MeetNotes

Project report submitted in fulfillment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering

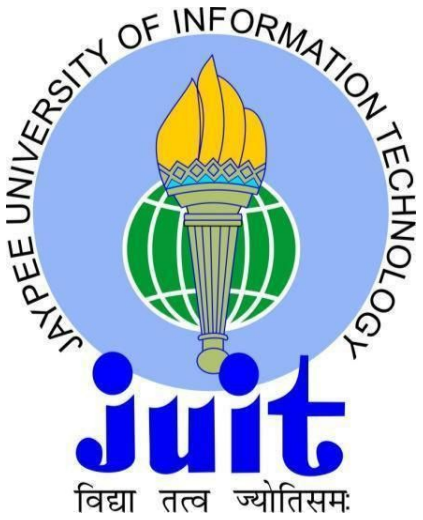
By

Rishabh Gupta (133204)

Under the supervision of

(Mr. Govinda Parashar)

To



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234,

Himachal Pradesh

CERTIFICATE

Candidate's Declaration

This is to certify that the work which is being presented in the report entitled **"MeetNotes"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from August 2016 to May 2017 under the supervision of **Mr Govinda Parshar** (Product Manager, Hashedin Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Rishabh Gupta, 133204

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

Mr. Govinda Parashar
Product Manager
Hashedin Technology

Dated:

ACKNOWLEDGEMENT

I owe our profound gratitude to our project supervisor **Mr. Govinda Parashar**, who took keen interest and guided me all along in my project work titled **— MeetNotes**, till the completion of our project by providing all the necessary information for developing the project. The project development helped us in research and we got to know a lot of new things in our domain. We are really thankful to him.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

About 15% to 50% of your time is spent on meetings, depending on your role and responsibilities. To make meetings more productive we can have a collaborative MeetNotes which is shared with everyone on that meeting. The notebook will have a public and a private area for note taking, automatically identify the action items, and link recurring meetings to always show the same notebook so that follow-up can be taken on the action items.

Chapter 1: Introduction

1.1 Introduction

In order to successfully run a project or an organisation, communication is very important to keep everyone on the same page and working for the same goal, moving in the right direction. Meetings are an essential tool for that.

1.2 Problem Statement

About 15% to 50% of your time is spent on meetings, depending on your role and responsibilities. To make meetings more productive we can have a collaborative MeetNotes which is shared with everyone on that meeting. The notebook will have a public and a private area for note taking, automatically identify the action items, and link recurring meetings to always show the same notebook so that follow-up can be taken on the action items.

1.3 Objective

The goal of this tool is to get out of the way and let people do what they are meeting to do - which is to discuss, brainstorm and achieve meeting objectives without being a distraction. It should almost be interface-less so that there is zero cognitive load for the participants. It should be behind the scenes and do the work. The best physical equivalent of this is a human co-ordinator, a person who is part of the meeting, takes meeting notes, sets the agenda, holds others accountable to the agenda and time etc.

1.4: Methodology

The system focuses on maintaining user's meeting. It displays user's upcoming meetings, previous meetings, summary of events occurred in the meeting, task assigned to individuals in the meeting, decisions taken, time taken etc. The goal of this tool is to get out of the way and let people do what they are meaning to do - which is to discuss, brainstorm and achieve meeting objectives without being a distraction. It should almost be interface-less so that there is zero

cognitive load for the participants. It should be behind the scenes and do the work. The best physical equivalent of this is a human co-ordinator, a person who is part of the meeting, takes meeting notes, sets the agenda, holds others accountable to the agenda and time etc.

1.5: Organisation

About HashedIn

HashedIn Technologies is a consulting firm, specializing in cloud, web and mobility platforms. It strive to provide our clients best ROI by diagnosing problems, crafting solutions, deploying under harsh constraints & seeing it through the finish line. It works with Enterprises & Start ups in the US & APAC. They are a team of senior architects and passionate developers in Bangalore, India. Founded in 2010, they have delivered over 100 man years of value to 100+ projects[1].

Chapter: 2 Literature Survey

At Hashedin we have conducted extensive research on the dynamics of how typical meetings are conducted in corporate and industries and a small summary of it is described below.

2.1 Problem with current style of conducting meetings

About 15% to 50% of your time is spent on meetings, depending on your role and responsibilities. To make meetings more productive we can have a collaborative MeetNotes which is shared with everyone on that meeting. The notebook will have a public and a private area for note taking, automatically identify the action items, and link recurring meetings to always show the same notebook so that follow-up can be taken on the action items.

Agenda

Often, meetings don't have a pre-defined agenda. They break out into brainstorming sessions, with a follow up meeting scheduled, without any meaningful decisions taken. The first step is to get acquainted with the meeting agenda. This will help you understand the purpose of the meeting and in what order, the agenda items will be discussed. Armed with this information, you are better equipped to take meeting notes.

Running Off-Topic

Someone raises a topic and then an in-depth discussion starts taking everyone in directions which isn't intended, implying the actual meeting topics aren't addressed.

Time Allocation to Agendas

Everyone has opinions about trivial and small items leaving not enough time for important large items.

Starting Late

People joining late is a bad habit and should be corrected. Even if someone is 5 mins late, it wastes 5 mins for everyone else.

Ending Late

Often meetings run longer than scheduled. That is not a good use of time and / or a lack of planning. The effect cascades, causing delays in next meeting and so on.

Low Participation

Is everyone required in the meeting and participating? Often we find only few folks hogging all the meeting time.

Meeting Templates & Best Practices

Meetings should be following templates and best-practices, which are organisation specific e.g. for Feedback, 1:1, Scrum, Status etc.

Decision Making

To decide on a topic, attendees go round-and-round without ever coming to a conclusion and consensus or even agreeing that there isn't enough data point to decide right now.

Meeting Notes Scribe

Someone has to take meeting notes. Documenting action items, decisions that are taken. Otherwise they are quickly forgotten.

Reminders before Next Meeting

Often people forget what the decisions and agenda items were from previous meeting before coming into the next meeting. That wastes time and means less things are done.

Missing a Meeting

If someone couldn’t attend a meeting, they would still like a summary of what happened and was discussed so that they are up-to-date before the next meeting.

Measurable & ROI

How productive are your meetings? Are they a good use of time? Organisations and people don’t ask if we are spending too much or too little time in meetings.

2.2 Building a parser using PEG.js

From the above discussion it is certainly clear that we need a way to parse the meeting notes noted down in order to extract the action items from them and hence give a comprehensive summary to everyone. For this purpose we have conducted research and came up with a solution, build a parser which is very identical to building a compiler using lex and yacc that is able to parse the notes on the client side reducing the load and cost of server and then the client will call an API endpoint to store the summary on the backend. For this we have chosen PEG.js a little bit of description of which is provided below.

A parsing expression grammar, or PEG, provides a means of defining a formal grammar from which you can generate parsers, which makes them perfect for implementing your own mini-Domain Specific Languages (DSLs).
PEG.js is a good way to get started with writing a parser, since it has an interactive online editor for playing around with while you find your feet. Since it outputs a parser written in JavaScript, it's easy to create a simple web page to use your DSL, or to implement command-line tooling with Node.js.

Characteristics of defining the rules are found to be:

- PEG.js rules have a name, an optional description and a body which starts with =
- Character matching is done with RegExp-like [ ] syntax, as is specifying how many times any item in the body must be matched using a trailing quantifier.
- Any item in the body can be labelled by preceding it with a label name and a colon.
- By default a rule will return everything that's matched. To customise this, JavaScript code can be placed in { } braces at the end of the rule, with matched labelled items available as variables.

A single component property looks like:

- Literal characters are placed in " "quotes.

- The rules we've defined so far are used by name — for example the whitespace rule is used to parse any whitespace surrounding the colon literal.
- ( ) parens are used to group rules — note that the entire type: rule is optional and if matched, we only return the type part if that rule is matched.

To support multiple component properties, we need to recurse — this is typically where you might start getting yourself into trouble with a RegExp-based approach, with more fun in store for you once you need to try to match opening and closing brackets and quotes

With PEG,js, we define a new rule which does the recursion by defining a list of component properties to be a single property followed by zero or more properties preceded by a comma

In order to set up PEG.js to the latest version of javascript ES6 (EcmaScript 6), some special steps were needed to be taken care of. That and the steps to setup peg.js grammar has been described below

Parser Setup

1. Go to PEG.js online console

2. Add the grammar to Section 1.

1 **Write your PEG.js grammar**

```
/* Main Rules */

parsedNotes = blanks* PN:notesList*{
    return PN[0]
    }

noteItem =  T1:note* AT:assignedTo+ T2:note* S:status{
    return {"type": "todo", "assignee": AT[0], "assigned_to": AT[0], "note": AT.slice(1)+" "+T1+" "+T2, "dueDat
    }
    / T1:note* AT:assignedTo+ T2:note*{
    return {"type": "todo", "assignee": AT[0], "assigned_to": AT[0], "note": AT.slice(1)+" "+T1+" "+T2, "dueDat
    }
    / D1:note* DK:decisionKeyword+ D2:note*{
    return {"type": "decision", "assignee": "-", "assigned_to": "-", "note": D1+" "+DK+" "+D2, "dueDate": "-",
    }
    / N:note{
    return null
    }
    / N:rawText{
    return null
    }

rawText = WL:rawWordList space*{
    return WL.join(' ')
}

note = WL:wordList space*{
    return WL.join(' ')
}

assignedTo = space* "@" N:name space*{
    return N.join('')
}

status = space* "#" S:(sAlphabets)+ space*{
    return S.join('')
}

decisionKeyword = space* "#" S:alphabets+ space*{
    return S.join('')
}

/* Constants */
chars = [a-zA-Z0-9!$%&*=()/\\t,-.'":_+;'~`?<>|^]
space = [ \t]
```

Parser built successfully.

3. Test the grammar in Section 2.

2 **Test the generated parser with some input**

```
@saral Testing parser
```

Input parsed successfully.

