

WorkEase

An Agentic Unified Intelligent Multi-Platform Communication System

Project Team

Hasnain Saleem 22P-9123
Alishba Tariq 22P-9112
Kashan Saeed 22P-9128

Session 2022–2026

Supervised by

Dr. Nouman Azam (Associate Professor)



Department of Computer Science

**National University of Computer and Emerging Sciences
Peshawar, Pakistan**

October, 2025

Student's Declaration

We declare that this project titled "*WorkEase An Agentic Unified Intelligent Multi-Platform Communication System*", submitted as requirement for the award of degree of Bachelors in Bachelor of Science in Computer Science, does not contain any material previously submitted for a degree in any university; and that to the best of our knowledge, it does not contain any materials previously published or written by another person except where due reference is made in the text.

We understand that the management of Department of Computer Science, National University of Computer and Emerging Sciences, has a zero tolerance policy towards plagiarism. Therefore, We, as authors of the above-mentioned thesis, solemnly declare that no portion of our thesis has been plagiarized and any material used in the thesis from other sources is properly referenced.

We further understand that if we are found guilty of any form of plagiarism in the thesis work even after graduation, the University reserves the right to revoke our BS degree.

Hasnain Saleem

Signature: _____

Alishba Tariq

Signature: _____

Kashan Saeed

Signature: _____

Verified by Plagiarism Cell Officer

Dated:

Certificate of Approval



The Department of Computer Science, National University of Computer and Emerging Sciences, accepts this thesis titled *WorkEase An Agentic Unified Intelligent Multi-Platform Communication System*, submitted by Hasnain Saleem (22P-9123), Alishba Tariq (22P-9112), and Kashan Saeed (22P-9128), in its current form, and it is satisfying the dissertation requirements for the award of Bachelors Degree in Bachelor of Science in Computer Science.

Supervisor

Dr. Nouman Azam (Associate Professor)

Signature: _____

Sir Riaz Nawab
FYP Coordinator
National University of Computer and Emerging Sciences, Peshawar

Dr. Qasm Jan, Head of Computer Science Department

HoD of Department of Computer Science
National University of Computer and Emerging Sciences

Acknowledgements

We would like to express our deepest gratitude to our supervisor, **Dr. Nouman Azam**, for his continuous guidance, motivation, and valuable knowledge throughout the development of our Final Year Project. We also thank our department faculty and peers for their constructive feedback and encouragement. Lastly, we appreciate our families for their constant support and patience during the completion of this project.

Hasnain Saleem

Alishba Tariq

Kashan Saeed

Abstract

This project presents the development of an **Agentic Unified Automation System**, a voice-assisted intelligent desktop application designed to integrate communication, file management, and task automation across multiple platforms such as Gmail, Slack, and local Linux utilities. The system operates as an agentic OS overlay, combining multiple autonomous agents coordinated by a supervisor agent to execute user commands seamlessly.

The key problem addressed is the fragmentation of productivity tools that forces users to constantly switch contexts, reducing focus and efficiency. The proposed solution provides a unified, AI-powered workspace that automates workflows, summarizes messages, manages notifications, and allows contextual voice interaction. Using natural language processing, multi-agent coordination, and secure API integration, the system can read, summarize, and respond to messages; execute terminal commands; and trigger custom event-based notifications. Evaluation was performed through detailed test cases to validate accuracy, response time, and integration stability. The results demonstrate enhanced productivity, reduced cognitive load, and effective cross-platform communication through automation.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Project Overview | 1 |
| 1.2 | Problem Statement | 1 |
| 1.3 | Proposed Solution | 2 |
| 1.4 | Objectives | 2 |
| 2 | System Analysis and Design | 3 |
| 2.1 | System Architecture | 3 |
| 2.2 | Use Case Diagram | 4 |
| 2.3 | Activity Diagrams | 6 |
| 2.3.1 | Overall System Activity | 6 |
| 2.3.2 | Activity 1: Voice-Based Interaction And Control | 7 |
| 2.3.3 | Activity 2: Smart Email and Message Automation | 9 |
| 2.3.4 | Activity 3: Multi-App Integration and Authentication | 11 |
| 2.4 | Component Diagram | 12 |
| 3 | Implementation | 13 |
| 3.1 | Development Environment | 13 |
| 3.2 | Implemented Features | 13 |
| 4 | Testing and Evaluation | 15 |
| 4.1 | Test Cases – Part 1 | 15 |
| 4.2 | Test Cases – Part 2 | 16 |
| 4.3 | Testing Outcomes | 16 |
| 5 | Project Management (CLO4) | 17 |

