



Medi-Caps Institute of Technology & Management

INDORE

SMARTDB

A Major Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

In partial fulfillment of the degree

of

Bachelor of Engineering

in

Computer Science & Engineering

Guided By:

Mr. Mohsin Sheikh

Submitted By:

Abhishek Jain (0812IT141001)

Surendra Lalwani (0812IT141053)

Suyash Jain (0812CS141102)

Varun Karandikar (0812CS141108)

Department of Computer Science and Engineering

June, 2018



Medi-Caps Institute of Technology and Management
INDORE

SMARTDB

A Major Project Report Submitted at
Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal
In partial fulfillment of the degree
of

Bachelor of Engineering
in
Computer Science & Engineering

Guided By:
Mr. Mohsin Sheikh

Submitted By:
Abhishek Jain (0812IT141001)
Surendra Lalwani (0812IT141053)
Suyash Jain (0812CS141102)
Varun Karandikar (0812CS141108)

Department of Computer Science and Engineering
June, 2018



INDORE

Medi-Caps Institute of Technology and Management

A.B. Road, Pigdamber, Indore – 453 331 (M.P.)
Ph. : (0731) 4259700, 4259500, Fax : 4259501
Website : www.medicaps-institute.awc.in
E-mail : info@medicaps-institute.ac.in

CERTIFICATE

This is to certify that **Abhishek Jain (0812IT141001)**, **Surendra Lalwani (0812IT141053)**, **Suyash Jain (0812CS141102)**, **Varun Karandikar (0812CS141108)** have completed their Major project work titled “**SmartDB**” as per the curriculum and submitted a satisfactory report on this project as a part of fulfillment towards the degree of **Bachelor of Engineering (Computer Science & Engineering)** from **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal**.

Signature: _____

Name: **Mr. Dheeraj Rane**

(Project Co-ordinator)

Signature: _____

Name: **Mr. Mohsin Sheikh**

(Project Guide)



INDORE

Medi-Caps Institute of Technology and Management

A.B. Road, Pigdamber, Indore – 453 331 (M.P.)
Ph. : (0731) 4259700, 4259500, Fax : 4259501
Website : www.medicaps-institute.awc.in
E-mail : info@medicaps-institute.ac.in

CERTIFICATE

This is to certify that **Abhishek Jain (0812IT141001), Surendra Lalwani (0812IT141053), Suyash Jain (0812CS141102), Varun Karandikar (0812CS141108)** have completed their Major project work titled “**SmartDB**” as per the curriculum and submitted a satisfactory report on this project as a part of fulfillment towards the degree of **Bachelor of Engineering (Computer Science & Engineering)** from **Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal.**

Signature: _____

Name : _____

(Internal Examiner)

Signature: _____

Name : _____

(External Examiner)

ACKNOWLEDGEMENT

The most awaited moment of successful completion of an endeavor is always a result of persons involved implicitly or explicitly in it. The successful completion of all the phases of our project is the result of dedicated efforts put in by many people and this report would be incomplete without giving due credits to them. This acknowledgment is but a small token of gratitude in recognition of their help in our endeavor.

We express a deep sense of gratitude to our project coordinator **Mr. Dheeraj Rane** for his invaluable support and constant encouragement which was of great support to bring this work to present shape.

No word will suffice to thank our project mentor **Mr. Mohsin Sheikh, Assistant Professor (CSE)** for his cooperation, invaluable guidance and encouragement which extended us for completion of our project, both in knowledge and morale

We would like to give our sincere thanks to **Dr. Suresh Jain, HOD of Computer Science and Engineering Department** from whom we have learnt a lot.

We are deeply indebted to **Dr. Sunil K. Somani, Director** who provided us a well-equipped computer laboratory with all required software and library, which was an essential requirement for the success of this project.

Last but not the least, we would like to thanks our colleagues, friends and our parents who were a constant and willing source of encouragement and inspiration for us throughout the project.

Abhishek Jain (0812IT141001)
Surendra Lalwani (0812IT141053)
Suyash Jain (0812CS141102)
Varun Karandikar (0812CS141108)

CONTENT

1. Introduction.....	1-3
1.1 Objective.....	1
1.2 Scope.....	1
1.3 Problems Domain.....	1
1.4 Solution Domain.....	2
1.5 Platform Specification.....	3
1.5.1 Hardware.....	3
1.5.2 Software.....	3
2. System Requirement Analysis.....	4-6
2.1 Information Gathering.....	4-5
2.1.1 Functional Requirement	4
2.1.1 Non Functional Requirement	5
2.2 System Feasibility.....	6-7
2.2.1 Operational	6
2.2.2 Technical.....	6
2.2.3 Economical.....	6
3. System Analysis.....	8-15
3.1 Use Case Diagram.....	8
3.2 Class Diagram.....	9
3.3 Sequence Diagram.....	11
3.4 Collaboration Diagram.....	13
3.5 Activity Diagram.....	13
3.6 Swim lane Diagram.....	14
4. Design.....	16-20
4.1 Data Design.....	16-17
4.1.1 ER Diagram.....	16

4.1.2 Data Dictionary.....	17
4.2 Architectural Design.....	18-20
4.1.2 Architectural Style Used.....	19
4.1.3 System Architecture.....	19
5. Implementation.....	21-29
5.1 Implementation of Modules.....	21
5.2 Results.....	29
6. Testing.....	30-33
6.1 Testing Methods used.....	30
6.2 Test Cases.....	31
7. Conclusion and Future Work.....	34
8. References.....	35

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1	Use Case Diagram	9
2	Class Diagram	11
3	Sequence Diagram	12
4	Collaboration Diagram	13
5	Activity Diagram	14
6	Swim Lane Diagram	15
7	ER Diagram	17
8	Architecture Design	19
9	Structure Chart	20

LIST OF TABLES

Table No.	Table Name	Page No.
4.1.2	Data Dictionary	17
4.1.2.1	Add Member Details	17
4.1.2.2	Registration Details	17
4.1.2.3	Chat Messages	18
4.1.2.4	Log Details	18
4.1.2.5	Recognizer	18

ABSTRACT

The Internet of Things (IoT) is the inter-networking of physical devices, vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data. The idea completely aims to make the routine easier and secure in a best possible efficient manner.

The process of controlling or operating various equipment, machinery, industrial processes, and other applications using various control systems and also with less or no human intervention is termed as automation. There are various types of automation based on the application they can be categorized as home automation, industrial automation, autonomous automation, building automation, etc. Home automation is the process of controlling home appliances automatically using various control system techniques. There are various techniques to control home appliances such as IoT based home automation over the cloud, home automation under Wi-Fi through android apps from any smartphone. Wireless home automation using IoT is an innovative application of internet of things developed to control home appliances remotely over the cloud. SmartDB is based on state of the art IoT technology and uses Android as the base platform for the development of our product. The idea of our product is to provide secure access to home, office buildings and educational institutions. It will be achieved through smart doorbell, once the visitor pressed the doorbell, it will publish an alert will be sent to all the family members using Firebase, which is a mobile and web application development platform by instant notification on Android Phone.

1. Introduction

1.1 Objective:

In today's world security is of at most concern and to achieve it with high level one needs to invest more amount of money which is not feasible for everyone in this world. So, SmartDB makes the task easier by providing the maximum security in minimal cost.

- Remote door control
- Surrounding Monitoring
- Instant notification alert
- Easy and fast logs management
- Authenticated users access

1.2 Scope:

Internet of Things (IoT) makes our world as possible as connected together. Nowadays, we almost have internet infrastructure everywhere allowing access as and when needed. Embedded computing devices would be exposed to internet influence. Common instances for embedded computing devices are MP3 players, MRI, traffic lights, microwave ovens, washing machines and dishwashers, GPS even heart monitoring implants or biochip and etc. In a nutshell, IoT aims to connect all potential objects to interact with each other on the internet to provide secure, comfort life for human.

SmartDB is based on state of the art IoT technology and uses Android as the base platform for the development of our product. The idea of our product is to provide secure access to home. It will be achieved through smart doorbell, once the visitor pressed the doorbell, it will publish an alert which will be sent to all the family members using Firebase Cloud Messaging service by instant notification on Android Phones, so the family members know someone is knocking at the door no matter where the person is. Visitors no longer need to call the person and simply let the smart doorbell to do the job, deaf people also benefit using it so they can get an alert from vibration of their phone. After receiving the image of the person who pressed the doorbell on family members' phone, one can designate the user as authorized or unauthorized depending upon the perspective and the same along with the image will be stored in the database which will be used in the future. If the person was categorized as authorized and whenever the same person arrives again then the same procedure won't be repeated and just a message will come along with their name and hence in this way security is achieved.

1.3 Problem Domain

In today's world, security is one of the major concern in households and industries. With so much advancement in technology, we can't leave our home with just a metal lock, we need intelligent

security system. With criminals, burglaries and theft on rise, has caused severe collateral damage to life and property.

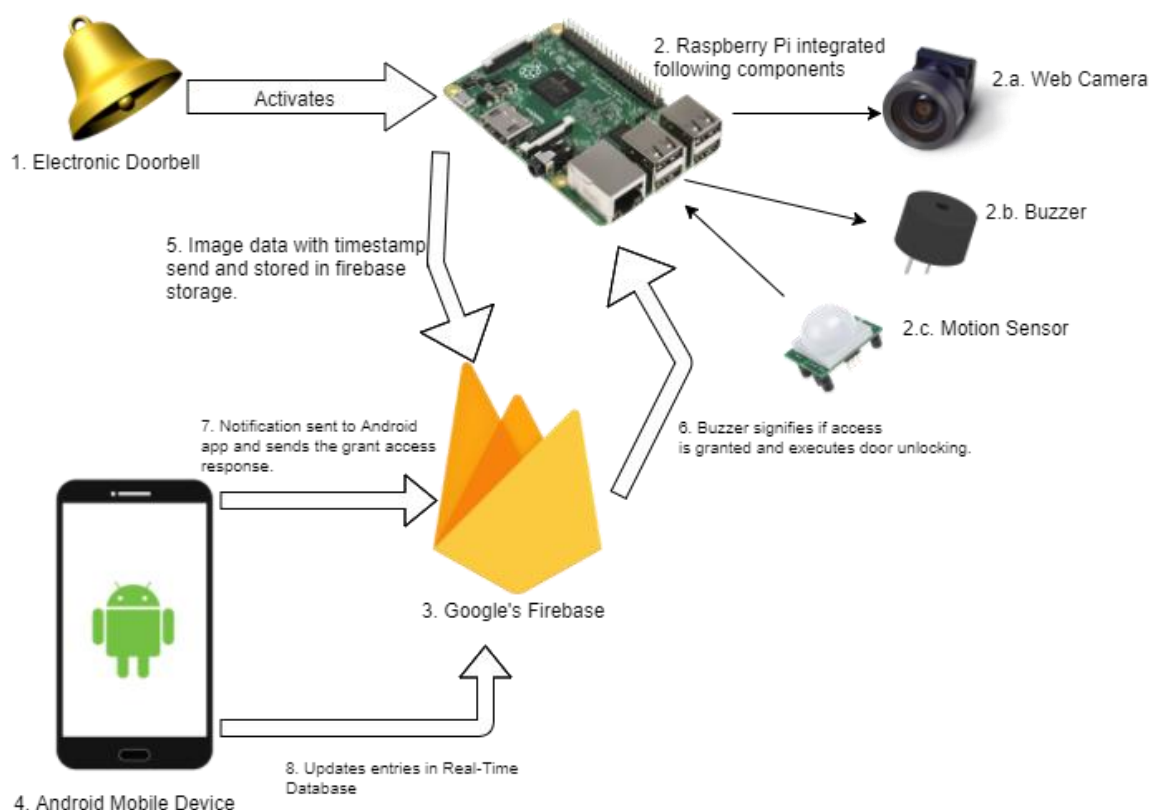
When people are away from their homes still they want to know if someone is at their doorstep and working parents do not want their children to answer the door and leaving the house without their knowledge. Also, when someone arrived with useful information at the door and no one is there to attend then there is a problem.

Currently available security system are very expensive as well as inflexible to be incorporated in existing buildings. By inflexible, we mean existing system are cumbersome to use as well as installed.

1.4 Solution Domain

SmartDB is an IoT based application aiming to ease the user problems with the smart automation and that too in an affordable manner. Android is having the largest share in today's corporate world and hence our product will be available to all at a reasonable cost. When a person is alone at their home SmartDB will alert family members about the one standing behind the door. With SmartDB, we aim to provide security in your hands via mobile phones.

Our product will revolutionize how one will track and manage access to their home. In the near future, our product should be able to keep tabs on inventory changes completely automatically, freeing up human interaction. It's not just about the house made of bricks rather it is home made up of intelligence. It's not all about speed—you'll also be able to get more done in less time. In addition to instant gratification, technological evolution also tends to favor productivity and efficiency. Our product will allow you to accomplish remote tasks faster and with greater precision, including data analysis and management.



1.5 Platform Specification

The use of embedded device allows to interface external devices with the electronic components and also functions as local server to act as an end point for socket programming with the other end being the android application on the mobile platform or the virtual machine to govern the flow of the system.

1.5.1 Hardware

- Embedded Device : Raspberry Pi 2 Model B
- Peripheral Kit (Network Wifi Adapter, Ethernet Cable)
- External Camera
- Any Android Enabled Handheld
- Basic Electronic components

1.5.2 Software

- Raspberry Pi 2 Model B running Raspbian Stretch OS
- Ubuntu 17.10 Virtual Machine
- Server Side : Database Server – Firebase Server
- Client End : Android
- Firebase Realtime Database
- Firebase Storage Service
- Firebase Authentication Service
- Firebase Cloud Messaging Service
- Tensorflow library
- Keras
- OpenCV

IDEs used are:

1. Android Studio
2. Sublime Text
3. IDLE

Tools used are:

- Adobe Illustrator
- Adobe Photoshop

Languages used are:

- Java
- Python
- XML
- SQL

2. System Requirements Analysis

2.1 Information Gathering

Information gathering refers to gathering information about the issue you're facing and the ways other organisations and communities have addressed it. The more information you have about the issue itself and the ways it has been approached, the more likely you are to be able to devise an effective program or intervention of your own.

2.1.1 Functional Requirements:

These are the statements of services the system should provide, how the system should react to a particular inputs and how the system should behave in a particular situation.

FR1: Pressing the Doorbell

DESC:

When the user will enter the house and press the doorbell and the required functions will be performed.

DEP: None

FR2: Saying the name

DESC: After pressing the doorbell, the bell will prompt user to tell his name or reveal any identity.

DEP: FR1

FR3: Message to owner

DESC: After the visitor has told his name, he/she (owner) will get the message about the visitor and he/she (owner) will instruct the doorbell and the message will be as it is printed on the LED SCREEN and the same will be pronounced by speaker.

DEP: FR2

FR4: User Authentication

DESC: After the message has been heard by visitor then too if he/she (visitor) tries to do the unauthorised access, then an alarm will buzz.

