Project Report On

Name of the System - REIG

by

1. Alkesh Kataria (Enrollment No:16102003)

2. Bhargav Borse (Enrollment No:16102023)

Under Guidance

of

Mr. Dhaval Mehta and Ms. Hemal Shah

Submitted to Bachelor of Computer Applications Program

in partial fulfillment of the requirements

for the award of degree

Bachelor of Computer Applications

Navrachana University

Vadodara

Nov 2018

Navrachana University
Vadodara

CERTIFICATE OF COMPLETION

This is to certify that report submitted along with the project entitled **REIG** has been carried out by Alkesh Kataria (16102003) & Bhargav Borse (16102023) under our guidance for partial fulfillment of the degree Bachelor of Computer Application from Navrachana University, Vadodara during the academic year 2018-19. This/These students have successfully completed project under our guidance.

Hemal Shah
(Internal Reviewer)
Date:

Mr.Dhaval Mehta
(Program Coordinator)
Date:

Table of Contents

Acknowledgement

The in-house project opportunity we had at Navrachana University was a great chance for learning and skill development. We consider ourselves very lucky individuals as we were provided with an opportunity to be a part of it. We are also grateful for having a chance to meet so many wonderful people and professionals who led me though us through the period.

We are using this opportunity to express our deepest gratitude and special thanks to everyone mentioned below and unmentioned, who helped us in taking part in useful decisions & giving necessary advices and guidance and arranging all facilities to make this project easier and a success. We take this moment to acknowledge their contribution gratefully.

We would like to thank all the people that have helped us to build this project. A special thanks to Ms. Hemal Shah and Mr. Dhaval Mehta to help us overcome any difficulties that came across and for guiding us throughout the project. We would like to thank Navrachana University for the opportunity and support provided. We would also like to thank all faculties for solving all

queries.

We perceive this opportunity as a big milestone in our career development. We will strive to use gained skills and knowledge in the best possible way, and we will continue to work on our improvement, in order to attain desired career objectives.

Abstract

The REIG is a Reminder-based web application which helps people to set reminders, share notes either individually or in a group. REIG would also allow groups to be created and members can chat, share notes, add reminders and events, add photos for that group.

REIG is developed as a website but it has the potential to behave as a Native Application. REIG can do both at the same time. REIG is also a real time application, the changes are reflected automatically without requirement of reloading the pages.

List of Tables

List of Figures

1. Introduction

1.1 Business Problem

Maintaining notes, sharing location, setting up reminders such features are never integrated in one app. To go with any of the one feature there's a different app. There are bunch of apps for using any one of the features. However, there is no app that can offer the functionality of the trio.

1.2 Project Summary

This project revolves around a website that in turn can convert itself to application. Such is the technology of PWA. It is based on PWA. Progressive Web App (PWA) is a term used to denote a new software development methodology. Unlike traditional applications, progressive web apps are a hybrid of regular web pages (or websites) and a mobile application. This application also covers the use of Real-Time Database. A real-time database is a database system which uses real-time processing to handle workloads whose state is constantly changing. This differs from traditional databases containing persistent data, mostly unaffected by time. For example, a stock market changes very rapidly and is dynamic.

With the use of PWA and Real-Time Database, this project will eventually help user to create reminders, events, store images, create groups, chat in groups, set reminders and events in groups, share images in groups too. PWA will convert that website into an application whenever user wishes to use it as an application and Real-Time Database will lead to reflect changes real-time eliminating the need to reloading the pages again and again.

1.3 Purpose

The purpose of this project is to give a detailed description of the working for the REIG application. The basic purpose of the software is to set reminders and events for an individual/group either in application or in website respective to their registered accounts. In group one can share images, location, notes, reminders or event. One can set reminders for special days of their loved ones or can add important dates related to health, functions, meetings and events. Addition to this, event can also be added or created on this application and website. Furthermore, the reminders as well as events can be saved on the same dates if required.

1.4 Scope

The REIG is a Reminder-based web application which helps people to set reminders, share notes either individually or in a group. The application should be free to access from either a mobile phone application point of view or similar. Users can provide their information using the web-portal. This information will act as the bases for the user to access the services. An administrator also uses the web-portal in order to administer the group and manage the same. The administrator can, for instance, verify members and manage user information. Furthermore, the software needs Internet connection to provide services. The software also needs GPS and Camera access for clicking pictures or sharing location accordingly. All system information is maintained in a database, which is located on a web-server.

2. System Requirements Study

2.1 User Characteristics

User/Actor	Characteristics	Constraint
Admin	○ Create group ○ View group activities ○ Set reminder or event for the group. ○ Share notes ○ Add or removal of members ○ Change group details ○ Set reminder or event for himself. ○ Save notes for himself.	
Group Member	○ View group activities ○ Set reminder or event for the group. ○ Share notes ○ Set reminder or event for himself. ○ Save notes for himself.	
Individual	○ View own activities ○ Set reminder or event ○ Share notes	Not in group.

Table 1 : User Characteristics

2.2 Hardware & Software Requirements

Development Requirements:

Tools	:	Visual Studio Code
Processor	:	*Computer:* Intel Core i5 – i7 recommended *Phone:* Qualcomm Snapdragon 450 – 845 recommended.
Hardware	:	32/64-bit machine
OS	:	*Computer:* Windows 7 or above *Phone:* Android 6.0.1 or above

RAM	:	*Computer:* 4gb minimum – 8gb recommended *Phone:* 3gb minimum – 6gb recommended.
Web Server	:	Xampp
Firebase Version	:	5.5.1

Table 2 : Development Requirements

Client – End Requirements:

Processor Windows	:	Minimum x64 Processor 1.4GHz
Processor Mobile	:	Minimum x64 Processor 1.2GHz
Min. RAM	:	*Computer:* 1gb of RAM *Mobile:* 1gb of RAM
Min. Disk Space	:	100 MB
Windows	:	Windows XP SP2
Android	:	6.0.1
OS X	:	OS X 10.5.6
Linux	:	Ubuntu 10.04
		Debian 6
		Fedora Linux 14
System Connected	:	To Internet

Table 3 : Client – End Requirements

2.3 Software Development Life Cycle

The Software Development Lifecycle is a systematic process for building software that ensures the quality and correctness of the software built. SDLC process aims to produce high-quality software which meets customer expectations. The software development should be complete in the pre-defined time frame and cost.

SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software. Every phase of the SDLC lifecycle has its own process and deliverables.

Here, are prime reasons why SDLC is important for developing a software system.

- It offers a basis for project planning, scheduling, and estimating
- Provides a framework for a standard set of activities and deliverables
- It is a mechanism for project tracking and control

- Increases visibility of project planning to all involved stakeholders of the development process
- Increased and enhance development speed
- Improved client relations
- Helps you to decrease project risk and project management plan overhead.

SDLC Phases

The entire SDLC process divided into the following stages:

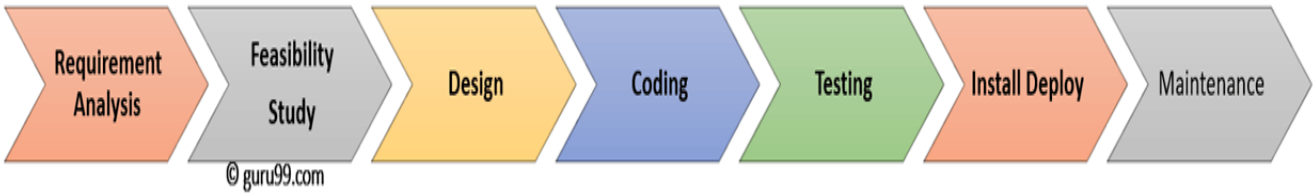

Figure 1 : SDLC Phases

- Phase 1: Requirement collection and analysis
- Phase 2: Feasibility study:
- Phase 3: Design:
- Phase 4: Coding:
- Phase 5: Testing:
- Phase 6: Installation/Deployment:
- Phase 7: Maintenance:

Phase 1: Requirement collection and analysis:

The requirement is the first stage in the SDLC process. It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and recognition of the risks involved is also done at this stage.

This stage gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives which triggered the project.

Requirements Gathering stage need teams to get detailed and precise requirements. This helps companies to finalize the necessary timeline to finish the work of that system.

We carried out this phase by asking the college students about what sort of different problems they face while chatting or setting reminders and events or while uploading photos. The responses led us towards creating an app that could erase such problems and can improve the in hand functionality. We even get to know that these features to be integrated in one app can fall on the beneficiary side for the students and normal users. Today's apps provide such functionality but aren't integrated into one. Our app would focus on integrating all of that into one. User doesn't need to switch apps for carrying out above said tasks. Rather one app would do it all.

Phase 2: Feasibility study:

Once the requirement analysis phase is completed the next step is to define and document software needs. This process conducted with the help of 'Software Requirement Specification' document also known as 'SRS' document. It includes everything which should be designed and developed during the project life cycle.

According to the feasibility study, we were able to prepare SRS document. A **software requirements specification** (**SRS**) is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

Phase 3: Design:

In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.

This design phase serves as input for the next phase of the model.

Phase 4: Coding:

Once the system design phase is over, the next phase is coding. In this phase, developers start build the entire system by writing code using the chosen programming language. In the coding phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process.

In this phase, Developer needs to follow certain predefined coding guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code.

Phase 5: Testing:

Once the software is complete, and it is deployed in the testing environment. The testing team starts testing the functionality of the entire system. This is done to verify that the entire application works according to the customer requirement.

During this phase, testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and send back to testing team for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system.

Phase 6: Installation/Deployment:

Once the software testing phase is over and no bugs or errors left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final software is released and checked for deployment issues if any.

Phase 7: Maintenance:

Once the system is deployed, and users start using the developed system, following 3 activities occur

- Bug fixing - bugs are reported because of some scenarios which are not tested at all
- Upgrade - Upgrading the application to the newer versions of the Software
- Enhancement - Adding some new features into the existing software

The main focus of this SDLC phase is to ensure that needs continue to be met and that the system continues to perform as per the specification mentioned in the first phase.

Here, are some most important phases of SDLC Life Cycle:

- Waterfall Model
    - In this approach, the whole process of the software development is divided into various phases.
- Incremental Approach
    - In this approach, the requirements are divided into groups at the start of the project. For each group, the SDLC model is followed to develop software.
    - The SDLC process is repeated, with each release adding more functionality until all requirements are met.
- V-Model
    - In this approach, the phase is planned in parallel. So, there are verification phases on the side and the validation phase on the other side.
    - V-Model joins by Coding phase.
- Agile Model
    - In this approach, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks.
- Spiral Model
    - In this approach, this model adopts the best features of the prototyping model and the waterfall model.
    - The spiral methodology is a combination of rapid prototyping and concurrency in design and development activities.
- Big Bang Model
    - Big bang model is focusing on all types of resources in software development and coding, with no or very little planning. The requirements are understood and implemented when they come. (SDLC, 2008)

2.4 Process Model

The process model used is called the Waterfall Model. The Waterfall Model was the first Process Model to be introduced. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Waterfall Model - Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.

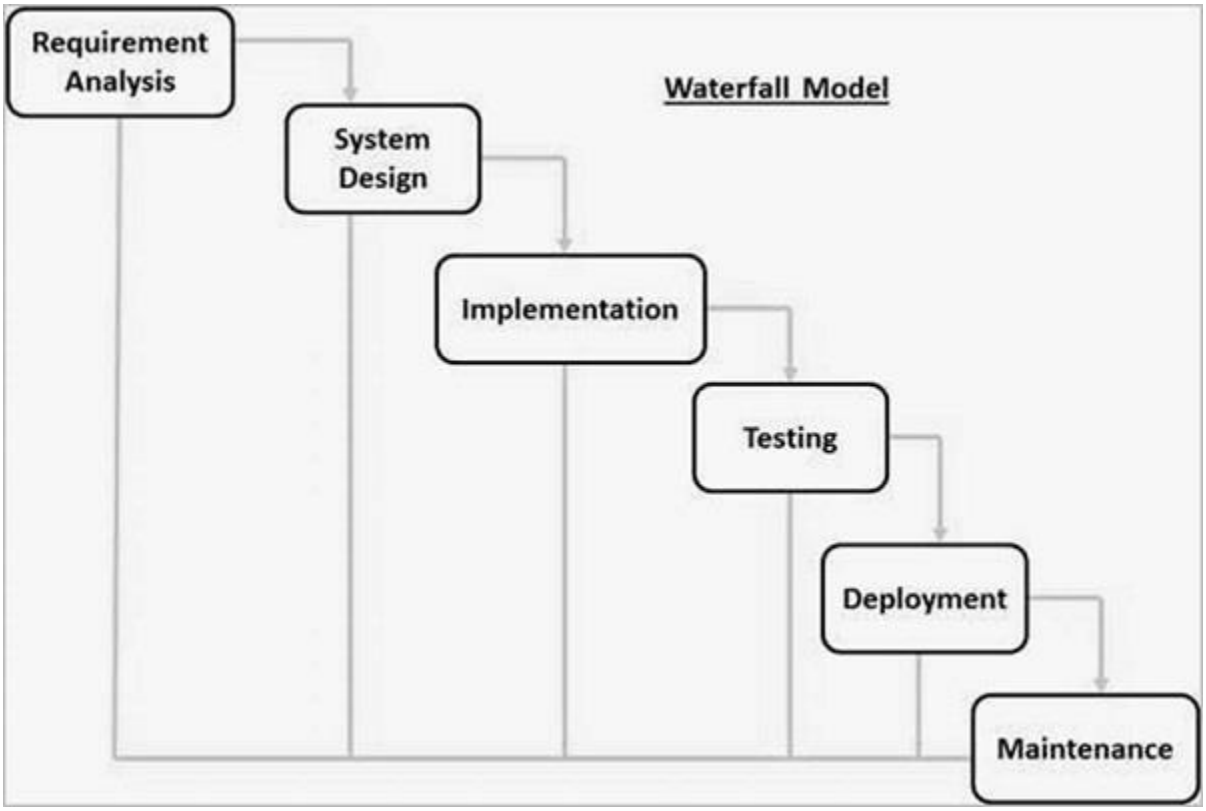

Figure 2 : Waterfall Model Phases

The sequential phases in Waterfall model are −

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows −

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model.
- Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows −

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.

- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early. (WaterFall, 2014)

3. System Analysis

3.1 Study of Current System

We have seen many applications with nice features. For example, one of the application is providing a feature for setting up reminder in which we can add something which we don't want to forget. By adding reminder, we can save a date for reminding us things. Every day, we have many chores to do, doing housework, business work, meetings, birthday parties, phone calls, billings, TV programs and so on. it is not easy to memorize all and handle all well, in order not to forget or miss any important things, we can ask help from some reminder application, and it does be workable and helpful. Reminder software is a simple event management program that can schedule our daily plans, reminder with pop-up window and sound to ensure that you don't miss an important appointment or deadline. If you spend a lot of time on your computer, it makes sense to keep your schedule or calendar there, too, since that's where you're most likely to see it.

Then there are some application providing feature to save images for their own purpose. This type of applications is of many types like vault, gallery, etc. We can safely store our images in this type of application. Security is an important issue to all of us today, and we are all realizing the value of our personal data. Images in particular are deemed to be important – we have a desire to protect these over most other types of data. And to address that need comes images storing apps. Whether we want to keep corporate, notes or different images safe, protect photos of ourselves from others with the access to our device, or keep a personal collection of pictures to ourselves, there are these apps available to keep them safe.

There is some application where user can chat with people. Chat apps have brought the world closer. They have connected you to your long lost school mates and have helped you make new acquaintances. These anywhere, anytime advantage has made these apps a popular medium of communication for businesses too. Some businesses specially request a chat app for internal business communications. Social chatting applications bring the world together and help you reach out to likeminded people. Instant messengers have become a way of life. They seem to be a more reliable form of communication than a phone call. Today's advanced chat apps also have the facility to check whether the recipient has received and actually read your message. You can catch-up with old friends and new, family and relatives – the whole world comes together on one virtual platform.

3.2 Problem and Weaknesses of Current System

There are many apps which provide interesting and useful features. Today, the availability of application is on the increase such that it is produce a noticeable change in the way humans feel and experience computing. On a general note, the presence of mobile apps on

Phones can be likened to be the decorations on a cake because they make mobile phones enjoyable and fun. Mobile apps are unlimited in number with usage that cuts across all walks of life and with people wanting more and more of these apps for easy lifestyle and living. Presently, the use of mobile apps can be seen in areas such as communication, education, cooking, social media, shopping, business (money making), matrimony, and banking. On daily basis individuals seek for updated versions of these apps.

