

RUBIX

Chat With Ease

Lakshmi Vyshnavi Perla

Bhargavi Sandhya Podile

Kamal Tej Veerapaneni

Vinay Maturi

MESSAGE SCHEDULING

1. Project Goals and Objectives	3
2. Project Plan	4
3. Second Increment Report	
a. Wire Frames	6
b. Architecture Diagram	9
c. Class Diagram	10
d. Sequence Diagram	11
e. Use Case Diagram	12
f. Implementation	17
g. Existing Service	19
h. User Stories	
4. Deployment(Wiki)	20
5. Testing	27
6. Project Management	29

PROJECT GOALS and OBJECTIVES

In general, texting of messages takes place instantly, but to make the messages to be delivered at a certain time or on some condition is called as Message Scheduling. It mainly helps in sending the messages to the recipients at certain time, because sometimes user may forget about that or the user might be busy with his work. This method can also be helpful in sending messages to the recipients based on their input message automatically. Sending the message based on the recipient message automatically will help a lot to the organizations such as colleges and recruitment departments.

Overall Goal:

Providing an environment to send a specific message at a certain time and to generate an automated reply for users to queries based on their request subject through mail.

Specific Features

- To set time and date for the message to be sent in a chatting environment.
- To setup a mailing system that generates an automated reply to the user based on the user's message based on the user's subject mentioned in the mail.

Significance:

- To send the desired message at specific time to the recipient. This will elevate the use of message scheduling.
- To increase the efficiency of response to the users through mailing process.

Specific Objectives:

- The first and foremost is to increase the immediate responsive nature for basic questions from an organization.
- To remove the problem of remembering to message every time for each event.

Specific Features:

- A regular chat is included along with the feature of specifying a particular time to send message or for a specific message. In addition to that, a system is created by sending the reply to the e-mails or the messages based on finding the key words of the subject that the user is sending. This can be done by using certain API. So finally using several APIs by adding certain features to the existing system would help us achieve the task completed

Related Works:

- SMS Sent TIME:

This is an android application which sends messages to the users by setting date and time.

- Text Magic

This is also a kind of website where a message can be sent to the user from its website by giving all the information and by setting the time. This can even send message to other countries also by giving the time of the time zone of that country.

Implementation:

This is a part of project, where our idea is implemented in an executive process. This is the phase where, we will give our inputs and follow the procedure to expect the desired outputs.

Main Modules:

The web base application requires the following languages.

HTML (Hyper Text Markup Language) : This is used to create the display of the pages on the websites that the user interacts with the application.

CSS (Cascading Style Sheets): It is used to give the styling for the documentation written using HTML for the web pages.

JavaScript: This is a type of programming language used to make the web pages interactive with the user by giving the logic for the items like buttons, links e.t.c,

Angular JS: It is a JavaScript based open-source framework which reduces the java script usage effort.

Boot Strap. It is a framework used to reduce the work from the scrap and gives an effective view of the web pages and add the functionalities like buttons with easy and good view.

Existing Services/REST API:

Oath Login:

For this increment the g-mail oAuth login API has been used login into the home page of our website.

WebStorm:

To execute the source code, webstorm has been used.

Chat Environment

Once the user logs into the account, user can have a chat room where multiple users can chat.

Mailing System

The user can have mailing system at which the user can send mail by setting a time to the mail. This environment is created using JavaScript.

Firebase

The messages that are being sent in the chatroom will be saved in the firebase database.

Detail Design of Features

Wireframes



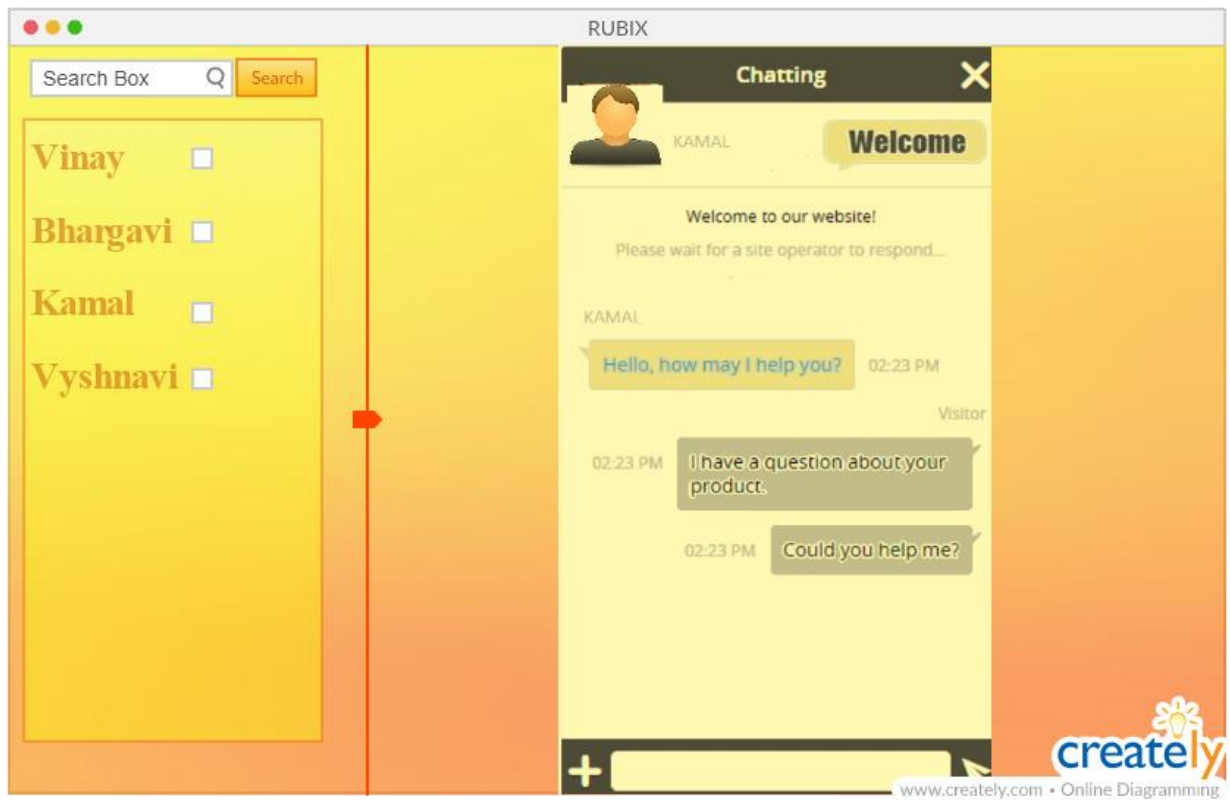
Register Page

The screenshot shows a web browser window titled "RUBIX". The page has a yellow-to-orange gradient background. At the top, a yellow banner contains the text "CREATE A NEW ACCOUNT". Below this, a yellow button labeled "HOME" is on the left. The main content area contains a registration form with the following labels and input fields: "FIRST NAME :", "LAST NAME :", "EMAIL :", "USERNAME :", "PASSWORD :", and "CONFIRM PASSWORD :". Each label is followed by a white rectangular input field. Below the form is a yellow button labeled "REGISTER". At the bottom left, a yellow banner contains the text "CONTACT US". At the bottom right, the "createely" logo is displayed, with the text "www.createely.com • Online Diagramming" below it.

