

TALKAHOLIC CHAT APPLICATION
A PROJECT REPORT

Submitted by

SELYA DHARSNEE M
(24MCR097)

SOWMIYA SREE M
(24MCR105)

SURYA N
(24MCR114)

*in partial fulfilment of the requirements
for the award of the degree
of*

MASTER OF COMPUTER APPLICATION

DEPARTMENT OF COMPUTER APPLICATION



KONGU ENGINEERING COLLEGE
(Autonomous)
PERUNDURAI, ERODE-638 060
DECEMBER 2024

**DEPARTMENT OF COMPUTER APPLICATION
KONGU ENGINEERING COLLEGE**

(Autonomous)

PERUNDURAI, ERODE-638 060

DECEMBER 2024

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**TALKAHOLIC CHAT APPLICATION**" is the bonafide record of project work done by **SELYA DHARSNEE M(Reg.No:24MCR097)**, **SOWMIYA SREE M (Reg.No:24MCR105)** and **SURYA N (Reg.No:24MCR114)** in partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications of Anna University, Chennai during the year 2024-2025.

SUPERVISOR

HEAD OF THE DEPARTMENT

(Signature with Seal)

Date:

Submitted for the end semester viva-voce examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We affirm that the project titled "**TALKAHOLIC CHAT APPLICATION**" being submitted in partial fulfilment of the requirements for the award of Master of Computer Application is the original work carried out by us. It has not formed the part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this occasion on this any other candidate.

Date:

SELYA DHARSNEE M

(REG.NO:24MCR097)

SOWMIYA SREE M

(REG.NO:24MCR105)

SURYA N

(REG.NO:24MCR114)

I certify that the declaration made by the above candidates is true to the best of my knowledge.

Date:

Name and Signature of the Supervisor

(Ms.S. HEMALATHA)

ABSTRACT

The Talkaholic Chat Application is an innovative messaging platform designed to enhance user communication by combining the familiar functionalities of existing chat applications with unique, engaging features. This project aims to provide a seamless user experience while fostering creativity and personalization in messaging. By allowing users to connect effortlessly through text, voice, and multimedia, Talkaholic prioritizes social interaction while incorporating elements that set it apart from conventional platforms.

One of the standout features of Talkaholic is the Timed Message Sending function, which enables users to schedule messages for delivery at a specific time. This feature not only helps manage communication more effectively but also adds an element of surprise and anticipation for the recipient. Whether it is sending a birthday greeting or a reminder, users can ensure their messages are received exactly when intended, thus enhancing the overall communication experience.

Additionally, the Theme Changing Feature empowers users to personalize their chat environment. With a variety of customizable themes, users can choose colors, fonts, and backgrounds that resonate with their individual styles and preferences. This customization promotes a sense of ownership and makes chatting more enjoyable. Overall, the Talkaholic Chat Application aims to redefine social messaging by merging fun, practicality, and personalization into a cohesive platform that caters to the diverse needs of modern users.

ACKNOWLEDGEMENT

First and foremost, we acknowledge the abundant grace and presence of almighty throughout different phases of the project and its successful completion.

We wish to express our hearty gratitude to our honorable Correspondent **Thiru.AKILANGO BCom., MBA., LLB.**, and other trust members for having provided us with all necessary infrastructures to undertake this projects. We extend our hearty gratitude to our honorable Principal **Dr.VBALUSAMY Mtech., Ph.D** for his consistent encouragement throughout our college days.

We would like to express our profound interest and sincere gratitude to our respected Head of the department **Dr.A.TAMILARASI MSc., MPhil., PhD., MTech.,** for her valuable guidance.

A special debt is owed to the project coordinator **Ms.S.HEMALETHA MCA.,** Department of Computer Technology for her encouragement and valuable advice that made us carry out the project work successfully.

We extend our sincere gratitude to our beloved guide Ms. **Dr.T. KAVITHA MCA., MPhil., PhD.,** Department of computer Technology for her ideas and suggestions, which have been very helpful to complete the project.

We are grateful to all the faculty and staff members of the Computer Technology Department and persons who directly and indirectly supported this project.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE
	ABSTRACT	
	ACKOWLEDGEMENT	
1	INTRODUCTION	
	1.1 OVERVIEW OF THE PROJECT	8
	1.2 PROBLRM DEFINITION	9
2	FEASIBILITY STUDY	11
	2.1 FUNCTIONAL FEASIBILITY	11
	2.2 OPERATIONAL FEASIBILITY	12
	2.3 ECONOMICAL FEASIBILITY	12
3	SYSTEM SPECIFICATIONS	13
	3.1 HARDWARE REQUIREMENTS	13
	3.2 SOFTWARE REQUIREMENTS	13
	3.3 SOFTWARE DESCRIPTION	13
	3.3.1 FRONT-END: FLUTTER	13
	3.3.2 BACKEND-END: FIREBASE	14
4	SYSTEM ANALYSIS	15
	4.1 EXISTING SYSTEM	15
	4.1.1 DRAWBACKS OF EXISTING SYSTEM	16
	4.2 PROPOSED SYSTEM	17
	4.2.1 NEED FOR PROPOSED SYSTEM	17
	4.2.2 ADVANTAGES OF THE PRPOSED SYSTEM	18

5	SYSTEM DESIGN	19
	5.1 USE CASE DIAGRAM	19
	5.2 ACTIVITY DIAGRAM	20
	5.3 DATA FLOW DIAGRAM (DFD)	21
6	CONCLUSION AND FUTURE ENHANCEMENT	22
	5.1 CONCLUSION	22
	5.2 FUTURE ENHANCEMENT	23
	APPENDICES	24
	A. SAMPLE CODING	24
	B. SCREENSHOTS	36
	REFERENCES	41

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Talkaholic Chat Application is a state-of-the-art messaging system that has been designed to revolutionise user interaction and communication in the digital sphere. The application, which offers a fresh take on conventional messaging platforms, prioritises individualisation and creativity while fusing ease of use with distinctive features.

By combining text, audio, and multimedia features, Talkaholic provides a dynamic communication experience that is intended to meet the needs of modern users. The goal is to enhance social interaction while incorporating innovative features that enhance the overall user experience.

One of the greatest advantages is the ability to send messages at specific times. This feature enhances the efficiency and thoughtfulness of communication by allowing users to schedule messages for delivery at certain times.

Whether it is a prearranged reminder for a big event or a surprise birthday greeting sent at midnight, users can ensure that their messages arrive at the precise moment they're meant to. By adding a feeling of anticipation and improving convenience, this feature makes occasions memorable for both senders and recipients.

The Theme Changing Feature, which lets users customise their chat experience, is another unique feature. By offering a variety of themes with editable fonts, colours, and backgrounds, Talkaholic enables users to express who they are through digital conversations. The customisation boosts the overall enjoyment of using the product and fosters a sense of ownership. The platform may be tailored to the preferences and moods of users, regardless of whether they like professional, minimalist themes or vibrant, lively visuals.

In addition to these features, the versatility of the Talkaholic chat enables seamless multimedia integration. The ease with which users may share photos, videos, and voice messages adds excitement and vibrancy to conversations.

This wide range of features ensures that Talkaholic caters to a wide range of clients, from individuals wanting a customised and creative chat experience to businesses searching for sophisticated and efficient communication solutions. Because it emphasises both individual and group interactions, it is the ideal platform for a range of uses.