| | | |
|----------|-----------------------------------|-----------|
| 5.1 | Timeline and Scheduling | 17 |
| 5.2 | Risk Management | 17 |
| 6 | Conclusion | 19 |
| | References | 19 |

List of Figures

| | | |
|-----|---|----|
| 2.1 | Use Case Diagram for WorkEase | 4 |
| 2.2 | Overall System Activity Diagram | 6 |
| 2.3 | Activity Diagram – Voice-Based Interaction And Control | 7 |
| 2.4 | Activity Diagram – Smart Email and Message Automation | 9 |
| 2.5 | Activity Diagram – Multi-App Integration and Authentication | 11 |
| 2.6 | Component Diagram of WorkEase System | 12 |

List of Tables

| | | |
|-----|---|----|
| 4.1 | Test Cases – Integration and Automation | 15 |
| 4.2 | Test Cases – Notifications, Security, and Reliability | 16 |

Chapter 1

Introduction

1.1 Project Overview

WorkEase is an intelligent desktop application that unifies multiple communication and productivity tools into a single workspace. It enables users to communicate, organize, and automate workflows using natural voice commands. The system integrates platforms such as Gmail, Slack, Discord, and file systems through secure APIs and runs natively on Linux. Its architecture is built around modular agents — each responsible for communication, file, or terminal tasks — coordinated by a central supervisor agent.

1.2 Problem Statement

Modern users struggle with managing multiple communication apps, notifications, and tasks simultaneously. This constant context switching causes inefficiency, distraction, and lost productivity. There is no unified, context-aware, and intelligent assistant that securely automates and integrates these tasks on a single desktop environment.

1.3 Proposed Solution

The proposed system introduces a **multi-agent intelligent platform** that integrates and automates workflows across Slack, Gmail, and local tools. It summarizes unread emails, manages Slack notifications, executes terminal commands, and triggers configurable alerts — all via natural speech or text input. The system learns user behavior and adapts accordingly.

1.4 Objectives

- Integrate multiple applications (Gmail, Slack, Discord, etc.) within one secure interface.
- Automate communication and task execution through voice commands.
- Generate contextual summaries and unified notifications.
- Provide configurable event-based alerts and reminders.
- Ensure data privacy with local storage and token-based access.

Chapter 2

System Analysis and Design

2.1 System Architecture

The system follows a **multi-agent architecture** where independent agents handle communication, file operations, or terminal commands, coordinated by a supervisor agent. This allows modular scalability, fault isolation, and efficient resource allocation.