DEP: FR3

FR5: Maintain Records

DESC: If an authorized person has arrived in the home so his/her data will be stored in the database which will be achieved through Machine Learning Concept, so that whenever the same person enters again the same procedure won't be repeated from the next time and owner gets instant notification about the same.

2.1.2 Non-Functional Requirements:

In requirement engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system rather than specific behaviours.

The performance requirements are as follows:

- System login/logout shall take less than 5 seconds.
- Searches shall return results within 10 seconds.

a. Reliable:

Our IoT system will be reliable as it will be completely fault prone and will also resist changes.

b. Portable:

Our system will be small in size and can easily be carried and implemented ubiquitously and hence our system will be portable.

c. Security:

Our system is solely made for security and will itself be secure enough and will be far away from any type of malware attacks.

d. Availability:

Our system will be active for 24*7 and 365 days.

• GPS Connection:

The app should be connected to the GPS device. In order for the app to get the user's location, the map and to calculate the distance.

• Internet Connection:

The app should be connected to the Internet. In order for the app to communicate with database.

e. Maintainability:

• Application extendibility:

Our software can easily extended in near future. The code should be written in a way that it favors implementation of new functions. Any updates or defect fixes shall be able to be made on server-side computers only without any patches required by the user.

- Application testability:

Test environments should be built for the application to allow testing of the application's different functions.

f. Portability:

Our application can be run on android enabled phones having android OS (SDK 17 to SDK 23).

2.2 System Feasibility

System feasibility aim to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects for success.

The feasibility study activity involves the analysis of the problem and collection of the relevant information relating to the project. The main aim of the feasibility study is to determine whether it would be financially and technically feasible to develop the product.

2.2.1 Operational Feasibility

Operational feasibility is the measure of how well the project will support the customer and the service provider during the operational phase. It is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented.

2.2.2 Technical Feasibility

This assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the needs of the proposed system. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements.

Technical feasibility also performs the following tasks.

- Analyzes the technical skills and capabilities of the software development team members
- Determines whether the relevant technology is stable and established
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

2.2.3 Economical Feasibility

This assessment aims to determine the positive economic benefits to the organization that the proposed system will provide. It typically involves a cost/ benefits analysis and it's the most frequently used method for evaluating the effectiveness of a new proposed system. Economic justification is generally

the bottom line consideration for most system. Economic justification includes broad range of concerns that includes cost benefit analysis. It involves estimating benefits and costs. These benefits and cost may be tangible or intangible.

3. System Analysis

3.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various sways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

Actors

The actors, usually individuals involved with the system defined according to their roles.

There are 3 actors:

- User
- Admin
- Visitor

Use Cases

The use cases, are the specific roles played by the actors within and around the system. This subsection extends upon the functional requirements through the presentation of detailed use cases. There are following Use cases in this project:

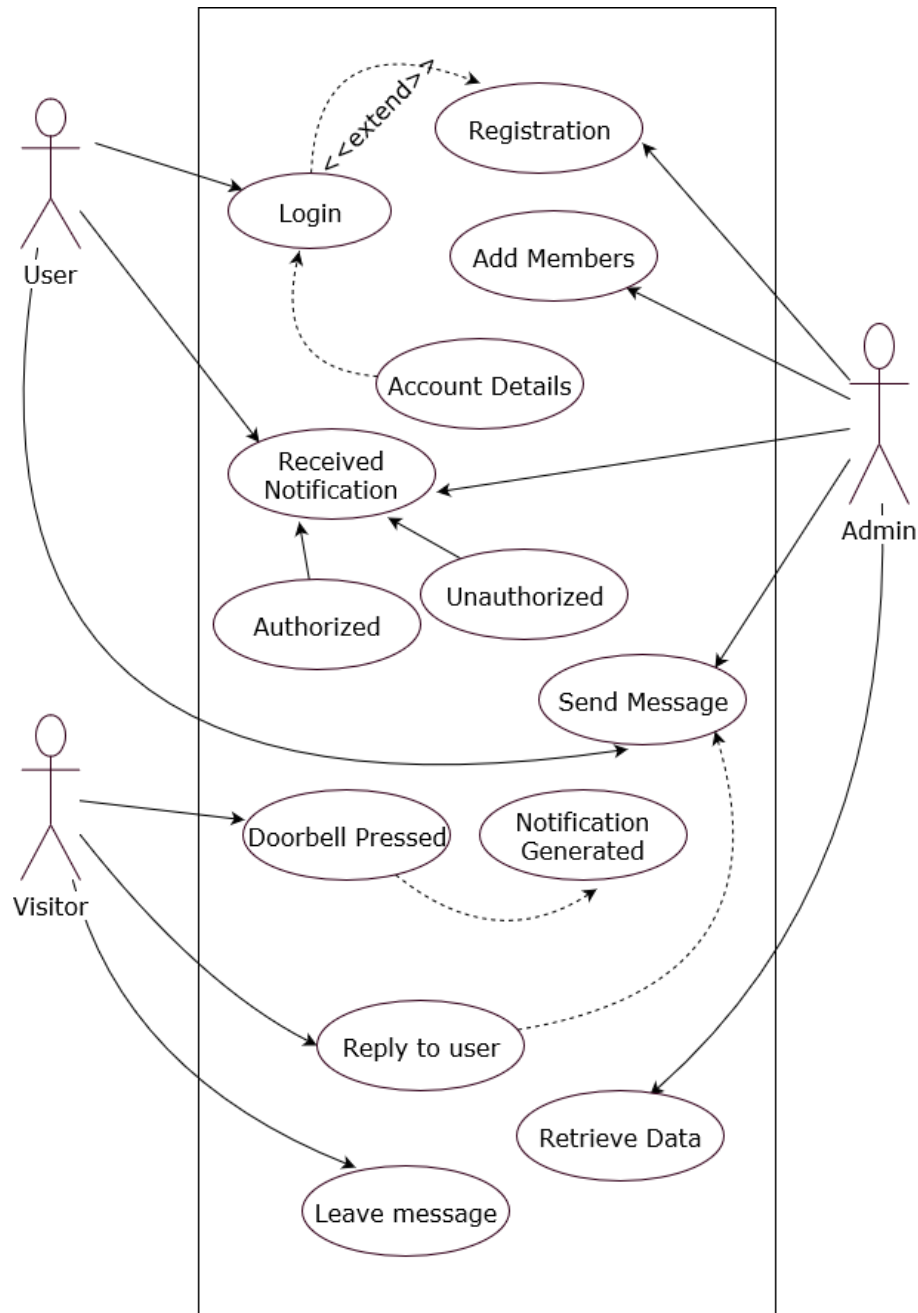
- Login
- Registration
- Add Members
- Account Details
- Received Notifications
- Send Message
- Doorbell Pressed
- Notification Generated
- Reply to User
- Retrieve Message
- Leave Message

System Boundary

It provides use case containment behaviour. It usually describes the system by grouping the use cases in the rectangle boundary in Visual Paradigm provides use case behaviour.

Use Case diagram

To facilitate an unambiguous and clear view of how the end-users interact, how the actors (end-users) are involved in the use cases, a use case diagram is provided:



3.2 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represents the whole system.

Classes

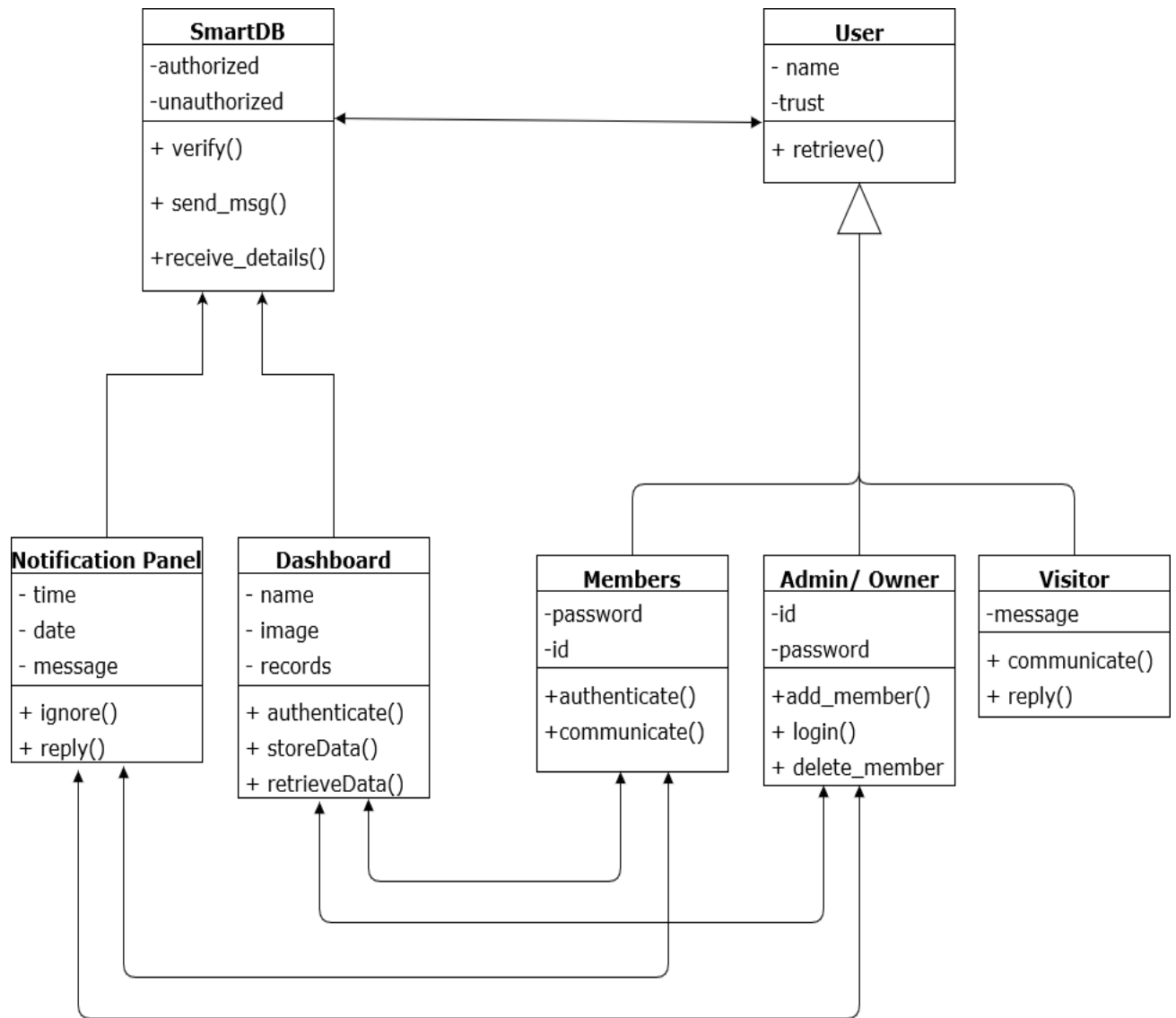
- User
- SmartDB
- Notification Panel
- Dashboard
- Members
- Admin/ Owner
- Visitors

Attributes

- User – name, trust
- SmartDB – authorize, unauthorized
- Notification Panel – time, date, message
- Dashboard – name, image, records
- Members- password, id
- Admin/ Owner – id, password
- Visitors- message

Generalization shown in:

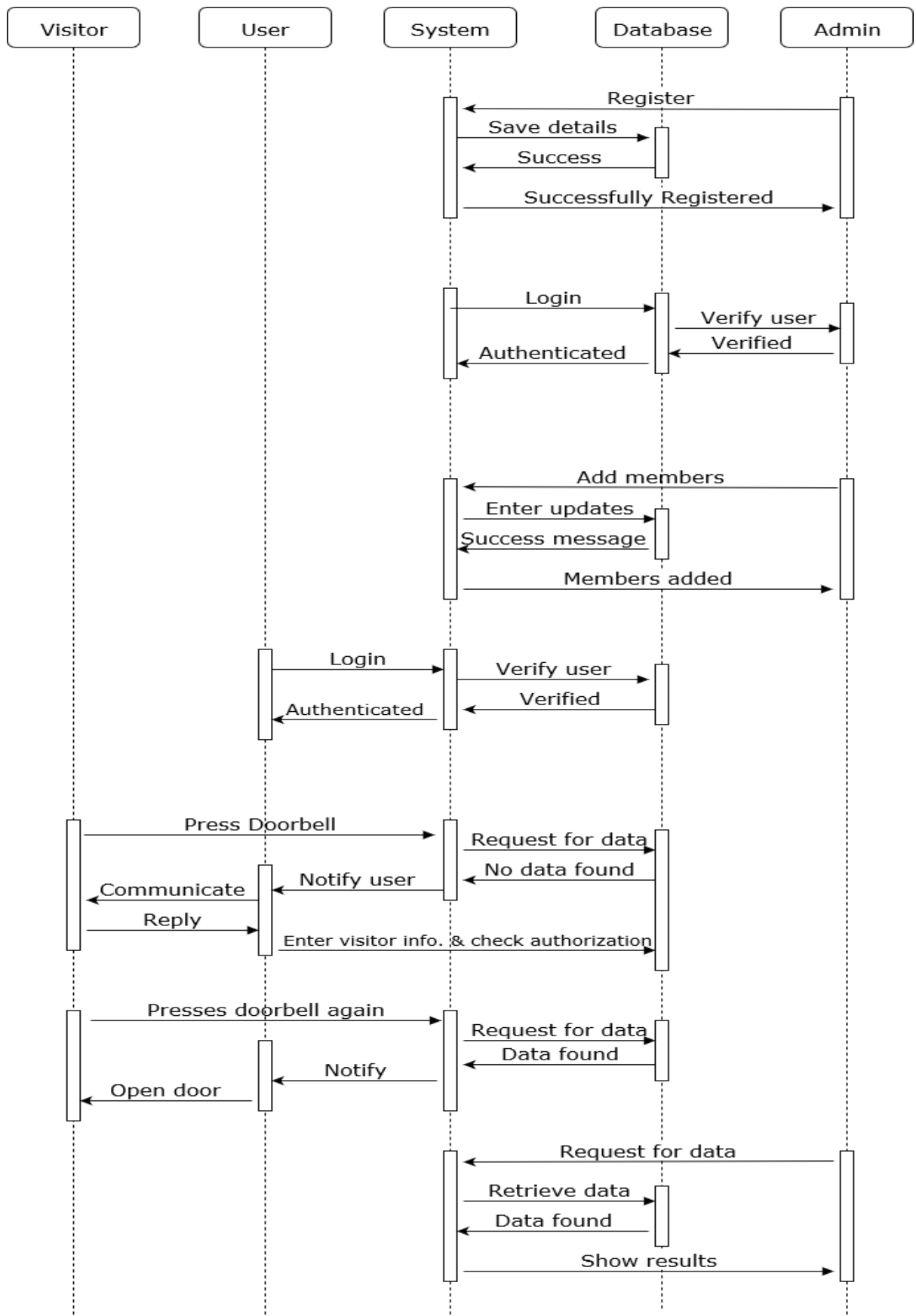
- Owner, Visitor and Admin To User



3.3 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.