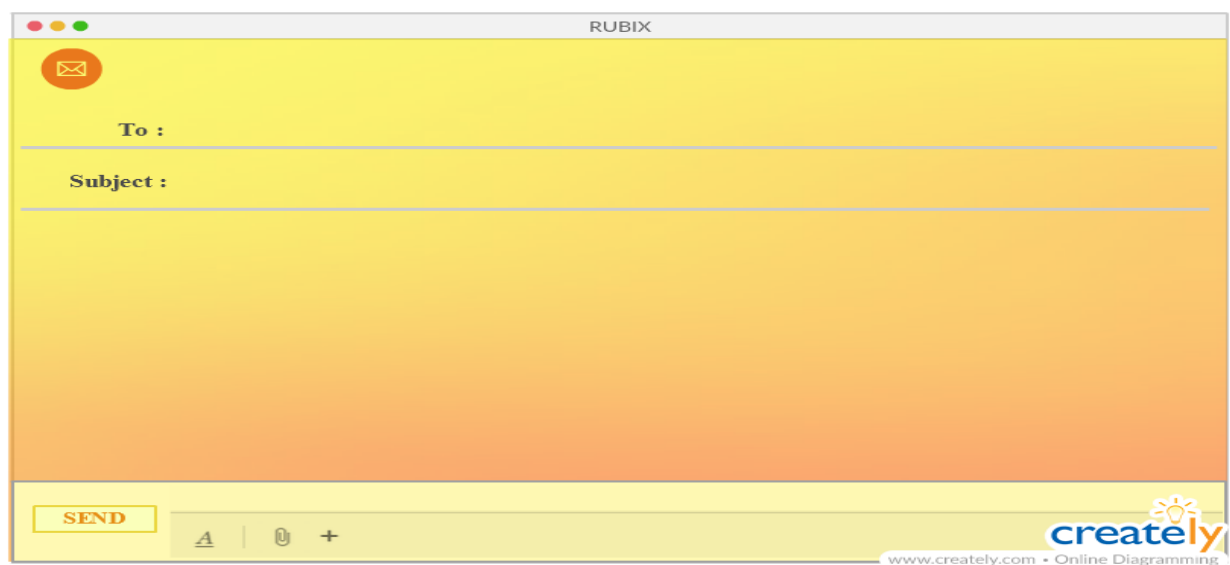
Home Page



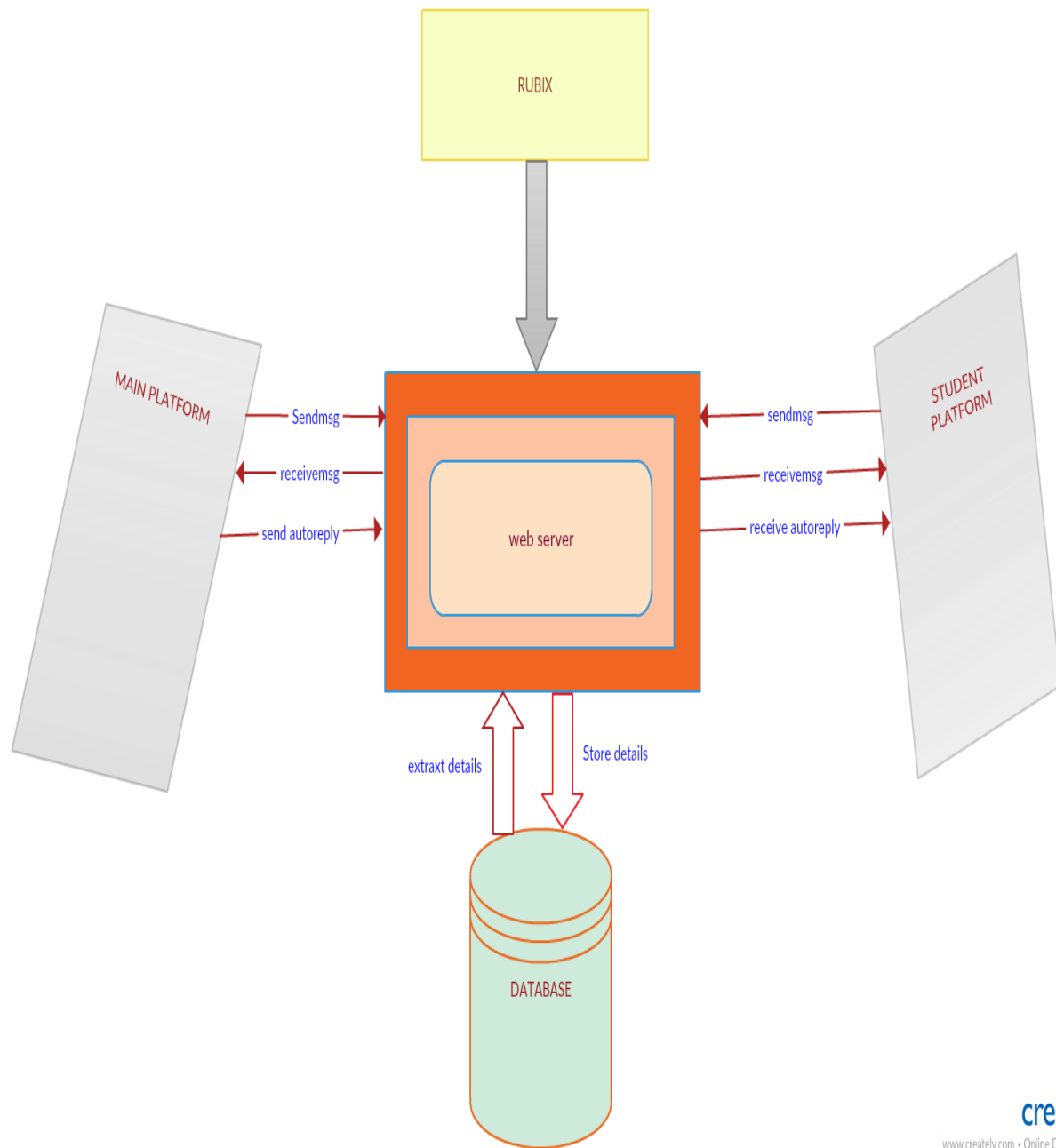
Chat Page



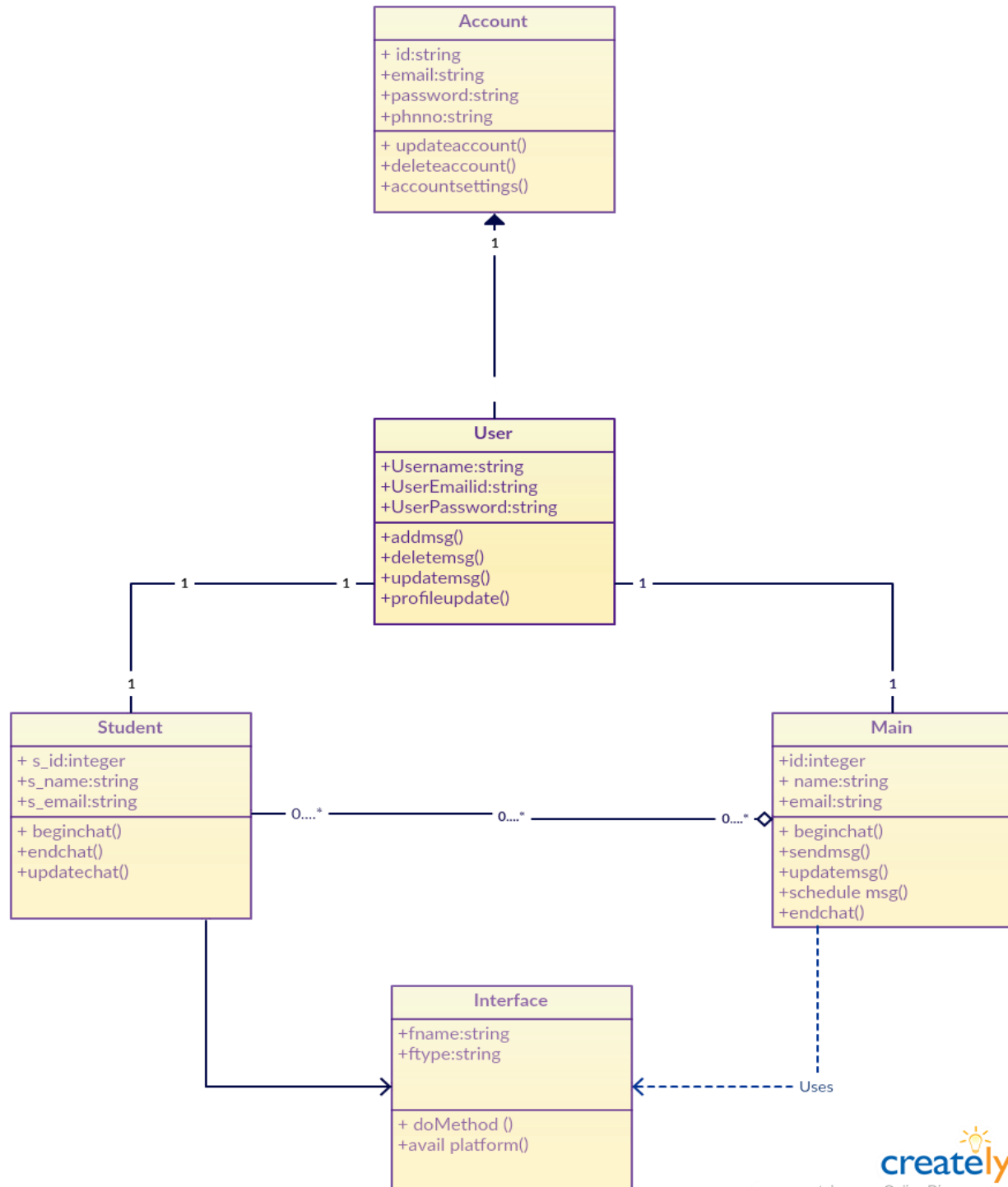
Mailing Page



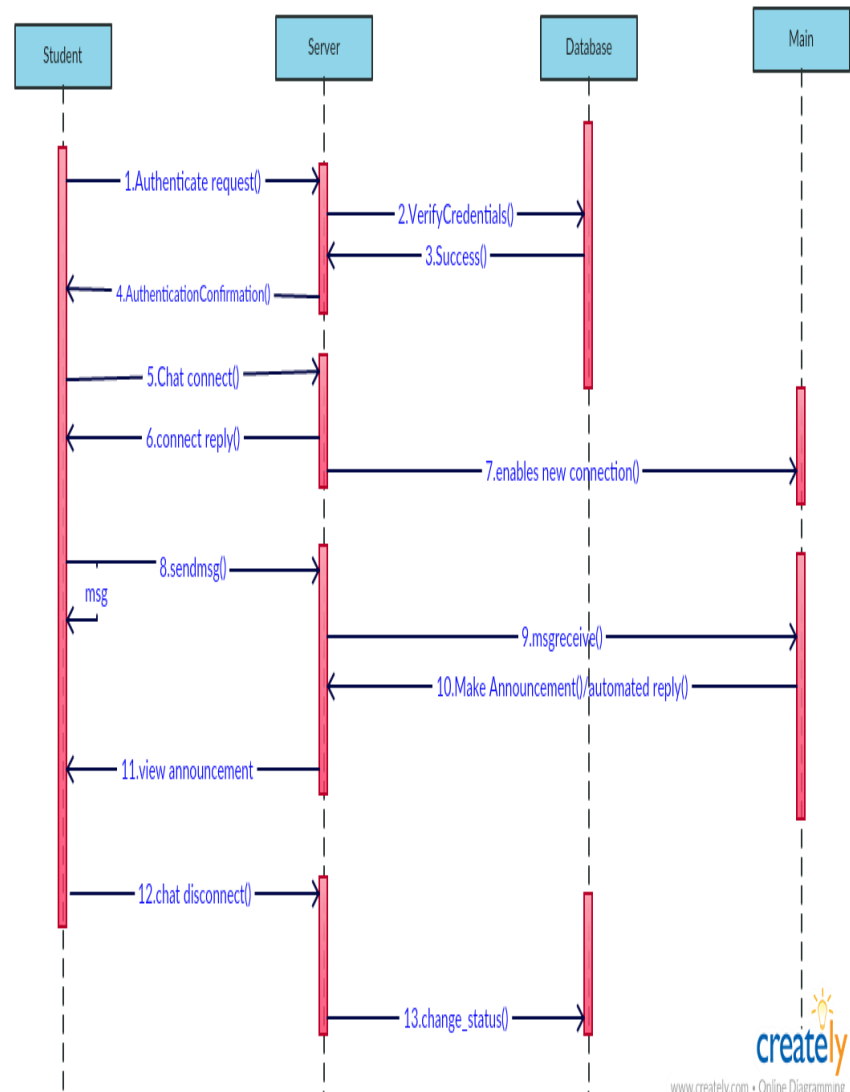
Architectural Diagram



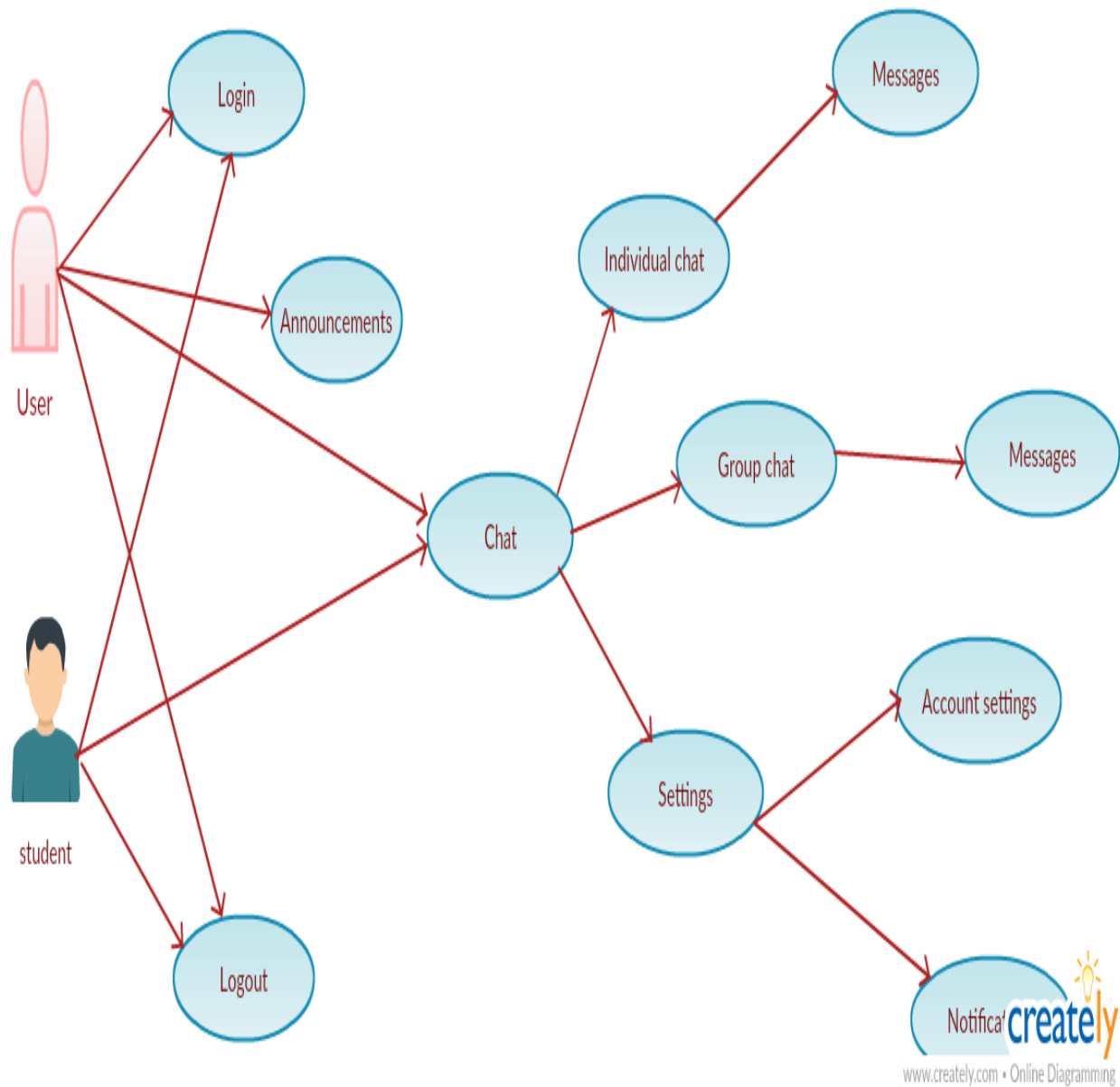
Class Diagram



Sequence Diagram



Use Case Diagram



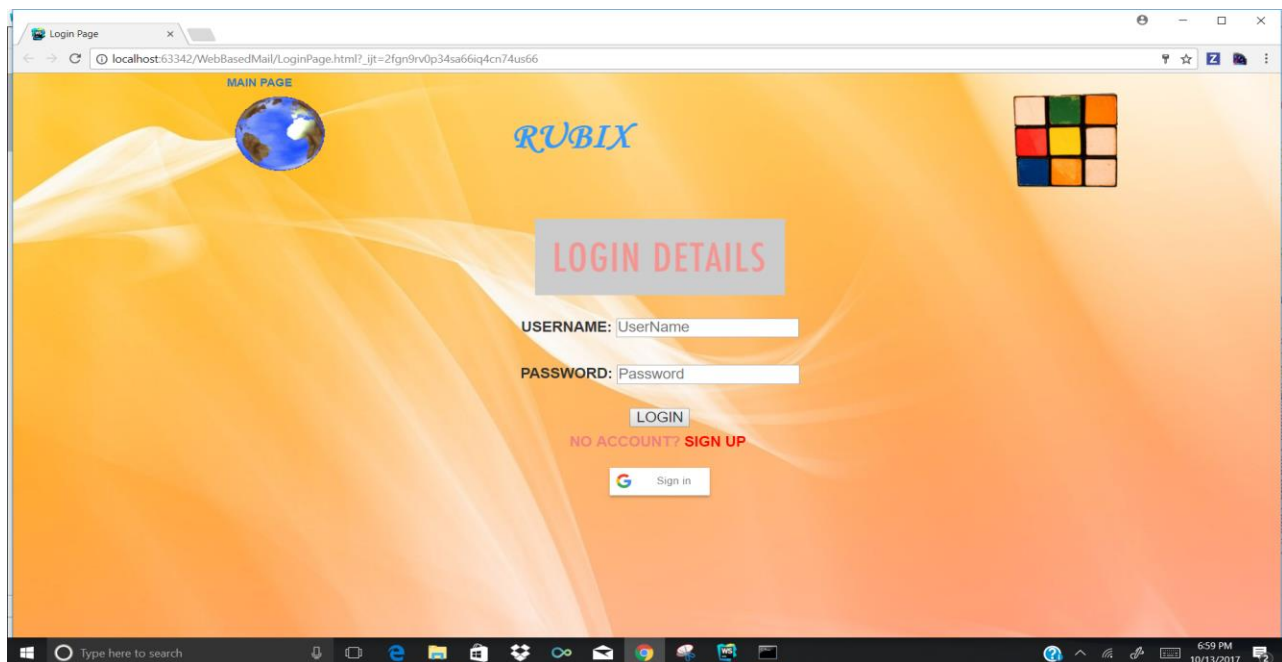
User Stories

1. As a user, I want to Login, so that I can ask queries
2. As Main, I want to Login and check the notifications about the doubts from students.
3. As a user, I want to chat with main to ask doubts, so that I can get my problem rectified.
4. As a main, I want to view incoming chat messages, so that I can respond to student's queries.