All things considered, by combining utility, fun, and personalisation, the Talkaholic Chat Application completely transforms the chatting experience. Talkaholic sets itself apart in the very competitive messaging app industry by addressing gaps in traditional platforms and incorporating unique features. Its commitment to fostering creativity and enhancing user experiences makes it a forward-thinking solution to today's communication issues, paving the way for a more engaging and flexible social messaging experience.

1.2 PROBLEM DEFINITION

Messaging systems are becoming a crucial component of everyday communication in the current digital era, supporting social, professional, and personal objectives. However, the majority of current applications offer a generic and repetitive user experience due to a lack of innovation in functionality and personalisation. Restrictive customisation possibilities, a lack of instruments to promote dynamic and engaging communication, and restricted capabilities for scheduling critical communications are some of the main obstacles. These restrictions hinder users' ability to communicate effectively, express themselves completely, and have a smooth, personalised texting experience.

Whether for professional follow-ups, timely greetings, or reminders, scheduling communications is a regular necessity. Unfortunately, the majority of platforms in use today need manual involvement throughout the sending process, which results in missed opportunities and hassle. Similar to this, many programs lack personalisation, which is an essential component of the contemporary user experience. Users frequently have limited customisation choices and are restricted to predetermined themes, which limits their ability to be creative and unique in their interactions.

Furthermore, the immersive communication atmosphere created by current messaging services is frequently lacking. Interactions may seem repetitious and one-dimensional if text, speech, and multimedia material cannot be effectively integrated. This dearth of variety in communication tools ignores the many demands of users, especially those seeking a platform that blends enjoyment, usefulness, and flexibility.

These restricted features provide difficulties, especially for creative users and businesses. While creative individuals want platforms that let them explore and express their personality, professionals need effective solutions like branded themes and scheduled messages to streamline operations and maintain a professional image. A void in the market for a flexible messaging platform is caused by the lack of such functionality.

By providing cutting-edge features like multimedia integration, customisable themes, and timed message sending, the Talkaholic Chat Application aims to solve these problems. In order to overcome the shortcomings of current programs and satisfy the requirements of contemporary communication, the platform seeks to reinvent the messaging experience by attending to the various needs of users and placing a high priority on both functionality and personalisation.

CHAPTER 2

FEASIBILITY STUDY

- Functional Feasibility
- Operational Feasibility
- Economical Feasibility

2.1 FUNCTIONAL FEASIBILITY

This section evaluates whether the proposed features of the Talkaholic application can be realistically developed and implemented using existing technology and tools.

Key Features Assessment

2.1.1 Timed Message Sending

- **Feasibility:** This functionality requires implementing a scheduled task system that triggers messages at predefined times. Cloud-based platforms like Firebase or AWS support scheduled functions, making implementation straightforward.
- **Challenges:** Ensuring accuracy in message scheduling and delivery during varying internet conditions.

2.1.2 Mini Games

- **Feasibility:** Simple games such as word puzzles and encryption challenges can be implemented using existing libraries and game engines optimized for mobile.
- **Challenges:** Ensuring these games run smoothly across different mobile platforms without overloading device resources.

2.1.3 Theme Customization

- **Feasibility:** With advancements in front-end frameworks like Flutter or React Native, implementing customizable UI themes is technically achievable.
- **Challenges:** Maintaining a balance between personalization options and application performance.

2.1.4 AI Chatbot

- **Feasibility:** Leveraging API-based AI models (like OpenAI or Google Dialog flow) makes chat bot integration possible.
- **Challenges:** Ensuring the chat bot delivers meaningful responses and handles a wide range of user queries.

2.2 OPERATIONAL FEASIBILITY

2.2.1 Target Audience Compatibility

- Talkaholic is designed for users seeking engaging communication tools combined with entertainment and productivity features.
- Its features, such as themes, mini-games, and AI assistance, cater to a broad user base, including casual users, professionals, and students.

2.2.2 User Accessibility

- The app is mobile-first and ensures compatibility across Android and iOS platforms.
- The absence of message notifications might require manual effort from users, but it offers opportunities to improve focus for those preferring less intrusive interactions.

2.2.3 Ease of Use

- Intuitive UI design ensures that users can navigate features, such as scheduling messages, customizing themes, or converting files, without extensive instructions.
- Engaging tools like mini-games enhance user retention and satisfaction.

2.3 ECONOMICAL FEASIBILITY

The evaluation of development costs is weighted according to the final revenue or benefits of the development system. Does not require additional hardware or software to develop this project. Therefore, the development of the project in the organization is economically reasonable.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE REQUIREMENTS

- Processor : Intel i5
- Processor Speed : Min 250MHz
- RAM : 8GB RAM
- Hard Disk : 512 GB

3.2 SOFTWARE REQUIREMENTS

- Front end : Flutter
- Back end : Sheets, Firebase

3.3 SOFTWARE DESCRIPTION

3.3.1 FRONT END: FLUTTER

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications from a single codebase for any web browser, Fuchsia, Android, iOS, Linux, macOS, and Windows. First described in 2015, Flutter was released in May 2017.

FEATURES

- Cross-platform Development Framework
- Same UI and Business Logic in All Platforms
- Fast Development due to the hot reload feature
- Highly Customizable UI Design
- A Wide Variety of Rich Widgets
- Access to Native Features
- Publicly Accessible

3.3.2 BACK END: FIREBASE PERFORMANCE

Firebase Performance Monitoring is a service that automatically captures performance data from your app, such as network requests, app start time, screen rendering time, and more. You can also add custom traces and attributes to measure specific aspects of your app's performance.

ADVANTAGES OF FIREBASE

- Reliable & Extensive Databases
- Fast & Safe Hosting
- Provides a Free Start to Newbies
- Google Analytics
- Firebase Cloud Messaging for Cross-Platform

SPREADSHEET

A spreadsheet is a computer program that can capture, display and manipulate data arranged in rows and columns. Spreadsheets are one of the most popular tools available with personal computers. A spreadsheet is generally designed to hold numerical data and short text strings.

ADVANTAGES OF SPREADSHEETS

- Spreadsheets require minimal training.
- Spreadsheets are customizable.
- Spreadsheets can be more collaborative than other tools.
- It's easy to manipulate and analyze data.

CHAPTER 4

SYSTEM ANALYSIS

4.1. EXISTING SYSTEM

The current system of communication primarily relies on conventional messaging platforms that offer basic features like text, audio, and multimedia messaging. While functional, these platforms have reached a saturation point in terms of innovation and fail to address evolving user demands for enhanced personalization and creative engagement.

The existing applications generally include straightforward chat interfaces with limited customization options, such as static themes or restricted file-sharing capabilities. Advanced communication experiences like scheduling messages, converting text to PDFs, or adding gamified interactions are either missing or require third-party tools, complicating the user experience.

Additionally, many of these platforms suffer from intrusive notifications or overly simplistic designs that lack interactive elements to keep users engaged. They are designed more for functionality than fun, leaving gaps in addressing user satisfaction. Security concerns with AI chat features and limited integration of payment gateways for premium content also hinder user adoption.

Moreover, these systems offer no mechanism to incorporate user moods into their interfaces, limiting creativity and personalization. As a result, users seeking more diverse, enjoyable communication tools often feel underwhelmed and unsupported.