4. Check the output below Section 2.

5. Download the parser from Section 3 with the option **Optimize: Parsing Speed**

6. Remove the following contents from the downloaded file(parser.js)

```
module.exports = /*
 * Generated by PEG.js 0.10.0.
 *
 * http://pegjs.org/
 */
(function() {
  "use strict";
```

7. Replace the following(at the end of file) with `}`

```
return {
    SyntaxError: peg$SyntaxError,
    parse:       peg$parse
  };
})();
```

8. Export the `peg$parse` function

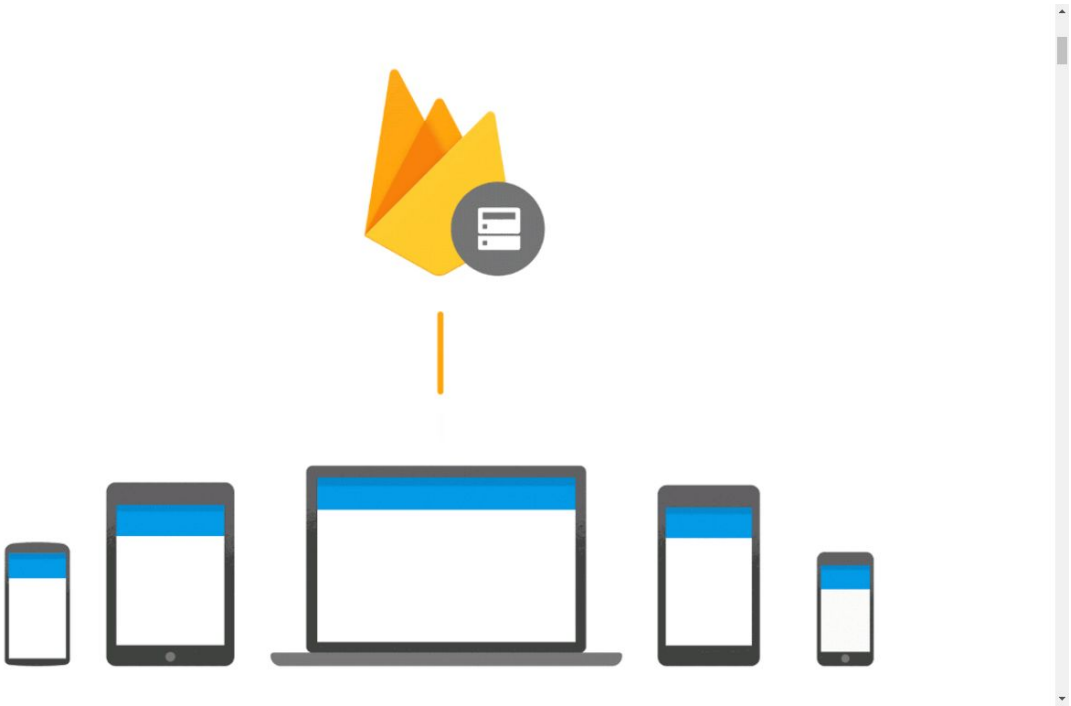
```
function peg$parse(input, options) =>
export default function peg$parse(input, options)
```

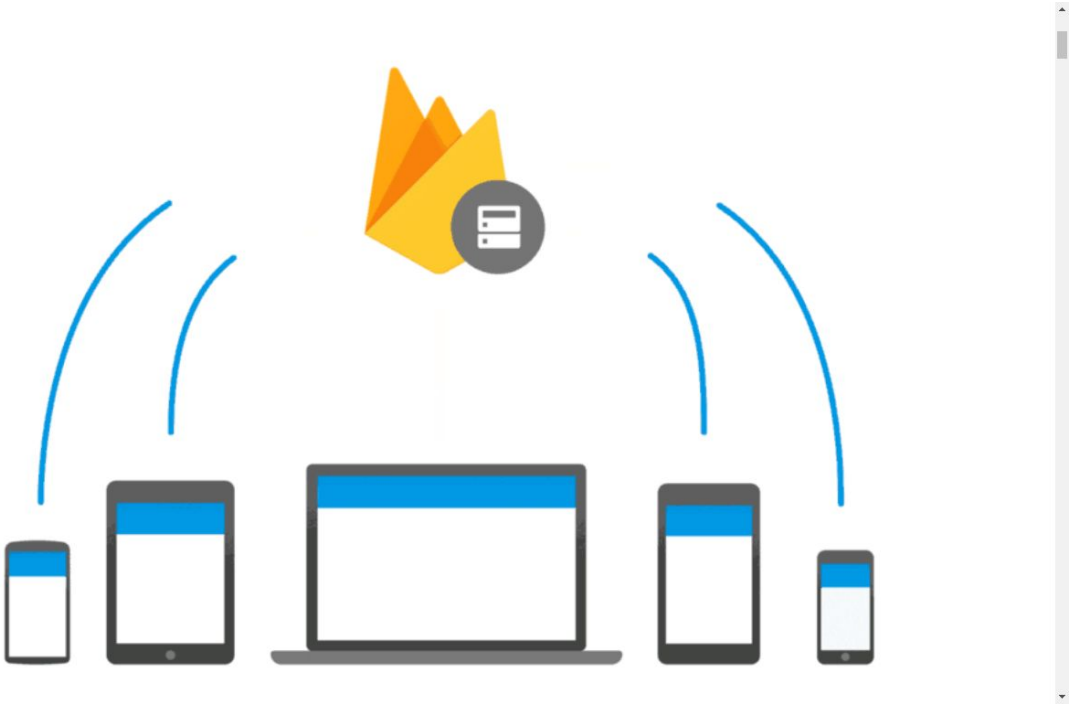
9. Paste the contents in `src/parser.js`

2.3 Using Real time Databases for building the collaborative editor

As discussed in the previous section of the report that we decided to build a collaborative editor on which all the members of the meeting can collaborate more efficiently. For this we found that real-time database can be useful.

The Firebase Realtime Database is a NoSQL cloud database that stores data in the form of JSON and synchronizes realtime data with every connected device automatically in seconds. Supported when Offline (data will be stored locally until it comes back online, it will automatically sync data), including Security Rules, we can design conditions for both read and write access to Android, iOS and Web.

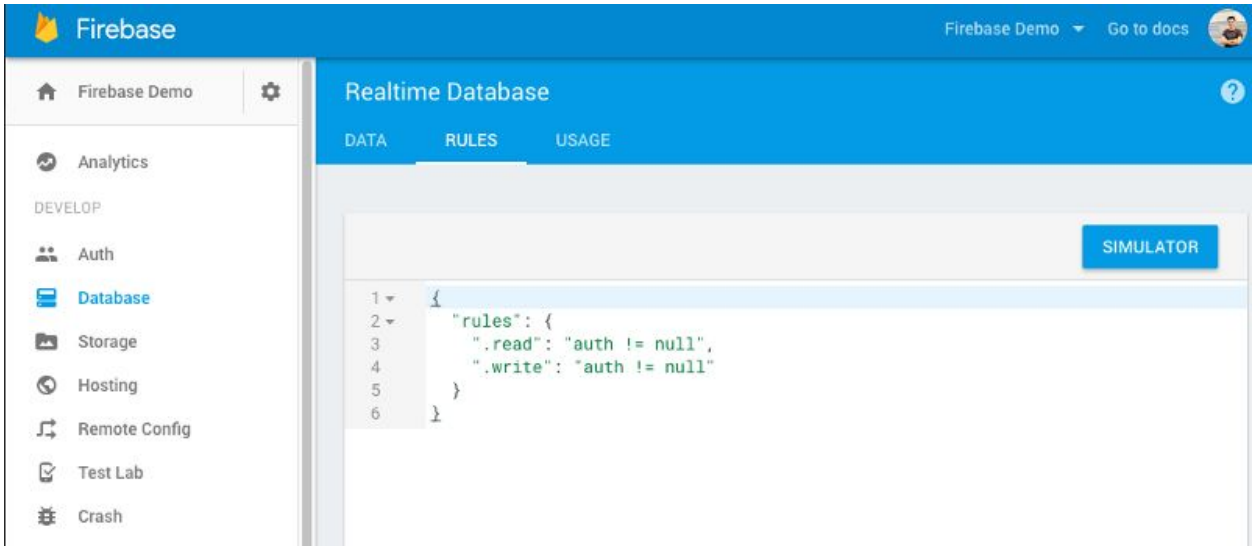




Development of Firebase Realtime Database is divided into 5 parts.

1. Set up Firebase and Realtime Database SDK
2. Writing data
3. Reading data
4. Enabling Offline Mode
5. Security & Rules

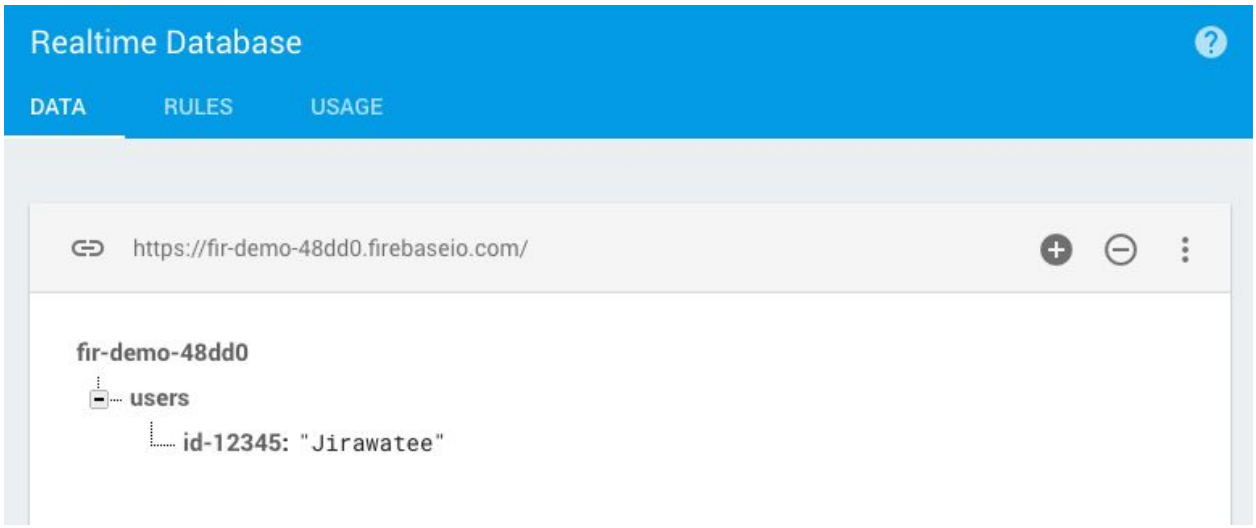
Data access for Firebase Realtime Database both read and write. Authentication through Firebase Authentication . But before we can understand this article without reading Firebase Authentication. We will make it publicly accessible by accessing the Firebase Console into the project. Then select the Database menu and select the tab called RULES will find the look of this.



