SOFTWARE PROJECT FINAL REPORT

Group: 11

Team members

Abdul Rehman **2836088**

Varun Nagalla **2828837**

Sravan Kumar **2837055**

Tejam Reddy **2835430**

Date

05/12/2022

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Purpose and Scope

Secure chat app Android chat app, which is a chat application used for personal and office purposes. A friend or office colleague can share any kind of private data through this application. It will help to chat with friends and office colleagues. Mainly, it will help people who want to hide their conversation. This app allows users to chat securely and

create groups for chatting to keep privacy in the group. Privacy is the top priority of this application. Languages that have been used are Java and XML, developed in Android. Android studio is the tool that is used for this chat application development.

1.1.1 Goals and Objectives

The main goal of a secure chat app is a secure messaging app featuring end-to-end encryption as well as the ability to encrypt your messages so no one can read them. You can safely text a friend or family member your details like banking info, phone numbers, health information, and much more

- Sign in or Sign up using your phone number
- One to One chat
- Encrypt and decrypt messages
- Preventing screenshots during chatting
- Firebase Real-Time Chat Integration
- Group chat
- Status Updating Feature
- Beautiful Material Design
- Sweet and Clean animations
- Image Sharing

1.1.2 Project Scope

The secure chat app is an Android chat app, focused on individuals who have trouble following safely texting with loved ones and want to hide their conversation. It is developed in a Java and XML environment, The Main Features of the application.

- Dashboard
- Chat
- Profile
- Contacts
- Settings
- Create group
- Group chat
- Encrypt message
- Decrypt message

This application is based on privacy and secure chatting among friends, family, and office colleagues. Different kinds of functionalities have been implemented to make sure the high

ratio success of this chat app. Functions like encryption and decryption, Group chatting with security and easy use of the application are the important factors of this developed application.

1.2. Product Overview (including capabilities, scenarios for using the product, etc.)

Secure chap app is an application that provides security and full privacy to the users for chatting. An encryption/decryption algorithm is implemented. All the users can add any of their friends to the group. A user can also share a message with another one for sharing the message with full privacy and encryption using a key. The main functions of this application are chat, profile management, settings, creating groups, encryption of the messages, and decryption of the messages.

1.3. Structure of the Document

This document is structured as we gave an introduction and details of the project in the 1st chapter. Chapter 2 discuss the organization of the project. The plans and the model we used to develop this project. Chapter 3 provides details for the use cases and requirements of the project. Chapter 4 discuss the architecture of the project and the details of the project. Chapter 5 discuss the design and interface of the project. Chapter 6 is related to test cases and testing. Chapter 7 finally concludes this project.

1.4. Terms, Acronyms, and Abbreviations

UI……User interface

UX……User experience

SRS……Software requirement specification

SDS……Software design specification

SPP……Software project report

STS……Software test specification

R1……Requirement 1

C1……Component 1

TC1……...Test case 1

UC……Use case

OS……..Operating system

RA……..…Risk analysis

OO……..…Object oriented

IDE……..……Integrated development environment

SDK……..……Software development kit

API……Application programming interface

CPA……Cross-platform app

APK…….……Android application package

GUI……..Graphical user interface

2. Project Management Plan

2.1. Project Organization

We used GitHub for the work of our project version control. We have developed a plan and followed the steps already defined by the team:

- SRS
- SDS
- STS
- SPR

During the development of the project, we followed these documents to complete and document our details.

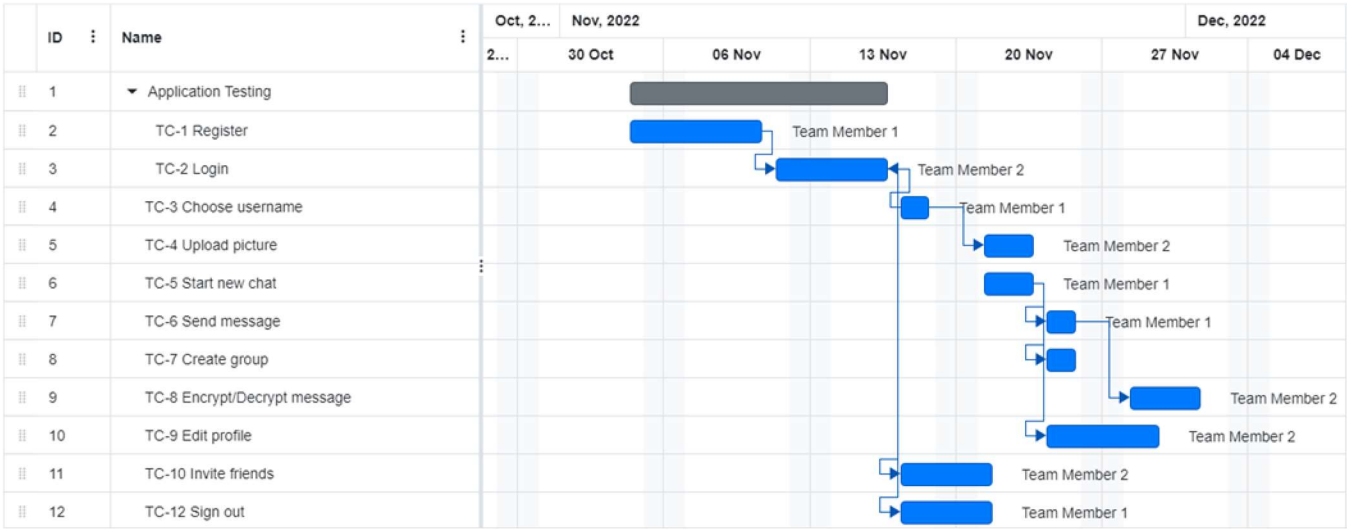
Figure 1: Schedule

2.2. Lifecycle Model Used

There are multiple development models are available for the development of the project. These models provide situation-based criteria for the development of the project.

We used the waterfall model as it is the project we have to present in the class. In the future, we will use an iterative model as we will add one-by-one functions to this application.

2.3. Risk Analysis



Restrictions

- Unauthorized users do not use this system.
- To use this system users will be registered first

Limitations

- Android based application
- Only firebase-implemented specifications are usable
- Do not support cross-platform working