2.2 Use Case Diagram

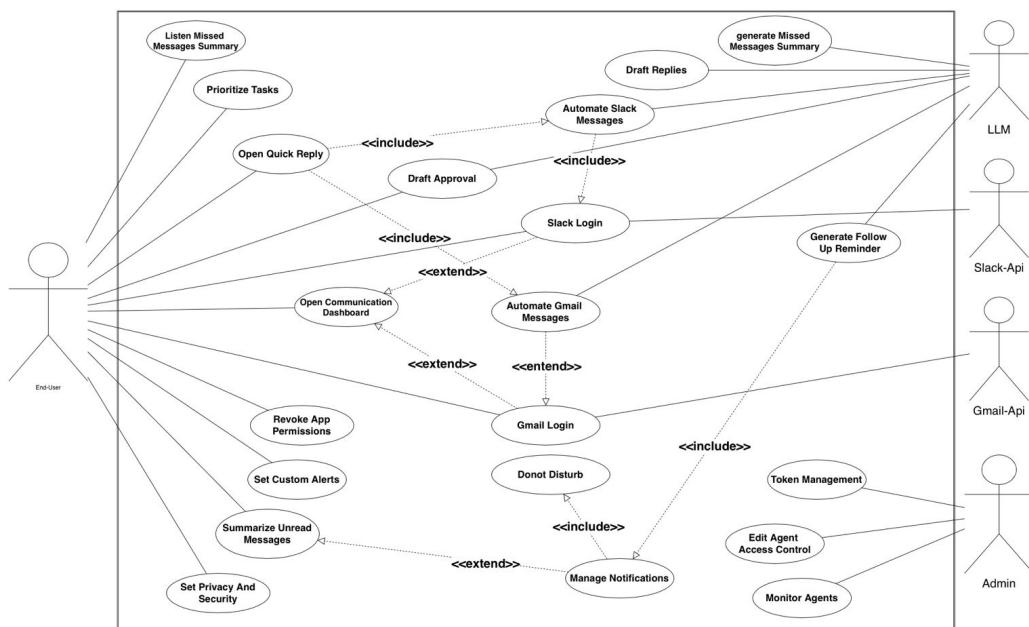
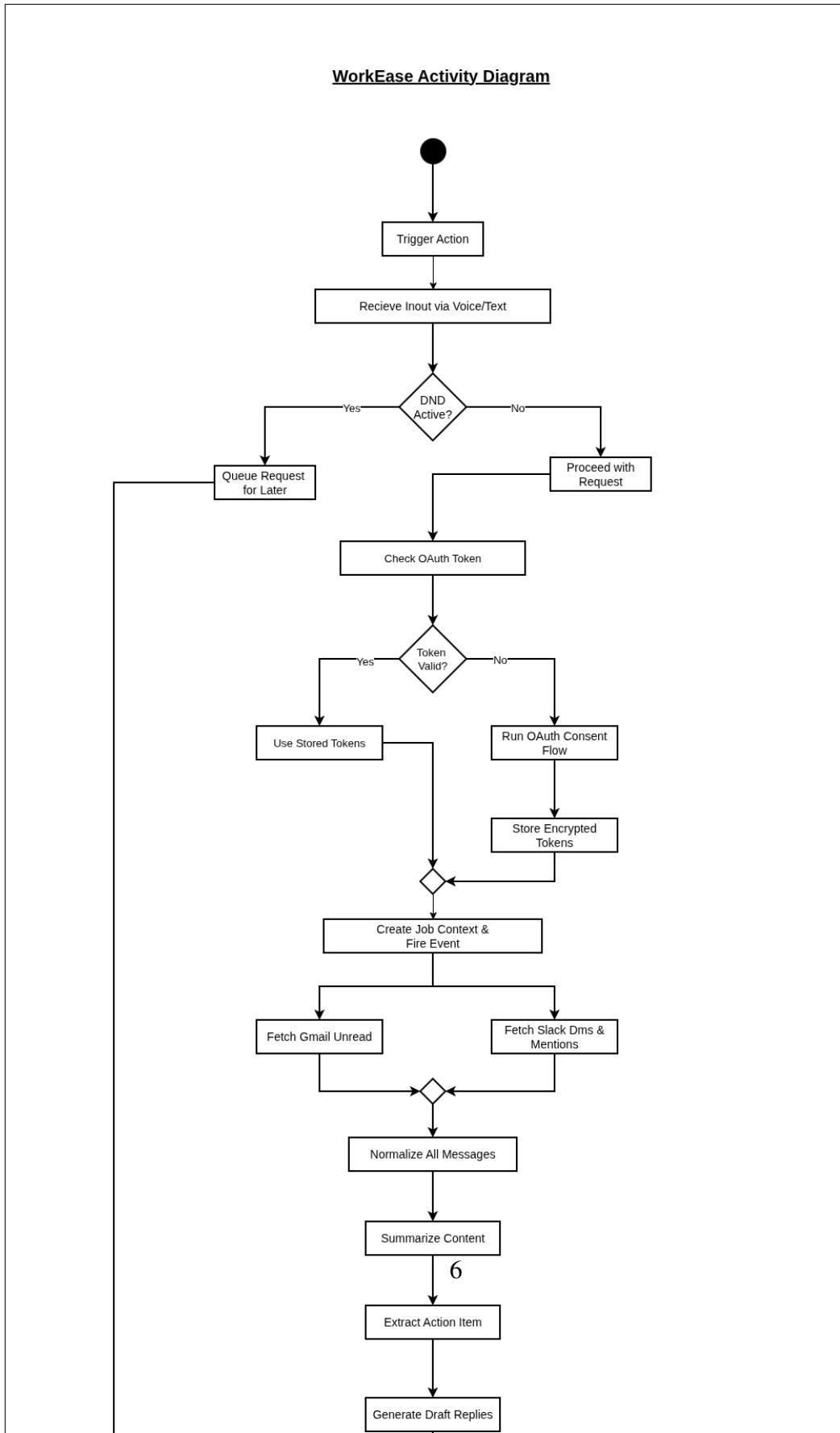