5. As a main, I want to post important announcements with a scheduled time, so that I can let students know about it.
6. As a Student, I want to view announcements, so that I can follow main's announcements.
7. As a main, I can send automated replies to students.
8. As a main, I want to search for a specific Student, so that I can ask or clarify the doubts

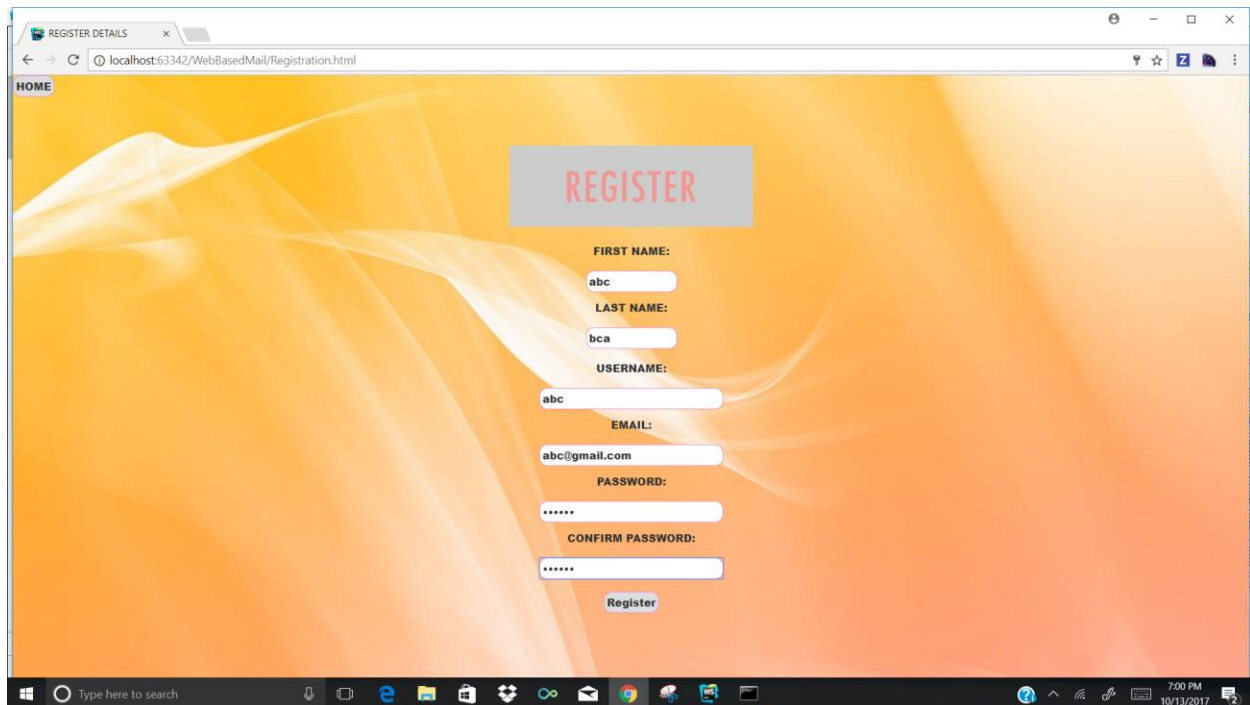
DEPLOYMENT

The implementation of the website works as follows.

The **HOME** page of our website is

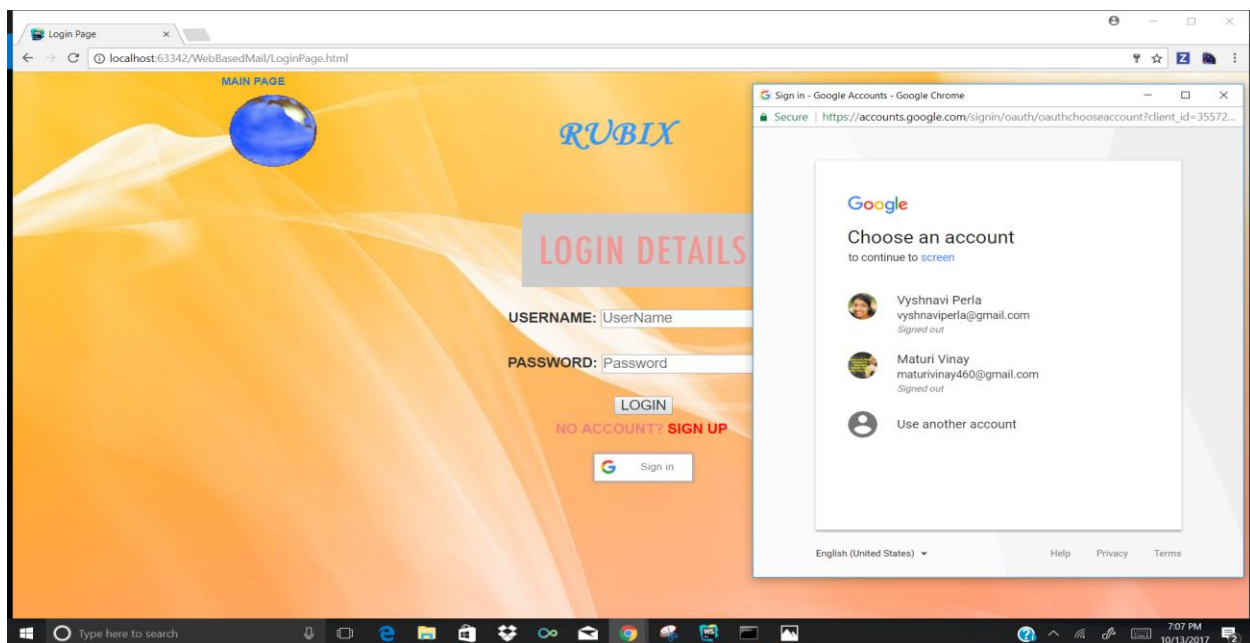


If the user is a new user, the **Registration page** can be obtained. The registration page looks as below.