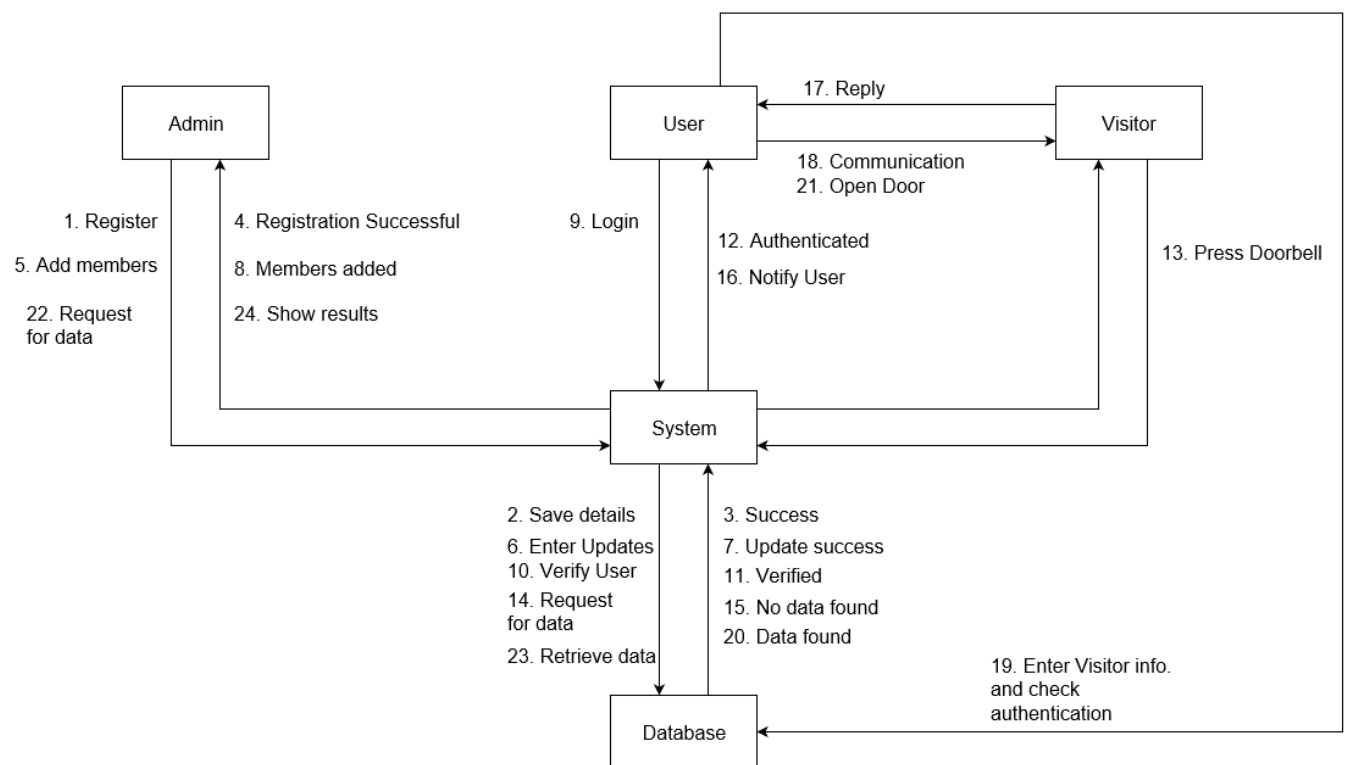
3.4 Collaboration diagram

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML).

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.

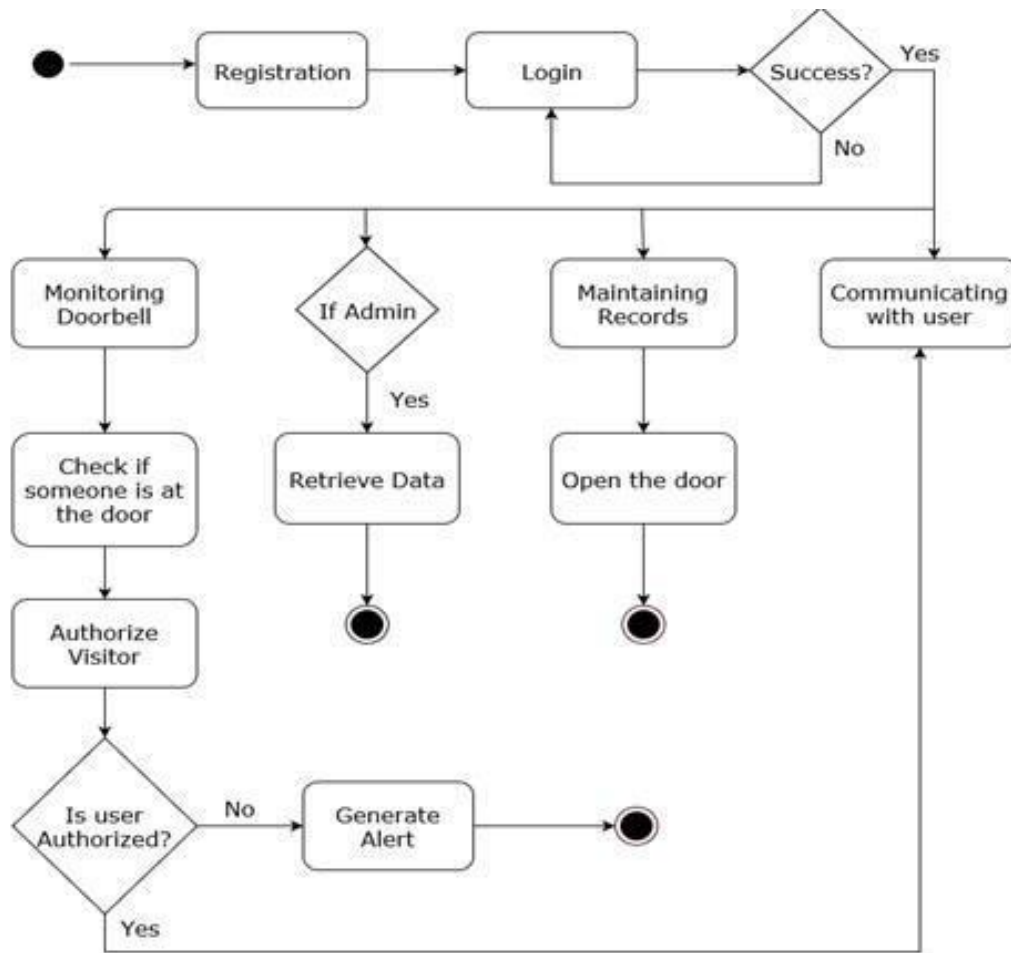
Objects:-

- Admin
- User
- Visitor
- System
- Database

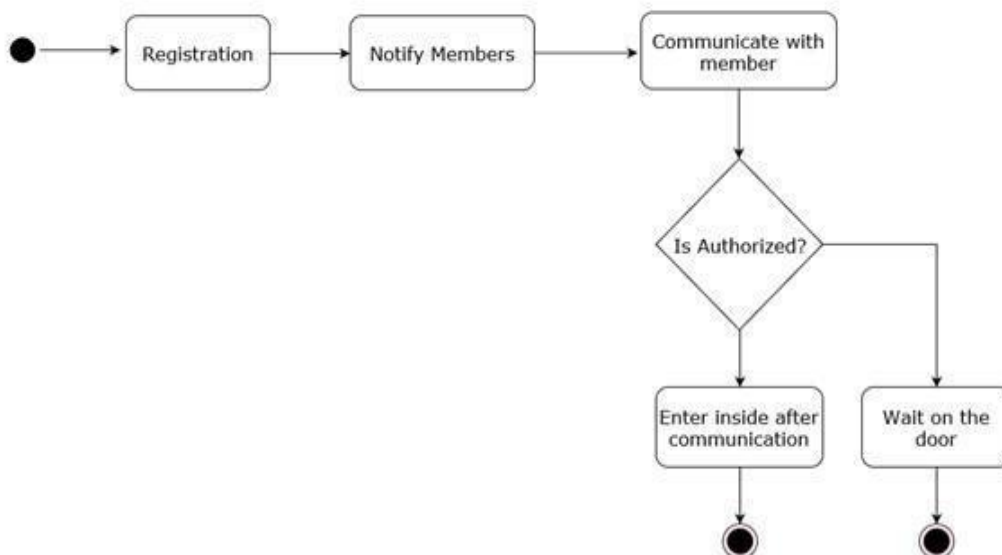


3.5 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control. The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system.



Visitor



3.6 Swim Lane Diagram

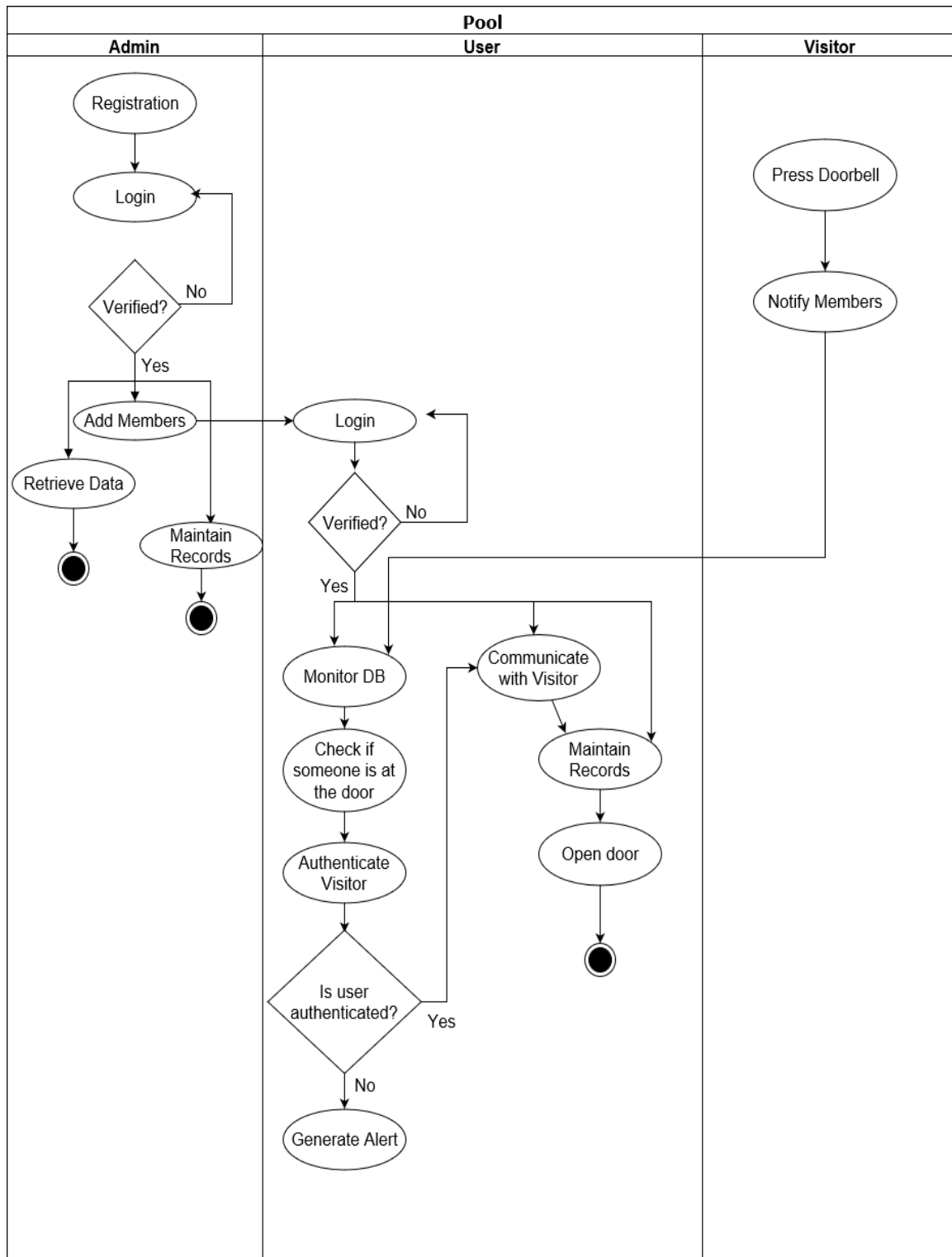
A swim lane (also known as swimlane) diagram is a type of flowchart. Like a flowchart, it diagrams a process from start to finish, but it also divides these steps into categories to help distinguish which departments or employees are responsible for each set of actions.

These lanes are columns that keep actions visually separated from others. A swim lane diagram makes responsibilities more clear than a regular flowchart. When looking to improve processes, knowing

which department is responsible for what can help speed up the process of correcting inefficiencies and eliminating delays.

Lanes-

- ADMIN
- USER
- VISITOR



4. Design

Designing data is about discovering and completely defining your application's data characteristics and processes. Data design is a process of gradual refinement, from the coarse "What data does your application require?" to the precise data structures and processes that provide it. With a good data design, your application's data access is fast, easily maintained, and can gracefully accept future enhancements.

4.1 Data Design

SmartDB make use of following data design for its working, namely:

- Users
- Admin
- Media

As the name suggests each table holds data about the respective field. The software makes use of the data stored in the database directly and so it needs no database operations to work on. All the primary keys are marked by a golden key opposite to it. It is important to note that at time of login, userId and its corresponding password is check form database of table Users table.