But in all these things we use many applications to use different and useful features. Like for example one app who gives facility to chat with people. Chatting applications bring the world together and help us reach out to likeminded people. But if user wants to talk in group with friends then some application doesn't have feature of creating group if some application provides feature to chat in group then that app don't display groups in different module. All chats are displayed in single module. User would be confused if all chats are displayed in on module. Another weakness of these type of apps is that anyone can change your created group's profile picture, we can't do anything, we don't have any option to give permission.

If a user is using a reminder app for remembering the important tasks, then there is also some problem in some apps that users are facing. Problems like while user save or add reminder their installed app is not able to sync for a long time. Users have to uninstall and again reinstall the app several times to keep it syncing.

User is using a app which stores image but takes so much space in phone and whenever we store our important images in app the size of application is also increased. Sometimes we lost our saved images from that app.

These were some problems and weakness of current system.

3.3 Requirements of New System

The above said problems can prove to be requirements of the new system. New system should comprise of the following requirements:

- User should be able to add reminder and event.
- User should be able to add photos with caption to create memories.
- User should be able to create group.
- User/Member should be able to chat.
- User/Member should be able to add reminder and event in groups.
- User/Member should be able to add photos with caption in groups.

All these requirements should be integrated into one app.

3.4 Feasibility Study

A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained.

A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based.

3.5 Requirements Validation

Validation is the process of confirming the completeness and correctness of requirements. Validation also ensures that the requirements:

1. Achieve stated business objectives
2. Meet the needs of stakeholders
3. The needs are clear and understood by the developers.

Validation is essential to identification of missing requirements and to ensure that the requirements meet certain quality characteristics. Validation addresses each individual requirement to ensure that it is:

- Correct – accurately states a customer or external need
- Clear – has only one possible meaning
- Feasible – can be implemented within known constraints
- Modifiable – can be easily changed, with history, when necessary
- Necessary – documents something customers really need
- Prioritized – ranked as to importance of inclusion in product
- Traceable – can be linked to system requirements, and to designs, code, and tests
- Verifiable – correct implementation can be determined by testing, inspection, analysis, or demonstration

3.6 Functions of System

Use Case System

Figure 3 : Use Case Diagram

3.7 Data Modelling

E-R Diagram

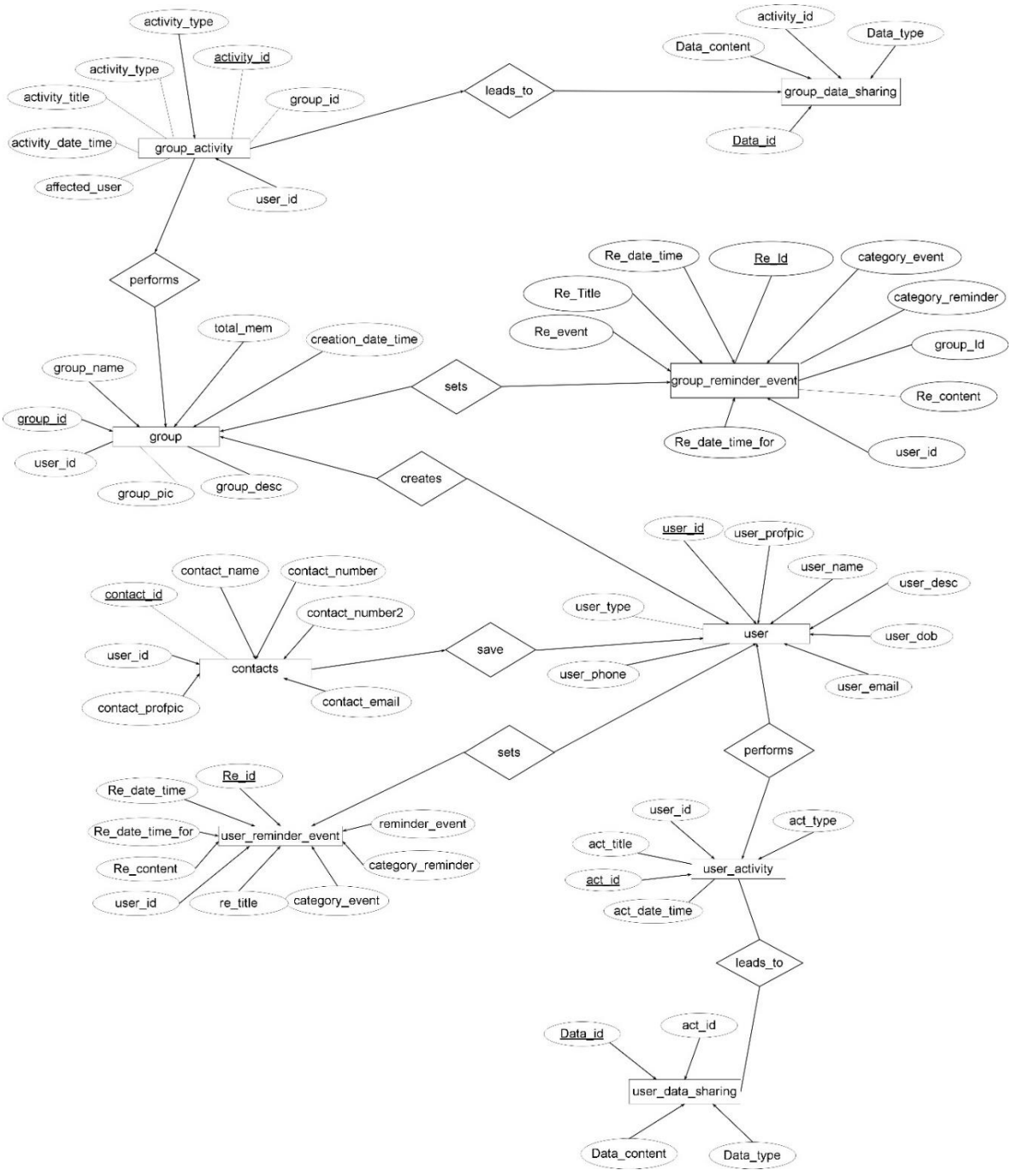

Figure 4 : E-R Diagram

Class Diagram

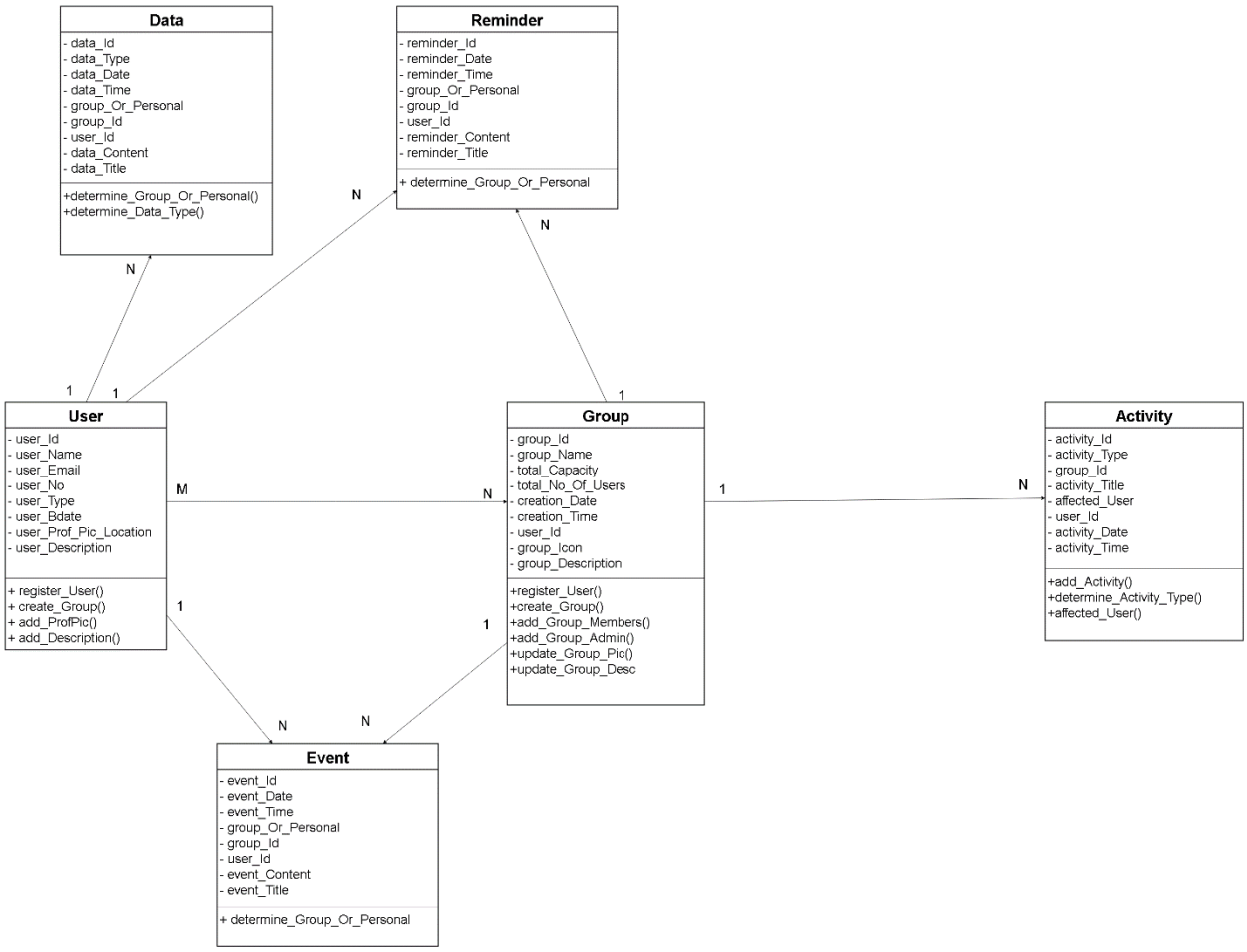

Figure 5 : Class Diagram

Database Diagram

Since our database structure is hierarchical (Tree-Like), there are no tables and relationships between them.

