SPL-2 Software Requirement Specification and Analysis Report, 2024

Personal Health Tracker using Automated Report Scanning

Submitted by

Md. Mahfuz Ibne Ali Ayon
BSSE - 1421

&

Mohammad Ismail Hossain
BSSE - 1433

Supervised by

Dr. Ahmedul Kabir
Associate Professor
Institute of Information Technology
University of Dhaka

(Supervisor’s Signature)

Submission Date: 09/12/2024

1. Introduction

In today’s busy world, health has become one of the most disregarded factors in a person's life. Being regularly updated about personal health is getting challenging day by day for most people. As a result, complex health issues are occurring frequently and prevention is becoming difficult.This Software Requirement Specification (SRS) document delves into the detailed requirements for the Personal Health Tracker using Automatic Report Scanning, highlighting its commitment to create an application to help managing personal health related informations.

1.1 Purpose

The purpose of this SRS document is to outline the functional and nonfunctional requirements, features, and system design of the Personal Health Tracker using Automatic Report Scanning. It serves as a formal agreement between stakeholders and the development team, ensuring a shared understanding of the system's objectives and specifications.By detailing requirements for registration, authentication, Image Capturing, History management and hosting, notifications, and user profiles, this document aims to guide the development process toward creating an application that meets the needs of all users effectively.

1.2 Intended Audience

This document is intended for: Patients, general users and Health care consultants.

1.3 Conclusion

The Personal Health Tracker using Automatic Report Scanning aims to develop an application that can help people to track their own health trends and become aware about their health condition.This system reduces the need for manual data entry, ensuring more accurate and better user experience for users .

2. System Overview

Personal Health Tracker using Automatic Report Scanning integrates various modules to create a comprehensive environment that supports managing users personal health related information in a friendly way.

2.1. Registration and Authentication System

Facilitates users to login to the system by providing gmail and password. Users can also login using their google account. The system ensures the security of users' credentials.

2.2. Profile Management

Provides detailed information about users account information and medical diagnosis reports. Enabling updating profile information and storing medical diagnosis reports in a secure way.

2.3. Input Processing

Acts as the core of the system , for extracting information from report images accurately.

2.4. Health History Visualization and Searching

Provides easy visualization of the medical diagnosis report history using statistical diagrams (bar chart) and selecting particular reports or time range for getting insights about health conditions.

2.5. Report sharing

Users can share their report with other users. He can control the report access privacy and add or delete users from his report access.

2.6. Notification System

Ensures timely getting notified regarding health issues, platform updates, and personalized notification preferences for an enhanced user experience.

2.7. Health News

This system will also provide the latest health news.

2.8. Stakeholders

1. General User: The stakeholders include **General Users**, who benefit from a centralized platform for managing health records, simplified medical data insights, reminders for medications and appointments, and robust privacy and data security.

2. Healthcare Consultants: **Healthcare Consultants** gain from streamlined sharing of organized patient information, quick access to summarized histories and trends, improved communication with patients, and customizable insights to support efficient decision-making.

3. Elicitation of Personal Health Tracker using Automatic Report Scanning

For the Personal Health Tracker using Automatic Report Scanning project, a comprehensive requirements elicitation process is crucial to ensure the application meets the expectations of its diverse stakeholders.This process involves collaborative requirements gathering, quality function deployment (QFD), and the development of usage scenarios to establish a solid foundation for the application's features and functionalities. The elicitation process for the Personal Health Tracker using Automatic Report Scanning has led to the identification of normal, expected, and exciting requirements, as detailed below:

3.1 Collaborative Requirements Gathering

Stakeholder meetings were conducted with patients, general users, health service providers to understand their needs, challenges, and expectations from the Personal Health Tracker. These discussions facilitated a mutual understanding of the problems and potential solutions, allowing for the negotiation of requirements and the specification of preliminary solution elements.

3.2 Quality Function Deployment

QFD was employed to align the application's development with the specific needs and wants of its users. This approach ensured that Personal Health Tracker using Automatic Report Scanning's functionalities directly responded to the stakeholders' expectations, enhancing user satisfaction and engagement.

3.2.1. Normal Requirements (Basic / Performance Needs)

These are the requirements that are essential for the project. If these are not met, users will be dissatisfied. However, meeting these requirements doesn’t create extra satisfaction—they are considered standard.

- User Authentication: Secure login and sign-up processes for users, and administrators and manage their accounts securely.

- User Profile: Customizable profiles displaying user personal information, health profile and health diagnosis history.
- Image Upload Capability: Users can upload images of their medical reports.
- Report scanning functionality: The system should automatically extract readable text from the image of the medical report.
- Report Storage and Retrieval: The system must store all the report and the extracted data and allow users to access their previous medical reports.
- Data Security: Basic data encryption and user authentication (e.g., password protection, medical history data).

3.2.2. Expected Requirements (Performance Features)

These are requirements that users explicitly ask for. They improve user satisfaction directly in proportion to how well they are implemented.

- User-Friendly Interface: The app should have a clean, intuitive UI for uploading, viewing reports, and managing user accounts.
- High Accuracy in Data Extraction: Users expect the system to extract data accurately from a variety of medical reports, regardless of layout or format.
- Fast Upload and Processing Time: Users expect the system to process and extract the report's text quickly without significant delays.
- Report Organization: Users expect the system to be able to organize their medical reports chronologically or by category (e.g., blood tests).
- Automatic Report Categorization: The system automatically classifies reports into different categories based on content (e.g., diagnostics type).
- Report Search: Users can search for a report of a specific test or specific date.
- History Visualization: Visualize diagnosis test history using a bar chart.
- Users will be notified to update their health and get warnings about their health condition.
- Sharing of Medical Reports: Users can share their report or health profile with other users. They can control the access permission of the report.

3.2.3. Exciting Requirements (Delighters)

These are features that users may not expect but would be delighted by. They can significantly enhance user satisfaction and differentiate your product from competitors.

- Health Insights or Trends: Offer users insights based on the extracted data, such as tracking health trends, health news, providing reminders based on personal medical history.
- Report Summarization: Instead of just extracting raw data, the system could summarize key points or insights from the medical report.
- Notify connected users: When a user adds a new report this system will automatically give notification to the connected users.
- Easily find other users: Users can search and get connected with other users through scanning QR code. Every user will have a unique QR code for his health profile and diagnostic reports.
- Offline Upload: Users can upload images even when offline, and the app syncs the data once an internet connection is available.
- Cross-Platform Availability: Access to the system from both mobile apps and a web interface is expected.

3.3 Usage Scenario (Module)

3.3.1. Registration and Authentication System

The system will have an authentication page. The page will have two options. sign in for new users and login for users that already have an account.

User Sign In

A new user will begin registration by visiting the app's sign in page by clicking on the Register Now button. The user will have to provide a valid gmail, password ,username, First name, Last name, age, date of birth, blood group etc.The user can choose whether to upload a profile photo of him/her. The system validates username uniqueness and gmail format compliance. Upon successful validation, an gmail verification link will be sent to the provided gmail address. Users can confirm their gmail by clicking the link and are redirected to the login page.There will be another option for validation through phone number as well.The system will send verification code to the phone number. Through this the users can confirm their verification.

User Login

Already registered users can login to the system by providing their username or gmail and password on the login page. There will be another option for the user to save their login details.The system will be able to maintain user sessions securely.This will allow the users to login to the system without needing to log in repeatedly. Also users can directly login using their gmail account that logged in on the device.

Login with Google account

Users can also create an account and login to the system using the Google account that is logged in on that device.

3.3.2. Profile Management

The logged in users will be able to manage their account details.The system will provide three options such as, Edit Profile, View Profile and Search User.

Edit Profile :

This option will allow the users to change their username, email, password, profile picture. For changing the username, gmail and password the user will have to provide the current username or gmail and password to validate. The system will then match the provided credentials with the database. If the credentials are verified then the changes will be applied to the users profile.

View Profile :

This option will allow the user to see their profile details and their uploaded medical reports. Each report will have a name, type, date-of-publish, report test name, test value. The user can search a particular medical report by date or report type.The system will display key data like test results, diagnosis and treatment details, linked to specific dates and health conditions.

Search User :

This option will allow the user to search another user by their username or gmail.By using this option the user can see other users medical reports as viewers which they gave access to. The system will generate a QR code for each user to be searched easily. Also users can share a report to another user through QR code sharing.

3.3.3. Input Processing

Image Capture:

The system will provide two options such as from camera and from gallery to the user to upload images of medical diagnosis reports.

Camera Option:

This option will allow the user to take an image of the medical diagnosis report by mobile phone camera. After taking an image the user can choose to take another image if they consider the image to be inappropriate.

Gallery Option:

This option will allow the user to select an image from the gallery.The system will provide the users to crop and rotate the image if they want to.

Manual Input:

Users can create a custom type of medical diagnosis report if they are unable to find any suitable report type. In this case, users will be able to create the diagnosis name, date of publish and test parameters.

Text Extraction:

After capturing the image, the system will send the image to a Text Extraction API. The API will preprocess the image to find the best possible text. After preprocessing the image the API will convert the texts into digital data by using LLM models. After performing text extraction the texts will be returned as a classified text to the system.Then the system will perform post processing such as checking the errors in the texts and save the texts to the user specific database with Report name,type,date of publish.Then the system will show the texts on the app's UI as texts like the medical report.

3.3.4 Report Management

After collecting the text data from the medical diagnosis reports,the system will store the data and fetch the data for showing details of the reports to the users.The system will provide options to the user to see analytics about their diagnosis reports.

Report Analysis:

This option will let the users choose a diagnosis report that they have already uploaded.When the user will select a report by name, the system will analyze the report based on the standard parameters and give a summary about the reports.During the selection of the report,the user can also specify a requirement as a prompt to get specific details about the report.The system will also warn about the potential health risks according to the report.The system will have a chatbot functionality that will provide the user to ask any question or filter the analysis about the selected report.

History Management:

This option will provide the users to select any diagnosis report to get a visual representation such as a bar chart of the parameters of the reports.This option will also allow the users to see a health trend based on previously uploaded medical diagnosis reports.This option will allow the users to search for a particular parameter from both the chart and the report.

3.3.5. Health profile and Report sharing

User search

Users can search by his username. If he has the permission to view the profile then he will get access to that user's profile. Besides each user will have a unique QR code, by using this qr code they can search the profile. Users can control the profile privacy. They can add new users or delete some users from the access permission list.

Report search

Users can also view a specific report by just entering the report id or scanning QR code of that report.

3.3.6. Notification System

The system will send verification code as a notification during registration and account recovery.The system will allow the users to consult with a health consultant if any uploaded diagnosis report has any dangerous measure.The system will also provide the users with daily health related news.The system will notify the users to get a routine health check up. User will get notification when his connected user adds a new report.

Users can customize their notification preferences to suit their needs and priorities.In their settings or preferences section, users access the Customize Notification Preferences feature. They select the types of notifications they want, such as health trend reminders or health related updates. Users also choose the frequency of notifications, like immediate updates or daily digests. This customization ensures users receive relevant information according to their preferences, enhancing their experience on the platform.

3.3.7. Health News

This system will provide the latest health news to users.

4. Use case diagrams

Use Case diagrams give the non-technical view of the overall system.

Level - 0

Use Case Name: Personal Health Tracker using Automatic Report Scanning (PHTAR)

Primary Actors: User , LLM API

Secondary Actors: Email, Summary API , News API

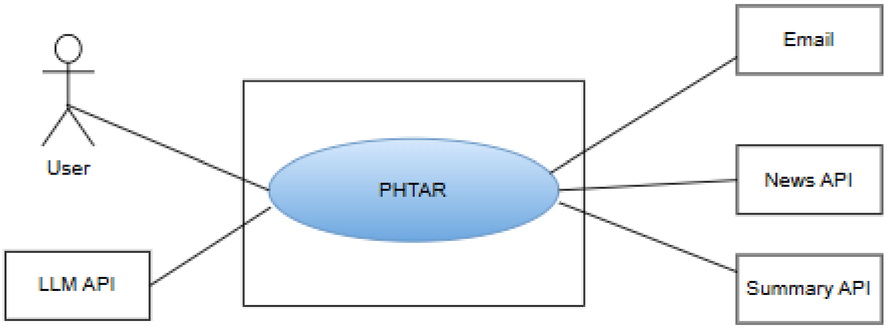


Figure 1: Use case diagram level 0: Personal Health Tracker using Automatic Report Scanning

Level - 1

Use Case Name: Personal Health Tracker using Automatic Report Scanning (PHTAR) (Detailed)
Primary Actors: User, LLM API
Secondary Actors: Email, Summary API , News API

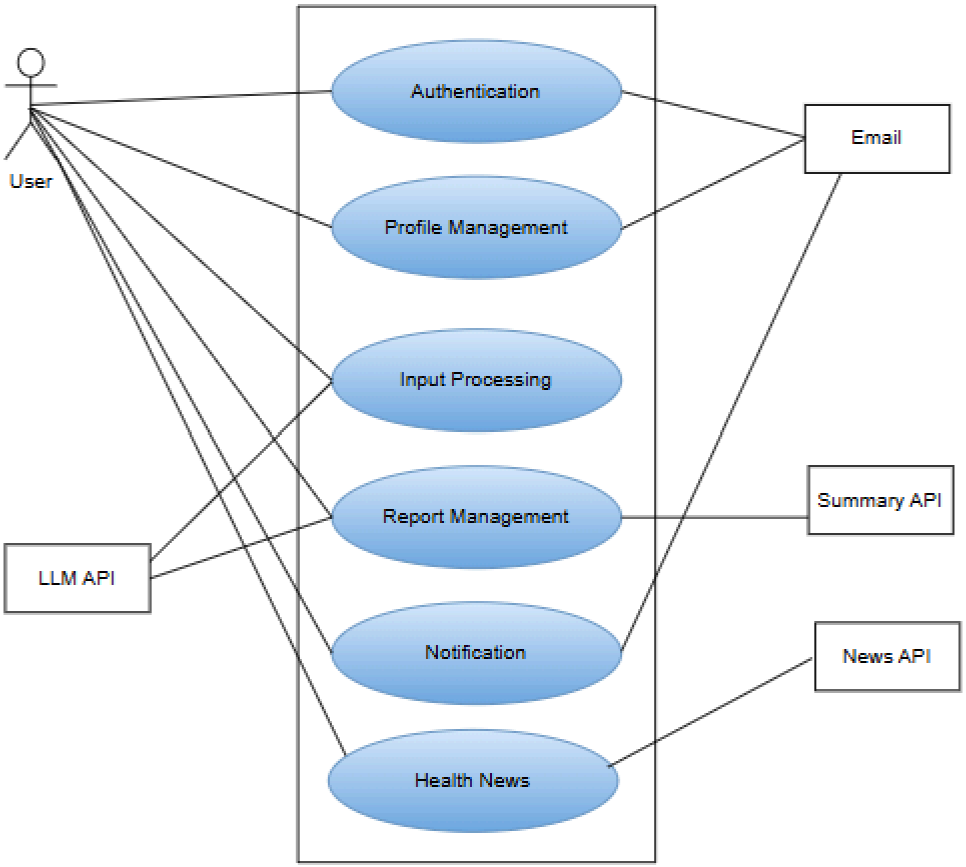


Figure 2: Use case diagram level 1: Personal Health Tracker using Automatic Report Scanning (PHTAR) (Detailed)

Description of Use Case Diagram level 1:

1. Registration & Authentication System: New users register by providing username, gmail, and password. The system checks username uniqueness and gmail format. After validation, a verification link is sent to gmail. Users confirm via the link and proceed to login. Registered users access their account dashboard by entering username/gmail and password. The system verifies credentials for login.

2. Profile Management: The logged in users will be able to manage their account details.The system will provide three options such as, Edit Profile , View Profile and Search User.Edit Profile will allow the users to change their username , gmail , password , profile picture.For changing the username , gmail and password the user will have to provide the current username or gmail and password to validate.View Profile will allow the user to see their profile details and their uploaded medical reports.The user can search a particular medical report by date or report type.Search User will allow the user to search another user by their username or gmail. Users can search another user or a report through QR code sharing.

3. Input Processing: The system will provide two options such as from camera and from gallery to the user to upload images of medical diagnosis reports.User will select any option and choose an image.The system will receive the image and show extracted text on the UI.

4. Report Management: After collecting the text data from the medical diagnosis reports,the system will store the data and fetch the data for showing details of the reports to the users.The system will provide options to the user to see analytics about their diagnosis reports.

5. Notification System: The system will send verification code as a notification during registration and account recovery.The system will allow the users to consult with a health consultant if any uploaded diagnosis report has any dangerous measure.The system will also provide the users with daily health related news feed.The system will notify the users to get a routine health check up.

Users can customize their notification preferences to suit their needs and priorities.In their settings or preferences section, users access the Customize Notification Preferences feature. They select the types of notifications they want, such as health trend reminders or health related updates. Users also choose the frequency of notifications, like immediate updates or daily digests. This customization ensures users receive relevant information according to their preferences, enhancing their experience on the platform.

Level - 1.1

Use Case Name: Registration and Authentication System
Primary Actors: User
Secondary Actors: Email

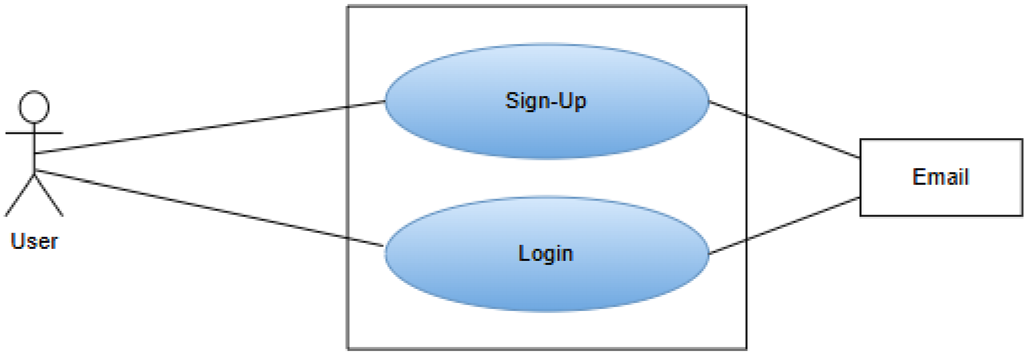


Figure 3: Use case diagram level 1.1: Registration and Authentication System

Action - Reply:

Action: New users initiate the registration process by selecting the Sign Up option.

