CS 5551 SECOND INCREMENT REPORT

Group: 1 Project Title: **Multi Messenger Application**

Nihar Dudam (16)
Rakesh Reddy Bandi (06)
Sri Naga Sarvani Jakkula (30)
Ravi kiran Yadavalli (67)

I. Introduction

Multi Messenger application is one of the best approaches for making people interact with their friends without any interruption. Messaging friends through various applications by switching over them is tedious. This overhead can be minimized by our Multi Messenger Application. A user can message to his friend through various messengers at same time using our application. Our Application mainly aims for the new feature "Search Conversation with a keyword" and displays the result as a whole conversation involving that keyword. Search Conversation with a keyword is the main advantage of our application. It makes the user to gather all the conversation with the keyword matched. This makes the user to easily gather the useful information all at once. By not switching between applications there can be relatively less battery drain. The idea of our project can be found in project proposal document.

The project has been divided into four phases with improving implementation features. For the first iteration of our project we want to complete all design section of the application with login, dashboard page design with synchronizing the local mobile message service into our application. We have chosen the android platform to develop our application. For this first iteration, we have designed the UML class diagram, activity diagram along with wireframes. We concentrated mainly on the design part which play a major role in implementing our project.

II. Objectives/Features

In recent times, the impact of mobile applications in socializing and communicating has been huge. With increase in demand for more sophisticated applications in managing and summarizing among tens of popular social networking mediums, it is great to have a single application which could aggregate and summarize through multiple messengers for different users.

Our application aims in providing a platform which could enable a user to send and receive messages from all friend lists across multiple instant messengers and synchronize them under a single window per unique person (friend) in your contacts overall. Our application also primarily focusses on analyzing and processing the conversation data per contact (friend) from multiple messengers. The application user would be able to "search" with a keyword in the window dedicated per friend of Multi Messenger App, to fetch the relevant data from the multiple messengers.

Below are the objectives of Multi Messenger Application:

1. To synchronize contacts from different instant messengers.
2. To enable each user with Send and Receive messages functionality from single window for the synchronised messenger.
3. Reduce the latency and overhead of switching across messengers.
4. To provide a user friendly and rich application.
5. To provide the user with the functionality of searching the conversation with a specific keyword to fetch the all relevant data across multiple messengers.

III. Project Background and Related Work

There have been multiple instant Messenger Aggregators like Meebo, Nimbuzz which have been successful in providing the users to converse through different messengers from a single login. Yet, there is no application which could let users to view messages from multiple messengers in a single window.

Achieving synchronization on conversation from multiple messengers would further lead an application to smartly search and process the information for better the knowledge on conversation history. To implement our project, we need several API's to integrate. For this second increment we want to implement the synchronize operation of local mobile SMS service within our application and fetch the contacts from GTalk and use the contacts API plugin to the fetch the contacts from Device. Instant Messenger Applications such as Yahoo and Twitter do provide their messenger API's to third Party Applications. We plan to receive the necessary OAuth credentials from respective applications and provide messaging facility through our application. Our upcoming deployments would be on IBM's Blue Mix service which could host Mobile Applications in a highly scalable environment. The database provider for App being MongoDB, which is highly suitable for the mobile content accessibility and retrieval from various mobile nodes.

Gtalk API to integrate Gtalk service API in our application we need the Gtalk API from developers.google.com. It needs authentication and authorization of the user. This can be done using OAuth 2.0. The respective credentials such as key for accessing google API and secret key and client id for OAuth 2.0 to implement in the application. We can obtain these credentials from developers.console.com.

Yahoo Messenger API We can get the Yahoo API from https://developer.yahoo.com/messenger/. We can get different services based on type of device on which we are implementing. The available and popular types are iOS, Android, Web etc. For the first increment we want to implement the native SMS service within the app.

IV. Proposed System

1. Requirement Specification:
   - Functional Requirements:
     - i. User should have a single sign in.
     - ii. User must able to view his recent chats.
     - iii. User must able to view all messenger's friends list.
     - iv. User must able to search in his friends list.
     - v. User must able to search in the recent conversation with a keyword.
     - vi. User's search should yield in a collaborative and meaningful data from all the instant messages.
     - vii. User must be able add various messenger accounts.
     - viii. User must be able to configure respective accounts.
     - ix. User must be able to sync his phone contacts.
     - x. User must be able to sync messenger contacts.
     - xi. User must be able to chat consecutively through various instant messengers.

V. Import Existing Services/API

5.1 Google Talk API

Google has provided its Instant Messenger Gtalk's API to third party Developers. Gtalk's API uses XMPP protocol to communicate the messages from Google's Server. Unlike the most of instant messaging protocols today, XMPP is defined in an open standard and uses an open systems approach of development and application, by which anyone may implement an XMPP service and interoperate with other organizations' implementations. Because XMPP is an open protocol, implementations can be developed using any software license; although many server, client, and library implementations are distributed as free and open-source software, numerous freeware and commercial software implementations also exist.

Below is the architecture of XMPP Client Server Model:

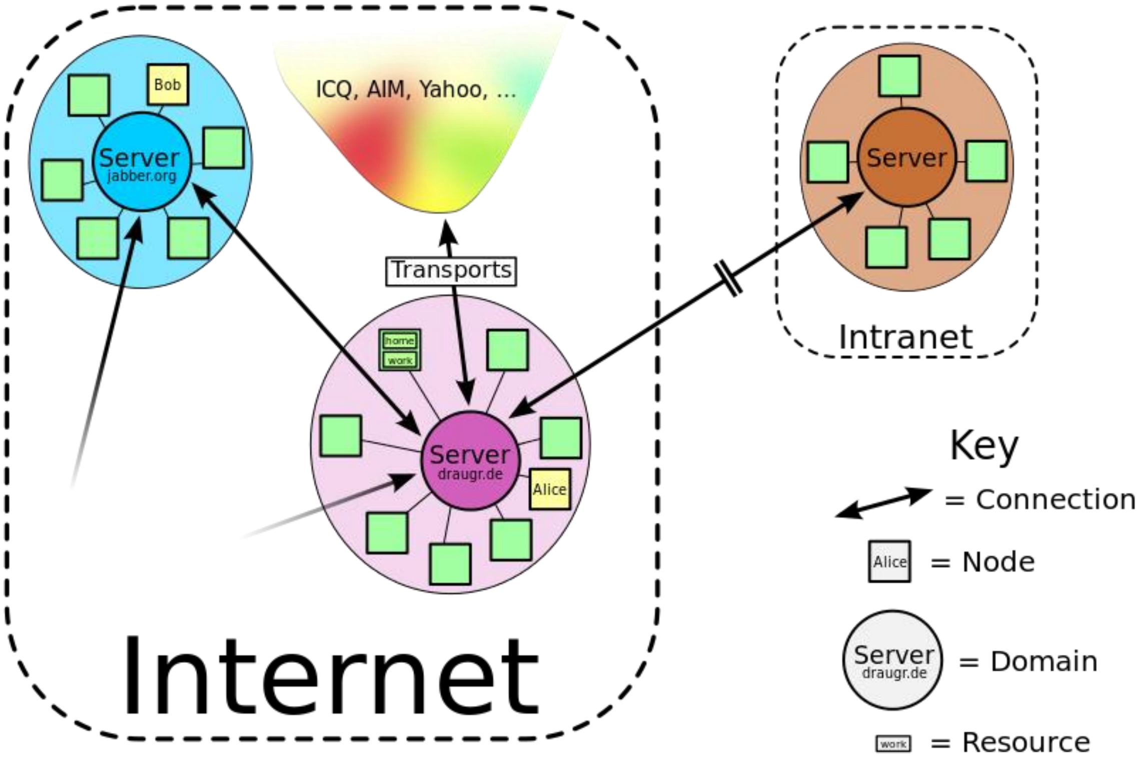


Figure 1: XMPP Architecture Diagram

The google OAuth is an Authorization API which is a two-step process in order to connect with GTalk.

- To Create a Token scoped for Chat Login.

- To Use the generated Token to authentication Google Talk Connection Servers.

5.2 Twitter API

Twitter is one of the most popular social Networking Site. In the current Increment we had been working on integrating three API's of Twitter .Below are the three API' and their functionality.

5.2.1 : GET direct messages/Show

Returns a single Directed message with the ID. Like the /1.1/direct_messages.format request, this method will include the user objects of the sender and recipient. This method requires an access token with RWD (read, write & direct message) permissions. Consult The Application Permission Model for more information.

Resource URL: https://api.twitter.com/1.1/direct_messages/show.json

5.2.2 : GET direct messages/Sent

Returns the 20 most recent directed messages sent by authenticating the user. Includes detailed information about the sender and recipient user. You can request up to 200 direct messages per call, up to a maximum of 800 outgoing DMs.

Resource URL: https://api.twitter.com/1.1/direct_messages/sent.json

5.2.3 *: POST direct messages/new*

Sends a new direct message to the specified user from the authenticating user. Requires both the user and text parameters and must be a POST. Returns the sent message in the requested format if successful.

Resource URL: https://api.twitter.com/1.1/direct_messages/new.json

Twitter requires Twitter OAuth to authenticate the Twitter User.

5.3 Yahoo API

The below are the steps in Integrating Yahoo Instant Messenger client.

- Create a Yahoo Messenger Open Authentication (OAuth) API Key.
- Authenticate with the Yahoo Messenger IM SDK servers.

- Create a new session.
- Obtain and update presence information.
- Obtain the current contact list, as well as more detailed information on a particular contact.
- Send and receive a message from another contact.

1. There are two API calls that are necessary to authenticate with the Yahoo Direct OAuth API. The first obtains a Pre-Authorized Request Token (PART), which requires user's login username, password, and the OAuth Consumer Key (API Key) that was just generated through developer.yahoo.com.

2. You would receive a Request token as an output when Log-in API is called. The below is login API: *https://login.yahoo.com/WSLogin/V1/get_auth_token?&login=username&passwd=mypassword&oauth_consumer_key=consumerkey*.

3. To receive pre-approved request token the API is: *https://login.yahoo.com/WSLogin/V1/get_auth_token*

4. To fetch Contacts from Yahoo Messenger the API is: *GET /v1/contacts?sid=msgrsessionid ; Host: rcore1.messenger.yahooapis.com*

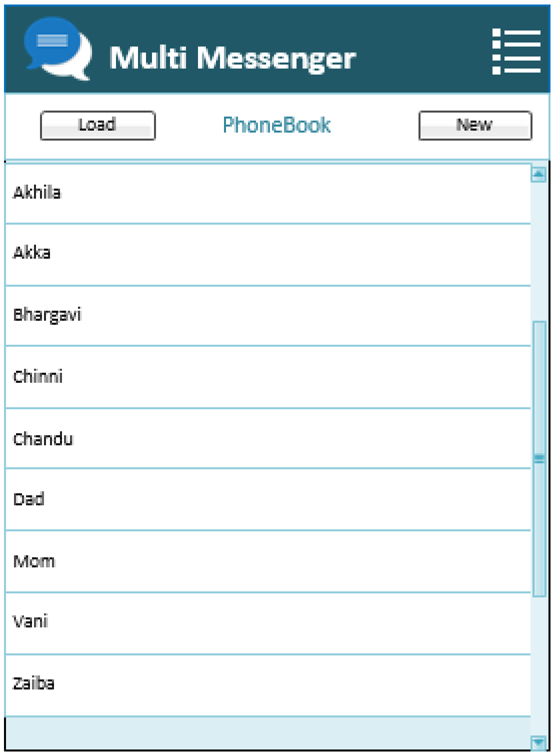
5. To receive notifications from Yahoo Messenger the API is: *GET/v1/notifications?sid=msgrsessionid&seq=5 ;Host: rproxy1.messenger.yahooapis.com*

6. To send messages through Yahoo Messenger is *POST/v1/message/yahoo/targetYahooId?sid=msgrsessionid Host: rcore1.messenger.yahooapis.com*

VI. Detail Design of Services

Wireframes:

1. Synchronizing Device Contacts Screen

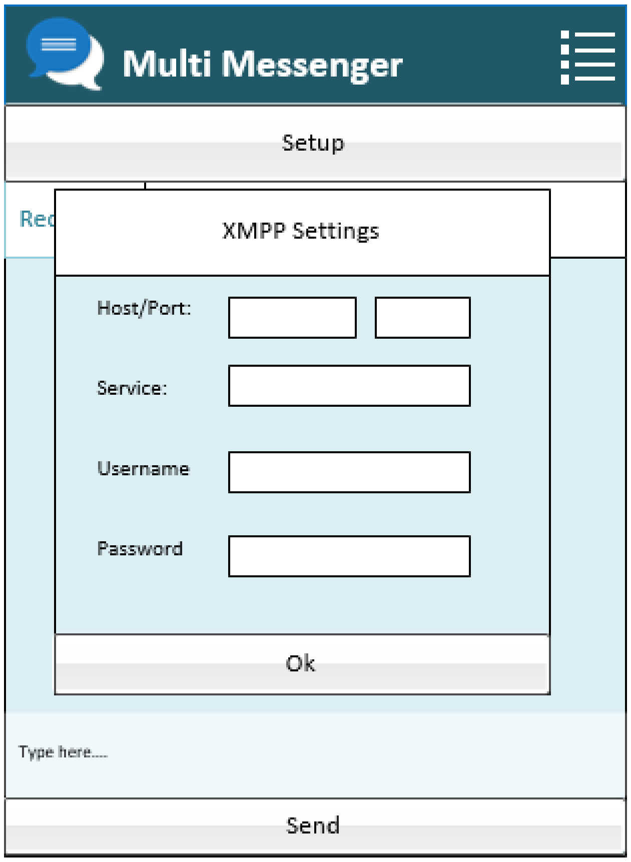


n

Description:

We get the list of all contacts that are on device when we attempt to synchronize the contacts into our application. We give an option to create a contact on the device to the user in our application.

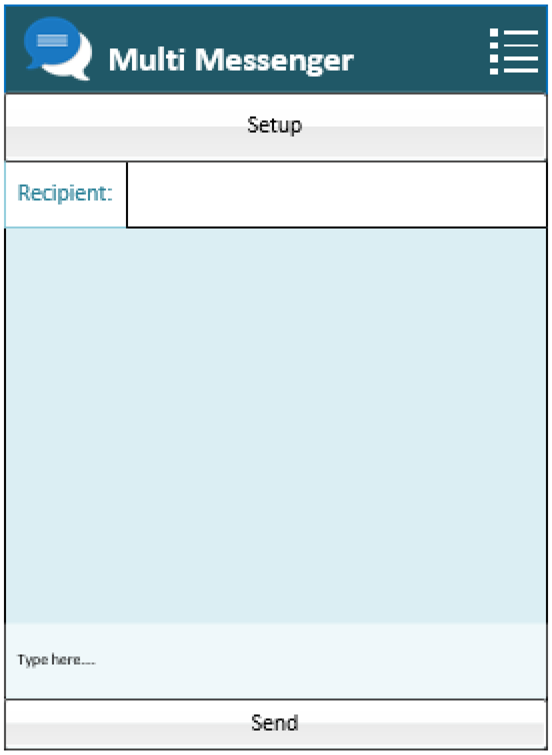
2. Gtalk XMPP settings page:



Description:

Here we give a provision of setting xmpp client in our application in order to connect to the Gtalk server on a button click.