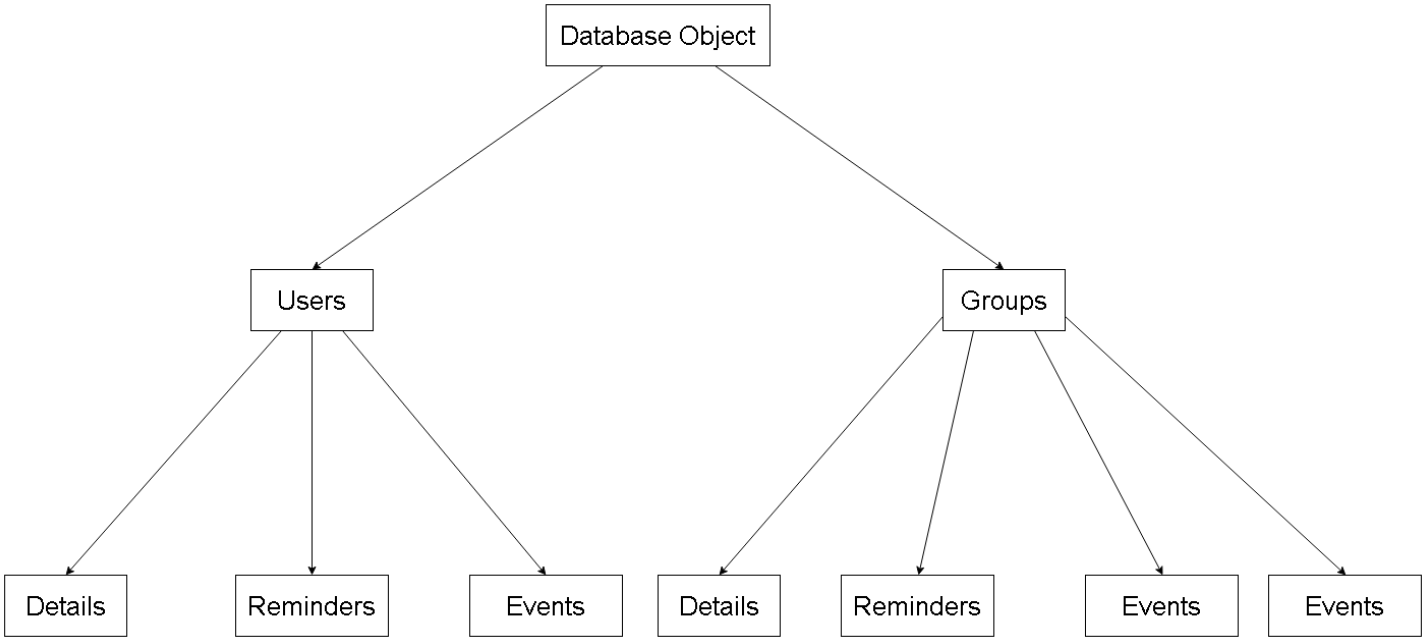

Figure 6 : Database Diagram

3.8 Functional and Behavioral Modelling

Activity Diagram – Adding Reminder/Event

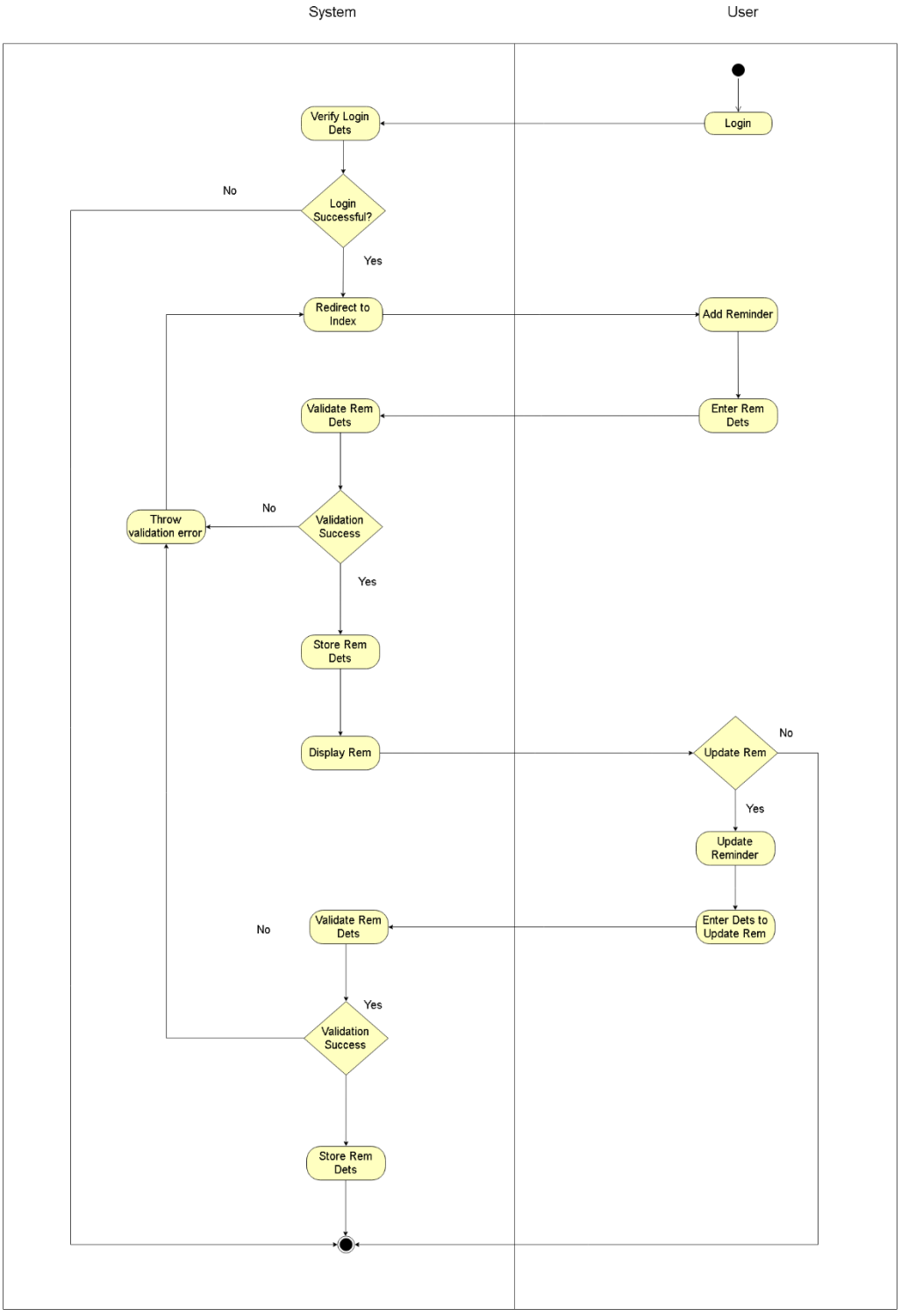

Figure 7 : Activity Diagram – Adding Reminder/Event

Activity Diagram – Creating Group

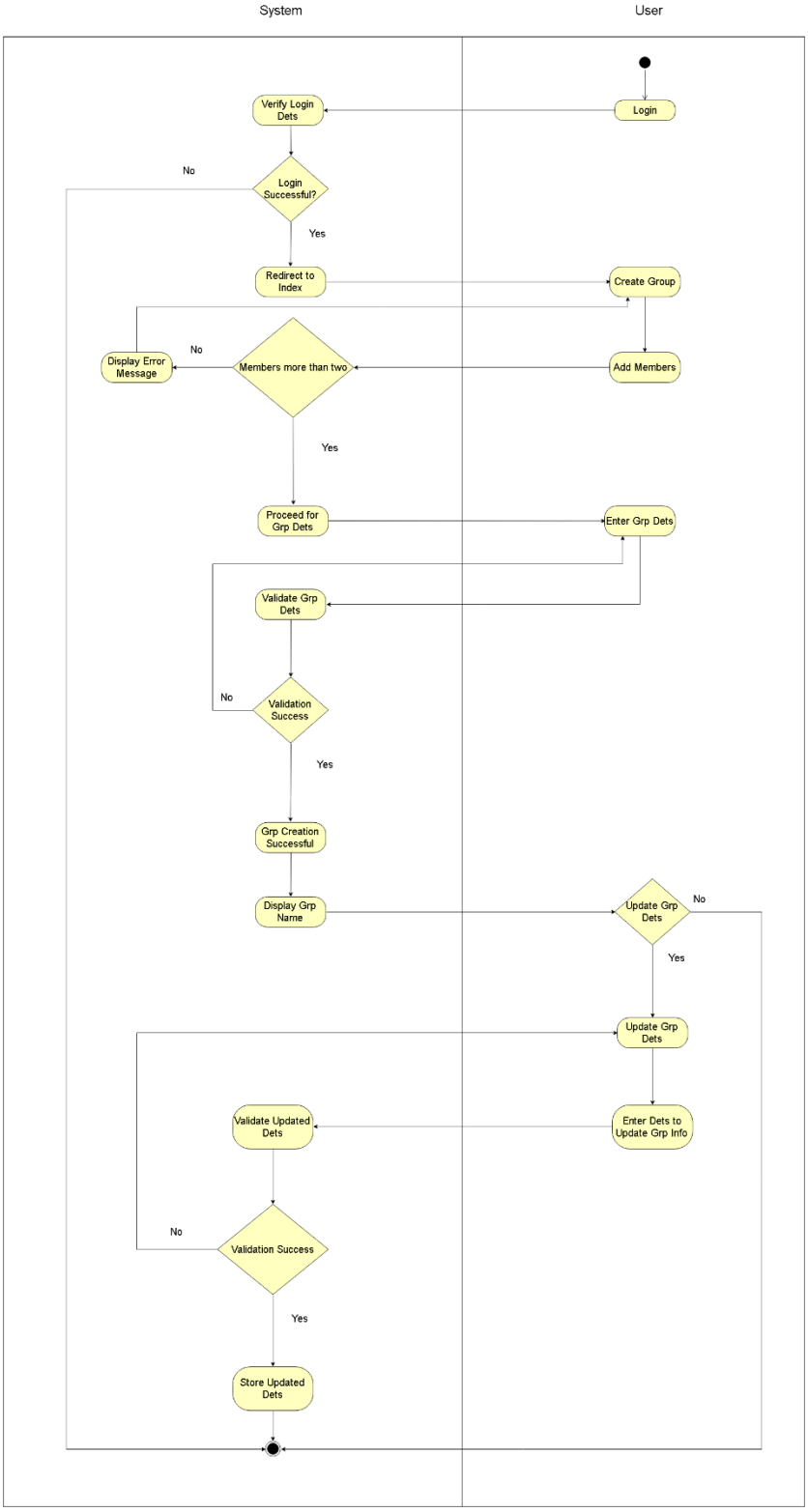

Figure 8 : Activity Diagram – Creating Group

Sequence Diagram – Add Reminder

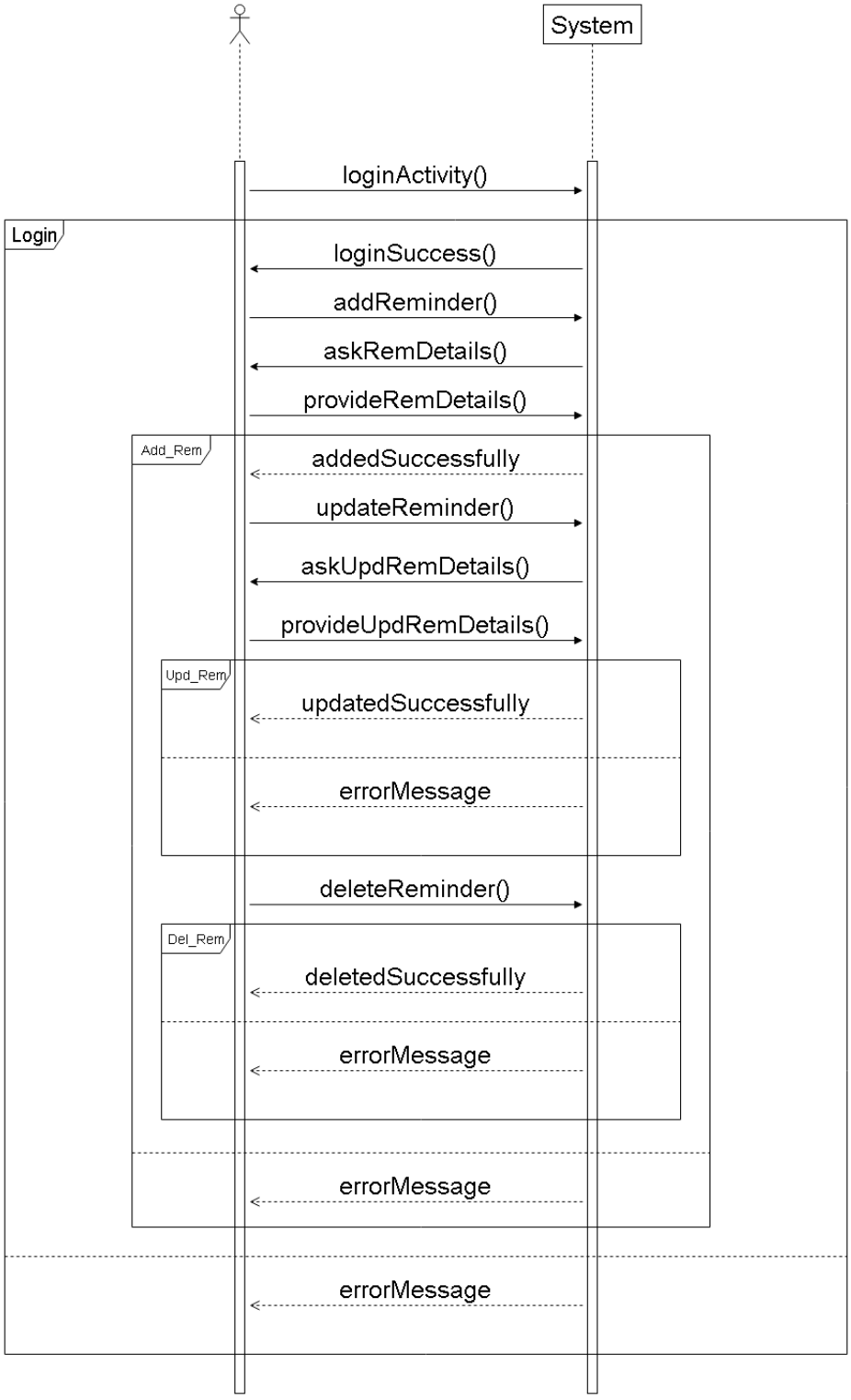

Figure 9 : Sequence Diagram – Add Reminder

Sequence Diagram – Upload Image

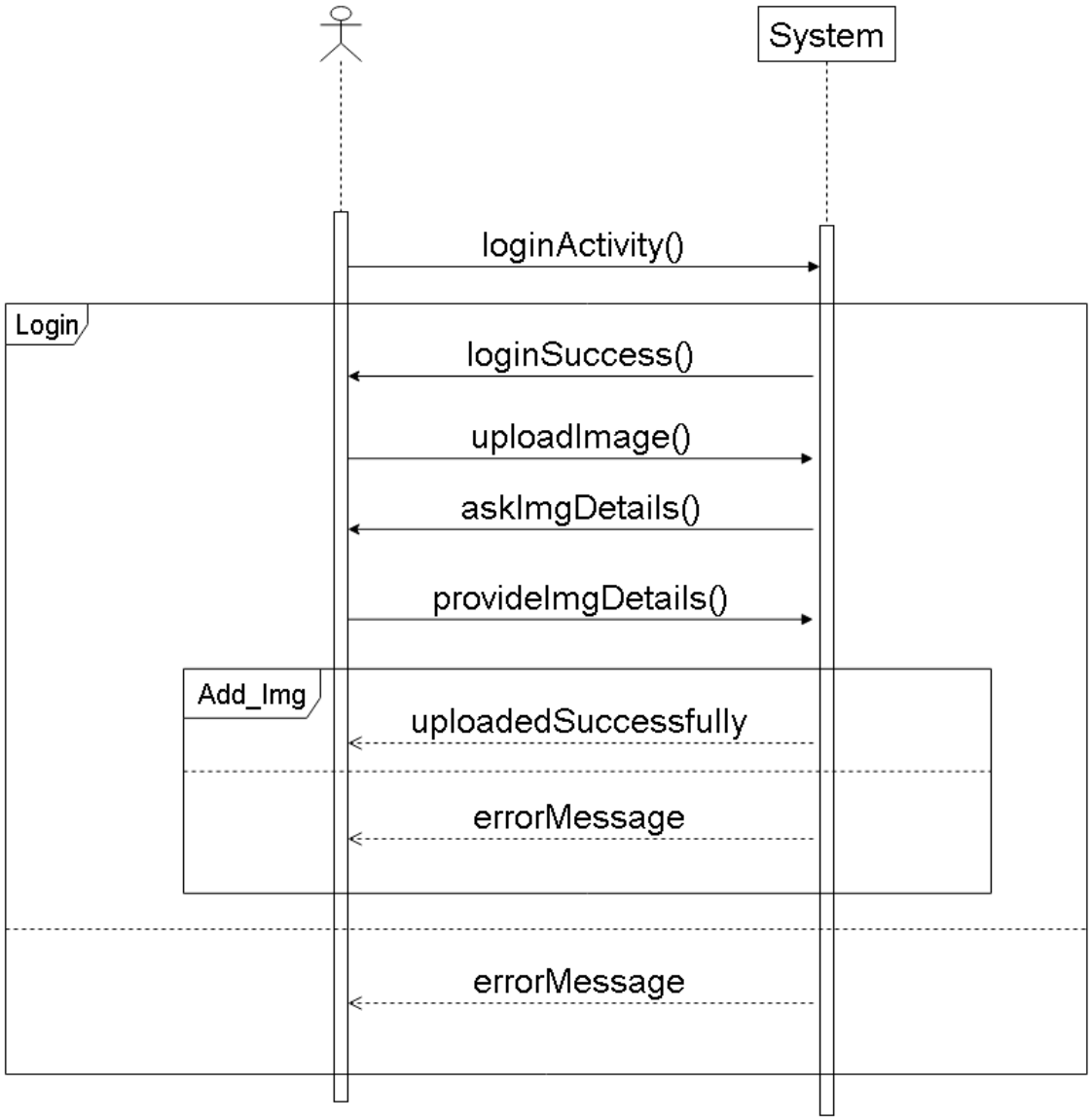

Figure 10 : Sequence Diagram – Upload Image

Sequence Diagram – Create Group

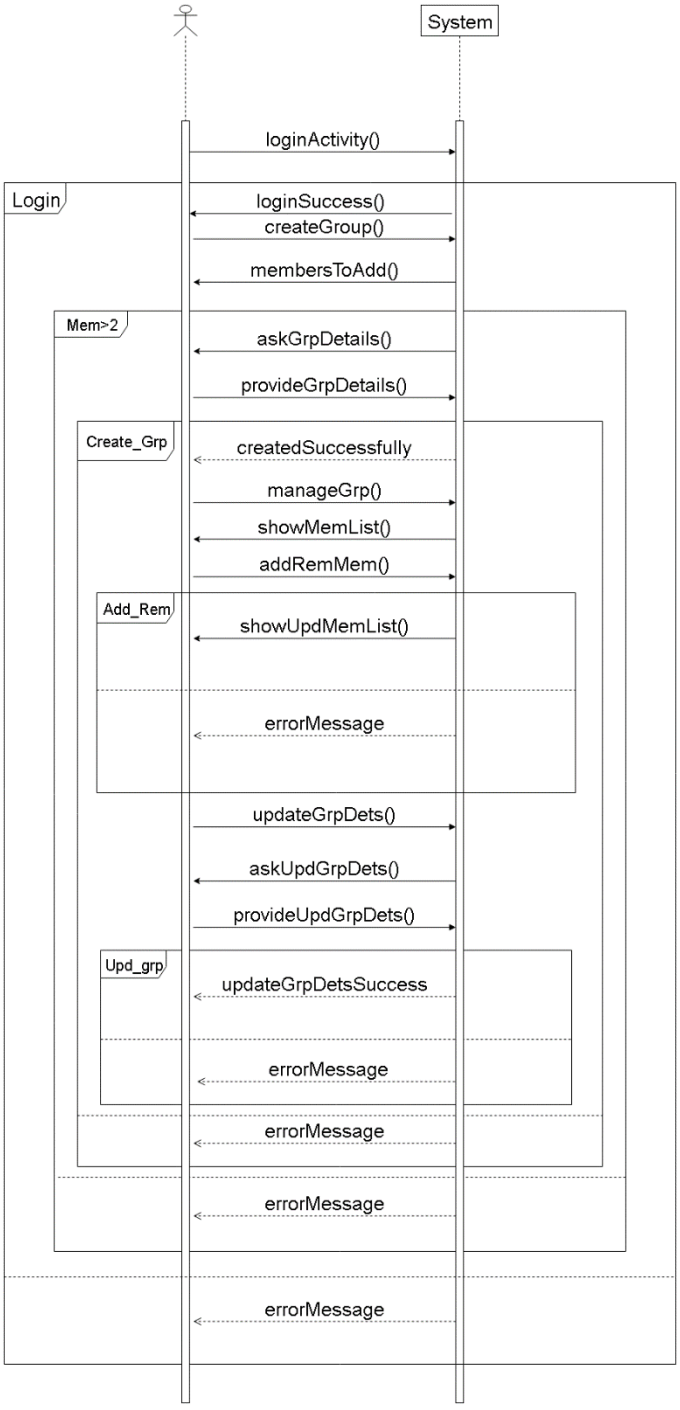

Figure 11 : Sequence Diagram – Create Group

3.9 Selection of Hardware & Software Justification

Since the project works as a website and application both, inclusion of mobile hardware and software is necessary.

The project is a reminder/event manager and a chat application. Such types of applications don't often occupy too much of mobile/desktop resources. Rather, they just require a minimum. It doesn't contain any graphic intensive element that would put load on the processor or any hardware component. It carries out basic tasks and behaves like a native application designed to eat much less resources then even a native application.

However, as this application interacts with the server most of the time, constant internet connection is required. Without internet, one can't set up reminders, events, create group and can't even carry out chat operations. The components that can be cached are the reminders/events already created, groups already created and the last message received in that group.

As this application is meant to carry out menial or day-to-day tasks, it won't be eating good amount of resources. This application could be made run even on extreme low end devices, however, with the constant interaction with server, they may show lags or stutters. Therefore, this application is for low to high end devices.