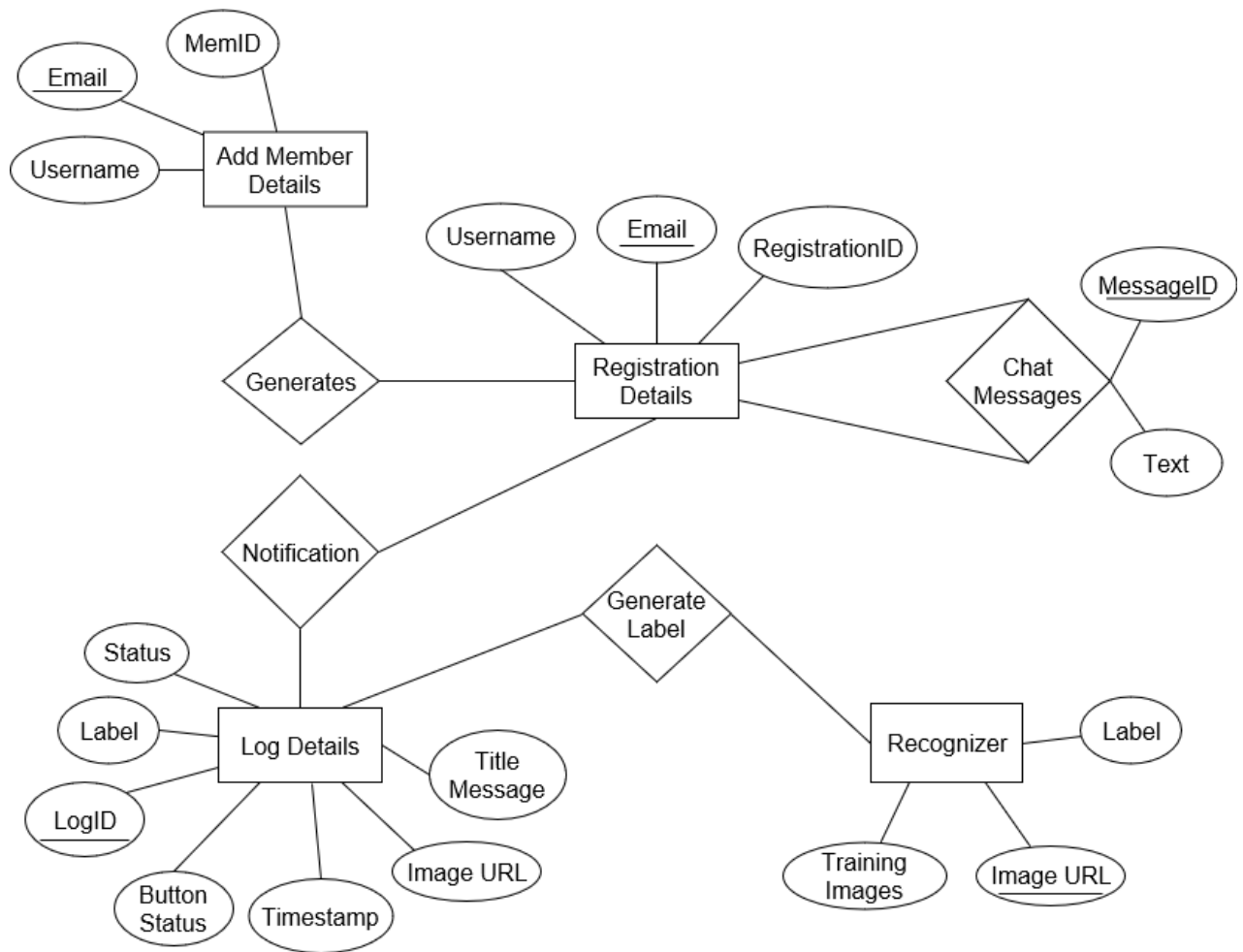
4.1.1 ER Diagram

An Entity-Relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database.

Entities are:

- Log_Details
- Registration_ Details
- Add Member_ Details
- Recognizer

Log_Details, Registration_ Details, Add Member_ Details, Recognizer are entities with properties associated with them. Relationships such as Generate, Chat messages, Generates and Notification exists among entities.



4.1.2 Data Dictionary

A **Data Dictionary**, also called a **Data Definition Matrix**, provides detailed information about the business data, such as standard definitions of data elements, their meanings, and allowable values. Essentially, a data dictionary provides a tool that enables you to communicate business stakeholder requirements in such a way that your technical team can more easily design a relational database or data structure to meet those requirements.

Add Member Details Table Schema:-

Attribute	Description	Type
Name	Name of the user	String
MemID	Unique key of the User	String
Email	Unique identification id	String

Registration Details Table Schema:-

Attribute	Description	Type
-----------	-------------	------

Username	Name of the user	String
RegistrationID	ID for user's device	String
Email	Unique identification id	String

Chat Messages Table Schema:-

Attribute	Description	Type
Name	Name of the user	String
Text	Entire Message sent by user	String

Log Details Table Schema:-

Attribute	Description	Type
Button Status	Allow/Disallow button's status	String
Timestamp	Timestamp generated at event of doorbell press	Integer
LogID	Unique ID for each log record	String
Image URL	Path of the uploaded image	String
Status	Indicates whether notification has been attended or not	String
Label	For authentication of Admin	String
Title Message	The text message displayed in the notification	String

Recognizer Table Schema:-

Attribute	Description	Type
Training Images	Set of images for training	Blob
Image URL	Path of the uploaded image	String
Label	Name of person recognized in the image	String

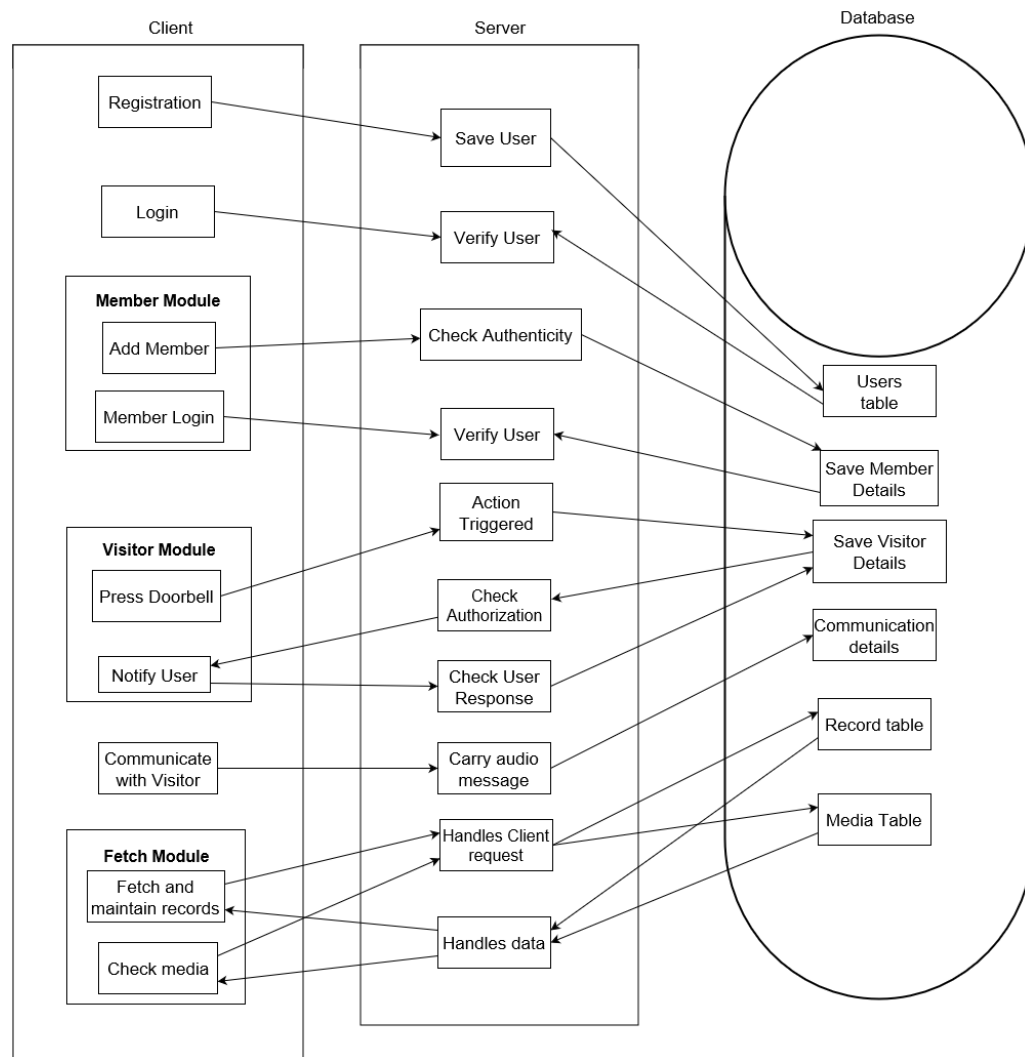
4.2 Architectural Design

The architectural design is the design process for identifying the subsystems making up the system and framework for subsystem control and communication. The goal of architectural design is to establish the overall structure of software system. An architectural style improves partitioning and promotes design reuse by providing solutions to frequently recurring problems. You can think of architecture styles and patterns as sets of principles that shape an application.

4.2.1 Architectural Style Used

Client Server Architecture Style:-

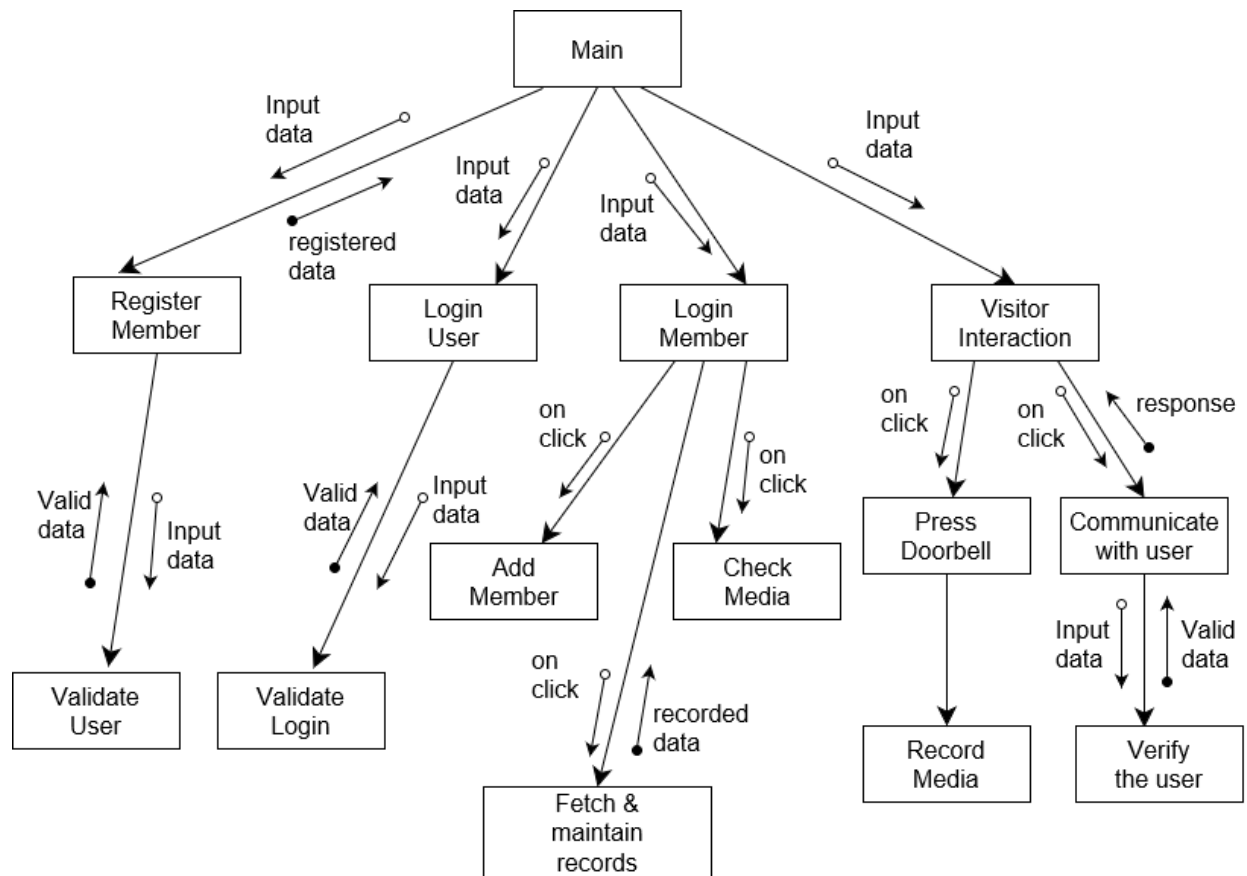
Segregates the system into two applications, where the client makes requests to the server. In many cases, the server is a database with application logic represented as stored procedures



4.2.2 System Architecture

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system.

Structure Chart:-



5. Implementation

5.1 Implementation of Modules

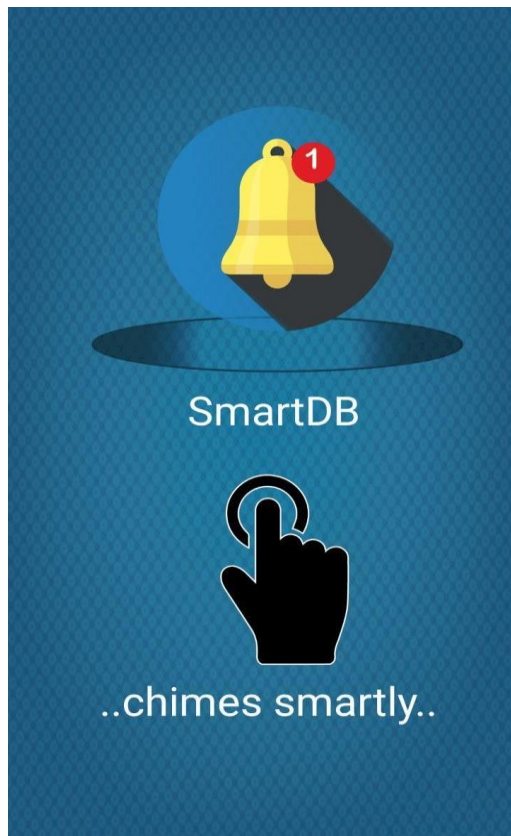


Fig.5.1.1 Splash Screen of SmartDB App.

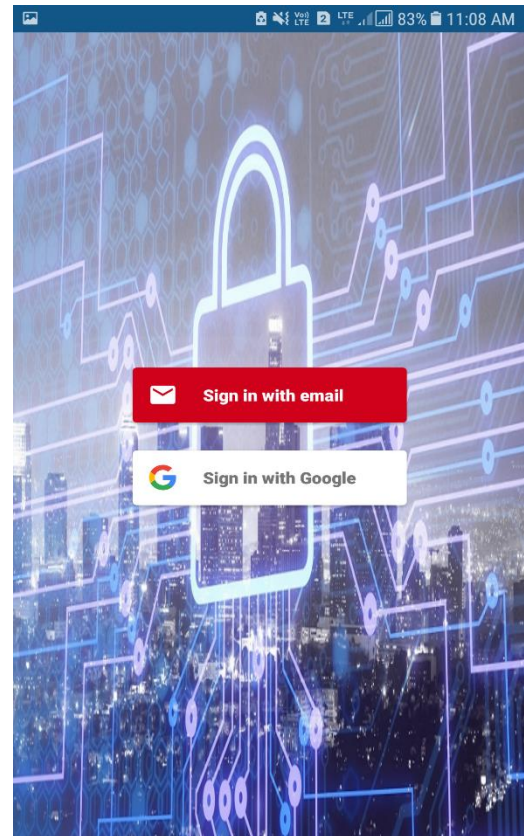


Fig. 5.1.2 Sign Up Screen.