3. Gtalk Chat Window:

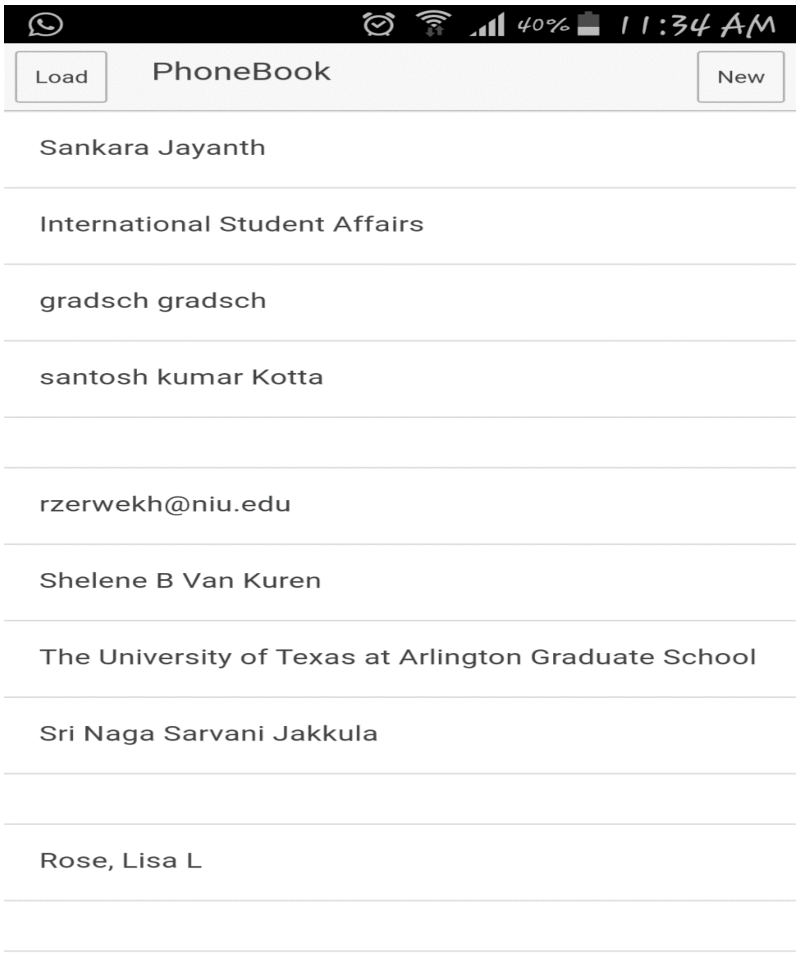


Description:

Here we give a provision of entering the recipient mail address in order to start a conversation. We give a text field to type a message and a button to send the message.

Mockups:

1. <u>Synchronizing Device Contacts Screen</u>



Sankara Jayanth

International Student Affairs

gradsch gradsch

santosh kumar Kotta

rzerwekh@niu.edu

Shelene B Van Kuren

The University of Texas at Arlington Graduate School

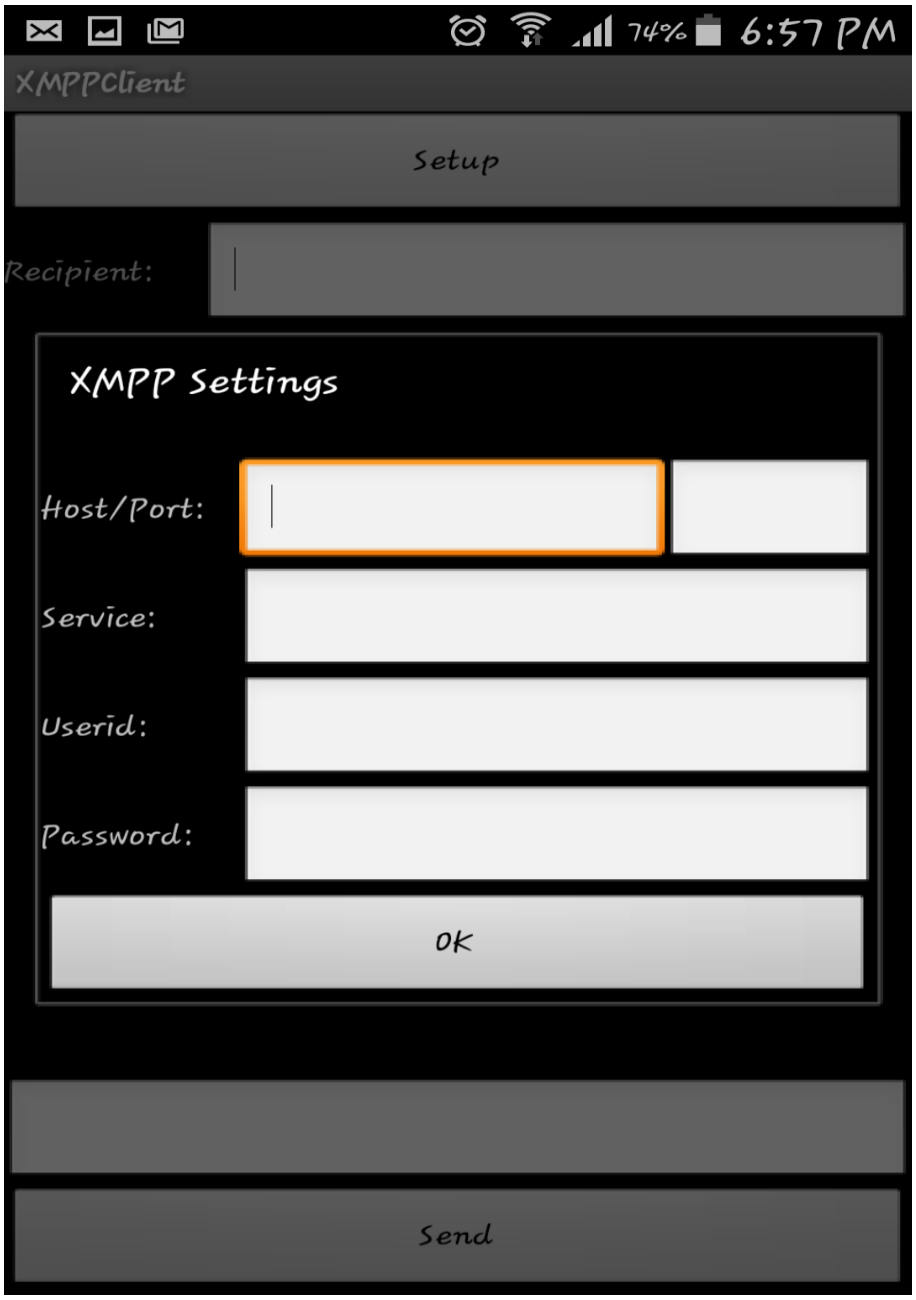
Sri Naga Sarvani Jakkula

Rose, Lisa L

Description:

The above screen displays the contacts that are synchronized from the device on clicking load button.

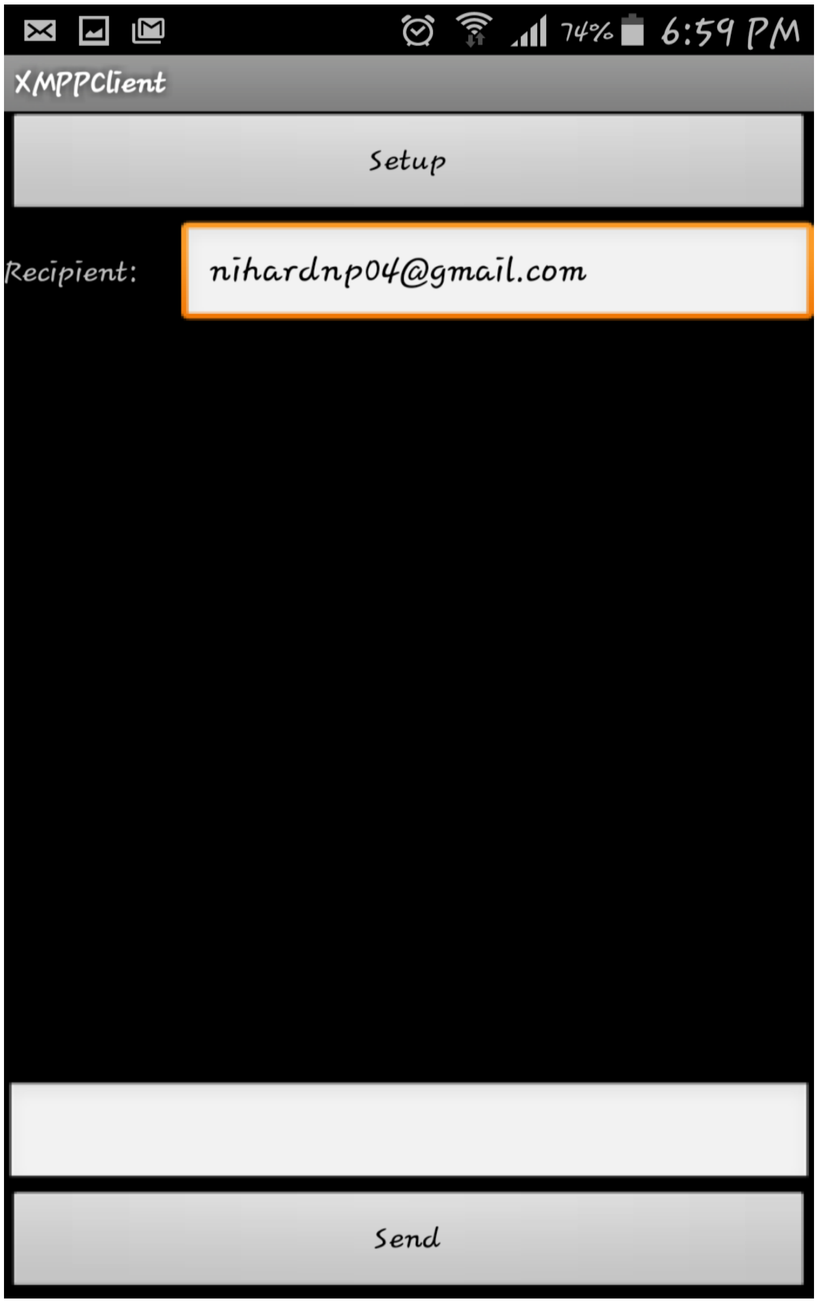
2. <u>Gtalk XMPP settings page:</u>



Description:

The above page enables the user to enter the XMPP setup configuration parameters in the specified fields so as to establish the connection with Gtalk server.

3. Gtalk Chat Window:



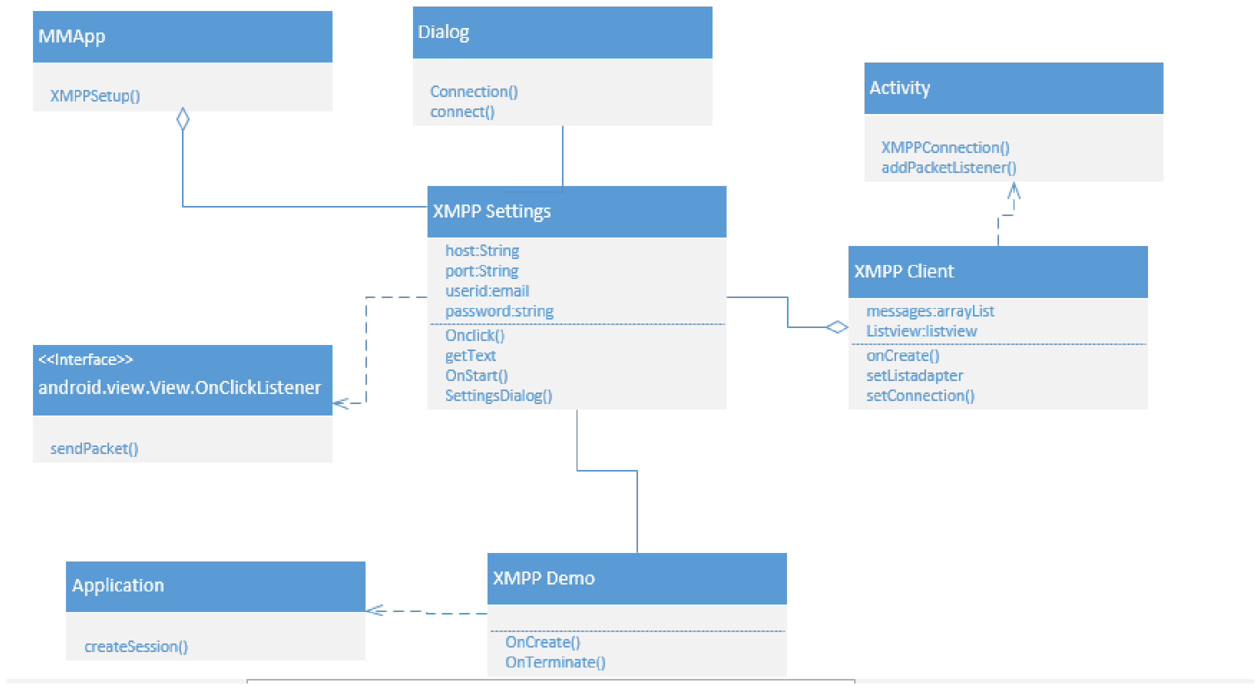
Recipient: nihardnp04@gmail.com

Description:

The above window enables user to enter the recipient details and send/receive messages in the window.

Class Diagrams:

1. Class Diagram for establishing XMPP client connection and sending a message.

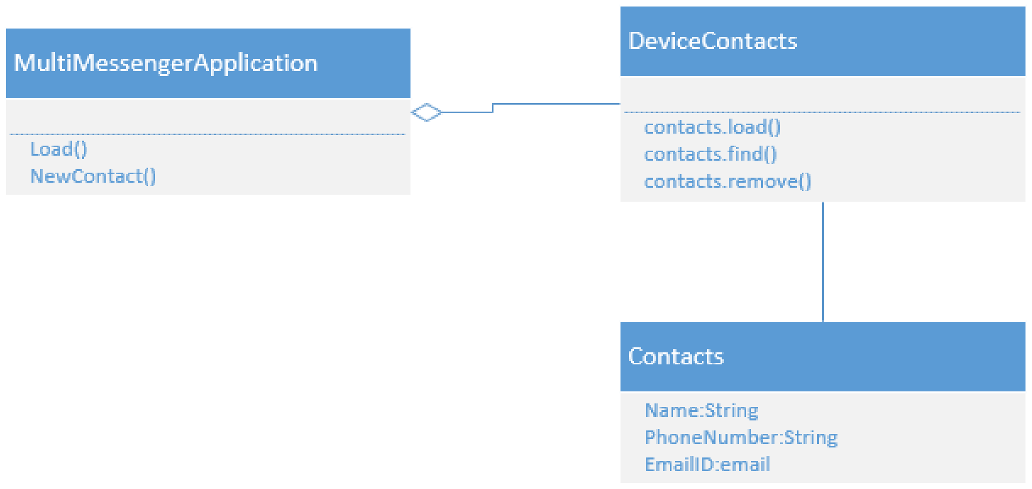


Description:

We have 3 main classes that needs to be implemented in order to establish a XMPP client connection and exchange packets.
They are:

- XMPP Settings: This class is used to setup XMPP connection with the Gtalk server by getting the necessary configuration parameters from the user. It extends Dialog class of Smack API
- XMPP Client: This class is used to send and receive packets of messages from the user and the user at other end. It extends Activity class of Smack API
- XMPP Demo: This class extends Application class of Smack API in which it is used to get the gtalk server interact with our host application.