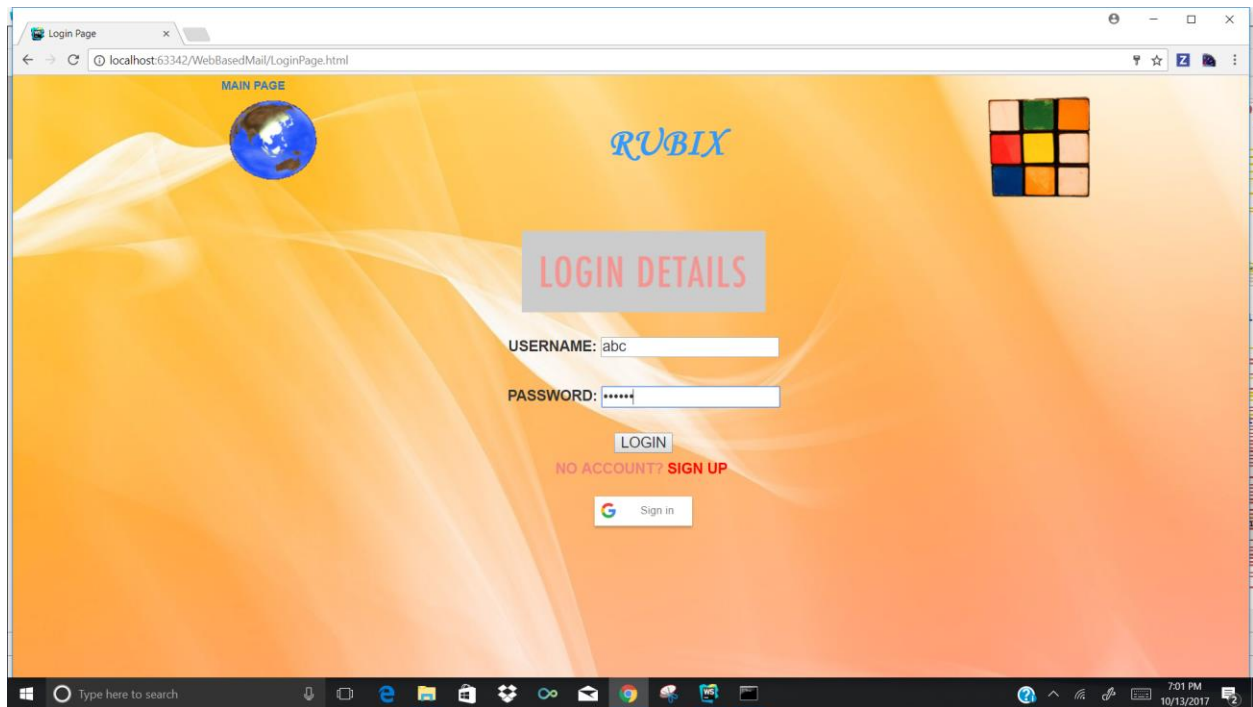
The screenshot shows a web browser window with the address bar displaying 'localhost:63342/WebBasedMail/Registration.html'. The page has a yellow and orange wavy background. A 'HOME' link is in the top left. A grey box with the word 'REGISTER' in red is centered. Below it, the form fields are: 'FIRST NAME:' with 'abc', 'LAST NAME:' with 'bca', 'USERNAME:' with 'abc', 'EMAIL:' with 'abc@gmail.com', 'PASSWORD:' with '*****', and 'CONFIRM PASSWORD:' with '*****'. A 'Register' button is at the bottom.

If the user logs in using G-mail Oath login the page looks as follows.

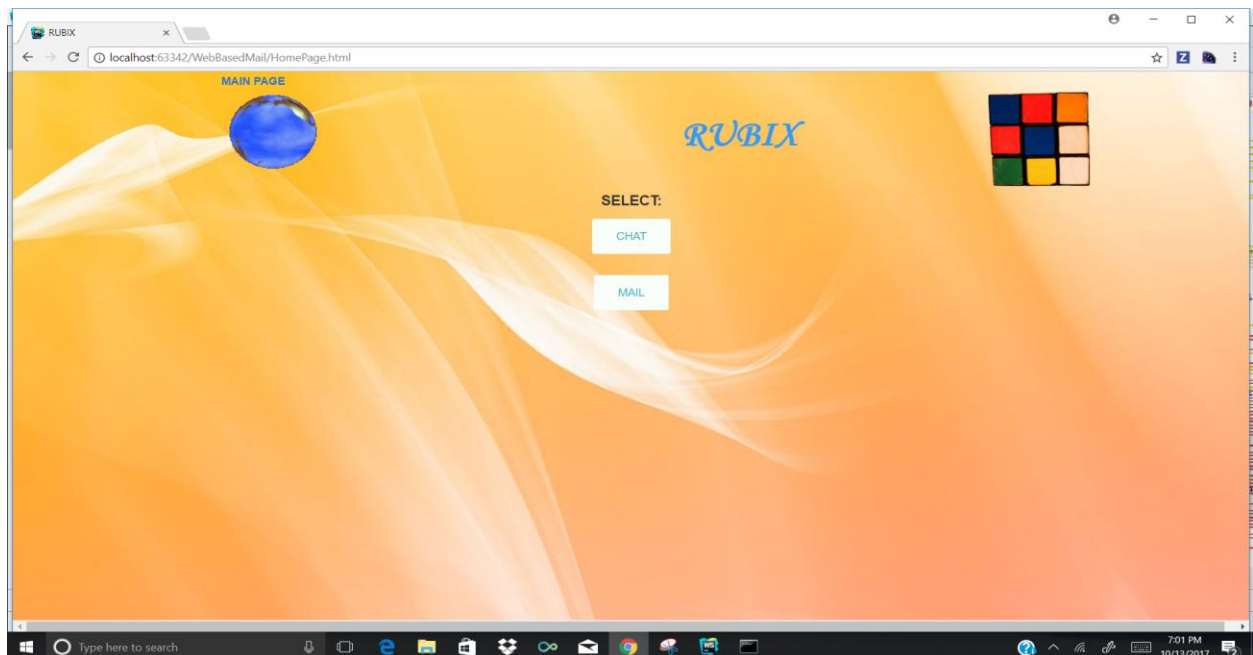


The screenshot shows a web browser window with the address bar displaying 'localhost:63342/WebBasedMail/LoginPage.html'. The page has a yellow and orange wavy background. A 'MAIN PAGE' link is in the top left. A blue globe icon is in the top center. The 'RUBIX' logo is in the top right. A grey box with the words 'LOGIN DETAILS' in red is centered. Below it, the form fields are: 'USERNAME:' with 'UserName', 'PASSWORD:' with 'Password', and a 'LOGIN' button. Below the 'LOGIN' button, there is a link 'NO ACCOUNT? SIGN UP' and a 'Sign in' button with a Google logo. An overlay window from Google is open on the right, titled 'Sign in - Google Accounts - Google Chrome'. It shows a 'Choose an account' dialog with two accounts: 'Vyshnavi Perla' (vysnaviperla@gmail.com) and 'Maturi Vinay' (maturivinay460@gmail.com), both signed out. There is also a 'Use another account' option. The overlay window has a 'Sign in' button and a 'Sign out' button.

If the user log in using the credentials it looks as follows.



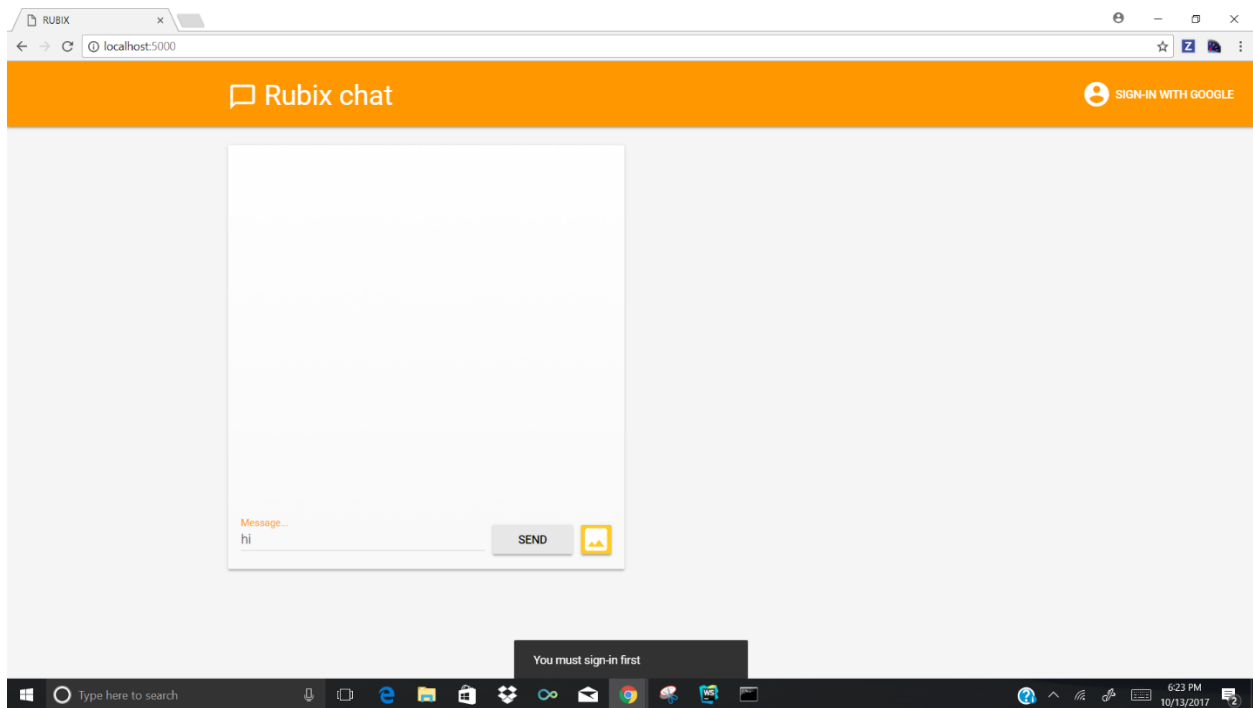
Once the user log in either using his credentials or using the g-mail oath login, the home page will be opened. It looks as follows.