4. System Design

4.1 Tables and Relationship

A hierarchical model represents the data in a tree-like structure in which there is a single parent for each record. To maintain order there is a sort field which keeps sibling nodes into a recorded manner. These types of models are designed basically for the early mainframe database management systems, like the Information Management System (IMS) by IBM.

This model structure allows the one-to-one and a one-to-many relationship between two/ various types of data. This structure is very helpful in describing many relationships in the real world; table of contents, any nested and sorted information.

The hierarchical structure is used as the physical order of records in storage. One can access the records by navigating down through the data structure using pointers which are combined with sequential accessing. Therefore, the hierarchical structure is not suitable for certain database operations when a full path is not also included for each record.

Data in this type of database is structured hierarchically and is typically developed as an inverted tree. The "root" in the structure is a single table in the database and other tables act as the branches flowing from the root. (Database, 2018)

4.2 System Procedural Design

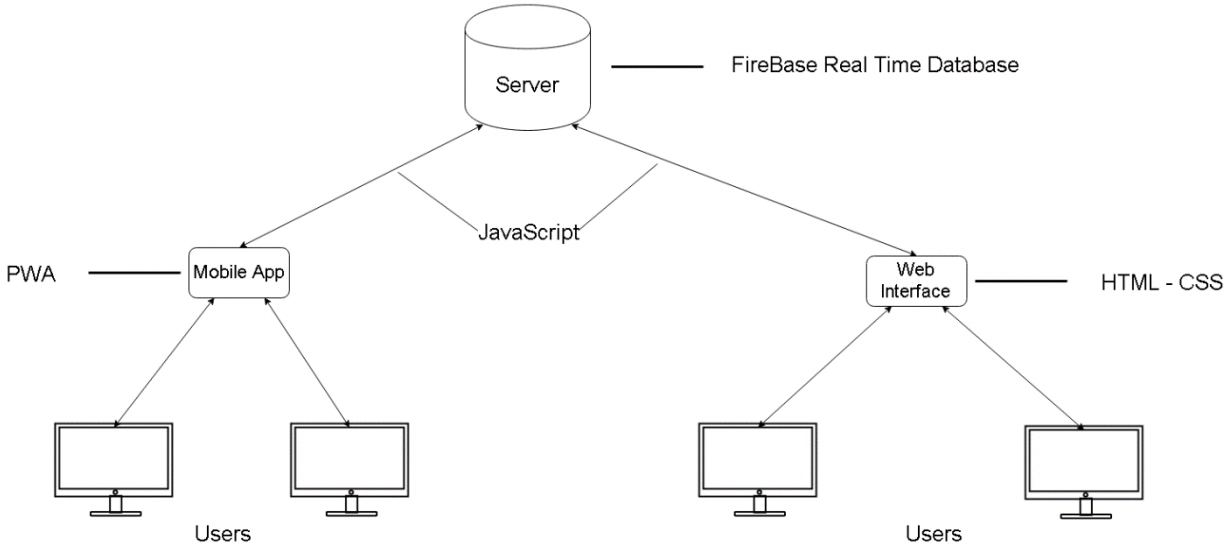

Figure 10 : Architectural System Design

The architectural design comprises of the structure of the system. The system designed would be able to behave as a native app and website at the same time. For designing such system, the biggest drawback is two different components. One would be the Android Application developed on Android Studio and other would be website developed using HTML and CSS.

To get rid of such drawback, PWA was introduced. PWA with the help of service worker and manifest file, can convert website into a native app such that the functionality doesn't take a hit. However, the website should be fully responsive and that was achieved with the help of Bootstrap. Bootstrap helps in making a website scale according to different sizes so that the GUI doesn't take a hit.

With the use of HTML, CSS and PWA components, the website should be scalable and can be a native app whenever the user wishes to.

The users can interact with any interface i.e. mobile or web, the requests are to processed by JavaScript and the database access to be handled by Firebase. The interaction with the Server would be done with the help of JavaScript.

Firebase provides a realtime database where changes are reflected eliminating the need of reloading. Without the use of reloading, the changes are automatically seen on the webpage or app.

Thus, this is the architectural design for the application.

4.3 Design Pseudo Code or Algorithm

1. Login User – Google, Other Email
   - If user have a google account, then they can login by clicking the 'Login with Google' button.
   - If user wants to login with different id, then they can login by clicking the 'Login with Mail' button.

2. If login successful, enter Index page.
   - After logging in, user would able to see their personal dashboard.
   - User can add reminder, set event, save images and also can make a group in their personal dashboard.

3. Add reminder, add event, Create Group, Logout on Index page.
   - Adding Reminder:
     - While adding reminder user have to enter the details which is asked in modal.
     - If user left the textbox empty named title, then user cannot proceed to the addition of reminder successfully. Title is mandatory while adding reminder.
     - User can also update or delete reminder by clicking "i" button.

   - Adding Event:
     - While adding event user have to enter the details which is asked in modal.
     - If user left the textbox empty named title, then user cannot proceed to the addition of reminder successfully. Title is mandatory while adding reminder.
     - User can also update or delete event by clicking "i" button.

   - Create Group:
     - While creating group, user have to add members by searching them with their email id. If email id is registered, then it will appear in another table with add button.
     - After adding email id, a modal will appear asking for group details to be filled.
     - User will be redirected to index page there he will be able to see the group which he/she made.
     - Now user can chat with group members, can share images to each other and also can add reminder and event for group members.

4. Display all reminders, events, created by that particular user.
   - Group members can see all reminder and event in reminder and event page.
   - On reminder and event page group members can see all the reminder and event and also can update or delete it.

5. Update reminders or events.
    - After adding reminder or setting up event, if user wants to change something then he can update all details by going to the more details page by clicking “i” button.

6. Delete reminders or events.
    - After adding reminder or setting up event, if user wants to delete reminder or event then he can delete by going to the more details page by clicking “i” button.

4.4 Timeline Chart

Task Name	**Start**	**End**	**Duration (days)**
Problem Statement finding	20/7/18	25/7/18	5
Formulating Problem Statement	26/7/18	31/7/18	5
Requirement Gathering	1/8/18	8/8/18	7
System Analysis	9/8/18	9/9/18	30
System Design	10/9/18	20/9/18	10
Database Design	21/9/18	28/9/18	7
Implementation	29/9/18	19/11/18	50
Testing	20/11/18	2/12/18	12
Deployment	3/12/18	6/12/18	3

Table 4 : TimeLine Chart

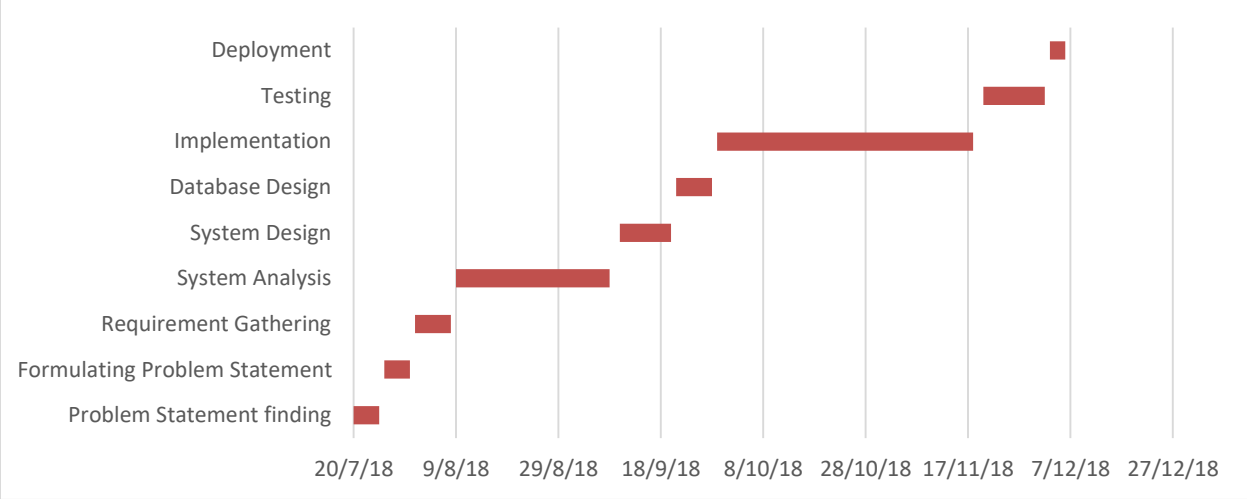

Figure 11 : Gantt Chart

5. Implementation Planning & Details

5.1 Implementation Environment

HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. (HTML, 2015)

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents. (CSS, 2012)

JAVASCRIPT

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme. (JavaScript, 2007)

JQUERY

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT License. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.

jQuery's syntax is designed to make it easier to navigate a 35 document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plugins on top of the JavaScript library.

This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications. (JQuery, 2014)

FIREBASE

Firebase is a mobile and web app development platform that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit.

REALTIME DATABASE

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets developers store and sync between your users in realtime. The Realtime Database is really just one big JSON object that the developers can manage in realtime.

With just a single API, the Firebase database provides app with both the current value of the data and any updates to that data. Realtime syncing makes it easy for your users to access their data from any device, be it web or mobile. Realtime Database also helps your users collaborate with one another.

Another amazing benefit of Realtime Database is that it ships with mobile and web SDKs, allowing you to build your apps without the need for servers. When your users go offline, the Realtime Database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronized.

The Realtime Database can also integrate with Firebase Authentication to provide a simple and intuitive authentication process.

AUTHENTICATION

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app.

Normally, it would take developers months to set up their own authentication system. And even after that, they would need to keep a dedicated team to maintain that system. But with the use of Firebase, they can set up the entire system in under 10 lines of code that will handle everything for them, including complex operations like account merging.

They can authenticate their app’s users through the following methods:

- Email & Password
- Phone numbers
- Google
- Facebook
- Twitter

Using Firebase Authentication makes building secure authentication systems easier, while also improving the sign-in and onboarding experience for end users.

Firebase Authentication is built by the same people who created Google Sign-in, Smart Lock, and Chrome Password Manager.

STORAGE

Firebase Storage is a standalone solution for uploading user-generated content like images and videos from an iOS and Android device, as well as the Web.

Firebase Storage is designed specifically to scale your apps, provide security, and ensure network resiliency.

Firebase Storage uses a simple folder/file system to structure its data. (Firebase, 2016)

JSON

In computing, JavaScript Object Notation (JSON) is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser–server communication, including as a replacement for XML in some AJAX-style systems.

JSON is a language-independent data format. It was derived from JavaScript, but as of 2017 many programming languages include code to generate and parse JSON-format data. The official Internet media type for JSON is application/json. JSON filenames use the extension .json. (JSON, 2015)

5.2 Module Specification

Login

- Login with google: User can login with his/her google id.
- Login with email: User can login with different account other than Gmail id.

Adding reminder/event – User

- Add – User can add reminder/event for personal use.
- Update – If earlier user added the reminder/event then user can update all details from more details page.
- Delete – If user wants to delete reminder/event then user can delete from more details page.

Data Sharing – User

- Upload – User can save images for their own purpose in saved images pages

Creating Group

- Create – User can create group by adding email id and after that entering group details, group will be created.

Adding reminder/event – Group

- Add – User can add reminder/event in group and for personal also.
- Update – If earlier user added the reminder/event then user can update all details from more details page.
- Delete – If user wants to delete reminder/event then user can delete from more details page.

Data Sharing – Group

- Upload – Group Members can save images for other group members in saved images pages

Group Management

- Add – New members can be added by clicking on add participants.
- Remove – Existing member can be removed by clicking on “x” button.

5.3 Security Features

User role management

Admin

1. User can create group.
2. He can add or remove group members.
3. He can even change the group settings.
4. He can set reminders or events.
5. He can share notes in the form of text or images.
6. He can view group activities.

Group Member

1. This type of user can't create group. He neither can add or remove group members.
2. He also hasn't got the access to change the group settings.
3. However, he can set reminders or events for the group.
4. He can also share notes in the form of text or images.
5. Lastly, he can also view group activities.

Individual (Not a group member)

1. This type of user is first of all not in any group.
2. He can set reminders or events for himself only.
3. He can also save notes in the form of text or images with himself only.
4. However, he can always become an admin by creating group.

Session Management

1. Whenever a user will start using our application, we will save a unique identification information about him, in an object which is available throughout the application, until its destroyed.
2. So wherever the user goes, we will always have his information and we can always manage which user is doing what.
3. Whenever a user wants to exit from your application, destroy the object with his information.

URL – encoding

1. URL encoding converts characters into a format that can be transmitted over the Internet.
2. Since our project revolves around JavaScript and firebase, therefore for URL encoding, we will using encodeURIComponent() function to encode URL properly.

Firebase Authentication

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app.

Normally, it would take developers months to set up their own authentication system. And even after that, they would need to keep a dedicated team to maintain that system. But with the use of Firebase, they can set up the entire system in under 10 lines of code that will handle everything for them, including complex operations like account merging.

They can authenticate their app's users through the following methods:

- Email & Password
- Phone numbers
- Google
- Facebook
- Twitter

Using Firebase Authentication makes building secure authentication systems easier, while also improving the sign-in and onboarding experience for end users.

Firebase Authentication is built by the same people who created Google Sign-in, Smart Lock, and Chrome Password Manager.

Firebase Push Method

Firebase provides a push method which generates a random key for that particular child and stores the data inside that key. One can't guess the pattern nor the way in which the key was generated. The key generated is being used for storing and retrieving data. It increases the load on developer since to store and retrieve that key is somehow needed, however for others it strengthens the security of the data.

The push() method will create a unique id when the data is pushed.

5.4 Coding Standards

Coding standards are a set of guidelines, best practices, programming styles and conventions that developers adhere to when writing source code for a project.

The standards which were followed are:

- Naming the variables and methods/functions in the most relevant way. Variables with the same name are differentiated with the purpose for which they are being used.

- Procedural structure was used up to a certain standard as function/methods names are used in different files.

- Different JavaScript files were used for different pages to avoid any consequences for the same variables and method/function names.

- CSS files were stored in a different folder and were called in different HTML pages.

- For Firebase integration, the same set of script files containing the same info needs to be called on every HTML page, so duplication of code is to be allowed here.

- Since Firebase deals with JavaScript & jQuery, the data retrieval and storage was done through JavaScript variables and methods/functions.