4.1.1. DRAWBACKS OF EXISTING SYSTEM

The existing chat systems exhibit several drawbacks that reduce their appeal and fail to meet modern user expectations. Key drawbacks include:

- **Lack of Personalization and Customization:**
Users cannot fully customize their chat interfaces with mood-based themes or fonts, which reduces engagement. Limited aesthetic appeal detracts from the overall experience.
- **Missing Unique Communication Features:**
Common platforms lack innovative tools such as "Timed Message Sending," integrated mini-games, or file format conversion features like text-to-PDF. This diminishes their utility.
- **Notification Overload or Absence of Control:**
Intrusive notifications often interrupt user workflows, while the lack of subtle reminders can result in missed communications, impacting user satisfaction.
- **No Fun or Productivity Boosters:**
Most platforms fail to incorporate recreational elements such as AI-driven games or functional tools like encryption-decryption utilities.
- **Minimal Adaptation to Modern Needs:**
As chat applications become more integrated with productivity tools, existing systems have failed to adapt by offering advanced features like AI chatbots or seamless payment integration for premium services.

Addressing these drawbacks through advanced, user-centric features forms the core rationale for developing Talkaholic Chat.

4.2. PROPOSED SYSTEM

Based on a detailed analysis of the existing system, Talkaholic Chat Application is proposed as a solution that overcomes the limitations of current messaging platforms. Designed with innovation and user engagement in mind, it integrates seamless communication with productivity and entertainment tools, offering users a versatile, personalized experience.

The system introduces creative features such as Timed Message Sending for scheduled delivery, interactive Mini-Games, and Mood-Based Theme Customization, enabling users to express themselves creatively. Additionally, tools for File Format Conversion and an AI Chatbot add value by enhancing user productivity while incorporating an element of playfulness.

Key functionalities include intuitive navigation, real-time chat options, interactive file-sharing tools, and secure WhatsApp Payment Integration for accessing premium content. This cross-platform application, built using the Flutter framework, ensures consistency across Android and iOS devices.

4.2.1. NEED FOR PROPOSED SYSTEM

- Offering enhanced customization, enabling users to tailor the app to their preferences.
- Simplifying communication through intelligent features like scheduled messaging and file conversion tools.
- Minimizing manual efforts, thus reducing user workload and improving efficiency.
- Enhancing overall user satisfaction through integrated entertainment options, such as mini-games.

4.2.2. ADVANTAGES OF PROPOSED SYSTEM

- **Innovative Features:** Tools like "Timed Message Sending" and integrated mini-games keep the platform engaging.
- **Customization:** Mood-based themes create a sense of individuality and offer better user engagement.
- **Cross-Platform Compatibility:** Seamless performance across Android and iOS broadens its appeal

4.2.3. PROJECT DESCRIPTION: Talkaholic Chat

Communication Module

This module encompasses the core messaging functionalities such as text and multimedia messaging, real-time chats, and Timed Message Sending.

Productivity Module

Features include tools for converting text to PDFs and encrypting/decrypting text, enhancing the platform's utility for both casual users and professionals.

Entertainment Module

Mini-games such as word puzzles and encryption challenges are integrated to keep users entertained within the app.

Customization Module

Mood-based themes, customizable fonts, and interactive user interfaces enable users to personalize their chat environments.

AI Chatbot Module

An intelligent chatbot assists users in generating messages, answering queries, and providing conversational support.

CHAPTER 5

SYSTEM DESIGN

5.1. USECASE DIAGRAM

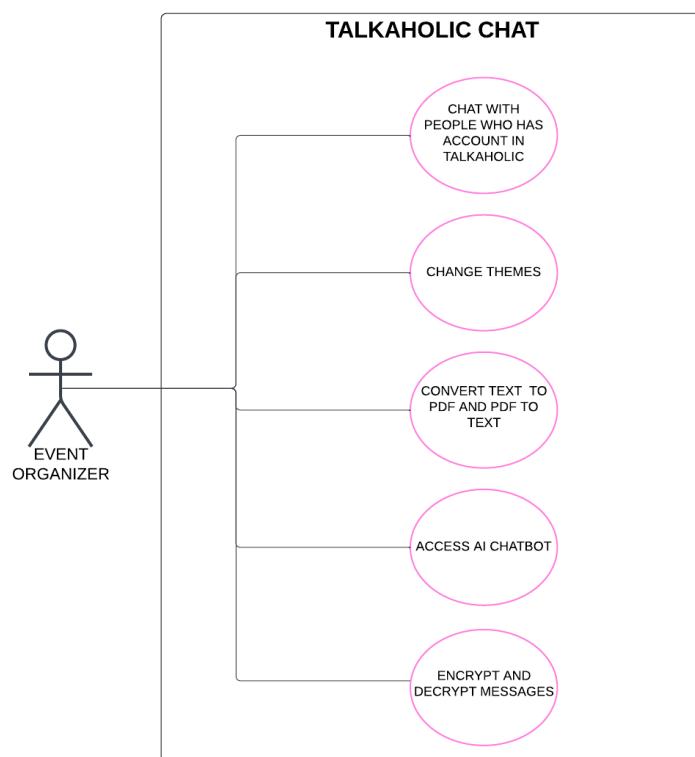


FIGURE 5.1 USE CASE DIAGRAM OF TALKAHOLIC CHAT

- **Chat with Other Users:** Engage with other registered users on the platform.
- **Theme Customization:** Personalize the interface by changing themes.
- **Text and PDF Conversion:** Convert text to PDF and extract text from PDF files.
- **AI Chatbot:** Interact with an AI chatbot for assistance or casual conversation.
- **Message Encryption/Decryption:** Securely send and receive messages using encryption.

5.2. ACTIVITY DIAGRAM

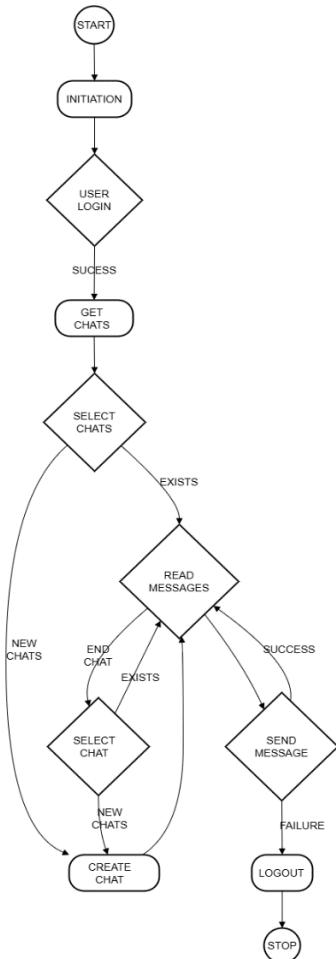


FIGURE 5.2 ACTIVITY DIAGRAM OF TALKAHOLIC CHAT

- **Start:** The process begins with initiation.
- **User Login:** Users are prompted to log in. If successful, the system retrieves available chats.
- **Select Chats:** Users can select existing chats. If a chat exists:
 - a. They can read messages.
 - b. Send messages successfully or handle failure.
 - c. End the chat if needed.
- **New Chat:** If no chats exist, users can create a new chat.
- **Logout/Stop:** If the process fails or a user chooses to exit, the system logs out and stops.