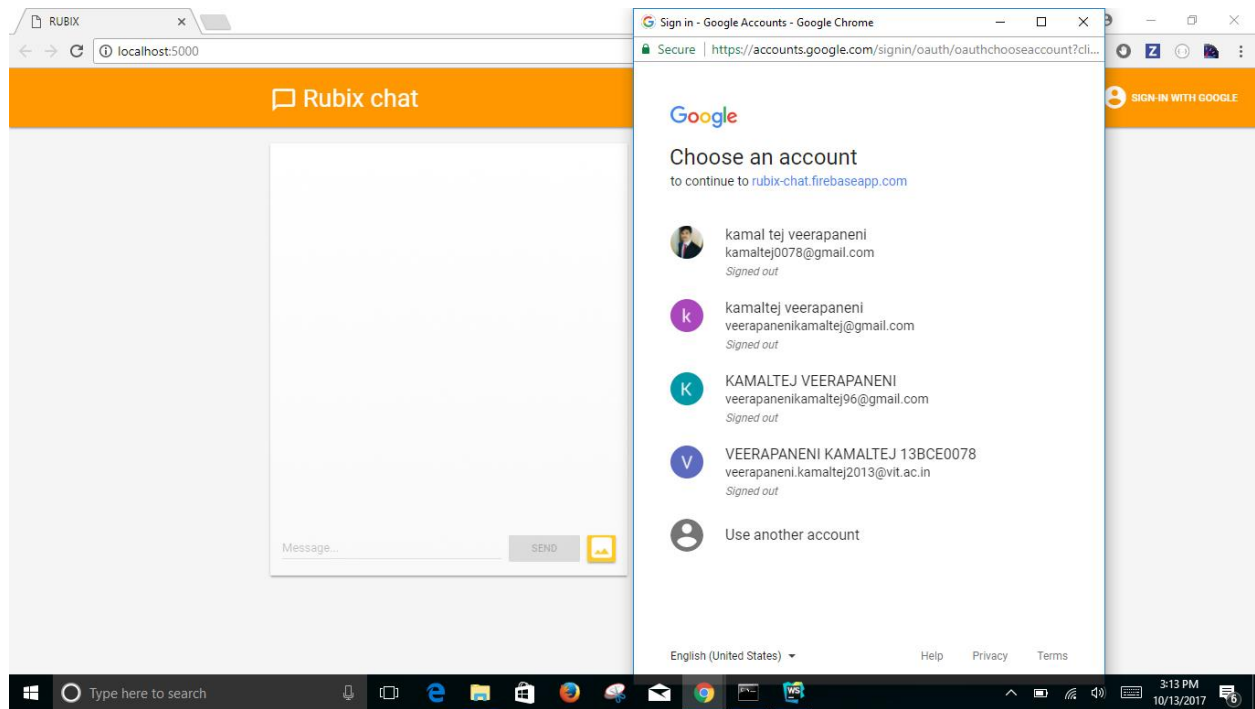
Chat Room

Once the user clicks the CHAT button, a chat room opens where the user can chat with multiple users.

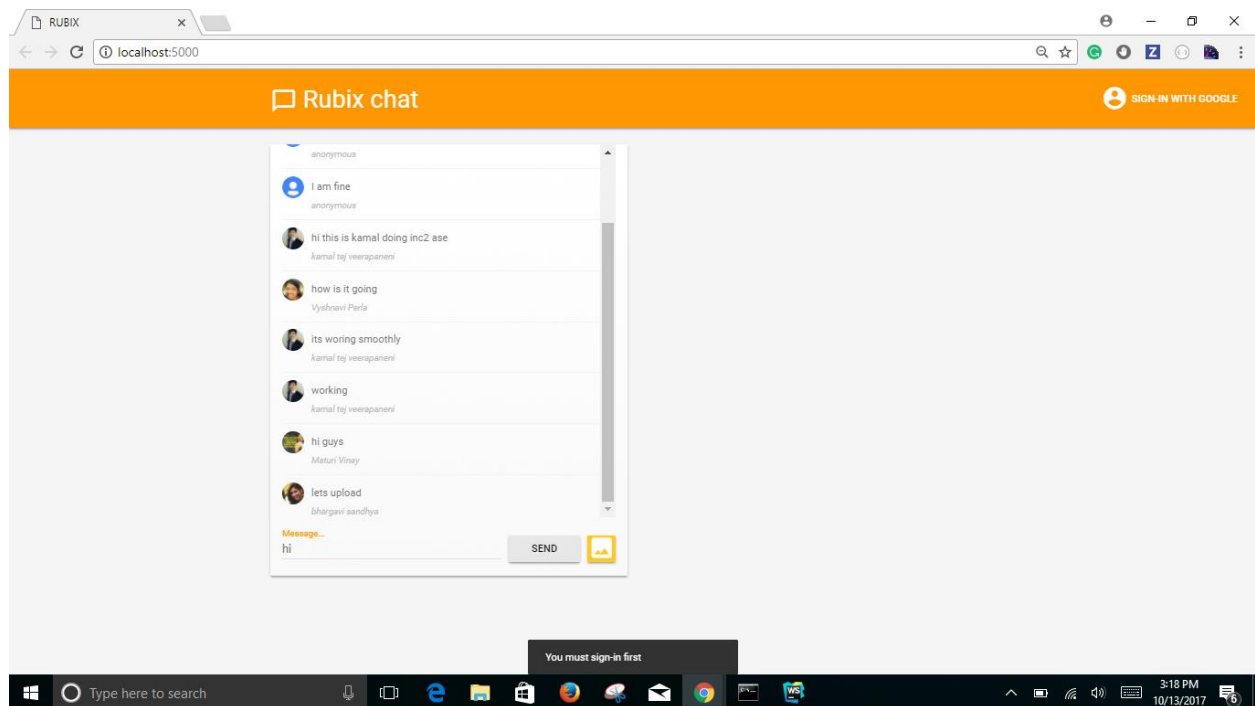
The chat room looks as below.



The users whoever want to join the chat room has to sign in using their G-mail login, It looks as follows.



The messages that are sent by the users will look as follows.



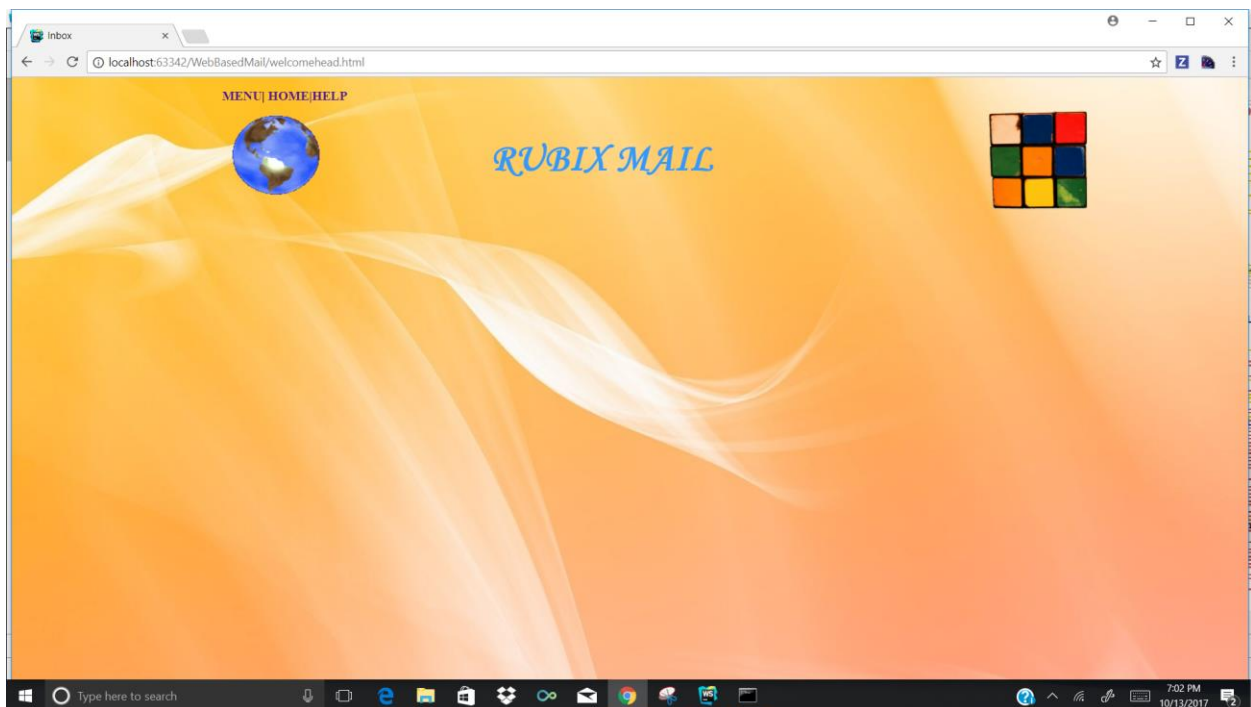
The user can send images in the chatroom, using the option and looks as below

The messages in the chat room looks as follows.

