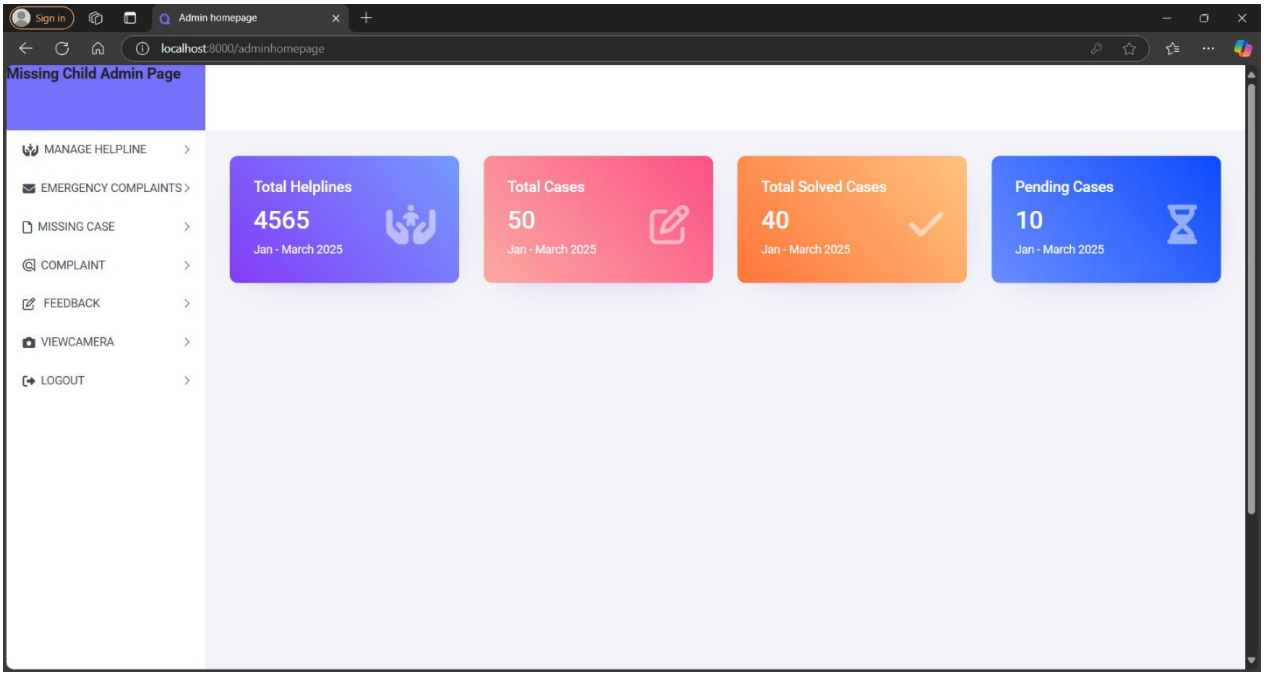
```
  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

  void sendData() async {
    String username = _emailController.text;
    String password = _passwordController.text;

    SharedPreferences sh = await SharedPreferences.getInstance();
    String url = sh.getString('url') ?? '';
    if (url.isEmpty) {
      Fluttertoast.showToast(msg: "Server URL not found in SharedPreferences");
      return;
    }

    final urls = Uri.parse("$url/android_login");
    print(urlsl);
    print('==================');

    try {
      final response = await http.post(urls, body: {'username': username,
'password': password});

      if (response.statusCode == 200) {
        final responseData = jsonDecode(response.body);
        if (responseData['status'] == 'ok') {
          Fluttertoast.showToast(msg: 'Login Success');
          sh.setString("lid", responseData['lid'].toString());
          if (responseData['type'] == 'user') {
            Navigator.pushReplacement(
                context, MaterialPageRoute(builder: (context) =>
MissingHome()));
          }
        } else {
          Fluttertoast.showToast(msg: 'User Not Found');
        }
      } else {
        Fluttertoast.showToast(msg: 'Network Error: ${response.statusCode}');
      }
    } catch (e) {
      Fluttertoast.showToast(msg: "Error: ${e.toString()}");
    }
  }
}
```