5.3. DATAFLOW DIAGRAM

LEVEL 0

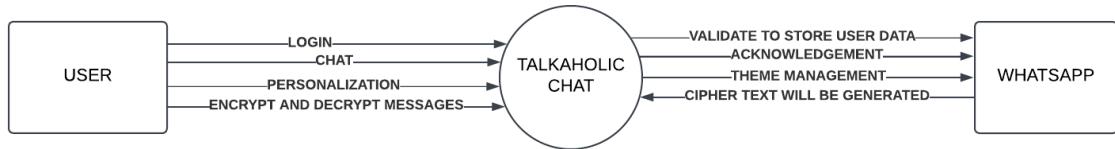


FIGURE 5.3 LEVEL 0 DATAFLOW DIAGRAM

The diagram shows a chat system, **Talkaholic Chat**, linking **User** and **WhatsApp**.

- **User ↔ Talkaholic Chat:** Handles login, chatting, personalization, and message encryption/decryption.
- **Talkaholic Chat ↔ WhatsApp:** Validates user data, manages themes, acknowledges actions, and encrypts messages for secure communication.

LEVEL 1

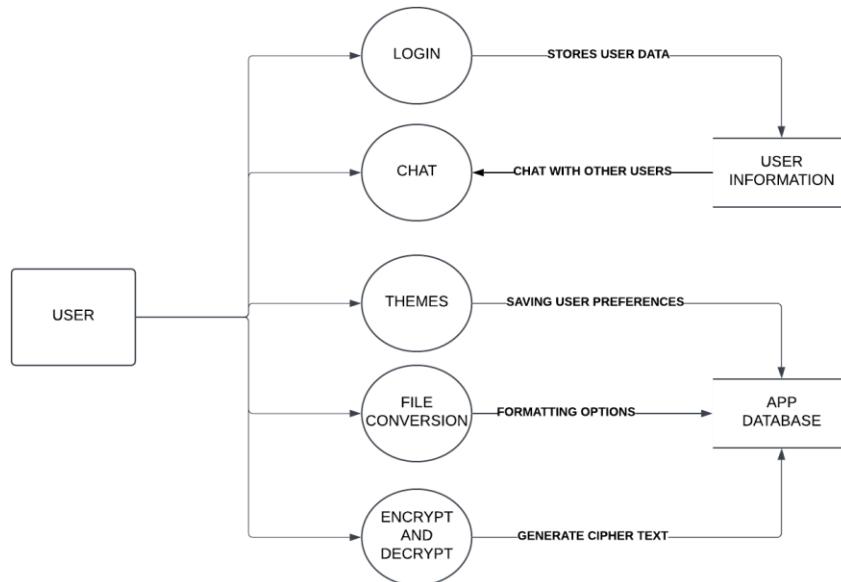


FIGURE 5.3 LEVEL 1 DATAFLOW DIAGRAM

- **Login** connects to **User Information** for user-specific records.
- **Chat, Themes, File Conversion, and Encrypt and Decrypt** connect to the **App Database** to store, retrieve, or manipulate application data

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1. CONCLUSION

The Talkaholic Chat Application is a dynamic messaging platform designed to enhance digital communication by integrating creativity, personalization, and efficiency. It offers real-time messaging, multimedia sharing, and group chat functionalities, coupled with innovative features like mood-based themes and timed message sending. Users can customize their chat experience by choosing from a variety of themes that adjust based on mood or personal preferences, creating a tailored and engaging environment. Additionally, interactive mini-games within the app allow users to play with friends or during chats, encouraging social connection and making conversations more lively.

One standout feature of Talkaholic is Text-to-PDF conversion tool, which allows users to effortlessly turn chat conversations or personal notes into shareable PDF files. This functionality supports customization, enabling users to format PDFs with personalized fonts and layout options. It also integrates seamlessly with other features like reminders and scheduled message delivery, offering a holistic approach to both productivity and personal engagement. Furthermore, the inclusion of an AI-powered chatbot helps improve user experience by acting as both an intelligent conversational partner and a virtual assistant, capable of setting reminders, managing tasks, and providing content recommendations based on user preferences.

With cross-platform compatibility for both iOS and Android, Talkaholic ensures a consistent and smooth experience across devices. The app emphasizes security and privacy with end-to-end encryption, data protection features, and AI-based moderation to ensure a safe communication space. The combination of fun and functional features, like mood-based theming, interactive games, AI chatbot assistance, and productivity tools, positions Talkaholic as a versatile and innovative platform ideal for users who value both enjoyment and efficiency in their digital interactions. Whether for personal chat or business communication, it delivers a comprehensive and engaging messaging experience.

6.2. FUTURE ENHANCEMENT

To enhance the Talkaholic Chat Application, we will introduce several exciting features, including **voice and video calling**, **file sharing**, and **scheduled message sending**. By adding voice and video calling capabilities, Talkaholic users will be able to transition from text-based conversations to real-time communication with high-quality audio and visual support. The integration of seamless video and voice calling will cater to users who require more personal or professional interactions, making it ideal for both casual chats and remote work collaborations. With **group call** support, users can engage in larger team discussions or social hangouts directly through the app, making Talkaholic a one-stop platform for all types of communication.

In addition to enhanced communication, **file sharing** will allow users to send documents, images, videos, and more directly within the chat interface. This new functionality will significantly improve user experience, especially for professional or academic users who frequently need to exchange important files. With a secure, easy-to-use file sharing system, Talkaholic will support multiple file formats and allow users to manage and download files with minimal friction. Whether for work documents, family pictures, or important receipts, the app will become a comprehensive hub for sharing not only text and media but also practical files in an efficient manner.

Finally, the **scheduled message feature** will take communication convenience to the next level. Users will be able to compose messages and schedule them to be sent at a specific time, which is perfect for sending reminders, surprise messages, or communicating across different time zones without the need for immediate interaction. This feature is highly practical for both personal use—such as sending birthday wishes or important reminders—and professional use, ensuring that business communications are timely and well-organized. Together, the introduction of these features will transform Talkaholic into a more robust and versatile messaging platform, balancing real-time interaction with efficiency, flexibility, and enhanced communication capabilities.

APPENDICES

A.SAMPLE CODING

Main.dart

```

import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'utility.dart';
import 'themes/theme_provider.dart'; // Ensure correct import
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ChangeNotifierProvider(
      create: (context) => ThemeProvider(),
      child: Consumer<ThemeProvider>(
        builder: (context, themeProvider, child) {
          return MaterialApp(
            debugShowCheckedModeBanner: false,
            theme: ThemeData(
              useMaterial3: true,
              colorScheme: themeProvider.lightScheme,
              scaffoldBackgroundColor: themeProvider.lightScheme.background),
            darkTheme: ThemeData(
              useMaterial3: true,
              colorScheme: themeProvider.darkScheme,
              scaffoldBackgroundColor: themeProvider.darkScheme.background),
            themeMode: themeProvider.themeMode,
            home: HomePage(),
          )
        }
      )
    );
  }
}
```

WORD DASH GAME.DART

```

import 'package:flutter/material.dart';
import 'dart:async';
import 'dart:math';
void main() {
  runApp(WordDashApp());
}
class WordDashApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: WordDashGame(),
    );
  }
}
class WordDashGame extends StatefulWidget {
  @override
  _WordDashGameState createState() => _WordDashGameState();
}
class _WordDashGameState extends State<WordDashGame> {
  // List of words with their hints and categories
  final List<Map<String, String>> masterWordList = [
    {"word": "apple", "hint": "A red or green fruit", "category": "Fruit"},
    {"word": "banana", "hint": "A yellow fruit", "category": "Fruit"},
    {"word": "zebra", "hint": "An animal with stripes", "category": "Animal"},
    {"word": "ocean", "hint": "Large body of water", "category": "Nature"},
    {"word": "planet", "hint": "A celestial body orbiting a star", "category": "Space"},
    {"word": "python", "hint": "A programming language", "category": "Technology"},
    {"word": "violin", "hint": "A string instrument", "category": "Music"},
    {"word": "chocolate", "hint": "A sweet brown treat", "category": "Food"},
    {"word": "volcano", "hint": "A mountain that erupts", "category": "Nature"},
    {"word": "parrot", "hint": "A colorful talking bird", "category": "Animal"},
```

```

late List<Map<String, String>> words; // Current word list
int currentWordIndex = 0;
String currentWord = "";
String displayWord = "";
String hint = "";
String category = "";
TextEditingController _controller = TextEditingController();
int hearts = 3;
int points = 0;
int timer = 30;
bool gameOver = false;
bool isHintUsed = false;
late Timer countdownTimer;
Random random = Random();
@Override
void initState() {
  super.initState();
  _shuffleWordList();
  startNewWord();
  startTimer();
}
// Shuffle the word list to create a fresh experience
void _shuffleWordList() {
  words = List.from(masterWordList)..shuffle(random);
}
// Start a new word in the game
void startNewWord() {
  if (currentWordIndex < words.length) {
    setState(() {
      currentWord = words[currentWordIndex]["word"]!;
      hint = words[currentWordIndex]["hint"]!;
      category = words[currentWordIndex]["category"]!;
      displayWord = _getUnderscoreWord(currentWord);
    });
  }
}

```

```
isHintUsed = false;
timer = 30; // Reset the timer for the new word
});
} else {
setState();
gameOver = true;
});
countdownTimer.cancel();
}
}

// Generate underscores for the word
String _getUnderscoreWord(String word) {
return List.generate(word.length, (index) => "_").join(" ");
}

void checkGuess(String guess) {
if (guess == currentWord) {
setState();
points += 10;
currentWordIndex++;
startNewWord();
});
} else {
setState();
hearts -= 1;
if (hearts <= 0) {
gameOver = true;
countdownTimer.cancel();
}
}
}

_controller.clear();
} void showHint() {
if (!isHintUsed) {
```

```
setState() {
    isHintUsed = true;
    int indexToReveal = random.nextInt(currentWord.length - 2) + 1;
    List<String> displayList = displayWord.split(" ");
    displayList[indexToReveal] = currentWord[indexToReveal];
    displayWord = displayList.join(" ");
});
}

} void startTimer() {
    countdownTimer = Timer.periodic(Duration(seconds: 1), (timer) {
        if (this.timer <= 1) {
            setState(() {
                gameOver = true;
            });
            countdownTimer.cancel();
        } else {
            setState(() {
                this.timer -= 1;
            });
        }
    });
}

}

@Override
Widget build(BuildContext context) {
    double screenWidth = MediaQuery.of(context).size.width;
    double screenHeight = MediaQuery.of(context).size.height;
    return Scaffold(
        appBar: AppBar(
            title: Text("Word Dash", style: TextStyle(fontSize: screenWidth * 0.07)),
            centerTitle: true,
        ),
        body: gameOver
            ? Center(

```

```

child: Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Text("Game Over!", style: TextStyle(fontSize: 30, color: Colors.red, fontWeight: FontWeight.bold)),
    Text("Your Score: $points", style: TextStyle(fontSize: 20)),
    SizedBox(height: 20),
    ElevatedButton(
      onPressed: () {
        setState(() {
          hearts = 3;
          points = 0;
          currentWordIndex = 0;
          _shuffleWordList();
          gameOver = false;
        });
        startNewWord();
        startTimer();
      },
      child: Text("Restart"),
    ),
  ],
),
),
),
)
: Padding(
  padding: const EdgeInsets.all(16.0),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Text("Category: $category", style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
      SizedBox(height: 10),
      Text("Hint: $hint", style: TextStyle(fontSize: 20, fontStyle: FontStyle.italic)),
      SizedBox(height: 20),
    ],
  ),
);

```

```
Text(displayWord, style: TextStyle(fontSize: 36, letterSpacing: 8)),  
SizedBox(height: 20),  
Row(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: [  
    Text("Time Left: $timer", style: TextStyle(fontSize: 18)),  
    SizedBox(width: 20),  
    Text("Hearts: ${"❤️ " * hearts}", style: TextStyle(fontSize: 18)),  
  ],  
),  
SizedBox(height: 20),  
TextField(  
  controller: _controller,  
  decoration: InputDecoration(  
    labelText: "Your Guess",  
    border: OutlineInputBorder(),  
  ),  
,  
SizedBox(height: 20),  
ElevatedButton(onPressed: () => checkGuess(_controller.text.toLowerCase()), child:  
  Text("Submit Guess")),  
SizedBox(height: 10),  
ElevatedButton(onPressed: showHint, child: Text("Show Hint")),  
],  
,  
,  
);  
}  
}
```

AI CHAT.DART

```

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:google_generative_ai/google_generative_ai.dart';
import 'package:intl/intl.dart';
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Ai chat',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const ChatScreen(),
    );
  }
}

class ChatScreen extends StatefulWidget {
  const ChatScreen({super.key});
  @override
  State<ChatScreen> createState() => _ChatScreenState();
}

class _ChatScreenState extends State<ChatScreen> {
  TextEditingController _userInput = TextEditingController();
  static const apiKey = "AIzaSyDu1ZzP1KKgb6slJURmDTrMlorPZyOkF3c";
  final model = GenerativeModel(model: 'gemini-pro', apiKey: apiKey);
}

```

```
final List<Message> _messages = [];
Future<void> sendMessage() async{
final message = _userInput.text;
setState(() {
_messages.add(Message(isUser: true, message: message, date: DateTime.now()));
_userInput.clear();
});
final content = [Content.text(message)];
final response = await model.generateContent(content);
setState(() {
_messages.add(Message(isUser: false, message: response.text?? "", date: DateTime.now()));
});
}
@Override
Widget build(BuildContext context) {
return Scaffold(
body: Container(
decoration: BoxDecoration(
image: DecorationImage(
colorFilter: new ColorFilter.mode(Colors.black.withOpacity(0.8), BlendMode.dstATop),
image:AssetImage('assets/bg.png'),
fit: BoxFit.cover
)
),
child: Column(
mainAxisAlignment: MainAxisAlignment.end,
children: [
Expanded(
child: ListView.builder(itemCount:_messages.length,itemBuilder: (context,index){
final message = _messages[index];
return Messages(isUser: message.isUser, message: message.message, date:
DateFormat('HH:mm').format(message.date));
}
)
]
)
]
)
);
```

```
)  
,  
Padding(  
padding: const EdgeInsets.all(8.0),  
child: Row(  
children: [  
Expanded(  
flex: 15,  
child: TextFormField(  
style: TextStyle(color: const Color.fromARGB(255, 255, 255, 255)),  
controller: _userInput,  
decoration: InputDecoration(  
border: OutlineInputBorder(  
borderRadius: BorderRadius.circular(15),  
,  
label: Text('Enter Your Message')  
,  
,  
,  
),  
Spacer(),  
IconButton(  
padding: EdgeInsets.all(12),  
iconSize: 30,  
style: ButtonStyle(  
backgroundColor: MaterialStateProperty.all(Colors.black),  
foregroundColor: MaterialStateProperty.all(Colors.white),  
shape: MaterialStateProperty.all(CircleBorder())  
,  
onPressed: (){  
sendMessage();  
},  
icon: Icon(Icons.send))  
],
```

```
),
)
],
),
),
),
);
}
}
```

```
class Message{
final bool isUser;
final String message;
final DateTime date;
Message({ required this.isUser, required this.message, required this.date });
}

class Messages extends StatelessWidget {
final bool isUser;
final String message;
final String date;
const Messages(
{
super.key,
required this.isUser,
required this.message,
required this.date
});
@Override
Widget build(BuildContext context) {
return Container(
width: double.infinity,
padding: EdgeInsets.all(15),
margin: EdgeInsets.symmetric(vertical: 15).copyWith(
left: isUser ? 100:10,
```

```
right: isUser ? 10: 100
),
decoration: BoxDecoration(
color: isUser ? Colors.blueAccent : Colors.grey.shade400,
borderRadius: BorderRadius.only(
topLeft: Radius.circular(10),
bottomLeft: isUser ? Radius.circular(10): Radius.zero,
topRight: Radius.circular(10),
bottomRight: isUser ? Radius.zero : Radius.circular(10)
)
),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text(
message,
style: TextStyle(fontSize: 16,color: isUser ? Colors.white: Colors.black),
),
Text(
date,
style: TextStyle(fontSize: 10,color: isUser ? Colors.white: Colors.black,),
)
],
),
);
}
```

