

Missing You My Beloved

1. Introduction

1.1 Purpose

We plan to create a decentralized system in which each volunteer can allocate a particular amount of space in his computer to this noble cause and provide an effective and easy way for police to search for a missing person.

1.2 Scope

- Create different user such as contributor, police department and administrator with their respective access levels.
- Providing a secure way to login for users to use this application preferably with 2 way authentication.
- Ability to do a background check of all the users who have registered with us.
- Ability to store the information of missing people according to their last seen location.
- Police department can access the database and if required can modify the last seen as observed.
- It reduce the manual efforts of police department to find the guardians of missing people.
- Provide analytics about the data of the missing people like number of missing person, for how long they are missing, the average time taken to find them and so on.

1.3 Definitions, Acronyms and Abbreviation

MP	-	Missing Person
LSS	-	Leaf search service
DSS	-	District search service
SCM	-	Service control manager
DHS	-	Database handling service
MD	-	Missing database
FD-		Foreign database
PK	-	Primary Key
NN	-	Not Null
UQ	-	Unique
UN	-	Unsigned Number
AI	-	Auto Increment
G	-	Generated

1.4 Technologies to be used

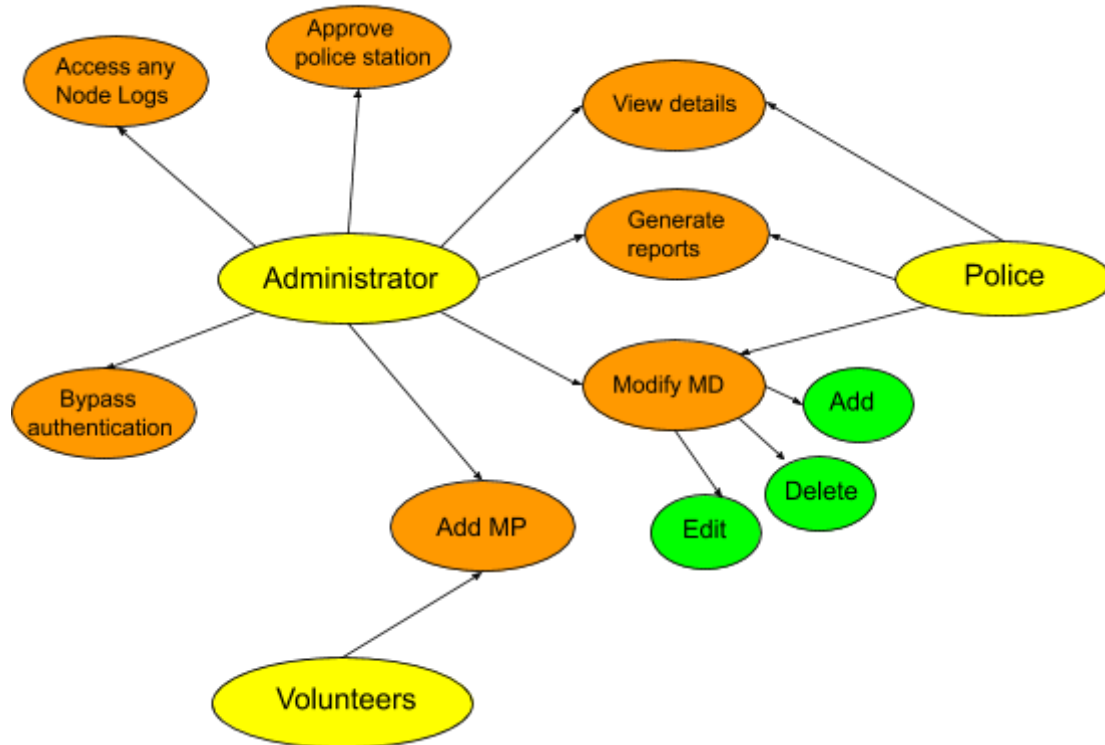
- Database application - PostgreSQL
- Web framework - Django
- Front-end Libraries - Bootstrap, JQuery ,JPopper and others if required.
- Processing and Back-end - Python 3.7
- Web deployment server - Apache Framework 2.0
- Designing Tool - Adobe XD

1.5 Overview

We plan to create a decentralized system in which each volunteer can allocate a particular amount of space in his computer to this noble cause. These nodes will keep the complete data encrypted so that people can access this whenever and as they require. These nodes will contain data hashed based on the shape of the image of the missing child. Once this is done there will be a master node monitoring these sub-nodes that will be allocated at district level. This node will contain the information off which node contains which information. Above this level there will be a master node at state level keeping track of all the district level nodes and what information it contains. This structure will be carried over to country level followed by continent and then the world. This structure will be beneficial to us considering that the MP over a particular region is most likely to be found over that same. In the above structure to find a person in district level we have to search only in one level and then the data in the computer of the destination. Present of upper level make search fast and easy. By doing so we will be able to reduce the load of the above level. These super nodes will always just need to return the IP address of the next node to be searched. When we have the IP of the end-node and the required information we will transfer the complete data between the two computers in peer-to-peer manner. To access the information we will have a complete suite of software which will be only capable to connect to the super nodes and decrypt the information transferred. This method of indexing will reduce the amount of data that need to be transferred by higher level nodes so that they can carter to more number of request whereas the lowest level will need to transfer more data independently through peer-to-peer method as it has to carter less amount of requests.

2. Overall Description

2.1 Use-Case Model Survey



Administrator- can access the data uploaded by the user and use it to find the missing person. They are responsible for giving access of data to the police stations.

- **Bypass authentication** - a new user account will be created or a user will log into an already existing user account.
- **Access any Node Logs** - activities at each and every node can be tracked and checked.
- **Approve police stations** - police stations will be given access to the database they want to access.
- **View details** - a new user account will be created or a user will log into an already existing user account.
- **Generate reports** - a new user account will be created or a user will log into an already existing user account.
- **Modify Missing Data** - can add, delete or update in the database.
- **Add Missing Persons** - *(explained below)*

