

**Pune Vidyarthi Griha’s**

**College of Engineering and Technology, Pune**

**T.E. MINI PROJECT**

**(Computer Engineering)**

**Skill Development Laboratory**

**Semester I**

**[Academic Year 2018-19]**

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Roll No.** | **Name** |
| 1 | 6008 | Shantanu Kalamdane |
| 2 | 6015 | Khushboo Mundada |
| 3 | 6024 | Parth Parikh |
| 4 | 6025 | Akash Rasal |

**GROUP ID**

**11**

**TITLE OF PROJECT**

SPArK - INSTANT MESSENGER

**PROBLEM STATEMENT**

Develop an Android application to enhance existing messaging and communication platforms..

**ABSTRACT**

Messaging apps are platforms that enable messaging, many of which started around but many of which have now developed into broad platforms. Messaging apps are the most widely used smartphone apps with in 2018. As people upgraded in the 2010s from feature phones to smartphones, they moved from traditional calling and SMS (which are paid services) to messaging apps which are free or only incur small data charges.

Our project focuses on creating an Instant Messaging Application which is freeware and user friendly. The project intends to enhance the features of some existing chatting applications. SPArK has the following unique features :

**Online and Offline Messaging :** SPArK let’s its users message even those friends who don’t have a smartphone or an active internet connection. The receiver need not have the app installed. If the receiver is a SPArK user, the message will be sent through our realtime database at no cost to the user. But, if the receiver is not a SPArK user, a normal SMS will be sent instead. The users enjoys hassle free communication powered by our user friendly interface.

This feature is very powerful in a market like India, where people turn off their data to save costs or battery or are just in a bad network area. **Pinned Message** : The user can pin any message i.e. if the user does a long press on any message, it will get pinned at the top of the chat window and stay there for 24 hours. This provides the other end user to take a quick look at an important message.

**Reminders :** SPArK lets users set and share a common reminder for on their respective phone as well as for the reciever. Just type ‘/’ in a chat window and a window pops up. Set the DD:MM:YY : HH:MM and the event and your reminder is all set.

**Blacklist** : SPArK lets users select certain keywords and messages for which they don’t want to be notified or disturbed. These messages will be silenced and will be shown later if the user wants to.

**Scheduled Messages** : SPArK gives users a handy tool called Scheduled Messages. With this, if a user is busy, he/she can set a timer to send a message and the app will do the rest.

**User Friendly Experience** : SPArK was designed for powerful features and tools but with simplicity in mind. The app has a clean and clutter free User Interface that is easy to use and quick to adapt to.

Customization in the chat colours as well as chat list are just some of stand-out UI features.

**TECHNICAL KEYWORDS**

* Listeners
* Handlers
* Hashmap / Map
* SMS Manager
* Alarm Manager
* SQLite
* Firebase Realtime Database
* Calendar and Date Input Listeners
* Swipe Listener

**GOALS & OBJECTIVES**

* To create an app that will enable people to easily and effectively communicate.
* Provide a User Friendly Interface.
* Incorporate features that are not found in existing messaging platforms.
* Create a small size android application.
* Compatible with 80% android devices.

**INTRODUCTION**

**“SPARK”** is an android app that provides smart enhancements for the ease of user communication. It connects people regardless where they are actually located.

In the company, colleagues can send and reply instant message in real time without face to face, meanwhile the work report can be shared during the instant chat session. People can speak to multiple people in the virtual conference, share ideas and get conclusions. People on a business trip can contact the co-workers inside the company through IM as well. What’s more, the staff can talk to customers or vendors online as well, in other word, now people can do business through the instant messenger direct rather than use the traditional method like make phone calls. Along with long distance messaging, Spark also provides various features to the users, like reminders, scheduled messages, ‘pin a message’ & blacklist messages.