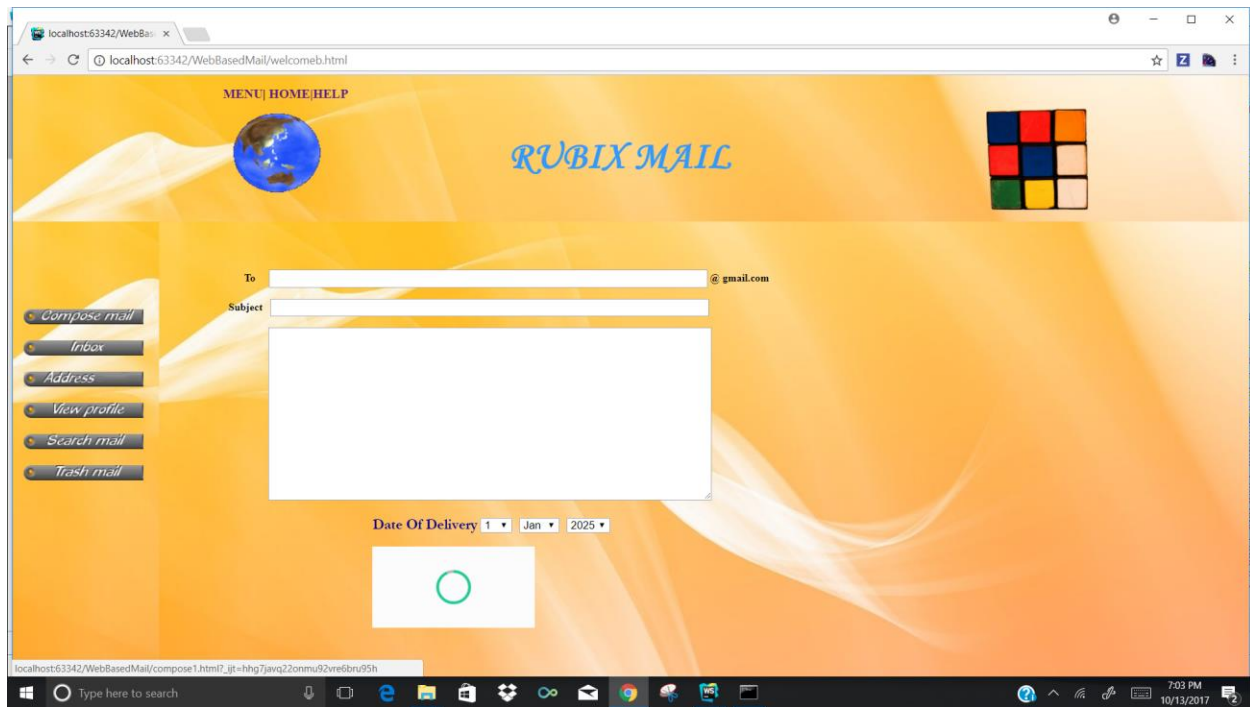
Mailing System

If the user selects the mailing system, the following page opens where user can send a mail to the desired persons mail.

The mailing home page looks as follows.

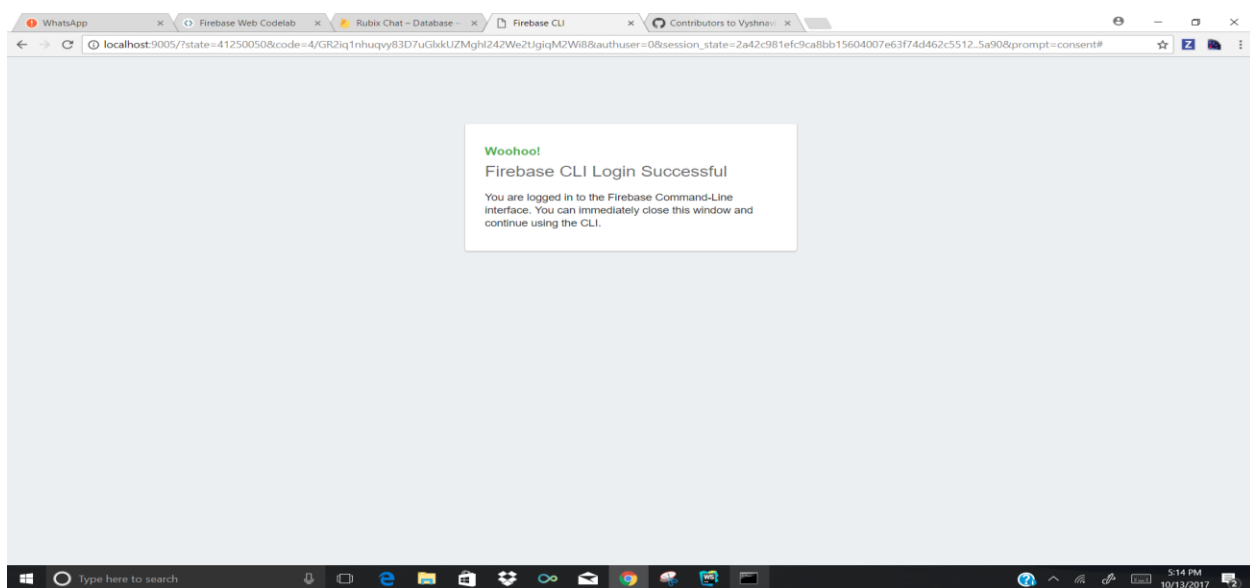


The mailing system of the page looks as follows with scheduling option.

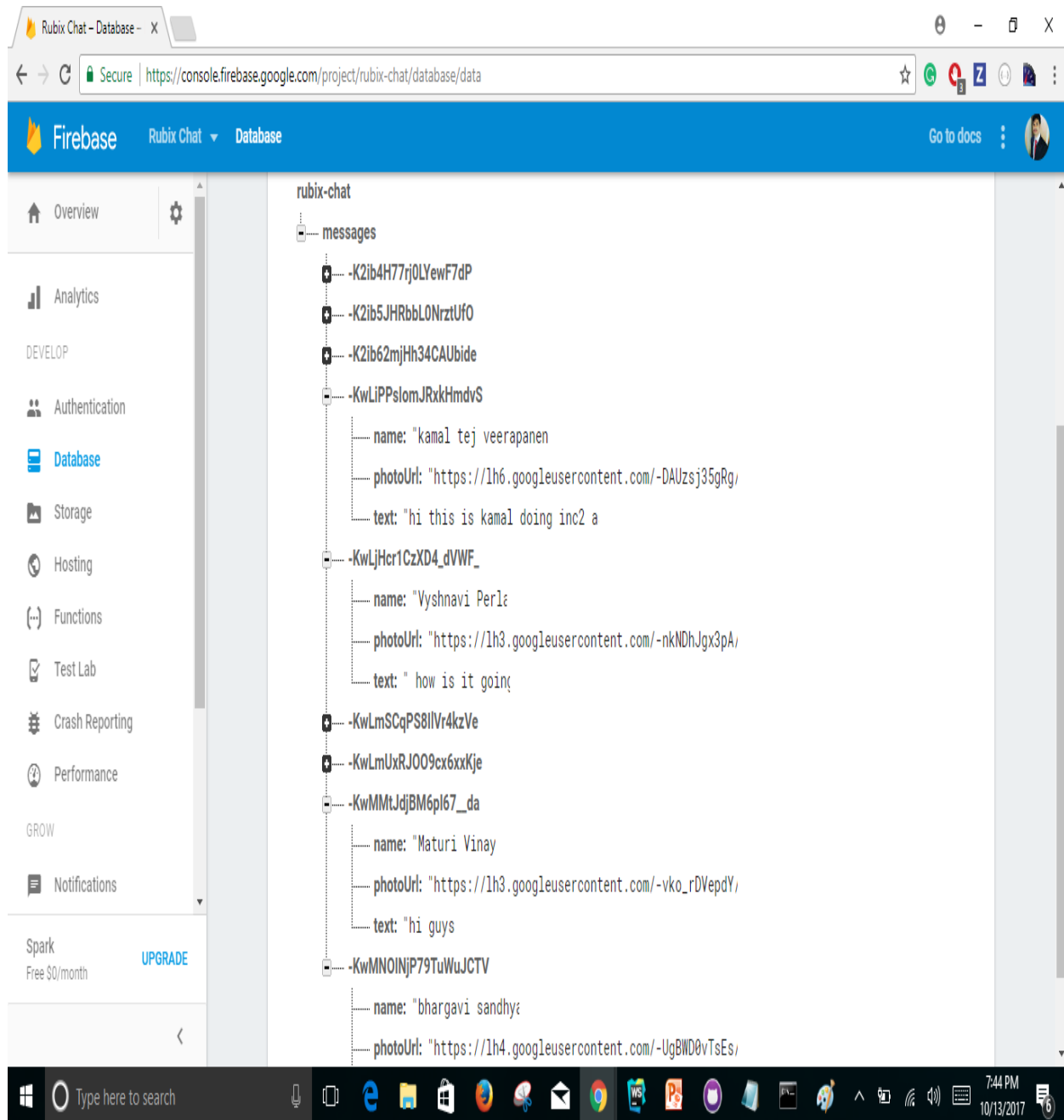


Firebase page

The firebase login for the application looks as follows.



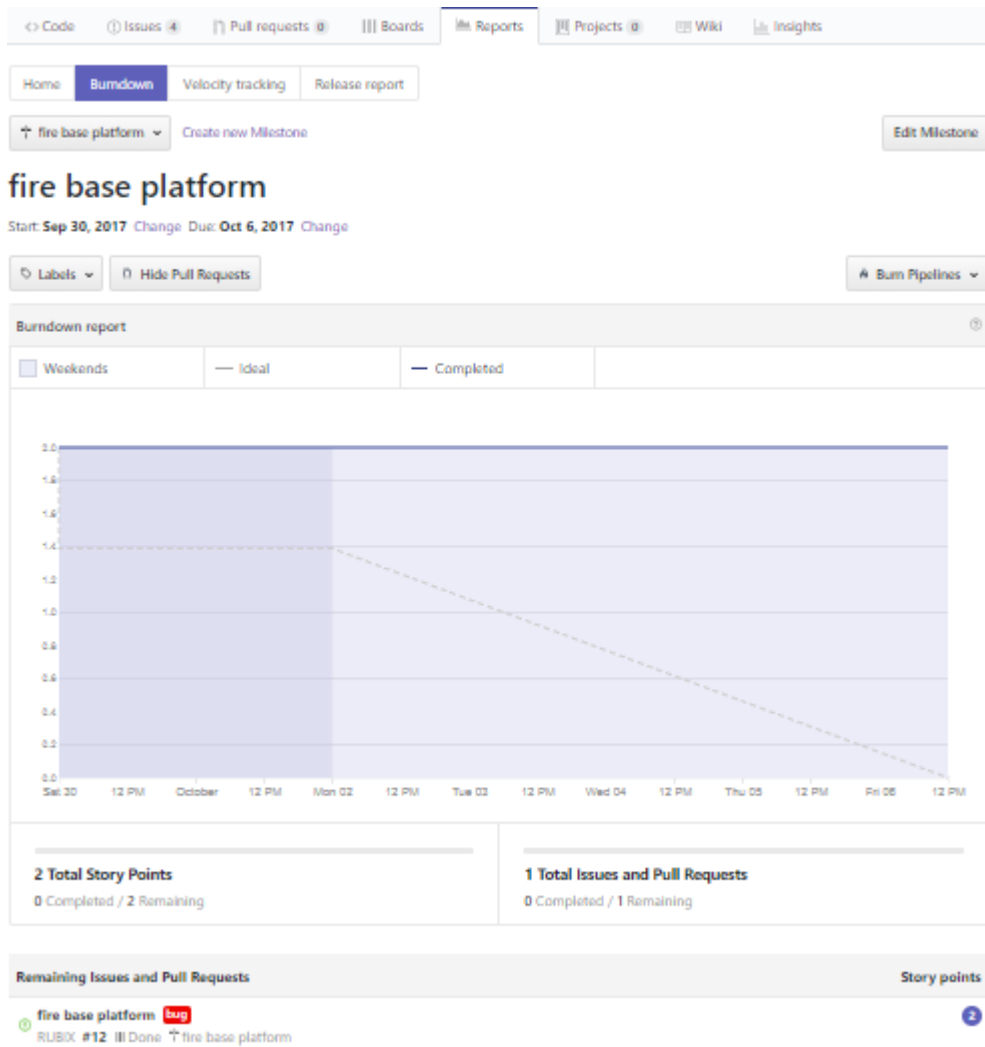
The data will be stored into the firebase and looks as follows.