Figure 2.1: Use Case Diagram for WorkEase

2.3 Activity Diagrams

2.3.1 Overall System Activity



2.3.2 Activity 1: Voice-Based Interaction And Control

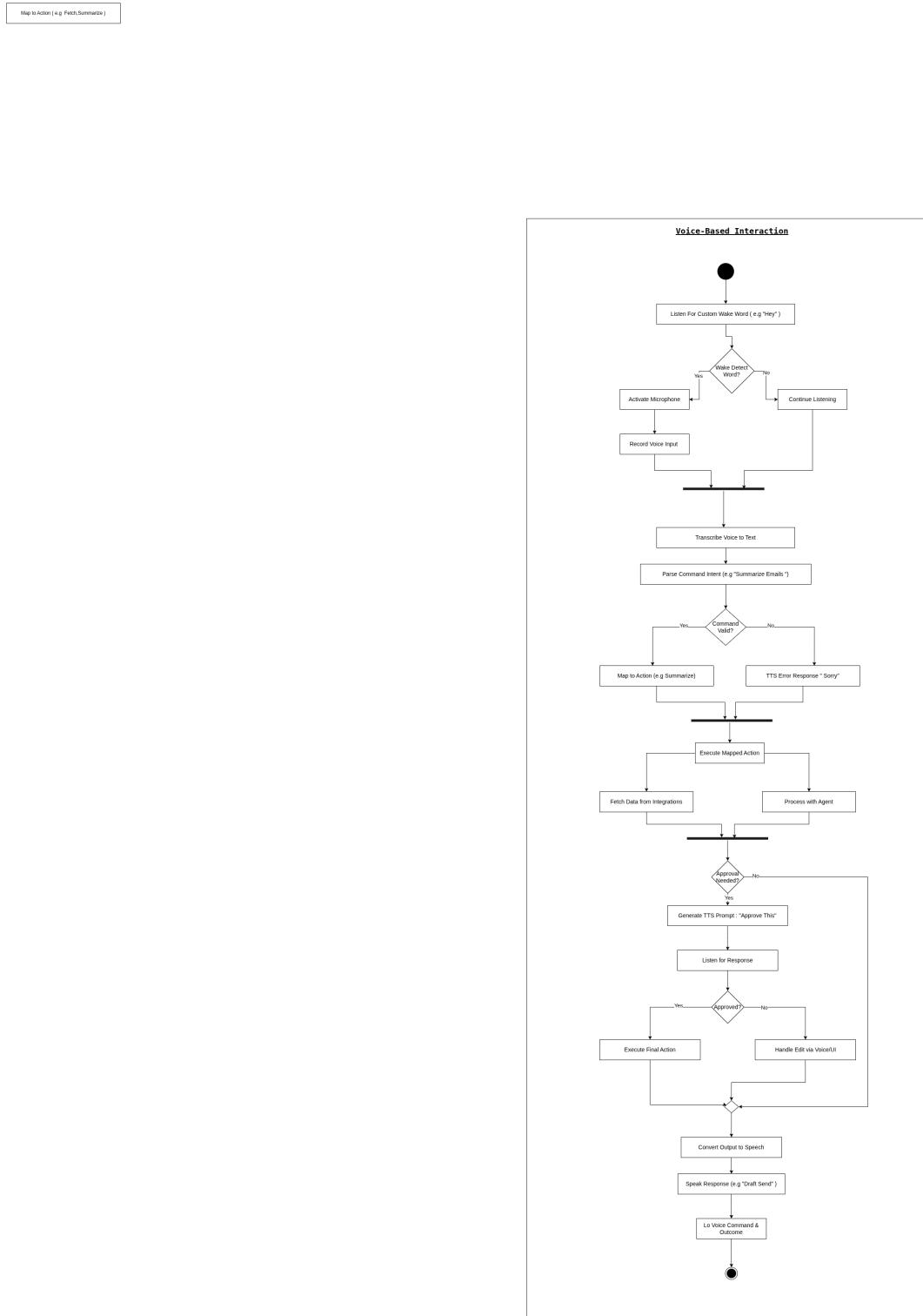