Reply: System stores response.

Action: The system prompts for providing valid gmail and password by user.

Reply: System check credentials validation and check for email verification.

Reply: Upon successful validation, a verification link is sent to the provided email address, enabling users to confirm their email by clicking the link and being redirected to the login page.

Action: Registered users access the login page and enter their email address along with the password.

Reply: The system verifies the entered information against stored data and grants access upon successful validation directing users to the HomePage.

Action: User selects login with gmail account that is logged in on that device.

Reply: The system logged in with that google account.

Level - 1.2

Use Case Name: Profile Management
Primary Actors: User
Secondary Actors: Email

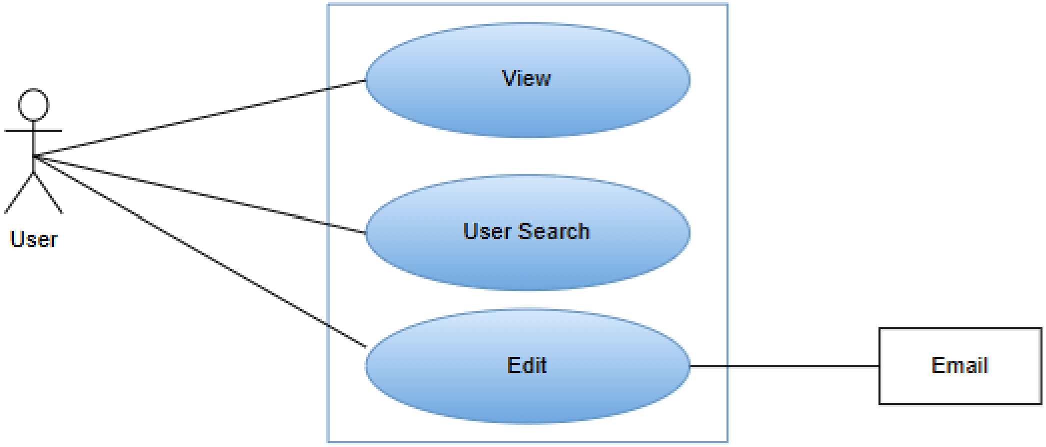


Figure 4: Use case diagram level 1.2: Profile Management

Action - Reply:

Action: Users select the profile option

Reply: System displays the profile page along with user's details.

Action: User Select Edit Profile option

Reply: System redirects user to Edit Profile Page.

Action: User changes profile information and provide password for validation

Reply: System checks for the validation and updates the profile on successful validation.

Action: User select User Search option from Homepage with another user's username.

Reply: System checks for access permission for viewing that users' profile.

Reply: If the user has access permission the system will show profile details of that searched user along with access provided medical diagnosis reports.

Action: User chooses QR code show option from profile.

Reply: System generates a QR code for the user's profile.

Action: User scans a QR code to search a user.

Reply: System shows the searched user's profile.

Level - 1.3

Use Case Name: Input Processing
Primary Actors: User
Secondary Actors: LLM API

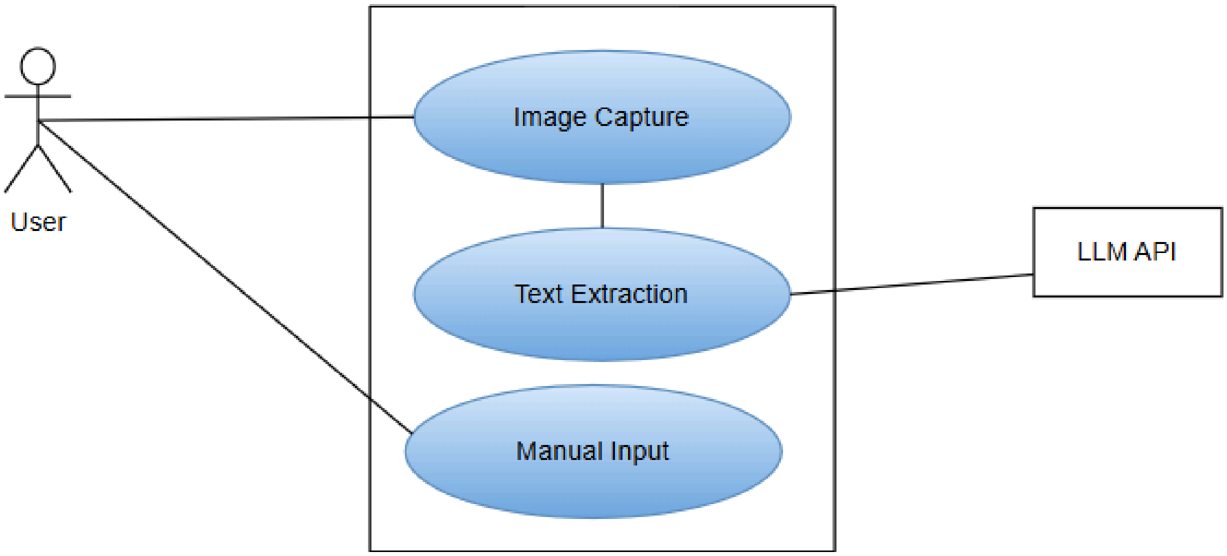


Figure 5: Use case diagram level 1.3: Input Processing

Action-Replay:

Action: Users select Image upload option

Reply: System redirects user to Image upload Page.

Action: User selects Gallery or Camera option.

Reply: System redirects user to camera or gallery based on the provided option.

Action: User selects or captures image.

Reply: System redirects user to image cropping screen.

Action: User can crop, rotate, resize the image.

Reply: System performs text extraction on that image.

Reply: System redirects to Input screen to show Extracted Text of that report.

Action: User can update the input value.

Action: User selects Manual Input option.

Reply: System redirects user to Manual Input Page.

Action: User input the test name and test value.

Reply: System stores provided Input along with the report image.

Level - 1.4

Use Case Name: Report Management
Primary Actors: User , LLM API
Secondary Actors: Summary API

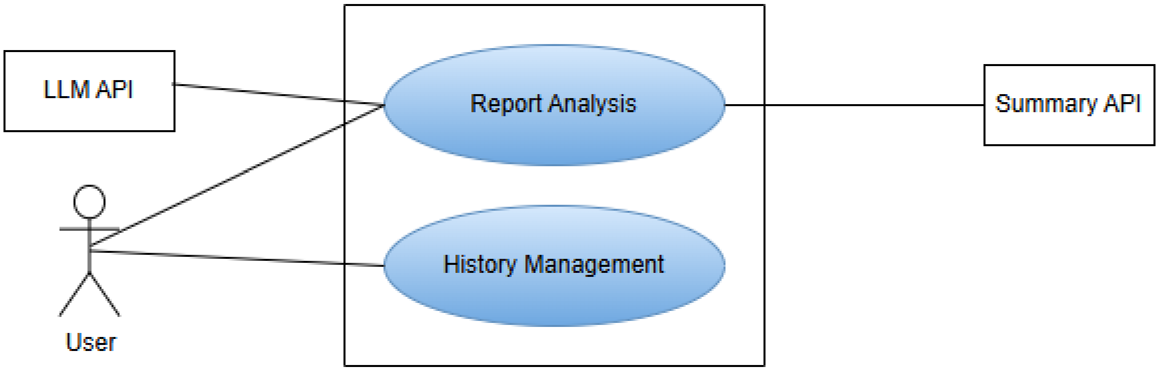


Figure 6: Use case diagram level 1.4: Health Report Management and Visualization

Action-Replay:

Action: User selects History Page.

Reply: System redirects user to History Page and shows all medical diagnosis reports of the user.

Action: User selects medical diagnosis report based on a criteria (Date, Report type, Test name)

Reply: System receives the criteria and shows that particular report.

Action: User selects Report Summary for the provided report

Reply: System sends the report to the Summary API.

Action: The Summary API sends a summary of the report to the system.

Reply: System receives the summary of the report and shows that to the user.

Action: User selects report history visualization for a specific test type.

Reply: System shows the statistical visualization (bar chart) of the test values of that test.

Action: User selects Search Report option with report details.

Reply: System receives report details and checks for availability of the report. By getting availability of the report system shows the report.

Action: User selects Share Report option with report details and accessed user details.

Reply: System updates report access privilege to the accessed user.

Action: User chooses a QR code show option for a particular Report .

Reply: System generates a QR code for that report.

Action: User scans a QR code to search a report.

Reply: System shows the searched report.

Level - 1.4.1

Use Case Name: Report Analysis
Primary Actors: LLM API, Summary API
Secondary Actors: User

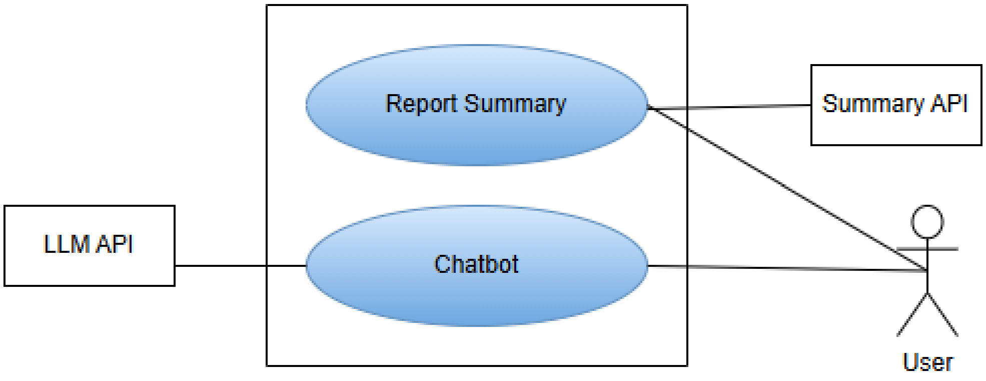


Figure 7: Use case diagram level 1.4.1: Report Analysis

Action-Replay:

Action: User selects Report Summary option with report details and a prompt about the report.

Reply: System receives the criteria and fetches report details from database.

Action: System sends report and prompt to Summary API.

Reply: The Summary API receives the report and sends a summary and prompts details of the report to the system.

Action: System receives the summary and prompt reply of the report from the Summary API.

Reply: System shows the summary and prompt reply of the report to the user.

Action: User gives a prompt for chatbot to the system .

Reply: System receives prompt and sends that to Chatbot.

Action: Chatbot generates answers to prompts.

Reply: System receives answer from chatbot and sends that to user.

Level - 1.4.2

Use Case Name: History Management
Primary Actors: User

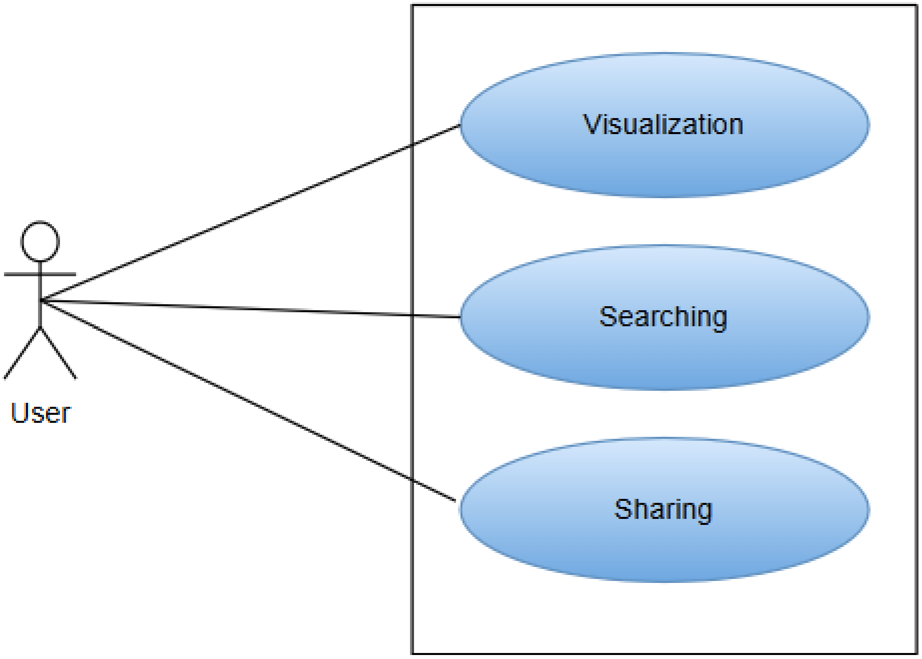


Figure 8: Use case diagram level 1.4.2: History Management

<u>Action-Replay:</u>

Action: User selects Report Visualization for a specific test (e.g.: RBC)

Reply: System shows the statistical visualization (bar chart) of the test value of that test.

Action: User selects Search Report option with report details.

Reply: System receives report details and checks for availability of the report. By getting availability of the report system shows the report.

Action: User selects Share Report option with report details and accessed user details.

Reply: System updates report access privilege to the accessed user.

Action: User add or delete use from the access permission list.

Reply: System update the access permission list and store it.

Level - 1.5

Use Case Name: Notification
Primary Actors: User , Email

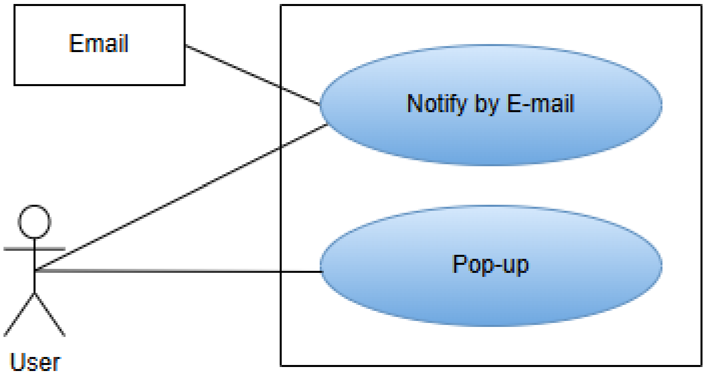


Figure 9: Use case diagram level 1.5: Notification

Action - Reply:

Action: User provides Email

Reply: System valides Email and sends a link to Email.

Action: User clicks on the link on Email.

Reply: System redirects user to Homepage.

Action: System gives pop up notification about health trend to user.

Reply: User clicks on the pop up and gets redirected to Health Tread Page.

Action: User added a new report.

Reply: The connected users will be notified that this user has uploaded a new report.

Action: Added a new user in his access permission list.

Reply: The newly added user will be notified.