5.5 Coding Sample

For integrating Firebase

```
<script>

// Initialize Firebase

var config = {

apiKey: "AIzaSyD24iQ_6nTppktpmW9kIIPkuOoZB2AxO5I",

authDomain: "reig-1538140691138.firebaseapp.com",

databaseURL: "https://reig-1538140691138.firebaseio.com/",

projectId: "reig-1538140691138",

storageBucket: "reig-1538140691138.appspot.com",

messagingSenderId: "76801264020"

};

firebase.initializeApp(config);

</script>
```

For adding Reminder

HTML

```
<!-- Modal Start reminder -->

    <div class="modal fade" id="darkModalForm" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

        <div class="modal-dialog form-dark" role="document">

            <!--Content-->

            <div class="modal-content card card-image">

                <div class="text-white rgba-stylish-strong py-5 px-4 z-depth-4">

                    <!--Header-->
```

```
<div class="modal-header text-center pb-4">

<h3 class="modal-title w-100 white-text font-weight-bold" id="myModalLabel"><strong>ADD</strong> <a class="green-text font-weight-bold"><strong> REMINDER</strong></a></h3>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<i class="fa fa-close" style="color: black !important; height: 5% !important;"></i>

</button>

</div>

<!--Body-->

<div class="modal-body">

<form id="reminderform" >

<!--Body-->

<div class="md-form mb-5">

<label data-error="Some error occured" data-success="Correct" for="Form-email5">Title</label>

<input type="text" id="reminder_title" oninput="InvalidMsg(this);" oninvalid="InvalidMsg(this);" class="form-control validate white-text" required>

</div>


<div class="md-form pb-3">

<br>

<label data-error="Some error occured" data-success="Correct" for="Form-pass5">Content</label>

<input type="text" id="reminder_content" class="form-control validate white-text">

</div>

<div class="md-form pb-3">
```

```
                    <br>
                    <label data-error="Some error occured" data-success="Correct" for="Form-pass5">Category</label>
                    <!-- <input type="text" id="Form-pass5" class="form-control validate white-text"> -->
                    <select class="form-control validate white-text" id="reminder_category">
                      <option value="Default">Default</option>
                      <option value="Personal">Personal</option>
                      <option value="Shopping">Shopping</option>
                      <option value="Shopping">Work</option>
                      <option value="Whishlist">Whishlist</option>
                    </select>
                  </div>
                  <div class="form-group">
                    <br>
                    <label class="label-control" style="color: black;">Select date & time</label>
                    <input type="text" id="reminder_datetime" class="form-control datetimepicker" value="10/05/2016">
                  </div>
                  <!-- <div class="form-group mt-4">
                    <input class="form-check-input" type="checkbox" id="checkbox624">
                    <label for="checkbox624" class="white-text form-check-label">
                      Accept the<a href="#" class="green-text font-weight-bold"> Terms and Conditions</a></label>
                    </div> -->
                    <!--Grid row-->
```

```
                    <div class="row d-flex align-items-center mb-4">


                        <!--Grid column-->

                        <div class="text-center mb-3 col-md-12">

                            <br>

                            <button class="btn btn-primary btn-block btn-rounded z-depth-1" id="btn_reminder_add">ADD</button>

                        </div>

                        <!--Grid column-->

                    </div>

                    </div>

                    <!--Grid row-->

                </form>

            </div>

        </div>

    </div>

    <!--/.Content-->

</div>

</div>

<!-- Modal -->
```

JavaScript with Firebase

```
document.getElementById('btn_reminder_add').onclick = function(){

    var reminder_title_val = document.getElementById('reminder_title').value;

    var reminder_content_val = document.getElementById('reminder_content').value;

    var reminder_category_val = document.getElementById('reminder_category').value;
```

```
    var reminder_datetime_val = document.getElementById('reminder_datetime').value;
    var reminder_datetime_split = reminder_datetime_val.split(" ");
    var reminder_date_only = reminder_datetime_split[0];
    var reminder_time_only = reminder_datetime_split[1];
    var reminder_ampm_only = reminder_datetime_split[2];
    var reminder_fulltime_only = reminder_time_only + " " + reminder_ampm_only;
    if(reminder_title_val == ""){
        eventRef.preventDefault();
    }
    eventRef.child(user.uid).child('reminders').push({
        title : reminder_title_val,
        content : reminder_content_val,
        category : reminder_category_val,
        date : reminder_date_only,
        time : reminder_fulltime_only
    });

    console.log('Done');
};
```

6. Testing

6.1 Test Plan

Introduction

Test planning, the most important activity to ensure that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project. It also defines the size of the test effort.

It is the main document often called as master test plan or a project test plan and usually developed during the early phase of the project.

Test Items

- Validation
- Navigation
- Functionality on every button click
- Script loading
- Random access
- Responsiveness of every page
- PWA behavior of application
- Module success
- Gorilla inputs

6.2 Test Strategy

A **test strategy** is an outline that describes the testing approach of the software development cycle. It is created to inform project managers, testers, and developers about some key issues of the testing process. This includes the testing objective, methods of testing new functions, total time and resources required for the project, and the testing environment.

Test strategies describe how the product risks of the stakeholders are mitigated at the test-level, which types of testing are to be performed, and which entry and exit criteria apply. They are created based on development design documents. System design documents are primarily used and occasionally, conceptual design documents may be referred to. Design documents describe the functionality of the software to be enabled in the upcoming release. For every stage of development design, a corresponding test strategy should be created to test the new feature sets.

Different Test approaches:

There are many strategies that a project can adopt depending on the context and some of them are:

- Dynamic and heuristic approaches
- Consultative approaches
- Model-based approach that uses statistical information about failure rates.
- Approaches based on risk-based testing where the entire development takes place based on the risk
- Methodical approaches which is based on failures.
- Standard-compliant approach specified by industry-specific standards.

Factors to be considered:

- Risks of product or risk of failure or the environment and the company
- Expertise and experience of the people in the proposed tools and techniques.
- Regulatory and legal aspects, such as external and internal regulations of the development process
- The nature of the product and the domain.

Strategies:

Graphical User Interface

GUI testing is the process of testing the system's Graphical User Interface of the Application Under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - toolbar, menu bar, dialog boxes and windows, etc.

GUI is what user sees. Say if you visit google.com what you will see say home page it is the GUI (graphical user interface) of the site. A user does not see the source code. The interface is visible to the user. Especially the focus is on the design structure, images that they are working properly or not.

Checklist under GUI:

1. Testing the size, position, width, height of the elements.
2. Testing of the error messages that are getting displayed.
3. Testing the different sections of the screen.
4. Testing of the font whether it is readable or not.
5. Testing of the screen in different resolutions with the help of zooming in and zooming out like 640 x 480, 600x800, etc.
6. Testing the alignment of the texts and other elements like icons, buttons, etc. are in proper place or not.
7. Testing the colors of the fonts.
8. Testing the colors of the error messages, warning messages.
9. Testing whether the image has good clarity or not.
10. Testing the alignment of the images.
11. Testing of the spelling.
12. The user must not get frustrated while using the system interface.
13. Testing whether the interface is attractive or not.
14. Testing of the scrollbars according to the size of the page if any.
15. Testing of the disabled fields if any.
16. Testing of the size of the images.

Database Testing

Database testing is one of the major testing which requires tester to expertise in checking hierarchical approach, writing data and methods. Testing can be performed in web application or desktop and database can be used in the application. There are many projects like banking, finance, health insurance which requires extensive database testing.

Database testing basically include the following:

1. Data validity testing.
2. Data Integrity testing
3. Performance related to data base.
4. Testing of triggers and functions.

- For doing data validity testing one should be good dealing with Firebase.
- For data integrity testing one should know about referential integrity and different constraint.
- For performance related things one should have idea about the hierarchical structure and design.
- For testing Procedure triggers and functions one should be able to understand the same.

6.3 Test Methods

Black Box Testing

Also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

Advantages

- Tests are done from a user's point of view and will help in exposing discrepancies in the specifications.
- Tester need not know programming languages or how the software has been implemented.
- Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer-bias.
- Test cases can be designed as soon as the specifications are complete.

Disadvantages

- Only a small number of possible inputs can be tested and many program paths will be left untested.
- Without clear specifications, which is the situation in many projects, test cases will be difficult to design.
- Tests can be redundant if the software designer/developer has already run a test case.
- Ever wondered why a soothsayer closes the eyes when foretelling events? So is almost the case in Black Box Testing. (Testing, 2014)

White Box Testing

Also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural 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.

This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees.

Advantages

- Testing can be commenced at an earlier stage. One need not wait for the GUI to be available.
- Testing is more thorough, with the possibility of covering most paths.

Disadvantages

- Since tests can be very complex, highly skilled resources are required, with a thorough knowledge of programming and implementation.
- Test script maintenance can be a burden if the implementation changes too frequently.
- Since this method of testing is closely tied to the application being tested, tools to cater to every kind of implementation/platform may not be readily available. (Testing W. B., 2014)

White Box testing will only be used when Black Box testing fails. During Black Box testing, if GUI fails, then White Box testing will come into place. We will analyze the code of that particular GUI part and correct the code and re-test it to check the changes.

6.4 Test Cases

Login

Test Scenario Id : SC1				**Test Case Id :** 101		
Test Case Description : Test Case for Login				**Test Priority :** High		
Pre – Requisite: 1. User should have knowledge about use of Internet. 2. User should have landed on the website or web app. 3. User should have a working email address.				**Post – Requisite:** 1. If inputs are matching the criteria, data will be stored in the system. 2. User will be taken to index page.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Click on Sign in with Google button.		A page/modal asking for login details.	A page asking for login details.	Pass	
2.	Email field is left blank		An error related to email field.	An error in the email field	Pass	
3.	Invalid email entered.	xyz@gmail.com	An error related to invalid email address.	An error related to invalid email address.	Pass	
4.	Entered correct email but wrong password	Email : rishi123@gmail.com Password : rishi12	An error related to email/password being wrong.	An error stating 'invalid email or password'	Pass	
5.	Entered correct email and password	Email : rishi123@gmail.com Password : rishi123	Redirected to Index page.	Redirected to Index page.	Pass	

Table 5 : Login Test Case

Index Page

Test Scenario Id : SC2				**Test Case Id** : 201		
Test Case Description : Test Case for directly accessing Index page.				**Test Priority :** High		
Pre – Requisite: 1. User should have landed on the website or web app. 2. User should have logged-in.				**Post – Requisite:** 1. User should be redirected to Login Page.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	After logging in successfully, user trying to copy the url and pasting it in an incognito tab.	URL of Index	Redirected to Login page without letting user access anything.	Redirected to Login page.	Pass	As no user was logged-in, the site redirected itself to Login page.

Table 6 : Index Page Test Case

Reminder – User

Test Scenario Id : SC3				**Test Case Id :** 301		
Test Case Description : Test Case for adding Reminder - User.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the website or web app. 2. User should have logged-in.				**Post – Requisite:** 1. Reminder form data would be added into the Firebase. 2. User would be able to see the reflected changes i.e. Added Reminder on Index Page without Reloading.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Leaving all fields blank.		Error related to empty field.	Error at first field titled 'Title'.	Pass.	
2.	Only entering field named 'Title'	Title : 'Test'	Error related to second field left empty.	Reminder Added.	Fail.	We have designed the system in such a manner that user is only required to add the 'Title' field and can leave all other fields empty.
3.	Entering every field expect 'Title'.	Content: 'Test – content', Category: 'Default', Date: '21/11/18', Time: '08:00 AM'	Error related to 'Title' field being left empty.	Error related to 'Title' field being left empty.	Pass.	As 'Title' field is the required one, leaving it blank will validate.

Table 7 : Reminder – User Test Case

Event – User

Test Scenario Id : SC3				**Test Case Id :** 302		
Test Case Description : Test Case for adding Event - User.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the website or web app. 2. User should have logged-in.				**Post – Requisite:** 1. Event form data would be added into the Firebase. 2. User would be able to see the reflected changes i.e. Added Event on Index Page without Reloading.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Leaving all fields blank.		Error related to empty field.	Error at first field titled 'Title'.	Pass.	
2.	Only entering field named 'Title'	Title : 'Test'	Error related to second field left empty.	Event Added.	Fail.	We have designed the system in such a manner that user is only required to add the 'Title' field and can leave all other fields empty.
3.	Entering every field expect 'Title'.	Content: 'Test – content', Category: 'Default', Date: '21/11/18', Time: '08:00 AM'	Error related to 'Title' field being left empty.	Error related to 'Title' field being left empty.	Pass.	As 'Title' field is the required one, leaving it blank will validate.

Table 8 : Event User – Test Case

Updating Reminder – User

Test Scenario Id : SC4				**Test Case Id :** 401		
Test Case Description : Test Case for updating Reminder - User.				**Test Priority :** Low		
Pre – Requisite: User should have landed on the website or web app. User should have logged-in. User should have at least one reminder added.				**Post – Requisite:** Reminder Update form data would be updated into the Firebase. User would be able to see the reflected changes i.e. Updated Reminder on Index Page without Reloading.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Clicked on 'i' button to edit current Reminder		A modal/page related to the Reminder details.	A page related to the Reminder details.	Pass.	
2.	Leaving all fields blank and clicking on 'Update' button.		Error related to empty field.	Error at first field titled 'Title'.	Pass.	
3.	Only entering field named 'Title' and clicking on 'Update' button.	Title : 'Test'	Error related to second field left empty.	Reminder Updated.	Fail.	We have designed the system in such a manner that user is only required to add the 'Title' field and can leave all other fields empty.
4.	Entering every field expect 'Title'	Content: 'Test – content', Category: 'Default', Date: '21/11/18',	Error related to 'Title' field being left empty.	Error related to 'Title' field being left empty.	Pass.	As 'Title' field is the required one, leaving

	and clicking on 'Update' button.	Time: '08:00 AM'				it blank will validate.
5.	Not changing any values and clicking on 'Update'.		Error related to updating values.	Redirected to Index Page.	Fail.	No error message was shown.
6.	Not changing any values and clicking on 'OK'		Redirected to Index page.	Redirected to Index page.	Pass.	

Table 9 : Updating Reminder – User Test Case

Updating Event – User

Test Scenario Id : SC4				**Test Case Id :** 402		
Test Case Description : Test Case for updating Event - User.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the website or web app. 2. User should have logged-in. 3. User should have at least one event added.				**Post – Requisite:** 1. Event Update form data would be updated into the Firebase. 2. User would be able to see the reflected changes i.e. Updated Event on Index Page without Reloading.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Clicked on 'i' button to edit current Event		A modal/page related to the Event details.	A page related to the Event details.	Pass.	
2.	Leaving all fields blank and clicking on 'Update' button.		Error related to empty field.	Error at first field titled 'Title'.	Pass.	
3.	Only entering field named 'Title' and clicking on 'Update' button.	Title : 'Test'	Error related to second field left empty.	Event Updated.	Fail.	We have designed the system in such a manner that user is only required to add the 'Title' field and can leave all other fields empty.
4.	Entering every field expect	Content: 'Test – content', Category: 'Default',	Error related to 'Title' field being left empty.	Error related to 'Title'	Pass.	As 'Title' field is the required one, leaving

	'Title' and clicking on 'Update' button.	Date: '21/11/18', Time: '08:00 AM'		field being left empty.		it blank will validate.
5.	Not changing any values and clicking on 'Update'.		Error related to updating values.	Redirected to Index Page.	Fail.	No error message was shown.
6.	Not changing any values and clicking on 'OK'		Redirected to Index page.	Redirected to Index page.	Pass.	

Table 10 : Updating Event – User Test Case

Adding Image – User

Test Scenario Id : SC5				**Test Case Id :** 501		
Test Case Description : Test Case for adding Image - User.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the website or web app. 2. User should have logged-in. 3. User should have an image to upload.				**Post – Requisite:** 1. Image would be uploaded into the Firebase. 2. User would be able to see the reflected changes i.e. Added Image on Images Page without Reloading.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Uploading a different type of file.	Test.pdf	Error related to type of file.	Error related to type of file.	Pass.	
2.	Only uploading image without entering Caption.	Test.jpg	Error related to 'Caption' being left empty.	File Uploaded.	Fail.	We have designed such a system where User can add caption to the image.
3.	Only entering caption without adding Image.	Caption: 'Test'	Error related to 'Image' field being left empty.	Error related to 'Image' field being left empty.	Pass.	
4.	Leaving both the fields empty.		Error related to fields being left empty.	Error related to 'Image' field being left empty.	Pass.	

Table 11 : Adding Image – User Test Case

Creating – Group

Test Scenario Id : SC6				**Test Case Id :** 601		
Test Case Description : Test Case for creating Group.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the website or web app. 2. User should have logged-in.				**Post – Requisite:** 1. User would be redirected to Index Page where Group name would be added.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Clicking 'Search' button leaving search field empty.		Error related to search field being left empty.	Error related to search field being left empty.	Pass.	
2.	Clicking on 'Next' button without adding any members.		Error related to no members being added.	Error related to no members being added.	Pass.	
3.	Clicking on 'Create' button without adding Group details.		Error related to fields being left blank.	Error related to 'Group Name' being left empty.	Pass.	
4.	Entering 'Group Name' field and leaving rest empty.	Group Name: 'Test'	Group should be created.	Group created.	Pass.	Group Description is optional.
5.	Entering 'Group Description' field and leaving rest empty.	Group Description: 'Test - Description'.	Error related to other fields being left empty.	Error related to 'Group Name' being left empty.	Pass.	

Table 12 : Creating – Group Test Case

Group Chat

Test Scenario Id : SC6				**Test Case Id :** 602		
Test Case Description : Test Case for Group Chatting.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the group page. 2. User should have internet connection. 3. User should have selected a group to chat.				**Post – Requisite:** 1. A message would be sent to the group members.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Typed a message in chat box.	Xyz test message.	A message would be sent to all group members.	A message would be sent to all group members.	Pass.	
1.	Directly clicked the 'Send' button without typing any message.		Error related to empty message textbox / Nothing should happen.	Nothing happened.	Pass.	