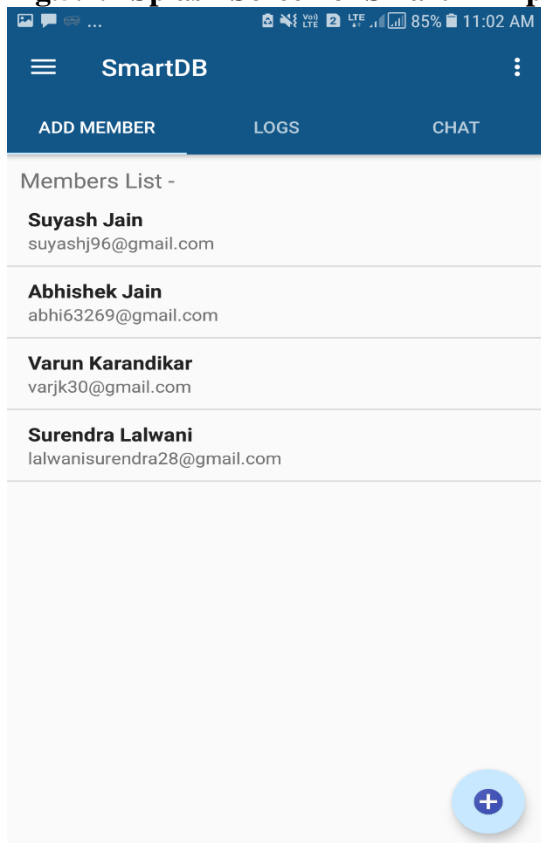


Fig. 5.1.3 Add Member Section for Admin.

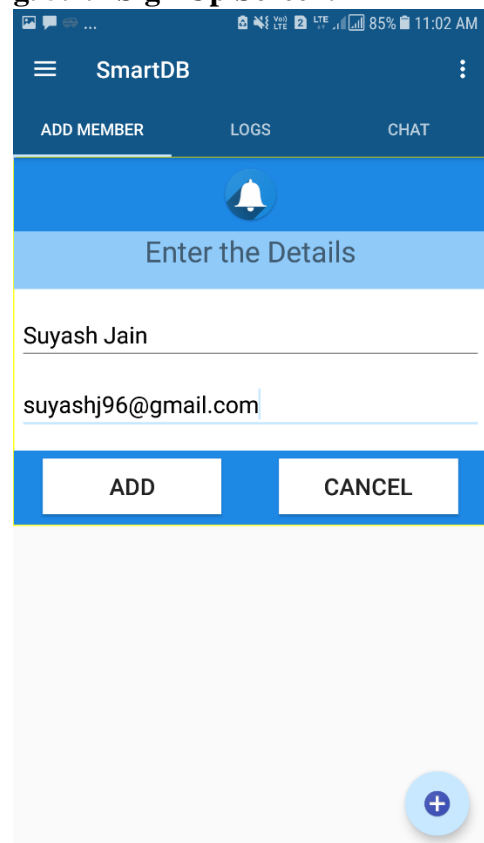


Fig.5.1.4 Admin adding members.

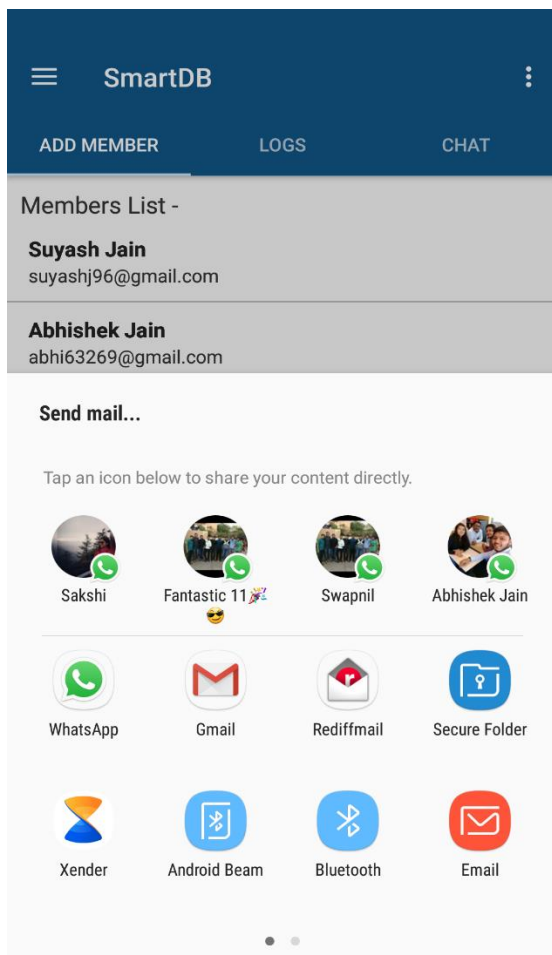


Fig. 5.1.5 Admin shares app with user.

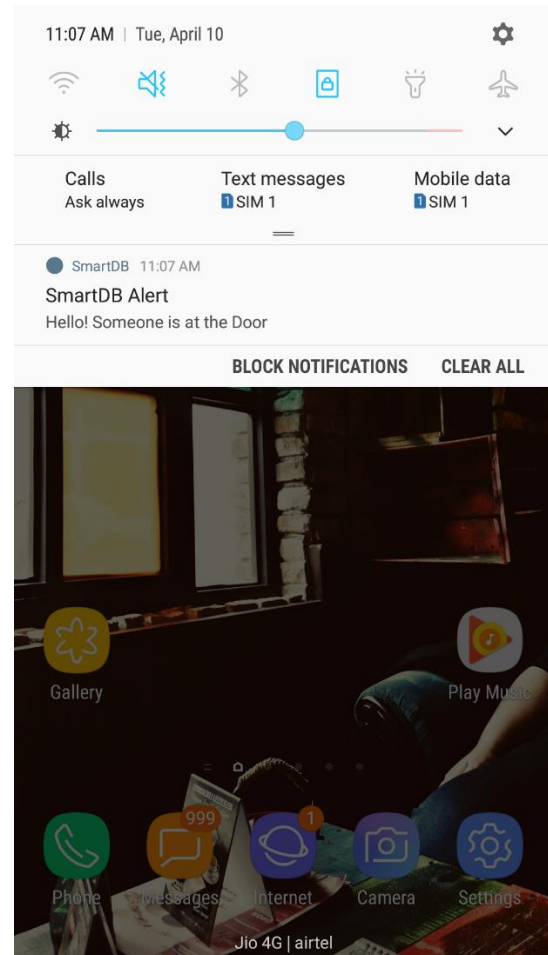


Fig. 5.1.6 Notification on the app arrives.

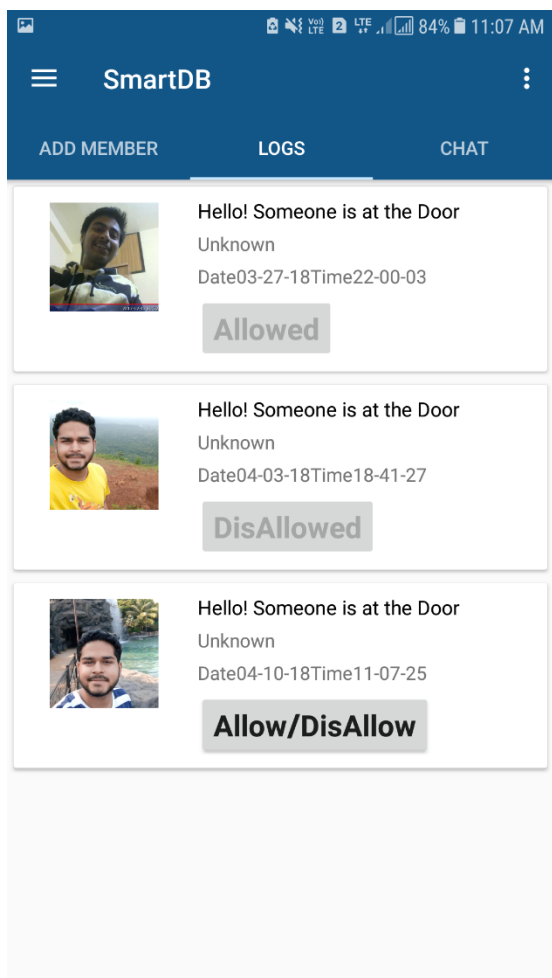


Fig. 5.1.7 Screen after opening notification

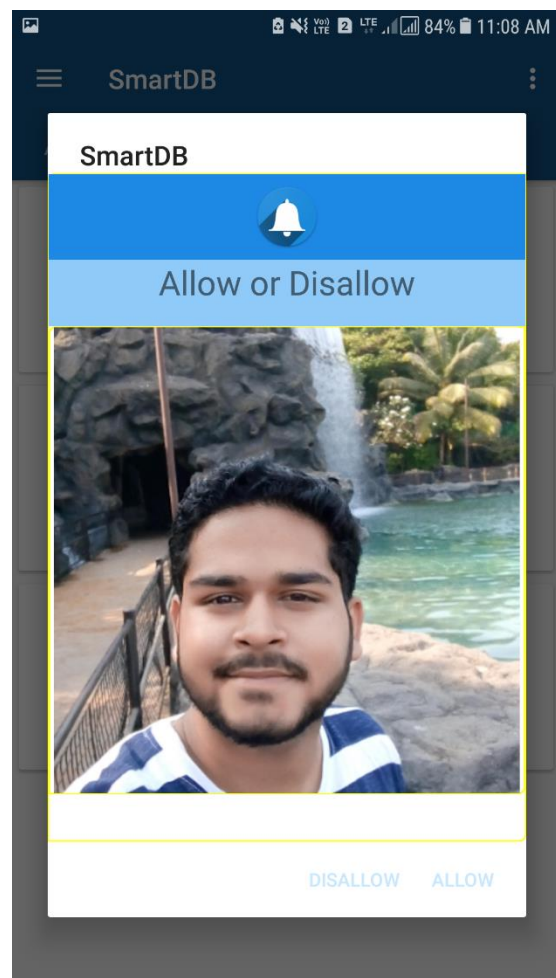


Fig.5.1.8 Screen after clicking button.

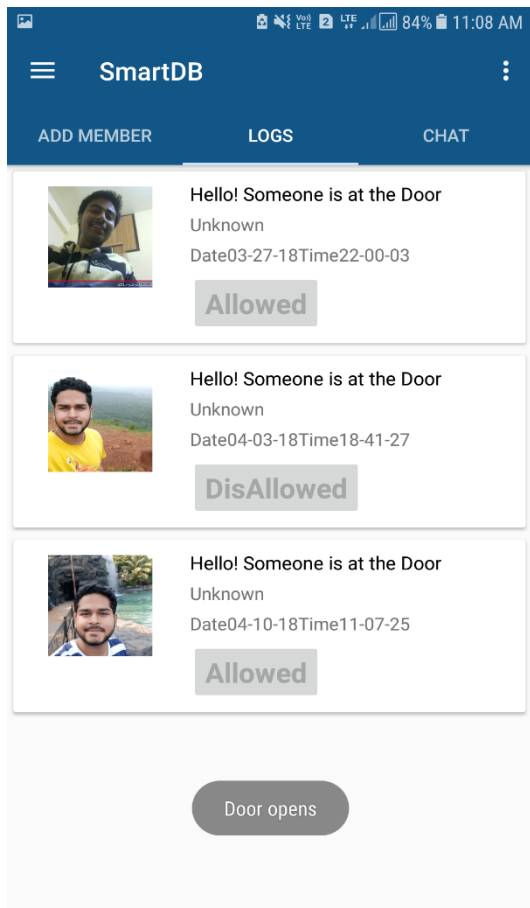


Fig. 5.1.9 Screen after allowing visitor.

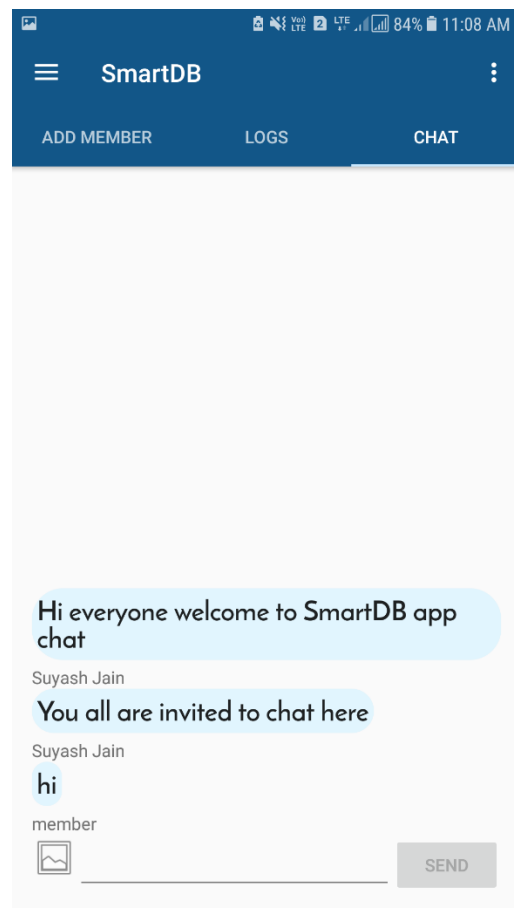


Fig. 5.1.10 Chat section of the application.

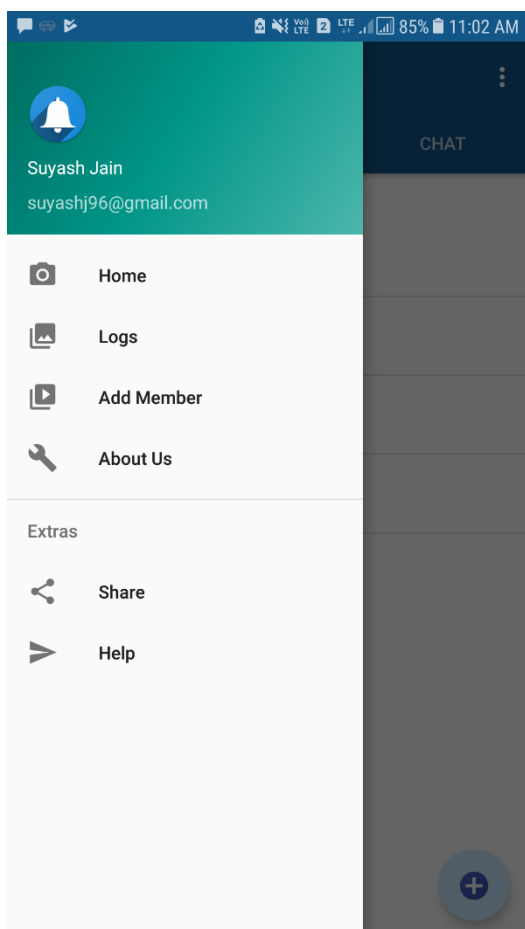


Fig. 5.1.11 Navigation Section of application.