Activity Diagram

Activity diagram ID: 01

Level 1.1

Name: Registration and Authentication System

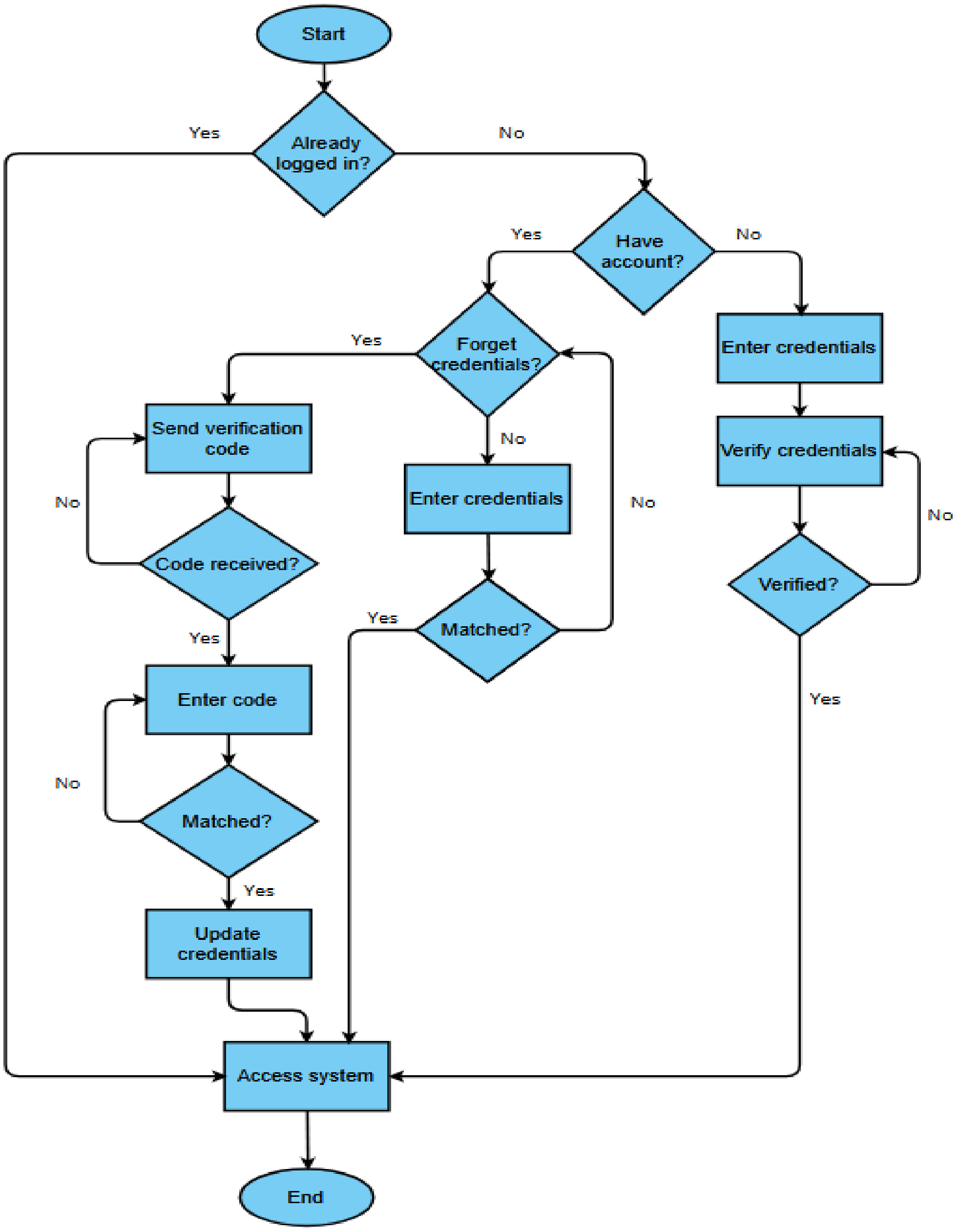


Figure 10: Activity diagram level 1.1: Registration and Authentication System

Activity diagram ID: 02

Level 1.2

Name: Profile Management

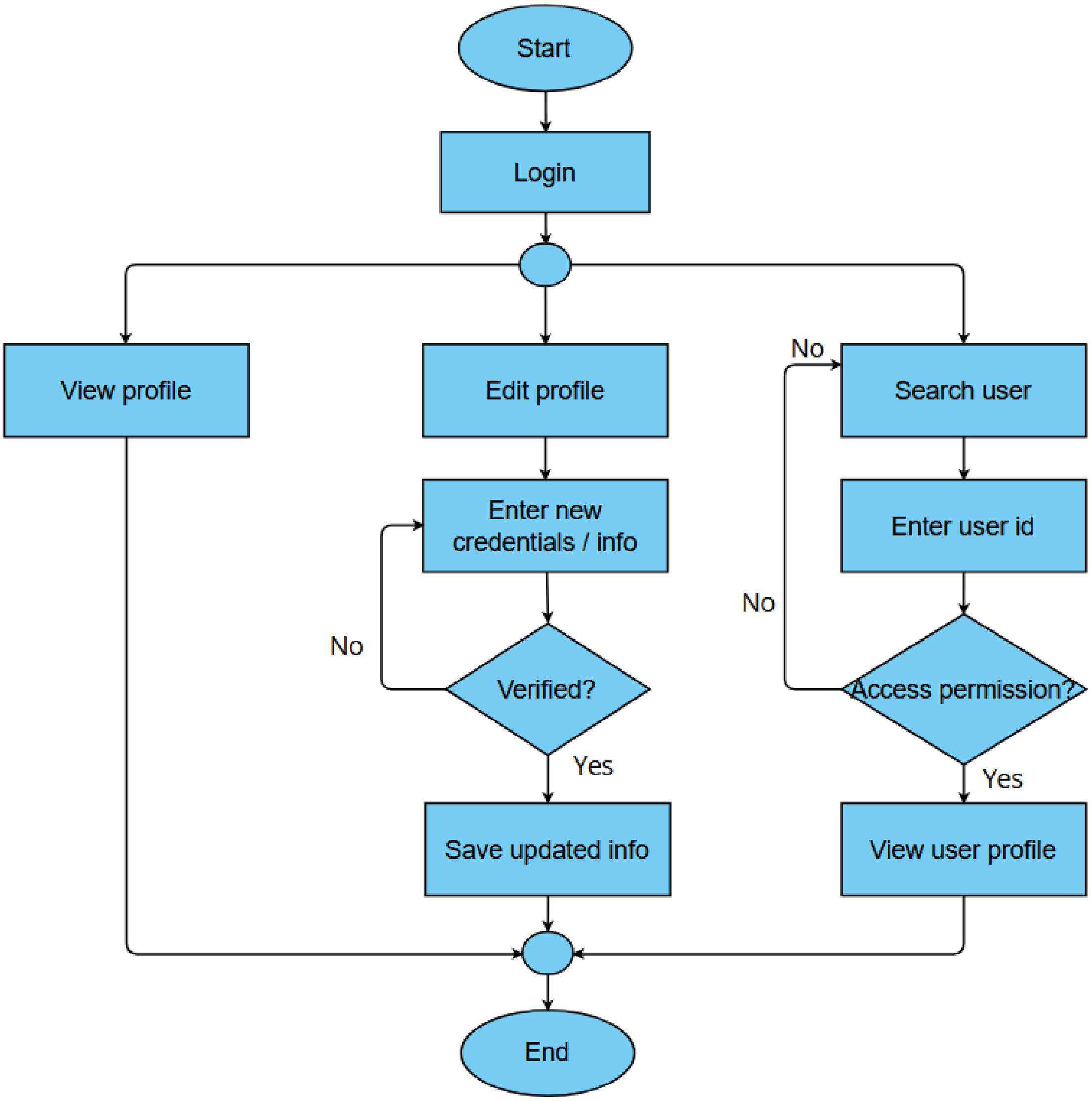


Figure 11: Activity diagram level 1.2: Profile Management

Activity diagram ID: 03

Level 1.3.1

Name: Image capture

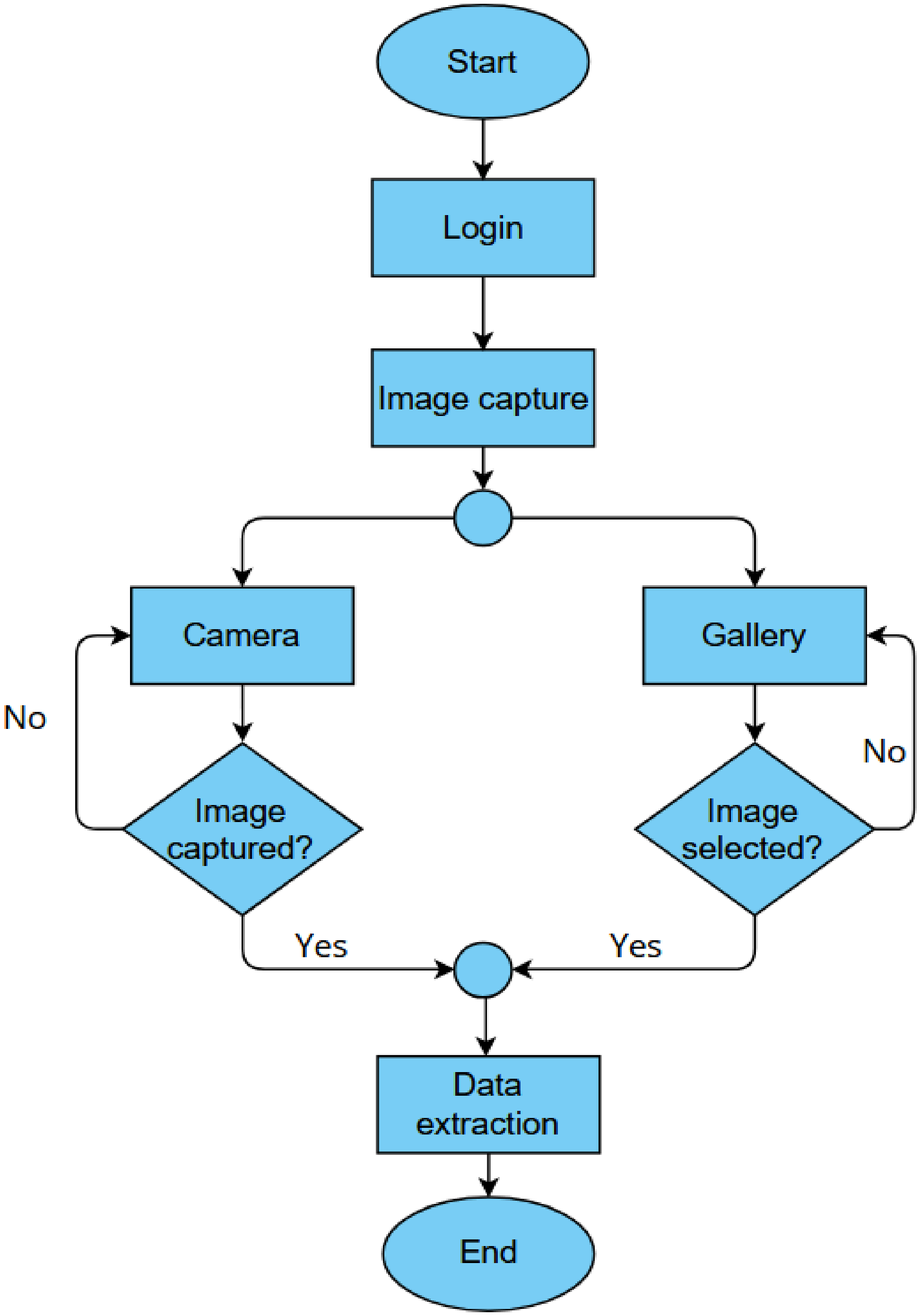


Figure 12: Activity diagram level 1.3.1: Image capture

Activity diagram ID: 04
Level 1.3.2
Name: Data extraction

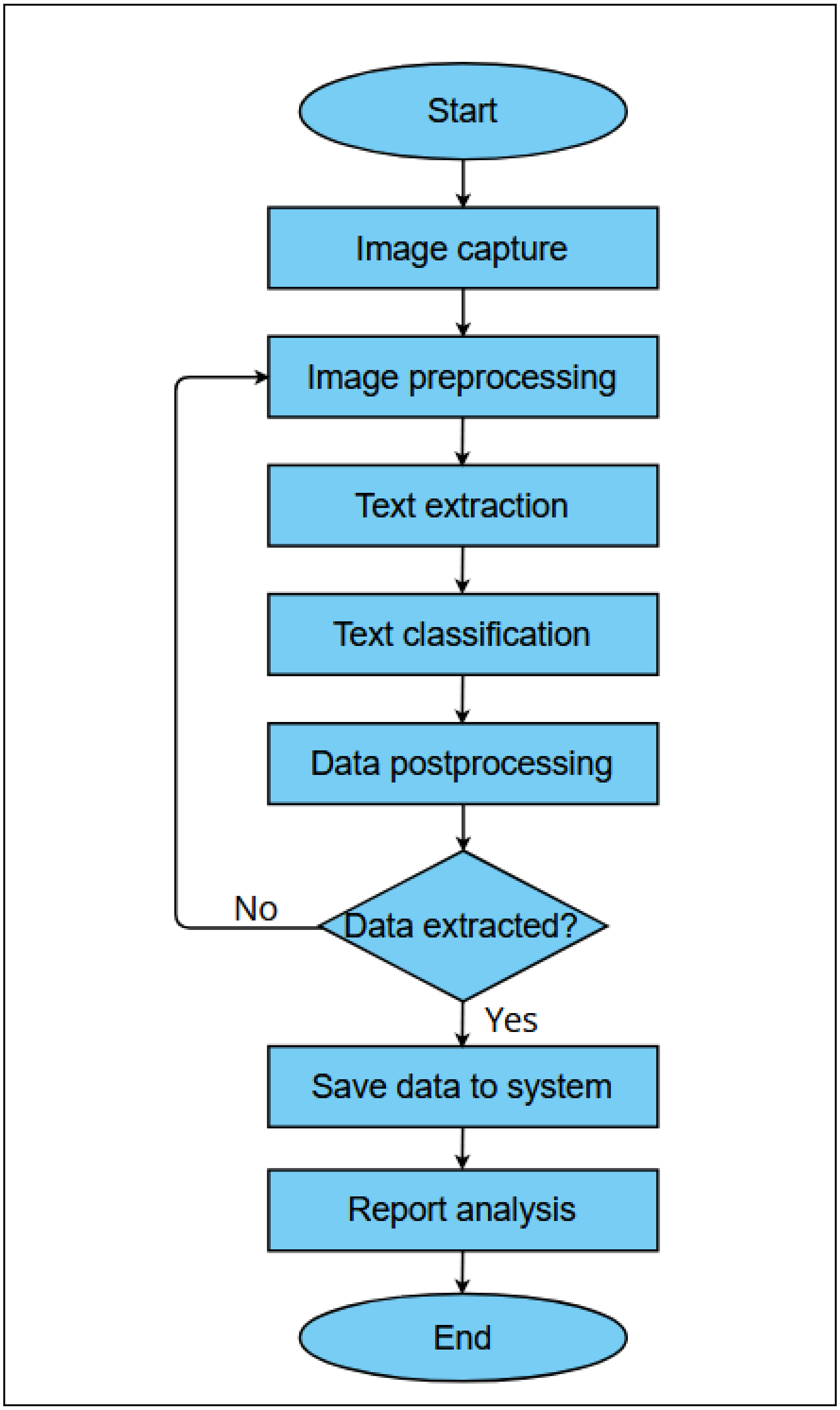


Figure 13: Activity diagram level 1.3.2: Data extraction

Activity diagram ID: 05
Level 1.4.1
Name: Report analysis

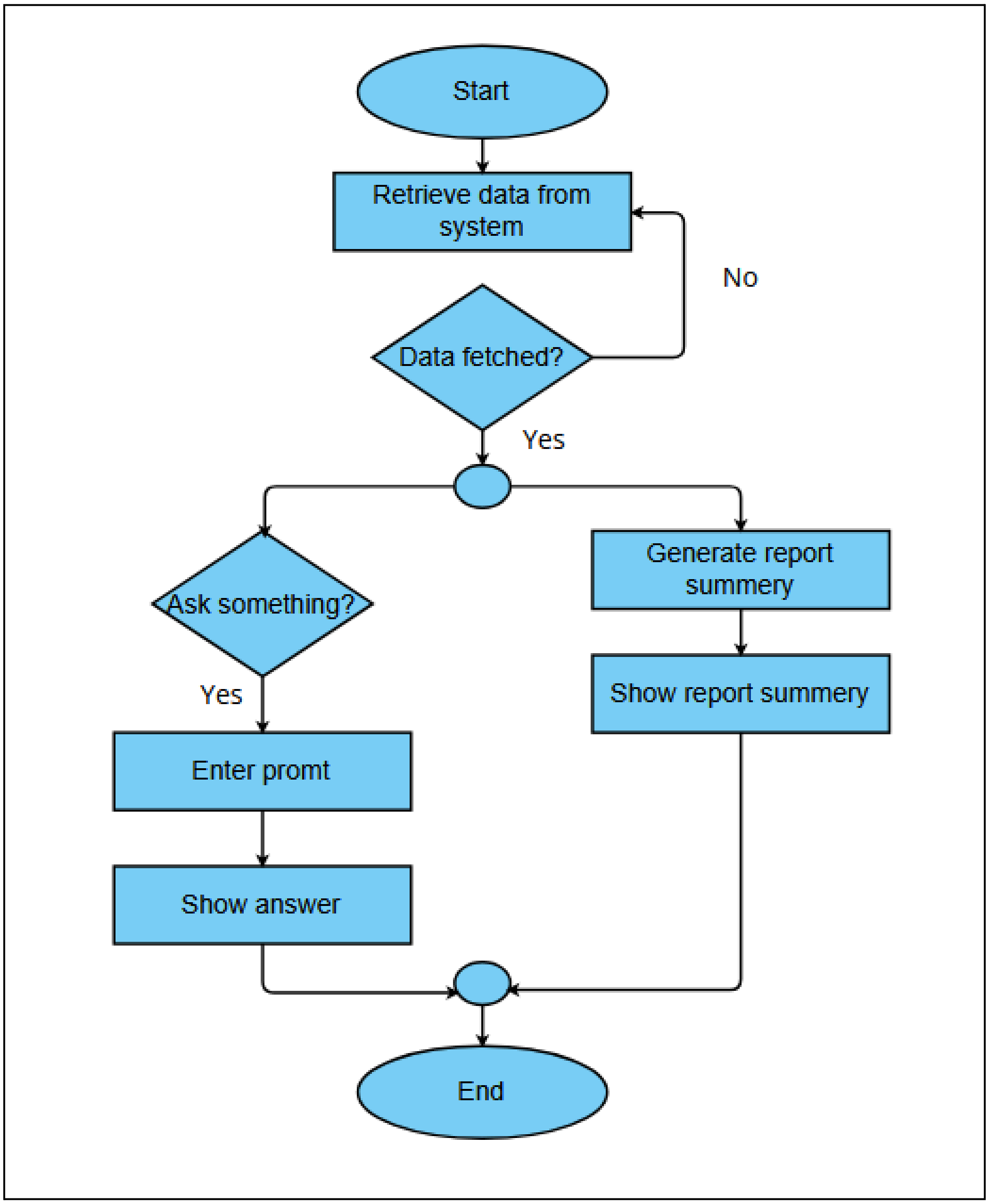


Figure 14: Activity diagram level 1.4.1: Report analysis

Activity diagram ID: 06

Level 1.4.2

Name: History management

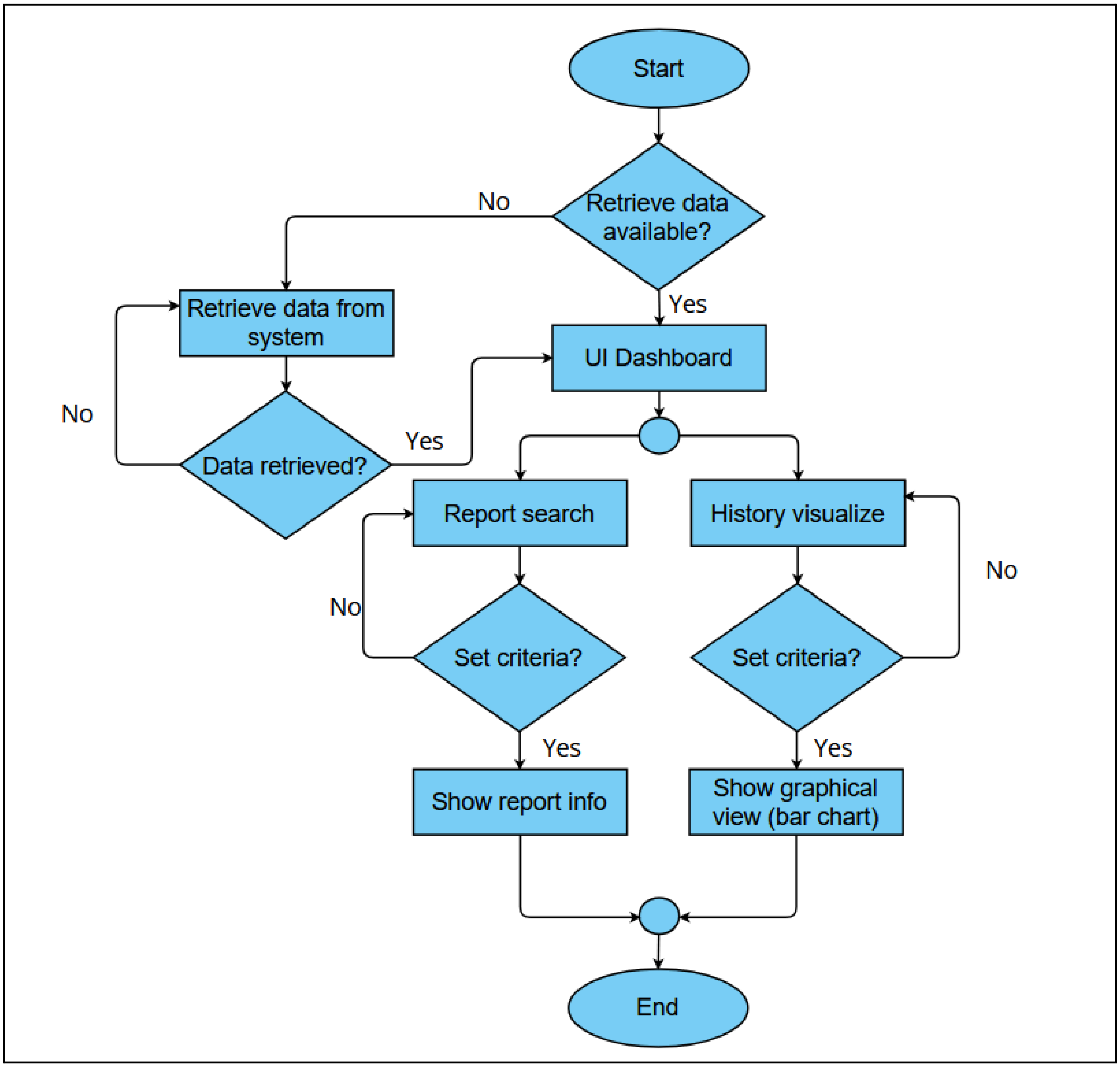


Figure 15: Activity diagram level 1.4.2: History management

Activity diagram ID: 07

Level 1.5

Name: Notification

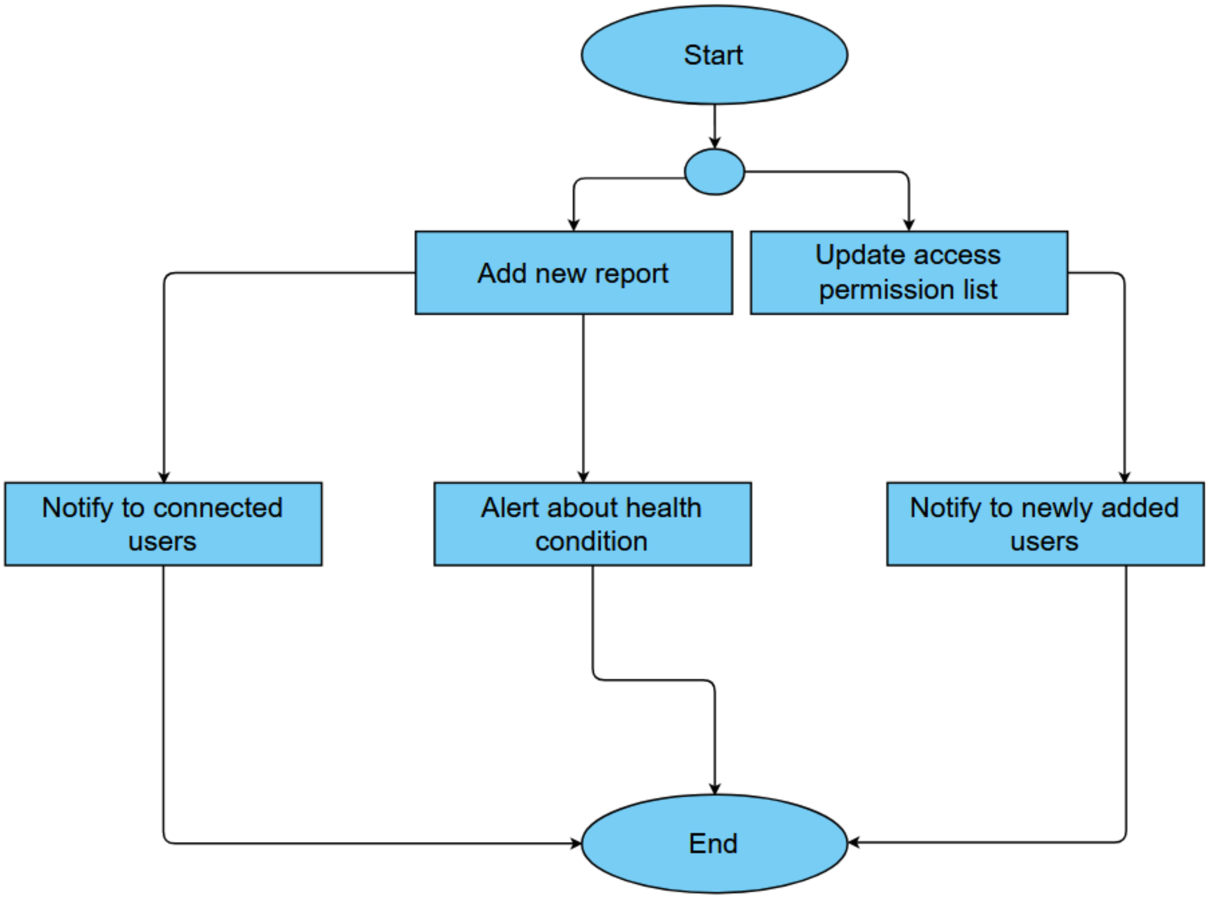

Start
Add new report
Update access permission list
Notify to connected users
Alert about health condition
Notify to newly added users
End