Constraints

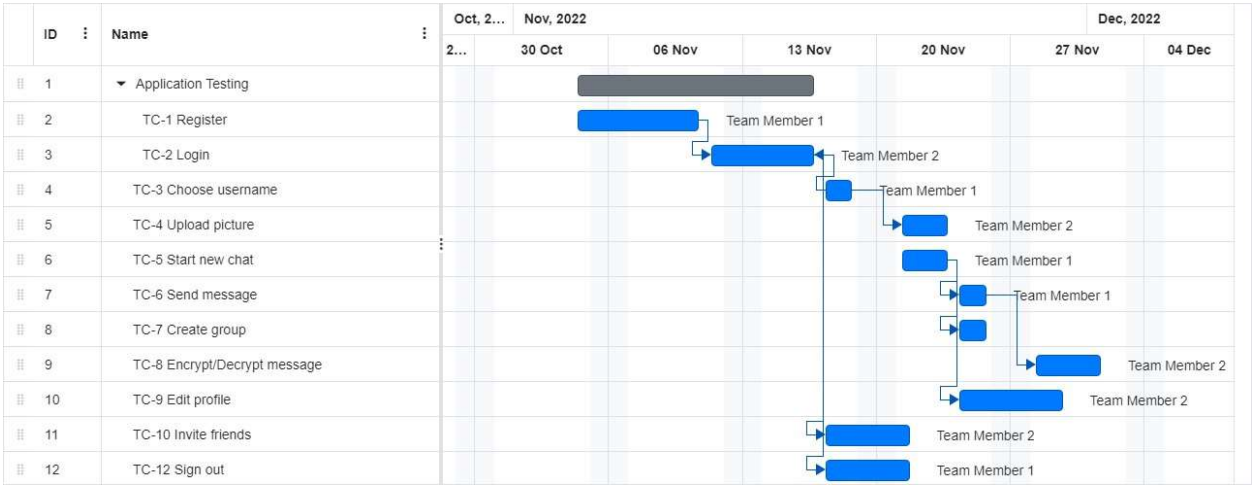
- All database maintenance will be handled by the users and it is available when the user needs it.
- Data is secure enough that no user can see the personal information of the other user.
- Data must be restored in any case of loss of data.
- The Internet should be available to use this system.
- All the records of the User are stored in the Database.

2.4. Hardware and Software Resource Requirements

For the development of this project, we required an android studio and an android device supporting OS level after android OS version 5.0, Lollipop.

2.5. Deliverables and schedule

- SRS
- SDS
- STS
- SPR



3. Requirement Specifications

3.1. Stakeholders for the system

3.1.1 External stakeholders

- Users who install this app
- Organizations that use this app for communication

3.1.2 Internal stakeholders

- Developers
- Team members for developing project
- Supervisor

3.2. Use cases

This section provides a usage scenario for this project. It organized information collected during requirements elicitation into use cases. Use cases are the description of how users will perform tasks. The following are our systems use cases:

3.2.1 Graphic use case model

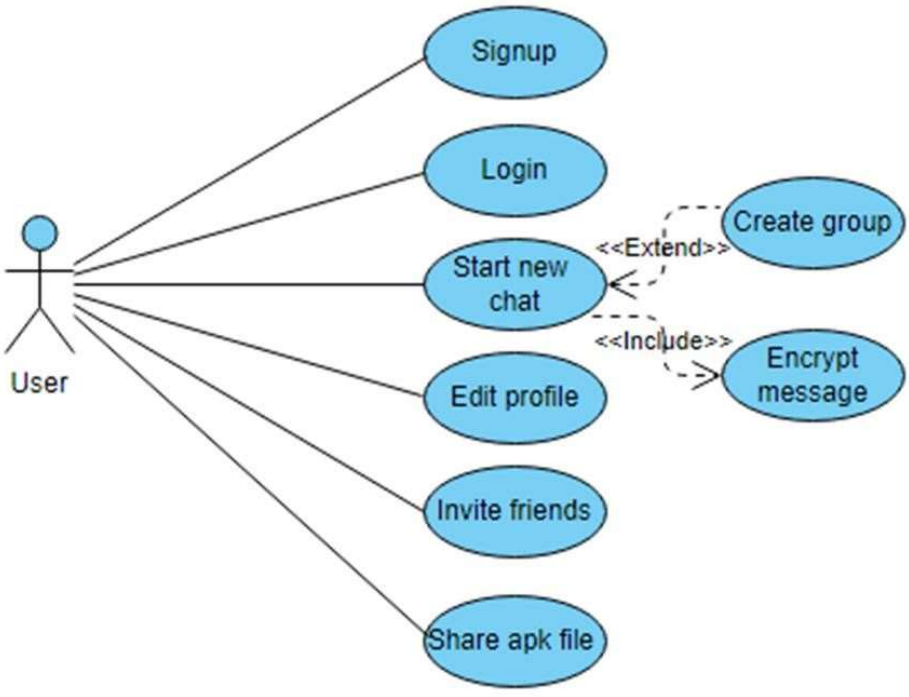


Figure 2: Use case diagram

3.2.2 Textual Description for each use case

3.2.2.1. User signup for the application by entering details.

- Users can sign up using the application.
- Validation made during sign-up:
    - Users must enter their phone number and verification from the system (Only numbers are allowed)
    - The phone number for account creation(numbers)
    - The phone number will be validated.
    - Phone number fields must be filled.

3.2.2.2. User should sign in by entering credentials.

- Users can log in to the app using login activity.
- Validation made during login:
    - Phone numbers should be present in the database. o Users must enter an OTP match from the system.
    - OTP can verify automatically also.

3.2.2.3 User must be able to send and receive messages.

- The user needs to log in to access the message activity.
- A message can be sent:
    - The receiver's number should be in the contact list. o Users can click on the receiver's username to start a chat.
    - Users can send a message:
        - By clicking the camera icon gallery will open.
        - Users can select any picture or video (up to 16MB).
        - Users can click send button to send pictures or videos.

- Users can send type messages in the text field.
- Users can click send button to send text messages.
- The text field cannot be empty to send the message.

3.2.2.4. Encryption of message should be available.

- Users can encrypt the message by clicking on the encryption button.
- Users can send an encrypted message:
    - The message must be entered in the text field. o Users can encrypt the message by clicking encrypt button.

3.2.2.5 Decryption of the message at the receiver end.

- The user (receiver) can decrypt the message by entering the password given by the sender:
    - The message must be received with encryption from the sender side.
    - Message can be unlocked by entering a given matching password by the sender.
    - Password should be available in the database.
    - The password field cannot be blank.

3.2.2.6. Group chat should be available.

- Users can create groups for group chat:
    - Users can click on the create group button to add other users to the group.
        o The user should be in the contact list of the admin.
    - On creating a group admin can add the user from the contact list.
    - Users can send or receive messages in the group.

- Encryption and decryption can be implemented using the password.

3.2.2.7 The System must be reliable so that database should not go down and can handle it effectively.

- System reliability depends on backups in case of a down server:
  - Firebase by Google will be used for this application to make it reliable.

3.3. Rationale for your use case model

To view in detail and valid description of the project with the help of the use case model. Defining the functions to get details and the big picture that how we are going to develop the project.

3.4. Non-functional requirements

This system should be available 24 hours and provide services to the users when the user wants to access the system.

- Availability:
  - Google play store platform will be used to make sure 24/7 availability.

    Our product will be easy to use by anyone because its interface is not complex.

- The app should be easy to use:

- Iconic guide for the user to use simply.
- General color scheme to make the application more usable.

The system should give a quick response to user actions.

- Performance:

  Firebase real-time database will be used to make sure quick responses.

- Reliability:

  Data must be restored in any case of loss of data.

- Security:

  Data is secure enough that no user can see the personal information of the other user.

4. Architecture

4.1. Architectural style(s) used

Software architecture pattern plays a crucial role in its ability to scale and meet users' demands over time.