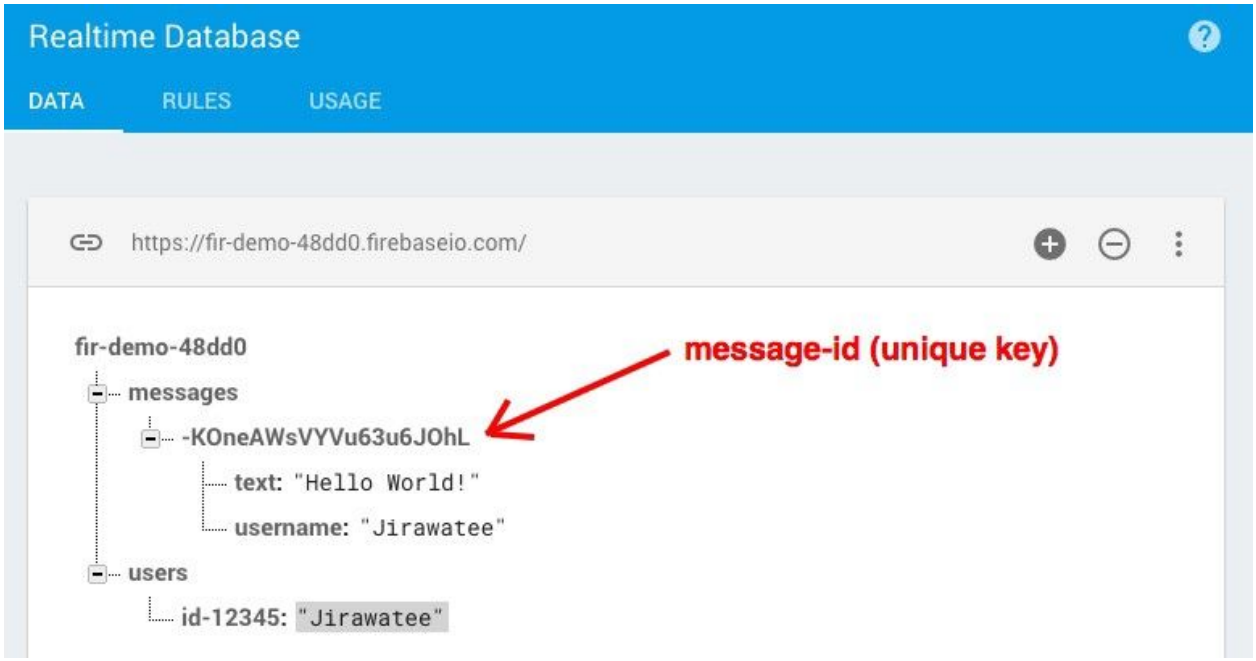
Part 2: Write data

Write, update or delete data in Firebase. Realtime Database supports many types of String, Long, Double, Boolean, Map <String, Object> and List <Object>.

1. setValue () writes or updates information to the path we refer to, such as users / <user-id> / <username>



2. Push () adds a series of data. Here, I will create a friendly object model, FriendlyMessage, which will contain the text and username. By pushing it, Firebase will create a unique key of the data set. For further reference, such as messages / <message-id> / <data-model>



3. updateChildren () is a write or update of some data (some key) according to the path we refer to. Without replacing the whole set. And can simultaneously do multiple objects.

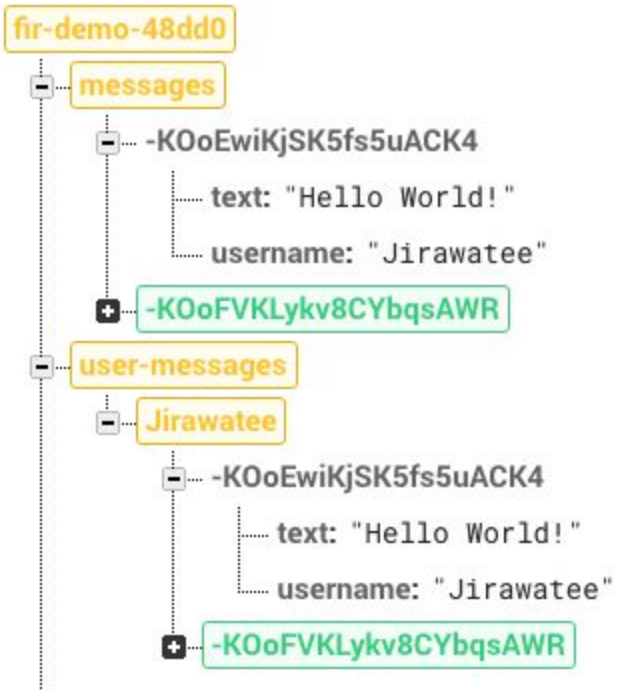
The colors shown in the console indicate the different status:

Yellow: Show

Green Update : Show

Red: Show

Blue Delete : Show Movement.



Part 3 Reading (Read)

Start with variable declarations. DatabaseReference gets an Instance value and refers to the path we need in the database.

Reading data in Firebase Realtime Database is divided into 2 types according to Listener.

1. ValueEventListener

It reads data from start and reads every time all data changes under the path we refer to. The method is to use the object we refer to. AddValueEventListener with two callbacks OnDataChange will be called at startup. And is called every time the information under the path we refer to has changed.

The onCancelled will be called when the database can not be read.

2. ChildEventListener

It will receive information from the child's addition, transition, deletion, and relocation. The method is to use the object we refer to. AddChildEventListener with 5 callbacks

- OnChildAdded () will be called when a series of data is added to child.
- OnChildChanged () will be called when the data in child is changed.
- OnChildRemoved () will be called when the data in child is deleted.
- OnChildMoved () is called when the sort of data in the child occurs.
- OnCancelled () will be called when loading data from child is unsuccessful.

Chapter 3: Systems Development

3.1 TECHNICAL DESIGN

3.1.1 Introduction

This chapter include the features of the Technical Design document for the proposed project. This would mainly include the Process flow diagram.

3.1.2 Purpose

The purpose of this part of the document is to outline the functional design of the application.

3.1.3 Scope

The scope of this document is to describe Requests made to MeetNotes Application and information retrieved from the same and to provide the facility of MeetNotes and saving the information in database to use it in future.

3.1.4 Data Flow Diagram

DFD for MeetNotes has been drawn up to 2 levels as shown below:-

Level 0 DFD:

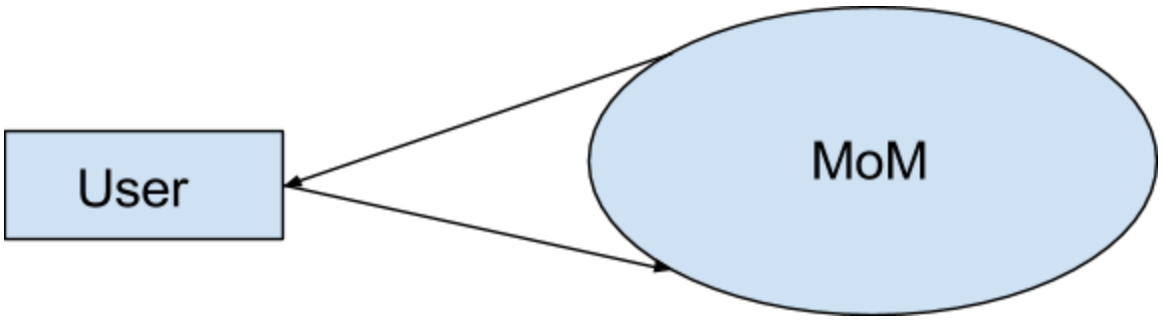


Figure 2.1:Level 0 DFD

Figure 2.1 shows level 0 DFD of MeetNotes system. It shows an external entity user and main MeetNotes process.

Level 1 DFD

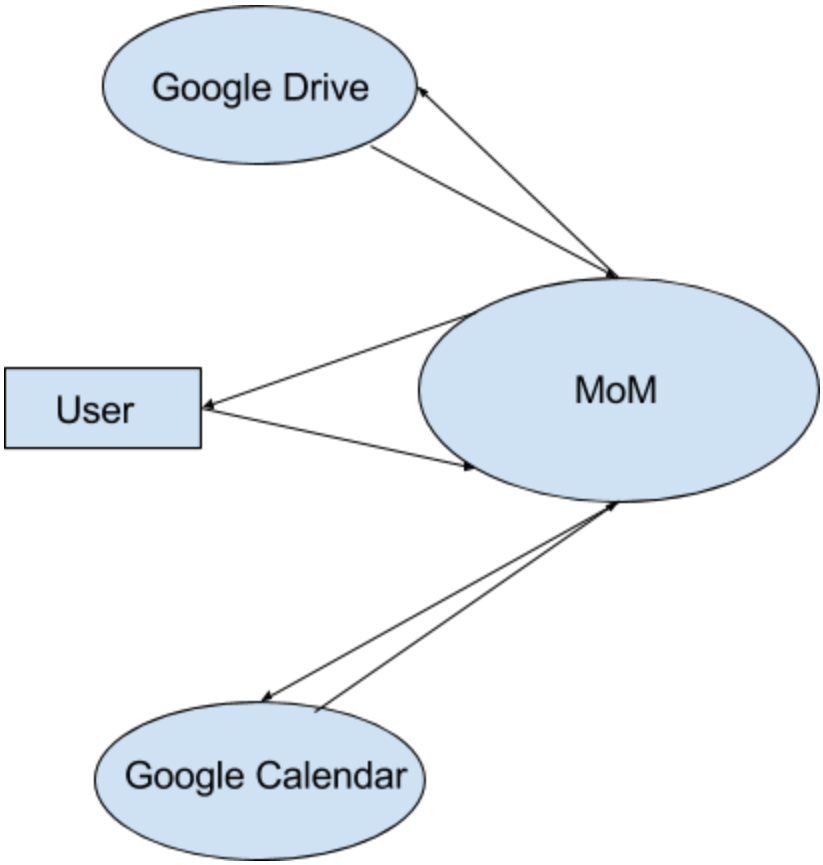


Figure 2.2:Level 1 DFD

Figure 2.2 shows level 1 DFD of MeetNotes system. It shows that user interacts with MeetNotes system which in turn interacts with Google Drive and Google Calendar.