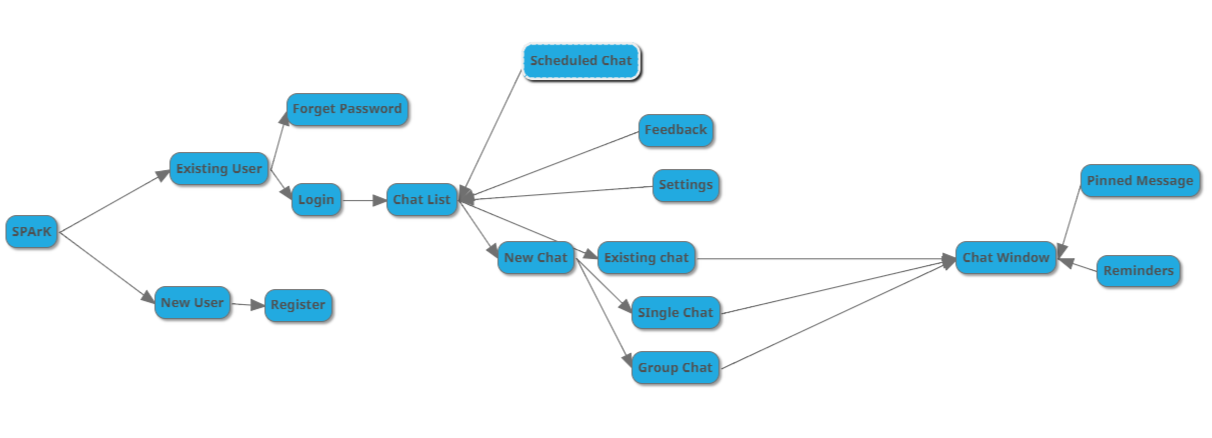
Our app is developed using firebase, java & SQLite at the backend & android studio as the front end tool. The recommended API for **SPArK** is 20 & above.

**Literature Survey**

A comprehensive comparison of existing messaging platforms like Whatsapp, Facebook Messenger, Line, Blackberry Messenger, etc. was done understand stand-out features of each application and to map out space for furture innovation.

**System Design & Implementation**

**Architecture :**



**Modules & Functionalities used :**

**Login.java :**

SMS permission is checked first.

Check if login Database has any value. If it has a value\, it returns the number and if its empty, it returns ‘No Value’. If a number is returned, login is done using the returned number else the user is asked to enter the id and password.

If the user enters the id and password, a firebase reference of user is created and the database is arranged by the id i.e. by email id or the phone number. Then the password is checked, if the entered password is correct or not.

FirebaseReference=new Firebase("https://spark-08152425.firebaseio.com/users/");

If the register button is swiped, register activity is invoked.

**Register.java :**

The required fields values are taken from user. The entered information is checked if it exists in the firebase database. If it exists, then the next activity is not invoked. The entered values are transferred to the next activity if the values don’t exist.

FirebaseReference= new Firebase("https://spark-08152425.firebaseio.com/users/");

**RegisterUserPassword.java :**

The password is checked for the constraints. If it satisfies all the constraints, then the password and all the information is saved in the firebase. The data is pushed in the firebase to save it in the database.

FirebaseReference= new Firebase("https://spark-08152425.firebaseio.com/users/");

And the data is pushed to firebase.

**chatList.java :**

The user number entered while in the login activity is first saved in variable. A firebase reference is created to retrieve all the chats of the user.

FirebaseReference= new Firebase("https://spark-08152425.firebaseio.com/messages/");

By using this reference we get the nodes in the firebase. A check is done if the node contains user number. If not, next node is fetched else the node is split by “\_” into two numbers.

At arr[0] first number and at arr[1] the second number. If it is a group, the arr[0] will contain alphabet at its 0th position. If it is a group a card is created by using

list.add(new cardDetails(arrOfStr[0], R.drawable.group, "GROUP",name));

herearrOfStr[0] contains the group name.

If not, the thearr[0] is checked if it is equal to user number. If it is equal a card is created with

list.add(new cardDetails(arrOfStr[1], R.drawable.ic\_account\_circle\_black\_24dp,"SMS CHAT", arrOfStr[1]));

else a card is created with

list.add(new cardDetails(arrOfStr[0], R.drawable.ic\_account\_circle\_black\_24dp,"SMS CHAT", arrOfStr[1]));.