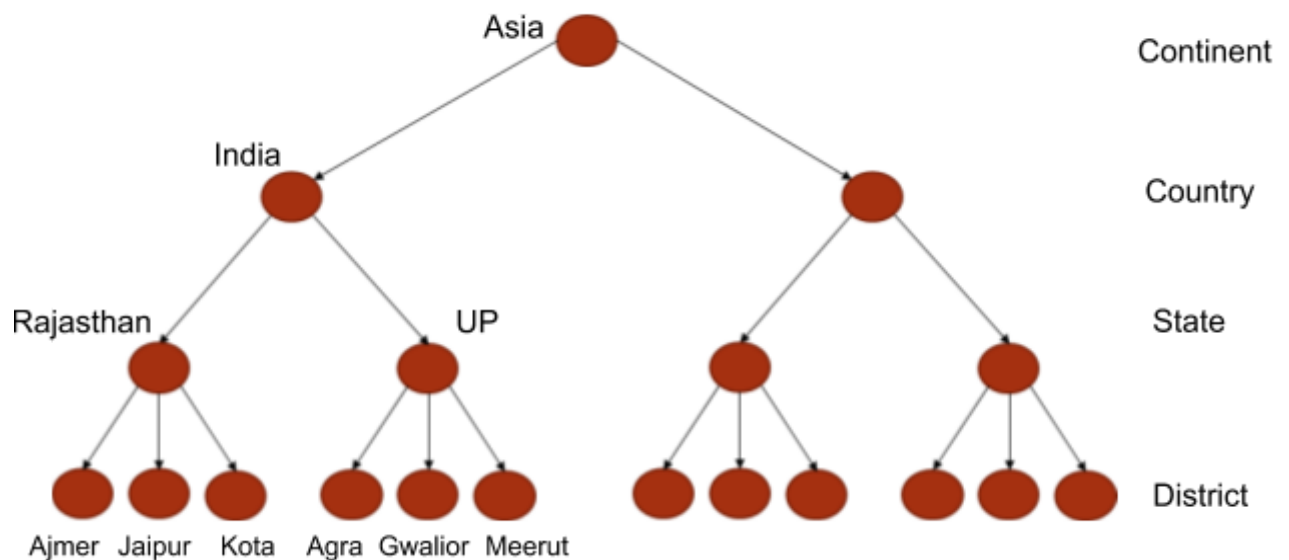
Volunteers – provide data by reporting missing people.

- **Add Missing Persons** – voluntarily report the missing persons.

Police Department -

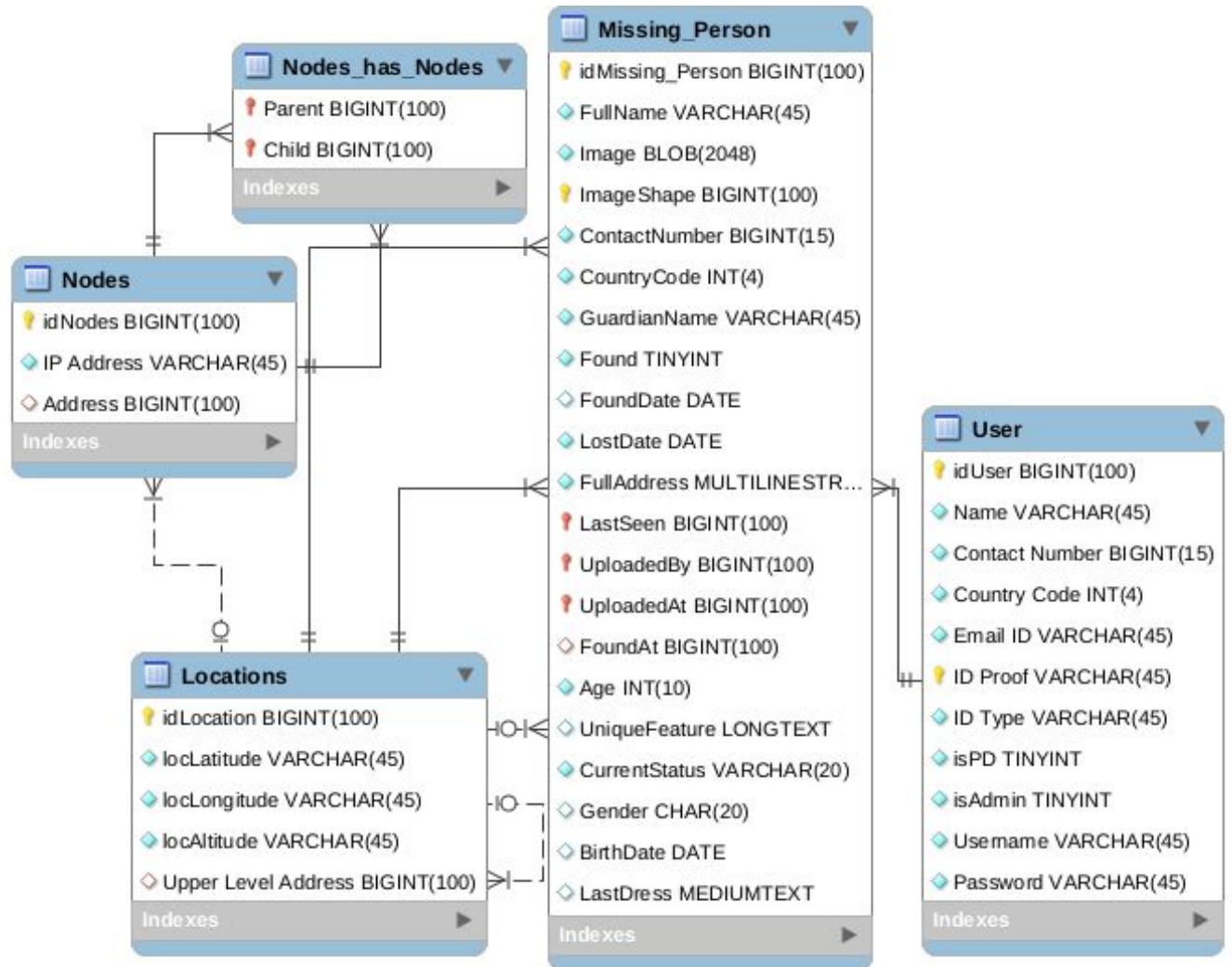
- **View details** - a new user account will be created or a user will log into an already existing user account.
- **Generate reports** - a new user account will be created or a user will log into an already existing user account.
- **Modify Missing Data** - can add, delete or update in the database.

2.2 Architecture Diagram



* levels can be increased or decreased as and when required

2.3 ER Diagram



2.4 Architecture Table

Nodes_has_Nodes - Table										
Table	Columns	Indexes	Foreign Keys	Triggers	Partitioning	Options	Inserts	Privileges		
Column Name		Datatype		PK	NN	UQ	BIN	UN	ZF	AI
Parent		BIGINT(100)		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Child		BIGINT(100)		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