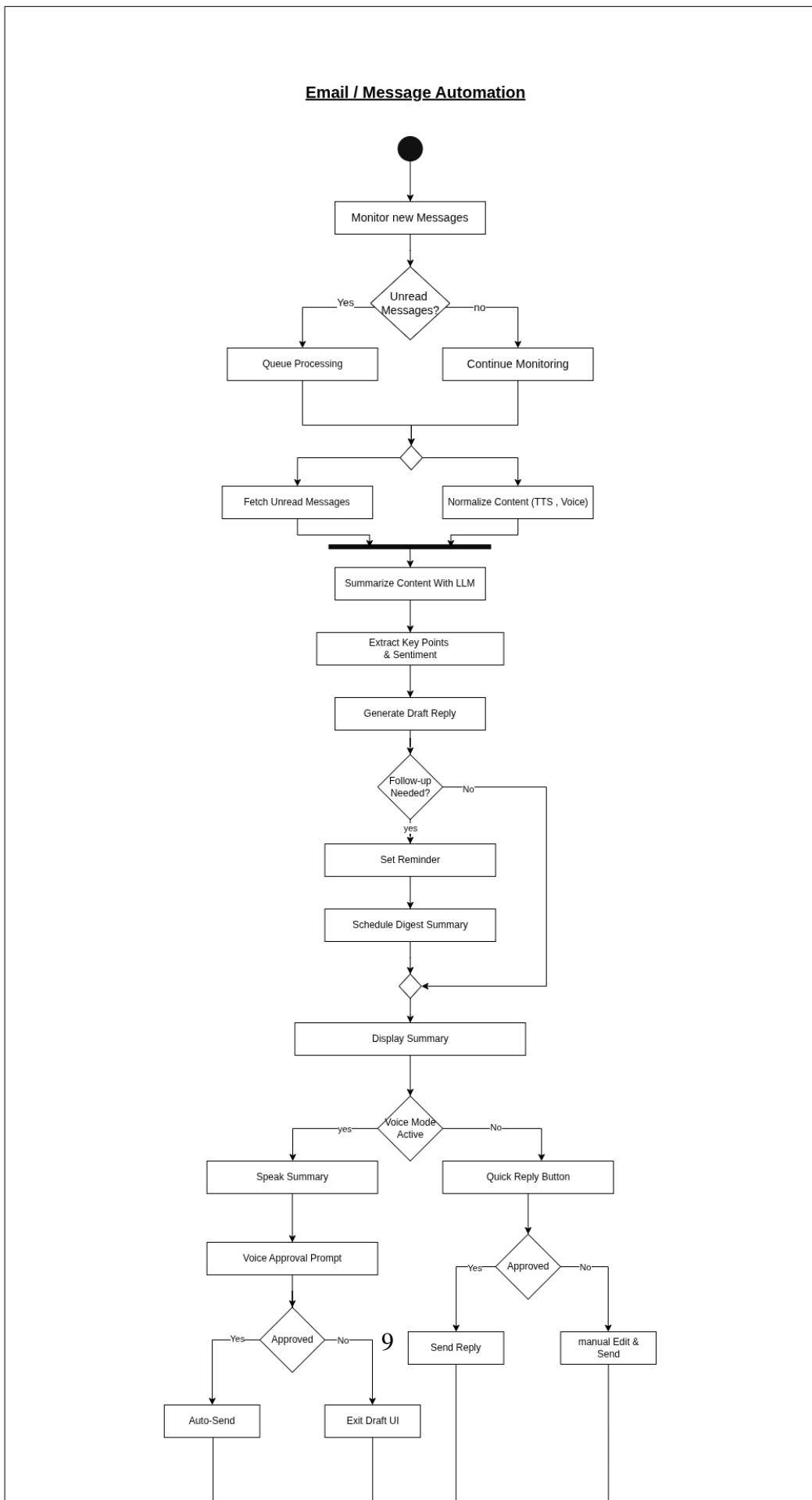
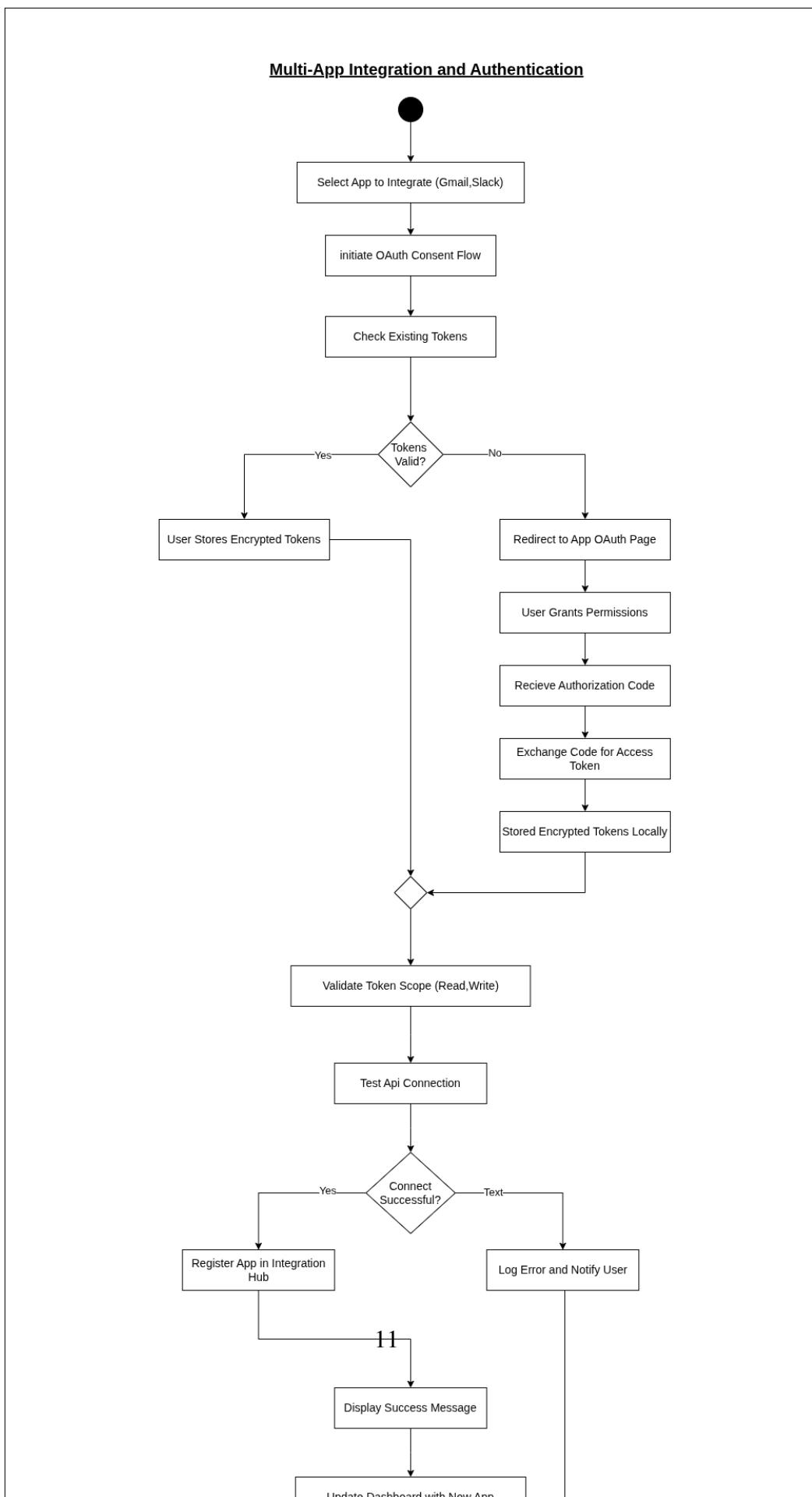