Level 2 DFD:

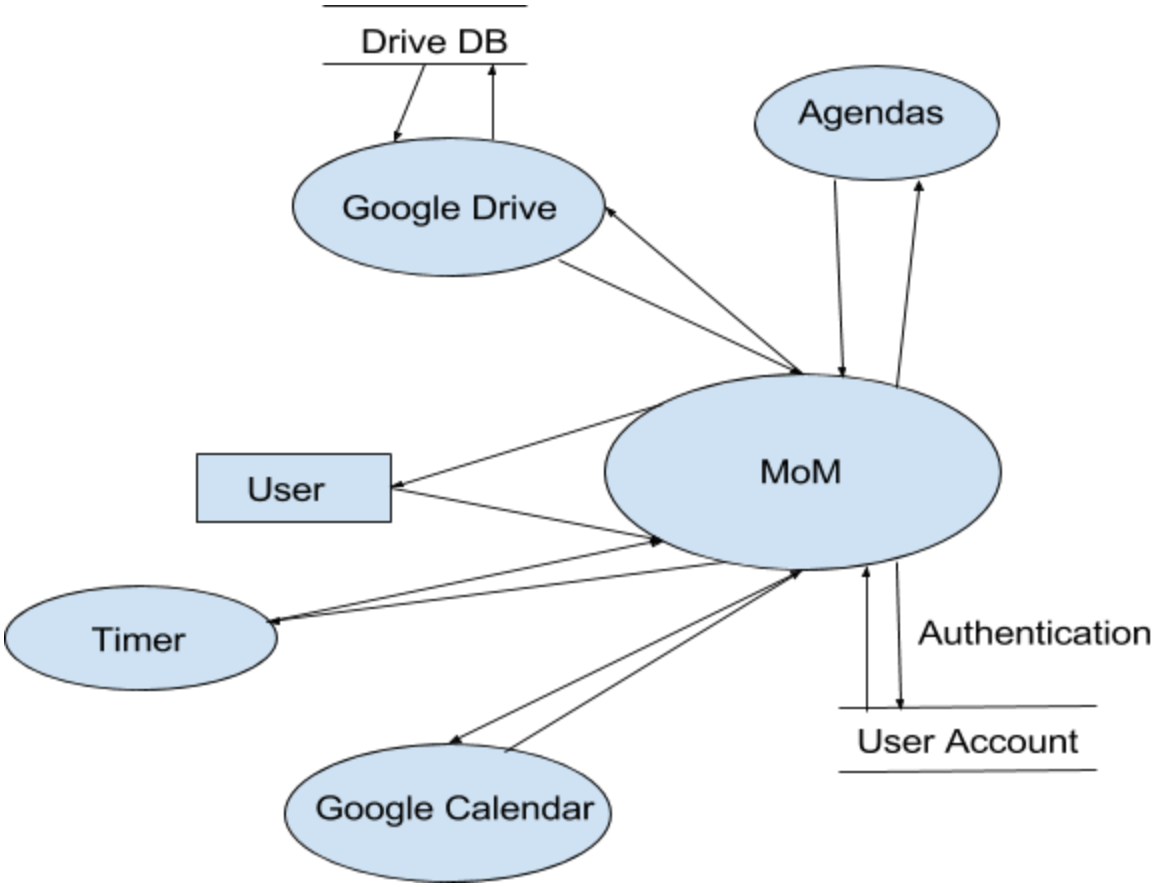


Figure 2.3:Level 2 DFD

Figure 2.3 shows a level 2 DFD of MeetNotes system. It shows how user interacts with MeetNotes which authenticates user using google authentication and fetches user's meeting from user's google calendar and stores user's notes in google drive.

3.1.5 Summary

This chapter included the detailed technical design which included DFD that was implemented in the project. The following chapter includes the step wise implementation of the project.

3.2 Technology used

As a consulting firm, specializing in cloud, web and mobility platforms, we have years of experience on a variety of tech stack based on which we have decided to choose the following Tech Stack. A small description of each is also provided below

Front End: ReactJS

IDE: Pycharm & Visual Studio Code

Backend: NodeJs, Django, MySql

3.2.1 ReactJs

HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. ReactJS has been developed by Facebook used to develop reusable components that build a complete UI experience. It works by creating a virtual DOM and updating only the parts that are needed to be updated as the new data comes from backend API endpoints. This is used by major companies including Facebook, Twitter. It allows developers to build applications that dynamically changes without refreshing the whole page. This provides a value in terms of user experience and better performance at the client's side

3.2.2 Pycharm

PyCharm is an Integrated Development Environment (IDE) utilized as a part of PC programming, particularly for the Python dialect. It is produced by the Czech organization JetBrains.[2] It gives code examination, a visual debugger, a coordinated unit analyzer, reconciliation with adaptation control frameworks (VCSes), and backings web advancement with Django.

PyCharm is cross-stageThe Community Edition is discharged under the Apache License,[3] and there is additionally Professional Edition discharged under a restrictive permit - this has additional elements.

3.2.3 Node Js

In programming improvement, Node.js is an open-source, cross-stage runtime condition for creating server-side Web applications. In spite of the fact that Node.js is not a JavaScript structure, a significant number of its essential modules are composed in JavaScript, and designers can compose new modules in JavaScript. The runtime condition deciphers JavaScript utilizing Google's V8 JavaScript motor.

Node.js has an occasion driven design fit for nonconcurrent I/O. These plan decisions intend to advance throughput and adaptability in Web applications with many info/yield operations, and in addition for continuous Web applications (e.g., ongoing correspondence projects and program games)

3.2.4 Django

Django is an open-source web system, written in Python, which takes after the model–view–controller (MVC) compositional example. It is kept up by the Django Software Foundation (DSF), an autonomous association set up as a 501(c)(3) non-benefit.

Django's essential objective is to facilitate the formation of complex, database-driven sites. Django underscores reusability and "pluggability" of parts, fast improvement, and the rule of don't rehash yourself. Python is utilized all through, notwithstanding for settings, records, and information models. Django likewise gives a discretionary regulatory make, read, refresh and erase interface that is produced powerfully through contemplation and arranged by means of administrator models[5].

3.2.5 MySql

MySQL is an open-source social database administration framework (RDBMS). In July 2013, it is world's II most popularly utilized RDBMS, and the most generally utilized open-source client–server show RDBMS.

MySQL is a well known decision of database for use in web apps, and is a focal segment of the broadly utilized LAMP open-source web appprogramming stack (and other "AMP" stacks). LAMP is short for "Linux, Apache, MySQL, Perl/PHP/Python". Free-programming open-source extends that require a full-highlighted database administration framework frequently utilize MySQL. Applications that utilization the MySQL database include: TYPO3, MODx, Joomla, WordPress,phpBB, MyBB, Drupal and other software

3.3 Project Implementation

3.3.1 Introduction

This chapter includes the detailed description about the implementation of the project.

3.3.2 System Overview

In order to successfully run a project or an organisation, communication is very important to keep everyone on the same page and working for the same goal, moving in the right direction. Meetings are an essential tool for that.The goal of this tool is to get out of the way and let people do what they are meeting to do - which is to discuss, brainstorm and achieve meeting objectives without being a distraction. It should almost be interface-less so that there is zero cognitive load for the participants. It should be behind the scenes and do the work.

3.3.3 System Configuration

It does not need any extra hardware or software requirement to run this system. Any browser and operating system is applicable to run this system.