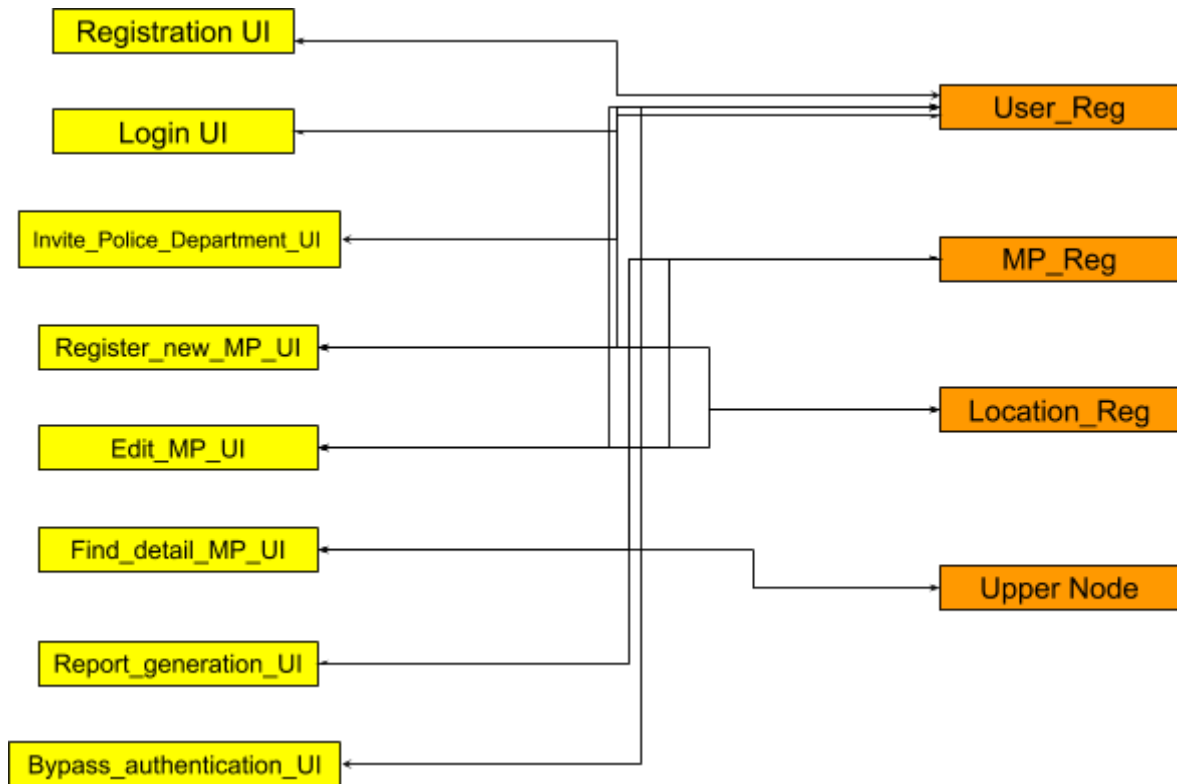
Nodes - Table										
Table	Columns	Indexes	Foreign Keys	Triggers	Partitioning	Options	Inserts	Privileges		
Column Name		Datatype		PK	NN	UQ	BIN	UN	ZF	AI
idNodes		BIGINT(100)		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IP Address		VARCHAR(45)		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Address		BIGINT(100)		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Missing_Person - Table ✕												
Table	Columns	Indexes	Foreign Keys	Triggers	Partitioning	Options	Inserts	Privileges				
Column Name		Datatype		PK	NN	UQ	BIN	UN	ZF	AI	G	
🔑 idMissing_Person		BIGINT(100)		✓	✓							
🔗 FullName		VARCHAR(45)			✓							
🔗 Image		BLOB(2048)			✓	✓						
🔑 ImageShape		BIGINT(100)		✓	✓	✓					✓	
🔗 ContactNumber		BIGINT(15)			✓			✓				
🔗 CountryCode		INT(4)			✓			✓				
🔗 GuardianName		VARCHAR(45)			✓							
🔗 Found		TINYINT			✓				✓			
🔗 FoundDate		DATE									✓	
🔗 LostDate		DATE			✓							
🔗 FullAddress		VARCHAR(100)			✓							
🔑 LastSeen		BIGINT(100)		✓	✓							
🔑 UploadedBy		BIGINT(100)		✓	✓							
🔑 UploadedAt		BIGINT(100)		✓	✓							
🔗 FoundAt		BIGINT(100)										
🔗 Age		INT(10)			✓			✓				
🔗 UniqueFeature		LONGTEXT										
🔗 CurrentStatus		VARCHAR(20)			✓							
🔗 Gender		CHAR(20)										
🔗 BirthDate		DATE										
🔗 LastDress		MEDIUMTEXT										

Locations - Table ✕												
Table	Columns	Indexes	Foreign Keys	Triggers	Partitioning	Options	Inserts	Privileges				
Column Name		Datatype		PK	NN	UQ	BIN	UN	ZF	AI	G	
🔑 idLocation		BIGINT(100)		✓	✓	✓						
🔗 locLatitude		VARCHAR(45)			✓							
🔗 locLongitude		VARCHAR(45)			✓							
🔗 locAltitude		VARCHAR(45)			✓							
🔗 Upper Level Address		BIGINT(100)										

User - Table ✕												
Table	Columns	Indexes	Foreign Keys	Triggers	Partitioning	Options	Inserts	Privileges				
Column Name		Datatype		PK	NN	UQ	BIN	UN	ZF	AI	G	
🔑 idUser		BIGINT(100)		✓	✓							
🔗 Name		VARCHAR(45)			✓							
🔗 Contact Number		BIGINT(15)			✓	✓		✓				
🔗 Country Code		INT(4)			✓			✓				
🔗 Email ID		VARCHAR(45)			✓	✓						
🔑 ID Proof		VARCHAR(45)		✓	✓	✓						
🔗 ID Type		VARCHAR(45)			✓							
🔗 isPD		TINYINT			✓							
🔗 isAdmin		TINYINT			✓							
🔗 Username		VARCHAR(45)			✓	✓						
🔗 Password		VARCHAR(45)			✓	✓						

2.5 Architecture Diagram



2.6 Assumption and Dependency -

- The IP of all the nodes should be Static IP else they should have a DHCP Server set-up for Dynamic IP.
- It would be ideal if the IP of devices are not changed frequently as it will cause updation of Index every time.

3. Specific Requirements

3.1 Use-Case Reports

Administrator- can access the data uploaded by the user and use it to find the missing person. They are responsible for giving access of data to the police stations.

- Bypass authentication
- Access any Node Logs
- Approve police stations
- View details
- Generate reports
- Modify Missing Data
- Add Missing Persons

Name of Use Case - Bypass authentication

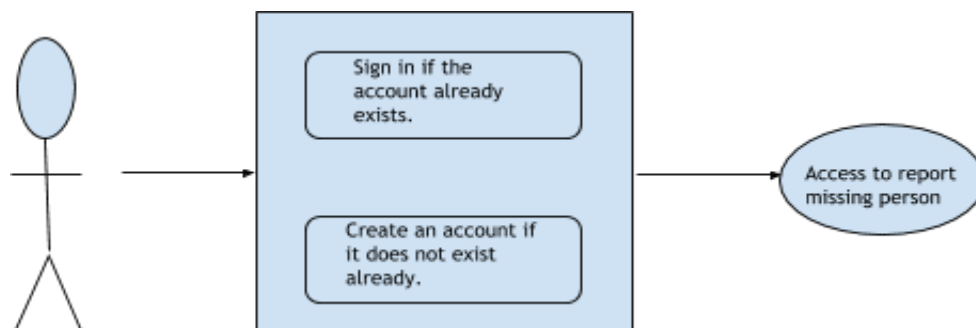
Description – a new user account will be created or a user will log into an already existing user account.