2. Class Diagram for Synchronizing Contacts from the device



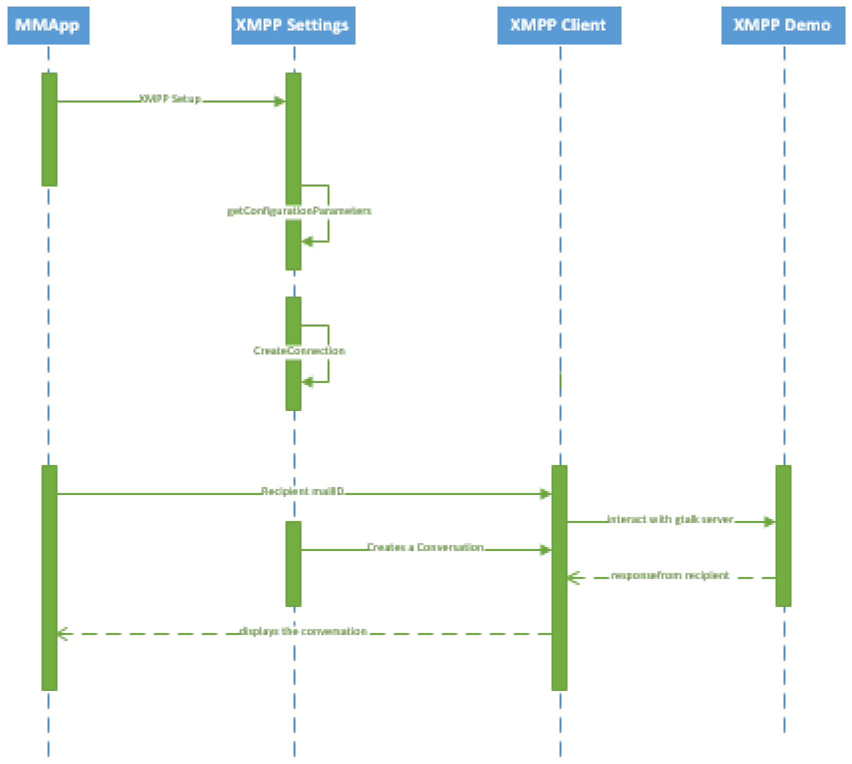
Description:

To synchronize device contacts from the phone, we implement this by using 3 classes.

- MultiMessengerApplication: This is our application class which is used to interact with the device to retrieve the contacts
- DeviceContacts: This class is used to retrieve the contacts from the device Database in order to display in our application
- Contacts: This class has the details of the contact.

Sequence Diagrams:

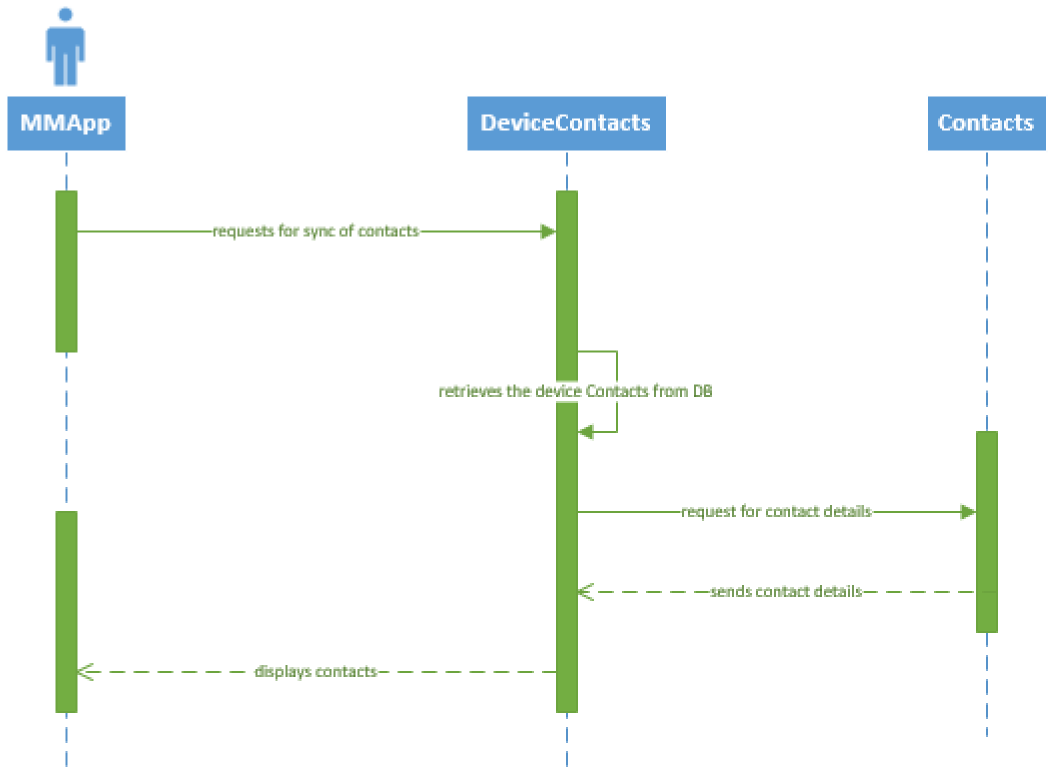
1. Sequence diagram for connecting to gtalk through XMPP



Description:

Here from user interface we request a setup for XMPP connection by entering configuration parameters. These are passed into XMPP settings class in which it creates a connection of XMPP client and sends the request for accessing gtalk to XMPP client class. Then we send the recipient mail ID entered from UI. It establishes a conversation with the recipient. It is displayed on the UI.

2. Sequence Diagram for Synchronizing Device contacts


MMApp
DeviceContacts
Contacts
requests for sync of contacts
retrieves the device Contacts from DB
request for contact details
sends contact details
displays contacts

Description:

Here, application requests for synchronizing the device contacts from the mobile. The contacts are retrieved from the DB through Device Contacts class then this requests the details of the contacts from Contacts class. The contacts are displayed on the UI.

VII. Testing

- **Unit Testing:**

  Unit Testing is software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

  - Testing login :

Test Data

Test Id	Library ID	Password
T001	tarun	a
T002	tarun	abc
T003	nihar	Nihar123
T004	Nihar	nihar123
T005	Ravi	Raviultimate
T006	Ravi	Ravinotultimate

Test Conditions

Test ID	Condition to be tested	Expected Result
T001	Login Validation	Login Successful
T002	Login Validation	Login Failed
T003	Login Validation	Login Successful
T004	Login Validation	Login Failed
T005	Login Validation	Login Successful
T006	Login Validation	Login Failed

- Gtalk Functionality:

Test Data

Test Id	Host	Port	Username	Password
T001	Talk.google.com	5222	sar@gmail.com	Sarvani
T002	Talk.google.com	5222	sar@yahoo.in	Sarvani
T003	Talk.google.com	5223	sar@gmail.com	Sarvani
T004	Talk.google.com	5222	sar@gmail.com	@@@@@
T005	Talk.yahoo.in	5222	sar@yahoo.in	Sarvani
T006	Talk.google.com	5225	Empty	Sarvani
T007	Talk.google.com	Empty	Empty	Empty
T008	Empty	Empty	Empty	Empty
T009	Talk.google.com	5222	sar@gmail.com	Empty

Test Conditions

Test ID	Condition to be tested	Expected Result	Test Cycle			
			S	1	2	3
T001	Gtalk Validation	Successful				
T002	Gtalk Validation	Failed				
T003	Gtalk Validation	Failed				
T004	Gtalk Validation	Failed				
T005	Gtalk Validation	Failed				
T006	Gtalk Validation	Failed				
T007	Gtalk Validation	Failed				
T008	Gtalk Validation	Failed				

T009	Gtalk Validation	Failed	

- Synchronize Contacts page:

1. **Test Data** : Click load, don't click load
2. **Test Conditions** :

Test ID	**Condition to be tested**	**Expected Result**
T001	Sync Contacts Validation On clicking Load	Successful
T002	Sync Contacts Validation On not clicking Load	Successful

- Performance Testing:

Performance testing is done using Yslow plugin of Google Chrome. This testing done to login page, register page and for Gtalk page where we will be sending messages to various other users.

VIII. Implementation

XMPP, Extensible Messaging and Presence Protocol is a communications protocol for message-oriented middleware based on XML (Extensible Markup Language). It enables the near-real-time exchange of structured data between any two or more network entities. Using this we can send or receive messages or data packets on a network.

Smack API, Smack is an Open Source XMPP (Jabber) client library for instant messaging and presence. A pure Java library, it can be embedded into your applications to create anything from a full XMPP client to simple XMPP integrations such as sending notification messages and presence-enabling devices.

Connecting to XMPP server requires knowing of configuration parameters set by XMPP Server.

XMPP connectivity details for Google Talk:

Host: talk.google.com

Port: 5222

Service: gmail.com

Username and Password

There are certain methods in Smack API in order to connect to server through XMPP. The *XMPPConnection* class is used to create the connection to the XMPP server specified by the *ConnectionConfiguration* class which has all the configuration parameters, which uses configuration parameters for establishing connection with the server and to disconnect, use the disconnect() method. Once a connection is established, the user should log in with username and password using the login () method of the Connection class.

We use Chat class of smack API to send messages for the specified recipient. The Chat Manager instance is obtained from XMPPConnection using the getChatManager () method. Chat Manager keeps track on all current chats. Chat is created which will now be series of messages exchanged between two users.

The send Message (String msg) or send Message (Message msg) method is used for sending text messages or message object in the context of a given chat session. Moreover, *MessageListener* can be used to get callbacks of notification of Message from other users on chat. To receive messages, we use asynchronous mechanism through the use of packet listener.

We used Contacts plugin available in ***ngCordova*** list of plugins to synchronize device contacts into our application. ***ngCordova*** Contacts plugin is used to create, remove and search contacts of the device. This plugin defines a global navigator. Contacts object which provides access to the device contacts database. The following are the methods which we use

- navigator.contacts.create: to create a contact on the device
- navigator.contacts.find: to search the contact on the device.
- navigator.contacts.pickContact: method launches the Contact Picker to select a single contact.

Yahoo API:

Yahoo Messenger is instant messaging platform, used on a wide variety of desktop and mobile clients. Yahoo SDK offers developers a platform to manage contacts, real-time instant communications, and data transfer to and from clients throughout the world.

Implementation Steps:

1) We need to create a project for our application in order to access messenger data in our application, we will need to get a Yahoo API key and configure it to use Yahoo Messenger.You need to have an yahoo account for creating a project.
2) There are mainly five steps for using this Yahoo API.
   a) Creating a Yahoo Messenger Open Authentication (OAuth) API Key.
   b) Authenticate with the Yahoo Messenger servers.
   c) Creating a new session
   d) Obtaining contact list.
   e) Sending and receiving messages.

a) Creating a Yahoo Messenger Open Authentication (OAuth) API Key:

We can get Customer Key and Customer secret for an individual user by creating an app and selecting option of creating API key in http://developer.yahoo.com

b) Authenticate with the Yahoo Messenger servers

There are two API calls that are required to authenticate with the Yahoo Direct OAuth API. The first obtains a Pre-Authorized Request Token (PART), and requires your login username, password, and the OAuth Consumer Key (API Key) that was just generated through developer.yahoo.com.

The API call looks like :

https://login.yahoo.com/WSLogin/V1/get_auth_token?&login=username&passwd=mypassword&oauth_consumer_key=consumerkey

The result of this call gives us a request token of format:

RequestToken=Juahfbjjbfsfsfnjjjnsj778sbfa3AYGU1KtB9vUbxlnzfIiFRLP...

The second API call gives us an access token which is used to perform various actions

https://api.login.yahoo.com/oauth/v2/get_token

Various input parameters for this API call are :

`oauth_consumer_key`	The consumer key that will be validated
`oauth_signature_method`	`PLAINTEXT`
`oauth_nonce`	random value
`oauth_timestamp`	A Unix timestamp, expressed as "*seconds* from epoch."
`oauth_signature`	Consumer Secret Key
`oauth_verifier`	None
`oauth_version`	1.0
`oauth_token`	Request token obtained.

Response of this API call gives us access token.

c) Creating session key :

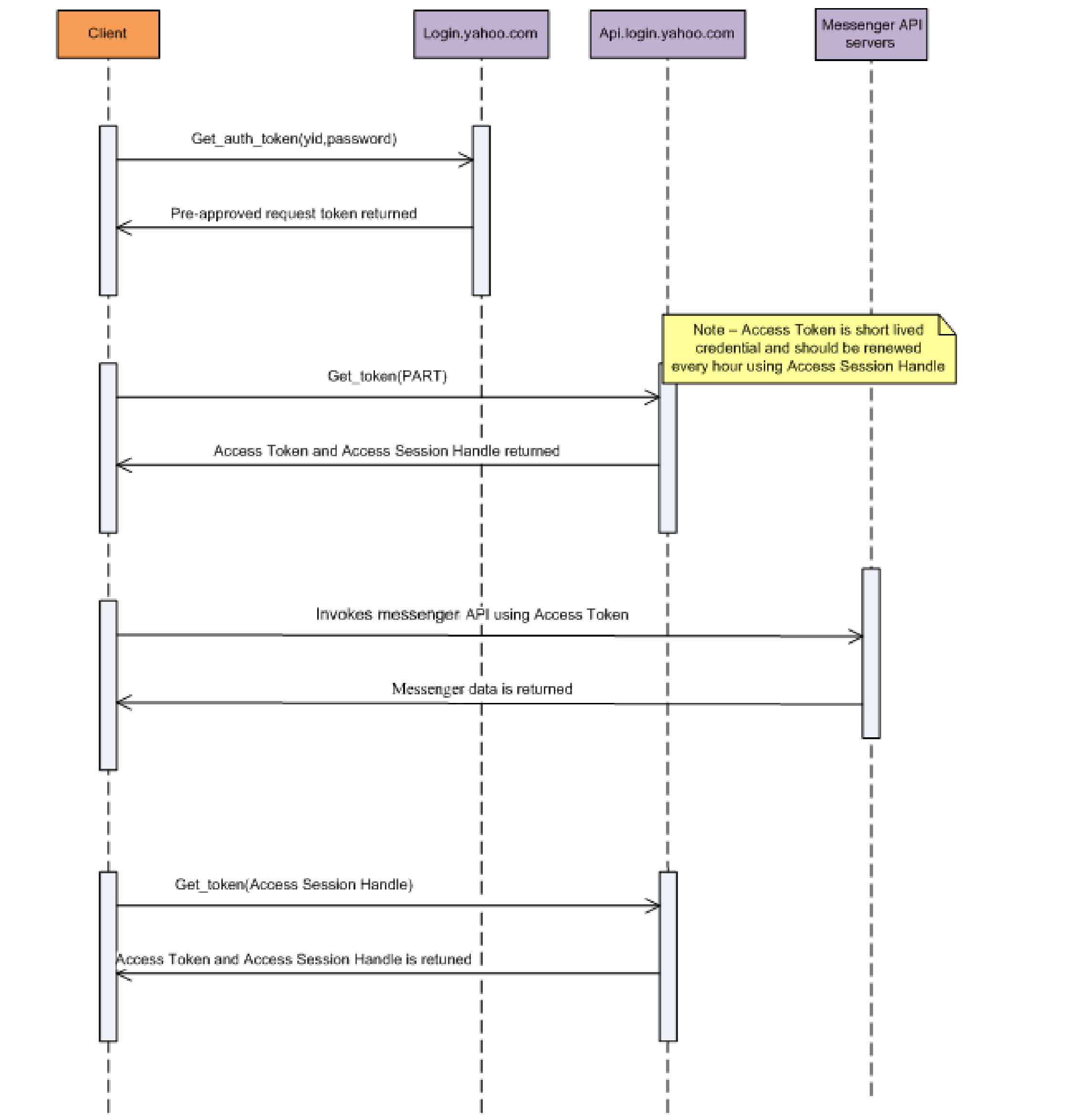
```
POST /v1/session
Host: developer.messenger.yahooapis.com
Authorization: < Standard OAuth credentials >
Content-Type: application/json;charset=utf-8
Content-Length: 2
{
}
```