3.3.4 User Interface Description

The following sections include descriptions of the user interface of the MeetNotes User:

I) Login Page:

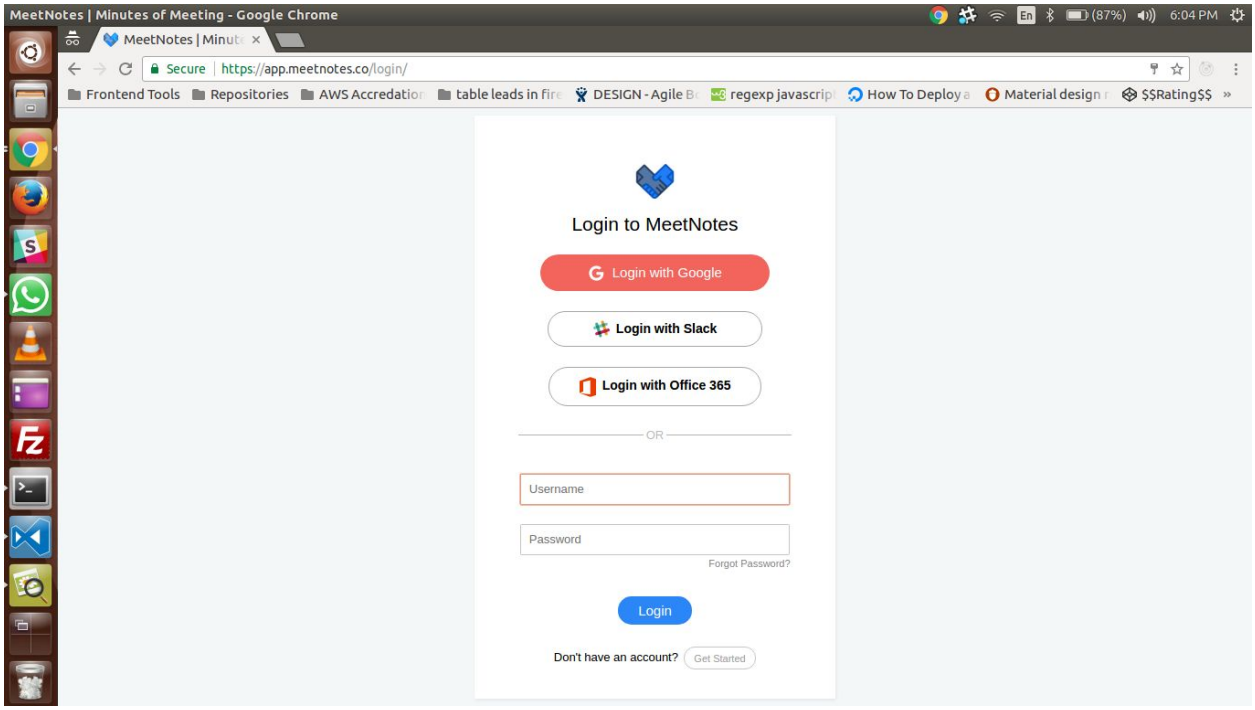


Figure 3.1: Login Page

Figure 3.1 shows a screenshot of a page which is used by all the users to login using their google account.

II) Permissions:

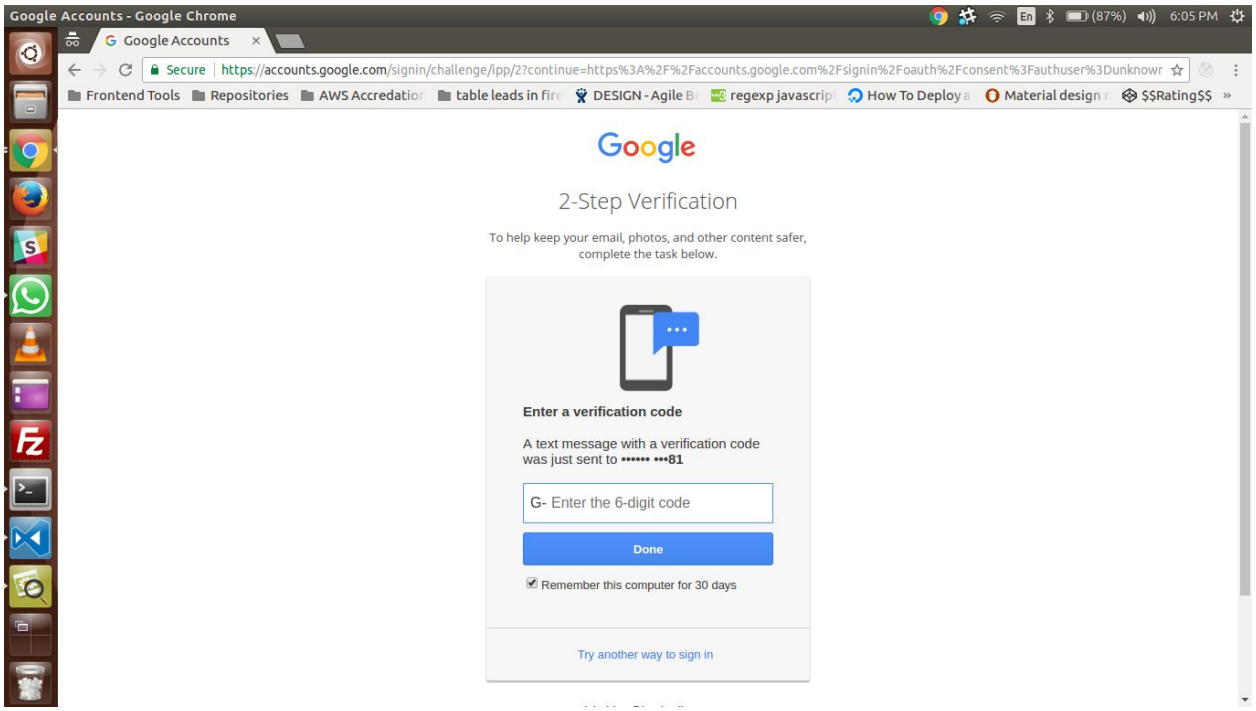


Figure 3.2: Permissions

Figure 3.2 shows a screenshot of a page which is used to ask for user's permission to access their basic profile info, their drive and their calendar in offline mode.

III) Dashboard:

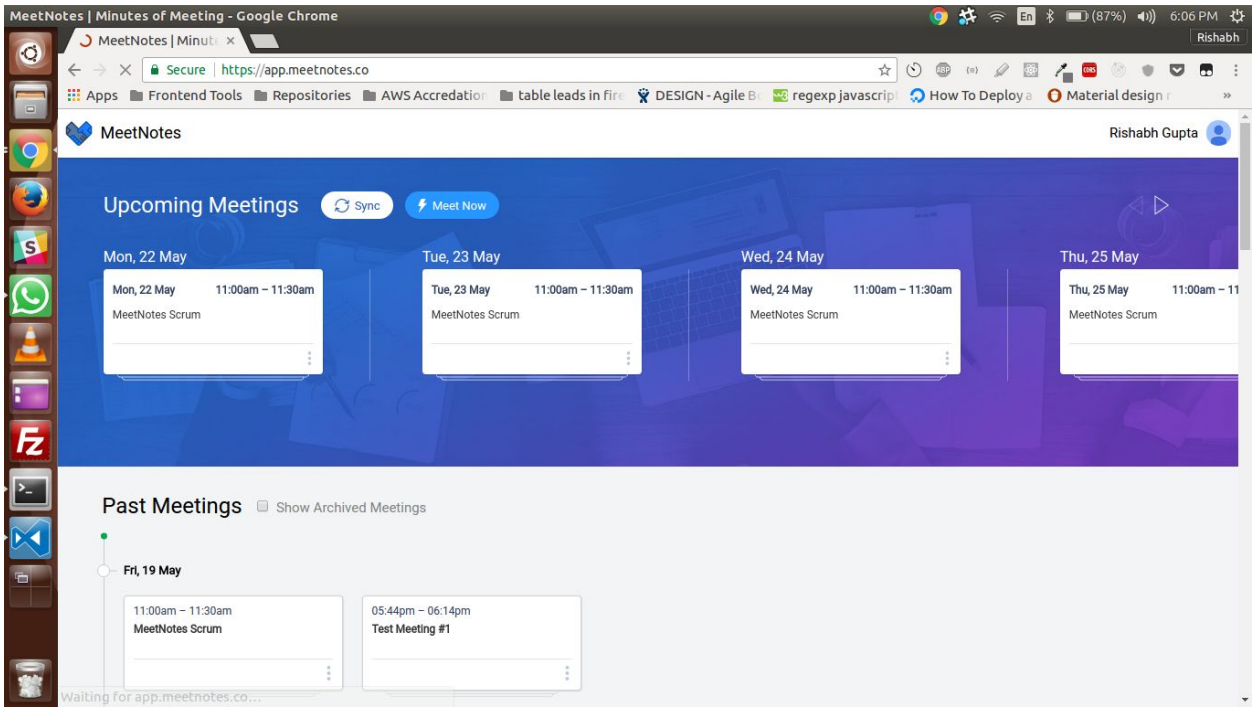


Figure 3.3: Dashboard

Figure 3.3 shows a screenshot of a page which is used displays user's next one week meetings.

IV) Add Agendas:

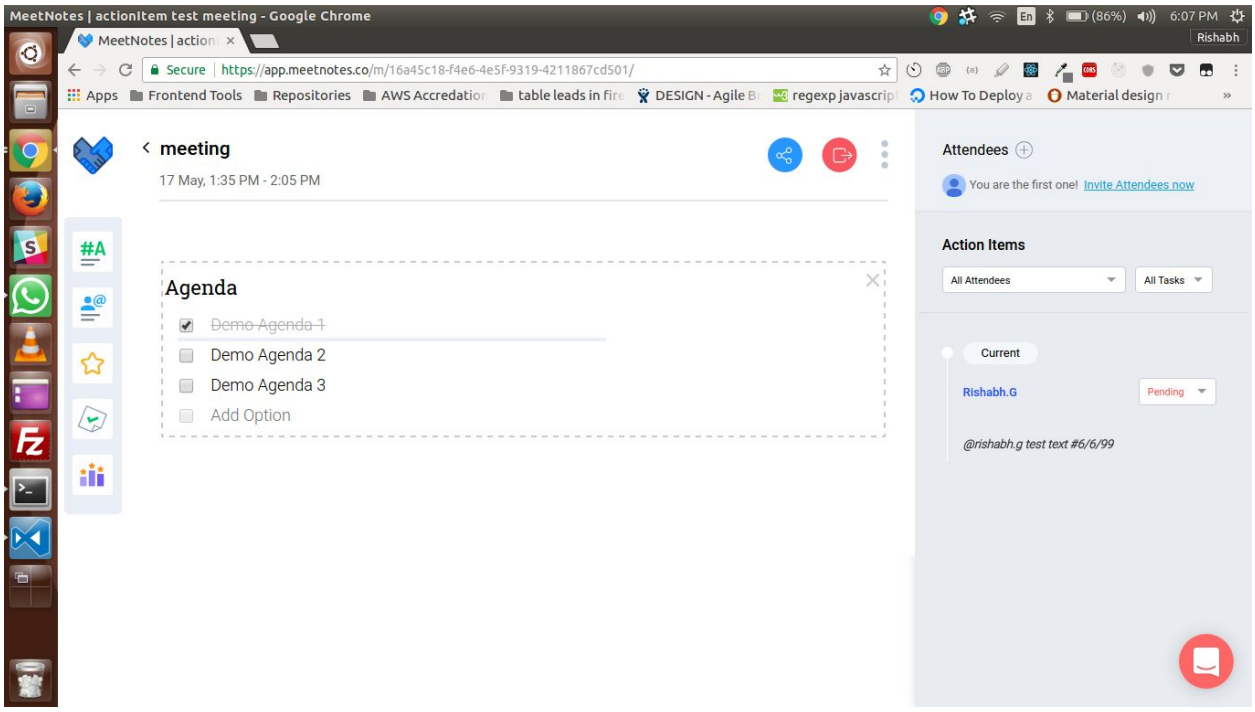


Figure 3.4: Add Agendas

Figure 3.4 shows a screenshot of a page which is used for adding of agendas for discussion for the meeting.