11.2 Screenshot

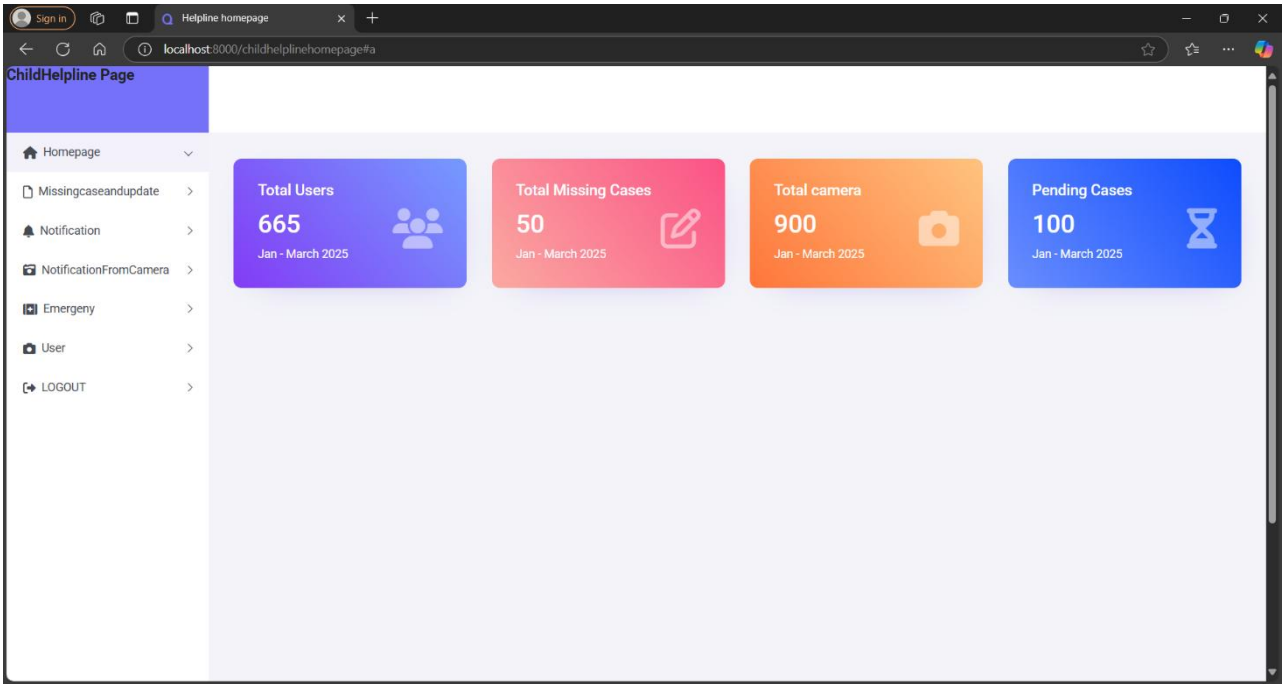
Login page:



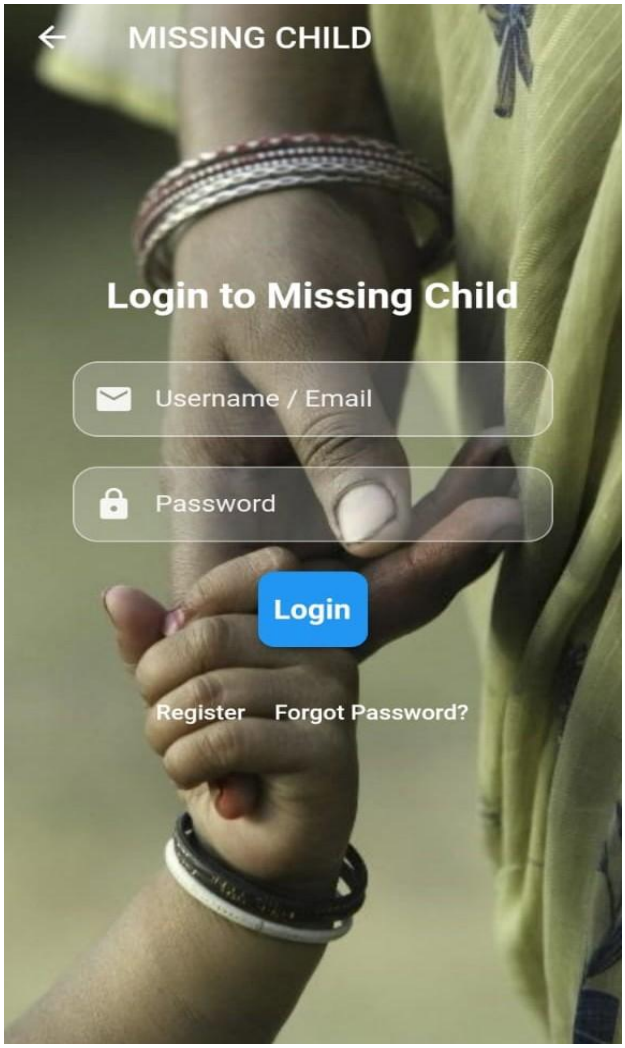
Admin home:

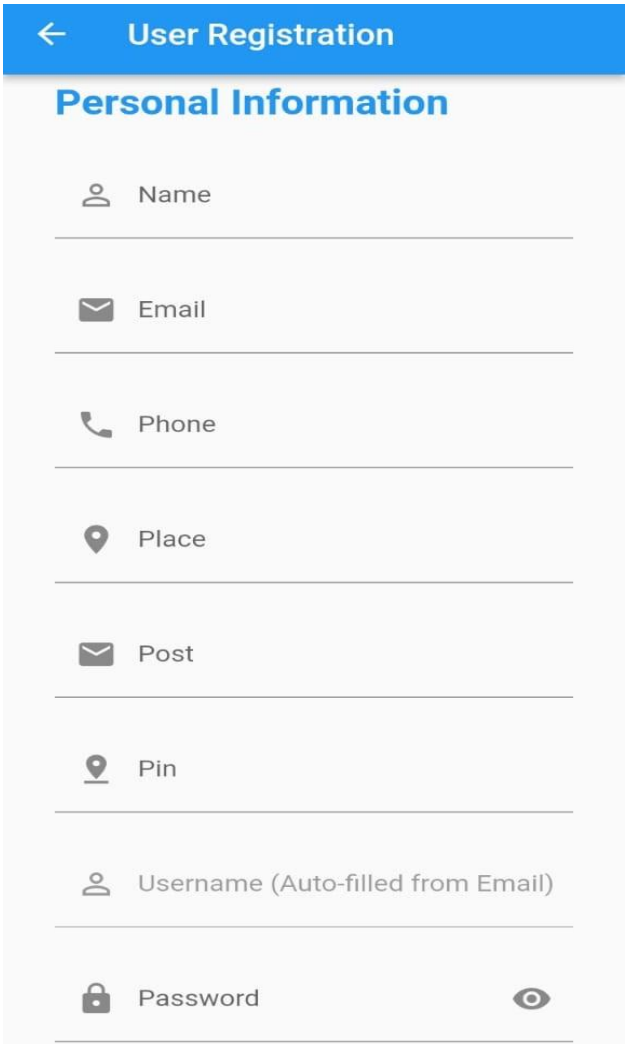

Sign in
Admin homepage
localhost:8000/adminhomepage
Missing Child Admin Page
MANAGE HELPLINE
EMERGENCY COMPLAINTS
MISSING CASE
COMPLAINT
FEEDBACK
VIEWCAMERA
LOGOUT
Total Helplines
4565
Jan - March 2025
Total Cases
50
Jan - March 2025
Total Solved Cases
40
Jan - March 2025
Pending Cases
10
Jan - March 2025

Child helpline home:

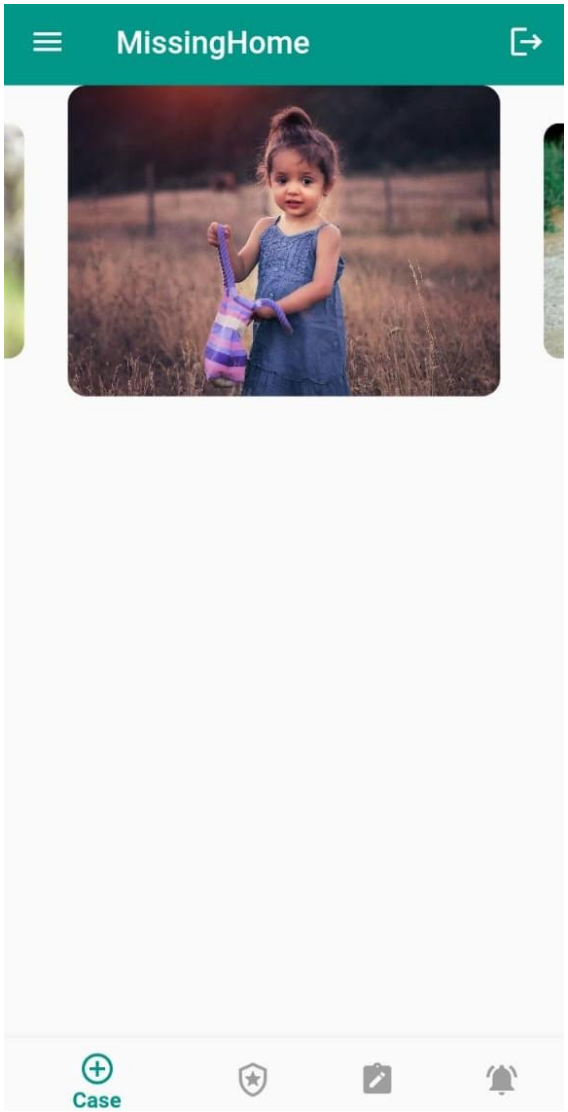


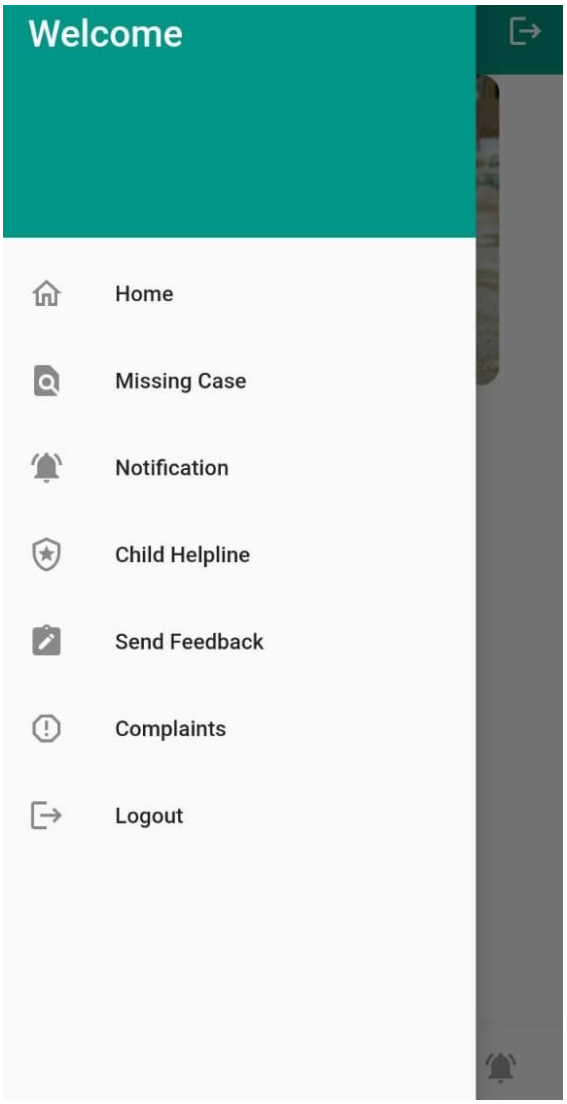
User Login and Registration:





User Home:


MissingHome
Case


Welcome
Home
Missing Case
Notification
Child Helpline
Send Feedback
Complaints
Logout

Bibliography

12.1 Bibliography

Websites:

- https://github.com/
- https://developer.android.com/
- https://www.w3schools.com/