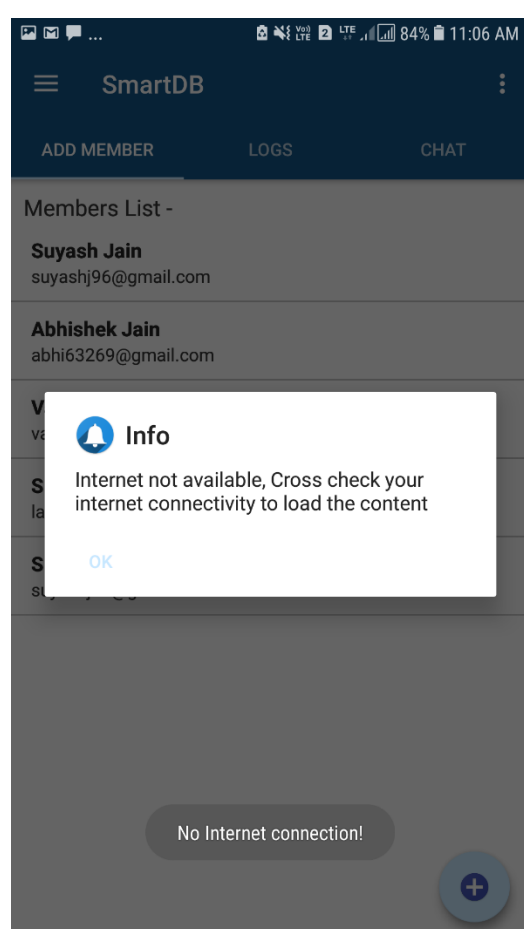


Fig.5.1.12 When no internet connection.

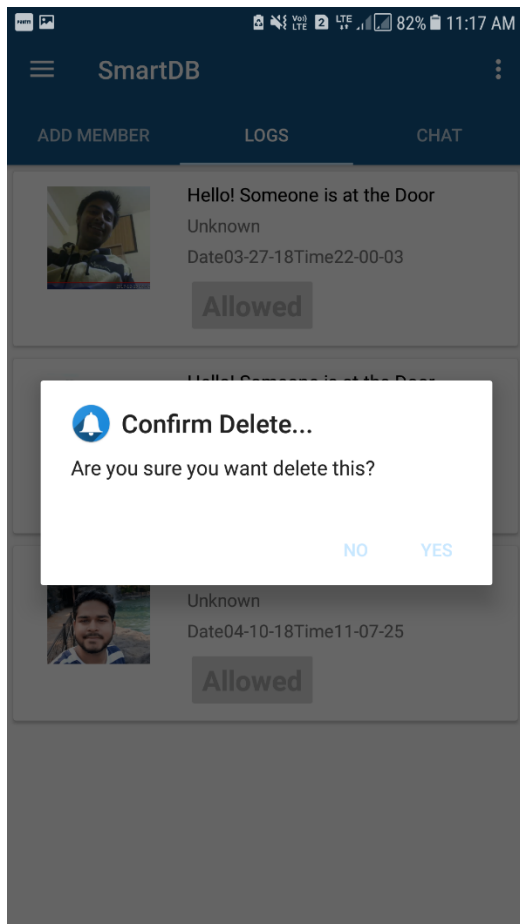


Fig. 5.1.13 Deleting an entry from Logs.

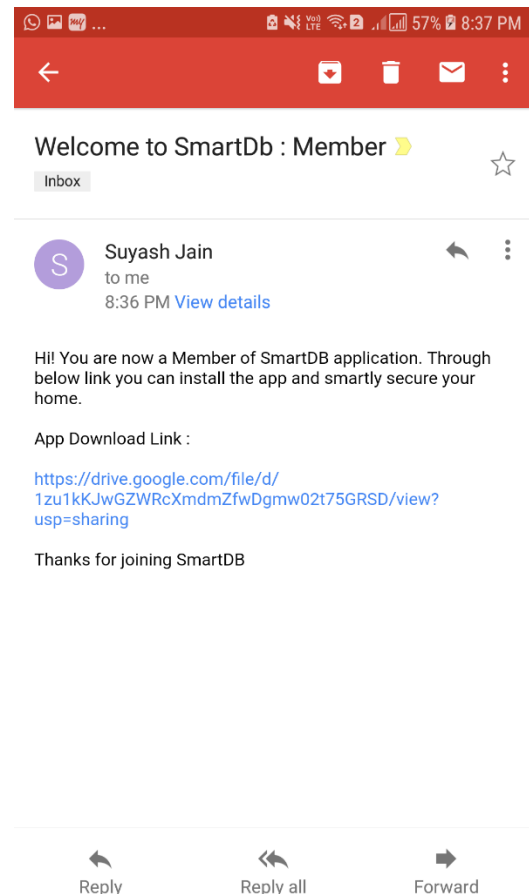


Fig.5.1.14 Users receiving mail from admin.

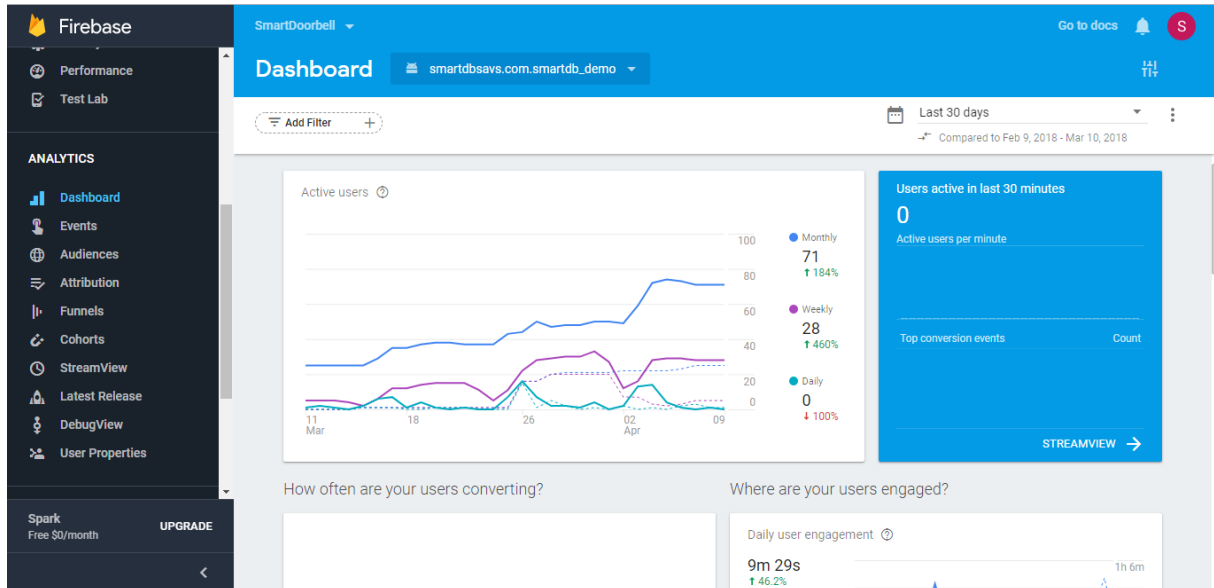


Fig. 5.1.15 Dashboard to the Firebase web and mobile application platform.

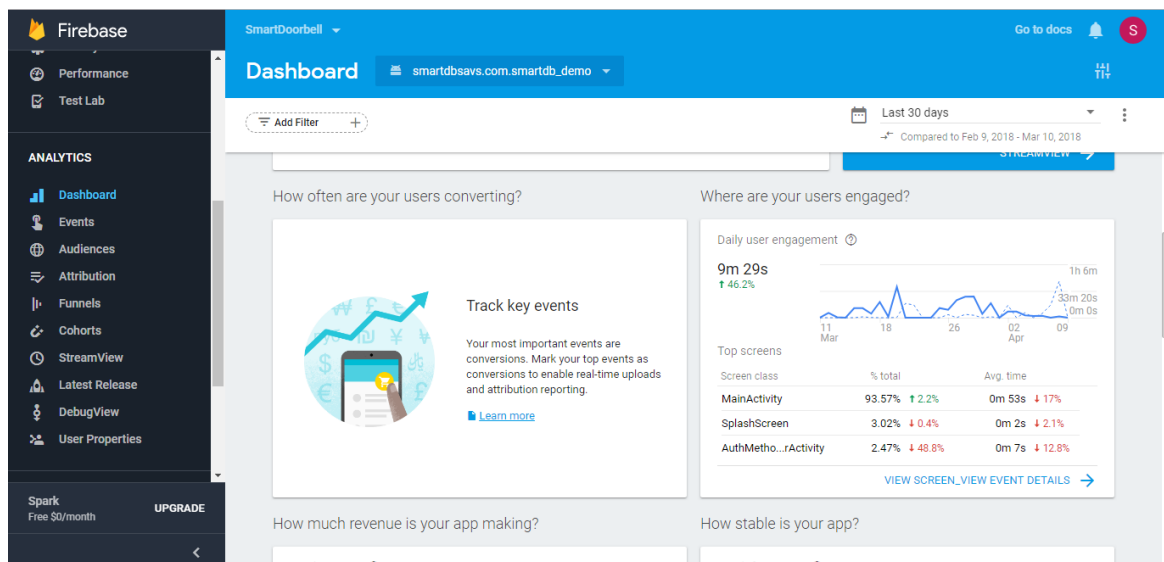


Fig. 5.1.16 Dashboard showing user statistics.

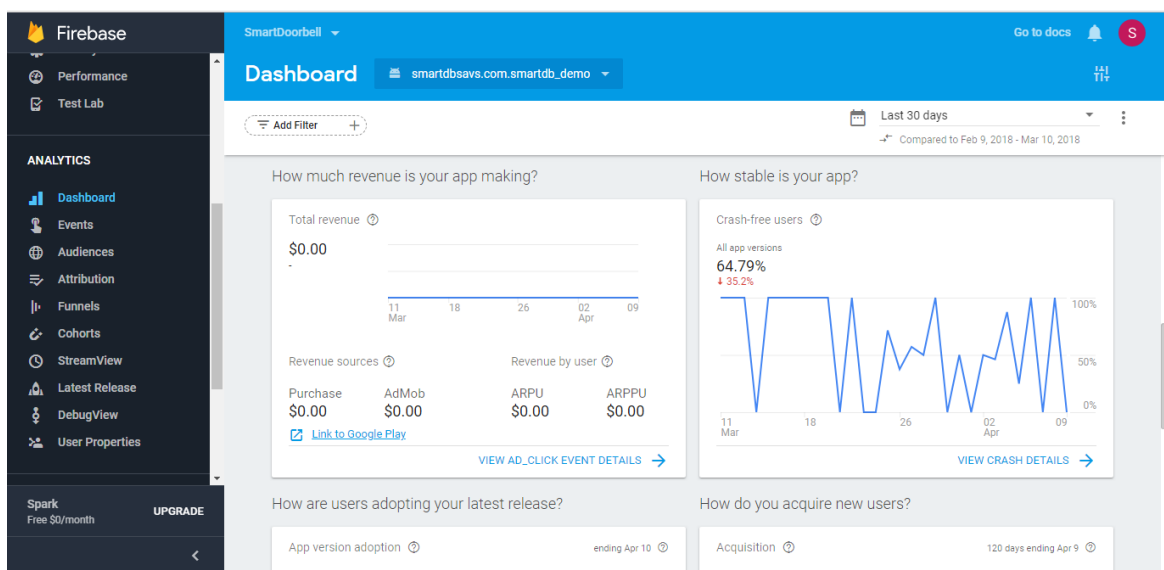


Fig. 5.1.17 Dashboard showing application statistics.

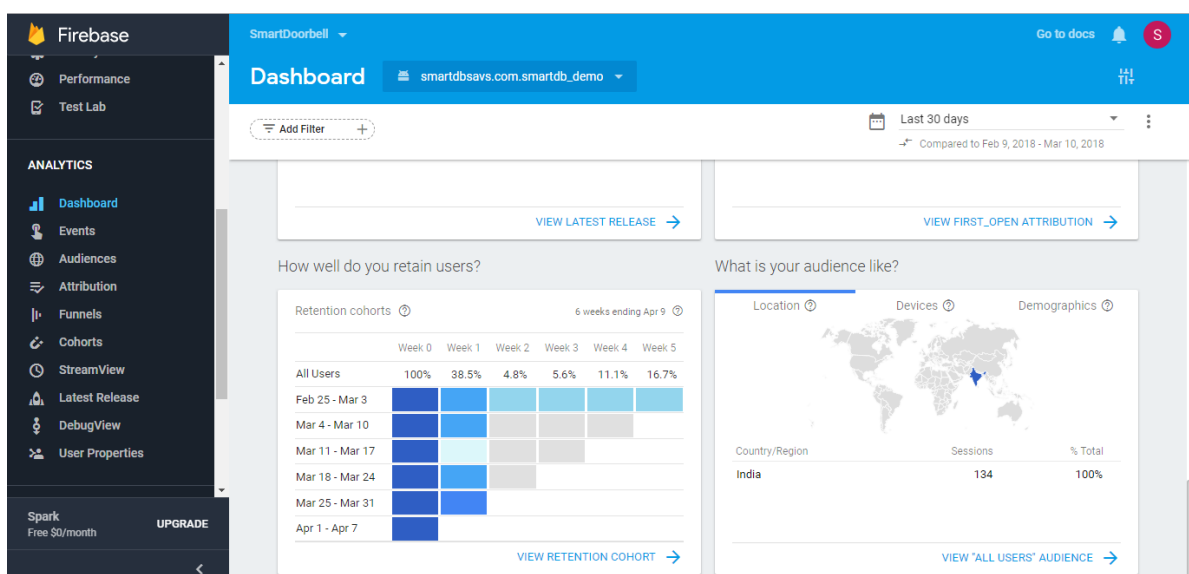


Fig. 5.1.18 Statistics about users and audience.

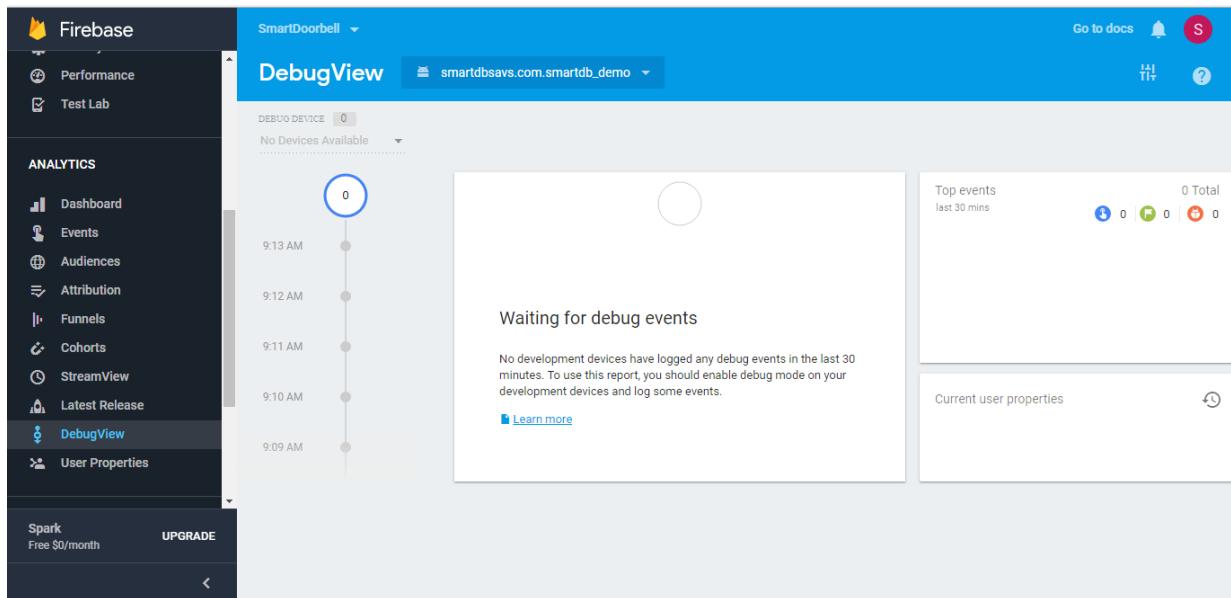


Fig. 5.1.19 Firebase’s debug view for debugging application.