By calling this we will get a session id which is user for various operations like fetching contact details, sending messages etc.

d) Obtaining contact list:

Example of fetching contact details:

```
GET /v1/contacts?sid=msgrsessionid
Authorization: OAuth
realm="yahooapis.com",
oauth_consumer_key="dj0yJmk9nM9Y29uc3VtZXJzZWNyZXQmeD1lMg--",
oauth_nonce="24829.2331",
oauth_signature_method="PLAINTEXT",
oauth_timestamp="1219450170",
oauth_token="A%3DuqkiebGpiTJl7ThQxU.jDXXaETYyfEy3xAKPyoavokwOOcZcz8Xs_l1Nv
nl._KmCEVCeLkxxT1Y6BgRqf5f98sQWHklBM_anetveR7okK_M_5XEmQ1_1reo3UgKQULT
_dQT8Gao3rgz5rJxgmnYrhdWWdfgTdMQVzpbJT2aGkz59NTK1O8yXVE1EvZUCqju7WiFY
u.WHNEw.9TWq3g--",
oauth_version="1.0",
oauth_signature="O2AQipLITO0aYHKZc9266RzC94%260af0ef7f79bfb89dd6af87589e4c9
7b022f594a3"
```


Client
Login.yahoo.com
Api.login.yahoo.com
Messenger API servers
Get_auth_token(yid,password)
Pre-approved request token returned
Note – Access Token is short lived credential and should be renewed every hour using Access Session Handle
Get_token(PART)
Access Token and Access Session Handle returned
Invokes messenger API using Access Token
Messenger data is returned
Get_token(Access Session Handle)
Access Token and Access Session Handle is retuned

This is the sequence diagram for Yahoo API implementation using Oauth Authentication.

IX. Deployment

Project Management:

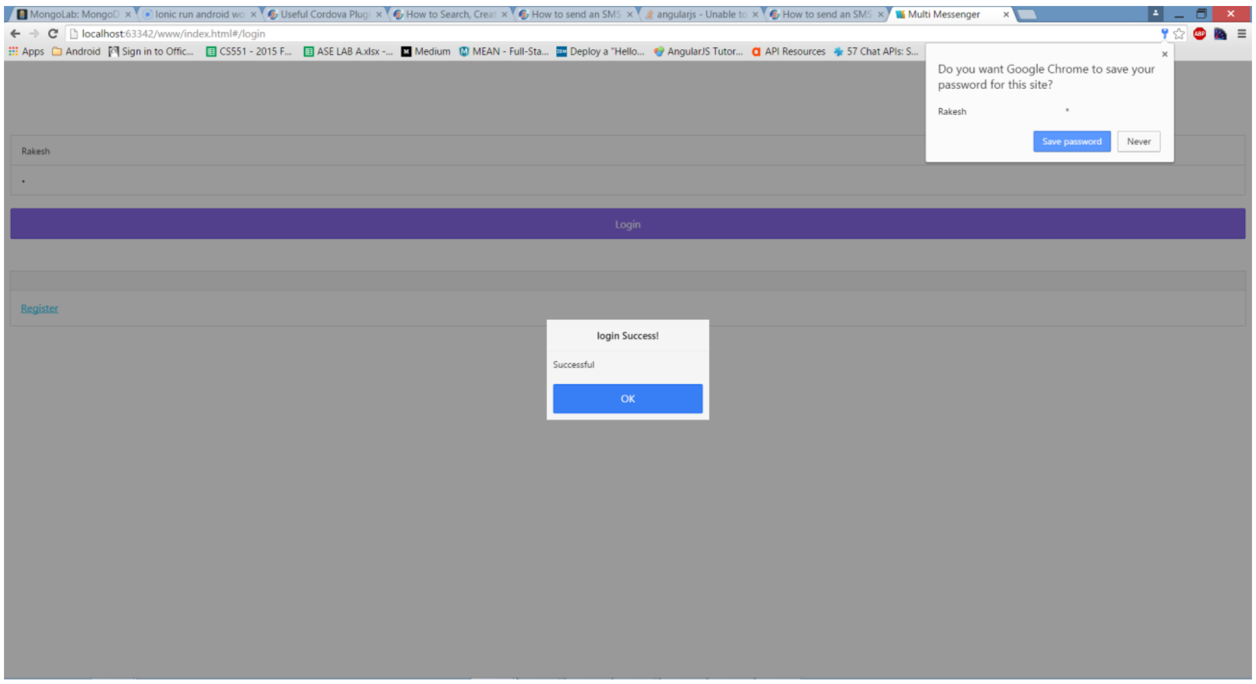
The Project management link of the Project is http://niharase.kanbantool.com/b/180890-ase_project.

GitHub Link:

The Link to second increment documentation and source code in our GitHub is https://github.com/rakeshreddybandi/Multi-Messenger-ASE-GROUP1.

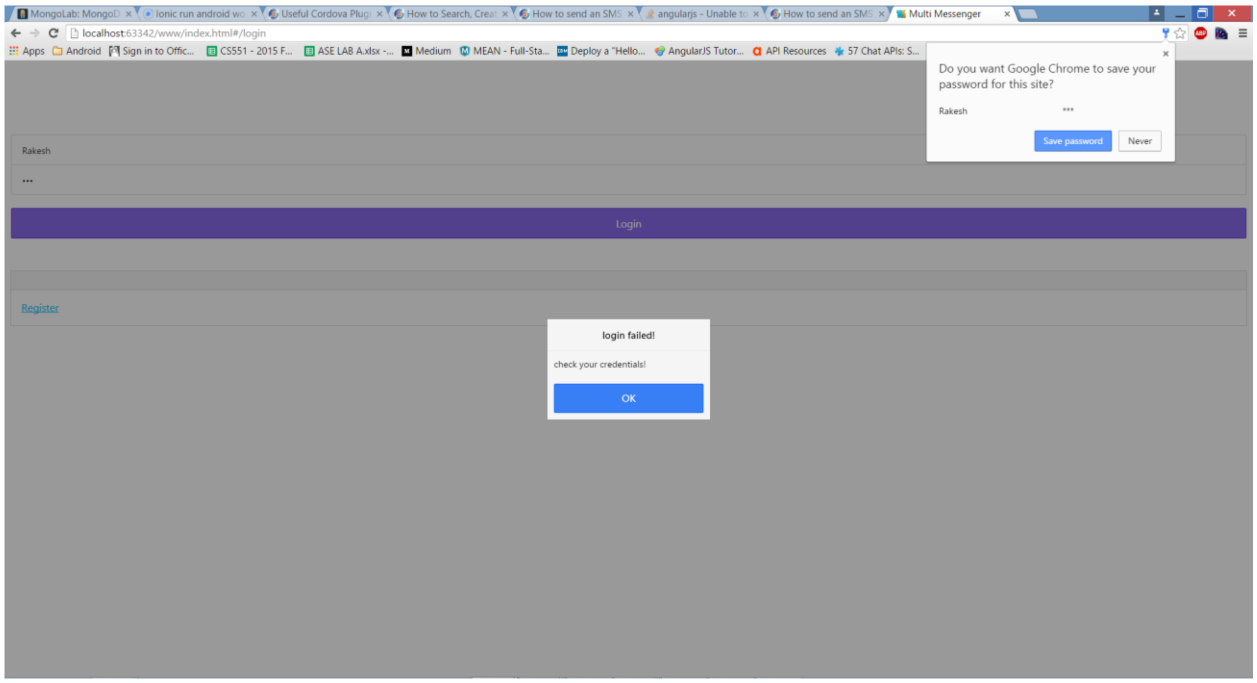
X. Report

- Unit test cases screen shots:
  - Login success :


localhost:63342/www/index.html#/login
Multi Messenger
Do you want Google Chrome to save your password for this site?
Rakesh
Save password
Never
Rakesh
Login
Register
login Success!
Successful
OK

In this we have given the correct credentials username and password of user and logged in to the app. We get the Success message for correct credentials.

- Login Failure:

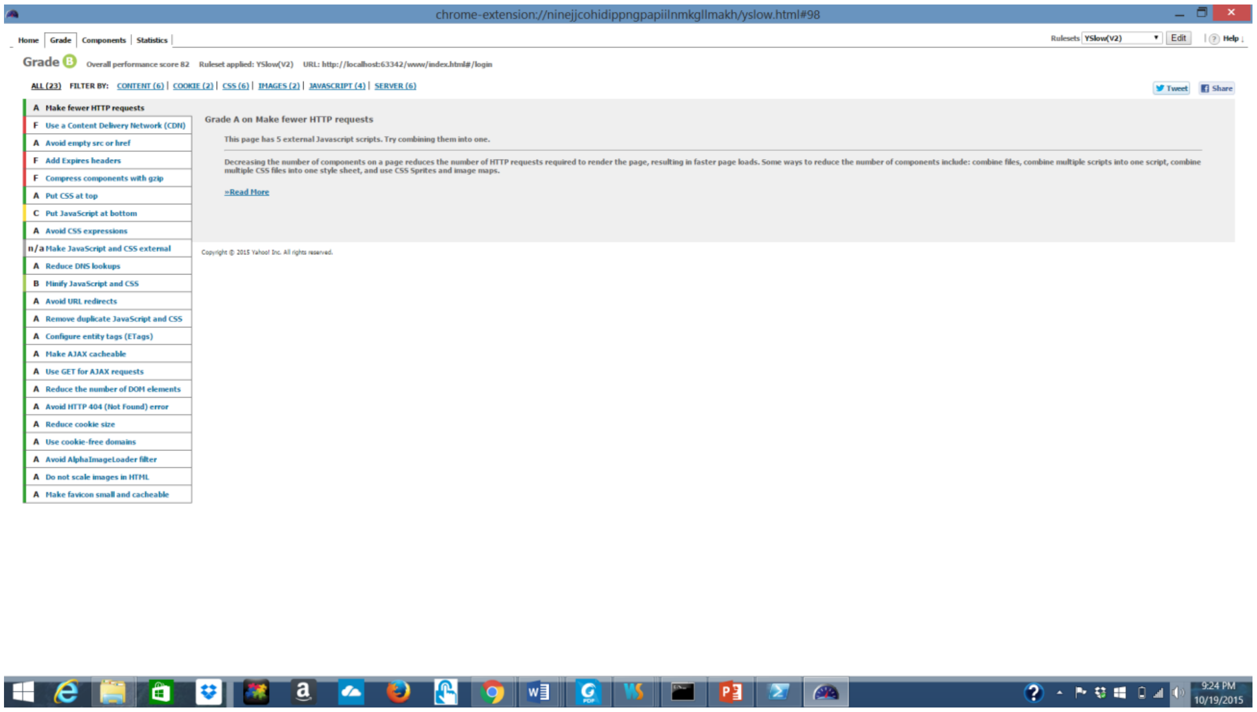

Rakesh
•••
Login
Register
Do you want Google Chrome to save your password for this site?
Save password
Never
OK

login failed!

check your credentials!

In this we have given the incorrect credentials username and password of user and logged in to the app. We get the Failure message for incorrect credentials.

- Performance testing Screen Shots:
    - Login:
        - Yslow Grade:



chrome-extension://ninejjcohidippngpapiilnmkgllmakh/yslow.html#98

Home | Grade | Components | Statistics

Rulesets YSlow(V2) Edit | Help

Grade B Overall performance score 82 Ruleset applied: YSlow(V2) URL: http://localhost:63342/www/index.html#/login

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

A	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
C	Put JavaScript at bottom
A	Avoid CSS expressions
n/a	Make JavaScript and CSS external
A	Reduce DNS lookups
B	Minify JavaScript and CSS
A	Avoid URL redirects
A	Remove duplicate JavaScript and CSS
A	Configure entity tags (ETags)
A	Make AJAX cacheable
A	Use GET for AJAX requests
A	Reduce the number of DOM elements
A	Avoid HTTP 404 (Not Found) error
A	Reduce cookie size
A	Use cookie-free domains
A	Avoid AlphaImageLoader filter
A	Do not scale images in HTML
A	Make favicon small and cacheable

Grade A on Make fewer HTTP requests

This page has 5 external Javascript scripts. 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.