Pre Condition – The Administrator should be logged into the system.

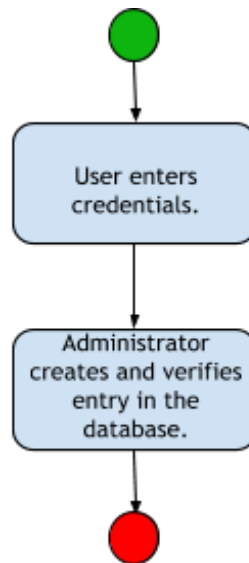
Normal Flow of Events –

- A form will open, necessary information will be entered
- A query will be fired to the database.
- If the account already exists, then the details will be loaded from the database.
- If the account does not exist already, it will be created and added to the database. The user then will be asked to log into the account to avail the services.

Use Case Diagram -



Workflow Diagram -



Name of Use Case – Access any Node Logs.

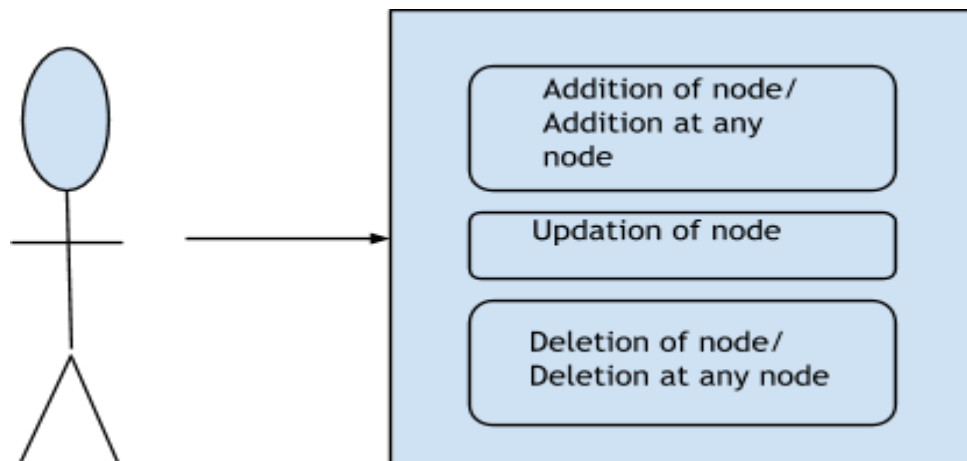
Description – activities at each and every node can be tracked and checked.

Pre Condition The Admin should be logged in the system.

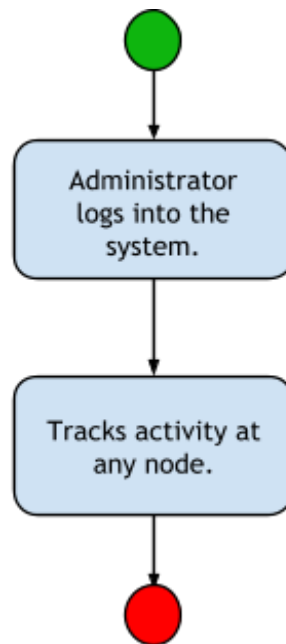
Normal Flow of Events –

- The addition of a new node or report of a missing person is recorded for each node.
- Admin can view these records.
- Admin can also undo any changes made to any node if he wants.

Use Case Diagram -



Workflow Diagram -



Name of Use Case - Approve Police Stations

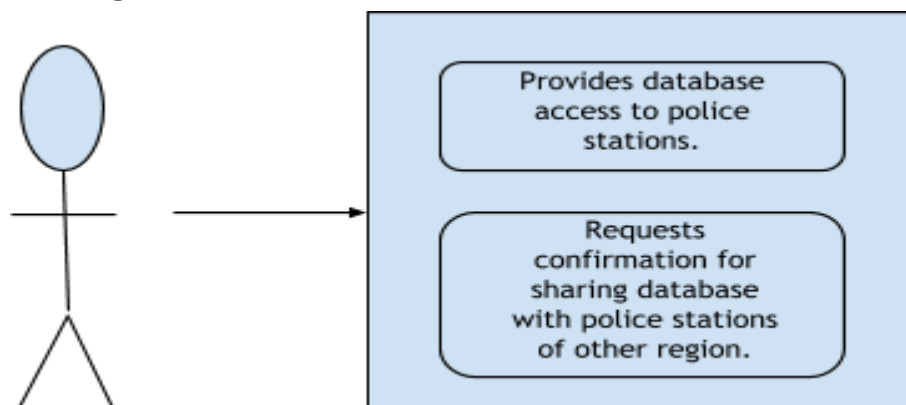
Description – police stations will be given access to the database they want to access.

Pre Condition – The Administrator should be logged into the system.

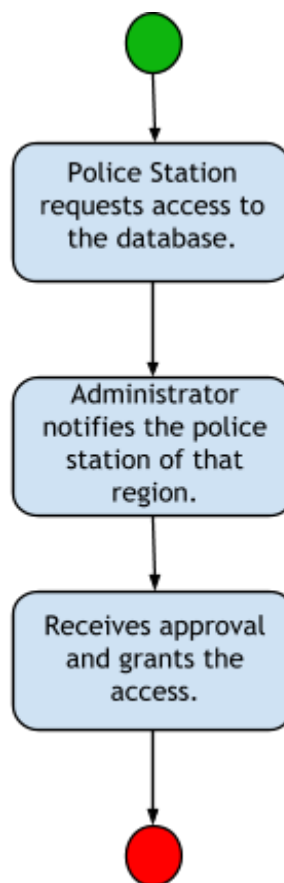
Normal Flow of Events –

- Police stations will be given access to the database of their region.
- If they wish to access any other database, they will request the administrator to do so.
- The administrator will notify the police station who already have access to the database and ask for confirmation or approval.
- Once the approval is received, the database will be shared.

Use Case Diagram -



Workflow Diagram -



Name of Use Case - View Details

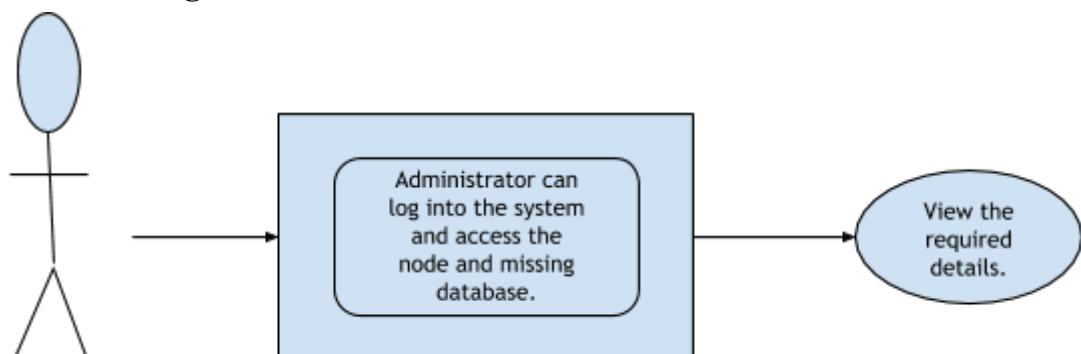
Description – administrator can view details of any node or any data in the missing database.