Figure 16: Activity diagram level 1.5: Notification

Swimlane Diagram

Swim lane Diagram ID: 01

Level 1.1

Name: Registration and Authentication System

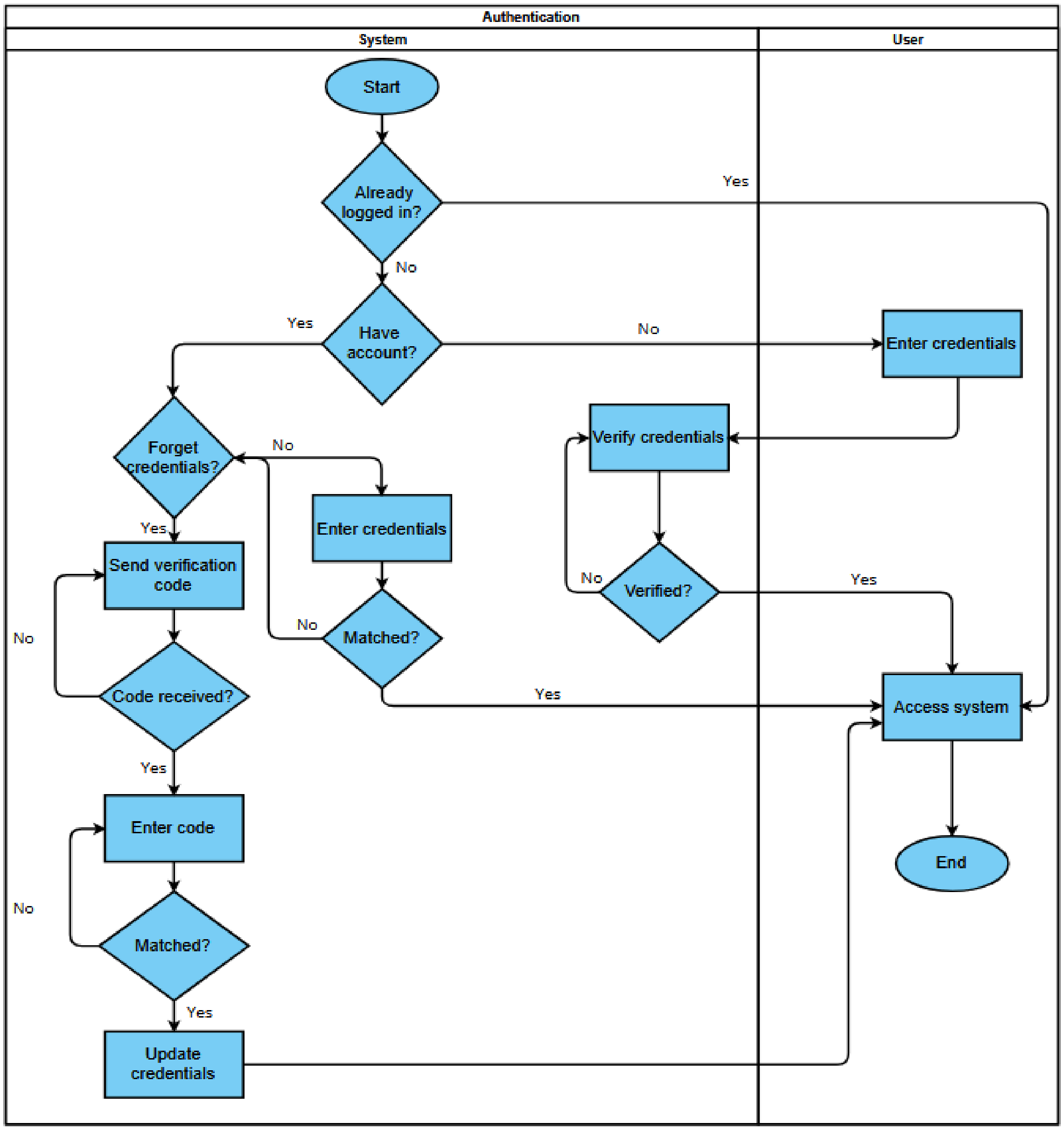


Figure 17: Swim lane diagram level 1.1: Registration and Authentication System

Swim lane Diagram ID: 02

Level: 1.2.1

Name: Profile management

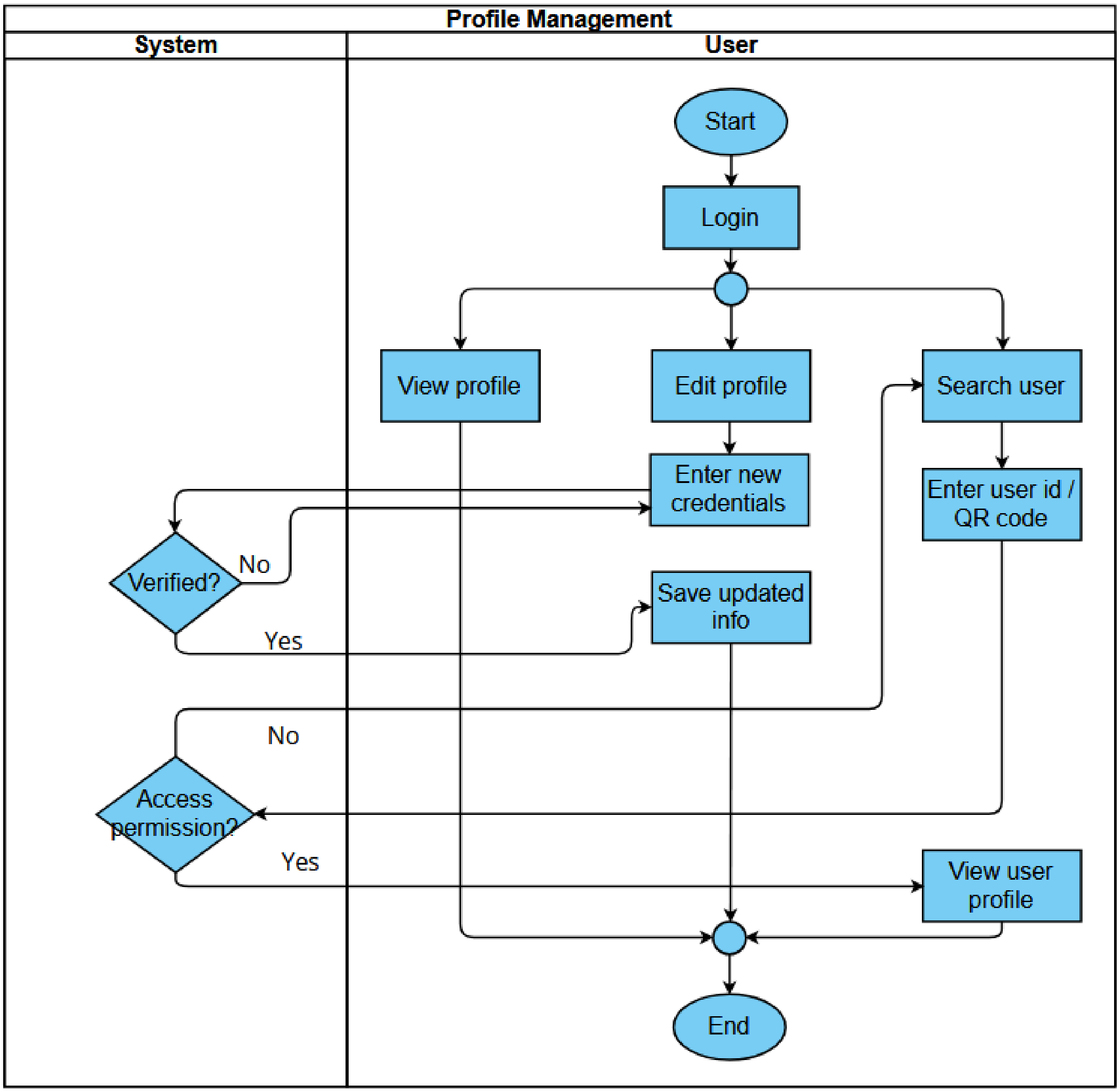


Figure 18: Swim lane diagram level 1.2.1: Profile Management

Swim lane Diagram ID: 03

Level 1.3.1

Name: Image capture

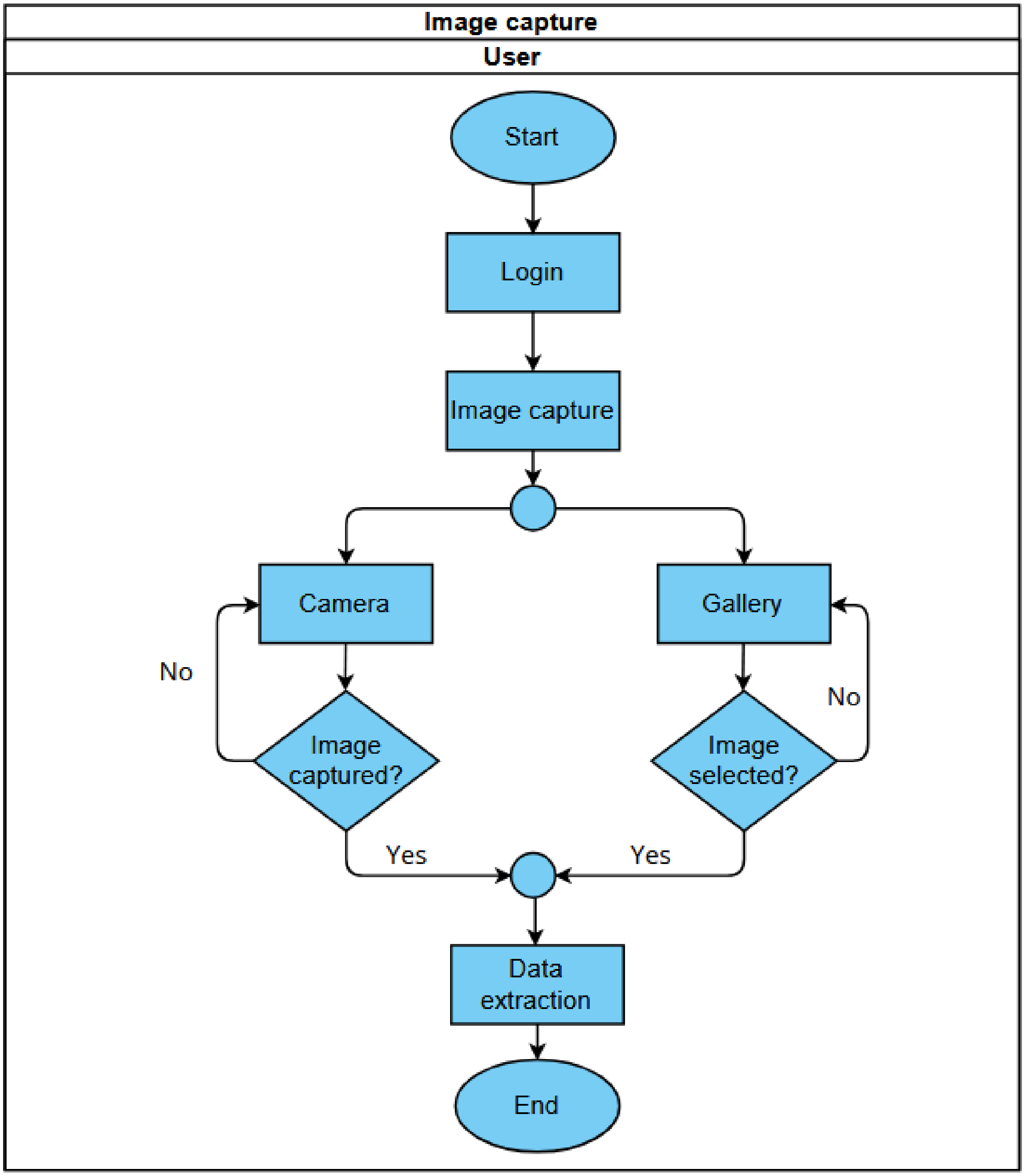


Figure 19: Swim lane diagram level 1.3.1: Image capture

Swim lane Diagram ID: 04

Level 1.3.2

Name: Data extraction

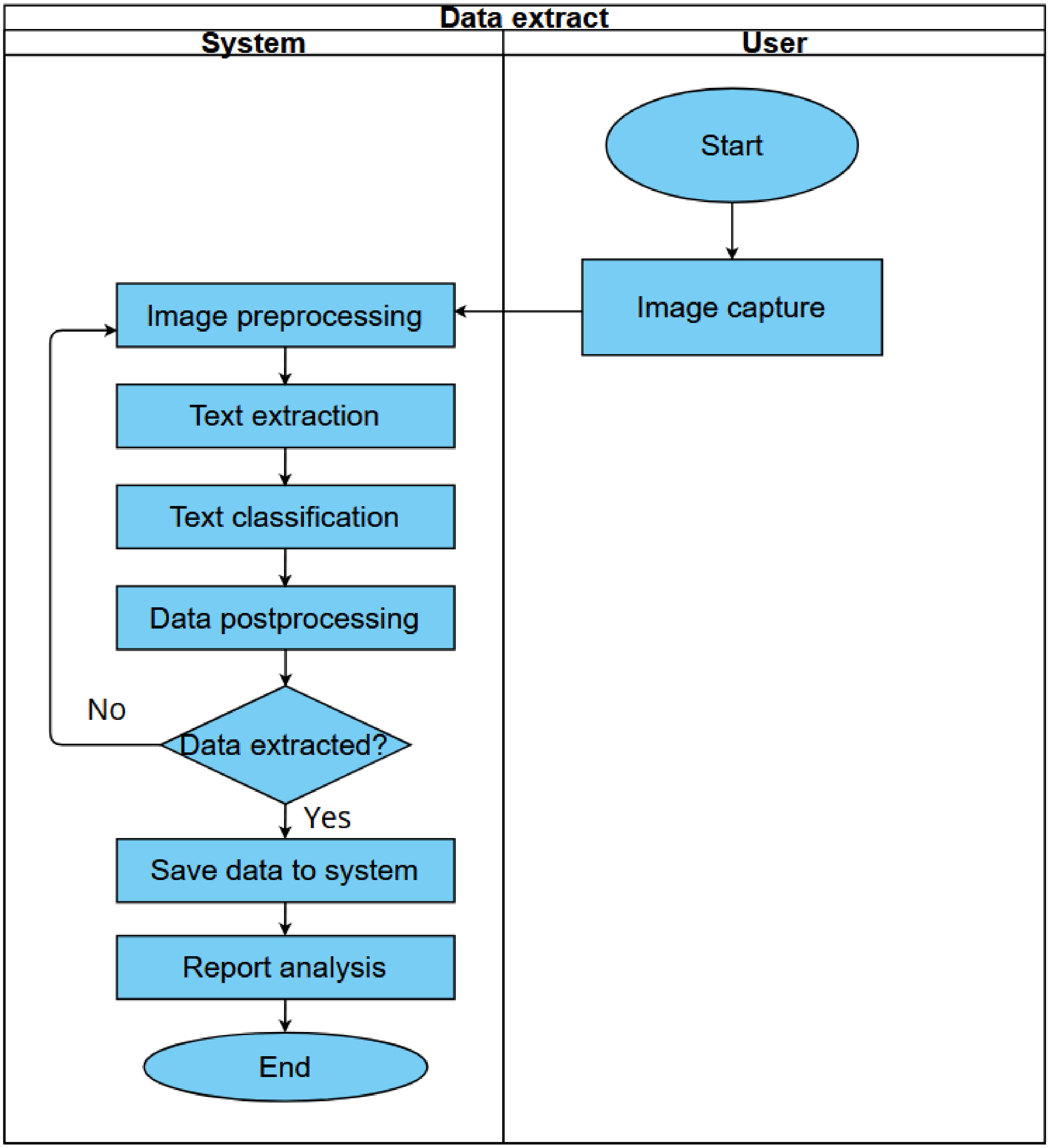


Figure 20: Swim lane diagram level 1.3.2: Data extraction

Swim lane Diagram ID: 05

Level: 1.4.1

Name: Report analysis

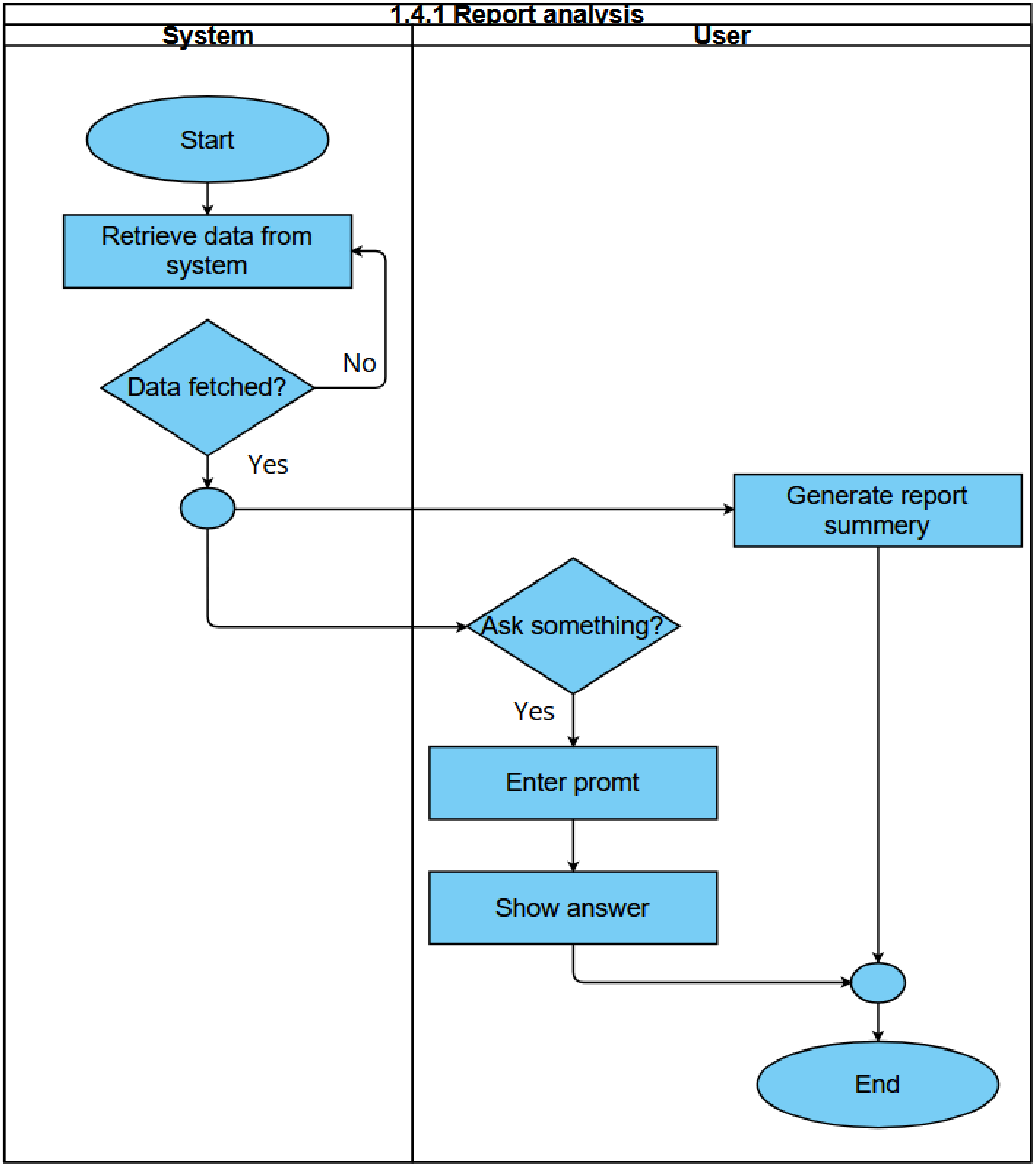


Figure 21: Swim lane diagram level 1.4.2: Report analysis

Swim lane Diagram ID: 06

Level: 1.4.2

Name: History management

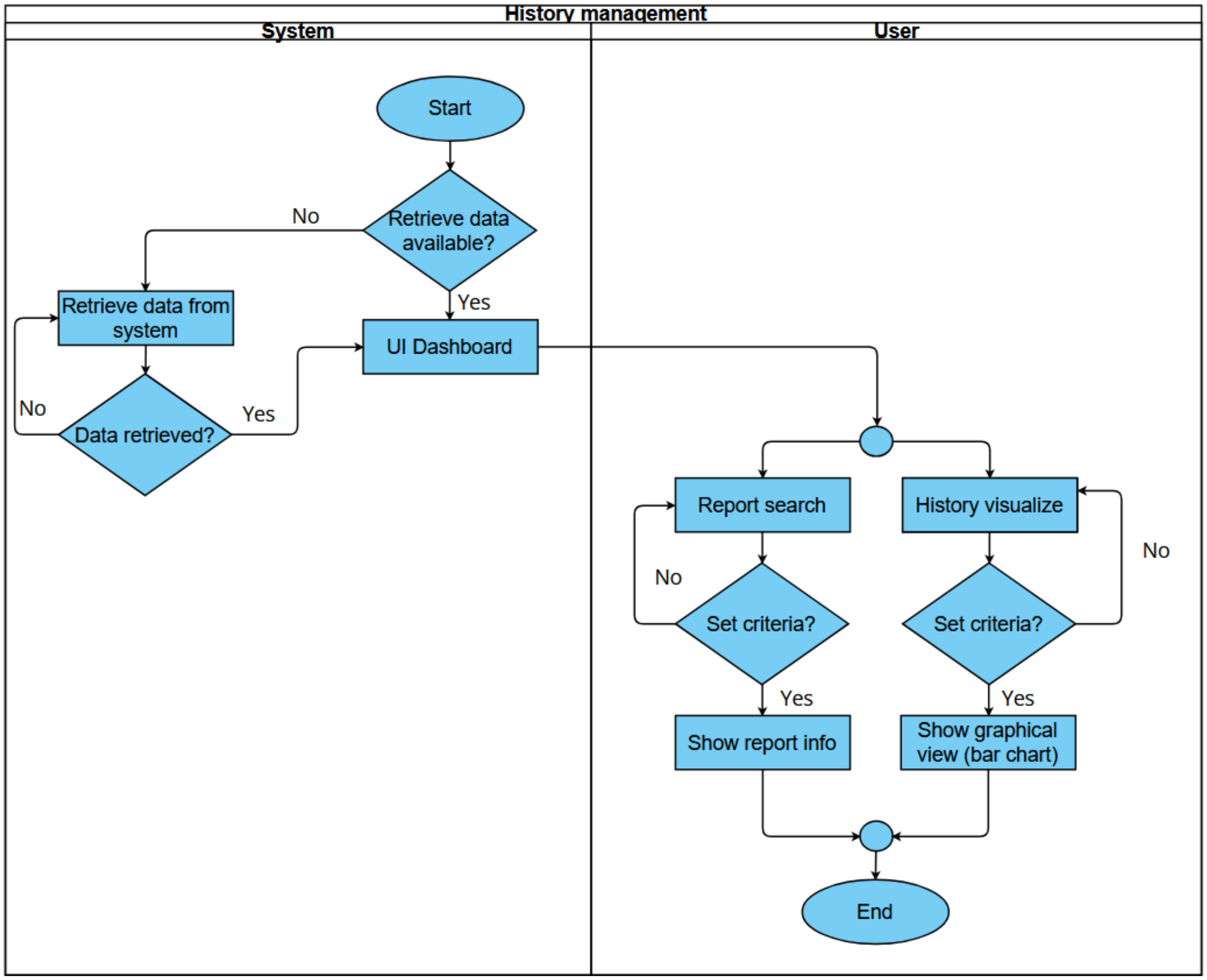


Figure 22: Swim lane diagram level 1.4.2: History management

Swim lane Diagram ID: 07

Level:1.5.1

Name: Notification

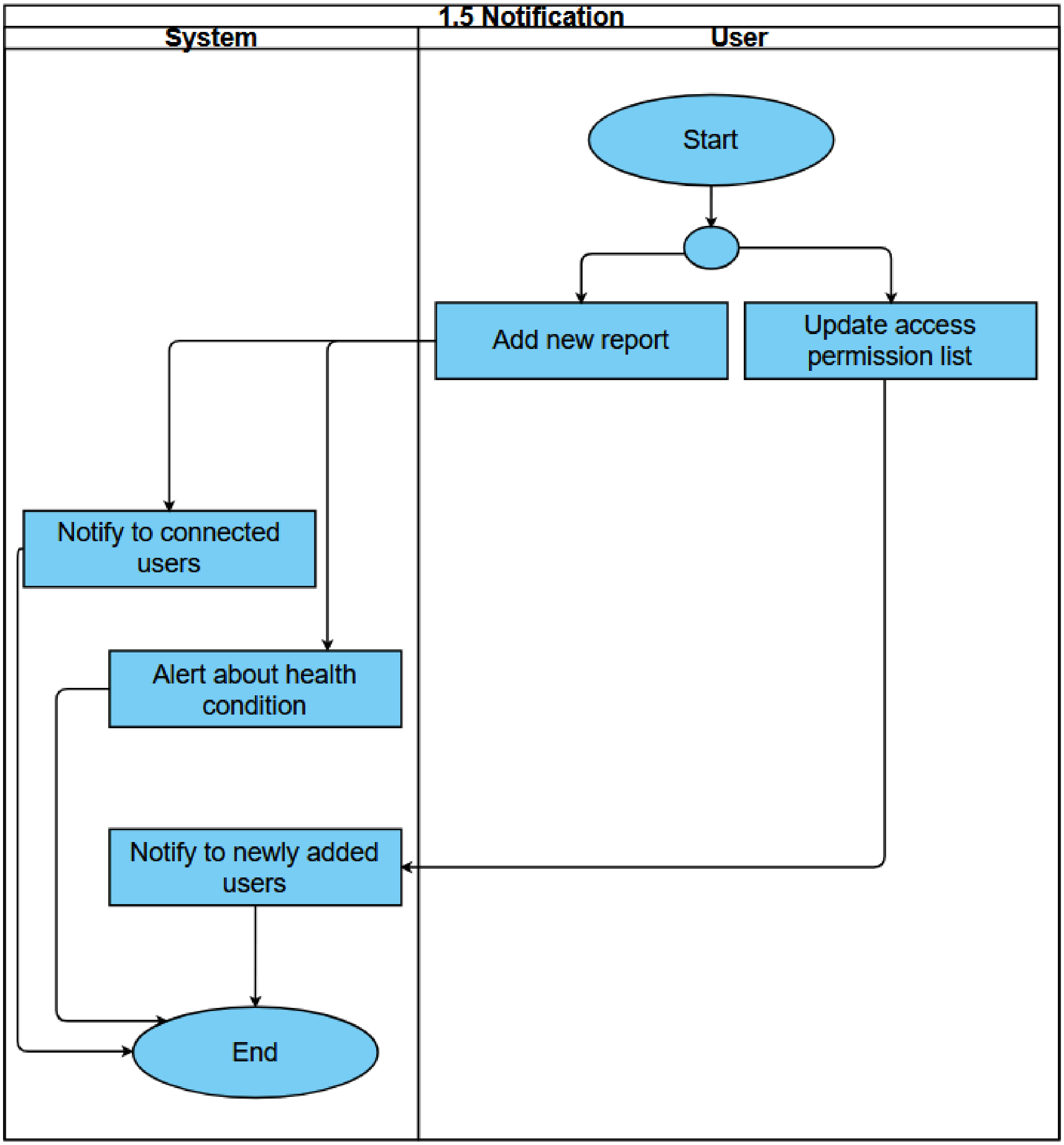

1.5 Notification
System
User
Start
Add new report
Update access permission list
Notify to connected users
Alert about health condition
Notify to newly added users
End

Figure 23: Swim lane diagram level 1.5: Notification

5. Data based modeling

Noun Listing

SL	Noun	Attribute	Problem/Solution Space
1.	Access		P
2.	Account		S
3.	Account recovery		S
4.	Address		S
5.	Age		S
6.	Analysis		P
7.	Analytics	Report name,Visual representation,Bar chart	S