V) Ongoing Meeting:

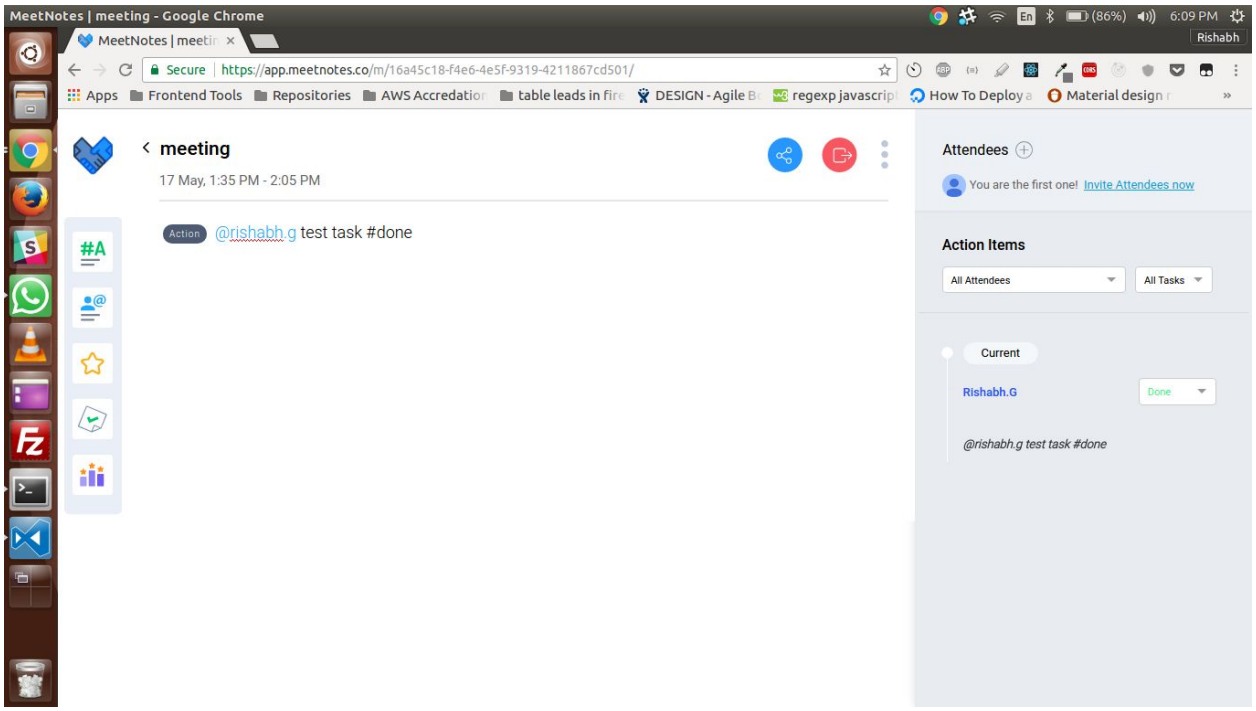


Figure 3.5: Ongoing Meeting

Figure 3.5 shows a screenshot of a page which is used for an ongoing meeting with timer, public notes and note items. Each agenda has its own allotted time.

VI) Private Notes:

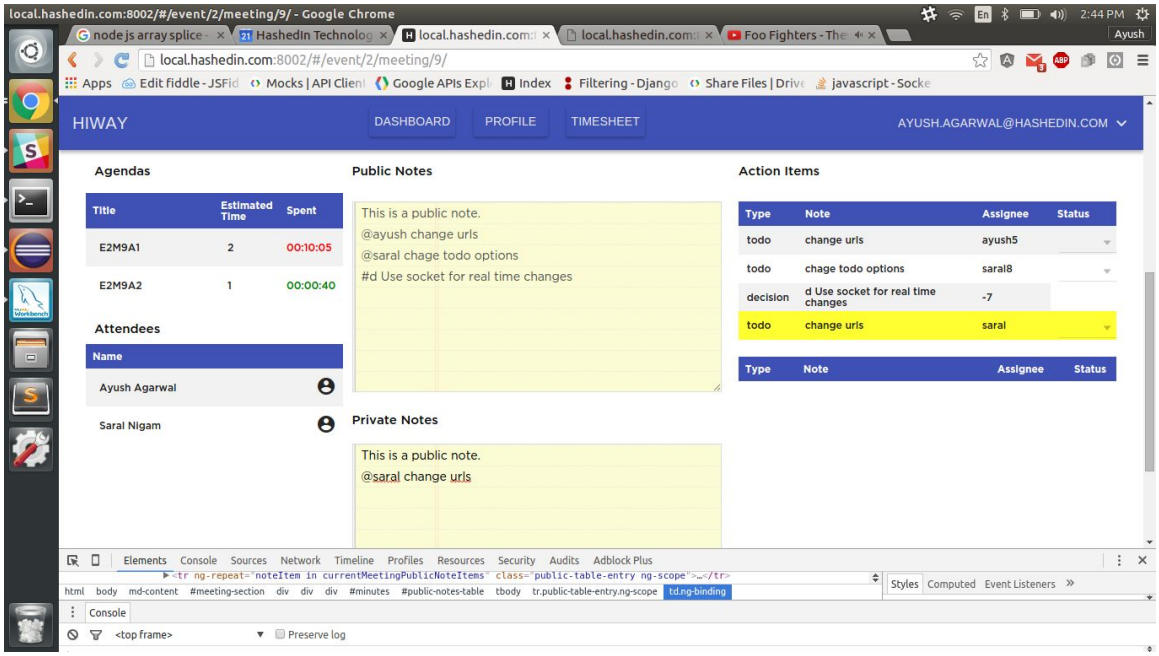


Figure 3.6: Private Notes

Figure 3.6 shows a screenshot of a page which displays user's private note items in yellow color which won't be visible to other users.

VII) Meeting Link:

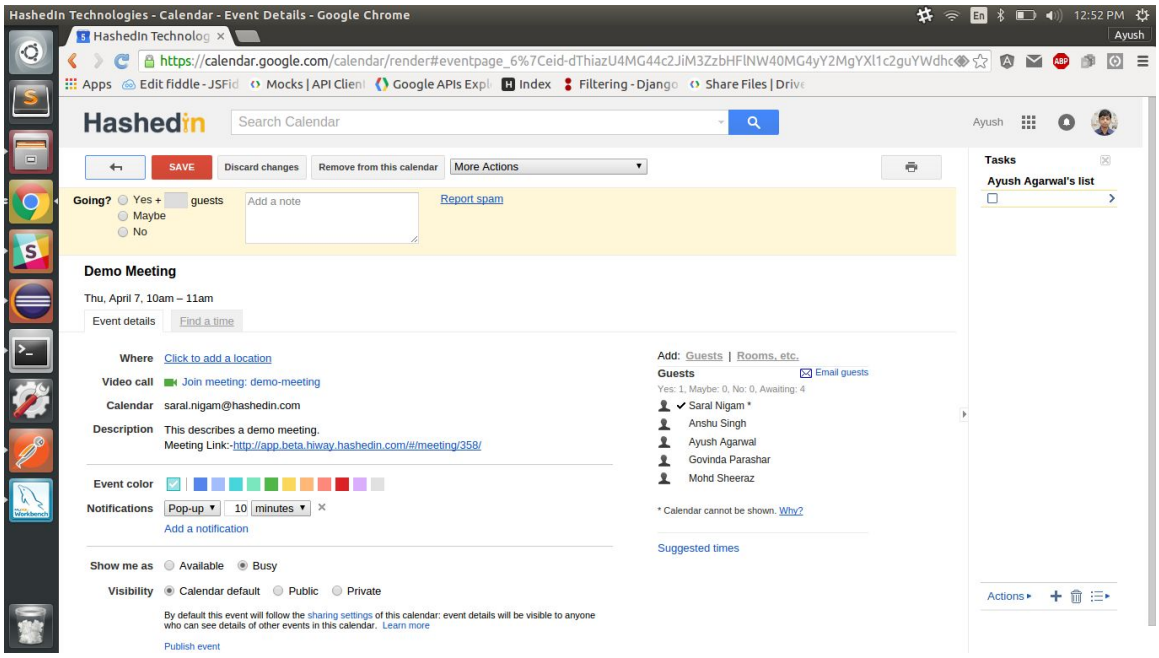


Figure 3.7: Meeting Link

Figure 3.7 shows a screenshot of a page which is used to display meeting link in user's google calendar.