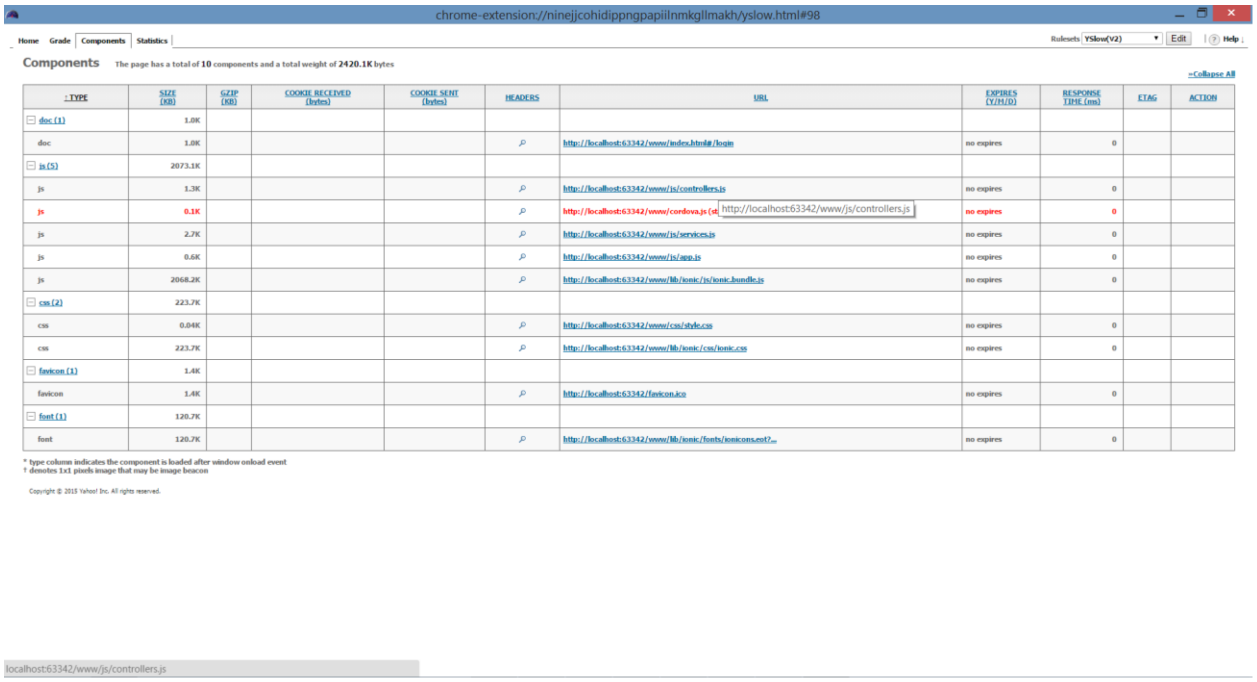
»Read More

Copyright © 2015 Yahoo! Inc. All rights reserved.

9:24 PM 10/19/2015

The above diagram describes about the Yslow grade for login

- Yslow components:



chrome-extension://ninejjcohidippngpapiilnmkgllmakh/yslow.html#98

Home Grade Components Statistics | Rulesets YSlow(V2) Edit | Help

Components The page has a total of **10** components and a total weight of **2420.1K** bytes

»Collapse All

↑TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)	RESPONSE TIME (ms)	ETAG	ACTION
doc (1)	1.0K									
doc	1.0K				⌕	http://localhost:63342/www/index.html#/login	no expires	0		
js (5)	2073.1K									
js	1.3K				⌕	http://localhost:63342/www/js/controllers.js	no expires	0		
js	0.1K				⌕	http://localhost:63342/www/cordova.js (st http://localhost:63342/www/js/controllers.js	no expires	0		
js	2.7K				⌕	http://localhost:63342/www/js/services.js	no expires	0		
js	0.6K				⌕	http://localhost:63342/www/js/app.js	no expires	0		
js	2068.2K				⌕	http://localhost:63342/www/lib/ionic/js/ionic.bundle.js	no expires	0		
css (2)	223.7K									
css	0.04K				⌕	http://localhost:63342/www/css/style.css	no expires	0		
css	223.7K				⌕	http://localhost:63342/www/lib/ionic/css/ionic.css	no expires	0		
favicon (1)	1.4K									
favicon	1.4K				⌕	http://localhost:63342/favicon.ico	no expires	0		
font (1)	120.7K									
font	120.7K				⌕	http://localhost:63342/www/lib/ionic/fonts/ionicons.eot?...	no expires	0		

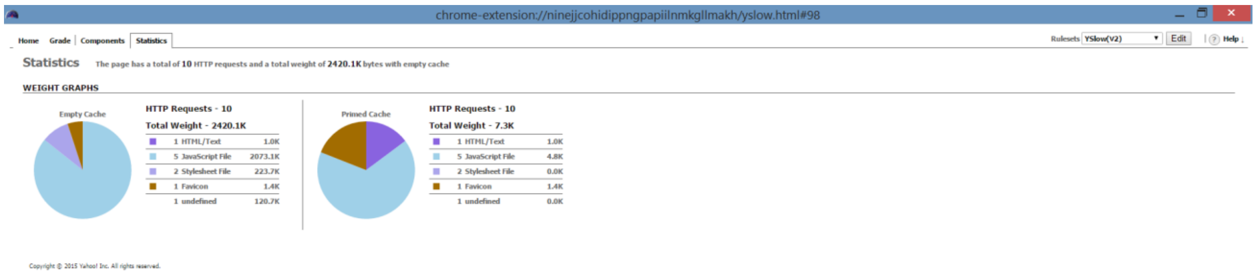
* type column indicates the component is loaded after window onload event
† denotes 1x1 pixels image that may be image beacon

Copyright © 2015 Yahoo! Inc. All rights reserved.

localhost:63342/www/js/controllers.js

The above diagram explains about Yslow components for login page

- Yslow statistics:


chrome-extension://ninejjcohidippngpapiilnmkgllmakh/yslow.html#98
Home
Grade
Components
Statistics
Rulesets YSlow(V2)
Edit
Help

Statistics The page has a total of **10** HTTP requests and a total weight of **2420.1K** bytes with empty cache

WEIGHT GRAPHS

Empty Cache

HTTP Requests - 10

Total Weight - 2420.1K

🟪	1 HTML/Text	1.0K
🟦	5 JavaScript File	2073.1K
🟪	2 Stylesheet File	223.7K
🟫	1 Favicon	1.4K
	1 undefined	120.7K

Primed Cache

HTTP Requests - 10

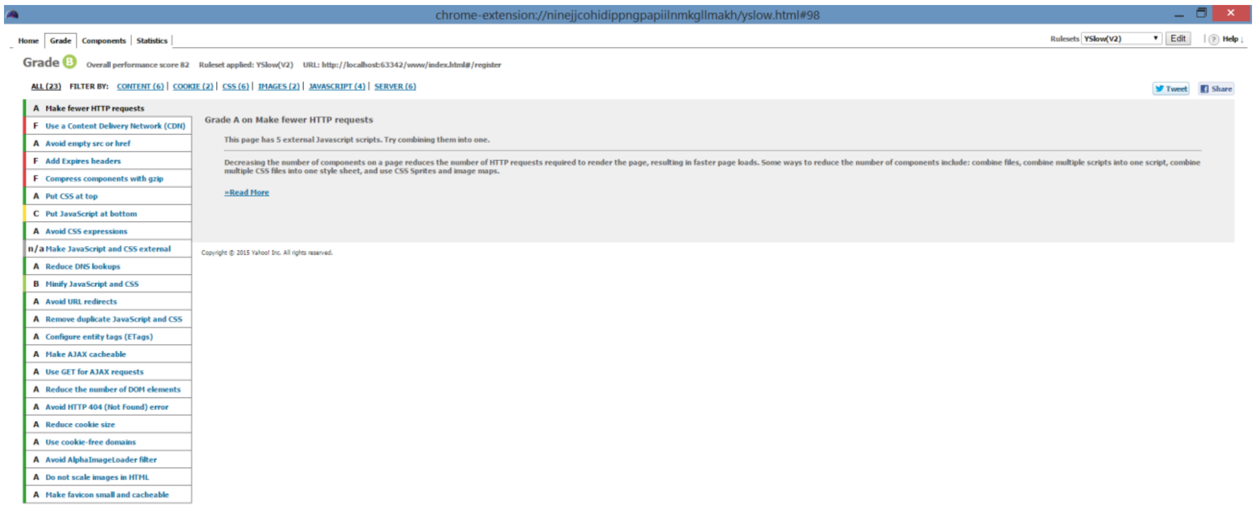
Total Weight - 7.3K

🟪	1 HTML/Text	1.0K
🟦	5 JavaScript File	4.8K
🟪	2 Stylesheet File	0.0K
🟫	1 Favicon	1.4K
	1 undefined	0.0K

Copyright © 2015 Yahoo! Inc. All rights reserved.

The above figure explains about Yslow statistics for login page.

- Register Page
    - Yslow Grade:


chrome-extension://ninejjcohidippngpapiilnmkgllmakh/yslow.html#98
Home
Grade
Components
Statistics
Rulesets
Edit
Help
Tweet
Share

Grade B Overall performance score 82 Ruleset applied: YSlow(V2) URL: http://localhost:63342/www/index.html#/register

Rulesets: YSlow(V2)

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

A	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
C	Put JavaScript at bottom
A	Avoid CSS expressions
n/a	Make JavaScript and CSS external
A	Reduce DNS lookups
B	Minify JavaScript and CSS
A	Avoid URL redirects
A	Remove duplicate JavaScript and CSS
A	Configure entity tags (ETags)
A	Make AJAX cacheable
A	Use GET for AJAX requests
A	Reduce the number of DOM elements
A	Avoid HTTP 404 (Not Found) error
A	Reduce cookie size
A	Use cookie-free domains
A	Avoid AlphaImageLoader filter
A	Do not scale images in HTML
A	Make favicon small and cacheable

Grade A on Make fewer HTTP requests

This page has 5 external Javascript scripts. Try combining them into one.

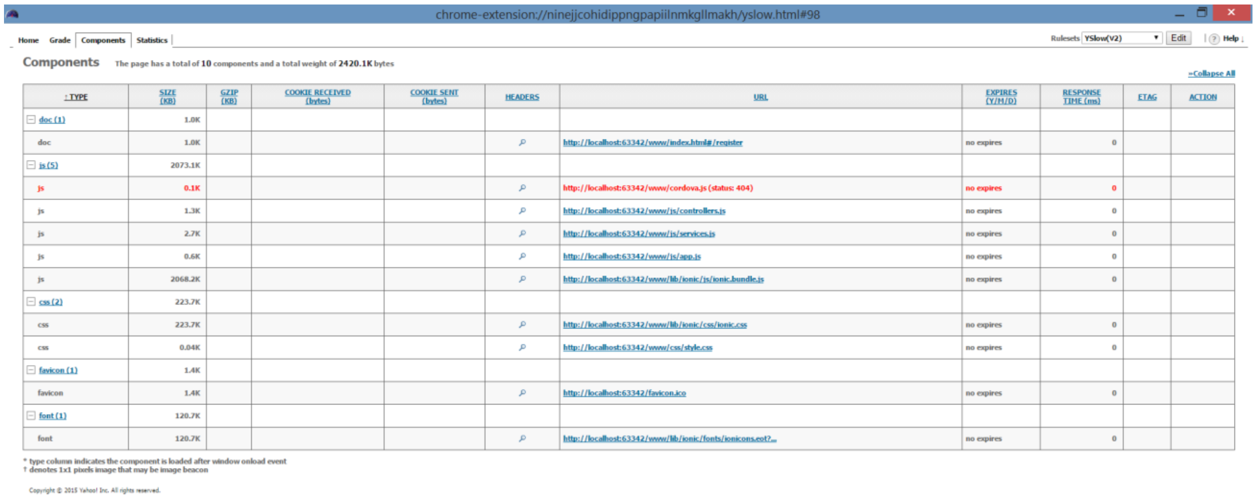
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet,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

»Read More

Copyright © 2015 Yahoo! Inc. All rights reserved.

The above figure explains about Yslow grade for register page.

- Yslow component:



chrome-extension://ninejjcohidippngpapiilnmkgllmakh/yslow.html#98

Home Grade Components Statistics

Rulesets YSlow(V2) Edit | Help

Components

The page has a total of **10** components and a total weight of **2420.1K** bytes

»Collapse All

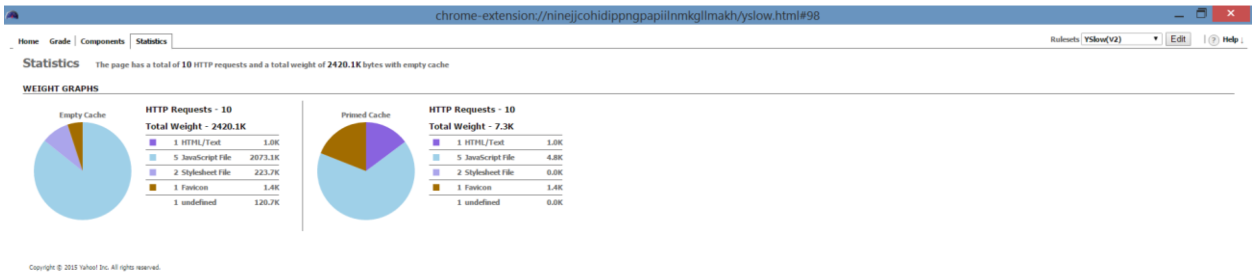
↕ TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)	RESPONSE TIME (ms)	ETAG	ACTION
⊟ doc (1)	1.0K									
doc	1.0K				🔍	http://localhost:63342/www/index.html#/register	no expires	0		
⊟ js (5)	2073.1K									
js	0.1K				🔍	http://localhost:63342/www/cordova.js (status: 404)	no expires	0		
js	1.3K				🔍	http://localhost:63342/www/js/controllers.js	no expires	0		
js	2.7K				🔍	http://localhost:63342/www/js/services.js	no expires	0		
js	0.6K				🔍	http://localhost:63342/www/js/app.js	no expires	0		
js	2068.2K				🔍	http://localhost:63342/www/lib/ionic/js/ionic.bundle.js	no expires	0		
⊟ css (2)	223.7K									
css	223.7K				🔍	http://localhost:63342/www/lib/ionic/css/ionic.css	no expires	0		
css	0.04K				🔍	http://localhost:63342/www/css/style.css	no expires	0		
⊟ favicon (1)	1.4K									
favicon	1.4K				🔍	http://localhost:63342/favicon.ico	no expires	0		
⊟ font (1)	120.7K									
font	120.7K				🔍	http://localhost:63342/www/lib/ionic/fonts/ionicons.eot?...	no expires	0		

* type column indicates the component is loaded after window onload event
† denotes 1x1 pixels image that may be image beacon

Copyright © 2015 Yahoo! Inc. All rights reserved.