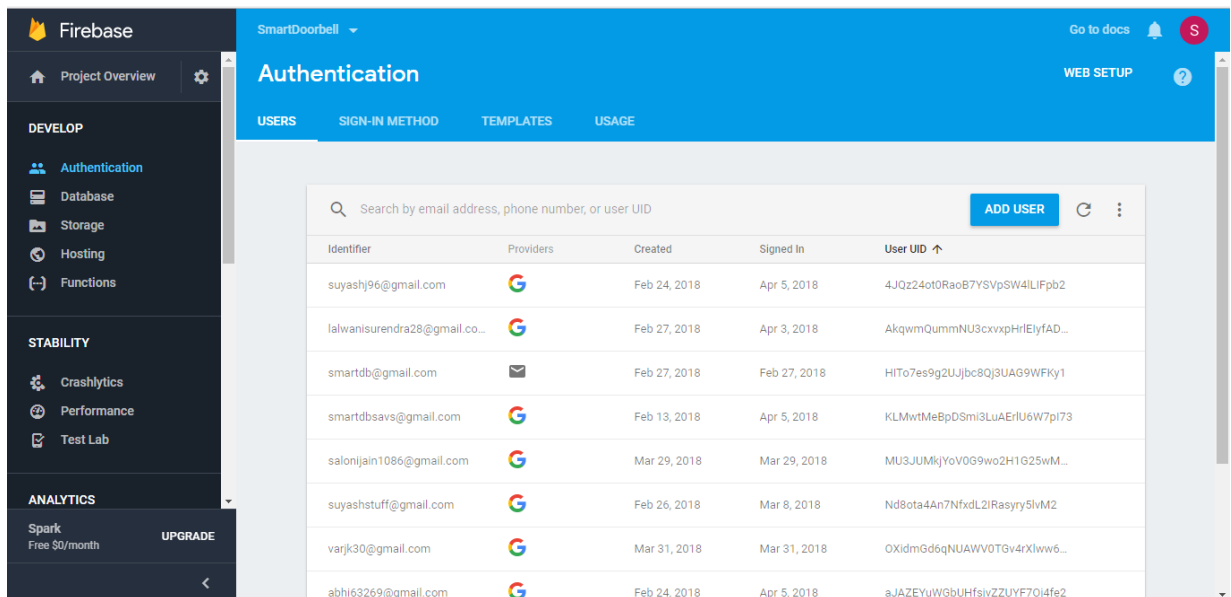


Fig. 5.1.20 Authentication panel on Firebase

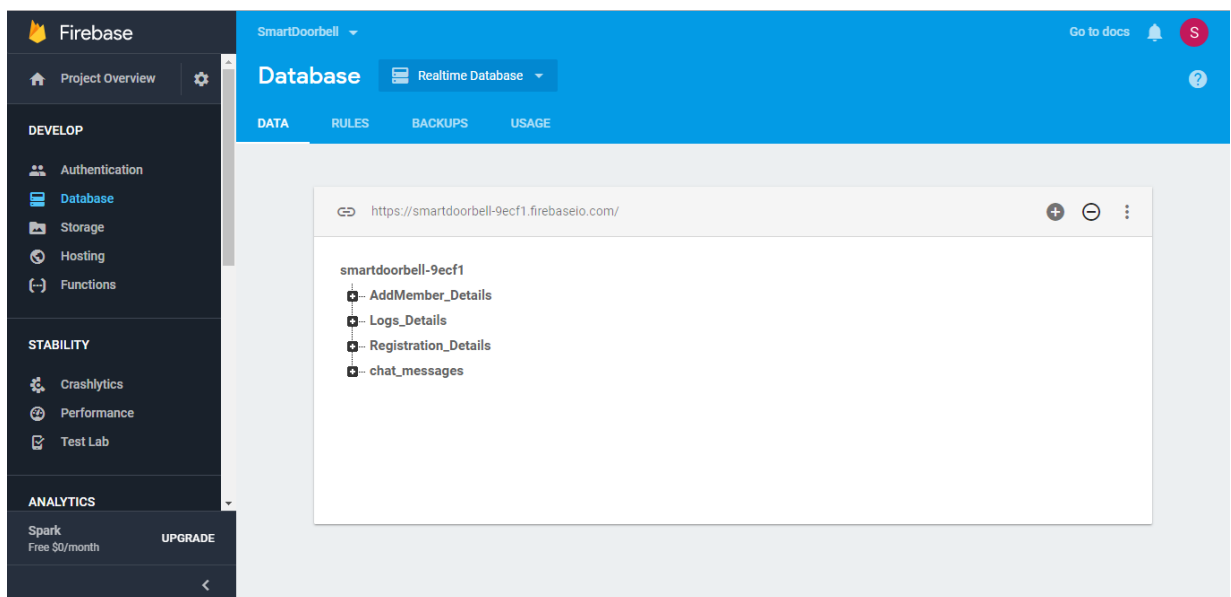


Fig. 5.1.21 Real-time Database on Firebase

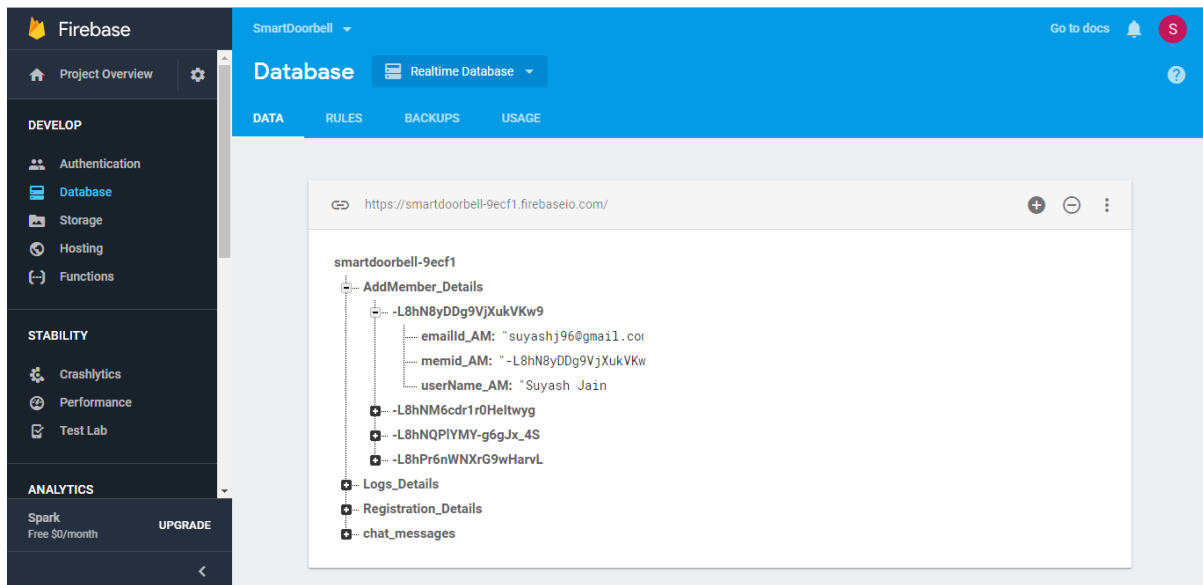


Fig. 5.1.22 AddMember_Details Table Record.

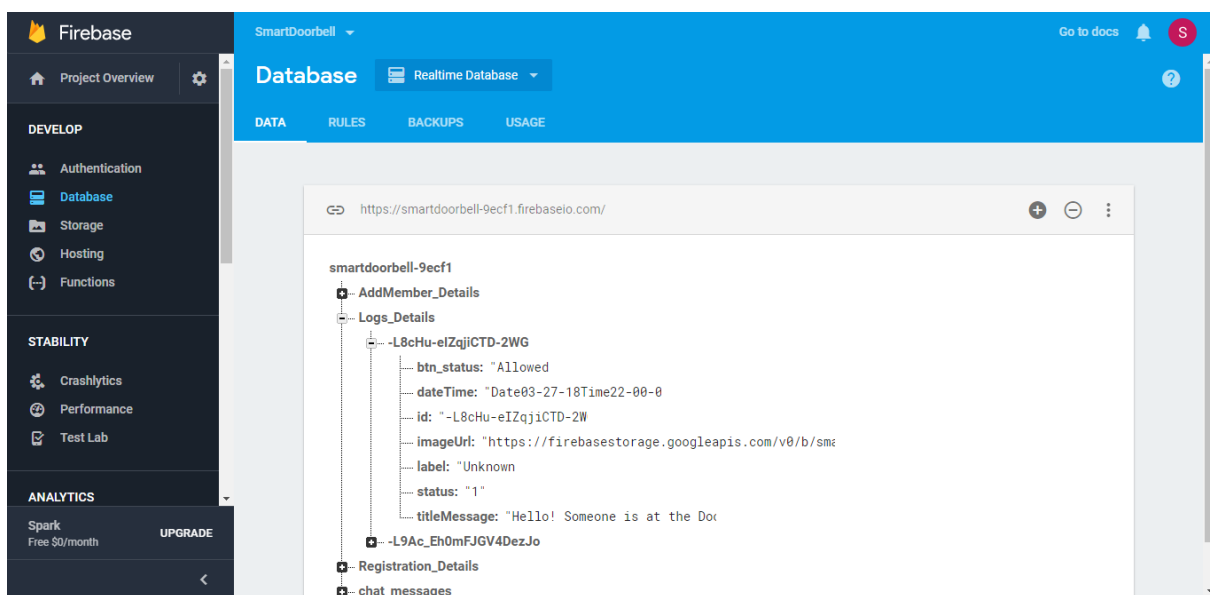


Fig. 5.1.23 Logs_Details Table Record.

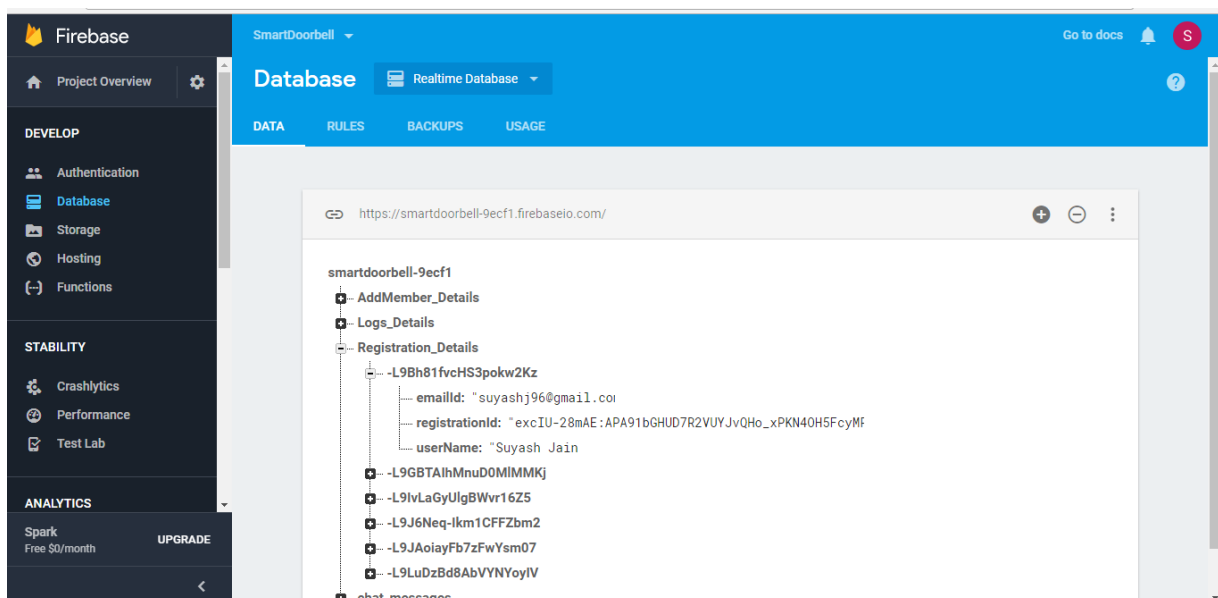


Fig. 5.1.24 Registration_Details Table Record.

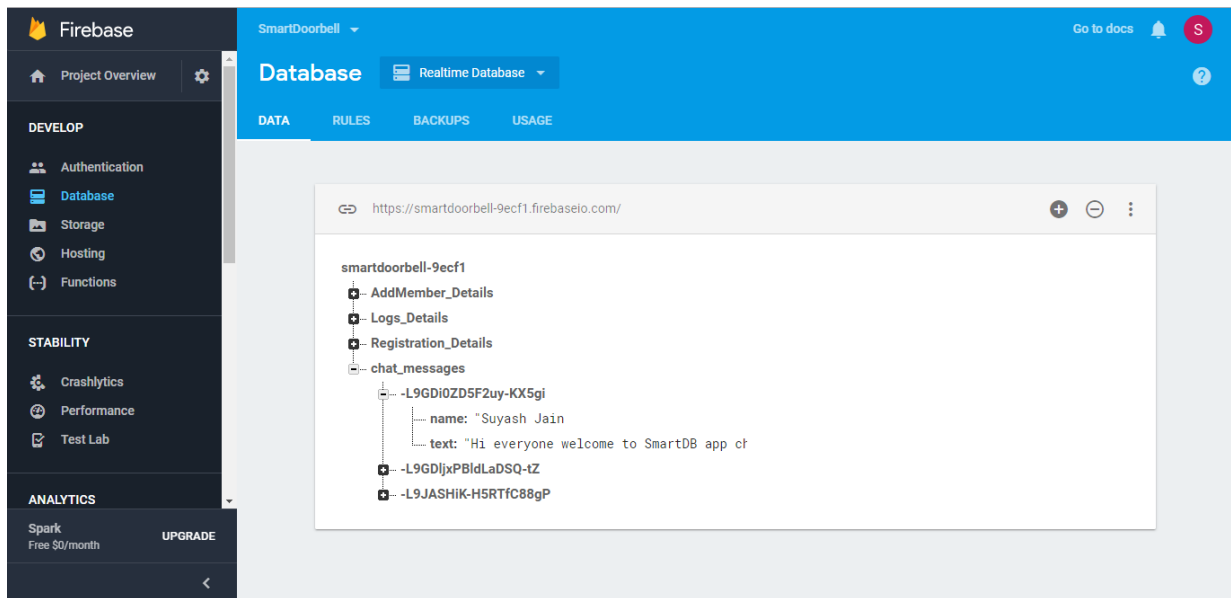


Fig. 5.1.25 chat_messages Table Record.