If the other user exists in database, the name is fetched using the firebase reference.

FirebaseReference= new Firebase("https://spark-08152425.firebaseio.com/users");

Then the card is created by replacing arrOfStr[0] with the name of other user in database.

To create a new, one to one chat the FAB button is clicked. It displays two more button. Single chat and group chat. In single chat the user is prompted to his contact activity and the number is selected from the contact list. In group button a new activity is opened. There the user needs to enter the name of group and select the participants of the group one by one from his contact list.

**chatWindow.java :**

theusernumber and the next user number is accepted from the chatList activity passed. A check is made if the other user number contains alphabet, then a group reference is created

FirebaseReference= new Firebase("https://spark-08152425.firebaseio.com/messages/"+otherNumber);

else a check is made if the users number is greater than the other user number and the reference is created as per the results.

FirebaseReference= new Firebase("https://spark-08152425.firebaseio.com/messages/" +Long.*toString*(firstNumber)+ "\_" + Long.*toString*(secondNumber));

A check is then made if the other user is registered in the app. If he is not registered a SMS will be sent to him. The user enters the message. It is accepted as following,

Map<String, String> map = new HashMap<String, String>();  
map.put("Message", messageText);  
map.put("User",userNumber);  
FirebaseReference.push().setValue(map);

This pushes the message in map format.

The previous messages are displayed using addchildeventlistener .if the message is from user

addMessageBox(message, 1,userName);

is called else

addMessageBox(message, 2,userName);

is called.

In function addmessagebox, all the constraints of the messagebox are set.

If the message contains "/reminder" a button is also inserted into scroll view below the message.

The message is displayed to right if it is from user else it is displayed to left.

A handler is used to check if the entered message starts with " / ". If yes, a reminder pops us. On clicking the reminder popup, message alert box pops up. On entering message, calendar pops up. Select date, then clock pops and on selecting the time a message is auto generated. It is then sent.

**Logout.java :**

The loginDatabase is deleted on clicking Logout navigation bar.