B.SCREENSHOTS

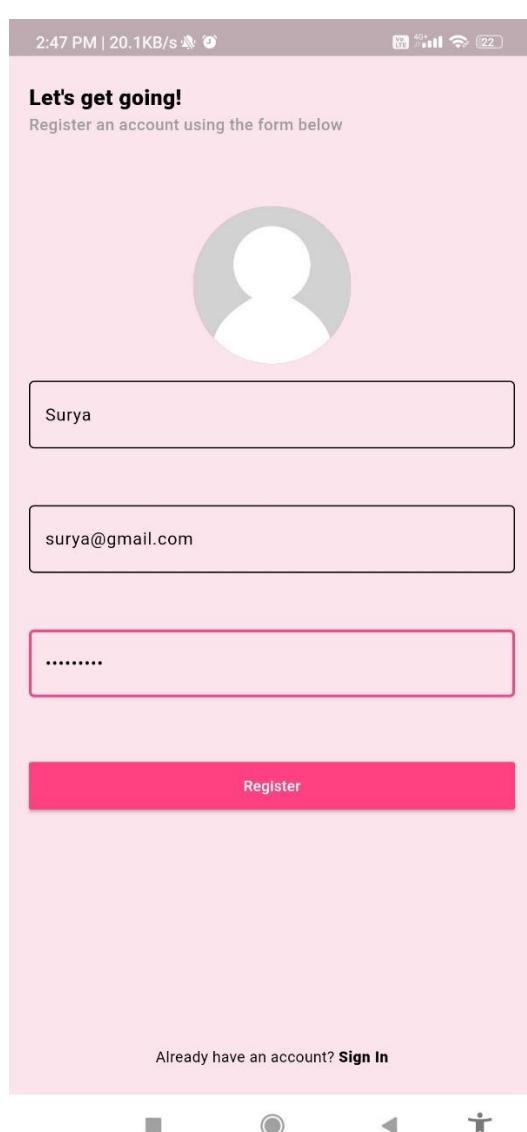


FIGURE B.1 SIGN UP PAGE

- A clean registration screen with fields for name, email, and password, plus a prominent Register button.
- A profile icon placeholder enhances user interface aesthetics.
- A bottom link allows navigation to the Sign In page for existing users.

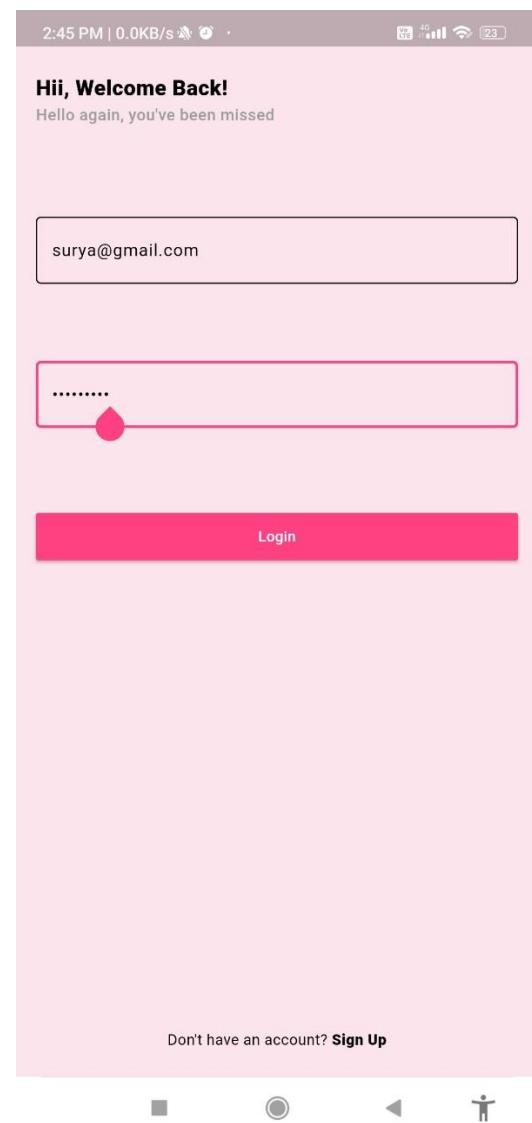


FIGURE B.2 LOGIN PAGE

- Login screen with email, password fields, and a pink Login button.
- Friendly welcome message: Hi, Welcome Back!
- Bottom link for navigation to the Sign Up page.

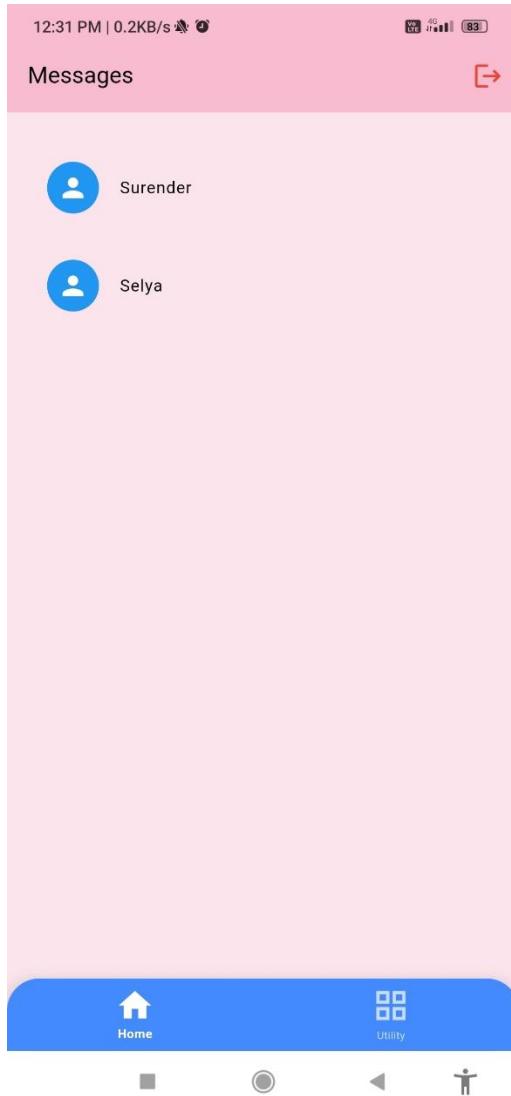


FIGURE B.3 CHATS

This Figure B.3 Represents Pink screen with the title Messages. Two contacts: Surender and Selya. Bottom navigation bar with Home and Utility icons.