Table 13 : Group Chat Test Case

Group Reminder

Test Scenario Id : SC7				**Test Case Id** : 701		
Test Case Description : Test Case for adding Reminder in group.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the website or web app. 2. User should have logged-in. 3. User should have selected a group to add reminder into.				**Post – Requisite:** 1. Reminder form data would be added into the Firebase. 2. Group members would be able to see the reflected changes i.e. Added Reminder on group chat box and on reminder and event Page without Reloading.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Leaving all fields blank.		Error related to empty field.	Error at first field titled 'Title'.	Pass.	
2.	Only entering field named 'Title'	Title : 'Test'	Error related to second field left empty.	Reminder Added.	Fail.	We have designed the system in such a manner that user is only required to add the 'Title' field and can leave all other fields empty.
3.	Entering every field expect 'Title'.	Content: 'Test – content', Category: 'Default', Date: '21/11/18', Time: '08:00 AM'	Error related to 'Title' field being left empty.	Error related to 'Title' field being left empty.	Pass.	As 'Title' field is the required one, leaving it blank will validate.

Table 14 : Group Reminder Test Case

Group Event

Test Scenario Id : SC7				**Test Case Id :** 702		
Test Case Description : Test Case for adding Event in group.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the website or web app. 2. User should have logged-in. 3. User should have selected a group to add event into.				**Post – Requisite:** 1. Event form data would be added into the Firebase. 2. Group members would be able to see the reflected changes i.e. Added Event on group chat box and on reminder and event Page without Reloading.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Leaving all fields blank.		Error related to empty field.	Error at first field titled 'Title'.	Pass.	
2.	Only entering field named 'Title'	Title : 'Test'	Error related to second field left empty.	Event Added.	Fail.	We have designed the system in such a manner that user is only required to add the 'Title' field and can leave all other fields empty.
3.	Entering every field expect 'Title'.	Content: 'Test – content', Category: 'Default', Date: '21/11/18', Time: '08:00 AM'	Error related to 'Title' field being left empty.	Error related to 'Title' field being left empty.	Pass.	As 'Title' field is the required one, leaving it blank will validate.

Table 15 : Group Event Test Case

Updating Reminder - Group

Test Scenario Id : SC8				**Test Case Id :** 801		
Test Case Description : Test Case for updating Reminder in group.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the website or web app. 2. User should have logged-in. 3. User should have selected a group to update reminder for. 4. User should have at least one reminder added.				**Post – Requisite:** 1. Reminder Update form data would be updated into the Firebase. 2. User and group members would be able to see the reflected changes i.e. Updated Reminder on reminder and event Page without Reloading.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Clicked on 'i' button to edit current Reminder		A modal/page related to the Reminder details.	A page related to the Reminder details.	Pass.	
2.	Leaving all fields blank and clicking on 'Update' button.		Error related to empty field.	Error at first field titled 'Title'.	Pass.	
3.	Only entering field named 'Title' and clicking on 'Update' button.	Title : 'Test'	Error related to second field left empty.	Reminder Updated.	Fail.	We have designed the system in such a manner that user is only required to add the 'Title' field and can leave all other fields empty.
4.	Entering every	Content: 'Test – content',	Error related to 'Title' field	Error related to	Pass.	As 'Title' field is the

	field expect ‘Title’ and clicking on ‘Update’ button.	Category: ‘Default’, Date: ‘21/11/18’, Time: ’08:00 AM’	being left empty.	‘Title’ field being left empty.		required one, leaving it blank will validate.
5.	Not changing any values and clicking on ‘Update’.		Error related to updating values.	Redirected to group Page.	Fail.	No error message was shown.
6.	Not changing any values and clicking on ‘OK’		Redirected to group page.	Redirected to group page.	Pass.	

Table 16 : Updating Reminder – Group – Test Case

Updating Event - Group

Test Scenario Id : SC08				**Test Case Id :** 802		
Test Case Description : Test Case for updating Event in group.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the website or web app. 2. User should have logged-in. 3. User should have selected a group to update event for. 4. User should have at least one event added.				**Post – Requisite:** 1. Event Update form data would be updated into the Firebase. 2. User and group members would be able to see the reflected changes i.e. Updated Event on Index Page without Reloading.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Clicked on 'i' button to edit current Event		A modal/page related to the Event details.	A page related to the Event details.	Pass.	
2.	Leaving all fields blank and clicking on 'Update' button.		Error related to empty field.	Error at first field titled 'Title'.	Pass.	
3.	Only entering field named 'Title' and clicking on 'Update' button.	Title : 'Test'	Error related to second field left empty.	Event Updated.	Fail.	We have designed the system in such a manner that user is only required to add the 'Title' field and can leave all other fields empty.
4.	Entering every field	Content: 'Test – content',	Error related to 'Title' field	Error related to 'Title'	Pass.	As 'Title' field is the required

	expect ‘Title’ and clicking on ‘Update’ button.	Category: ‘Default’, Date: ‘21/11/18’, Time: ’08:00 AM’	being left empty.	field being left empty.		one, leaving it blank will validate.
5.	Not changing any values and clicking on ‘Update’.		Error related to updating values.	Redirected to Group Page.	Fail.	No error message was shown.
6.	Not changing any values and clicking on ‘OK’		Redirected to Group page.	Redirected to Group page.	Pass.	

Table 17 : Updating Event – Group Test Case

Upload file - Group

Test Scenario Id : SC09				**Test Case Id :** 901		
Test Case Description : Test Case for uploading files (.docx, .pdf, etc.)				**Test Priority :** Low		
Pre – Requisite: 1. User should have selected a group to send file into.				**Post – Requisite:** 1. Selected file will be sent in group to all members.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Clicked on 'file' icon		A modal/page will appear.	A modal to select file.	Pass.	
2.	Selected file from the modal.		File sent in group.	File sent in group.	Pass.	User will be able to see only files like .docx, .pdf, .pptx, .xlsx etc

Table 18 : Upload File – Group Test Case

Upload image - Group

Test Scenario Id : SC09				**Test Case Id :** 902		
Test Case Description : Test Case for uploading images.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the group page. 2. User should know the path of image which is to be sent in group. 3. User should have selected a group to send image into.				**Post – Requisite:** 1. Selected image will be sent in group to all members.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Clicked on image icon		A modal/page will appear.	A modal to select image.	Pass.	
1.	Selected image from the modal.	"test – image.jpeg"	Image to be send in group.	Image sent in group.	Pass.	User will be able to see only images
2.	Selected an image and then pressed on cancel.		Return to previous page.	Returned to previous page.	Pass.	If user cancels the image uploading then process should terminated.

Table 19 : Upload Image – Group Test Case

Group Details

Test Scenario Id : SC10				**Test Case Id :** 1001		
Test Case Description : Test Case for updating group details.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the group details page. 2. User should have internet connection to update detail. 3. User should have selected a group to see/update details for.				**Post – Requisite:** 1. Selected image will be updated in group profile picture. 2. Group description will be updated.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Clicked on image icon		A modal/page will appear.	A modal to select image.	Pass.	
2.	Selected an image from the modal.	"test – image 2.jpg"	Image updated for group profile picture.	Successfully updated group profile picture.	Pass.	User will be able to see only images
3.	Entered the description	Description "xyz"	Description will be updated.	Successfully updated the description.	Pass.	
4.	Left description empty.		Description should not be updated.	Description updated.	Fail.	We have designed such a system in which admin/group member leaves the description blank, its valid

Table 20 : Group Details Test Case

Group Members

Test Scenario Id : SC10				**Test Case Id :** 1002		
Test Case Description : Test Case for group members.				**Test Priority :** Low		
Pre – Requisite: 1. User should have landed on the group members page. 2. User should have internet connection to remove a group member. 3. User should have selected a group. 4. User should be an admin.				**Post – Requisite:** 1. A group member would be removed from group.		
Sr. No	Action	I/Ps	Expected O/Ps	Actual O/Ps	Test Result	Test Comments
1.	Clicked on cross icon		A modal/page will appear asking for confirmation to remove selected member.	A modal asking for confirmation.	Pass.	
2.	Admin clicked on 'OK' button.		User will be removed from group.	User will be removed from group.	Pass.	
3.	Admin clicked on 'Cancel' button.		User will be redirected to group member page.	User will be redirected to group member page.	Pass.	

Table 21 : Group Members Test Case

7. Screenshots

7.1 Login Page

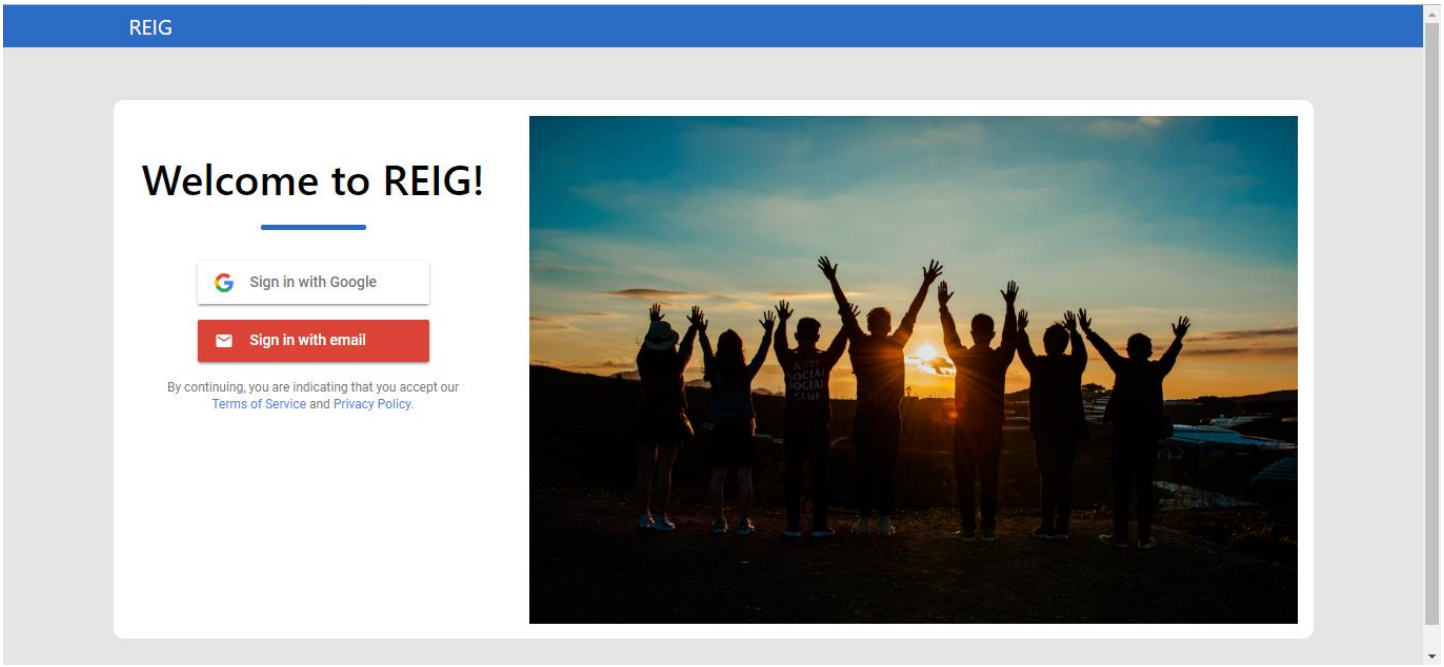

Figure 12 : Login Page

7.2 Index Page

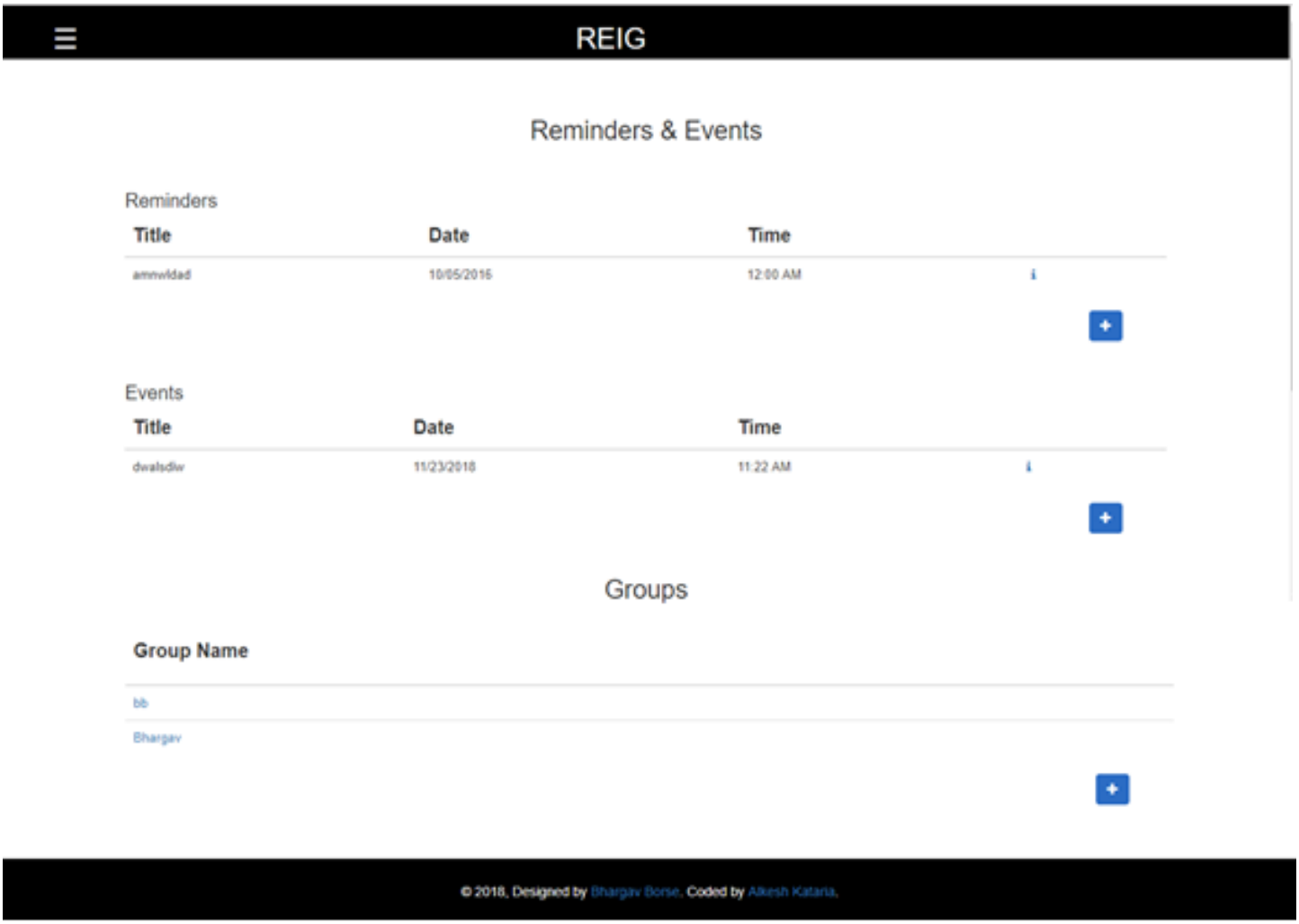

Figure 13 : Index Page

7.3 Add Reminder – Modal (For User/Group)

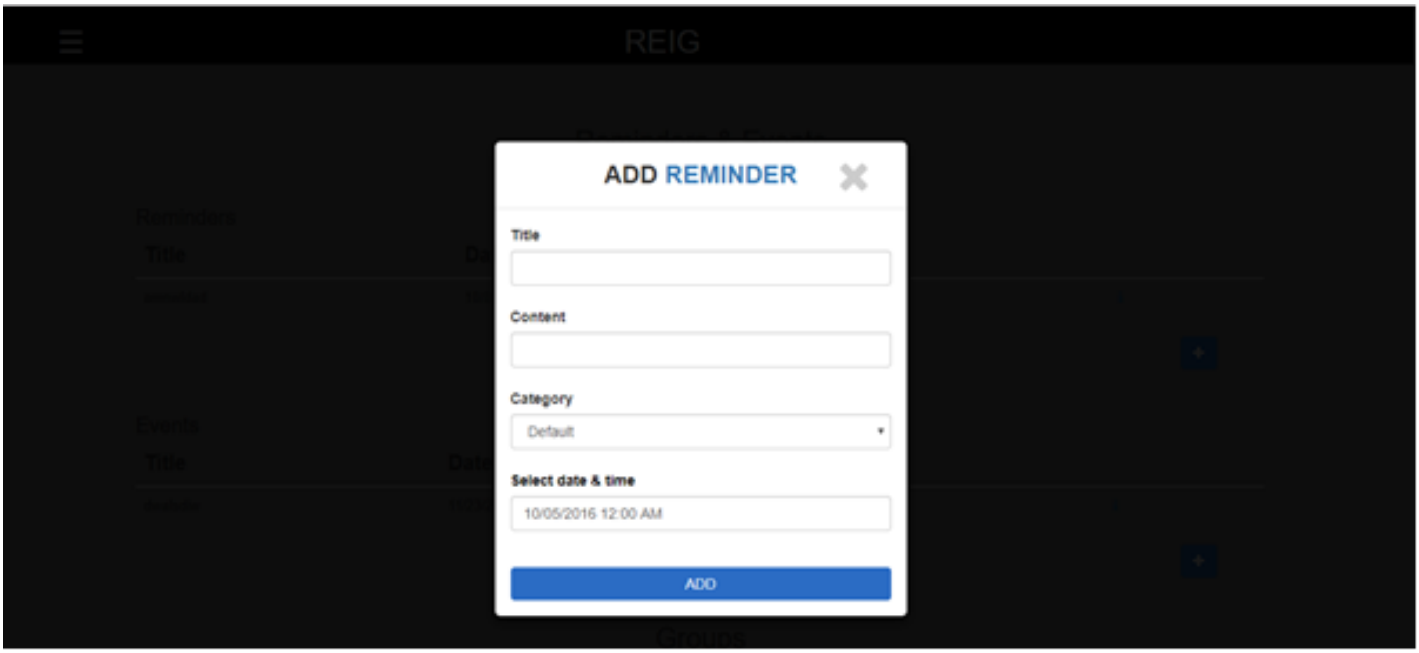

Figure 14 : Add Reminder Modal (For User/Group)