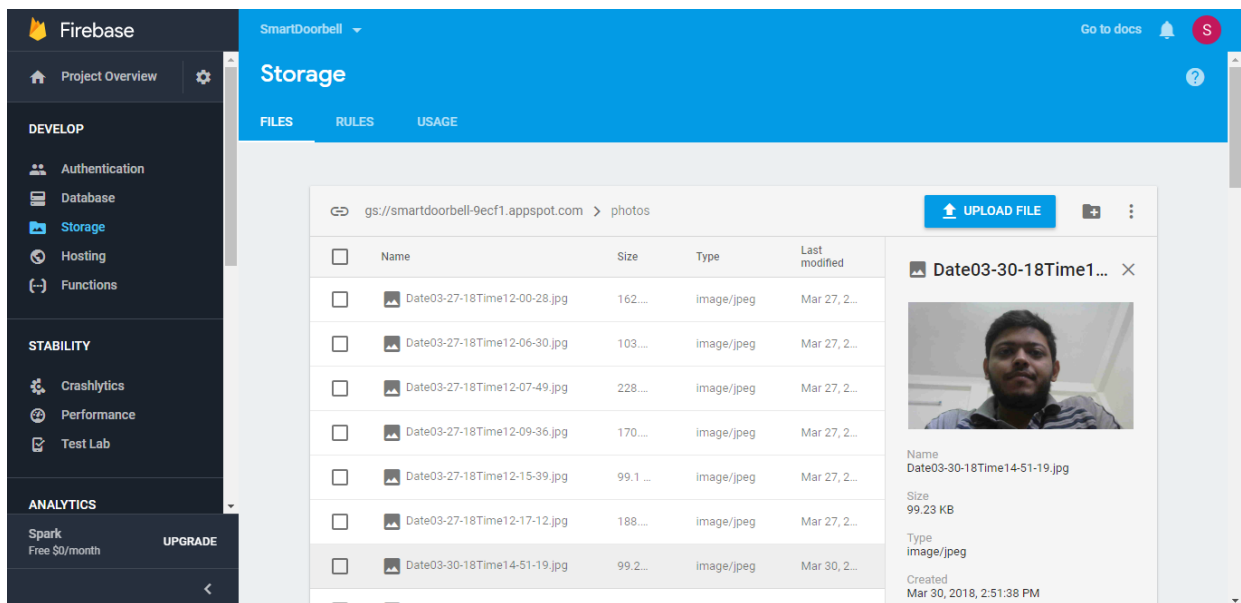


Fig. 5.1.26 Firebase's Storage Service.



Fig. 5.1.27 System Setup along with hardware.

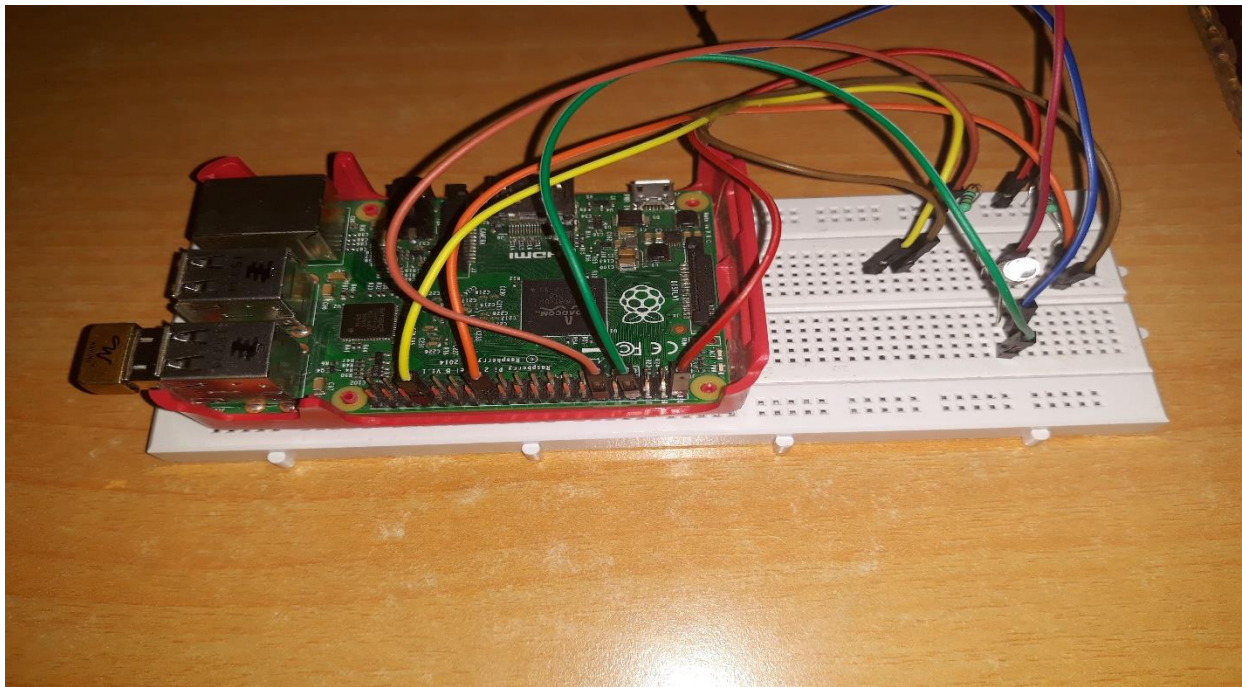


Fig. 5.1.28 SmartDB System to be installed.

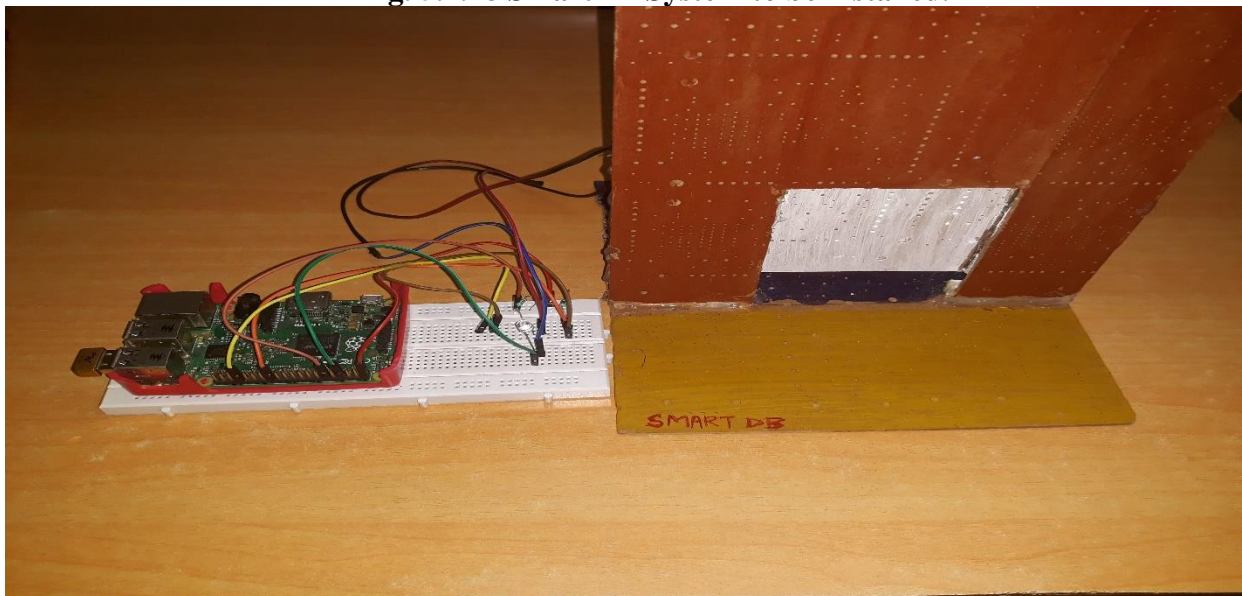


Fig. 5.1.28 SmartDB System along with door prototype.

5.2 Results

The working of SmartDB starts with the pressing of doorbell by the visitor. Once visitor presses the doorbell, the webcam attached with the system installed on the door captures the image of the visitor and uploads it to Firebase storage. At the same time, it triggers a notification to the users's mobile phone through Firebase Cloud Messaging. The system now establishes connection with the server using socket programming which performs Face Recognition through facenet algorithm on the captured image. This results in the generation of label corresponding to the person in the image and same is reflected dynamically on the mobile application with the help of Real Time database. Now the user can perform appropriate action using dialog box on the mobile application resulting in opening or closing of door remotely.

This shows SmartDB system works properly as expected within the acceptable margins of error.

6. Testing

6.1 Testing Methods used

1. **Black Box Testing** - Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications. In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.
2. **White Box Testing** - White Box Testing is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.
3. **Regression Testing** - Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. Regression Testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine. This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

6.2 Test Cases

6.2. TEST CASES				Submission Date: 05/04/18		
BLACK BOX						
Serial No.	Name of Test Case	Input to Test Case	Test Objective	Expected Output	Actual Output	Remarks
1)	Registration	1. abhi632	To check whether user enters valid email or not	Redirected to Login	Email must be in correct format	Failed
		2. @abhi632	To check whether user enters valid email or not	Redirected to Login	Email must be in correct format	Failed
		3. .com@abhi632	To check whether user enters valid email or not	Redirected to Login	Email must be in correct format	Failed
2)	Login	1. abhi632269@gmail.com & abhi123456	User must be registered and login with correct credentials	Redirected to Main Activity	Redirected to Main Activity	Passed
		2. abhi632269@gmail.com & abhi14567	User must be registered and login with correct credentials	Redirected to Dashboard	Incorrect password	Failed
		3. @abhi632269 & abhi123456	User must be registered and login with correct credentials	Redirected to Dashboard	Incorrect password	Failed
3)	Pressing the doorbell	User presses Doorbell	Notify the system users about visitor	User is notified	User is notified	Passed
4)	Image uploaded to firebase	Captured image of visitor	To check whether image is uploaded to firebase storage server or not	Image uploaded	Image uploaded	Passed
5)	Notification contains image of visitor	Image details	To check if notification contains image or not	Received notification contains image	Received notification contains image	Passed
6)	Pressing the AUTHORIZE button	User presses the AUTHORIZE button	To check if authorize button functions properly or not	Door opens remotely	Door opens	Passed
7)	Door action	User presses the AUTHORIZE/UNAUTHORIZE button	To check whether door automatically opens or not.	Door opens remotely	Door does not open	Failed
8)	No internet connectivity	User is not connected to any network.	To check whether android application works without internet or not.	Application displays error message	Application displays error message	Passed

WHITE BOX						
Serial No.	Name of Test Case	Input to Test Case	Test Objective	Expected Output	Actual Output	Remarks
1)	Redundant Registration	Already registered	Already registered user must	Already	Already	Passed
2)	Giving name to image	Generated	To check if proper name is	Proper name is	Proper name is	Failed, possible
3)	Real time value updation of	Changes	To check whether changes to	Changes	Changes	Passed
		Changes performed on one	To check whether changes to records are reflected across	Changes reflected	Changes were not reflected	Failed, possible reason may be
4)	Redundant addition of same	Attempting to add	To check whether multiple	Member already	Member already	Failed, it might
5)	Connection to server	Client providing	To check whether connection	Connection	Connection is lost	Failed
6)	Add Courses	Course Name	To check that admin must	Course updated	Course updated	Passed
7)	Redundant Entries in Log	User views log	To check whether logs	Logs have no	Logs have no	Passed
8)	Capture Image	User presses	To check whether image is	Image is	Image is captured	Passed
9)	Face Recognition	Captured image of	To check whether distorted	Recognition	Recognition fails	Failed

REGRESSION TESTING						
Serial No.	Name of Test Case	Input to Test Case	Test Objective	Expected Output	Actual Output	Remarks
1)	Login	abhi63269@gmail.com and abhi1234567	User must be registered and login with correct credentials	Redirected to Dashboard	Redirected to Dashboard	Passed
2)	Door action	User presses the AUTHORIZE/UNAUTHORIZE button	To check whether door automatically opens or not.	Door opens remotely.	Door opens remotely.	Passed
3)	No internet connectivity	User is not connected to any network.	To check whether android application works without internet or not.	Application displays error message	Application displays error message	Passed
4)	Giving name to image	Generated Timestamp	To check if proper name is assigned to image	Proper name is assigned	Proper name is assigned	Passed
5)	Connection to server	Client providing image	To check whether connection remains active throughout the process	Connection remains active	Connection remains active	Passed
6)	Face Recognition	Captured image of visitor	To check whether distorted image is recognized or not.	Recognition works	Recognition works	Passed

7. Conclusion and Future work

IoT aims to connect all potential objects to interact each other on the internet to provide secure, comfort life for human. SmartDB is based on state of the art IOT technology and uses Android Things as the base platform for the development of our product. The idea of our product is to provide secure access to home. It will be achieved through Smart Doorbell.

It's not all about speed—you'll also be able to get more done in less time. In addition to instant gratification, technological evolution also tends to favor productivity and efficiency. Our product will allow you to accomplish remote tasks faster and with greater precision, including data analysis and management.

Future Scope

SmartDB can be extended as we can make the opening of the door automatic and also providing the smart lock with it and also instead of image we can store videos which can even be retrieved at latter times. We will be including sensors such as night vision to increase security. Also we will adding voice/video call capability in our product so we can talk to visitor over the microphone and speaker attached at door and we will be adding sensor to get alert when someone try to break into our house. It can also interact with other IoT appliance inside home to pretend someone is here!

8. References

- Developer Android Things Platform
<https://developer.android.com/things/hardware/index.html>
- Android API guide - <https://developer.android.com/reference/packages.html>
- Lynda - <https://www.lynda.com/in/it2>
- Stack Overflow - <https://stackoverflow.com/>
- Quora - <https://www.quora.com/>
- Raspberry Pi official website - <https://www.raspberrypi.org/forums/>
- FaceNet: A Unified Embedding for Face Recognition and Clustering.
<https://arxiv.org/abs/1503.03832>
- Building a Facial Recognition Pipeline with Deep Learning in Tensorflow.
<https://hackernoon.com/building-a-facial-recognition-pipeline-with-deep-learning-in-tensorflow-66e7645015b8>
- Python Server /Client Socket Programming : <https://docs.python.org/3.7/howto/sockets.html>
- Firebase in Android : <https://firebase.google.com/docs/>
- Pyrebase : <https://github.com/thisbejim/Pyrebase>
- Webcam configuration : <https://www.raspberrypi.org/documentation/usage/webcams/>
- Putty Configuration : <https://www.raspberrypi.org/documentation/remote-access/ssh/windows.md>
- RaspberryPi setup : <http://www.circuitbasics.com/how-to-connect-to-a-raspberry-pi-directly-with-an-ethernet-cable/>
- Android App Development : <https://developer.android.com/index.html>