An architectural pattern can be called an outline that allows you to express and define a structural schema for all kinds of software systems. It's a reusable solution that provides a predefined set of subsystems, roles, and responsibilities, including the rules and roadmap for defining relationships among them. It helps you address various software engineering concerns such as performance limitations, high availability, minimizing business risk, etc.

For this project, we have used a **Client-Server Architecture** style. We developed an app that supports real-time support to the users. For this type of project, client-server architecture is the best to use.

4.2. Architectural model (includes components and their interactions)

A list of the software architecture components is presented.

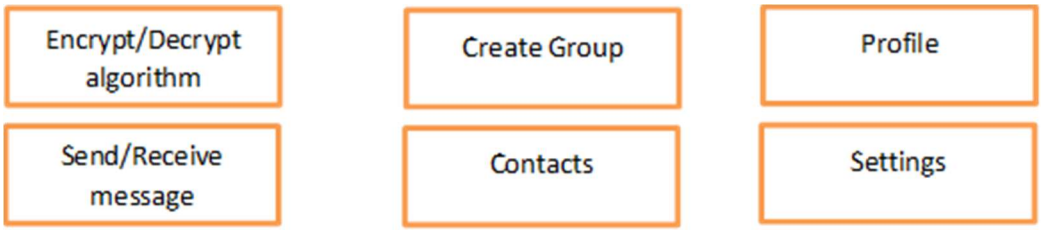


Figure 3: Architecture Components

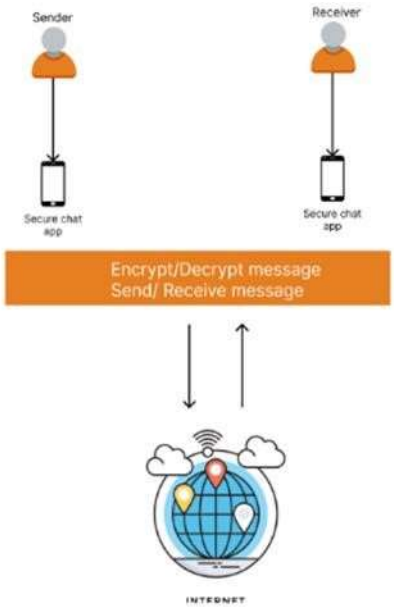


Figure 4: Architecture diagram

External Interface Description

No external design is needed. Only application designs are implementable and shared.

4.3. Technology, software, and hardware used

For the development of these diagrams, MS word built-in tools were used. For different diagrams, the UMLet tool is used.

4.4. Rationale for your architectural style and model

For this project, we have used a **Client-Server Architecture** style. We developed an app that supports real-time support to the users. For this type of project, client-server architecture is the best to use.

As mentioned in the style's description this style is best for chat applications.

5. Design

5.1. User Interface design

User interfaces are shared:

Figure 5: UI

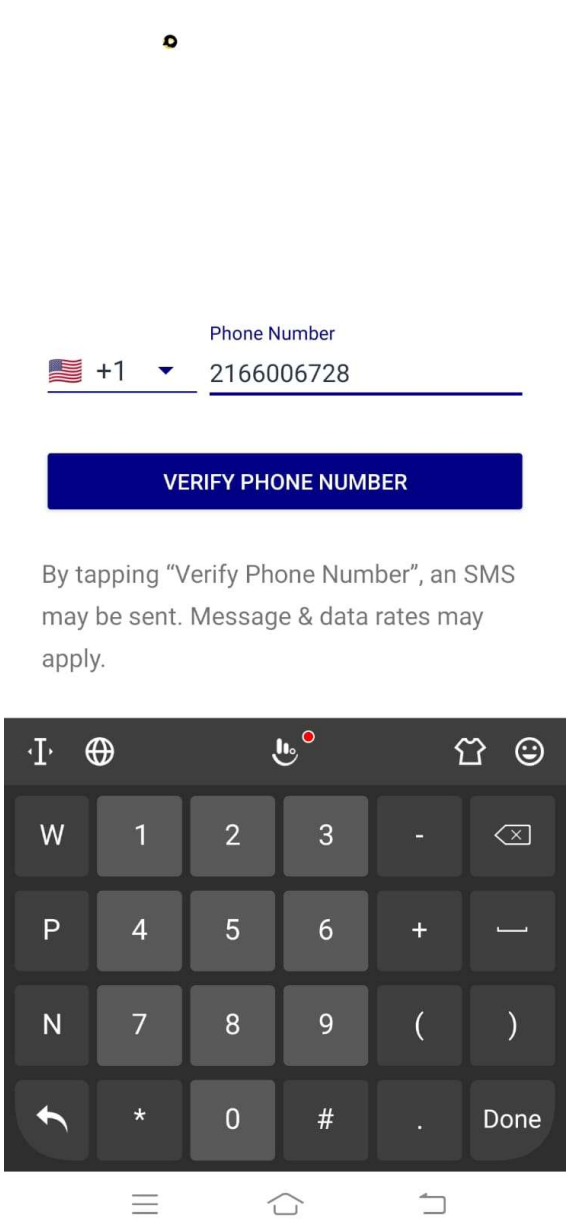
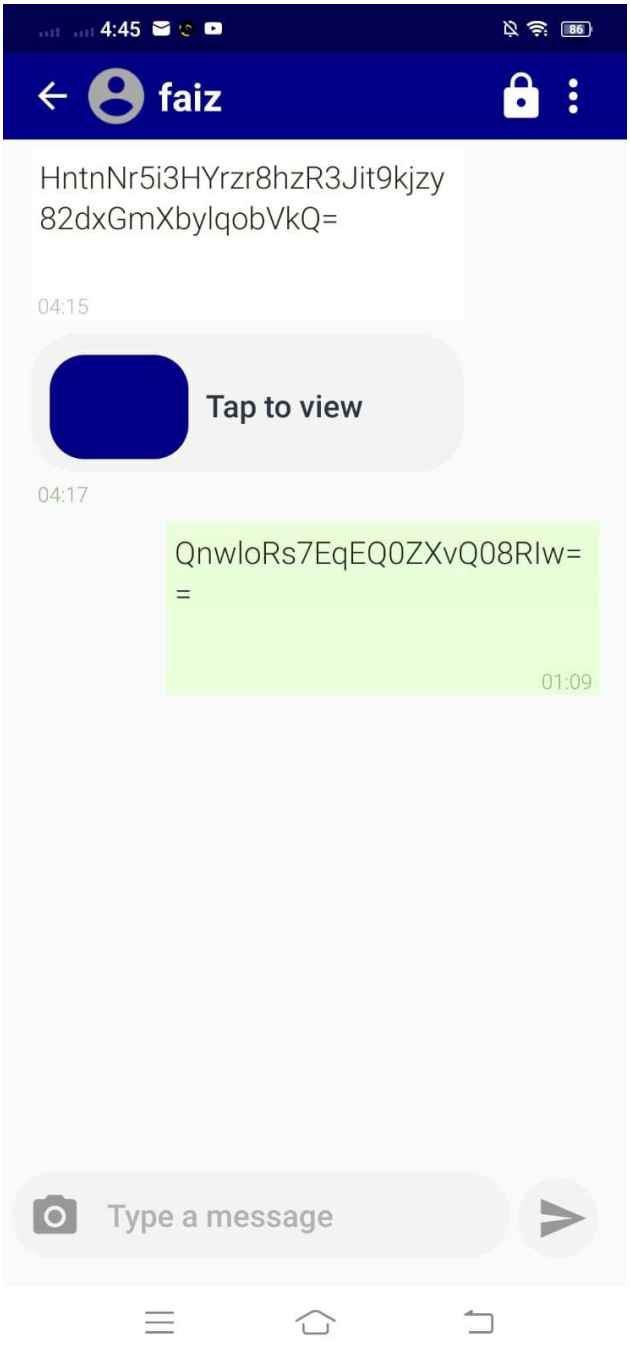
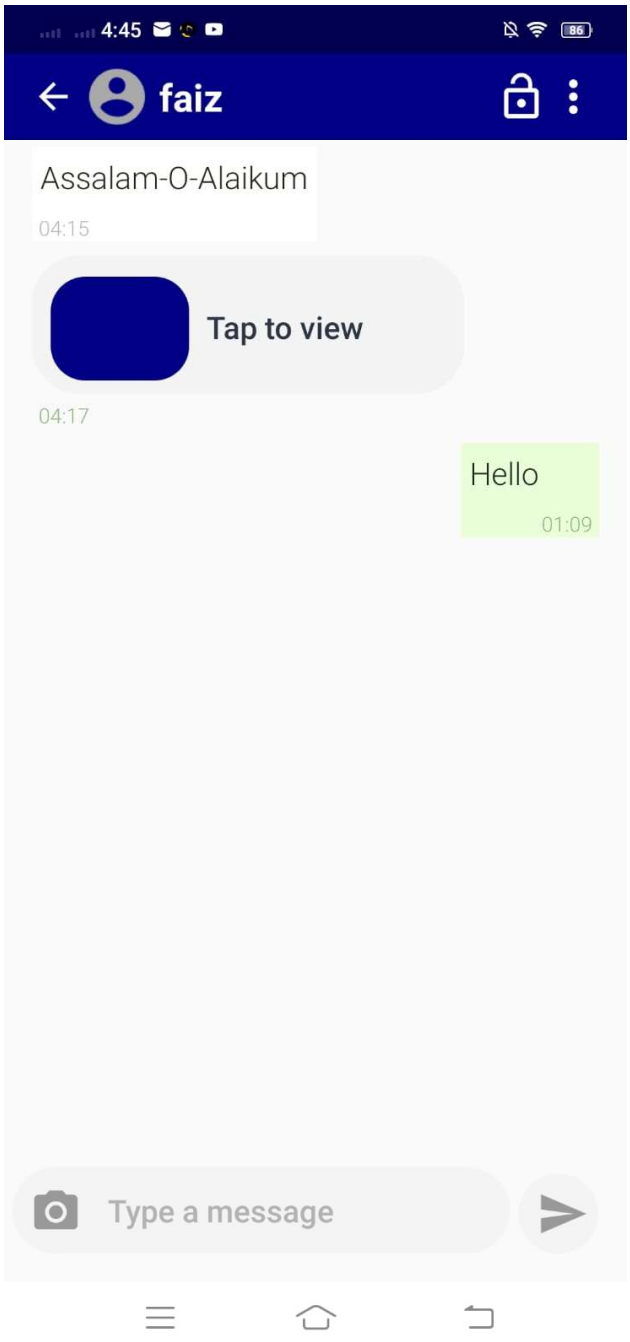
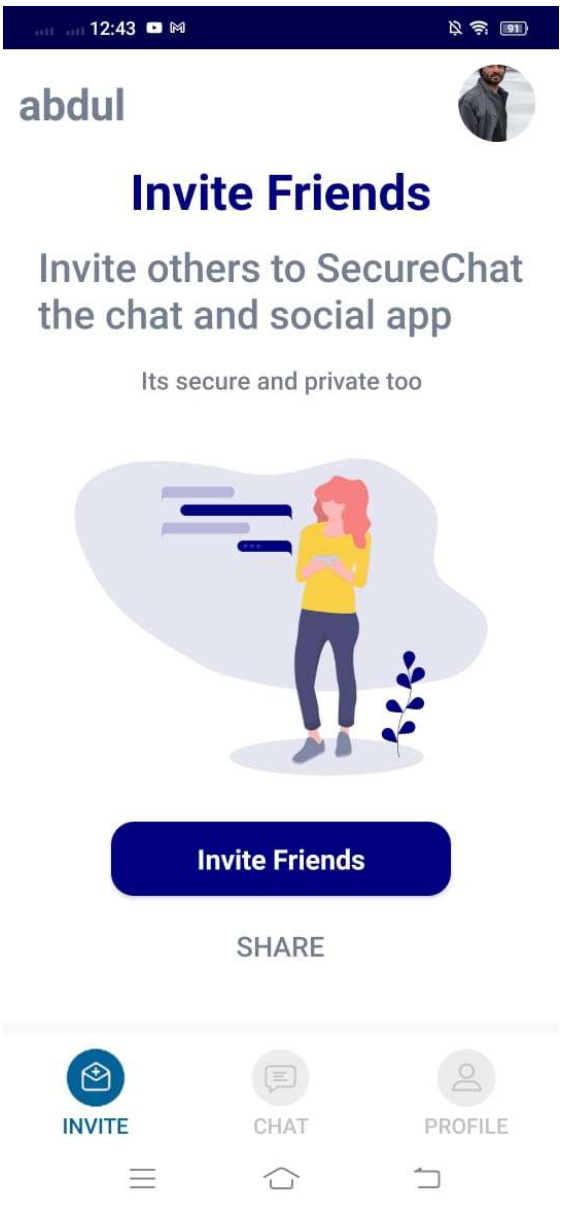
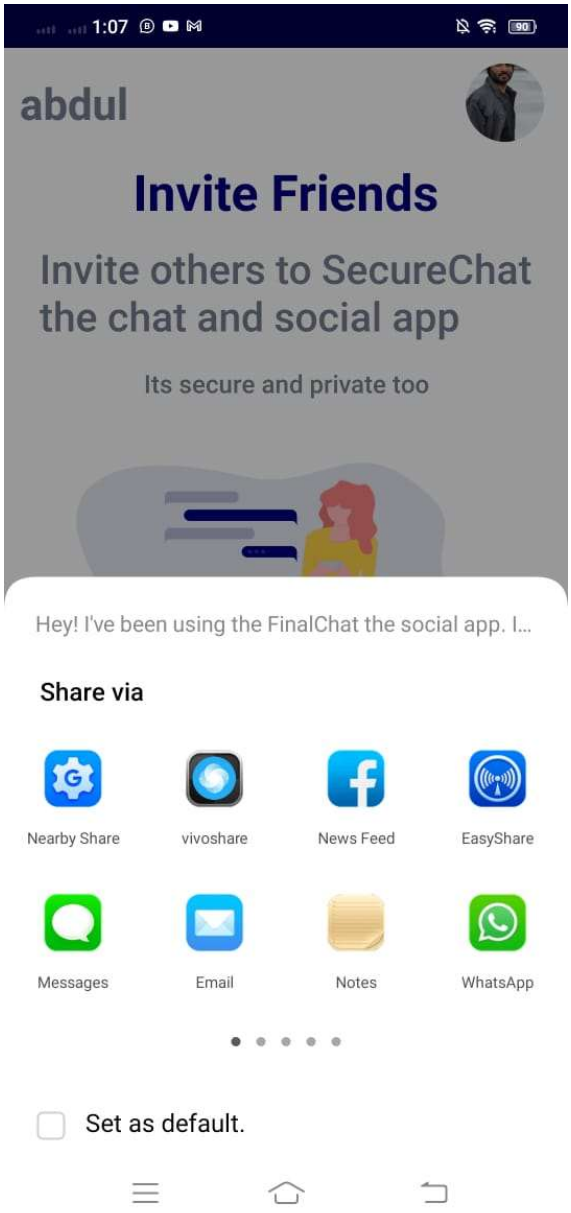