7.4 Add Event – Modal (For User/Group)

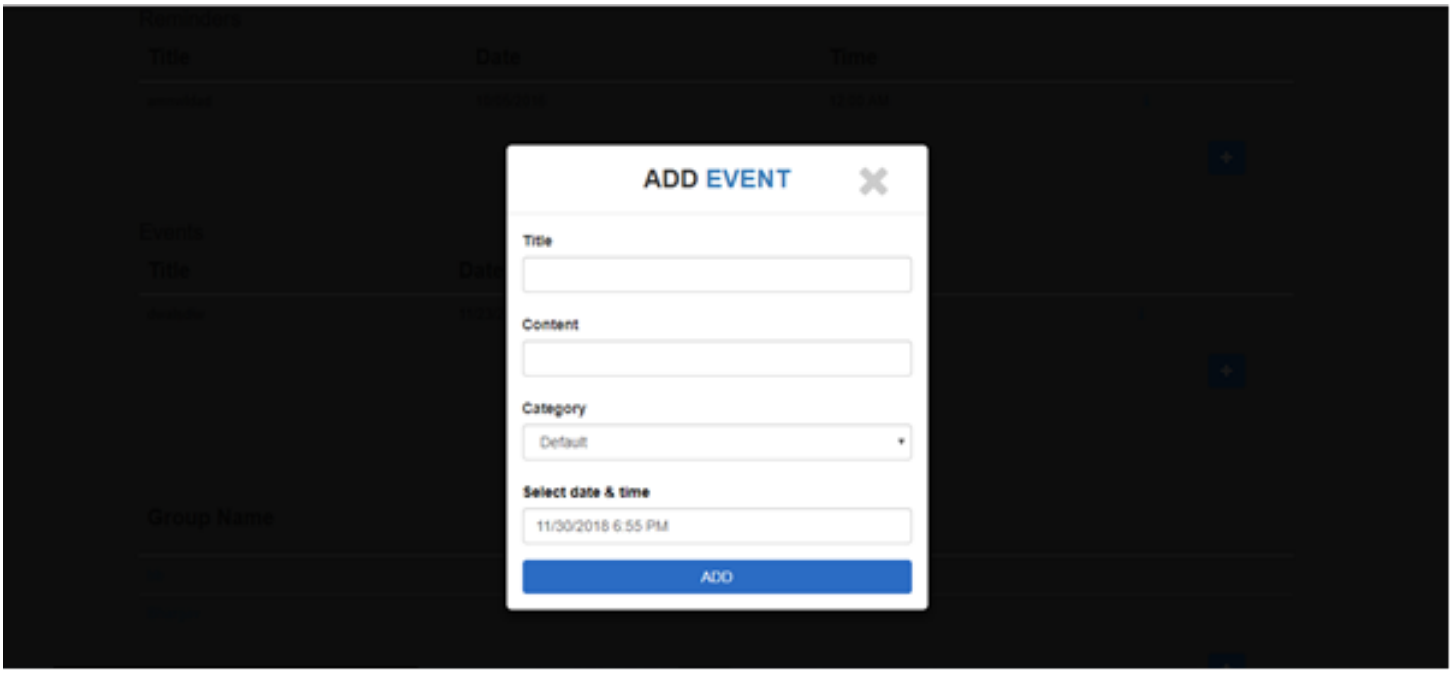

Figure 15 : Add Event – Modal(For User/Group)

7.5 Reminder/Event – More Details (User/Group)

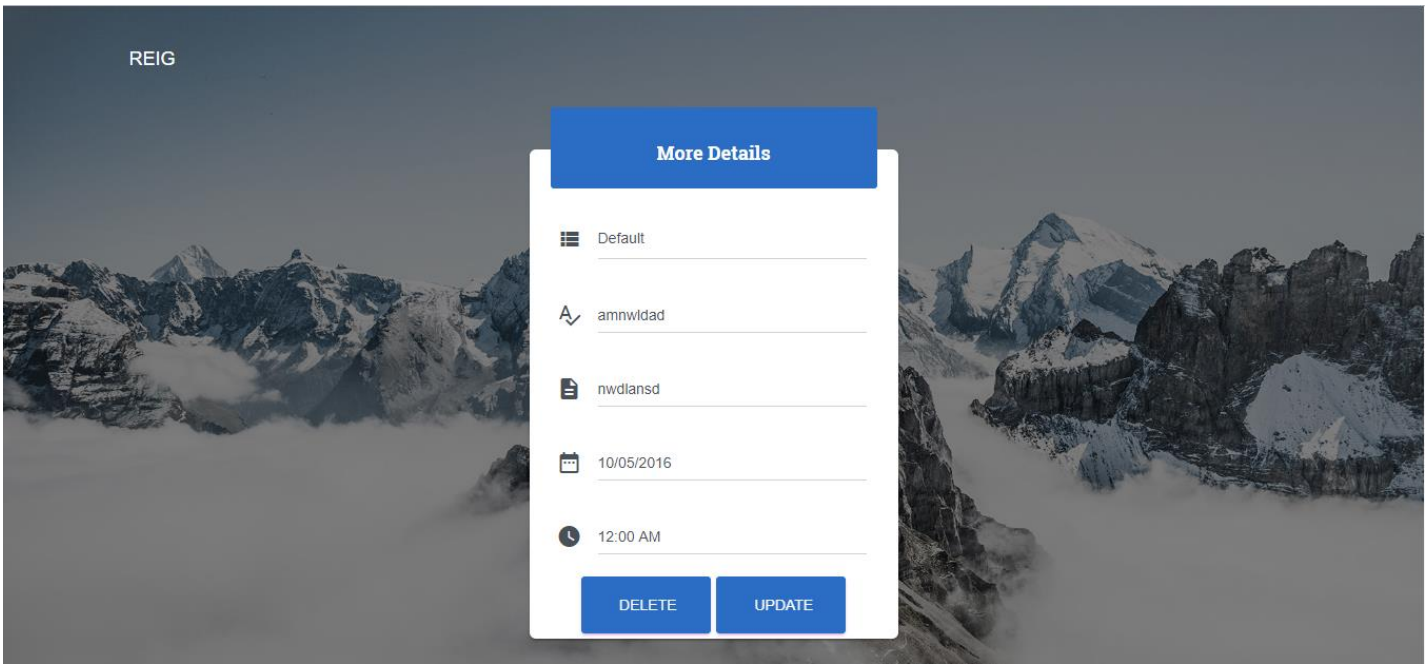

Figure 16 : Reminder/Event – More Details (User/Group)

7.6 Add Members (Group)

Figure 17 : Add Members (Group)