Figure 2.3: Activity Diagram – Voice-Based Interaction And Control

2.3.3 Activity 2: Smart Email and Message Automation



2.3.4 Activity 3: Multi-App Integration and Authentication



2.4 Component Diagram

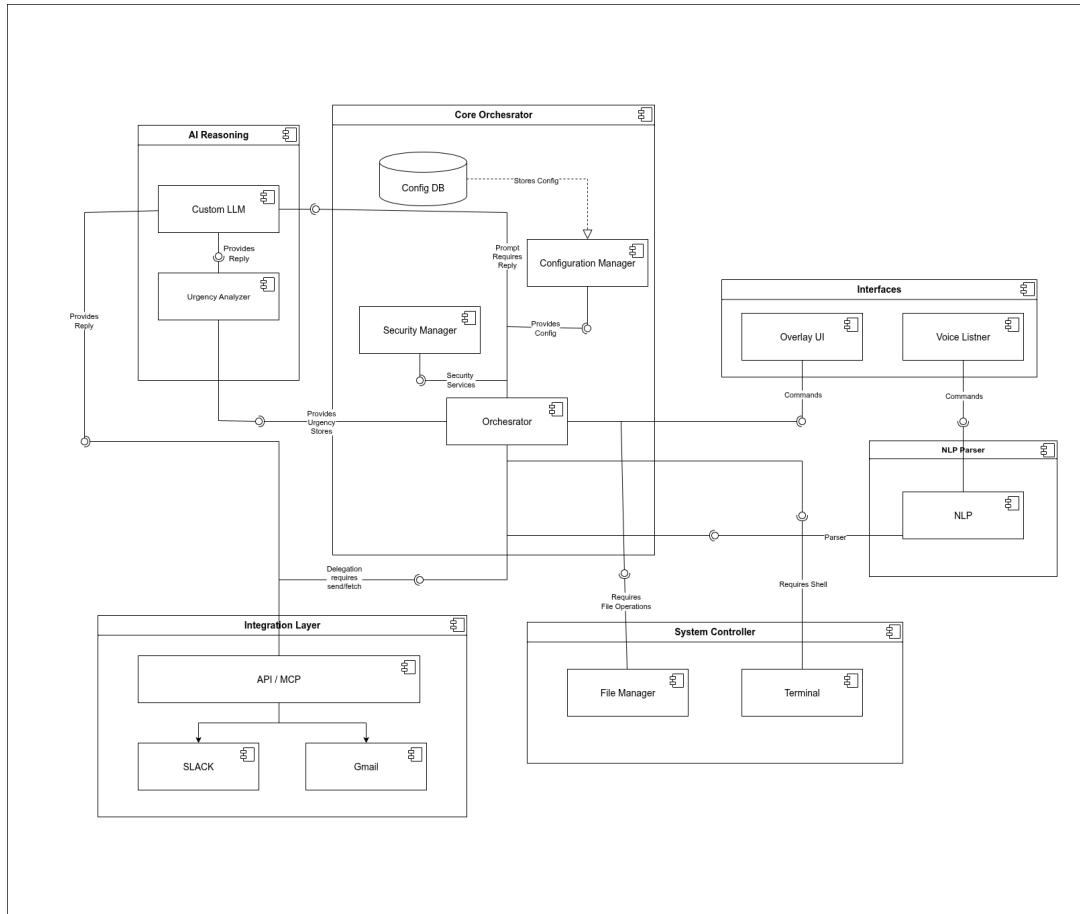


Figure 2.6: Component Diagram of WorkEase System

Chapter 3

Implementation

3.1 Development Environment

- **Language:** Python (FastAPI, LangChain)
- **OS:** Linux (Ubuntu)
- **Database:** SQLite / JSON-based storage
- **Frontend:** Electron + React
- **AI Models:** Whisper (STT), Mistral (LLM), Coqui (TTS)