The above figure explains about Yslow components for register page

- Yslow statistics:



Rulesets YSlow(V2)

Statistics The page has a total of **10** HTTP requests and a total weight of **2420.1K** bytes with empty cache

WEIGHT GRAPHS

HTTP Requests - 10

Total Weight - 2420.1K

🟪	1 HTML/Text	1.0K
🟦	5 JavaScript File	2073.1K
🟪	2 Stylesheet File	223.7K
🟫	1 Favicon	1.4K
	1 undefined	120.7K

HTTP Requests - 10

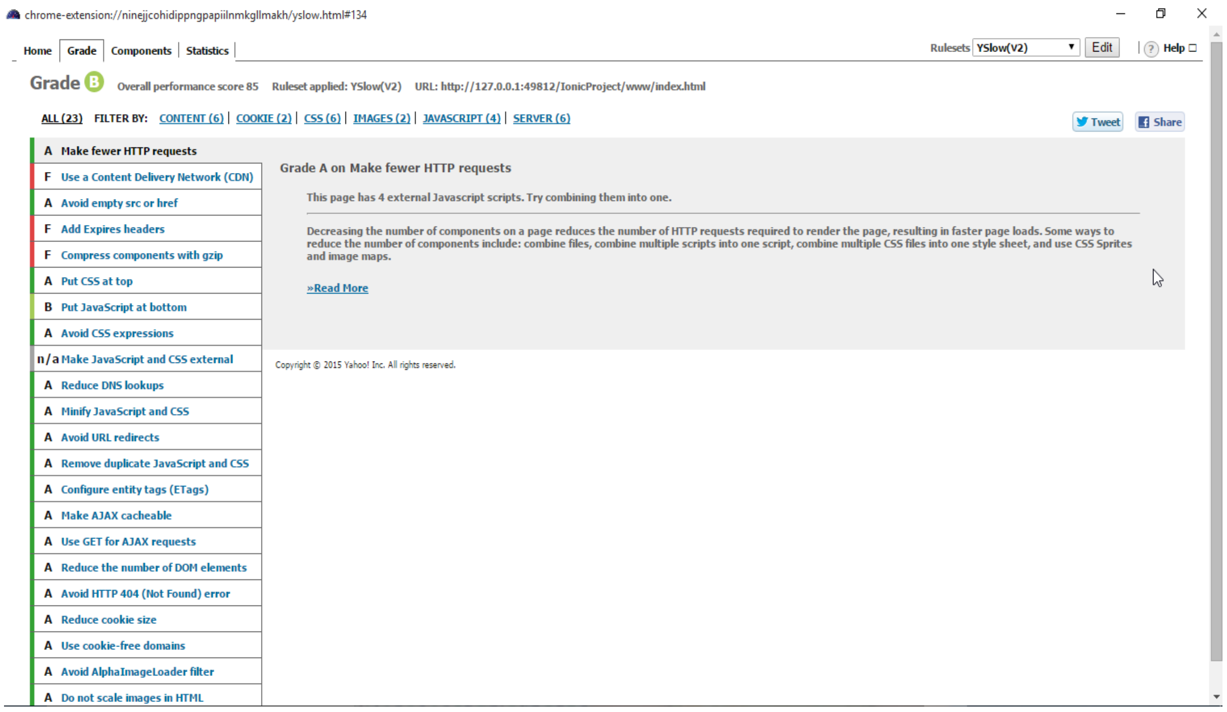
Total Weight - 7.3K

🟪	1 HTML/Text	1.0K
🟦	5 JavaScript File	4.8K
🟪	2 Stylesheet File	0.0K
🟫	1 Favicon	1.4K
	1 undefined	0.0K

Copyright © 2015 Yahoo! Inc. All rights reserved.

The above diagram describes about Yslow Statistics for register page:

- Contacts Synchronization page :


chrome-extension://ninejjcohidippngpapiilnmkgllmakh/yslow.html#134

Home | Grade | Components | Statistics

Rulesets YSlow(V2) Edit | Help

Grade B Overall performance score 85 Ruleset applied: YSlow(V2) URL: http://127.0.0.1:49812/IonicProject/www/index.html

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

A	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
B	Put JavaScript at bottom
A	Avoid CSS expressions
n/a	Make JavaScript and CSS external
A	Reduce DNS lookups
A	Minify JavaScript and CSS
A	Avoid URL redirects
A	Remove duplicate JavaScript and CSS
A	Configure entity tags (ETags)
A	Make AJAX cacheable
A	Use GET for AJAX requests
A	Reduce the number of DOM elements
A	Avoid HTTP 404 (Not Found) error
A	Reduce cookie size
A	Use cookie-free domains
A	Avoid AlphaImageLoader filter
A	Do not scale images in HTML

Grade A on Make fewer HTTP requests

This page has 4 external Javascript scripts. 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.

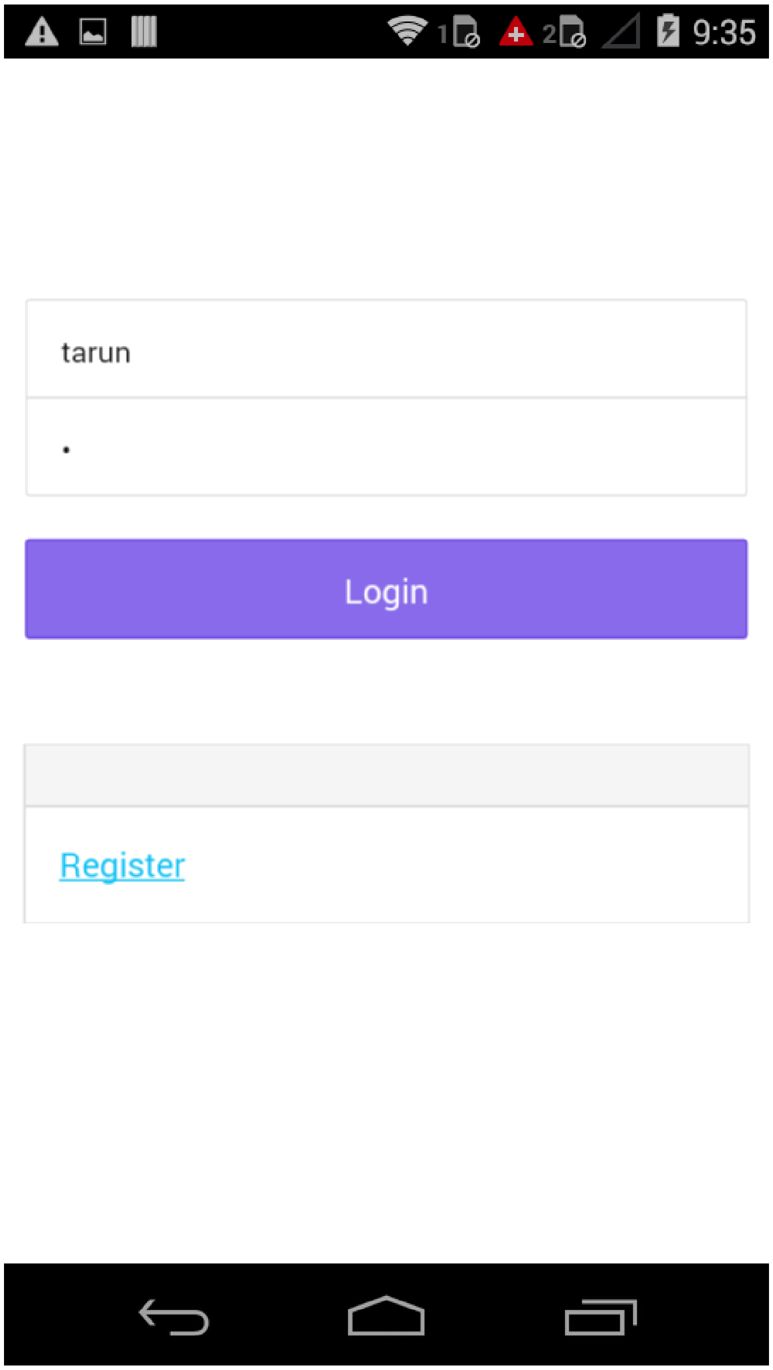
»Read More

Copyright © 2015 Yahoo! Inc. All rights reserved.

The above diagram explains about yslow grade for contacts synchronization page.

Project Implemenation Screenshots:

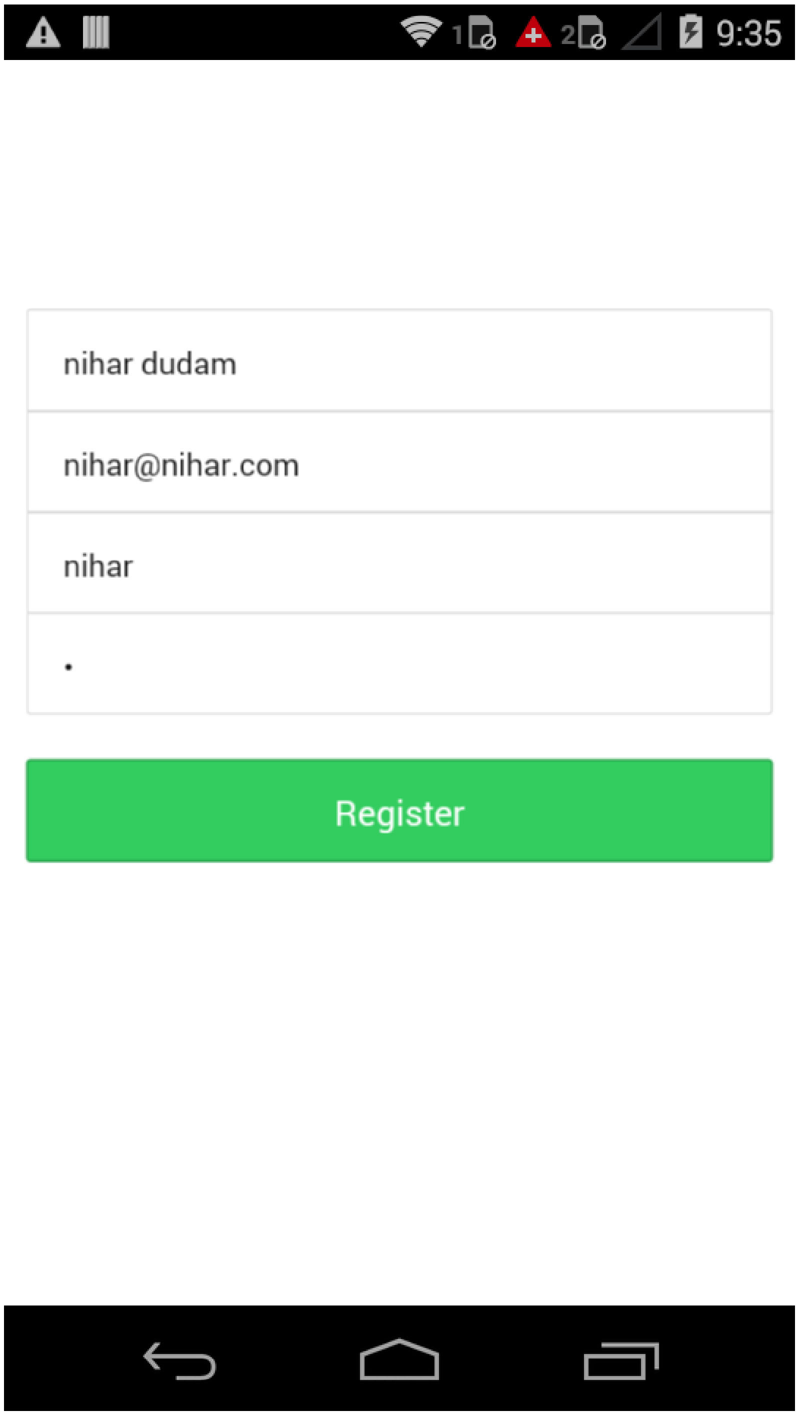
1. Login Activity:



Description:

This is the Login Activity of the Multi-Messenger app where user can login to the app using the correct credentials.

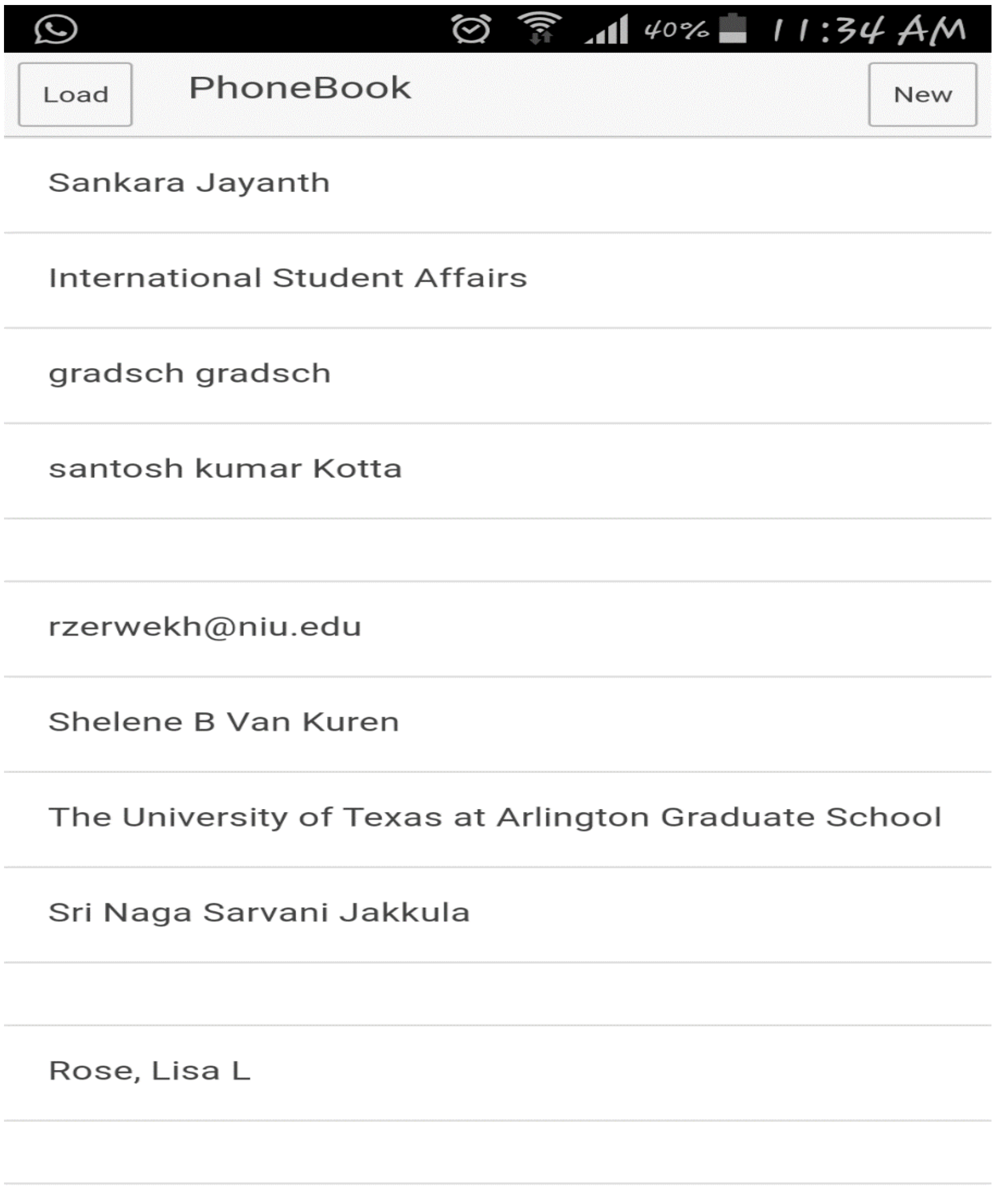
2.Register Activity:



Description:

This is the Register Activity of the Multi-Messenger app where user can register to the app where the user details are stored in Mongolab using MongoDB.