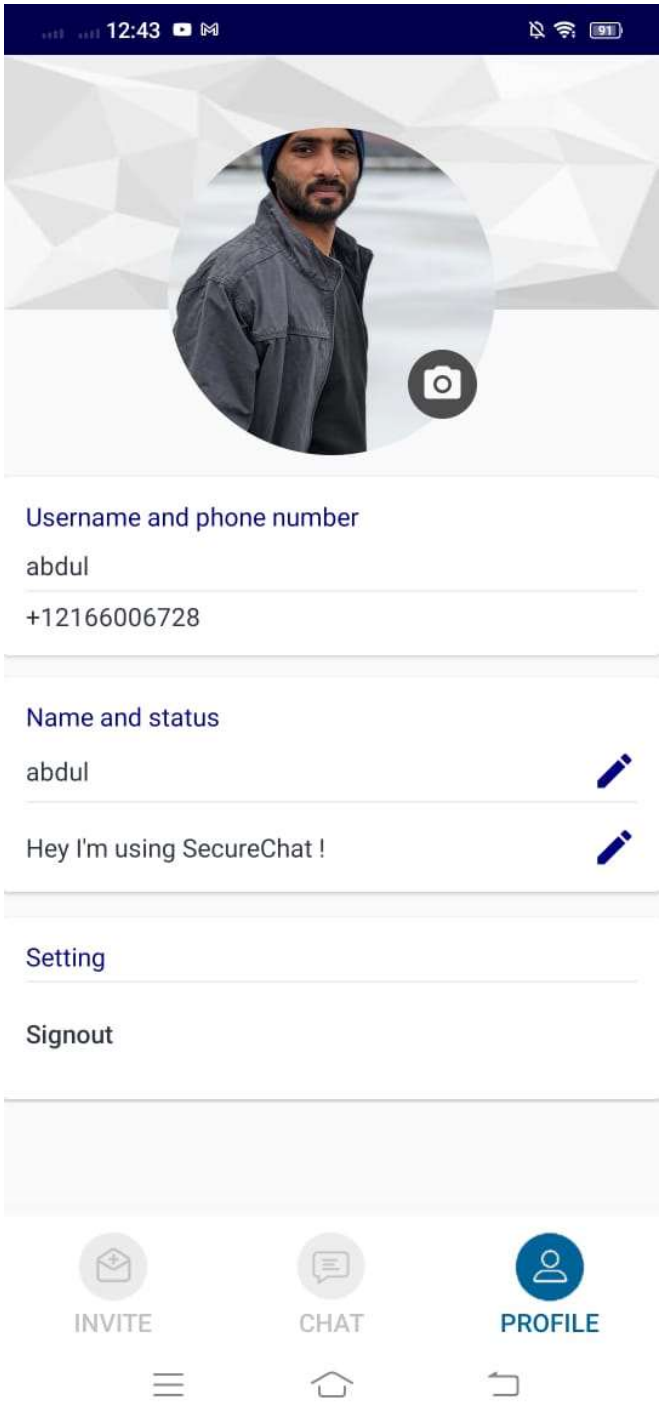
Figure 6:UI

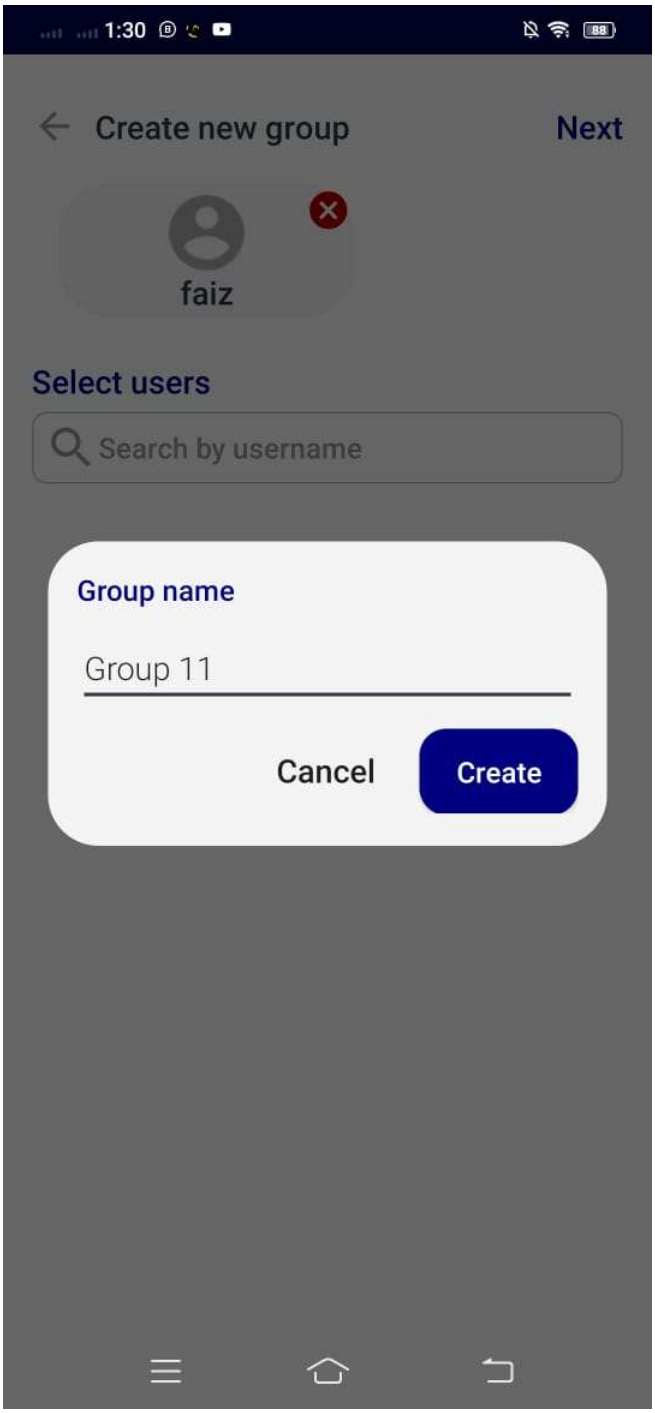

4:45
faiz
HntnNr5i3HYrzr8hzR3Jit9kjzy
82dxGmXbylqobVkQ=
04:15
Tap to view
04:17
QnwloRs7EqEQ0ZXvQ08RIw=
=
01:09
Type a message


4:45
faiz
Assalam-O-Alaikum
04:15
Tap to view
04:17
Hello
01:09
Type a message


12:43
abdul
Invite Friends
Invite others to SecureChat the chat and social app
Its secure and private too
Invite Friends
SHARE
INVITE
CHAT
PROFILE


1:07
abdul
Invite Friends
Invite others to SecureChat the chat and social app
Its secure and private too
Hey! I've been using the FinalChat the social app. I...
Share via
Nearby Share
vivoshare
News Feed
EasyShare
Messages
Email
Notes
WhatsApp
Set as default.