3.2 Implemented Features

1. Unified communication dashboard (Slack + Gmail)
2. Auto email summarization and voice drafting
3. Task extraction and prioritization
4. Configurable alerts and quiet hours
5. Secure OAuth authentication and local token management

Chapter 4

Testing and Evaluation

4.1 Test Cases – Part 1

| ID | Scenario | Objective | Pre-conditions | Test Steps | Test Data | Expected Result |
|--------------|------------------------------|---------------------------------------|----------------------|---------------------------------|------------------------|---------------------------------|
| TC_INT_01 | Verify multi-app integration | Ensure APIs connect successfully | Valid tokens | Connect apps and fetch messages | OAuth keys | Unified inbox retrieved |
| TC_AUTO_01 | Auto email summarization | Verify NLP summaries | Gmail linked | Run summarization | Sample emails | Concise summaries shown |
| TC_TASK_01 | Task extraction | Ensure tasks identified | Slack + Gmail active | Run task extractor | Message text | Prioritized task list generated |
| TC_NOTIF_01 | Configurable alerts | Validate custom notifications | Event rule set | Trigger alert condition | “Interview call” email | Notification triggered |
| TC_DASH_01 | Unified inbox display | Verify dashboard merges notifications | Apps linked | Open dashboard | Message events | Combined view accurate |
| TC_STATUS_01 | Slack status voice control | Check voice-based status update | Mic active | Say “Mute Slack 1h” | Voice input | Slack muted for set time |

Table 4.1: Test Cases – Integration and Automation

4.2 Test Cases – Part 2

| ID | Scenario | Objective | Pre-conditions | Test Steps | Test Data | Expected Result |
|----------------|-----------------------|------------------------------------|-----------------------|------------------------------|--------------------------|-----------------------|
| TC_REMIND_01 | Follow-up reminder | Ensure reminders trigger correctly | Gmail + Slack active | Set reminder, simulate delay | Reminder = 3h | Alert generated |
| TC_SENTI_01 | Sentiment analysis | Validate emotion/urgency detection | NLP model active | Analyze strong message tone | “URGENT! Project failed” | Marked as urgent |
| TC_LOG_01 | Audit trail logging | Ensure action logs recorded | Admin access | Perform actions, open log | Log entries | Accurate timestamps |
| TC_ACCESS_01 | Access revocation | Check permission removal | Active session | Revoke Gmail access | Gmail token | Access denied message |
| TC_OFFLINE_01 | Offline functionality | Validate limited offline support | Cached data available | Disconnect, run summary | Cached threads | Summary shown locally |
| TC_SECURITY_01 | Token encryption | Verify secure local storage | Tokens issued | Inspect encrypted store | Access tokens | Encrypted securely |

Table 4.2: Test Cases – Notifications, Security, and Reliability

4.3 Testing Outcomes

All modules passed functional testing. Minor issues (API rate limits, occasional STT inaccuracies) were mitigated with retry logic and confidence thresholds.

Chapter 5

Project Management (CLO4)

5.1 Timeline and Scheduling

- **Phase 1:** Requirements & Design (Weeks 1–4)
- **Phase 2:** Core Development (Weeks 5–9)
- **Phase 3:** Integration & Testing (Weeks 10–13)
- **Phase 4:** Documentation & Presentation (Weeks 14–16)

5.2 Risk Management

- **R1:** API token expiry → Mitigation: auto-refresh mechanism.
- **R2:** Voice recognition errors → Mitigation: adaptive thresholds.
- **R3:** Integration failures → Mitigation: retry and logging systems.

Chapter 6

Conclusion

The **WorkEase** system demonstrates how intelligent automation and multi-agent coordination can unify productivity platforms into a seamless, privacy-first workspace. With successful Slack–Gmail integration, voice automation, and contextual task handling, it significantly reduces manual workload. Future improvements include integrating more third-party APIs, advanced on-device LLMs, and community-driven agent extensions.