3. Synchronizing Device Contacts Screen:



Sankara Jayanth

International Student Affairs

gradsch gradsch

santosh kumar Kotta

rzerwekh@niu.edu

Shelene B Van Kuren

The University of Texas at Arlington Graduate School

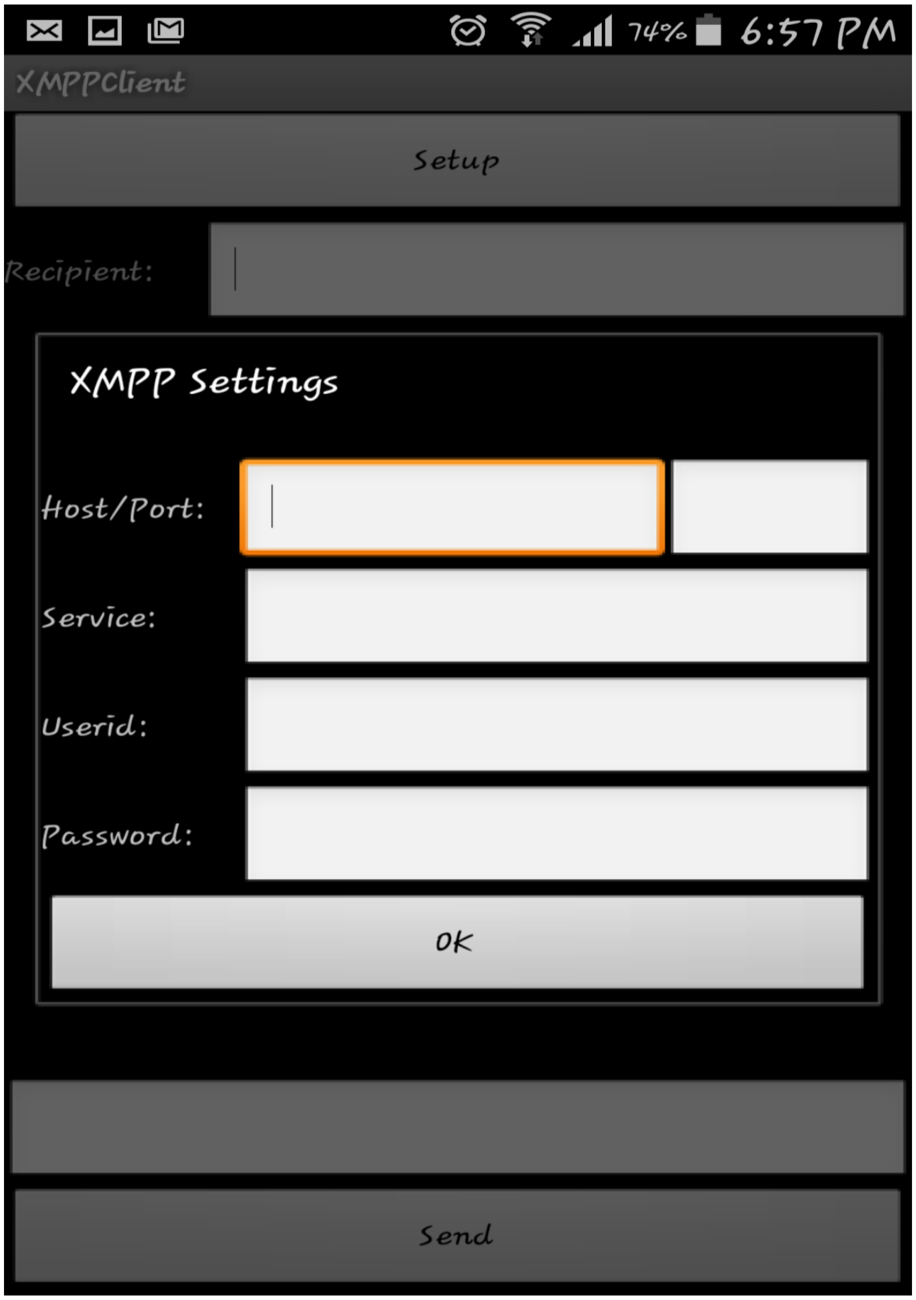
Sri Naga Sarvani Jakkula

Rose, Lisa L

Description:

The above screen displays the contacts that are synchronized from the device on clicking load button.

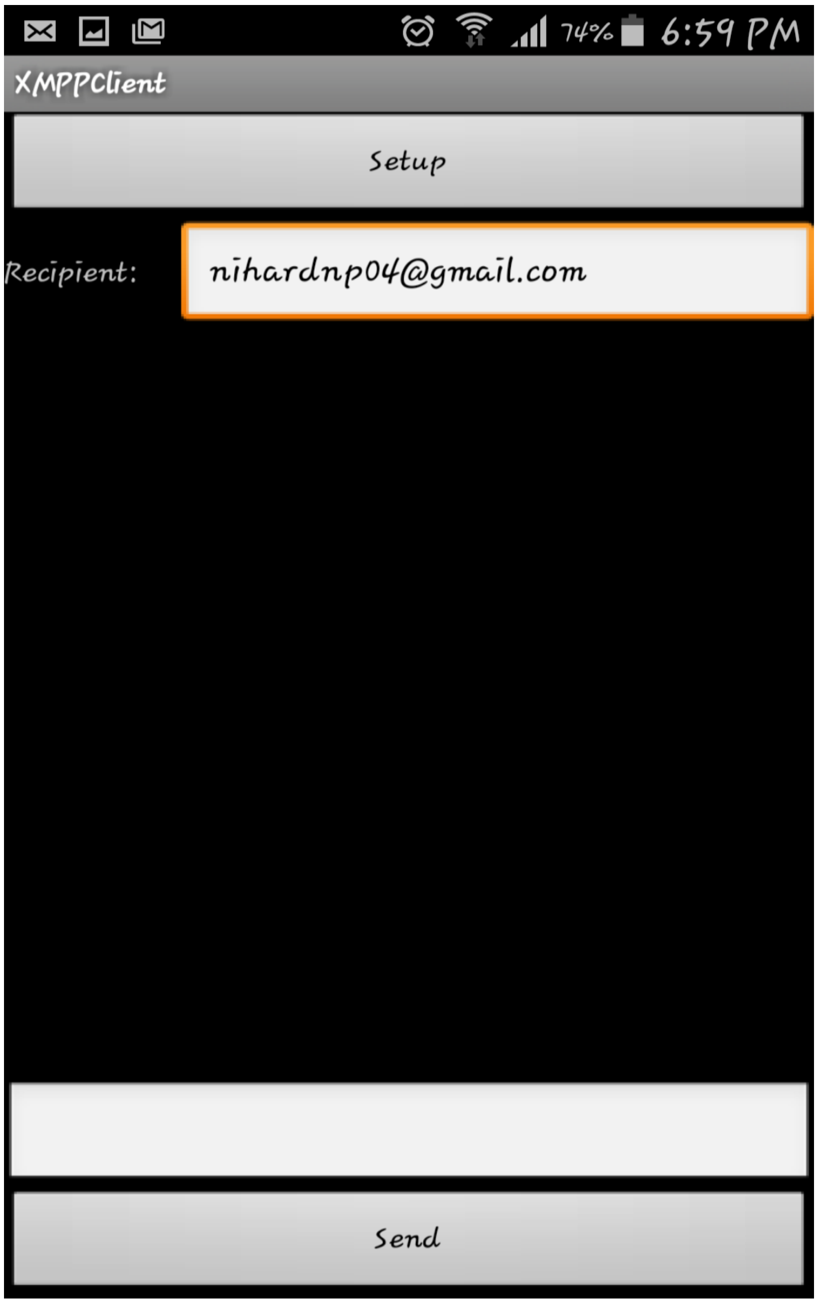
4.Gtalk XMPP settings page:



Description:

The above page enables the user to enter the XMPP setup configuration parameters in the specified fields so as to establish the connection with Gtalk server.

5.Gtalk Chat Window:

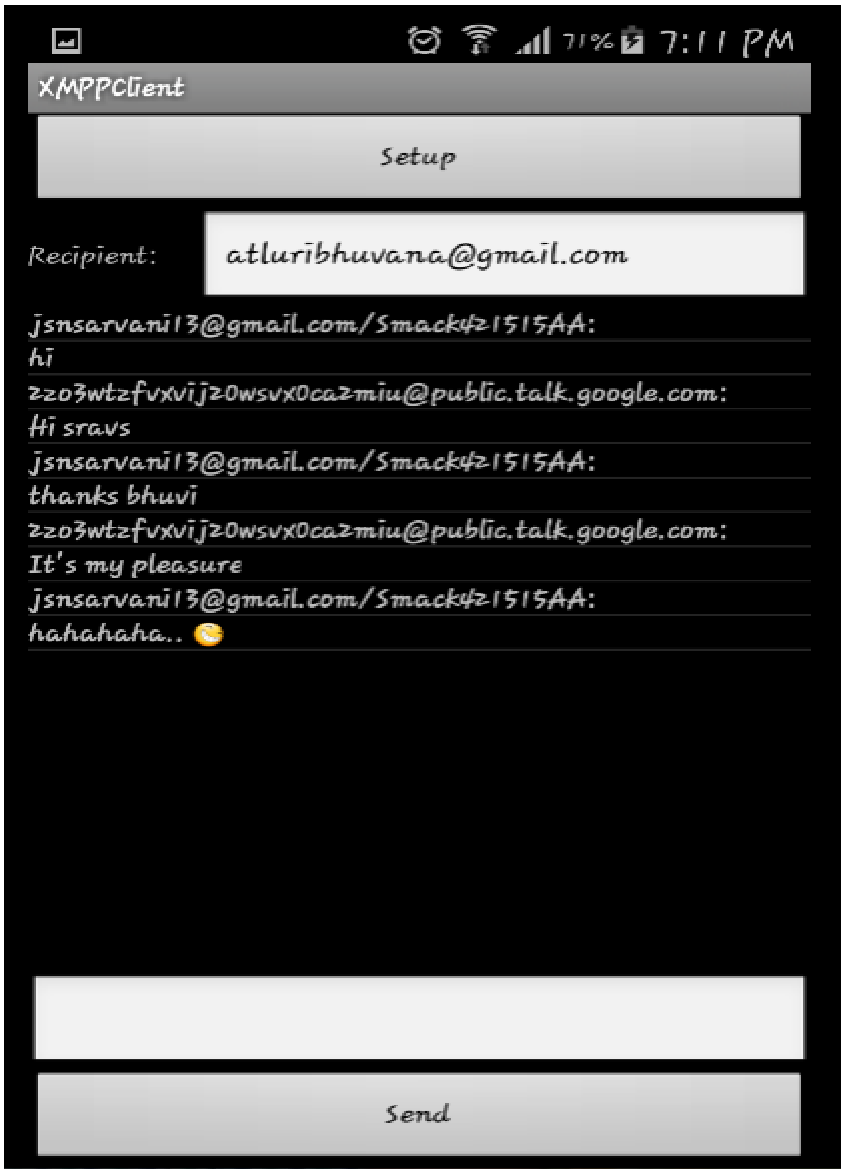


nihardnp04@gmail.com

Description:

The above window enables user to enter the recipient details and send/receive messages in the window.

6. Conversation Window:



Recipient: atluribhuvana@gmail.com

jsnsarvani13@gmail.com/Smack4215154A:
hi
zzo3wtzfvxvijz0wsvx0cazmiu@public.talk.google.com:
Hi sravs
jsnsarvani13@gmail.com/Smack4215154A:
thanks bhuvi
zzo3wtzfvxvijz0wsvx0cazmiu@public.talk.google.com:
It's my pleasure
jsnsarvani13@gmail.com/Smack4215154A:
hahahaha..

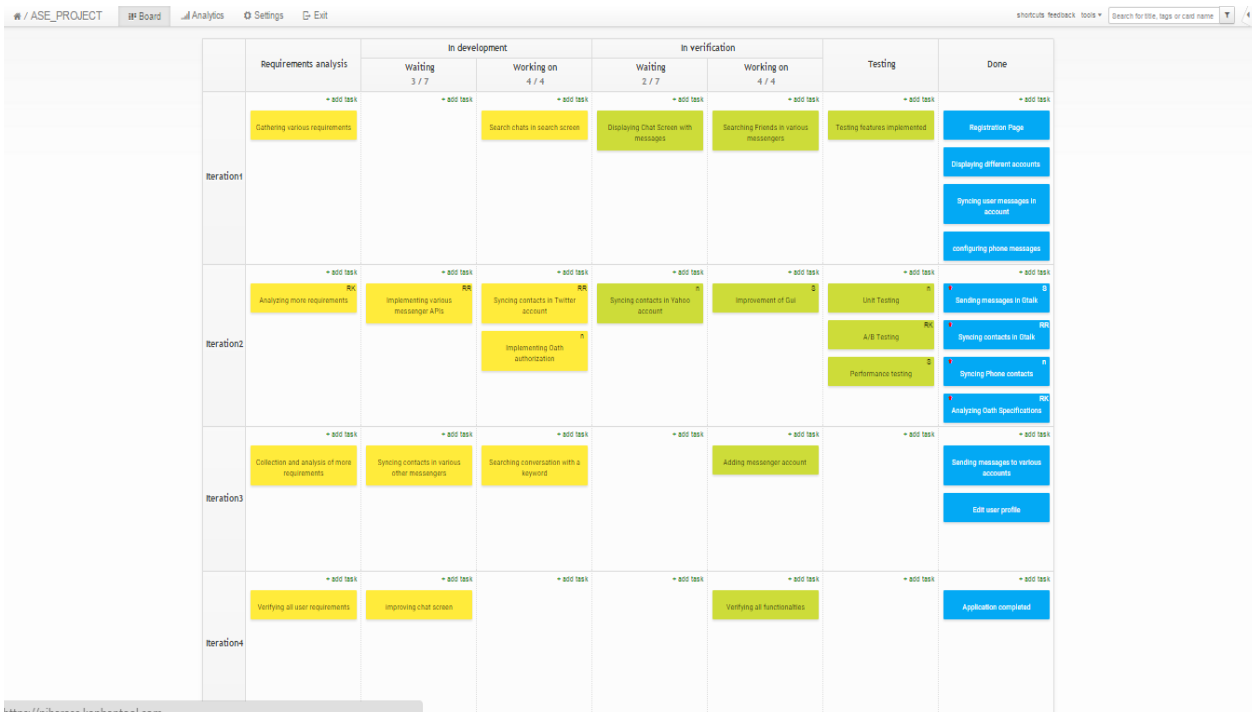
Conversation window of the Gtalk chat.

XI. Project Management

Project Management is done through popular Kanban tool where entire project can be completed efficiently by dividing the tasks among all group members and analyzing the completed work and work yet to be completed effectively.

Project Management URL : https://niharase.kanbantool.com/

- Kanban tool iterations



	Requirements analysis	In development		In verification		Testing	Done
		Waiting 3 / 7	Working on 4 / 4	Waiting 2 / 7	Working on 4 / 4		
Iteration1	Gathering various requirements		Search chats in search screen	Displaying Chat Screen with messages	Searching Friends in various messengers	Testing features implemented	Registration Page Displaying different accounts Syncing user messages in account configuring phone messages
Iteration2	RK Analyzing more requirements	RR Implementing various messenger APIs	RR Syncing contacts in Twitter account n Implementing Oath authorization	n Syncing contacts in Yahoo account	S Improvement of Gui	n Unit Testing RK A/B Testing S Performance testing	S Sending messages in Gtalk RR Syncing contacts in Gtalk n Syncing Phone contacts RK Analyzing Oath Specifications
Iteration3	Collection and analysis of more requirements	Syncing contacts in various other messengers	Searching conversation with a keyword		Adding messenger account		Sending messages to various accounts Edit user profile
Iteration4	Verifying all user requirements	improving chat screen			Verifying all functionalties		Application completed

The above figure explains about various iterations of the project.

As we can see, for the first increment Registration of a page and syncing phone messages are completed.