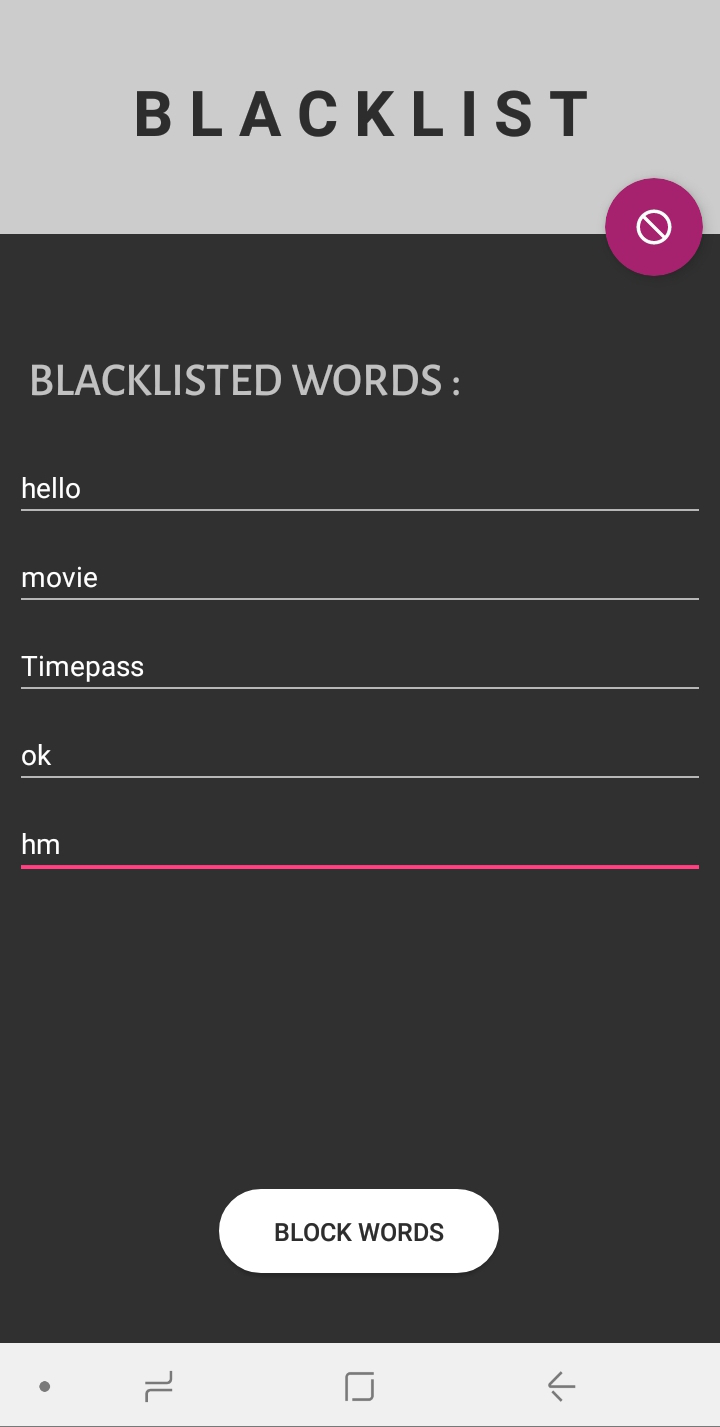
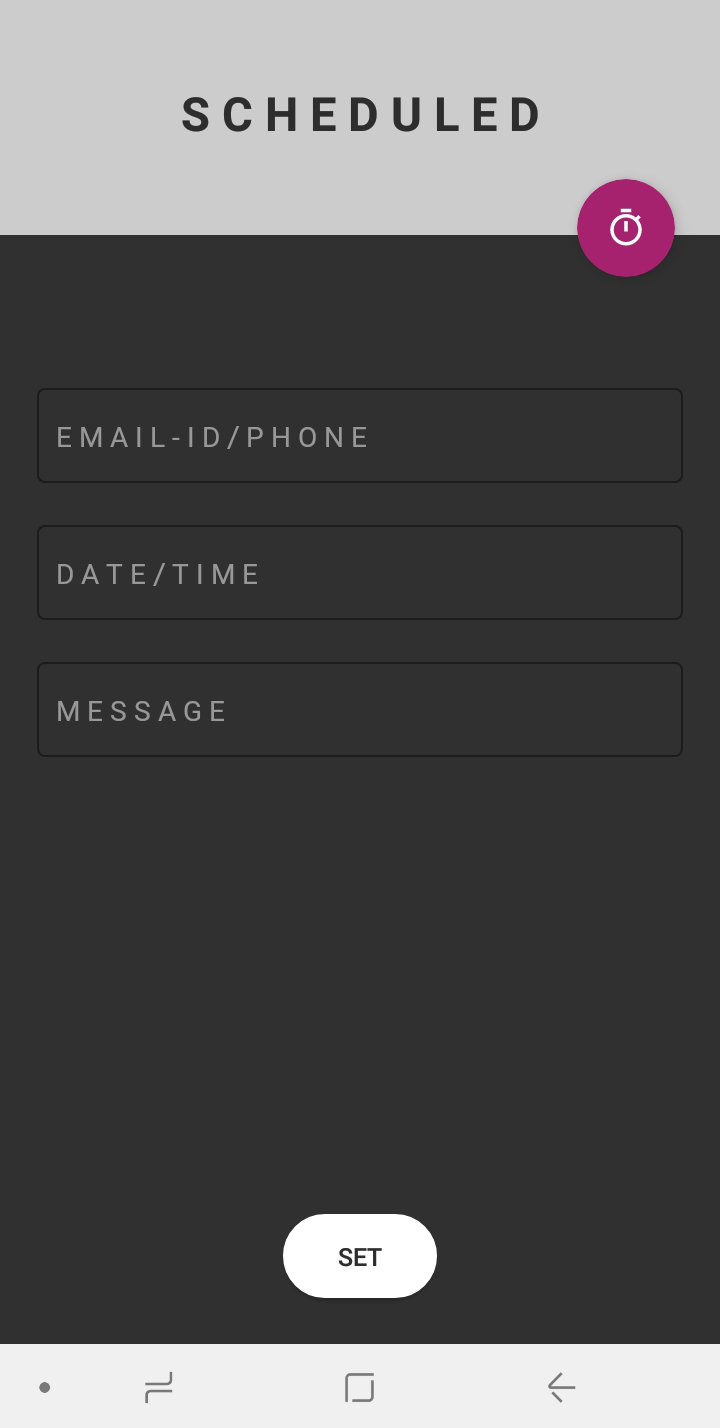
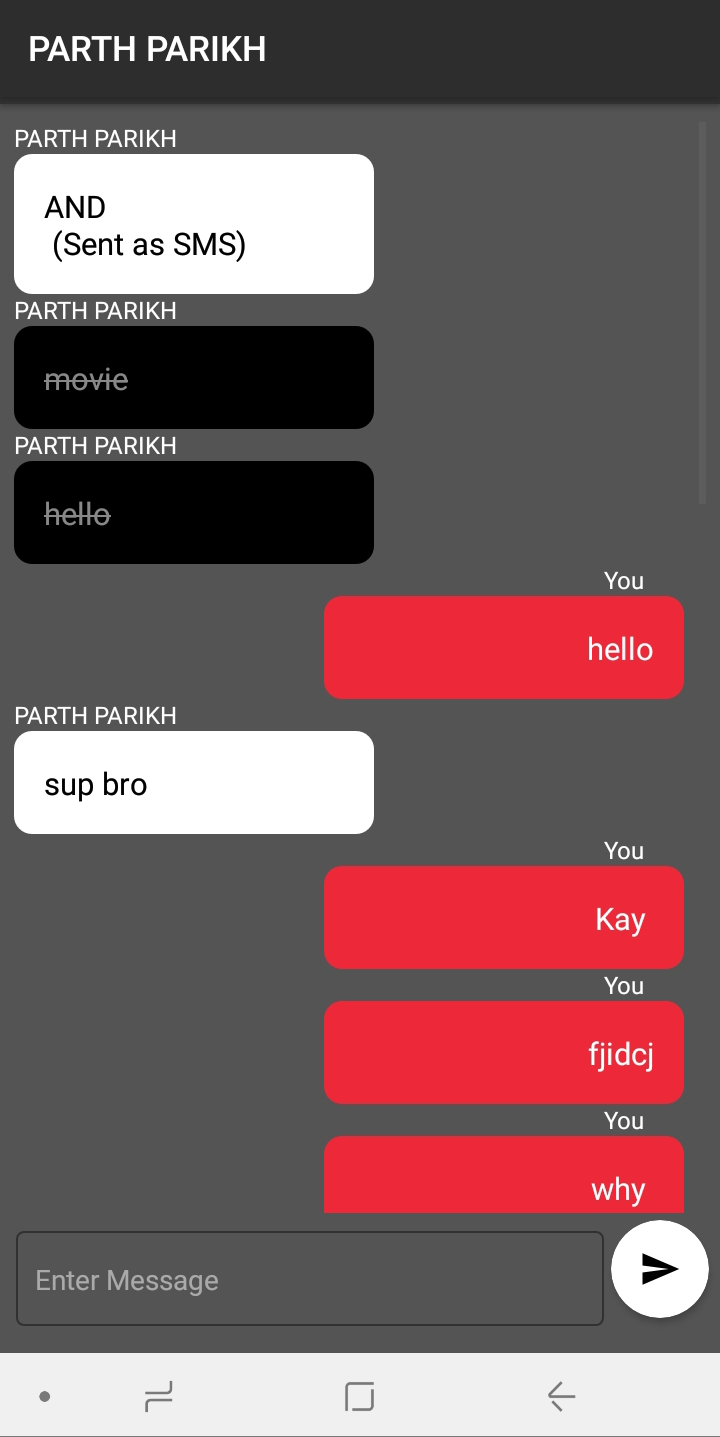
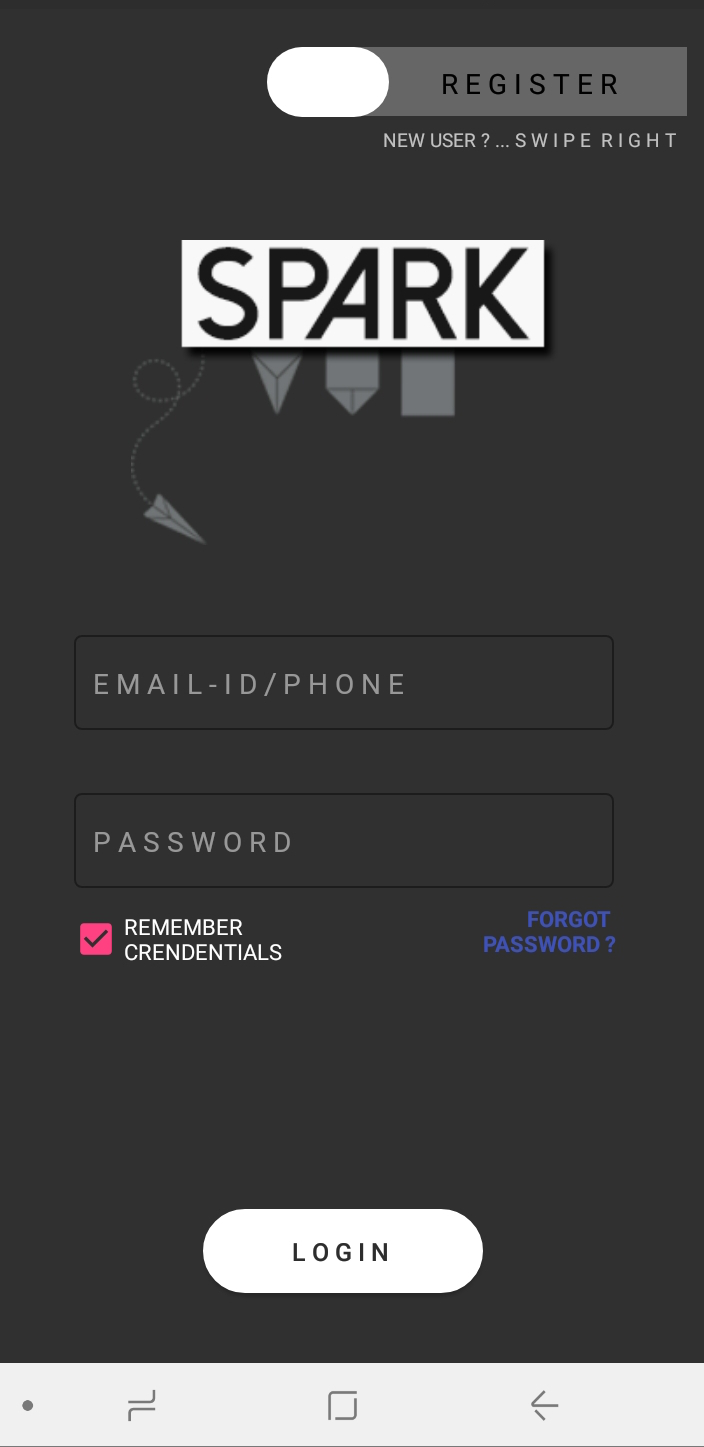
**System Requirements**

**Hardware Requirements :** Android Phone

**Software Requirements :** Android Studio, Firebase realtime database, SQLite, Android v5.0 and above.

**Result**

**GUI :**



**Conclusion**

The instant messenger SPArK is a feature rich means of communication with a user friendly.

**References**

* Android Developers
* Stack Overflow
* Firebase Website
* Udemy
* Quora
* Google Play console