Pre Condition – The administrator should be logged into the system.

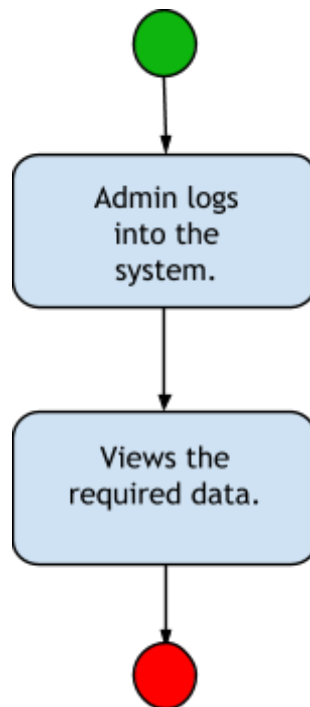
Normal Flow of Events –

- Log into the system.
- View data from any node in the database.

Use Case Diagram -



Workflow Diagram -



Name of Use Case - Generate reports

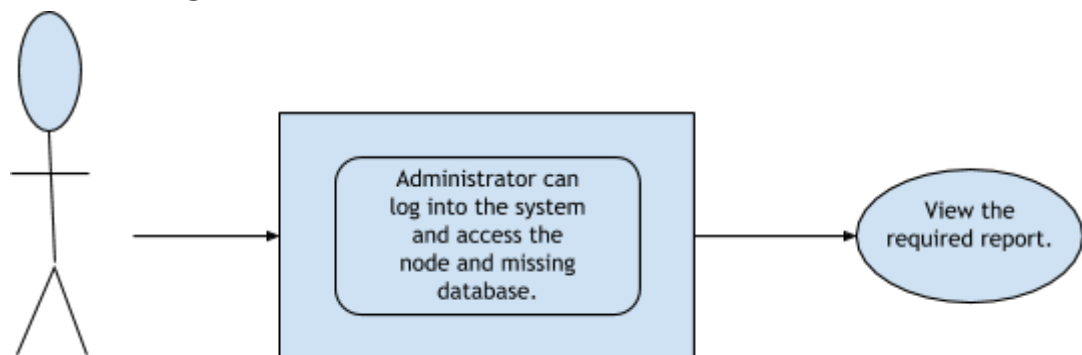
Description – administrator can produce a summary of the data in the database for any node.

Pre Condition – The administrator should be logged into the system.

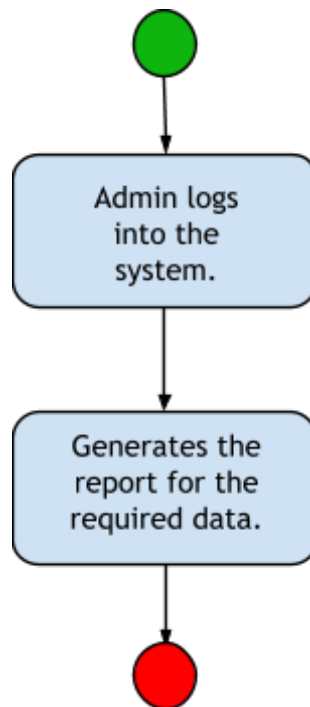
Normal Flow of Events –

- Log into the system.
- View consolidated information for the required data.

Use Case Diagram -



Workflow Diagram -



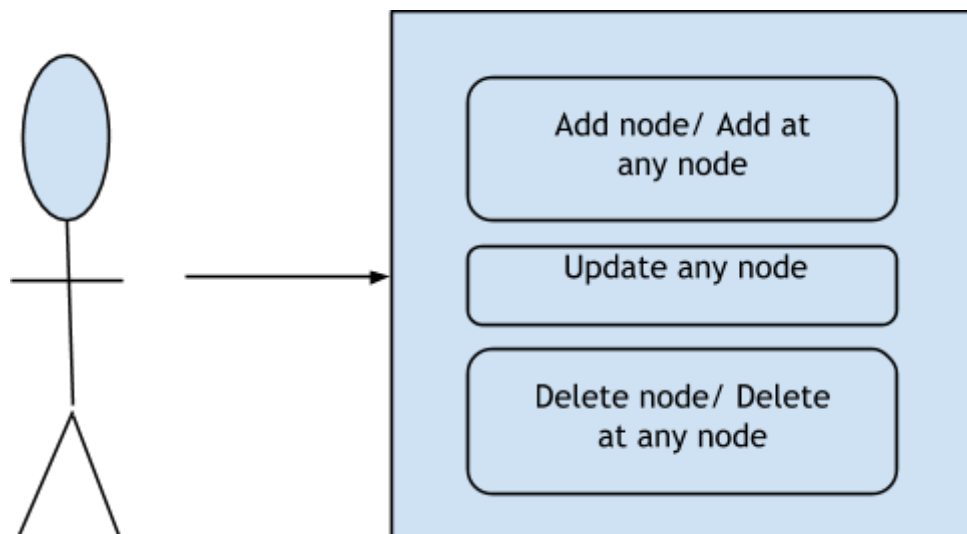
Name of Use Case - Modify Missing Data

Description – can add, delete or update in the database.

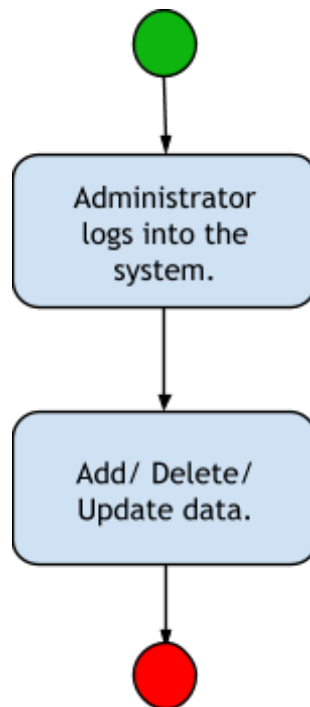
Normal Flow of Events –

- Log into the system.
- Add/ Delete/ Update data in the database.

Use Case Diagram -



Workflow Diagram -



Volunteers – Provide data by reporting missing people.

- **Add Missing People** – voluntarily report the missing people.