In the second increment, various requirements users may have are gathered and analysed. Later we have concentrated in fetching phone contacts and Gtalk API and we were successful in fetching contacts of the user phone and also we were able to send messages through our messenger to another Gtalk user using Gtalk API and XMPP connections.

Syncing contacts of Twitter and Implementing Oath authorization are under development stage. In the verification stage fetching contacts from Yahoo account and improvement of GUI are in verification stage.

All the tasks are divided in a good proportion among the group members by assigning specific time and deadline to a task. Tasks which are in the development and verification stage could be completed in iteration 3.

- Implementation status report:

I. Work completed

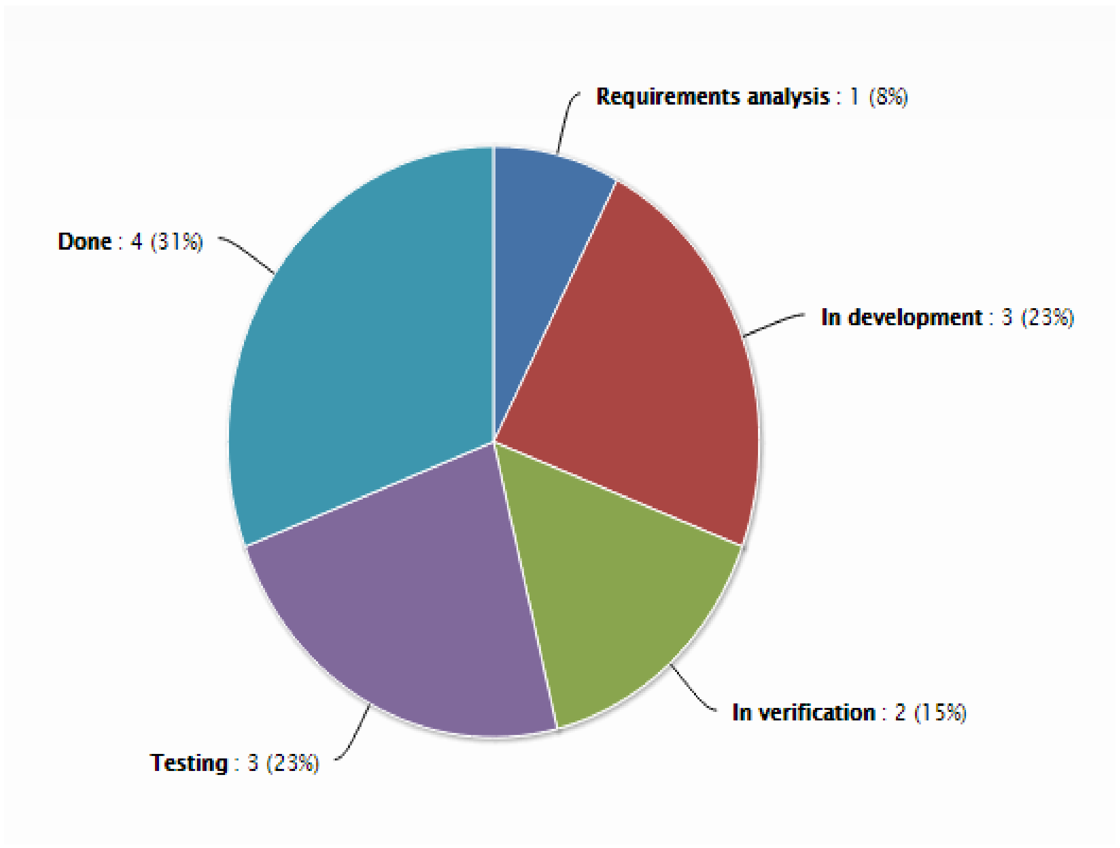
Member	Task Description	Member Responsibility	Contribution %	Time (hours)	Comments /Issues
Rakesh	1. Syncing contacts in Gtalk. 2. Implementing various messenger API's 3.Syncing contacts in Twitter	Develop, design, build and testing the task	100	18	twitter API in progress
Nihar	1.Implementing Oath authorization 2.Syncing contacts in Yahoo API 3. Unit testing 4. Syncing Phone contacts	Develop, design, build and testing the task	100	20	Yahoo API in progress
Sarvani	1.Sending messages in Gtalk 2. Improvement of GUI 3. Performance Testing	Develop, design, build and testing the task	100	22	
Ravi KIran	1.Analyzing more requirements 2.Analyzing Oath specifications 3.sending messages Yahoo API 4.A/B Testing	Develop, design, build and testing the task	100	20	

II. Work yet to be completed

Member	Task Description	Member Responsibility	Time (hours)	Comments /Issues
Rakesh	1.Implementation of Twitter API and various other APIs in our Application 2. Testing the application 3. Improvement of GUI 4. Integration of API's in single window	Develop, design, build and testing the task	40	
Nihar	1. Implementation of Yahoo API and various other API's. 2. Testing the application 3. Integration of API's in single window 4. Keyword search of messages.	Develop, design, build and testing the task	40	
Sarvani	1.Sending messages in Gtalk 2. Improvement of GUI 3. Testing the application 4. Integration of API's in single window	Develop, design, build and testing the task	40	
Ravi KIran	1. Implementation of Yahoo API and various other API's. 2.Testing the application 3. Integration of API's in single window 4. Keyword search of messages.	Develop, design, build and testing the task	40	

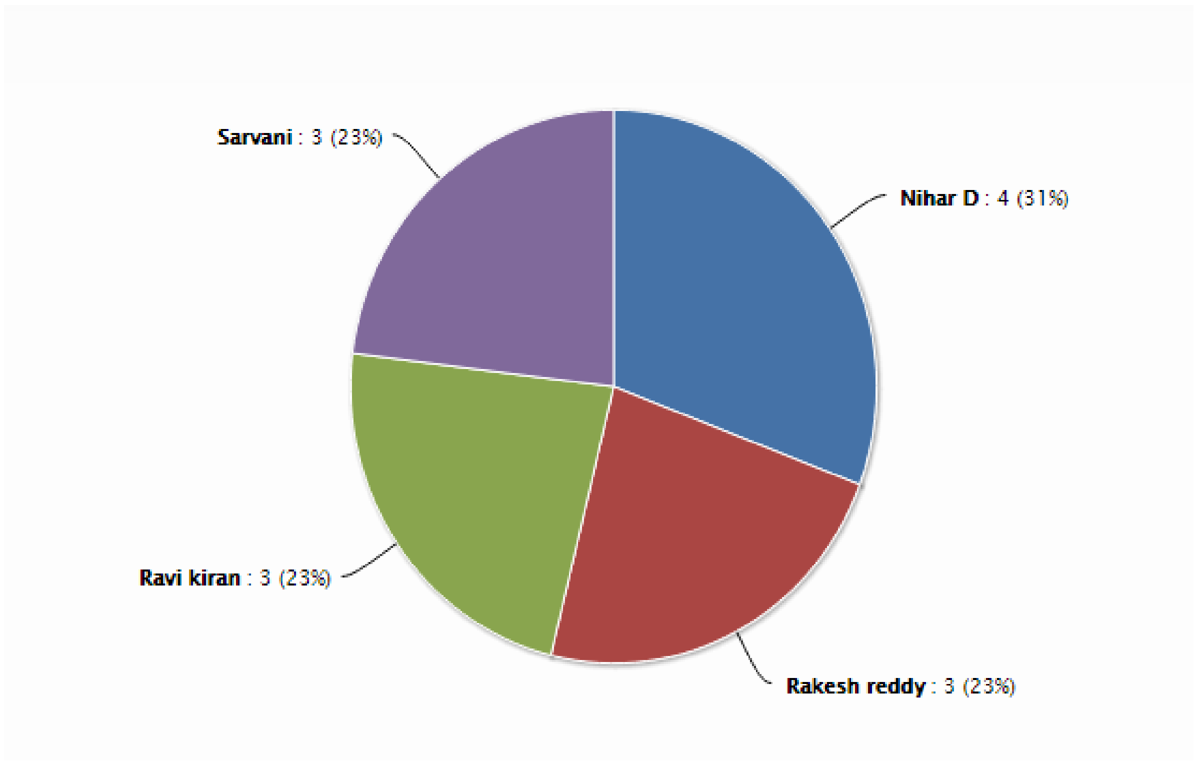
- **Analysis graphs**:

  - Increment Analysis graph:



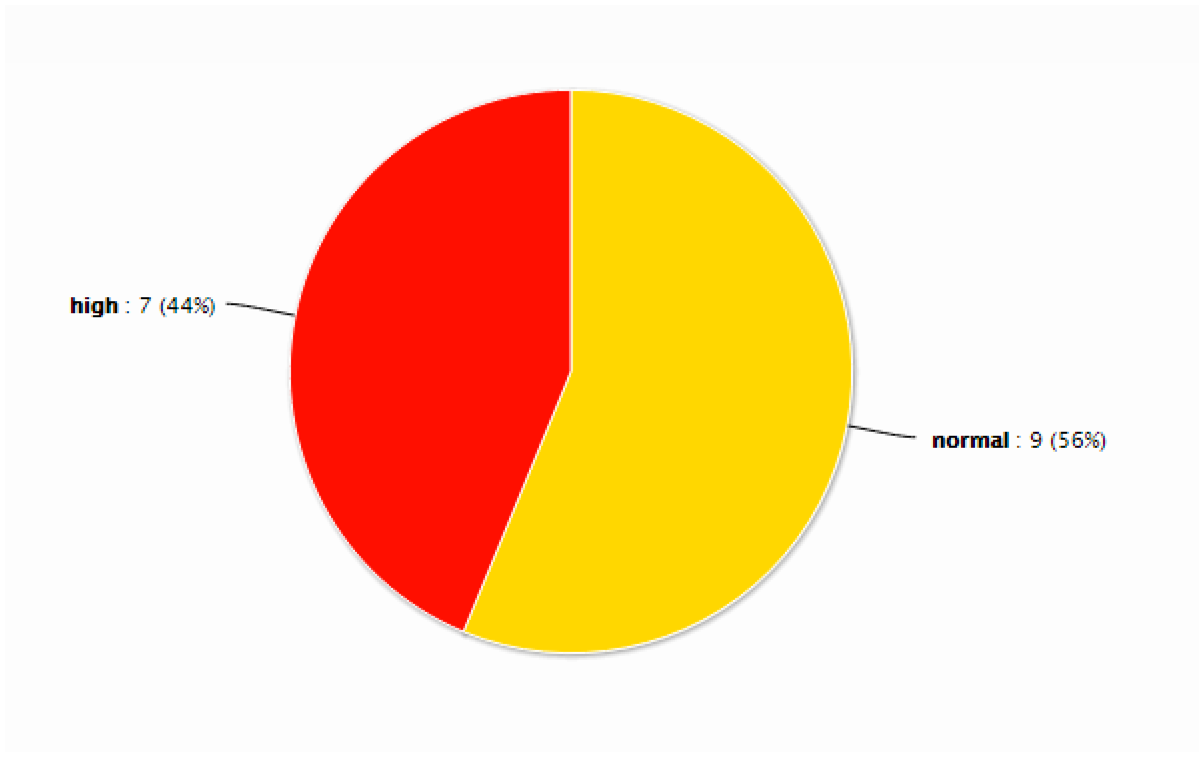
The above diagram explains about the various tasks position in various divisions available in Kanban tool

- Assignment division graph



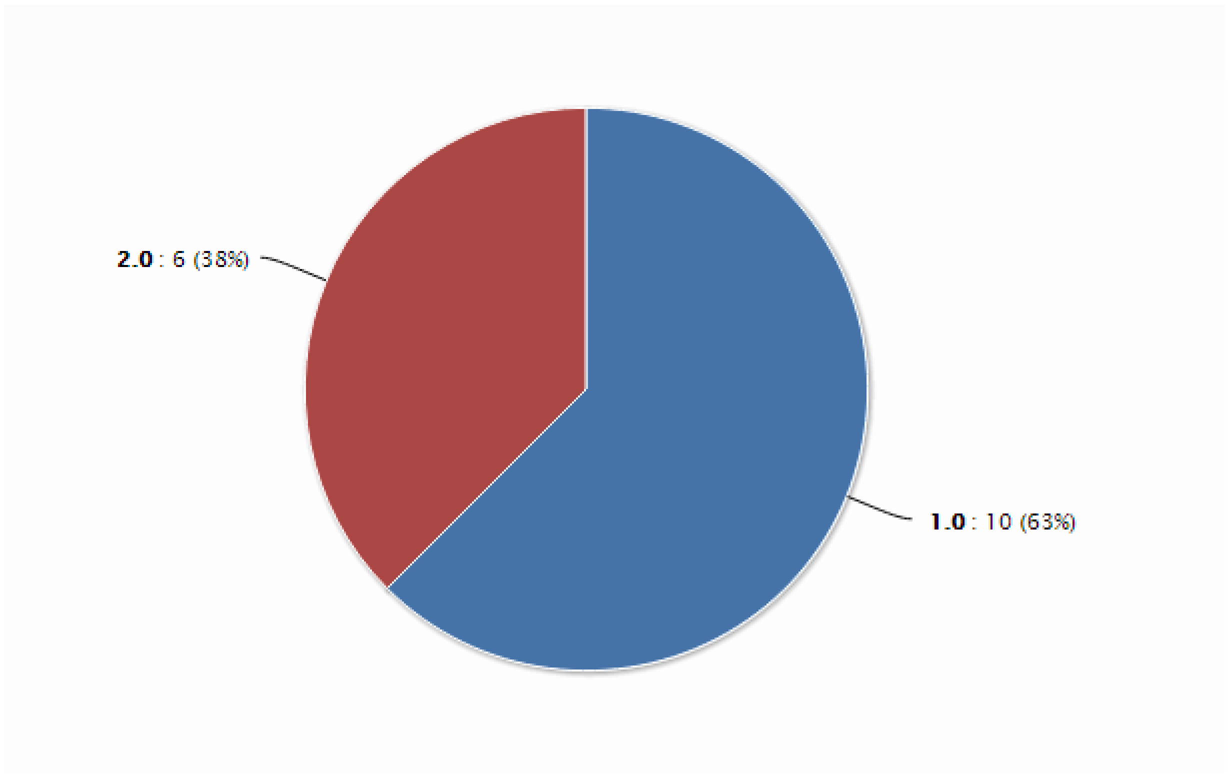
The above diagram describes about the amount of task difficulties faced by each group member.

- Priorities analysis graph



The above diagram explains about priority level given to various tasks.

- Task difficulty graph:



The above figure describes about difficulty level of various tasks in this specific increment.

XII. Bibliography

https://developer.yahoo.com/messenger/

https://dev.twitter.com/rest/reference/get/direct_messages/sent

https://dev.twitter.com/rest/reference/get/direct_messages/show

https://dev.twitter.com/rest/reference/post/direct_messages/new

https://en.wikipedia.org/wiki/XMPP

https://developers.google.com/talk/jep_extensions/oauth

http://developer.samsung.com/technical-doc/view.do?v=T000000119

http://ngcordova.com/docs/plugins/

https://en.wikipedia.org/wiki/Unit_testing