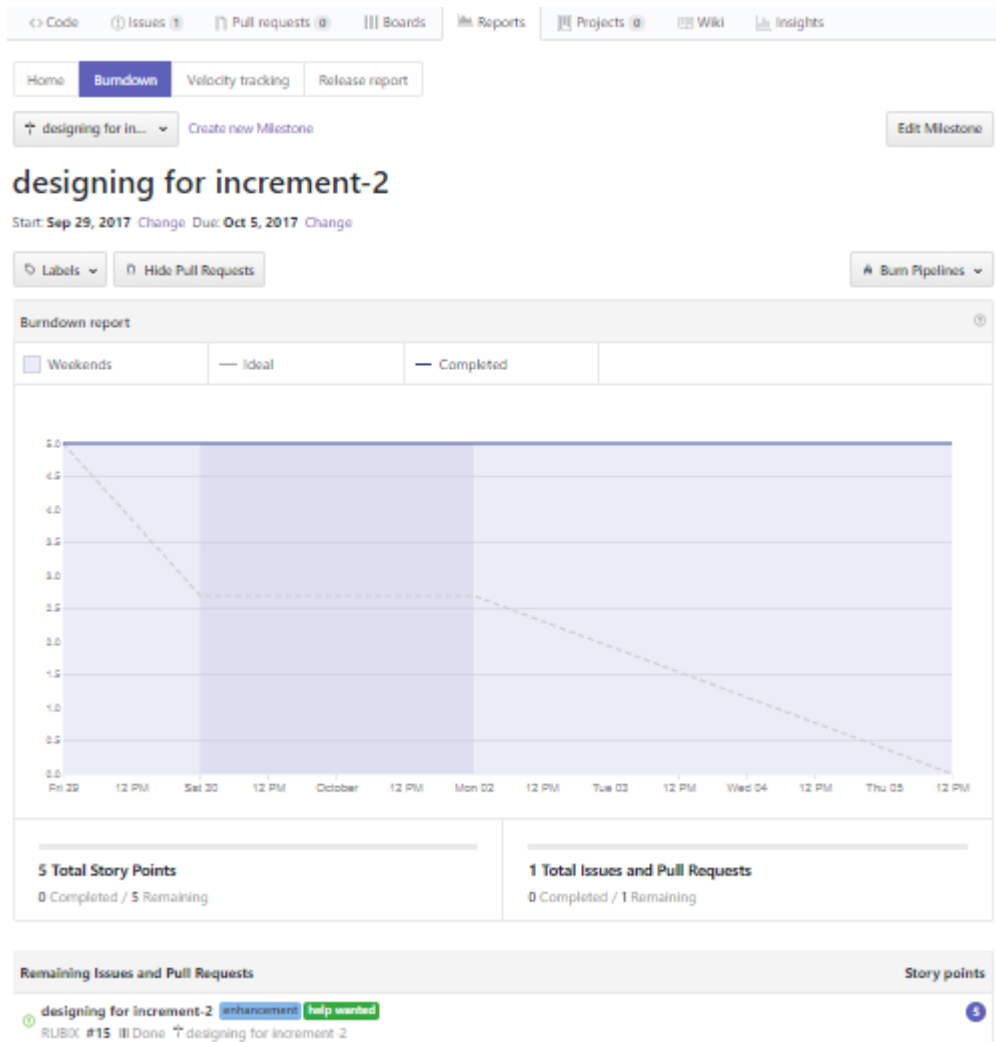
PROJECT PLAN:

Burndown Charts of the issues are as follows.

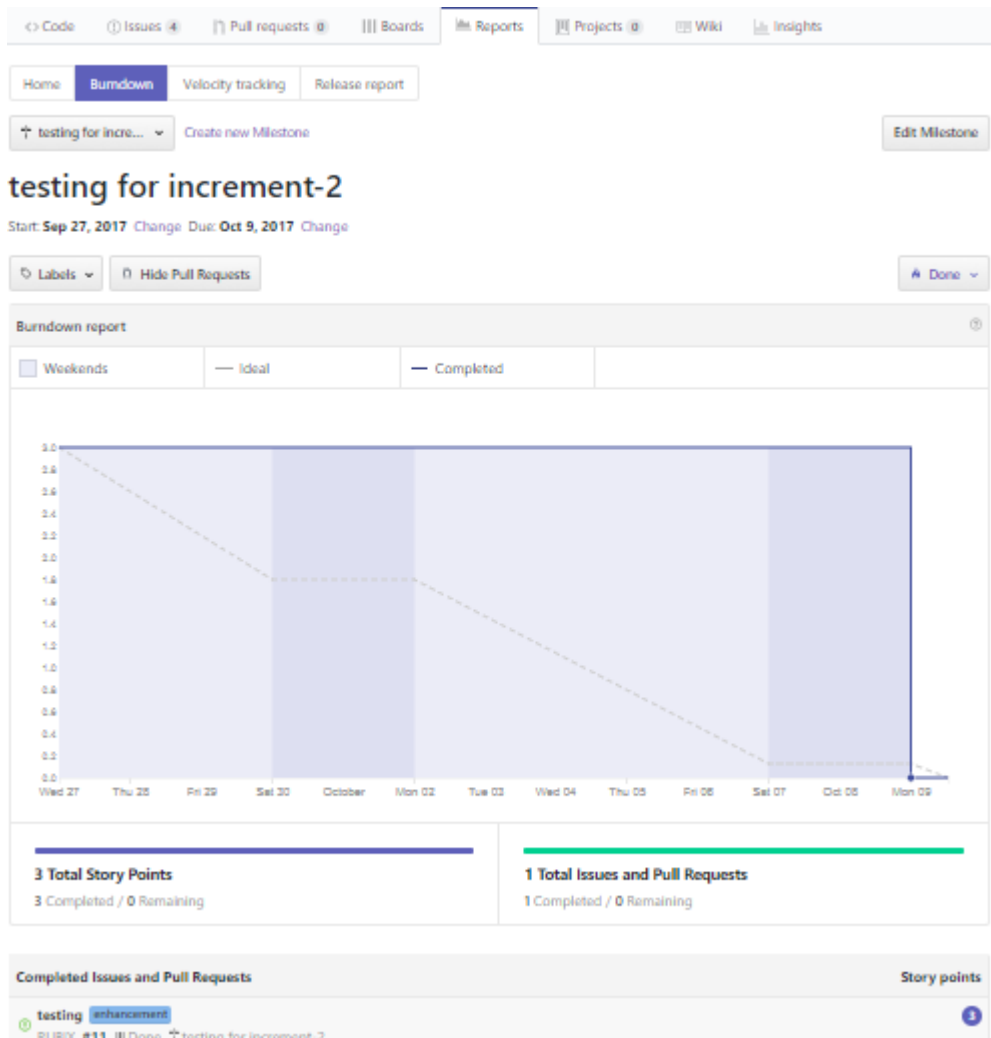
Burndown Chart for firebase platform



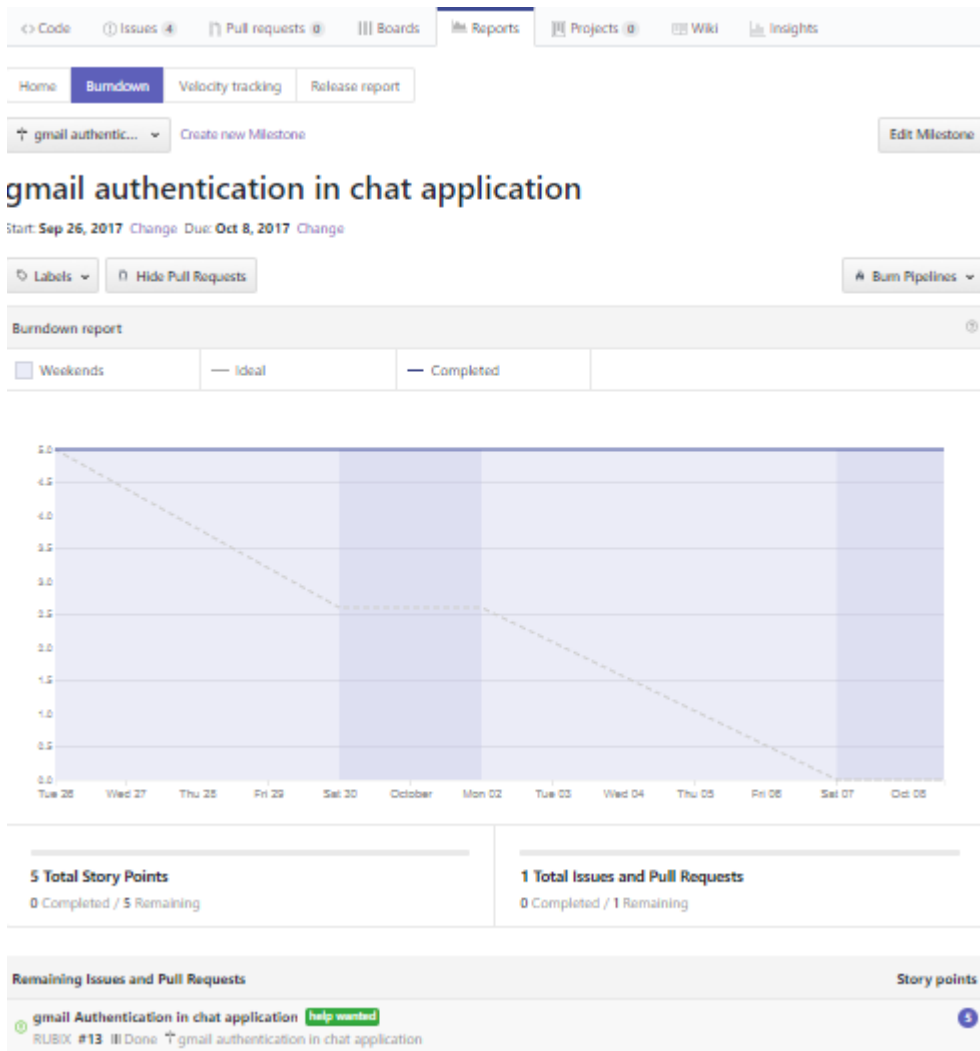
Burndown chart for designing of increment



Burndown chart for testing



Burndown chart for chat application



YSlow Code quality of the pages are as follows.