12:43
Username and phone number
abdul
+12166006728
Name and status
abdul
Hey I'm using SecureChat !
Setting
Signout
INVITE
CHAT
PROFILE


1:30
Create new group
Next
faiz
Select users
Search by username
Group name
Group 11
Cancel
Create

5.2. Components design (static and dynamic models of each component)

A description of major software components contained within the architecture is presented.

Interface description

Input

Button click only required. No special input for the encryption of the message. But for the decryption user must enter the key to decrypt the message.

Output

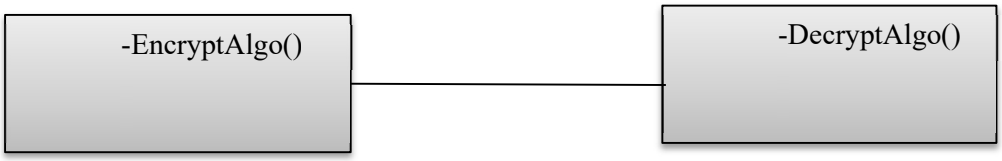
The user will send an encrypted message and will be able to decrypt the received message after entering the key.

Exceptions

The message is encrypted or decrypted.

Static models

Class diagrams



composite structure diagram

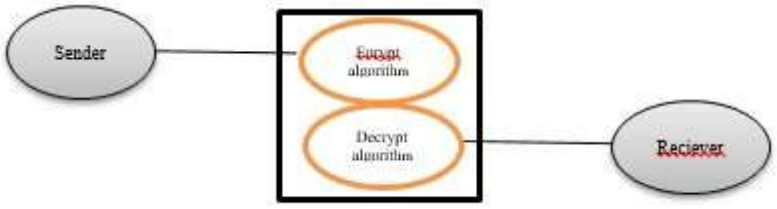
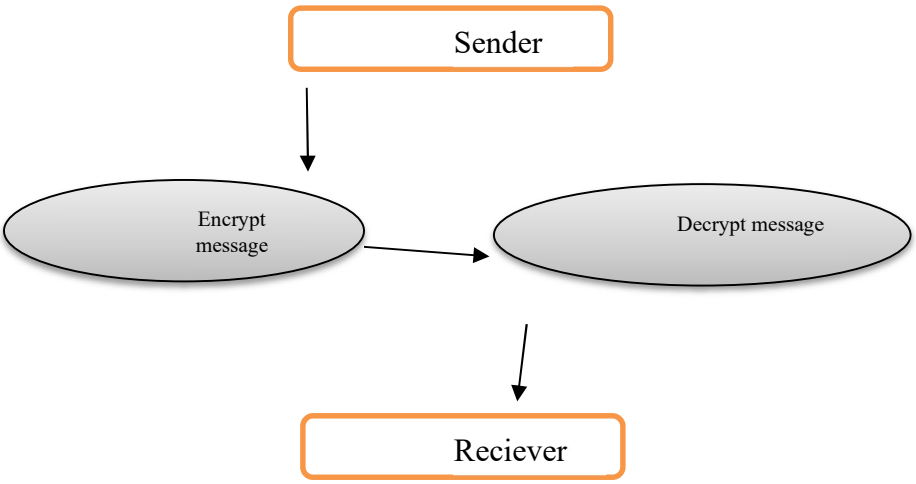


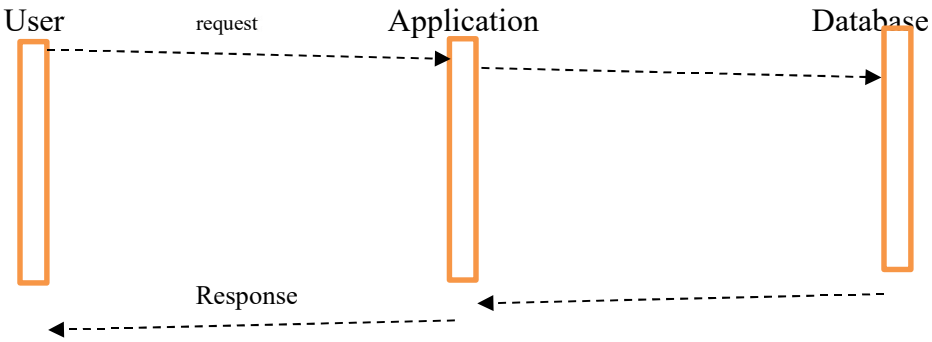
Figure 9: component 1 CSD

Dynamic models

Activity diagrams



sequential diagrams



Interface description
Input

Button click only required. No special input for the encryption of the message.

But for the decryption user has to enter the key to decrypt the message.

Output

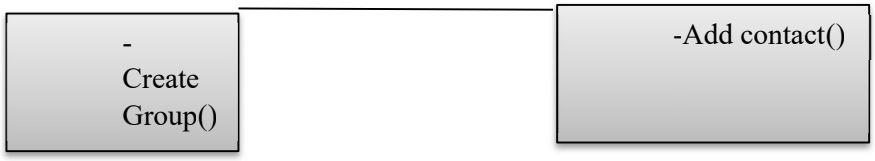
The user will send an encrypted message and will be able to decrypt the received message after entering the key.

Exceptions

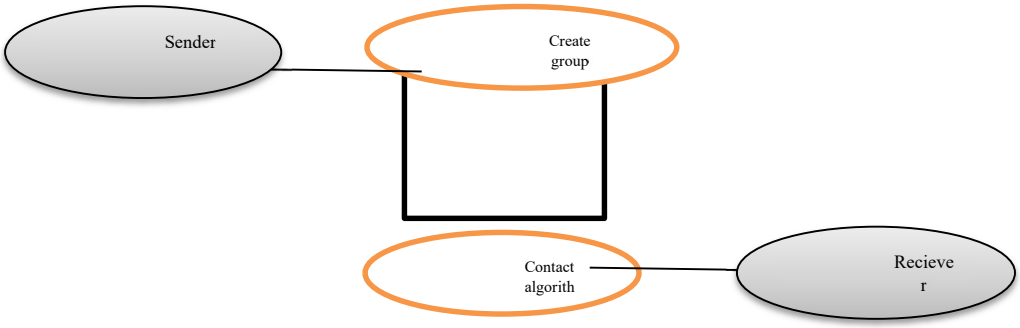
The message is encrypted or decrypted.