Name of Use Case – Add Missing People

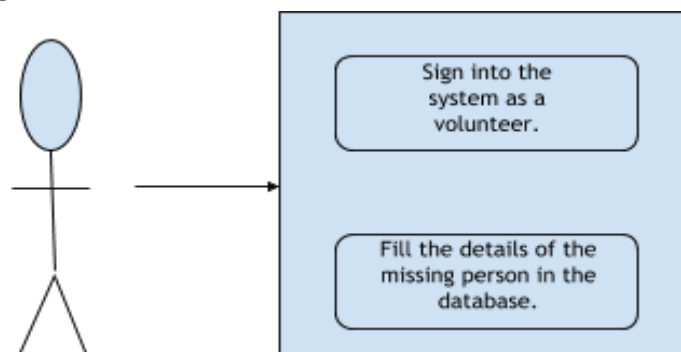
Description – addition to the database

Pre Condition –The volunteer must be logged into the system

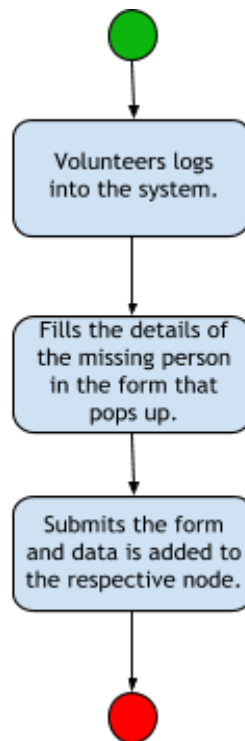
Normal Flow of Events –

- As the volunteer chooses to report a missing person, a form pops up.
- The details filled in the form are used to create an entry in the database.

Use Case Diagram -



Workflow Diagram -



Police Department - Use the database to find the details of missing person and update the details of those found

- **View details** - a new user account will be created or a user will log into an already existing user account.
- **Generate reports** - a new user account will be created or a user will log into an already existing user account.
- **Modify Missing Data** - can add, delete or update in the database.

Name of Use Case - View Details

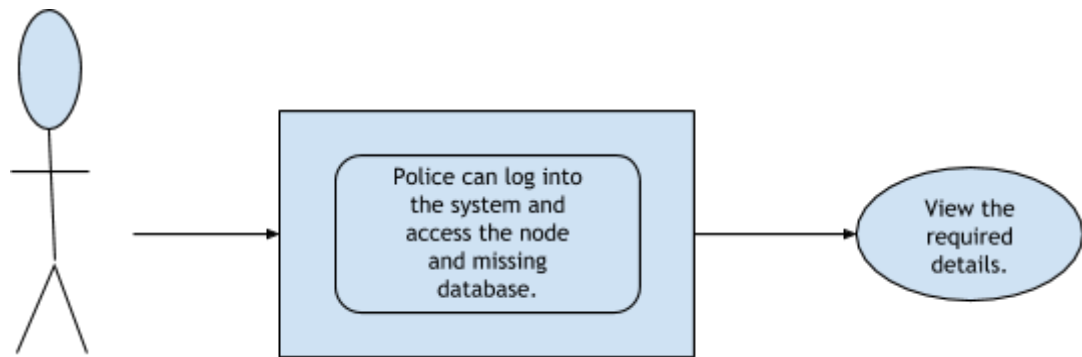
Description – Police can view details of any node or any data in the missing database.

Pre Condition – The police should be logged into the system.

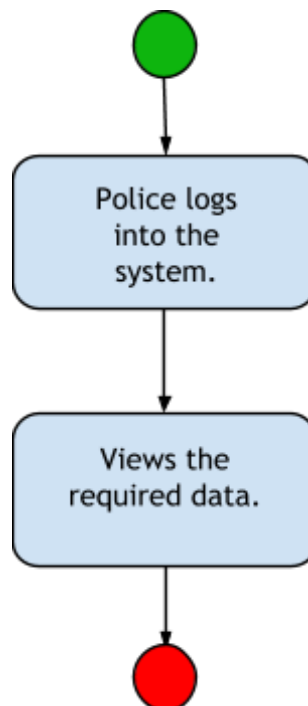
Normal Flow of Events –

- Log into the system.
- View data from any node in the database.

Use Case Diagram -



Workflow Diagram -



Name of Use Case - Generate reports

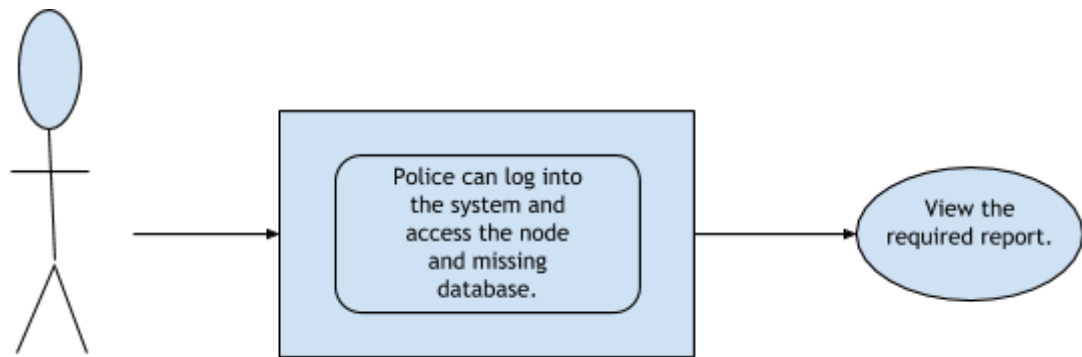
Description – Police can produce a summary of the data in the database for any node.

Pre-Condition – The Police should be logged into the system.

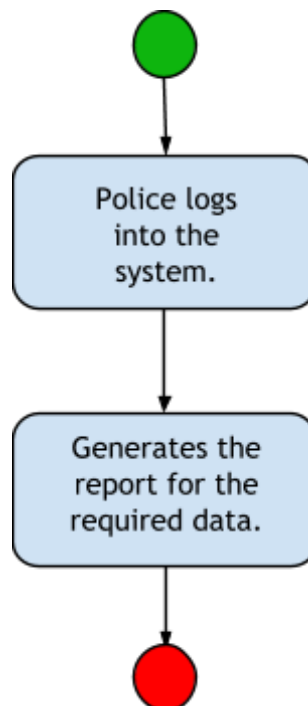
Normal Flow of Events –

- Log into the system.
- View consolidated information for the required data.

Use Case Diagram -



Workflow Diagram -



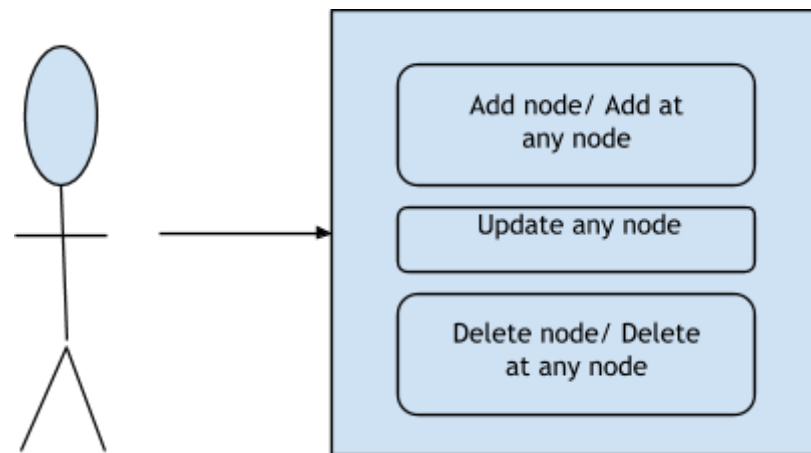
Name of Use Case - Modify Missing Data

Description – can add, delete or update in the database.