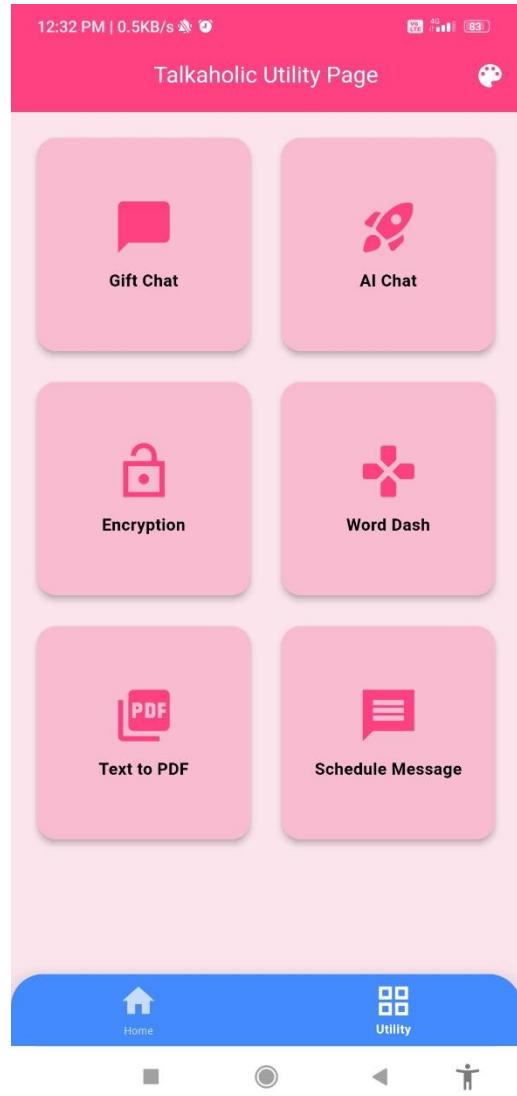


FIGURE B.4 UTILITY PAGE

This Figure B.4 Represents Six options: Gift Chat, AI Chat, Encryption, Word Dash, Text to PDF, Schedule Message. Bottom bar with Home and Utility.

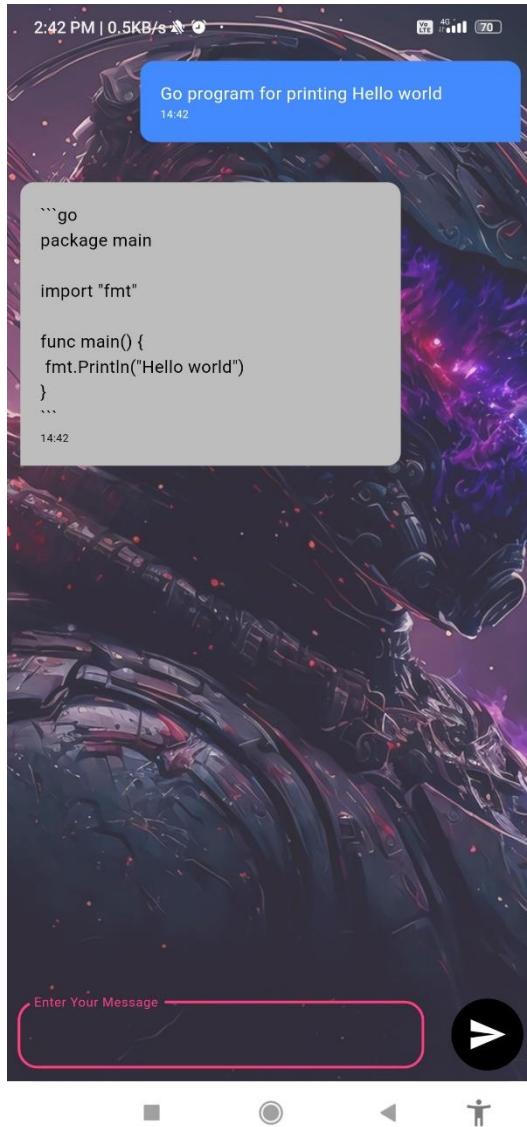


FIGURE B.5 AI CHAT BOT

This Figure B.5 Represents AI Chat screen showing a Go program to print Hello world.

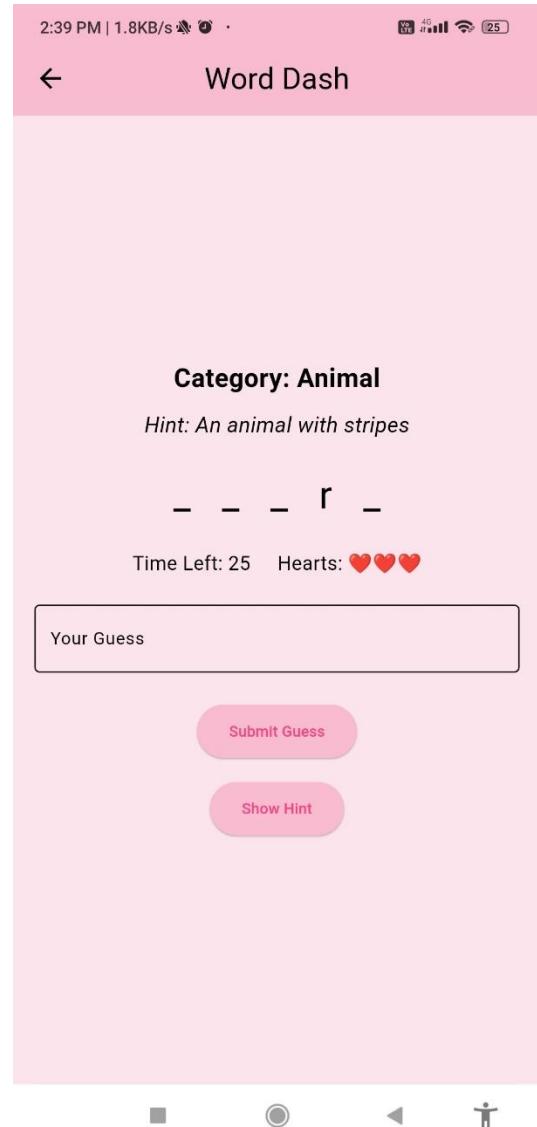


FIGURE B.6 WORD GAME

This Figure B.6 Represents The game is called Word Dash and is about guessing words. The hint is An animal with stripes, with the word _ _ _ r _ .The player has 25 seconds, three hearts, and options to guess or get a hint.