Static models

Class diagrams

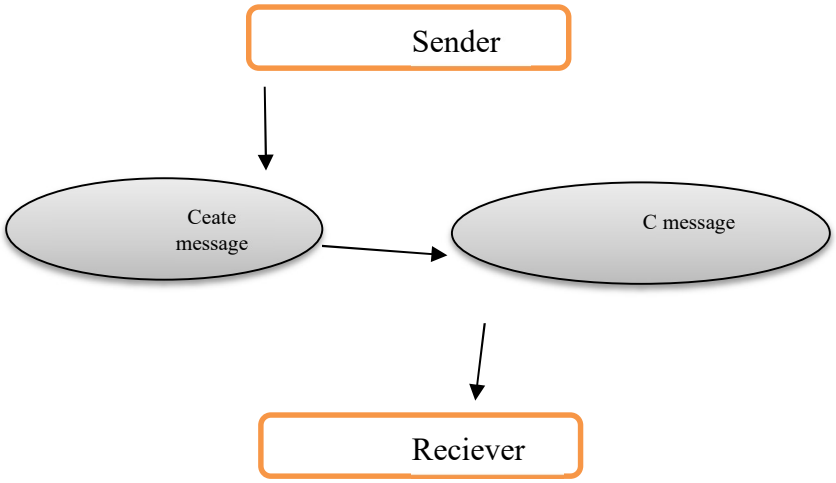


composite structure diagram

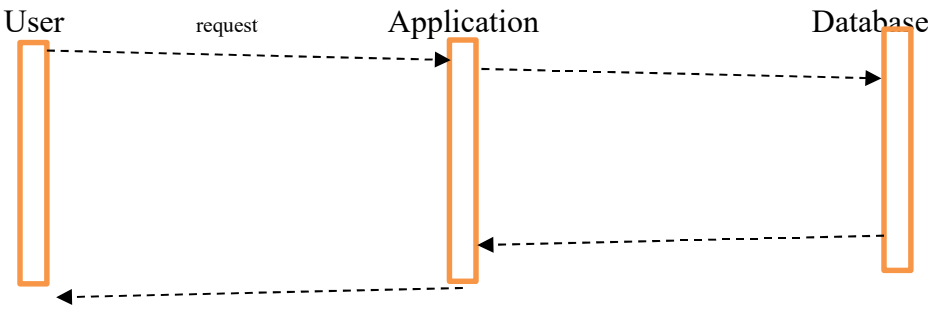


Dynamic models

Activity diagrams



sequential diagrams



Interface description

Input

Button click only required. No special input for the encryption of the message. But for the decryption user must enter the key to decrypt the message.

Output

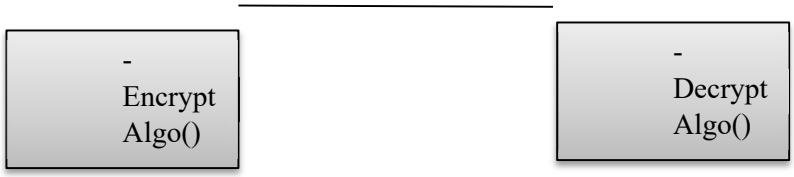
The user will send an encrypted message and will be able to decrypt the received message after entering the key.

Exceptions

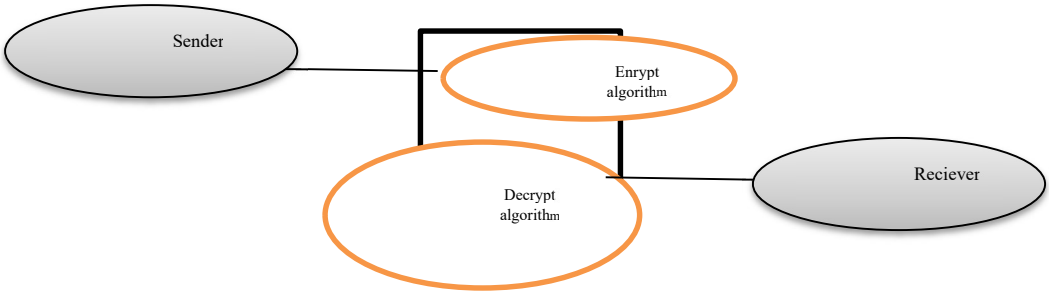
The message is encrypted or decrypted.

Static models

Class diagrams

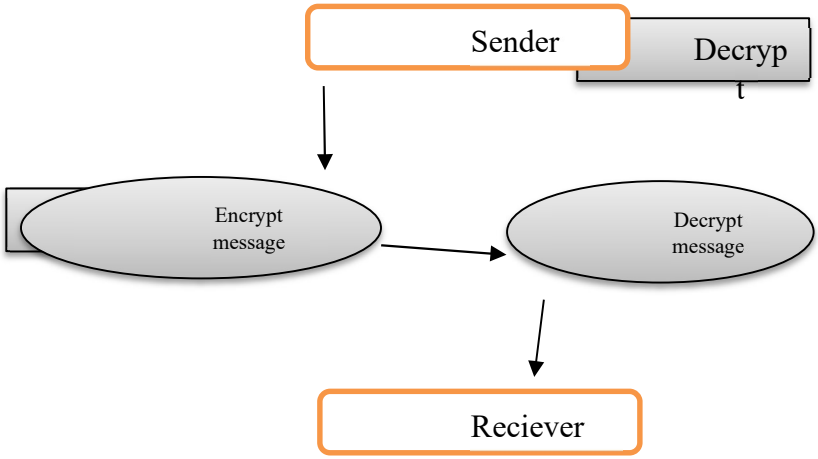


composite structure diagram

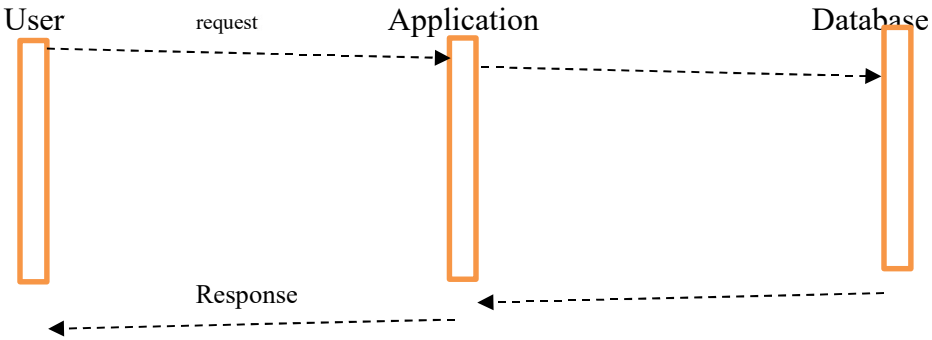


Dynamic models

Activity diagrams



sequential diagrams



5.3. Database design

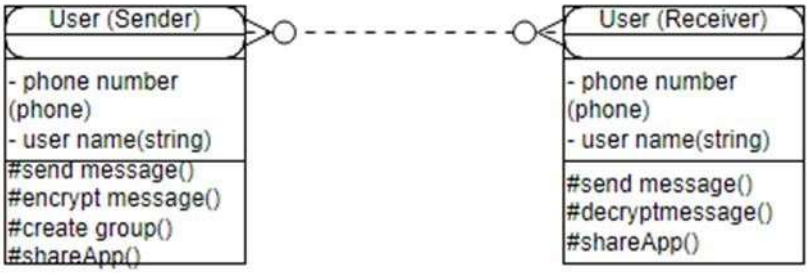


Figure 10: Firebase-based ERD

5.4. Rationale for your detailed design models

This design model is used for the verification, evaluation, maintenance, reuse, teaching, communication, and documentation of the project.