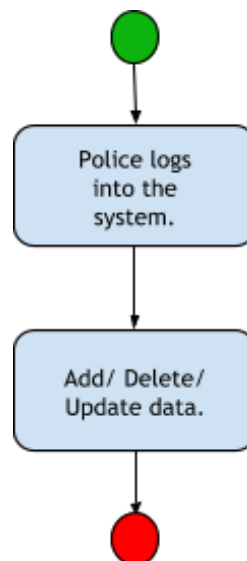
Normal Flow of Events –

- Log into the system.
- Add/ Delete/ Update data in the database.

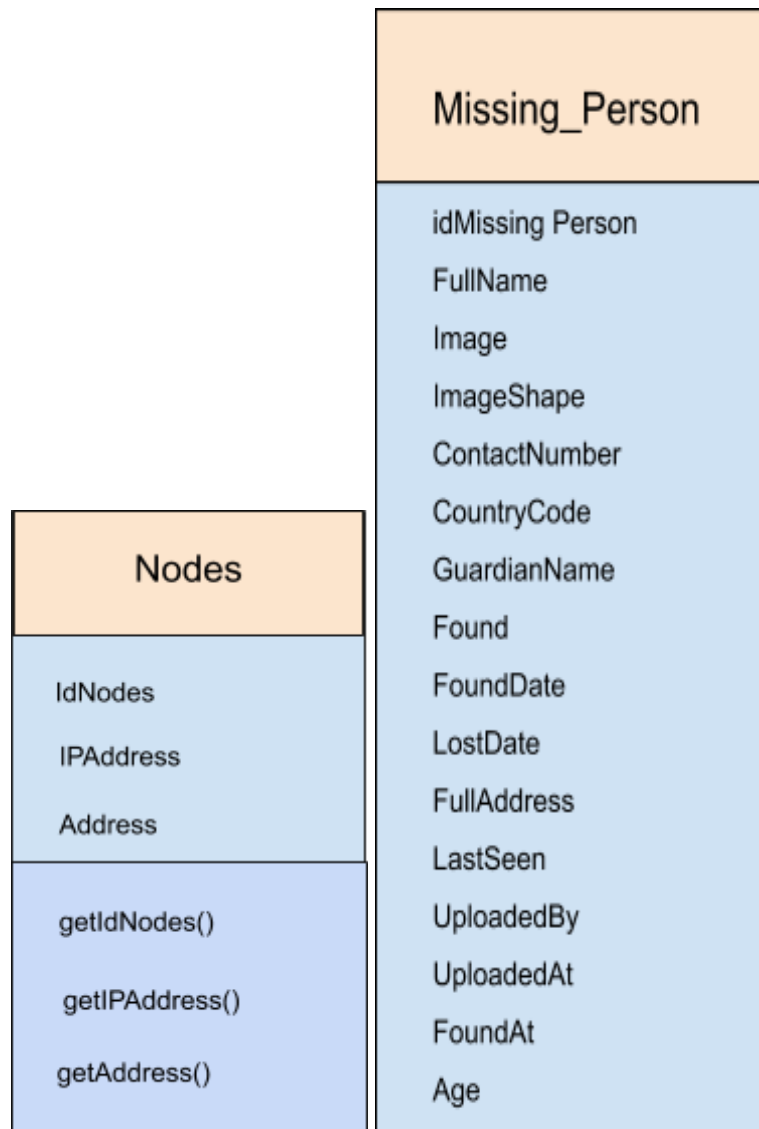
Use Case Diagram -



Workflow Diagram -



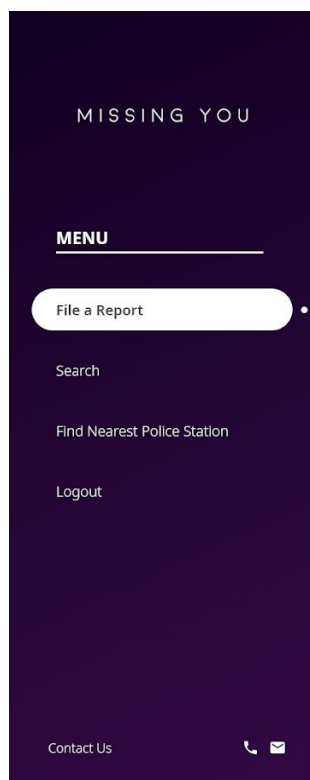
3.2 Class Diagrams




Locations	User
idLocations locLatitude locLongitude locAltitude UpperLevelAddress	idUser Name ContactNumber CountryCode EmailID IDProof IDType isPD isAdmin Username Password
getLocationID() getlocLatitude() getlocLongitude() getlocAltitude()	

4. User Interface

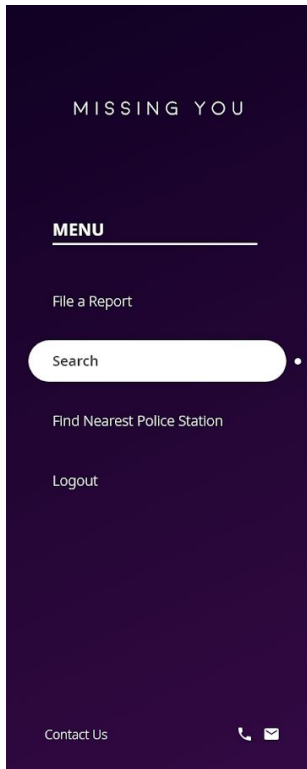
4.1 Reporting of Missing Person



File a Report

Full Name	<input type="text"/>		
Contact Number	+91 <input type="text"/>	Date Of Birth	<input type="text"/>
Guardian Name	<input type="text"/>		
Image Upload	<div>  <div>Image size up to 4 MB</div> <div>Upload Image</div> </div>		
Full Address	<input type="text"/>		
Lost Date	<input type="text"/>		
Unique Features	<input type="text"/>		

