**MINI PROJECT**

**(2020-22)**

**‘WhatsApp Clone’**

**Project Report**

**Institute of Engineering & Technology**

**Submitted By-**

Vishnu Pratap Singh Chauhan (201500803)

Shruti (201500676)

Rohit Kumar (201500586)

**Under the Mentorship Of-**

Mr. Bhanu Kapoor

Technical Trainer

Department of Computer Engineering & Applications

**Department of Computer Engineering and Applications**

**Declaration**

I hereby declare that the work which is being presented in the Bachelor of technology. Project ‘WhatsApp Clone’, in partial fulfilment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the Mentorship of Mr. Bhanu Kapoor, Mentor, Dept. of CEA, GLA University.

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

**Sign:**

**Name of Candidates:** Vishnu Pratap Singh Chauhan

Shruti

Rohit Kumar

**Department of Computer Engineering and Applications**

**Certificate**

This is to certify that the project entitled ‘WhatsApp Clone’, carried out in Mini Project - I Lab, is a Bonafede work by Vishnu Pratap Singh Chauhan, Rohit Kumar and Shruti and is submitted in partial fulfilment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Mentor:

Name of Mentor: Mr. Bhanu Kapoor

Date:

**Department of Computer Engineering and Applications**

**ACKNOWLEDGEMENT**

Presenting the ascribed project paper report in this very simple and official

form, we would like to place my deep gratitude to GLA University for providing

us the mentor Mr. Bhanu Kapoor, our Mentor.

He has been helping me since Day 1 in this project. He provided me with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meetings to check the progress of the project and providing us with the resources related to the project. Without his help, I wouldn't have been able to complete this project.

And at last but not the least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

Sign:

Name of Candidate:

University Roll No.:

**ABSTRACT**

Our project is to clone very famous social media app “WhatsApp”

WhatsApp is a free cross-platform messaging service. It lets users of iPhone and

Android smartphones and Mac and Windows PC call and exchange text, photo,

audio and video messages with others across the globe for free, regardless of the

recipient's device.

Our Aim is to cover all Main features of chatting App like Phone Number

Authentication, One to one & Group chatting with contacts only - includes text,

image, gif, video, audio (with recording), emoji sharing and image and video

caching, status visible to contacts only and disappears after 24 hours, video

calling (one-one and group), online/offline status, Message Seen Feature,

Automatic Scrolling on New Message, Replying to Messages and much more!

**INDEX**

|  |  |  |
| --- | --- | --- |
| **CONTENTS** | **PAGE NUMBERS** | **SIGNATURES** |
| Cover page | 1 |  |
| Declaration | 2 |  |
| Certificate | 3 |  |
| Acknowledgment | 4 |  |
| Abstract | 5 |  |
| Index | 6 |  |
| Chapter 1:Introduction | 7 |  |
| Chapter 2:Software Requirement Analysis | 8-10 |  |
| Chapter 3:Language, Technology and Tools used | 11-13 |  |
| Chapter 4:Implementation and Interface | 14 |  |
| Chapter 5:Testing | 15-18 |  |
| Chapter 6: Conclusion | 19 |  |
| References | 20 |  |

**CHAPTER-1**

**INTRODUCTION**

**1.1 CONTEXT**

This Application ‘WhatsApp clone’ has been submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at GLA University, Mathura mentored by Mr. Bhanu Kapoor. This project has been completed approximately two months and has been executed in modules, meetings have been organised to check the progress of the work and for instructions and guidelines.

**1.2 MOTIVATION**

With social media and streaming services easily accessible, we may have a great online presence already. Therefore we need to learn about the presence and development of such powerful communication app. Therefore, we bring to you ‘WhatsApp Clone’, the one stop solution for exchanging text messages, photographs, videos across all devices and all platforms be it android or ios.

**1.3 OBJECTIVE**

The main objective of this website is to create ‘WhatsApp Clone’ is to create a link between people using different os devices. Similar to WhatsApp, out app provides real time one to one as well as group chat.

**1.4 SOURCES**

The source of our project (including all the project work, documentations and presentations) will is available at the following link:

https://github.com/shru-ty/wa\_clone

**CHAPTER -2**

**SOFTWARE REQUIREMENT ANALYSIS**

**2.1 HARDWARE AND SOFTWARE REQUIREMENTS**

**Hardware Requirement**

**Processor :** Intel

**Operating System :** Any Operating System

**RAM:** 8 GB (or higher)

**Hard disk:** 256GB

**Software Requirement**

**Software used:** Android Studio, Flutter SDK, Dart SDK

**Language used:** Dart

**Database:** Firestore Database

**User Interface Design:** Dart with flutter

**2.2 MODULES AND FUNCTIONALITIES**

**Modules**

* **Common :** Used for managing the properties which are common throughout the app and are used a number of times.
* **Config:** Used for configuration part of the app.
* **Features:** Used to create several diff features such as call,text,select contacts etc.
* **Models :** Used to create model of every property such as message,user etc

**Functionalities**

* Landing Screen UI
* Login Screen UI
* Flutter Firebase Phone Authentication
* Adding Riverpod
* Auth Controller
* About Riverpod
* Resolving iOS App Error
* OTP Screen UI
* Verifying OTP
* User Information Screen UI
* Saving User Data To Firestore
* Persisting Auth State
* Displaying Contacts in User’s Phone
* Displaying Name and online/offline status
* Modifying Bottom Chat Field UI
* Sending Text Message
* Displaying Chat Contacts
* Displaying Messages
* Automatic Scrolling on New Message

**2.3 RIVERPOD**

**Plugin**

A reactive caching and data-binding framework. Riverpod is inspired by Provider but solves some of it's key issues such as supporting multiple providers of the same type; awaiting asynchronous providers; adding providers from anywhere. Create/share/tests providers, with no dependency on Flutter. This includes being able to listen to providers without a BuildContext.

**At the end it is concluded that we have made effort on following points-**

* A description of the background and context of the project and its relation to work already done in the area.
* Made statement of the aims and objectives of the project.
* The description of Purpose, Scope, and applicability.
* We describe the requirement Specifications of the system and the actions that can be done on these things.
* We understand the problem domain and produce a model of the system, which
* describes operations that can be performed on the system.
* We included features and operations in detail, including screen layouts.
* We designed user interface and security issues related to system.
* Finally the system is implemented and tested according to test cases.

**CHAPTER-3**

**LANGUAGES, TECHNOLOGY AND TOOLS USED**

**3.1 FLUTTER**

Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase.

**3.2 DART**

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

**3.3 RIVERPOD**

A reactive caching and data-binding framework. Riverpod is inspired by Provider but solves some of it's key issues such as supporting multiple providers of the same type; awaiting asynchronous providers; adding providers from anywhere. Create/share/tests providers, with no dependency on Flutter. This includes being able to listen to providers without a BuildContext.

**3.4 FIREBASE**

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in realtime . It helps you develop high-quality apps, grow your user base, and earn more money. Each feature works independently, and they work even better together.

**3.5 FLUTTER PLUGINS USED**

The following plugins of flutter are used while making this application :

* cupertino\_icons: ^1.0.2
* firebase\_auth: ^3.4.1
* cloud\_firestore: ^3.2.1
* firebase\_storage: ^10.3.1
* country\_picker: ^2.0.17
* firebase\_core: ^1.24.0
* firebase\_core\_platform\_interface: 4.5.1
* flutter\_riverpod: ^2.0.0-dev.9
* image\_picker: ^0.8.6
* flutter\_contacts: ^1.1.5+1
* uuid: ^3.0.6
* intl: ^0.17.0
* cached\_network\_image: ^3.2.1
* cached\_video\_player: ^2.0.3
* emoji\_picker\_flutter: ^1.3.0
* flutter\_sound: ^9.2.13
* path\_provider: ^2.0.11
* permission\_handler: ^9.2.0
* audioplayers: ^1.0.1
* swipe\_to: ^1.0.2
* story\_view: ^0.13.2
* agora\_uikit: ^1.0.2
* flutter\_platform\_widgets: ^1.11.1

**3.5 GITHUB**

GitHub is an immense platform for code hosting. It supports version controlling and collaboration and allows developers to work together on projects. It offers both distributed version control and source code management (SCM) functionality of Git. It also facilitates collaboration features such as bug tracking, feature requests, task management for every project.

Essential components of the GitHub are:

* Repositories
* Branches
* Commits
* Pull Requests
* Git (the version control tool GitHub is built on)

**3.6 Android Studio**

Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version."

**CHAPTER-4**

**IMPLEMENTATION AND INTERFACE**

**4.1 IMPLEMENTATION**

The project is built with Dart Language using Flutter Framework in android studio. Front-end is built using various flutter properties and plugins and on the contrary, back-end is handled with the firebase console provided by google.

**4.2 USER INTERFACE DESIGN**

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

The following steps are various guidelines for User Interface Design:

1. The system user should always be aware of what to do next.

2. The screen should be formatted so that various types of information, instructions

and messages always appear in the same general display area.

3. Message, instructions or information should be displayed long enough to allow

the system user to read them.

4. Use display attributes sparingly.

5. Default values for fields and answers to be entered by the user should be specified.

6. A user should not be allowed to proceed without correcting an error.

7. The system user should never get an operating system message or fatal error.

**CHAPTER-5**

**TESTING**

**Implementation and Software Specification Testing**

**Detailed Design of Implementation**

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

**Technical Design**

**This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.**

**Test Specifications and Planning**

This activity prepares detailed test specifications for individual modules and programs, job streams, subsystems, and for the system as a whole**.**

**Programming and Testing**

This activity encompasses actual development, writing, and testing of program units or modules.

**User Training**

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

**Acceptance Test**

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

**Installation Phase**

In this phase the new Computerized system is installed, the conversion to new procedures is fully implemented, and the potential of the new system is explored.

**System Installation**

The process of starting the actual use of a system and training user personnel in its operation.

**Review Phase**

This phase evaluates the successes and failures during a systems development project, and to measure the results of a new Computerized Transystem in terms of benefits and savings projected at the start of the project.

**Development Recap**

A review of a project immediately after completion to find successes and potential problems in future work.

**Post-Implementation Review**

A review, conducted after a new system has been in operation for some time, to evaluate actual system performance against original expectations and projections for cost-benefit improvements. Also identifies maintenance projects to enhance or improve the system.

**THE STEPS IN THE SOFTWARE TESTING**

The steps involved during Unit testing are as follows:

a. Preparation of the test cases.

b. Preparation of the possible test data with all the validation checks.

c. Complete code review of the module.

d. Actual testing done manually.

e. Modifications done for the errors found during testing.

f. Prepared the test result scripts.

**The unit testing done included the testing of the following items:**

1. Functionality of the entire module/forms.
2. Validations for user input.
3. Checking of the Coding standards to be maintained during coding.
4. Testing the module with all the possible test data.
5. Testing of the functionality involving all type of calculations etc.
6. Commenting standard in the source files.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While System Integration, We integrated the modules one by one and tested the system at each step. This helped in reduction of errors at the time of the system testing.

**The steps involved during System testing are as follows:**

* Integration of all the modules/forms in the system.
* Preparation of the test cases.
* Preparation of the possible test data with all the validation checks.
* . Actual testing done manually.
* Recording of all the reproduced errors. Modifications done for the errors found during testing.
* Prepared the test result scripts after rectification of the errors.

**The System Testing done included the testing of the following items:**

1. Functionality of the entire system as a whole.

2. User Interface of the system.

3. Testing the dependent modules together with all the possible test data scripts.

4. Verification and Validation testing.

5. Testing the reports with all its functionality.

After the completion of system testing, the next following phase was the Acceptance Testing. Clients at their end did this and accepted the system with application. Thus, we reached the final phase of the project.

**There are other six tests, which fall under special category. They are described below:**

* Peak Load Test: it determines whether the system will handle the volume of handles of activities that occur when the system is at the peak of its processing demand. For example, test the system by activating all terminals at the same time.
* Storage Testing: It determines the capacity of the system to store transaction data on a disk
* or in other files.
* Performance Time Testing: it determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine how long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.
* Recovery Testing: This testing determines the ability of user to recover data or re-start system after failure. For example, load backup copy of data and resume processing without data or integrity loss.
* Procedure Testing: It determines the clarity of documentation on operation and uses of system by having users do exactly what manuals request. For example, powering down
* system at the end of week or responding to paper-out light on printer. Human Factors Testing: It determines how users will use the system when processing data or preparing reports.

**CHAPTER-6**

**CONCLUSION**

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

**At the end it is concluded that we have made effort on following points...**

* + A description of the background and context of the project and its relation to work already done in the area. Made statement of the aims and objectives of the project.
  + The description of Purpose, Scope, and applicability.
  + We define the problem on which we are working in the project.
  + We describe the requirement Specifications of the system and the actions that
  + can be done on these things.
  + We understand the problem domain and produce a model of the system, which
  + describes operations that can be performed on the system.
  + We included features and operations in detail, including screen layouts.
  + We designed user interface and security issues related to system.
  + Finally the system is implemented and tested according to test cases.

**REFERENCES**

• [www.google.com](http://www.google.com)

• <https://docs.flutter.dev/>

• <https://dart.dev/guides>

• https://developers.google.com/fonts/docs/material\_icons

• <https://appicon.com/>

* www.youtube.com