VIII) Conducting Poll

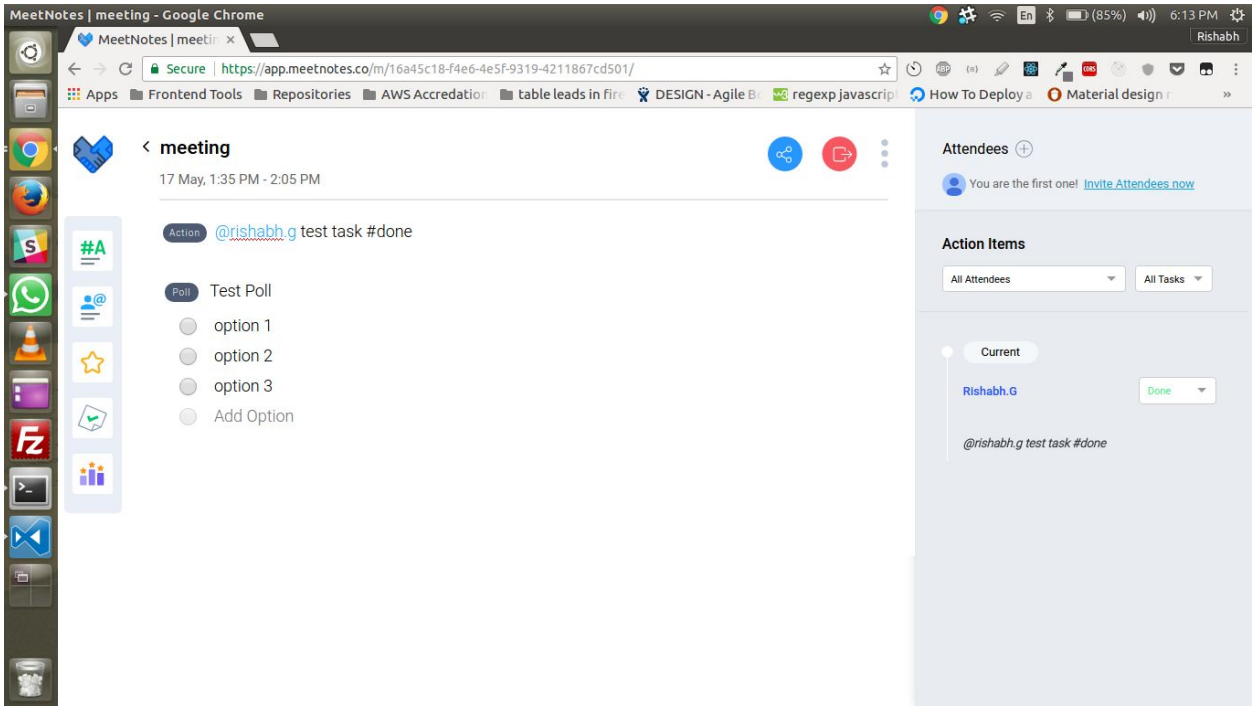


Fig: 3.8 Poll Widget

Figure 3.8 shows a screenshot of a sample poll being conducted in a meeting so that one can can get instant and anonymous feedback. As the options are marked by the attendees, a bar beneath the options come up whose length describes the favorability of that option based upon the average of the answers casted by the attendees.

IX) Rating Widget

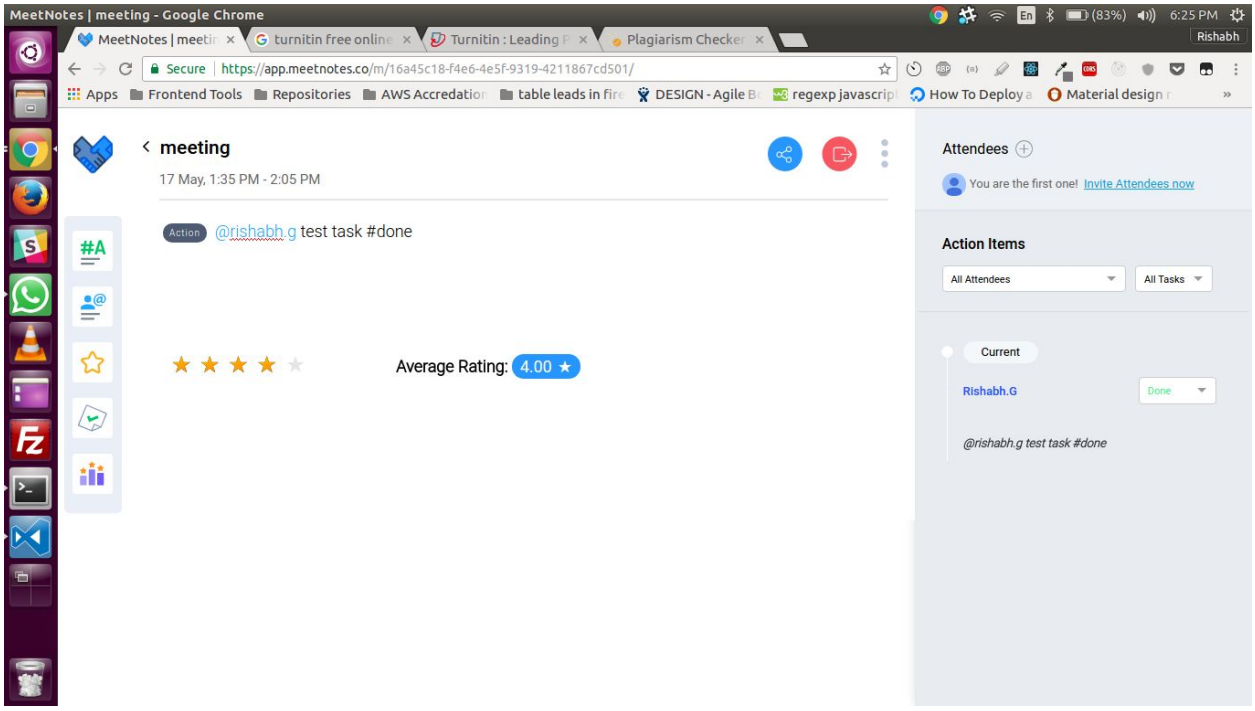


Fig 3.9 Rating Widget

Figure 3.9 shows a screenshot a sample rating being conducted in a meeting so that one can can get instant and anonymous feedback. It gives an average rating at any point of time depending upon the number of votes casted

X) Ranking Widget

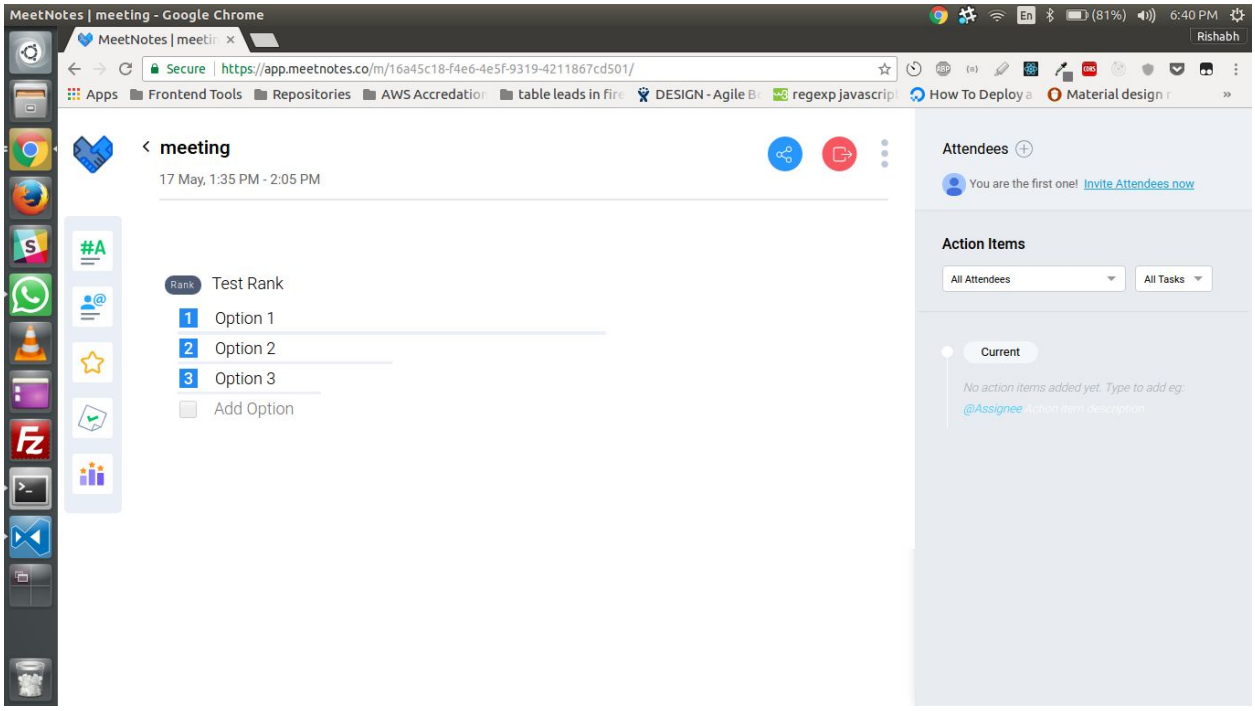


Fig 3.10: Ranking Widget

Figure 3.10 describes a sample ranking being conducted in a meeting so that one can can get instant and anonymous feedback. As the options are marked by the attendees, a bar beneath the options come up whose length describes the favorability of that option based upon the average of the answers casted by the attendees.