4.2 Searching For Missing Person



Search

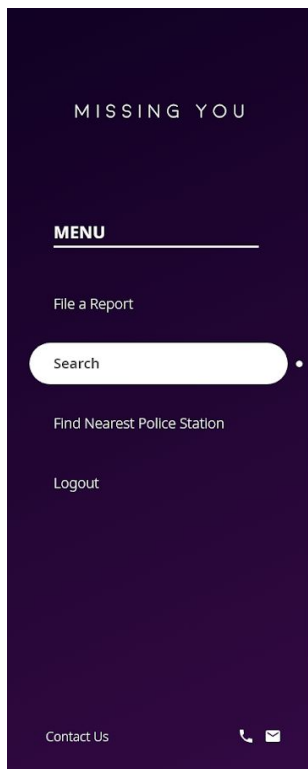
Mark Desmond

| or |

Search by Image



— End of Results —



Search

Mark Desmond

| or |

Search by Image

[Back to Results](#)

Mark Desmond

Male, 19 **Still Missing**

Lost On	29/02/2019
Contact	+91 9477909374
Address	221 - B, Baker Road, Sector 16 Noida, Gurgaon - 302031 India
Gaurdian Name	Mike Desmond
Unique Features	Birthmark on upper right thigh Scar on forehead
Birth Date	15/07/2000
Last Dress Worn	Blue checkered shirt, Dark blue denim jeans, white wrist band on left



[View Full Image](#)

More Pictures



[View All](#)

5. Working

5.1 Foreground

Installation, registration and authentication

- Each user will require an account to activate his application because that will be the main source which will tell the higher level nodes about its children nodes. ***So with every installation the user will be asked to either sign-up/sign-in.*** If he already has an account and decides to sign-in all the previous information will be loaded and verified for the same location as before. On the other hand if he decides to sign-up for an account his sign-up request will be sent to higher level nodes and the following steps will be taken -
 - His request will be ***matched to the district level node*** based on the district he/she chooses. This district level node will create an entry of this application and account.
 - The above node will send the geographical information to it and save the same details with the associated account. This will help in relogin of the user after he/she has logged-out.
- If a user migrates to a new location ***there will be an option to change his parent node*** and doing so will associate his account and application to the new node and deleting the information from the old node.

Reporting a missing person

Well for reporting a missing person (let's call him/her MP) application user will undergo three stages:

- I. Filling details of MP: To file a missing case application would need some details of that person like ***photo, full name, age, sex, height, skin complexion etc.*** This details would be then used to confirm his/her identity.
- II. Locating where MP was last seen: To start the search for him/her application would need to know from where the MP was gone missing. Which if not entered the current location of the person entering the details will be taken. From this point search will begin. ***The last seen location will be used to find the nearest leaf node.*** This node will be served as the starting node for search. However if this starting node is far away then nearest node will store MP data and address of this node in Foreign Database (say FD, another term to denote important thing). The nearest node will migrate this data of MP from FD to its starting node.
- III. Confirm upload : Asks a final confirmation on all the details client has entered. After confirmation MP details will be delivered in what we call as Missing

Database (shortened as MD) of the leaf node. A node's MD actually stores all the MP which were last seen near that node. Finally its search will be initiated.

5.2 Background

User Login and Signup

While Signing-up the user will require to provide the following details.

- Government Issued ID card (Can be restricted to particular category if required)
- E-Mail ID
- Phone Number

Both the E-Mail ID and the Phone Number will be authenticated with OTP.

While signing-in the user will be asked to enter the same details as before every time. If the details match then an OTP will be sent to the phone number and that OTP will be required for a successful log-in.

Storing and accessing data in the nodes

- Each leaf node will have an index of all the elements that it is storing in its database. This index will be based on the alphabetical names of the MPs. There will be a secondary index based on the shape of the image of the missing people. We are deciding this because most of the queries will be based on these types of search only. Sometimes the people will have just *the name of the person* and the rest of time only *the image of the person*.
- Each parent node will have the merged index of all its child nodes. This will be helpful as it will be able to tell the node that needs to be searched in *O(1) time*. During any updation operation the updation has to be done till the highest level.
- For searching the database will primarily be searching on the image's shape or the name of the person. The search will be based mainly by these two service
 - I. **Leaf Search Service(LSS)** - It will first search for the information in the index of the local node itself. If it finds so then it will be great and it will return the IP address of himself.
 - II. **District Search Service(DSS)** - If not it will make query to the district level node. That node will give the IP of the child node that is most prominent to have the data stored in it. Then it will search the index of that node. If it is found the IP address of that node is sent to the search request node. If still it is not found this step kept on following up to the higher level till we get the IP of the node that is an exact match.
- Once we have the IP from the above method, the node will connect with that IP for a *peer-to-peer data transfer* and downloads the complete information of the data from it.

- Making this kind of modal the above layers will have less load to handle and all the high load data transfer will go to peer-to-peer connection reducing the load level extremely low. Moreover, *it is most likely that a person lost in a particular locality will be found in that same locality* for most of the cases so similarly over here *we will not need to access the higher level node on most of the cases.*

Connections and Networks

- All the communication will be made on **TLS** verified connection. This secure connection will only be allowed to be created between the **known IP address** that were stored in the database during account creation.
- All communications are encrypted in **AES256**. This is to ensure that unauthorised users or fake nodes can not modify data or system variables.

Main Server

Main server will be accessible to all nodes every time and it will contain these three things :

- I. Information of Country, State, District, Name, Mobile Number and Email ID of each registered user.
- II. Information of each and every node like its geographical location(position coordinates), node id and its logs. Where logs are necessary in case any node fails.
- III. Index column of MD of nodes along with their pointer(IP address) for searching an MP of different node.

Main server will contain data in hash tables for faster access. Its data will be managed using SQL as it is very efficient and fast for even large data.

Nodes

Leaf nodes have two services running parallelly and independently which are -

- I. **Database Handling Service (DHS)** - DHS handles the Missing Database (MD) and Foreign DataBase (FD). MD contains info of missing persons like their Full names, Face photo (8-bit color lossless PNG compressed image), last seen location, contact-if-found phone number, etc. Updation of MD is more frequent than that of FD. Although sending FD dataframes to another node makes it comparatively more slow process on ideal working conditions.
- II. **Search Service** - Search service will be the most prominent work done by the leaf node. To pull this off it will maintain two indexes one based on the shape of the image of the MP and the name of the person.

Other than Search service some nodes (other than leaf nodes) also have Connection service. This service is responsible for connection in Backbone Networks. Backbone networks are direct connections to siblings. It helps to reduce communication overload from parent node.

Node Failure or Data corruption

- To escape from issues arising out of node failure a regular data backup is facilitated. Every (say) 4th hour all data is compressed and stored to local storage. This task is owned by Service Control Manager(SCM) a daemon process. SCM also manages logs and monitors the activities of other services. If any service throws an error it halts other services and informs parent node and main server with a ***SYSFAIL*** message. After which node is manually started.
- If SCM detects data corruption it immediately restores session to most recently backed up one and signals ***SYSRPT*** message to main server and parent. SCM buffers all the activities when system is undergoing restoring. So that -
 - If a new MP was recorded in between restoring session it reads SCM buffer and requests the client to try that again.
 - If a MP is found in between this then parent can inform it in reply ***SYSRPT*** message (since whenever an MP is found parent is notified first).