The screenshot shows the YSlow analysis interface for a page titled "Login Page" at the URL `localhost:63342/WebBasedMail/LoginPage.html`. The page has an overall performance score of 86 and a Grade of B. The analysis is based on the YSlow(V2) ruleset. The page contains 23 elements, categorized by type: Content (6), Cookie (2), CSS (8), Images (2), JavaScript (4), and Server (6). The primary recommendation is "A Make fewer HTTP requests", which is a Grade A suggestion. The analysis notes that the page has 3 external stylesheets and suggests combining them into one to reduce the number of HTTP requests. Other recommendations include using a Content Delivery Network (CDN), avoiding empty `src` or `href` attributes, adding Expires headers, compressing components with gzip, putting CSS at the top, putting JavaScript at the bottom, and avoiding CSS expressions. The page also includes a "Read More" link for further details.

Grade **B** Overall performance score 86 Ruleset applied: YSlow(V2) URL: `http://localhost:63342/WebBasedMail/LoginPage.html`

ALL(23) FILTER BY: CONTENT(6) | COOKIE(2) | CSS(8) | IMAGES(2) | JAVASCRIPT(4) | SERVER(6)

A Make fewer HTTP requests Grade A on Make fewer HTTP requests

This page has 3 external stylesheets. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[Read More](#)

Copyright © 2017 Yahoo! Inc. All rights reserved.

The screenshot shows the YSlow analysis interface for the same "Login Page". The page has a total of 9 HTTP requests and a total weight of 907.4K bytes with an empty cache. The analysis is based on the YSlow(V2) ruleset. The page contains 23 elements, categorized by type: Content (6), Cookie (2), CSS (8), Images (2), JavaScript (4), and Server (6). The primary recommendation is "A Make fewer HTTP requests", which is a Grade A suggestion. The analysis notes that the page has 3 external stylesheets and suggests combining them into one to reduce the number of HTTP requests. Other recommendations include using a Content Delivery Network (CDN), avoiding empty `src` or `href` attributes, adding Expires headers, compressing components with gzip, putting CSS at the top, putting JavaScript at the bottom, and avoiding CSS expressions. The page also includes a "Read More" link for further details.

Statistics The page has a total of 9 HTTP requests and a total weight of 907.4K bytes with empty cache

WEIGHT GRAPHS

Empty Cache:

Component	Weight
3 JavaScript File	90.5K
2 Stylesheet File	144.6K
1 CSS Image	404.1K
2 Image	266.4K
1 Favicon	1.5K

Printed Cache:

Component	Weight
2 JavaScript File	45.8K
1 CSS Image	0.0K
2 Image	8.0K
1 Favicon	1.5K

RUBIX
localhost:5000

Rubix chat

SIGN-IN WITH GOOGLE

chrome-extension://ninejcohdiippngpapiinmkgilmakhsyslow.html#1

HomeGradeComponentsStatistics

RulesetsYSlow(V2)EditHelp

GradeB

Overall performance score 82 Ruleset applied: YSlow(V2) URL: http://localhost:5000/

ALL(23) FILTER BY: CONTENT(6) COOKIE(2) CSS(6) IMAGES(2) JAVASCRIPT(4) SERVER(6)

B Make fewer HTTP requests

F Use a Content Delivery Network (CDN)

A Avoid empty src or href

F Add Expires headers

F Compress components with gzip

A Put CSS at top

A Put JavaScript at bottom

A Avoid CSS expressions

n/a Make JavaScript and CSS external

Grade B on Make fewer HTTP requests

This page has 4 external Javascript scripts. Try combining them into one.

This page has 4 external stylesheets. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

Read More

Copyright © 2017 Yahoo! Inc. All rights reserved.

RUBIX
localhost:5000

Rubix chat

SIGN-IN WITH GOOGLE

chrome-extension://ninejcohdiippngpapiinmkgilmakhsyslow.html#1

HomeGradeComponentsStatistics

RulesetsYSlow(V2)EditHelp

Statistics

The page has a total of 11 HTTP requests and a total weight of 486.6K bytes with empty cache.

WEIGHT GRAPHS

Empty Cache

HTTP Requests - 11

Total Weight - 486.6K

1 HTML/Text	4.7K
4 JavaScript File	434.3K
4 Stylesheet File	43.4K
1 CSS Image	3.9K
1 Favicon	0.1K

Primed Cache

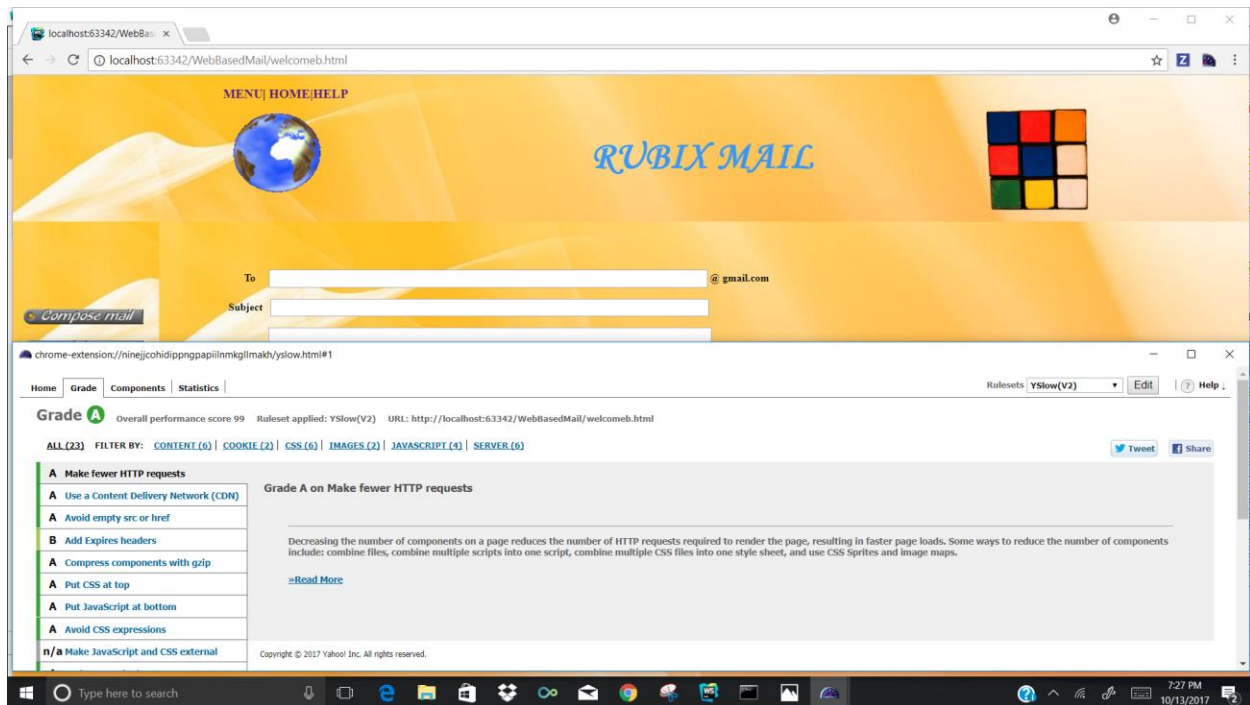
HTTP Requests - 8

Total Weight - 35.6K

1 HTML/Text	4.7K
2 JavaScript File	11.8K
3 Stylesheet File	18.8K
1 CSS Image	0.0K
1 Favicon	0.1K

javascript:document.ySlow.showStats()

26



Testing

1. Unit Testing

This testing is performed by the Developers in stages to test the design of the application. The small components of the code are tested individually to check if they are working correctly. Unit testing involves testing of the functionalities which are most crucial for that unit or component. This helps the developers to rectify the code at the same instance of the testing is performed. The testing will be limited to the specific components only and rectifying or correcting the code of that unit will not hamper the functionality of other components of the application. Once all the units are identified and tested separately. White Box Testing is used for executing a Unit Test. Debugging the code in this stage is helpful and advantageous. Once the all units/components of the application are working efficiently, error free and per expected results, these can be integrated with larger components and Integration Testing can be done.

Case	Test case Description	Expected Result	Actual Result
Login	Enter invalid username and password	Error message should pop up saying invalid credentials.	PASS
Login	Enter valid username and password	Page should redirect to Home page	PASS
Signup	Mail id and wrong password	Gives error message	PASS
Signup	Password and confirm password should	If this succeeds it redirects to login page	PASS
Chat Button	Entering the Chat environment	Chat Room for the users	PASS
Chat Message	Enter Message only after signing in through Gmail	Shows small popup at bottom if not signed in	PASS
Send Message	The message will be sent	Send message in the chat room	PASS
Email Environment	Mailing system	Mailing environment will be opened.	PASS

Project Management

Implementation Status Report:

Work completed and Contributors:

1. **Firestore Platform** : Kamal Tej Veerapaneni , Lakshmi Vyshnavi Perla, Bhargavi Sandhya Podile , Vinay Maturi
2. **Testing** -Bhargavi Sandhya Podile,Kamal Tej Veerapaneni,Maturi Vinay,Lakshmi Vyshnavi Perla
3. **Designing**-Lakshmi Vyshnavi Perla,Bhargavi Sandhya Podile,Vinay Maturi,Kamal Tej Veerapaneni
4. **G-mail OAuth** :Vinay Maturi,Lakshmi Vyshnavi Perla,Kamal Tej Veerapaneni,Bhargavi Sandhya Podile

BIBLIOGRAPHY

- 1.www.wikipedia.org
- 2.www.firebase.com
- 3.developers.google.com