5.5. Traceability from requirements to detailed design models

C1 => R1, C2=> R2, C3=> R3, C4=> R4, C4=> R5, C4=> R6, C5=> R7

6. Test Management

6.1. A complete list of system test cases

This section enumerates a complete list of test cases for the software.

Table 1: TC 1

ID	**TC-1 Register**
Test Input	Phone number
Expected Output	A verification message should be received
Description	To register on the app phone number will be entered and a verification from firebase will be sent to ensure security.

Table 2: TC 2

ID	**TC-2 Login**
Test Input	Phone number
Expected Output	A verification message should be received
Description	To log in to the app phone number will be entered and verification from firebase will be sent to ensure security.

Table 3: TC 3

ID	**TC-3 Choose username**
Test Input	Enter the string for the username
Expected Output	The username should be assigned
Description	To choose a username the app string will be entered, and data will be saved into firebase.

Table 4: TC 4

ID	**TC-4 Upload picture**
Test Input	Click on the button to upload a profile picture
Expected Output	The picture should be uploaded
Description	To upload a user profile picture gallery should be appeared for choosing pictures.

Table 5: TC 5

ID	**TC-5 Start a new chat**
Test Input	Click on the button and select contact
Expected Output	Contacts should be appeared to start the chat
Description	To start chatting on the app clicking on the button will lead to new activity where a contact needed to be selected and starts a chat.

Table 6: TC 6

ID	**TC-6 Send message**
Test Input	Click on the button, select contact, enter a character send a message
Expected Output	The user should be able to send a message.
Description	To send a message from the app clicking on the button will lead to sending a message.

Table 7: TC 7

Test Input	Click on the button and select contact
Expected Output	Contacts should be appeared to start the chat
Description	To create a group on the app clicking on the button will lead to new activity where a contact needed to be selected and created for the group.

8: TC 8

ID	**TC-8 Encrypt/Decrypt message**
Test Input	Click on the button
Expected Output	The message should be encrypted/decrypt with a key.
Description	To encrypt/decrypt on the app clicking on the button will lead to a function where the contact needs a key to encrypt/decrypt messages.

Table 9: TC 9

ID	**TC-9 Edit profile**
Test Input	Click on the button, enter a new username, and Enter new status.
Expected Output	The username or status should be updated.
Description	To update the profile on the app clicking on the button will lead to a function where the app needs input to update the status.

Table 10: TC 10

ID	**TC-10 Invite friends**
Test Input	Click on the button.
Expected Output	Contacts should be appeared to share links.
Description	To share the app, clicking on the button will lead to a layout where the app link can be shared.

11: TC 11

ID	**TC-11 Share .apk**
Test Input	Click on the button.
Expected Output	Contacts should appear in the .apk file.
Description	To share the .apk file, clicking on the button will lead to a layout where .apk can be shared.

Table 12: TC 12

ID	**TC-12 Sign out**
Test Input	Click on the button
Expected Output	The user should sign out and the login layout should appear.
Description	To sign out from the app clicking on the button will lead to login activity.

6.2. Traceability of test cases to use cases

TC1 => signup

TC2 => login

TC3 => edit profile

TC4 => edit profile

TC5 => start new chat

TC6 => send message

TC7 => send message

TC8 => Encrypt/decrypt

TC9 => edit profile

TC10 => invite friends

TC11 => share .apk

TC12 => signout

6.3. Techniques used for test case generation

6.3.1 Decision Table-Based Testing

A decision table is also known as to Cause-Effect table. This software testing technique is used for functions that respond to a combination of inputs or events. For example, a submit button should be enabled if the user has entered all required fields.

We used this technique for test case generation.

6.4. Test results and assessments (how good are your test cases? How good is your software?)

We tested our application and test cases show that our software is fit for use. Requirements are implemented correctly. All the test cases show that our application is providing all the features implemented correctly.

6.5. Defects reports

Our developed application is working perfectly as implemented and intended.

7. Conclusions

Secure chap app is an application that provides security and full privacy to the users for chatting. An encryption/decryption algorithm is implemented. All the users can add any of their friends to the group. A user can also share a message with another one for sharing the message with full privacy and encryption using a key. The main functions of this application are chat, profile management, settings, creating groups, encryption of the messages, and decryption of the messages.

7.1. Outcomes of the project (are all goals achieved?)

All the goals were settled at the start of this project, and we have achieved all the goals driven out from the requirements. The idea was purposed and after that, we wrote all the requirements. Step-by-step working of our team leads to success. This project provides a new way to chat with users securely. Users can share their ideas with full privacy. The main functions of this project were:

- Dashboard
- Chat
- Profile
- Contacts
- Settings
- Create group
- Group chat
- Encrypt message

- Decrypt message

All the functions have been implemented as we can show in the demo.

7.2. Lessons learned

This was the first project we were doing as a team. We learned teamwork. Through the step-by-step development of this project, we came to know the depth of time-saving and cost-effective project development. Developing the right product is a challenge for any developer or organization providing services. But following this process not only saves time but also makes it possible to understand and meet requirements. Developing and documenting the project step by step makes it easy to maintain the project.

7.3. Future development

We found this project very useful and a need for these types of apps is required. We are planning to extend its functions like audio/video calls, status, or story sharing, and sharing posts. We have the plan to make it a social platform in the future.

References

https://web.cs.wpi.edu/Research/aidg/DR-Rpt98.html

https://www.jot.fm/issues/issue_2005_11/article4/

https://www.chsbuffalo.org/sites/default/files/files/for-physicians/secure-chat-quick-startguide-101620.pdf https://developer.android.com/studio https://www.umlet.com/

https://www.oracle.com/java/technologies/downloads/

https://www.tutorialspoint.com/software_testing_dictionary/software_requirement_specification.htm https://www.toolsqa.com/software-testing/test-casespecification/#:~:text=Test%20Case%20Specification%20document%20described,forth %2C%20for%20a%20given%20feature.

http://www.cs.iit.edu/~oaldawud/Slides/Class12_ConfigurationManagement.pdf

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

https://www.softwaretestinghelp.com/how-to-test-software-requirements-specificationsrs/

https://www.synopsys.com/glossary/what-is-sdlc.html

https://www.techtarget.com/searchsoftwarequality/definition/systems-development-lifecycle https://www.indeed.com/career-advice/career-development/project-organization

https://cs.uwaterloo.ca/~m2nagapp/courses/CS446/1195/Arch_Design_Activity/ClientSer ver.pdf https://www.guru99.com/software-testing-techniques.html