7.7 Add Group Details

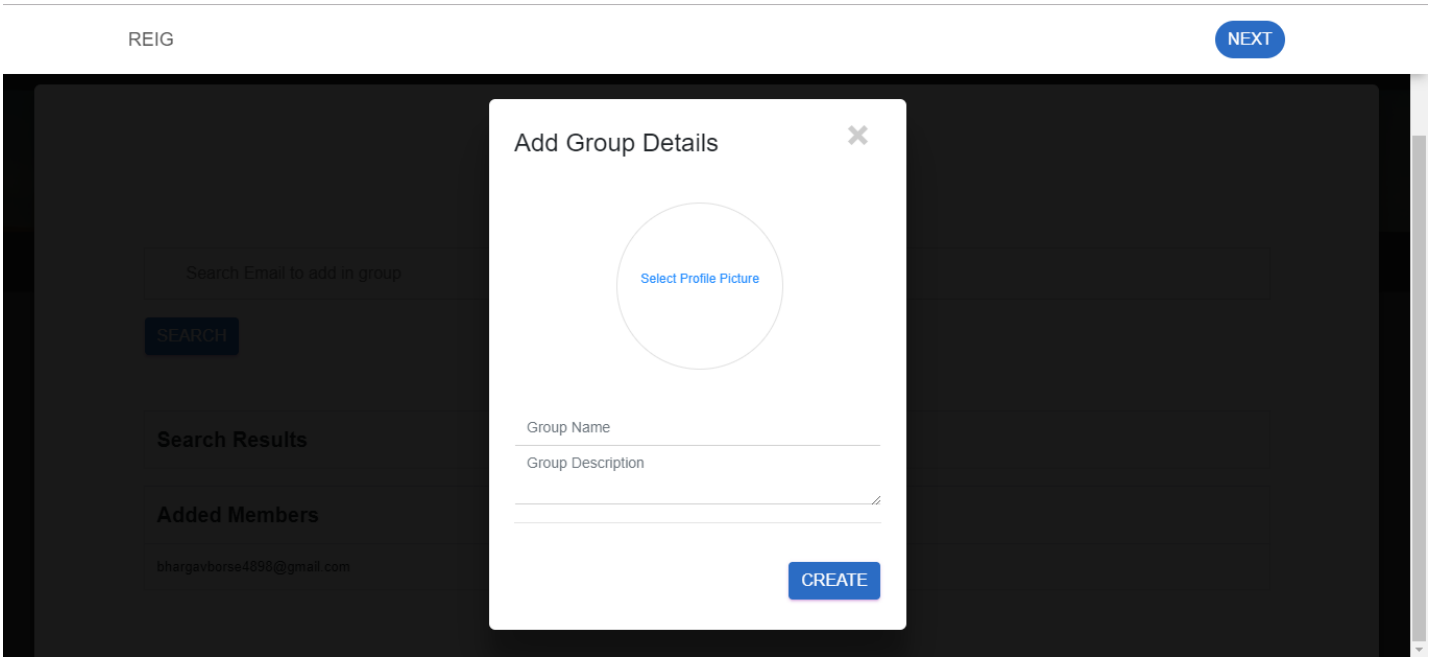

Figure 18 : Add Group Details

7.8 Group Index

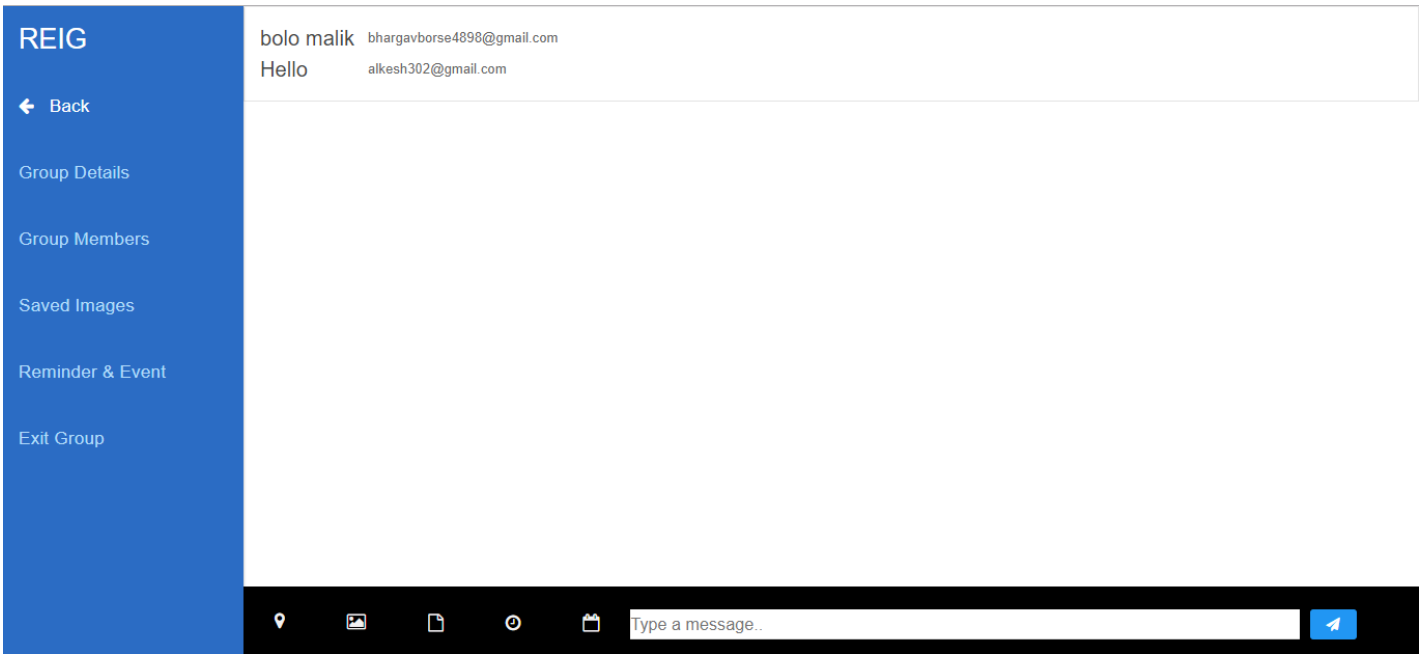

Figure 19 : Group Index

7.9 Group Details

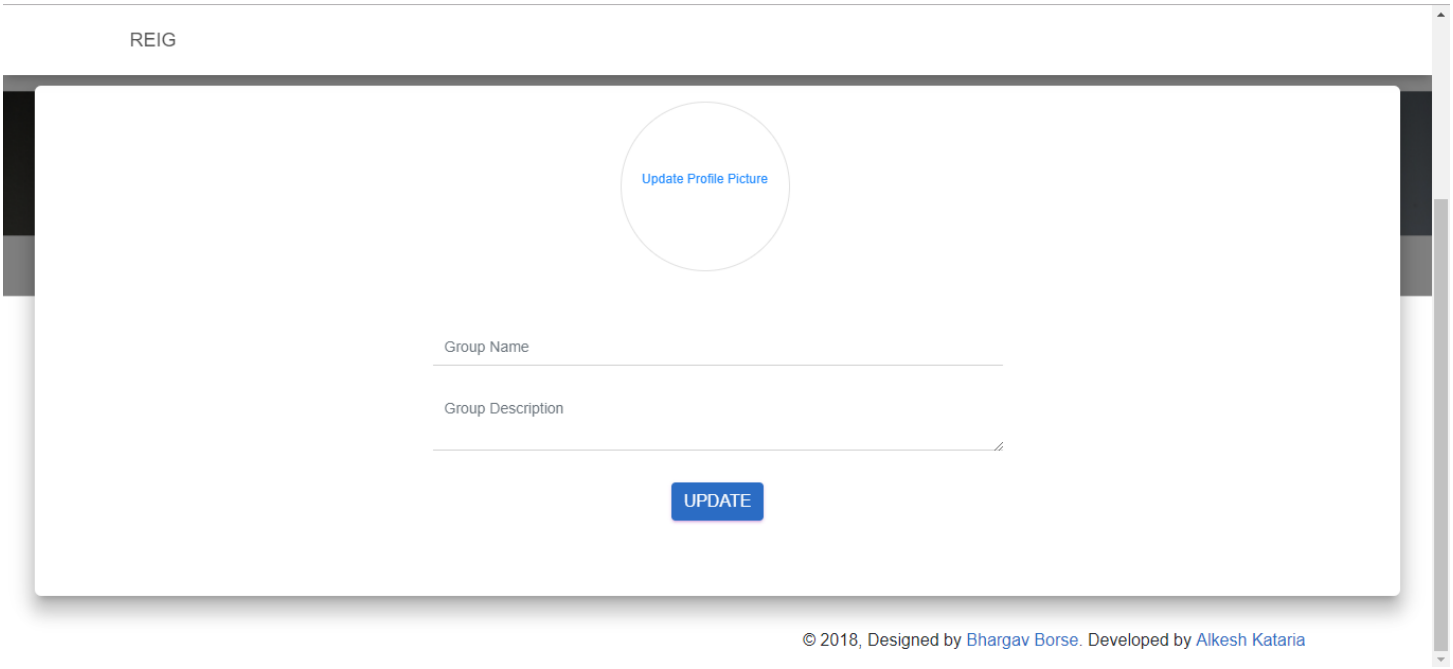

Figure 20 : Group Details

7.10 Manage Group Members

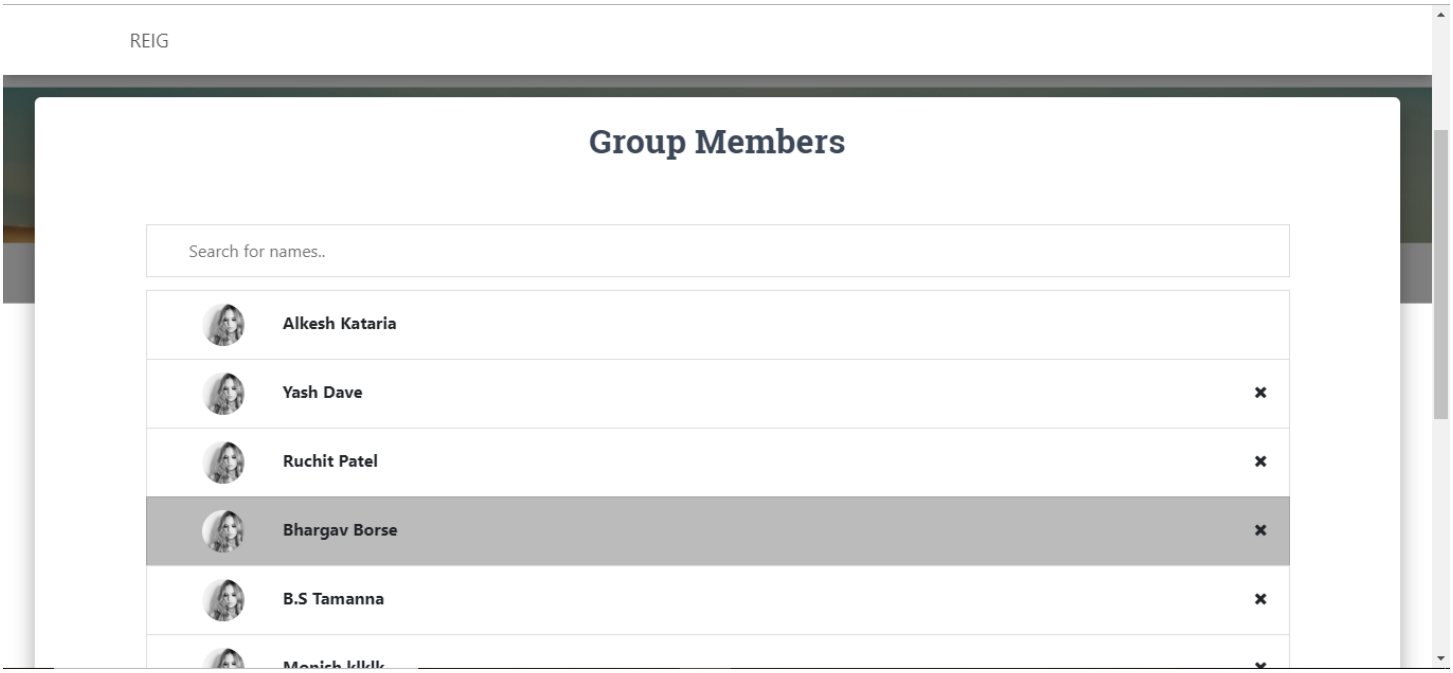

Figure 21 : Manage Group Members

7.11 Saved Images (User/Group)

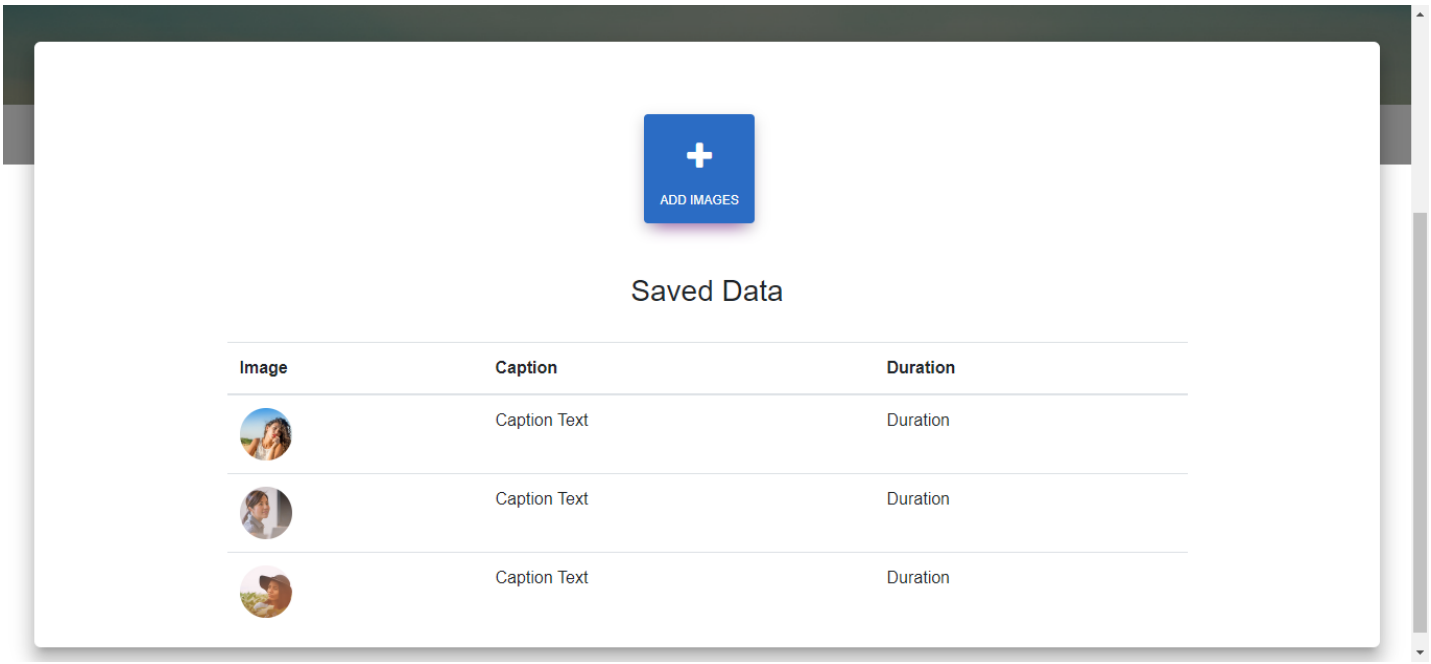

Figure 22 : Saved Images (User/Group)

7.12 Add Image – Modal (User/Group)

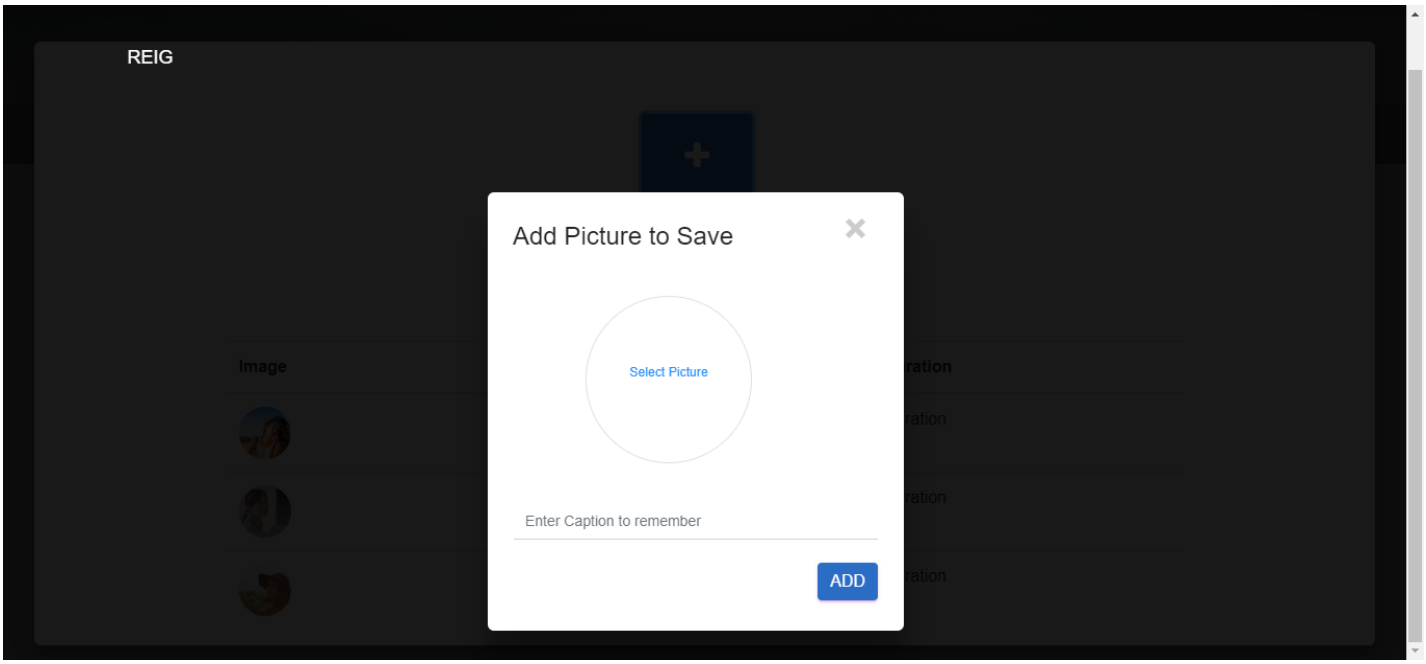

Figure 23 : Add Image – Modal (User/Group)

7.13 View/Update Reminder/Event (User/Group)

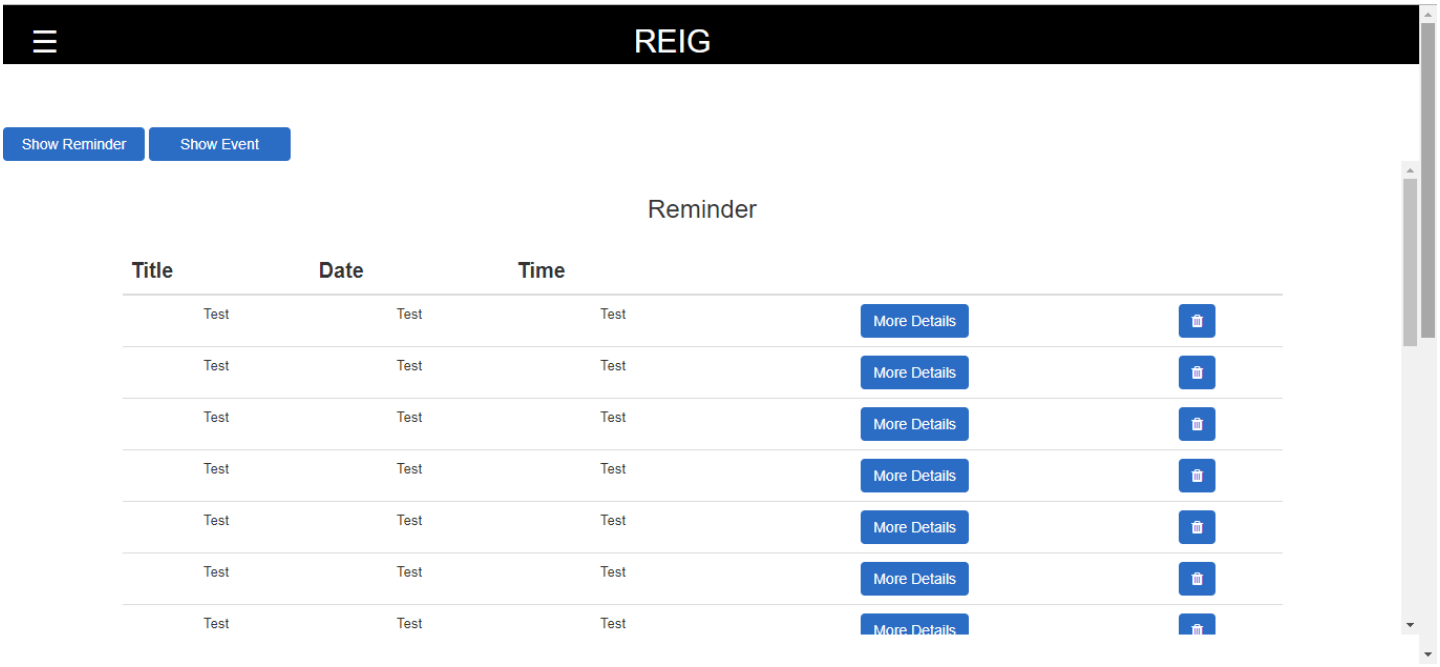

Figure 24 : View/Update Reminder/Event (User/Group)

7.14 Limitations and Enhancements

Limitations

1. The biggest limitation is without the use of internet; this application won't be able to carry out its operations. While using this application user will be to add reminder, create event, or to create groups, for carrying out such operations user is required to have internet connection.

2. When user carries out an operation since it's a real-time database, it takes time to save and load data from database. Sometimes, it takes longer than usual where user may leave interest to carry out further operations.

3. Unlike every other chat application where user can attach and send multiple files, this application doesn't have that feature, user is required to select single image to save individually or in group.

Enhancements

The enhancements targeted are:

1. To improve user experience by adding more features like attach multiple files at once.
2. To improve GUI and make more user-friendly.
3. To improve the performance of the system and reduce load times.
4. To work on the user's feedback and improve the overall experience.

7.15 Conclusion

The In-House Project provided us an opportunity to learn and develop our skills. It was an extra-ordinary experience. Many new things, languages, ways we came across and overall it was the required part that was given to us. This chance to improve ourselves and paring knowledge was the biggest we could've get. It led us towards exploring new languages and functions and also made us understand the importance of developing a project from scratch till the end.

This project led us towards the road of preparation for the industrial projects and as a foundation to develop some in the industry.

We came across many challenges but with the guidance of our faculties, we were able to win over each one of it.

7.16References

1. CSS. (2012, Dec 29). *CSS.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Cascading_Style_Sheets

2. Database, H. (2018, July 23). *Hierarchial Database.* Retrieved from TutorialsPoint: https://www.tutorialspoint.com/Hierarchical-Database-Model

3. Firebase. (2016, Oct 24). *Firebase.* Retrieved from HowToFirebase: https://howtofirebase.com/what-is-firebase-fcb8614ba442

4. HTML. (2015, April 2). *HTML.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/HTML

5. JavaScript. (2007, May 22). *JavaScript.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/JavaScript

6. JQuery. (2014, June 4). *JQuery.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/JQuery

7. JSON. (2015, Jan 7). *JSON.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/JSON

8. SDLC. (2008, March 21). *Software Development Life Cycle.* Retrieved from Guru99 - SDLC: https://www.guru99.com/software-development-life-cycle-tutorial.html

9. Testing, B. B. (2014, Nov 1). *Black Box Testing.* Retrieved from Software Testing Fundamentals: http://softwaretestingfundamentals.com/black-box-testing/

10. Testing, W. B. (2014, Nov 10). *White Box Testing.* Retrieved from Software Testing Fundamentals: http://softwaretestingfundamentals.com/white-box-testing/

11. WaterFall, T. (2014, July 28). *WaterFall Model.* Retrieved from TutorialsPoint: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm