iHUB DivyaSampark

@IIT Roorkee

Project Title: Making Your WhatsApp Conversations Smarter with Intelligent Reminders

Month 1: Report

Initiation, Requirement Analysis, Tech Stack

Mentor:

Dr M N Thippeswamy, Head and Professor

Department Of CSE(AI&ML) and CSE(Cyber Security) Engineering

Submitted By:

Anu S M (UG 211)

Akshay S (UG 212)

Venkatesh R (UG 213)

1. Initiation

In today's digital age, the sheer volume of information exchanged on platforms like WhatsApp, Messages, and Email creates a unique challenge known as information overload. As users engage in a variety of conversations, ranging from personal discussions to professional collaborations, critical details embedded in messages, such as event dates, locations, and times, can easily get lost amidst the digital noise. This problem is particularly prevalent in WhatsApp, where conversations can pile up, leading to confusion and reduced productivity.

To address this issue, we introduce "TimeCraft (Making WhatsApp Conversations Smart with Intelligent Reminders)", a system designed to extract dates, times, and locations from WhatsApp messages and automatically set reminders before the event. Leveraging the power of cross-platform frameworks like Flutter and backend services such as Firebase, TimeCraft uses Natural Language Processing (NLP) models like HuggingFace to detect entities in the messages. These detected entities are then stored in the Cloud Firestore database and sent back to the application as reminders. The system ensures that no crucial information gets lost in the digital sea, enhancing user productivity and improving overall communication efficiency

Problem Statement

- The sheer volume of information exchanged on WhatsApp, Messages, and Email poses a unique challenge- Information overload.
- As users engage in diverse conversations, from personal discussions to professional collaborations, the critical details embedded in messages, such as event dates, locations, and times, can easily get lost amidst the digital noise.

Problem Solution

Introducing “TimeCraft(Making Whatsapp Conversations Smart with Intelligent Reminders)," a system that extracts dates, times, and locations from WhatsApp messages and automatically sets reminders before the event.

Project Scope:

- **Inclusions:**
    - Develop an app that allows users to log in with their phone numbers, extract event details from WhatsApp messages, store them in a database, and receive timely reminders.
- **Exclusions:**

- Advanced features such as integrations with other messaging platforms could be considered in future iterations.

Stakeholders

Identified Stakeholders: Users (event attendees), Developers, and Administrators.

2. Requirement Analysis

2.1 User Stories:

Example User Stories:

- As a user, I want to log in with my phone number to access the app.
- As a user, I want to view upcoming events and their details.
- As a user, I want to edit the details of an event.

2.2 Functional Requirements:

Key Functionalities:

- User authentication via phone number.
- Message extraction for event details (type, name, date, time).
- Database storage of events.
- Viewing and editing events.
- Reminder notifications.

2.3 Non-functional Requirements:

- **Performance:**
  - The app should respond within a reasonable time frame.
- **Security:**
  - User data, including phone numbers and event details, must be securely stored and transmitted.
- **Usability:**
  - The user interface should be intuitive and user-friendly.

2.4 Use Case Diagram

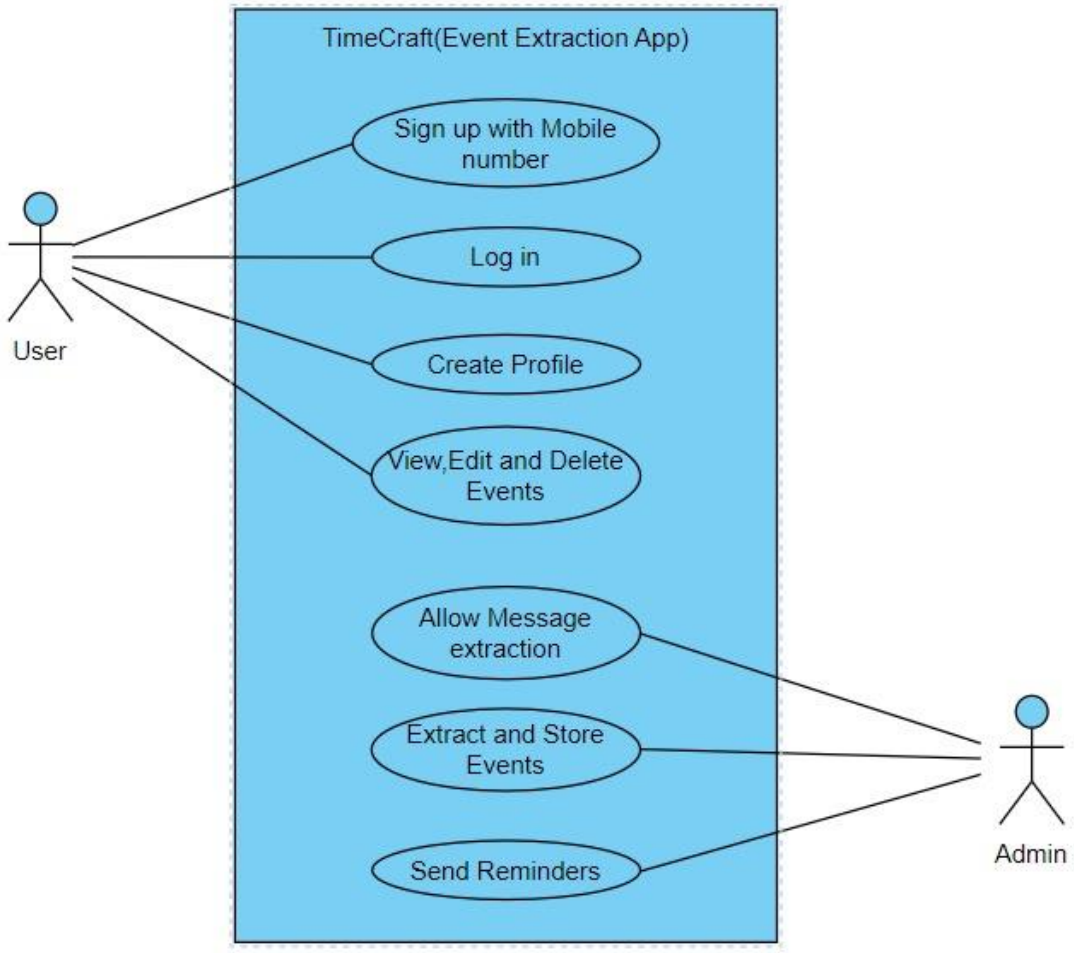

Fig 1: Use Case Diagram

2.5 Data Requirements:

- **Entities:**
  - User (phone number, profile information)
  - Event (type, name, date, time)
- **Database Structure:**
  - Users Table (user_id, phone_number, ...)
  - Events Table (event_id, user_id, type, name, date, time, ...)

2.6 Sample User Interface of the app

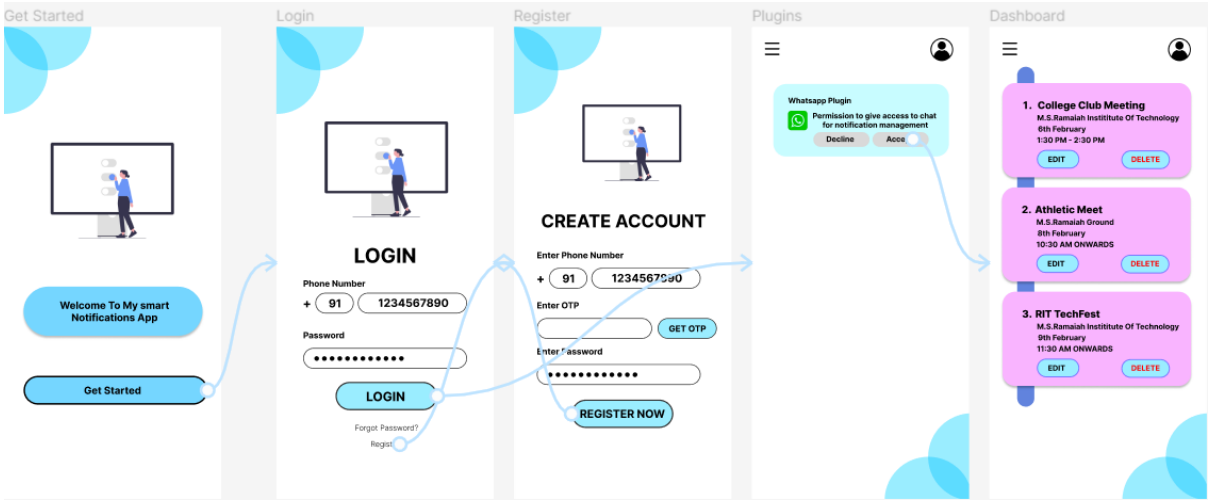

Fig 2: a)Get Started b)Login 3)Register 4)Plugins 5)Dashboard

2.7 Architecture Design

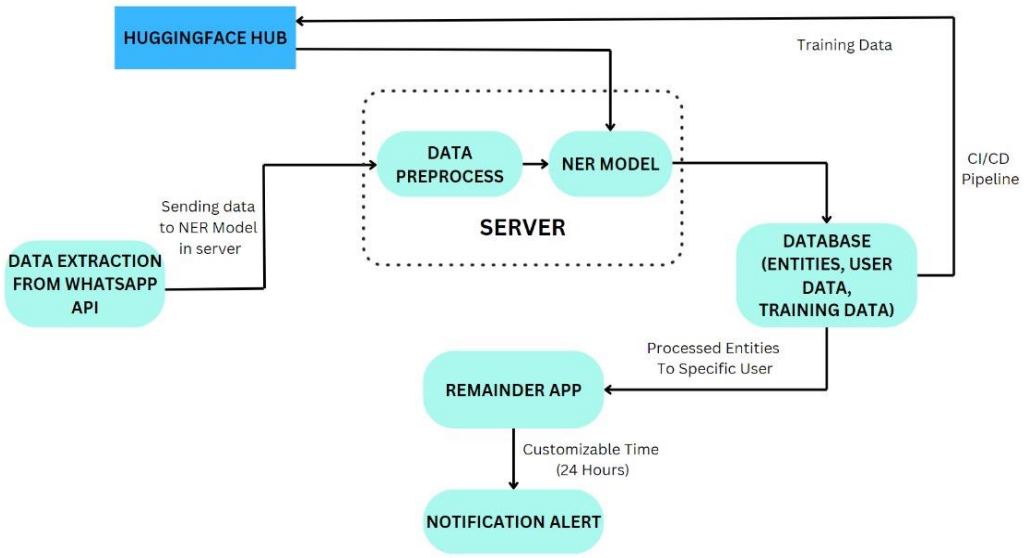

Fig 3: Architecture Design

Data Collection: The app collects WhatsApp messages via its API. This requires user permission and must comply with WhatsApp's terms of service.

Data Preprocessing: The collected data is preprocessed to ensure it is in a suitable format for further processing. This could involve cleaning the data and converting it into a structured format.

Entity Extraction: The preprocessed data is then passed to a Natural Language Processing (NLP) model, specifically a Named Entity Recognition (NER) model, to extract relevant entities such as Event Name, Type of Event, dates, times, and locations. This can be achieved using libraries like HuggingFace. Optimization techniques like converting to ONNX environment and 8-bit Quantization are performed to reduce latency and deployment cost.

Data Storage: The extracted entities are then stored in a database, specifically Cloud Firestore. This allows for easy retrieval and manipulation of the data.

Reminder Setup: A reminder is set 24 hours prior to the extracted date and time. This can be achieved using Firebase Cloud Messaging (FCM).

User Interface: The user interface of the app, built using Flutter, displays the reminders to the user. The reminders are fetched from the database and displayed in a user-friendly manner.

Methodology

- Start
- Initialize Firebase
- Set Up Firebase Authentication (Phone Number)
- Set Up Cloud Firestore for User and Event Data
- Set Up Cloud Functions for Serverless Backend Logic
- Set Up HuggingFace(or other suitable NLP library) for Named Entity Recognition (NER)
- Set Up Firebase Cloud Messaging (FCM) for Sending Reminder Notifications
- Develop App in Flutter (using Dart)
- Data Extraction: Extract dates, times, and locations from WhatsApp messages
- Send Data to Server for NER Detection
- Store Output Entities in Cloud Firestore
- Send Data Back to App
- Present Entities as Reminders in App
- End

3. Tech Stack Collection:

3.1 Development Frameworks:

Flutter:

- Cross-platform framework for mobile app development.

3.2 Backend Services:

Firebase:

- Authentication: Firebase Authentication for phone number authentication.
- Database: Cloud Firestore for storing user and event data.
- Cloud Functions: For serverless backend logic.

3.3 Natural Language Processing (NLP) Model:

Suitable NLP library like HuggingFace

3.4 Notification Service:

Firebase Cloud Messaging (FCM):

- For sending reminder notifications.

3.5 Authentication:

Firebase Authentication:

- Phone number authentication.

3.6 IDE and Version Control:

IDE:

- Visual Studio Code.

Version Control:

- Git for version control.

3.7 Programming Language:

Dart:

- Programming language for Flutter development.