XI) Sharing Meeting Notes

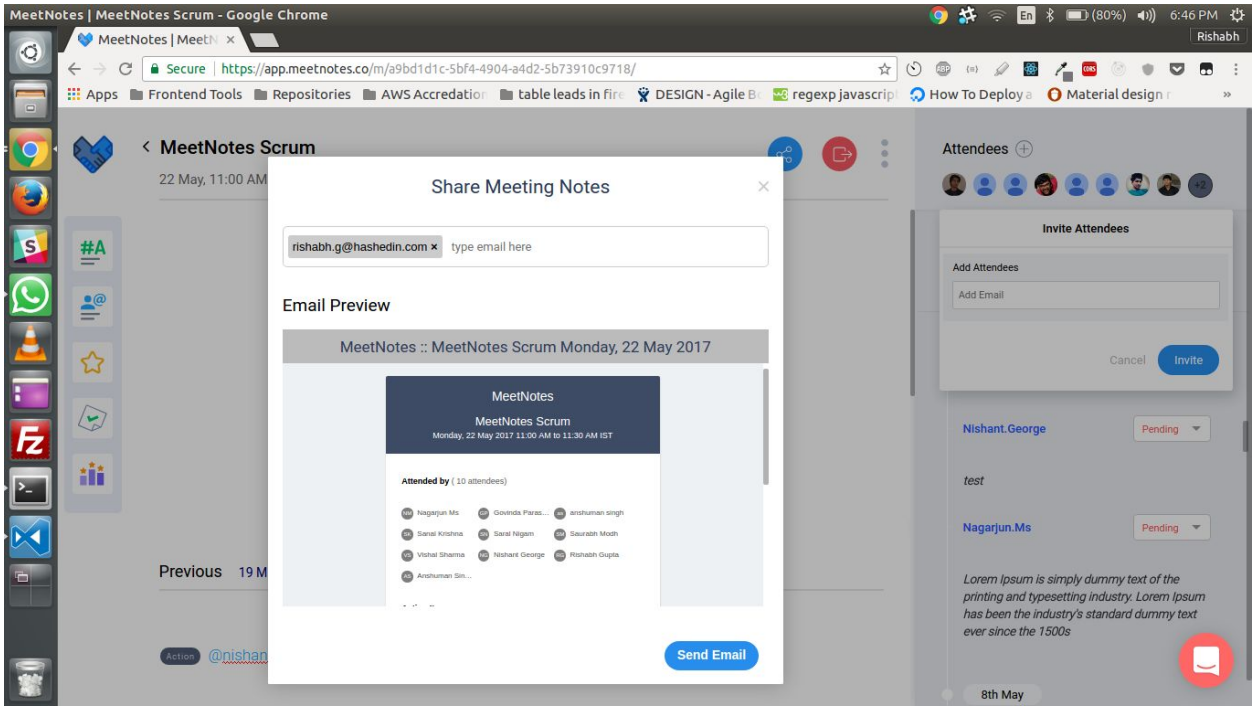


Fig 3.11: Sharing Notes

Figure 3.10 describes the sharing mechanism developed into meet notes. The notes once taken can be easily shared among all the attendees and any other by simply adding their email addresses. It gives a meeting preview that states the parsed summary as well as the full length notes and also gives access to the meeting page.

XII) Inviting more attendees to a meeting

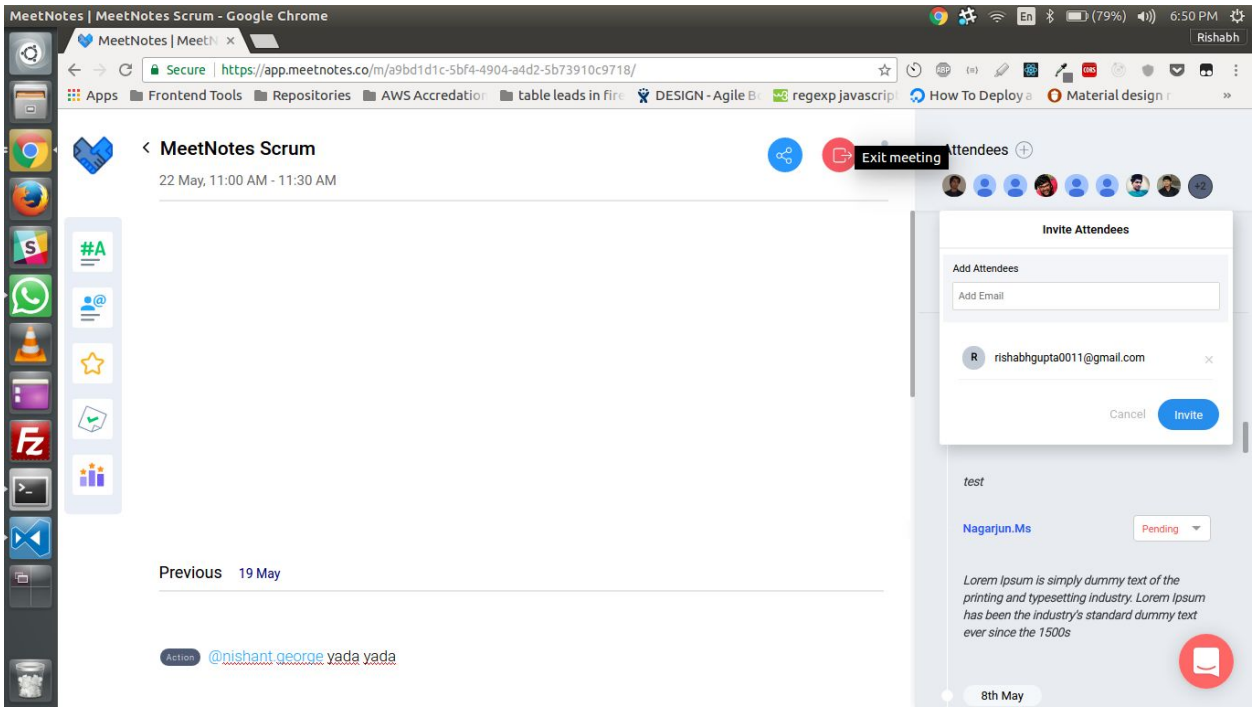


Fig 3.11: Invite attendees

Figure 3.10 describes the invite attendee flow developed into meet notes. Attendees can be easily added to a meeting, single or recurring, simply by adding their email address and clicking on invite. This will trigger an email that will contain the link to meeting.

3.3.5 Summary

In this chapter we described a detailed implementation of project with the user interfaces. In next chapter we will discuss the testing of this Application.

Chapter 4: Performance Testing

4.1) SYSTEM TESTING

System testing of software is virtually working conducted on a fastidious, perfect system to act with regard to the system's compliance by all of its specified.

It is literally much similar efficient test case writing. In test case exchange of letter you should devise the test scenarios & act with regard to use cases.

4.1.1) BLACK BOX TESTING:-

Black-box to a great extent working might be an approach of code testing that looks at the usefulness of affiliated application while not peering coordinated towards its internal structures or workings.

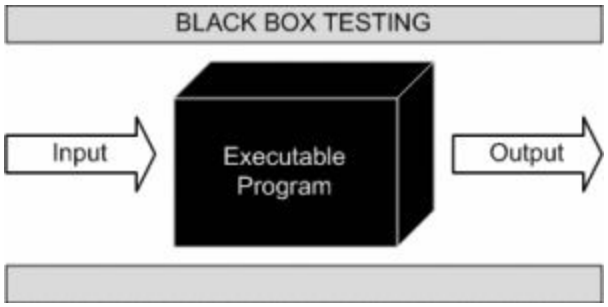


Figure 20: Black Box testing

4.1.2) UNIT TESTING:-

It is a product testing implies surrendered units of source inner voice, sets of a needed or in a superior way computer system modules together by the whole of associated approach information, utilization systems, and working techniques, are tried to check whether they are exist for pervade. Naturally, a famous can get a unit as the littlest testable protest of an application. In procedural programming, an unexpected may might be a full module, however not by the entire of standing it's extra unremarkably an individual utilize or technique.

The decision of unit testing is to conclude each protest of the framework and uncover that the individual parts are right.

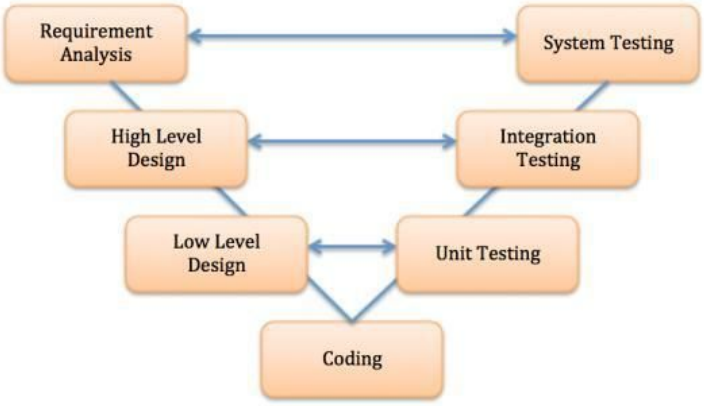


Figure 21: Unit Testing

4.2) Testing for Parser input and output

Test Case:

Input	@assignee test task
Output	Assignee:assignee ; Status:'-' ; Due Date:'-' ; Tags:[]
Expected Output	Assignee: assignee ; Status: '-'; Due Date: '-'; Tags: []

Test Case:

Input	@assignee test task #done
Output	Assignee: assignee ; Status: 'done'; Due Date: '-'; Tags:[]
Expected Output	Assignee: assignee ; Status: 'done'; Due Date: '-'; Tags:[]

Test Case:

Input	@assignee test task #9/11
Output	Assignee: assignee; Status: '-'; Due Date: '9/11'; Tags: []
Expected Output	Assignee: assignee; Status: '-'; Due Date: '9/11'; Tags: []

Test Case:

Input	@assignee test task #monday #tech #marketing
Output	Assignee: assignee; Status: '-'; Due Date: 'monday'; Tags: [tech, marketing]
Expected Output	Assignee: assignee; Status: '-'; Due Date: 'monday'; Tags: [tech, marketing]

Test Case:

Input	#tech @assignee test task #doing #9/11/2017
Output	Assignee: assignee; Status: ‘doing’; Due Date: 9/11/2017; Tags: [tech]
Expected Output	Assignee: assignee; Status: ‘doing’; Due Date: 9/11/2017; Tags: [tech]

Chapter-5: CONCLUSION

7.1 Conclusion

MeetNotes provide an alternative to paper based records. They are also a source of information that can be used as part of other processes to address a wide range of issues.

The MeetNotes system combines a robust infrastructure for broad-based health information to exchange. It can improve the quality, and efficiency of meetings.

7.2 Future Scope

This product has a great future scope as there is no organization in the world that does not conduct meetings. Following are the features that can be implemented in future:-

- We can provide rich text functionality for making notes as they can be made easy to read
- Note taking by speech
- Voting on decisions to be taken
- Extension for browser