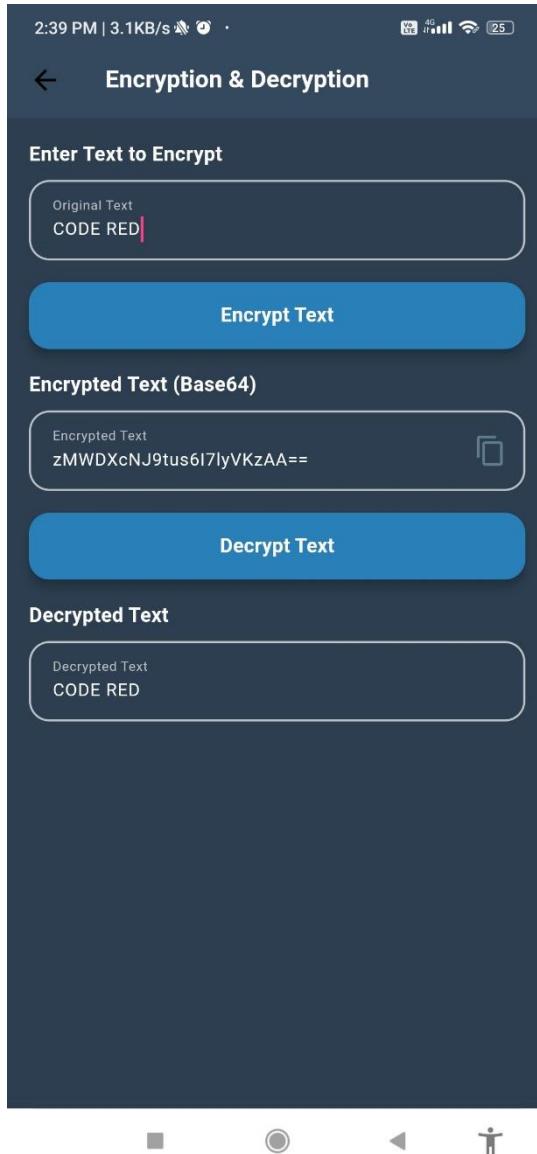


FIGURE B.7 ENCRYPTION AND DECRYPTION

This Figure B.7 Represents The app performs encryption and decryption of text using Base64. The input CODE RED is encrypted and then decrypted.

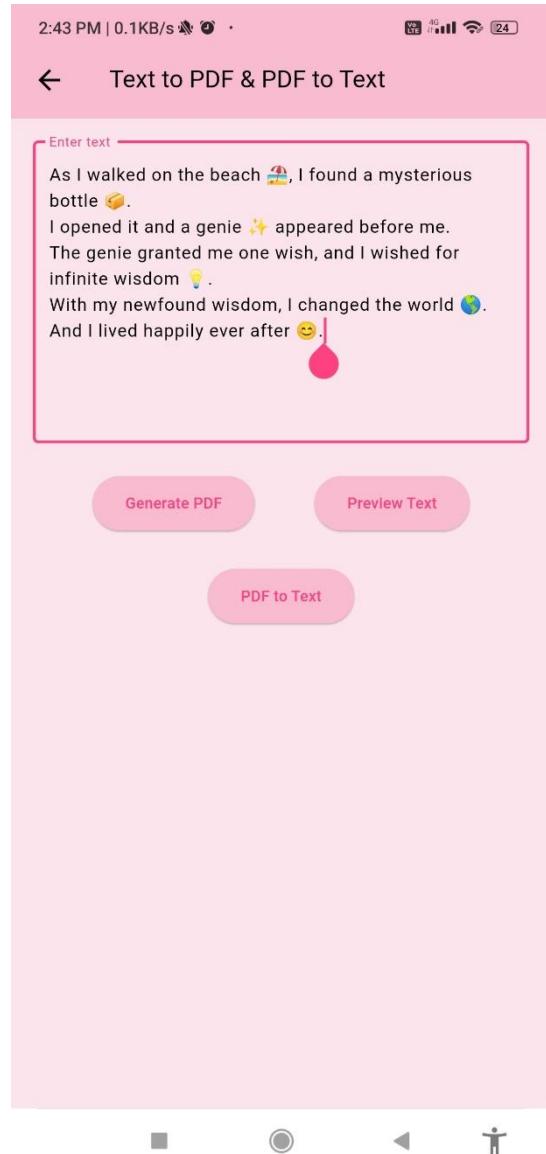
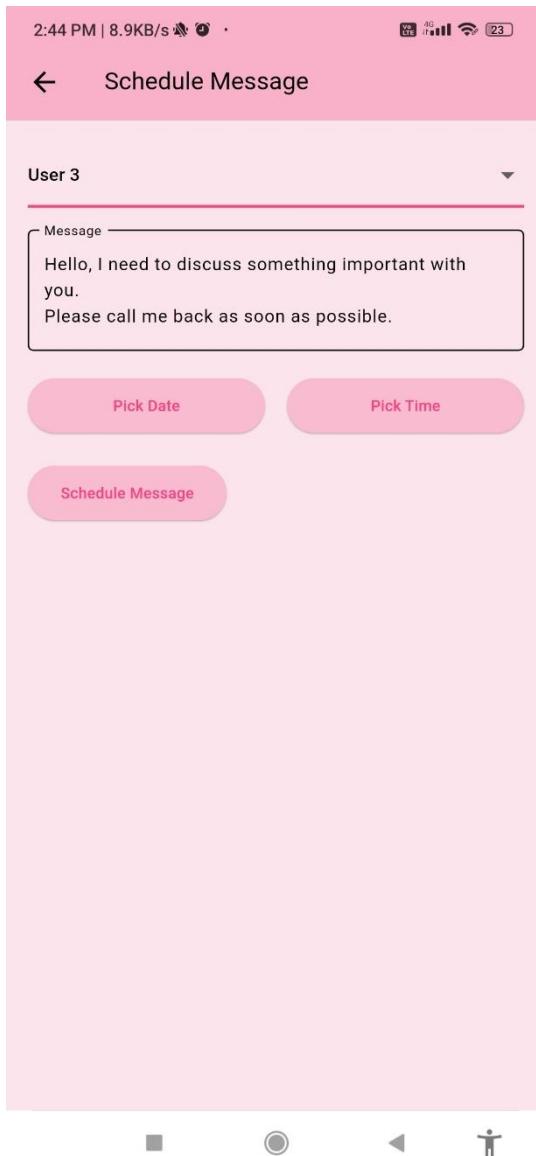


FIGURE B.8 FILE CONVERTER

This Figure B.7 Represents A pink-themed app for converting text to PDF or extracting text from PDF. Text box with a short story about a genie and wisdom..Three buttons: Generate PDF, Preview Text, and PDF to Text.



**FIGURE B.9 SCHEDULED
MESSAGE**

This Figure B.8 Represents Message Scheduling Interface: A pink-themed app screen for scheduling messages. Message Input: A text box contains a sample message asking for a callback.

REFERENCES

- [1] Rap Payne(2019),”Beginning App Development with Flutter: Create Cross-Platform Mobile Apps 1st ed. Edition”.
- [2] Thomas Bailey ,Alessandro Biessek(2021) ,”Flutter for Beginners - Second Edition: An introductory guide to building cross-platform mobile applications with Flutter 2.5 and Dart 2nd ed. Edition”.
- [3] Carmine Zaccagnino(2020),”Programming Flutter”.
Eric Windmill (2019),”Flutter in Action 1st Edition”.
- [4] https://github.com/flyerhq/flutter_firebase_chat_core.
- [5] <https://github.com/ArunBalajiR/Flutter-Chat-Application>.