8.	API		P
9.	App		P
10.	Authentication		S
11.	Bar chart		S
12.	Blood group		S
13.	Blood Pressure		S
14.	Button		P
15.	Camera		S
16.	Chatbot		S
17.	Classification		S
18.	Code		P
19.	Compliance		P
20.	Confirmation		P

21.	Consultant		P
22.	Credentials		P
23.	Crop		S
24.	Customize Notification Preferences		S
25.	Daily digest		S
26.	Dashboard		P
27.	Database		S
28.	Date		P
29.	Date of birth		S
30.	Date of publish		S
31.	Details		P
32.	Diagnosis		P
33.	Digital data		P
34.	Edit Profile		S
35.	Email		S
36.	Error checking		P
37.	Experience		P
38.	Filter		P
39.	First name		S
40.	Frequency		P
41.	Feed		P
42.	Gallery		S
43.	Gmail		S
44.	Glucose level		S

45.	Health conditions		S
46.	Health consultant		P
47.	Health data		S
48.	Health risks		S
49.	Hematology		S
50.	Health summary		S
51.	Health trend	Update,Daily digest,Health related news	S
52.	Health-related news		S
53.	History management		P
54.	Image capture		S
55.	Image preprocessing		S
56.	Input processing	Image capture,Camera, Gallery,Crop, LLM API	S
57.	Key data		P
58.	Last name		S
59.	Link		P
60.	Login		S
61.	Measure		P
62.	Medical diagnosis		S
63.	Medical report	Name,Report type,Date of publish,Blood pressure,Glucose level,Test Parameters	S

64.	Mobile device		P
65.	Mobile Phone		P
66.	Name		S
67.	News		P
68.	Notification	Username,Gmail, Customize Notification Preferences, Routine check-up, Reminder	S
69.	LLM API		S
70.	Option		P
71.	Page		P
72.	Parameters		P
73.	Password		S
74.	Phone number		S
75.	Photo		S
76.	Platform		P
77.	Post processing		S
78.	Preferences		P
79.	Profile		S
80.	Profile picture		S
81.	Question		P
82.	Registration		P
83.	Reminder		S
84.	Report		P
85.	Report management		P

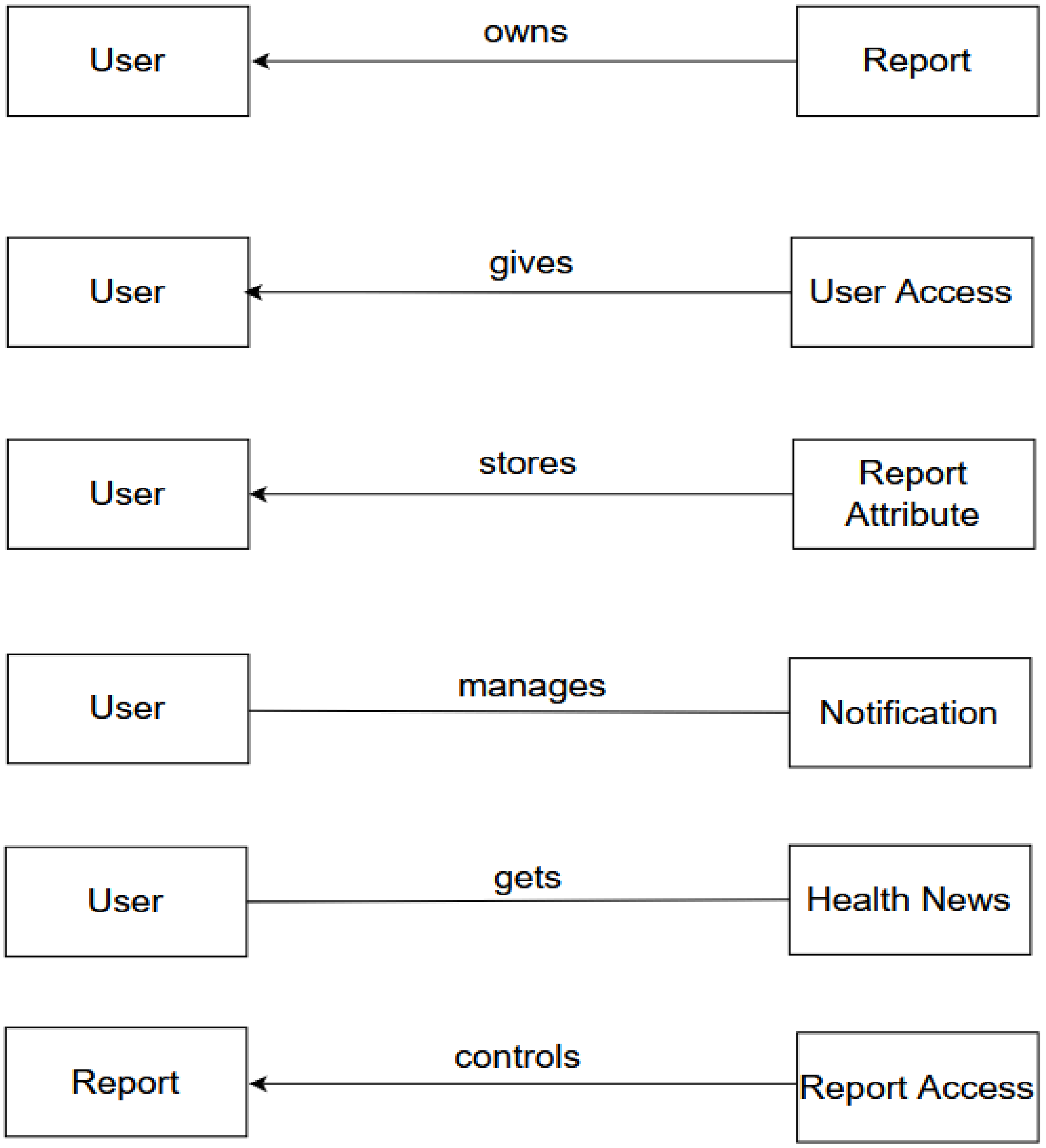
86.	Report type		S
87.	Report Analysis		P
88.	Routine check-up		S
89.	Search User		S
90.	Session		S
91.	Setting		P
92.	Sign-in		S
93.	Standard parameters		S
94.	Summary		S
95.	Summary report	Name,Health Summary,Standard parameters, Summary API	S
96.	System		S
97.	Test result		S
98.	Text		S
99.	Text errors		S
100.	Text Processing		
101.	Treatment details		S
102.	Trend analysis		S
103.	Type		P
104.	UI		S
105.	Uniqueness		P
106.	Update		S
107.	User	Username,First name,Last name, Gmail,	S

		Password,Age, Date of birth , Blood group , Photo	
108.	Username		S
109.	Validation		P
110.	Validation process		P
111.	Verification		P
112.	Verification Link		S
113.	View Profile		S
114.	Visual representation		S
115.	Viewers		P
117.	Custom		P
118.	Manual Input		S
119.	Test Parameters		S
120.	LLM API		S
121.	Login Page		S
122.	Report History		S
123.	QR code		S
124.	Summary API		S

Probable data objects

SI No.	Data Object	Attributes
1.	User	User_id, Username, First name, Last name, email, Password, Date of birth, Blood group, Photo
2.	Report	Report_id, report_image_link, Date, Report_Summary
3.	Notification	User_id, Username, email, Time , Content ,NID
4.	Share Access	User_id,Report_id, viewer_id
5.	Health news	News_id,News_text
6.	Report Attributes	User_id,Report_id,Attribute_name,value ,AID

Relationship between Data Objects



ER Diagram

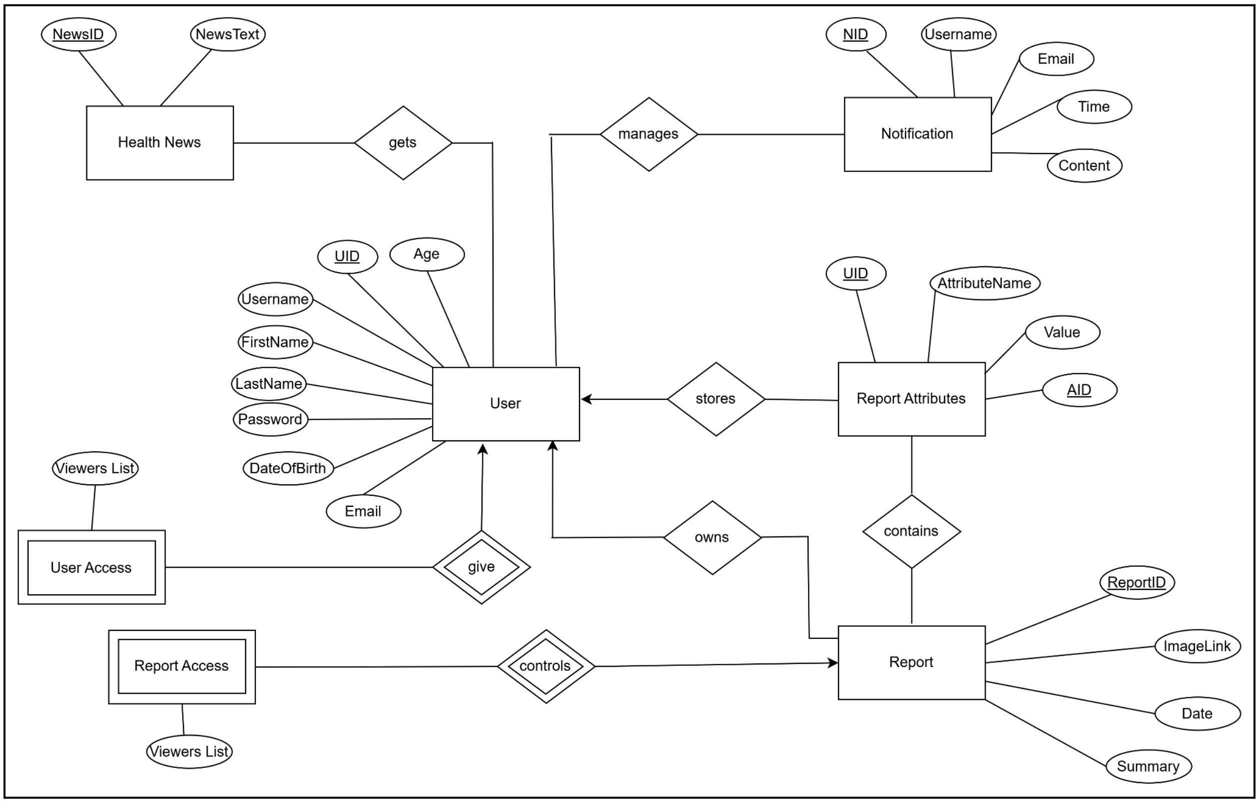


Figure 24: ER Diagram

SCHEMA

User	
Attributes	**Data type**
UID	String
UserName	String
FirstName	String
LastName	String
Age	Number
Password	String
Email	String

Report	
Attributes	**Data types**
ReportID	String
UID	String
ReportType	String
Date	String
Summary	String

Notification	
Attributes	**Data type**
NID	String
Username	String
Email	String
Time	String
Content	String

User Access	
Attributes	**Data type**
UID	String
ViewersList	JSON

Health News	
Attributes	**Data type**
NewsID	String
NewsText	String

Report Access	
Attributes	**Data type**
ReportID	String
ViewersList	JSON

Report Attributes	
Attributes	**Data type**
AID	String
UID	String
AttributeName	String
Value	Number

Manages	
Attributes	**Data type**
NID	String
UID	String

Gets	
Attributes	**Data type**
NewsID	String
UID	String

Contains	
Attributes	**Data type**

ReportID	String
UID	String

6. Class-based Modeling

The objects that the system will manipulate, the operations (also known as methods or services) applied to manipulate these objects, relationships (some hierarchical) between the objects, and interactions among the defined classes—all are represented by class-based modeling. Classes, objects, attributes, operations, Class-Responsibility-Collaborator (CRC) models, collaboration diagrams, and packages constitute components of a class-based model.

6.1 List of verbs

SL	Verbs	SL	Verbs
1.	Have	2.	Save
3.	Sign in	4.	Maintain
5.	Login	6.	Allow
7.	Begin	8.	Change
9.	Visit	10.	See
11.	Click	12.	Search
13.	Provide	14.	Display
15.	Choose	16.	Take
17.	Validate	18.	Consider
19.	Send	20.	Select
21.	Confirm	22.	Crop
23.	Redirect	24.	Straighten
25.	Send	26.	Warn
27.	Preprocess	28.	Ask

29.	Find	30.	Filter
31.	Call	32.	Select
33.	Convert	34.	Get
35.	Return	36.	Allow
37.	Perform	38.	Search
39.	Check	40.	Send
41.	Save	42.	Consult
43.	Show	44.	Notify
45.	Store	46.	Customize
47.	Fetch	48.	Access
49.	See	50.	Select
51.	Choose	52.	Ensure
53.	Select	54.	Enhance
55.	Analyze	56.	Give

6.2 General Classification

1. External entities (e.g., other systems, devices, people) that produce or consume information to be used by a computer-based system.
2. Things (e.g., reports, displays, letters, signals) that are part of the information domain for the problem.
3. Occurrences or events (e.g., a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
4. Roles (e.g., manager, engineer, salesperson) played by people who interact with the system.
5. Organizational units (e.g., division, group, team) that are relevant to an application.
6. Places (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the system.
7. Structures (e.g., sensors, four-wheeled vehicles, or computers) that define a class of objects or related classes of objects.

SL	Nouns	G.C
1.	System	2,7
2.	User	1,4,7
3.	Sign-in	2,3,7
4.	Login	2,3,7
5.	Gmail	2
6.	Password	2
7.	Username	2
8.	First name	2
9.	Last name	2
10.	Age	2
11.	Date of birth	2
12.	Blood group	2
13.	Photo	2
14.	Verification Link	2
15.	Address	2
16.	Phone number	2
17.	Session	2
18.	Profile	2,5,7
19.	Account	2
20.	Edit Profile	2
21.	View Profile	2
22.	Search User	2
23.	Database	1,2,5,7
24.	Name	2

25.	Date of publish	2
26.	Test result	2
27.	Treatment details	2
28.	Health conditions	2
29.	Input processing	2,3,7
30.	Image capture	1,3,7
31.	Camera	1,2
32.	Gallery	1,2
33.	Crop	1,2
34.	Summary API	1,3,7
35.	Text	2
36.	LLM	1,3,7
37.	Post processing	2
38.	UI	2
39.	Medical report	1,2,7
40.	Analytics	2
41.	Summary	2
42.	Health risks	2
43.	Chatbot	1,7
44.	Visual representation	2
45.	Bar chart	2
46.	Health trend	2
47.	Notification	1,2,3,7
48.	Account recovery	1,2,7
49.	Routine check-up	2

50.	Customize Notification Preferences	2
51.	Reminder	2
52.	Update	2
53.	Email	2
54.	Profile picture	2
55.	Image preprocessing	2
56.	Classification	2
57.	Text errors	2
58.	Medical diagnosis	2
59.	Summary report	2
60.	Analysis	2,3
61.	Health summary	2,3
62.	Trend analysis	2,3
63.	Daily digest	2
64.	Health data	2
65.	Report type	2
66.	Standard parameters	2
67.	Health-related news	1,2,7
71	Text Extraction API	1,2,7
72.	Login Page	2
73.	Authentication	1,2,7
74.	Report History	2,3,7

6.3 Potential Class List

1. User
2. Sign-in
3. Authentication
4. Login
5. Account Recovery
6. Profile
7. Database
8. Input processing
9. Image capture
10. Report analysis
11. Medical report
12. Report History
13. Notification
14. Health-related news
15. Text Extraction API

6.4 Selection Criteria

1. Retained information: The potential class will be useful during analysis only if information about it must be remembered so that the system can function.
2. Needed services: The potential class must have a set of identifiable operations that can change the value of its attributes in some way.
3. Multiple attributes: During requirement analysis, the focus should be on "major" information; a class with a single attribute may, in fact, be useful during design, but is probably better represented as an attribute of another class during the analysis activity.
4. Common attributes: A set of attributes can be defined for the potential class and these attributes apply to all instances of the class.
5. Common operations: A set of operations can be defined for the potential class and these operations apply to all instances of the class.
6. Essential requirements: External entities that appear in the problem space and produce or consume information essential to the operation of any solution for the system will almost always be defined as classes in the requirements model.

SL	Nouns	Selection Criteria
1.	User	1,2,3,4,5,6
2.	Sign-in	1,2
3.	Login	1,2
4.	Profile	1,2
5.	Database	1,2,3,6
6.	Input processing	1,2,5,6
7.	Image capture	1,2,6
8.	Report Analysis	1,2,6
9.	LLM	1,2
10.	Medical Report	1,2,3,4,5,6
11.	Notification	1,2,7
12.	Health related news	1,3,7
13.	Text Extraction API	1,2,7
14.	Account Recovery	1,2,3,7
15.	Authentication	1,2,3,7
16.	Report History	1,2,3,7

6.5 Selected Classes

1. User
2. Database
3. Input processing
4. Image capture
5. Report analysis
6. Medical Report
7. Notification
8. Health related news
9. Text Extraction API
10. Account recovery

11.Authentication
12.Report History

6.6 Finally selected classes

1. User
2. Authentication
3. Account recovery
4. Gemini API
5. Database
6. Report analysis
7. Medical report
8. Report history
9. Notification
10. Health News

6.7 Class Cards

Class-1: User

Class : User	
Attribute	**Methods**
-firstName -lastName -userName -password -date_of_birth -bloodGroup -age	change_user_info(),
Responsibilities	**Collaborator Class**
Manage User Registration	Database , Authentication
Manage User Login	Database , Authentication
Password Recovery	Database , Account Recovery
General accessibility and Notification	Database , Notification

Class-2: Authentication

Class : Authentication	
Attribute	**Methods**
- verification link - userName - password	get_user_info(), validate_user_info(), get_verification_link(), send_verification_link(), login(), register(), isAllowed()
Responsibilities	**Collaborator class**
User info verification	User , Database
Send verification link	Database

Class-3: Account recovery

Class : Account recovery	
Attribute	**Methods**
- verification link	recover_password(), recover_email(), update_user_info()
Responsibilities	**Colaborator class**
User info verification	User , Database
Send verification link	Database
Update user Info	Database

Class-4: GeminI API

Class : Input processing	
Attribute	**Methods**
- image	select_image_capture_mode(), choose_image(), get_extracted_text(),
Responsibilities	**Colaborator class**
Extract Text	
Store extracted text	Database

Class-5: Database

Class : Database	
Attribute	**Methods**
-user list -report list -report attribute list	create(), delete(), update(), get()
Responsibilities	**Collaborator class**
Store information	User , Authentication , Account recovery, Medical report , Report analysis , Report history , Notification , Health news
Maintain stable connection	
Provide required information	

Class-6: Medical report

Class : Medical report	
Attribute	**Methods**
- report id - report type - date and time - report summary	create_report(), update_report(), provide_report()
Responsibilities	**Collaborator class**
Provide report details	Database , User , Report history

Class-7: Report analysis

Class : Report analysis	
Attribute	**Methods**
- report bar chart	provide_visualization()
Responsibilities	**Collaborator class**
Show visualization	Report History

Class-8: Report history

Class : Report history	
Attribute	**Methods**
- report list	provide_report_details(), show_report_details(), share_report()
Responsibilities	**Collaborator class**
Show report history	User , Medical Report

Class-9: Notification

Class : Notification	
Attribute	**Methods**
- notification_list - user_list	send_general_notification(), receive_general_notification(), receive_user_info(), send_alert(), send_verification_email(),
Responsibilities	**Collaborator class**
Send notification	User
User information reception	Database

Class-10: Health news

Class : Health news	
Attribute	**Methods**
- news Link - news	get_daily_news(), send_news_to_user(),
Responsibilities	**Collaborator class**
Send News	User

CRC Diagram

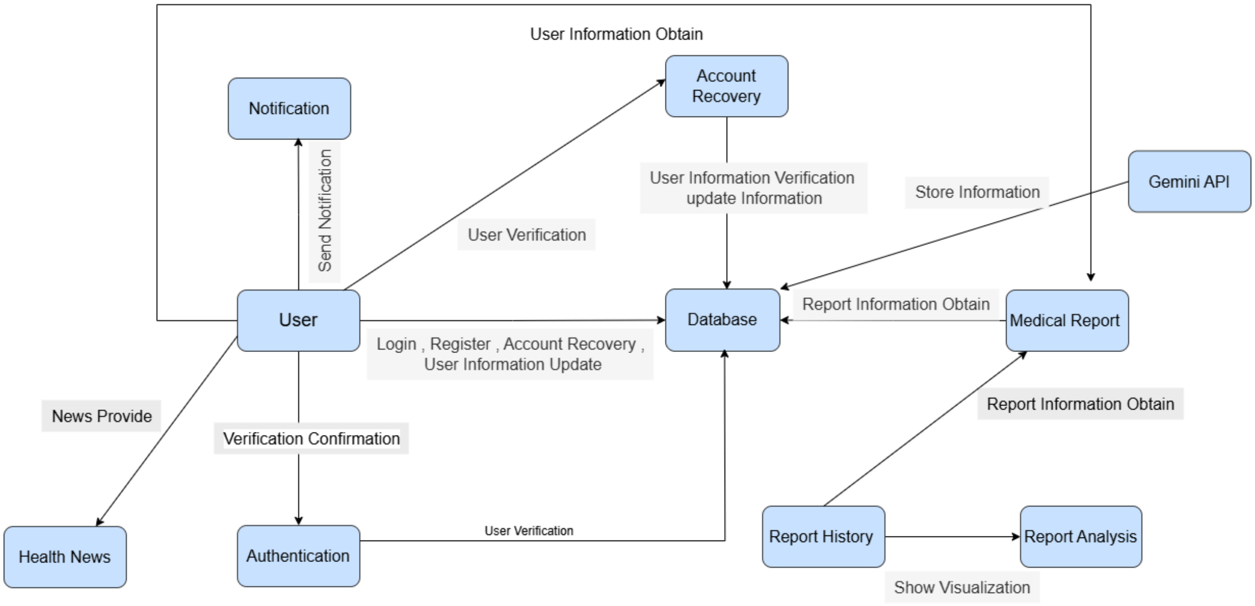


Figure 25: CRC Diagram

7. Behavioral Modeling

The Behavior Modeling indicates how the system will behave to external events or stimuli. It is represented as a function of time and event, It describes interactions between objects. It shows how individual objects collaborate to achieve the behavior of the system as a whole. In UML behavior of a system is shown with the help of use case diagram, sequence diagram and activity diagram. To create behavioral model following things can be considered-

- Evaluation of all use-cases to fully understand the sequence of interaction within the system.
- Identification of events that drive the interaction sequence and understand how these events relate to specific classes.
- Creating sequence for each use case.
- Building a state diagram for the system.
- Reviewing the behavioral model to verify accuracy and consistency

7.1 Event Identification

In the event identification table, events are mentioned in the leftmost column. The initiator class of the event and collaborator classes are mentioned in the following two columns and special cases for each event are mentioned in the rightmost column, if there are any such special cases

Event Table

SL	Event Name	Initiator	Collaborator	Associated Method
1.	Login	User	Database , Authentication	login(), get(), validate_user_info()
2.	Register	User	Database , Authentication	register(), create() , get_user_info()
3.	Password Recovery	User	Database , Account Recovery	recover_password() update(),
4.	Account Information Update	User	Database, Account Recovery	update_user_info(), create(), get() , delete(), update()
5.	User Information Validate	Authentication	User , Database	validate_user_info(),get()
6.	Verification Link Receive	User	Authentication	get_verification_link()
7.	Email Recovery	User	Account Recovery , Database	recover_email(), update()
8.	Image Capture Mode Selection	User	Input Processing	select_image_capture_mode()
9.	Image Selection	User	Input Processing	choose_image()
10.	Extracted Text Receive	User	Input Processing	get_extracted_text()
11.	Manual Input Provide	User	Database , Input Processing	create_manual_report(), create()
12.	Save User Information	Database	User ,	validate_user_info(),c

			Authentication	reate() , update()
13.	Update User Information	Database	User , Authentication	validate_user_info(),update(), delete()
14.	Save Report Information	Database	Medical Report, Authentication	validate_user_info(),create_report()
15.	Update Report Information	Database	Medical Report, Authentication	validate_user_info(),update_report()
16.	User Search	User	Database , Authentication	get_user_info(), isAllowed() , get()
17.	Report Creation	User	Medical Report	create_report()
18.	Create Visualization	Report Analysis	Report History , User	provide_visualization(), provide_report_details()
19.	Notify User	Notification	Database	send_general_notification(), get()
20.	Provide News	Health News	User	send_news()
21	Verification Link sent	Authentication	User	send_verification_link()
22	Provide Report Details	User	Report History	provide_report_details()
23	Show Report Details	Report History	Database , User	show_report_details()
24	Show Report Visualization	Report Analysis	Database , Report History , User	provide_visualization()
25	Report Share	User	Report History , User , Authentication	share_report()
26	Show User Information	Database	User	get()
27	Report Search	User	Authentication ,	show_report_details()

			Database, Report History	
28	Receive News	Health News	User	send_news_to_user()

7.2 State Transition Diagram

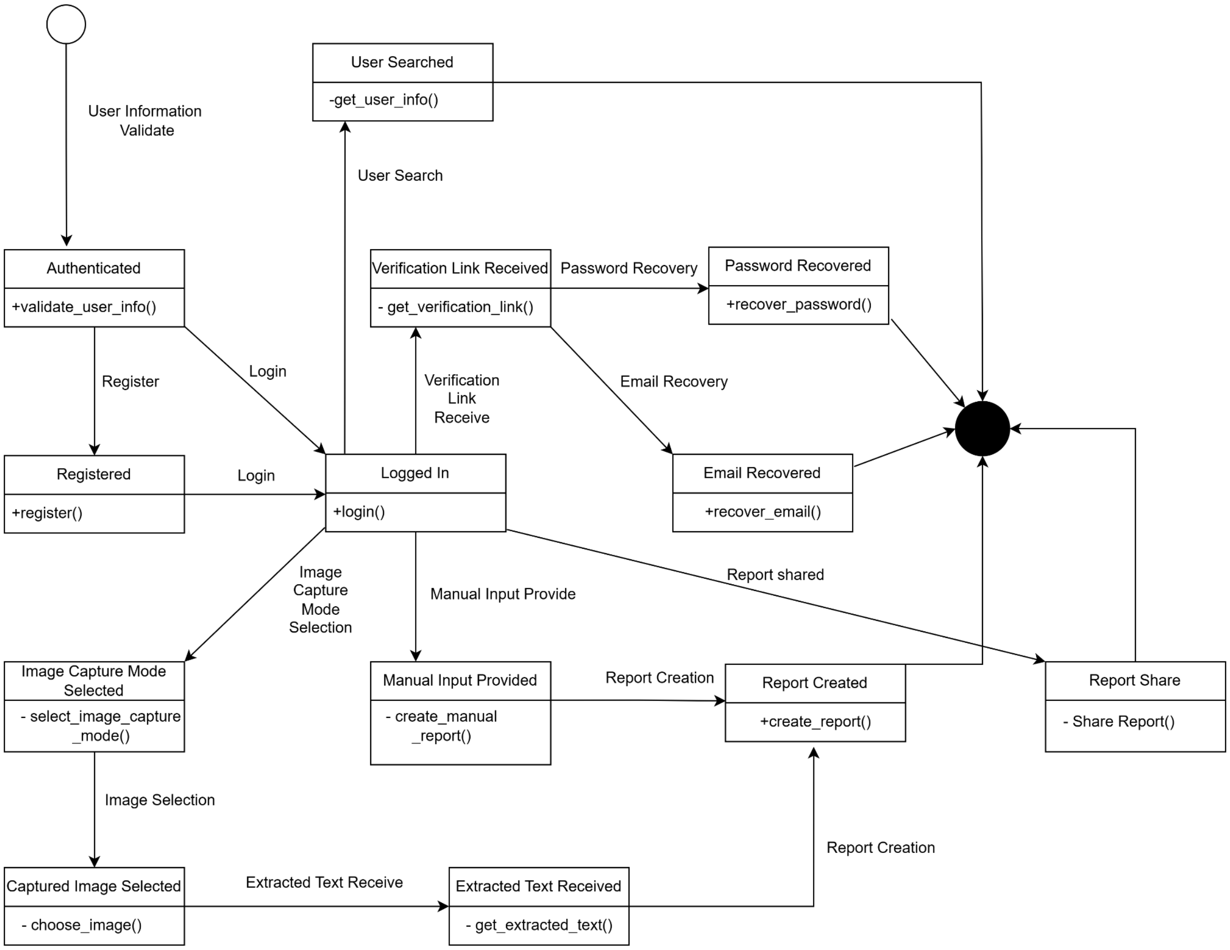
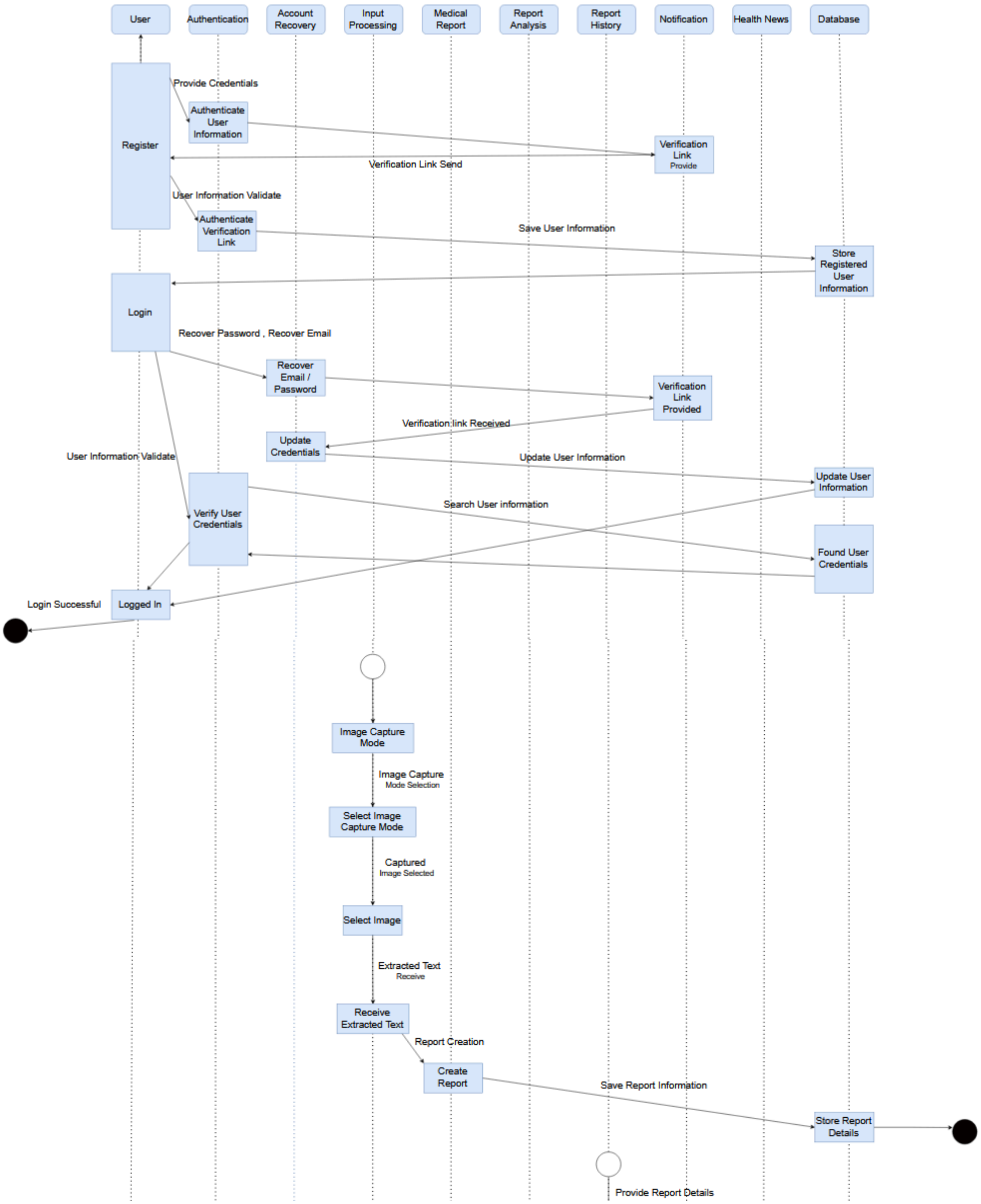


Figure 26: State transition Diagram

7.3. Sequence Diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration.Sequence Diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time, what messages are sent and when.



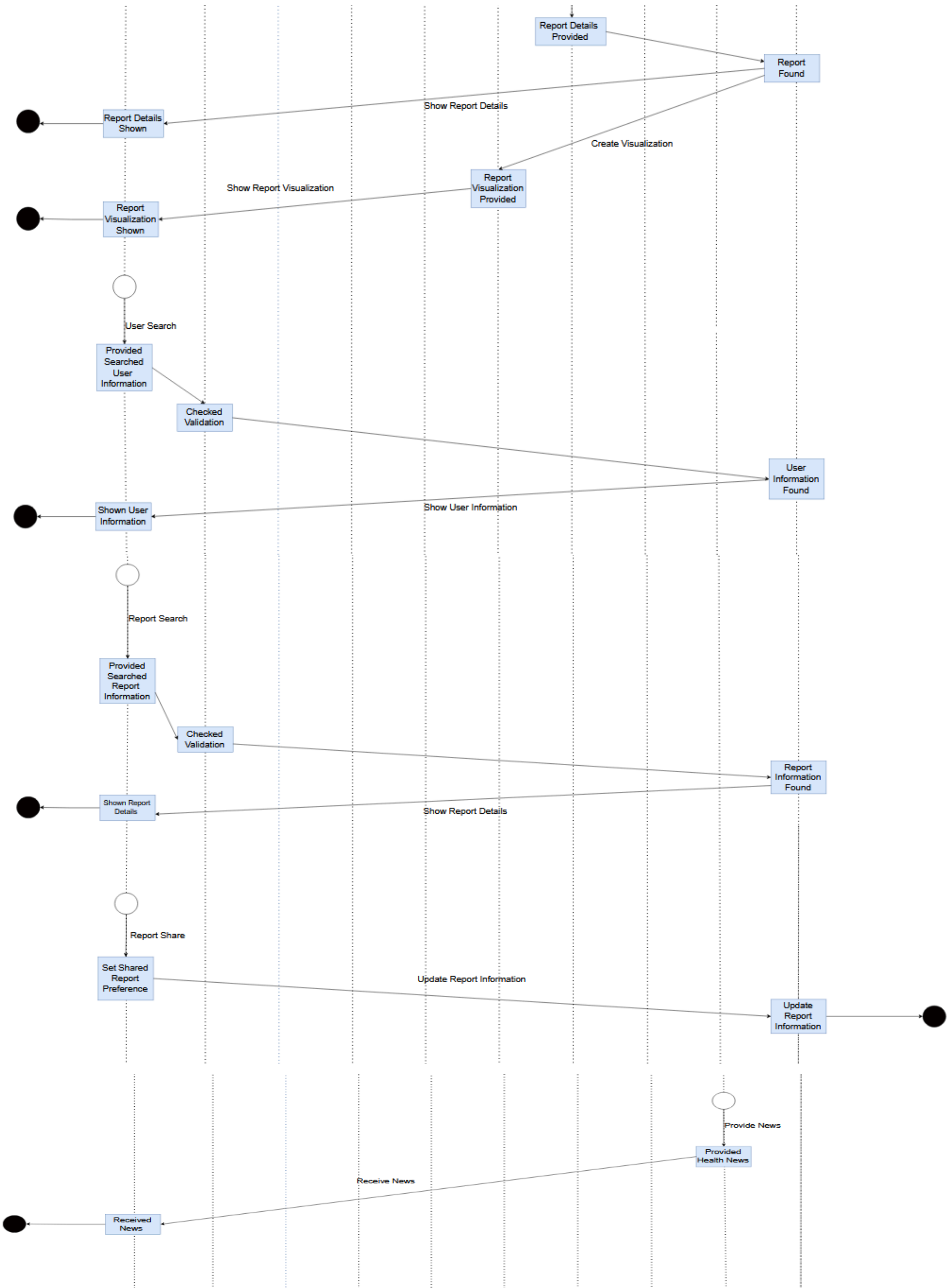

Report Details Provided
Report Found
Show Report Details
Report Details Shown
Create Visualization
Report Visualization Provided
Show Report Visualization
Report Visualization Shown
User Search
Provided Searched User Information
Checked Validation
User Information Found
Show User Information
Shown User Information
Report Search
Provided Searched Report Information
Checked Validation
Report Information Found
Show Report Details
Shown Report Details
Report Share
Set Shared Report Preference
Update Report Information
Update Report Information
Provide News
Provided Health News
Receive News
Received News

Figure 27: Sequence Diagram

7. Data flow Diagram

8 Data flow diagram

Figure ID - 0 : PHTAR

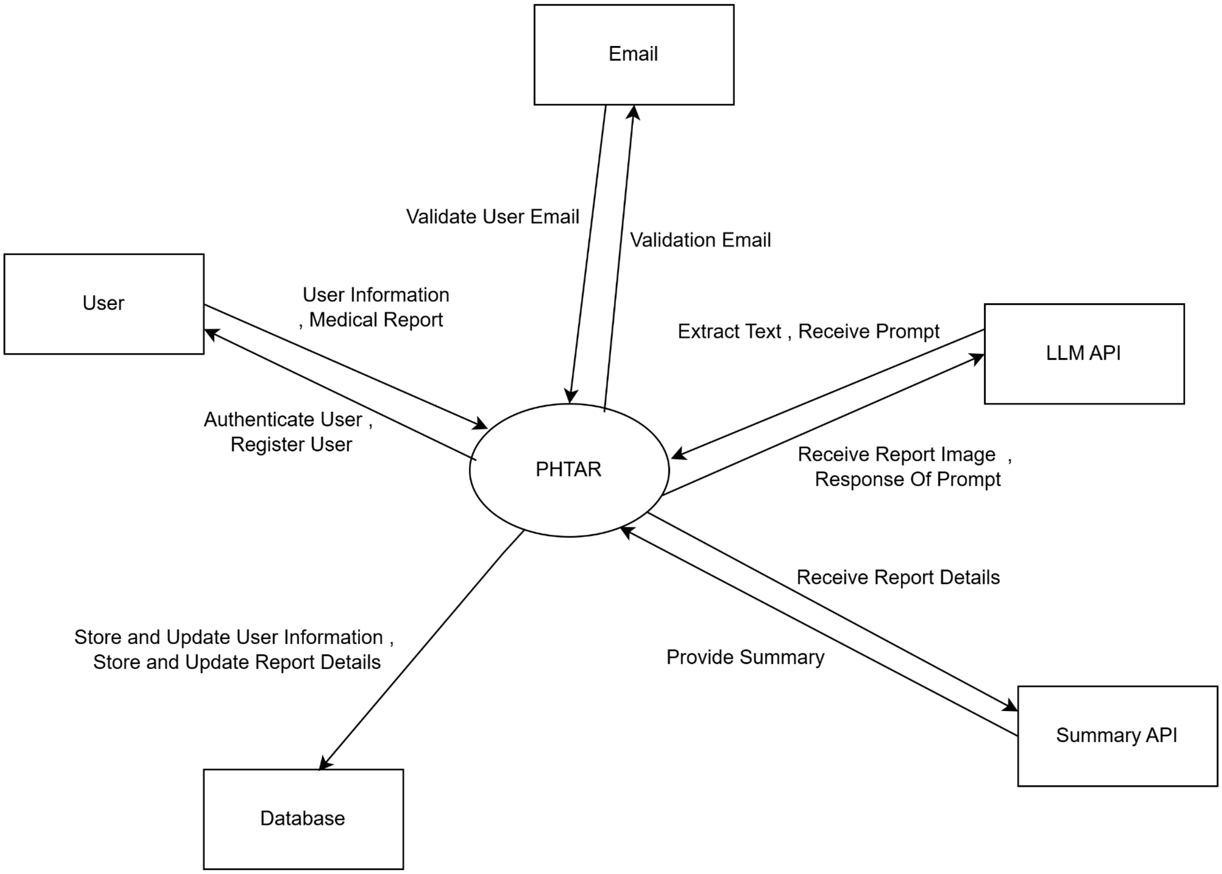


Figure 28: Level-0 Data flow Diagram

Figure ID - 1: PHTAR (Detailed)

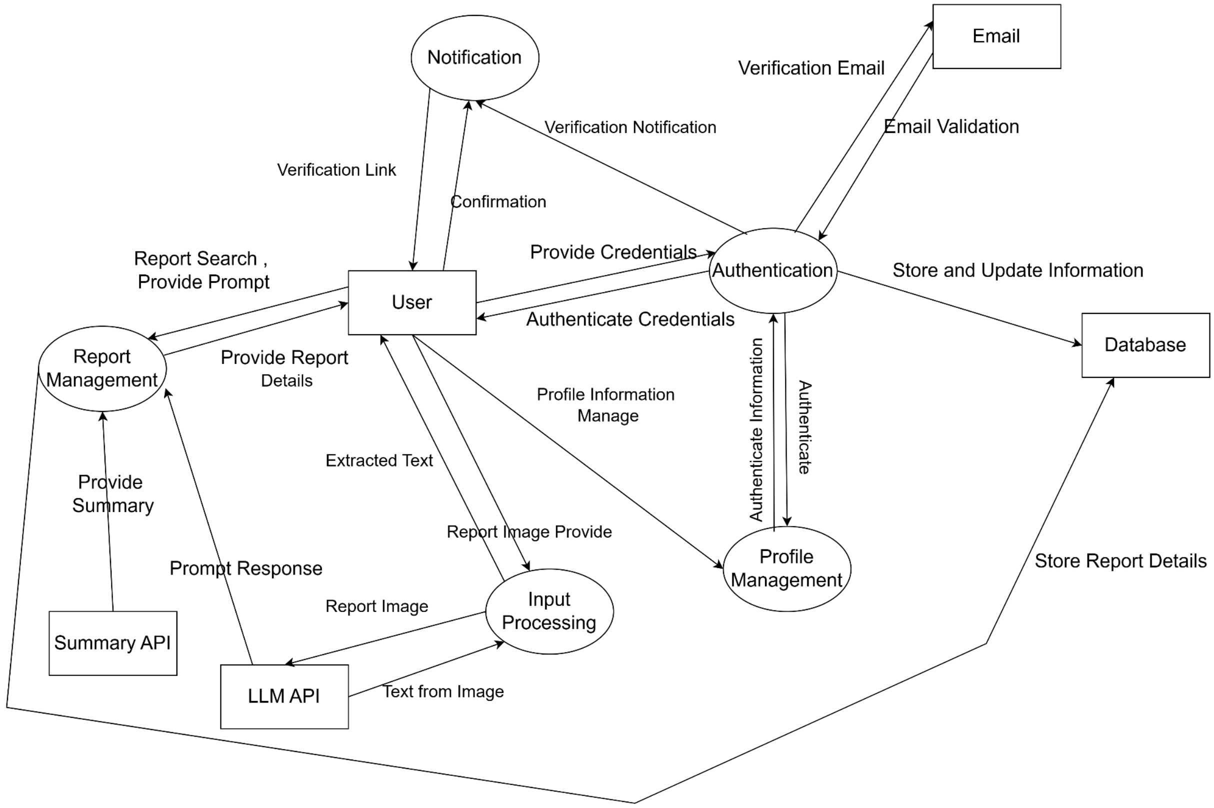


Figure 29: Level-1 Data flow Diagram

Figure ID - 2 : Authentication

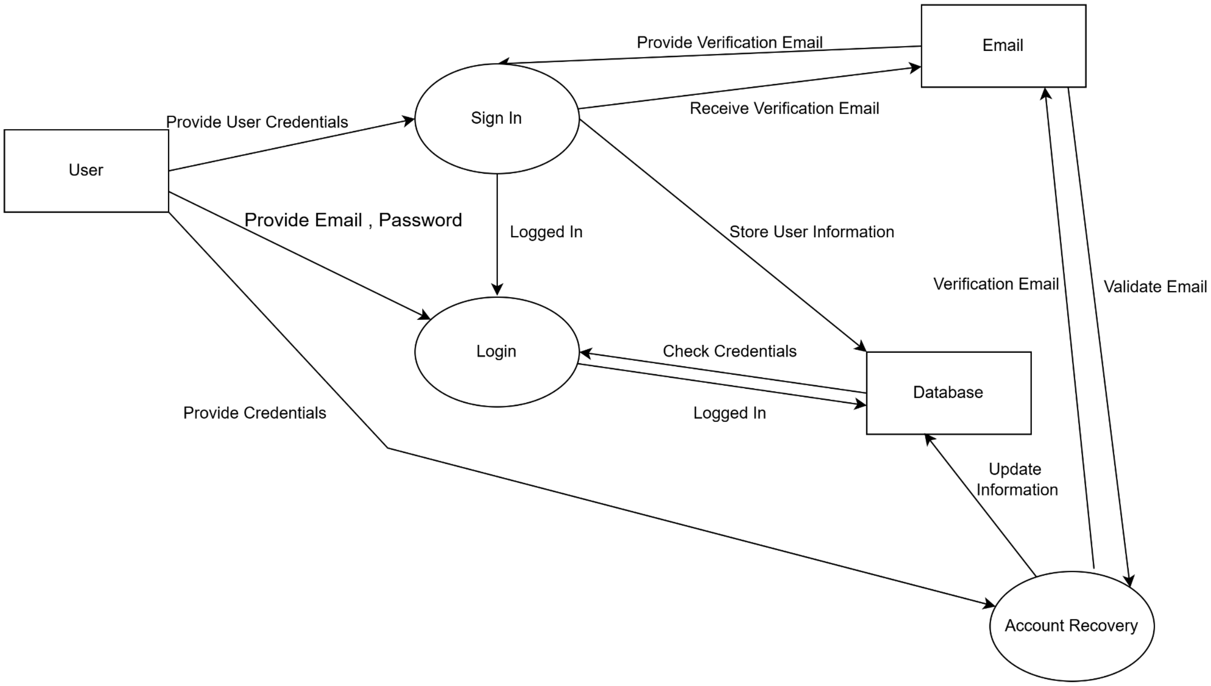


Figure 30: Level-1.1 Data flow Diagram

Figure ID - 3 : Profile Management

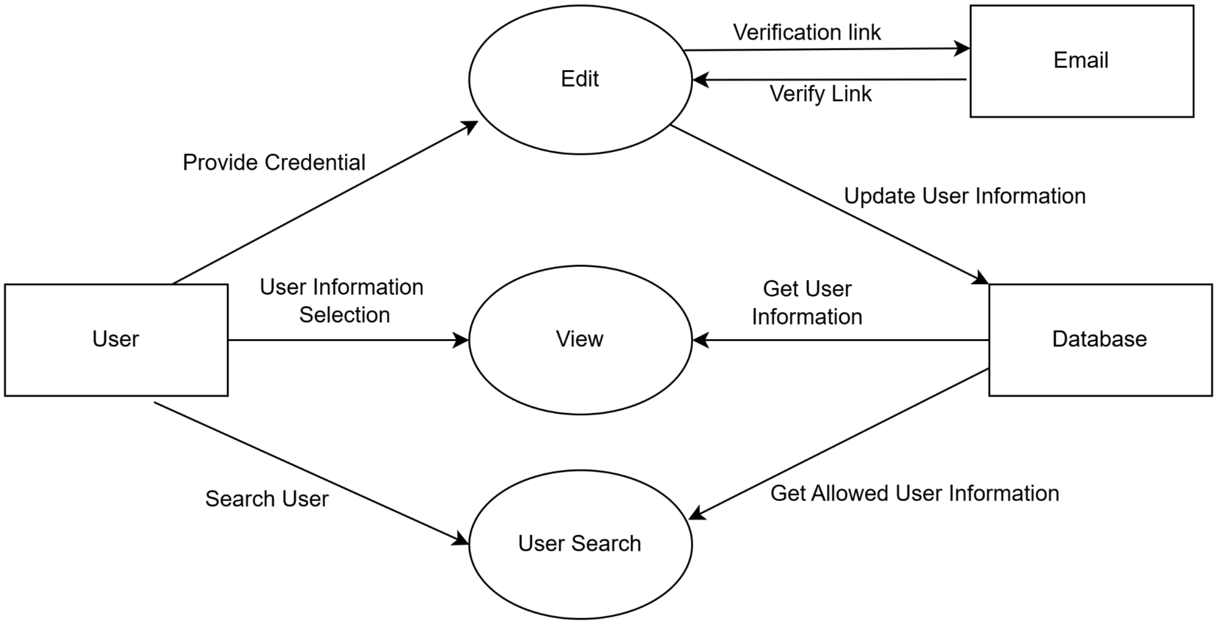


Figure 30: Level-1.2 Data flow Diagram

Figure ID - 4 : Input Processing

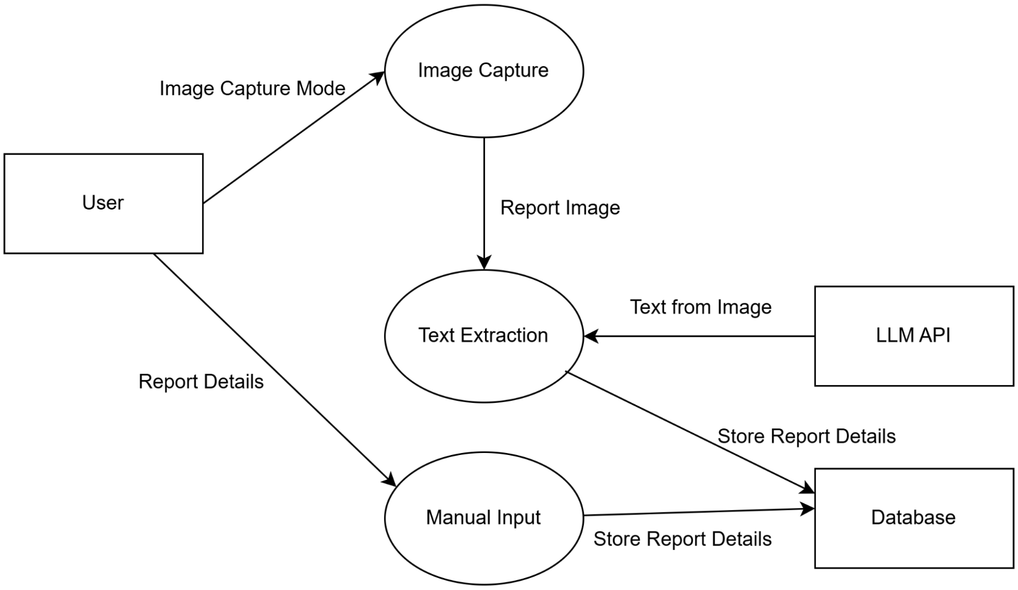


Figure 31: Level-1.3 Data flow Diagram

Figure ID - 5 : Report Analysis

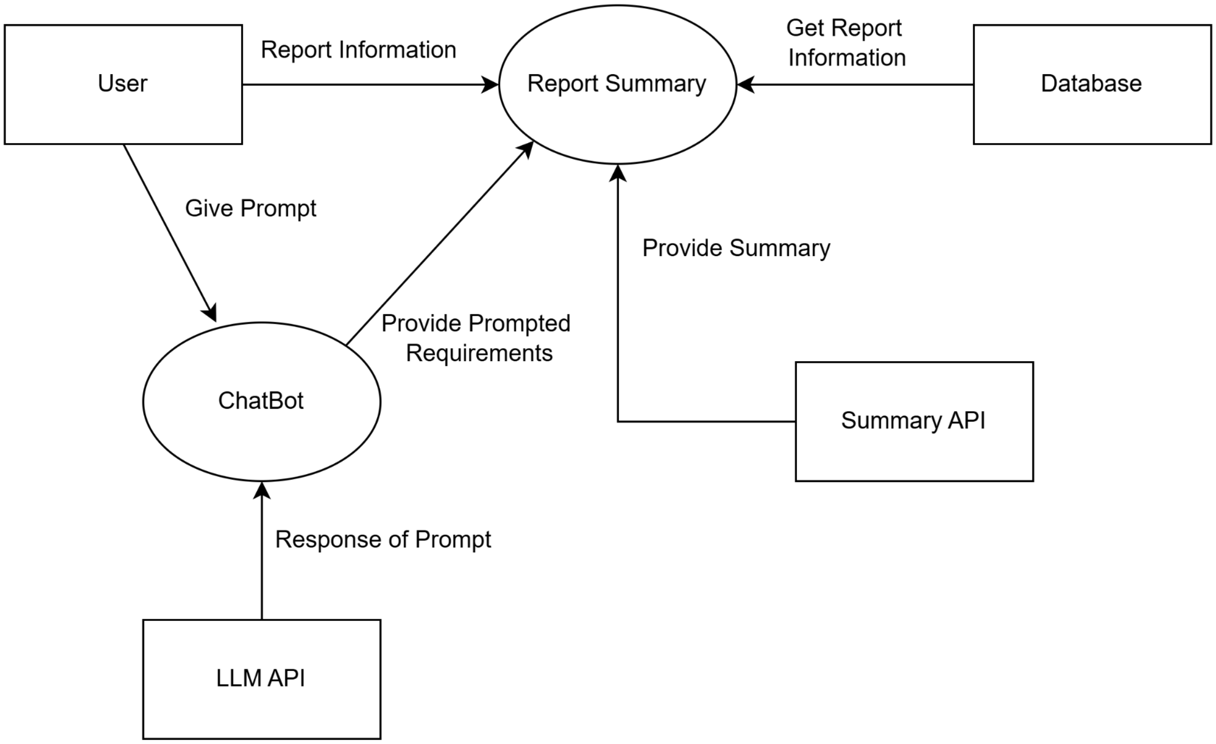


Figure 31: Level-1.4.1 Data flow Diagram

Figure ID - 6 : History Management

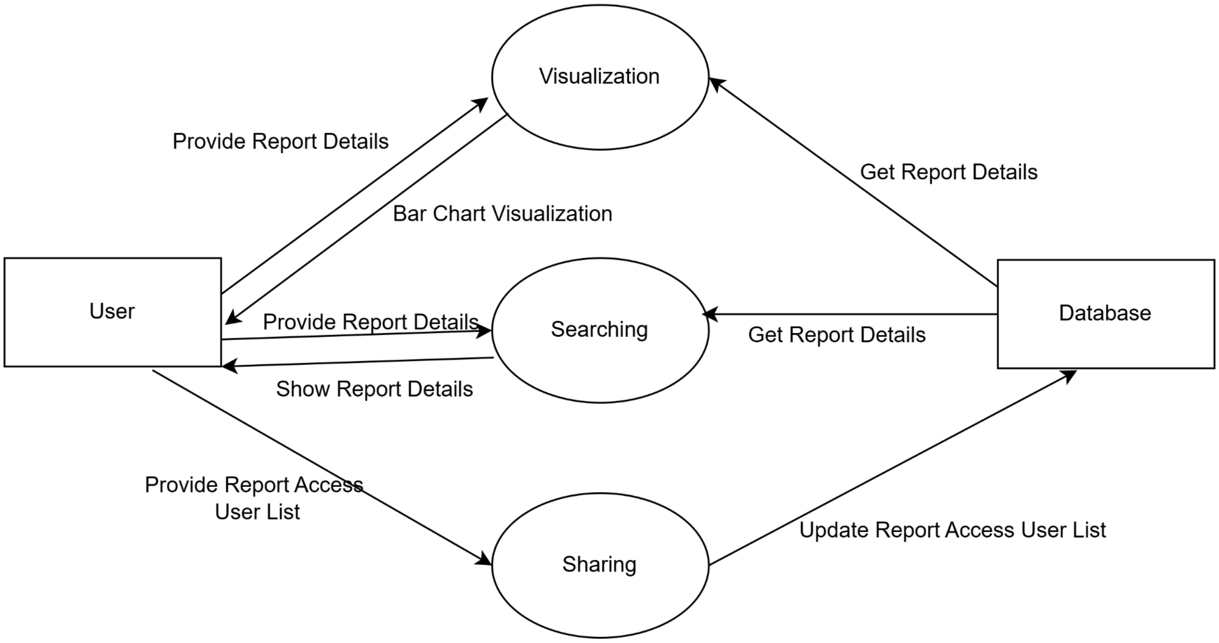


Figure 32: Level-1.4.2 Data flow Diagram