

Faculty of Engineering

Department of Informatics Engineering

Software and Information System Engineering



Agile-Focused AI Note-Taking App

(AgileMeets)

A senior 2 project report - submitted to complete the requirements for
obtaining a bachelor's degree in informatics engineering - Software
Engineering and Information Systems

Prepared by

Raghad al-hossny Mohamed Al Balkhi

Supervised by

Dr. Eng. Riad Sonbol Eng. Anas Abdulaziz

February 2025

Agile-Focused AI Note-Taking App (AgileMeets)

مشروع (تخرج 2) قدم إستكمالاً لمتطلبات الحصول على درجة
البكالوريوس في هندسة المعلوماتية- هندسة البرمجيات ونظم المعلومات

إعداد

راغد عمار الحصني محمد أحمد البلخي

إشراف

د. رياض سنبل م. أنس عبد العزيز

شباط 2025

SUPERVISION CERTIFICATION

Supervisor Certification

I Certify that the preparation of this project entitled [**Agile-Focused AI Note-Taking App**]

Prepared by [**Raghad al-hossny & Mohamed Al Balkhi**] was made under my supervision at Faculty of Informatics Engineering in partial Fulfillment of the Requirements for the Degree of Bachelor of Software and Information system Engineering

Name:

Signature:

الاهداء (رغم)

الاهداء (محمد)

RAGHAD'S ACKNOWLEDGEMENTS

الحمدُللهُ الذي هيأَ البدء ويسِّرَ الطريق وطَيِّبَ المنهى، الحمدُللهُ على التمام وحسنِ الخاتمة

*First of all, I would like to thank my supervisor, **Dr. Riad Sonbol**, for his constant support, guidance, and encouragement throughout this journey.*

Over three semesters and two projects, he has inspired me to grow and strive for excellence, I am deeply grateful for his impact on my academic and personal development....

If I ever become a lecturer, you will be the first model I aim to follow.

Dr. Mouhib al noukari - Dean of the College

Thank you doctor for every moment of support, encouragement, and trust you've given me. I am truly grateful for all the hard work you've dedicated to us, from the very beginning until the end...

For the best project manager, engineer, therapy for team internal conflicts :)

Eng. Anas Abdulaziz

Thank you for all the efforts you've made for us...

*And finally, to my partner, **Mohamed Al-Balkhi**, I wish you a future filled with success and achievement. Thank you for your hard work.*

MOHAMED'S ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to everyone who has supported me throughout my academic journey at the college.

First and foremost, I am deeply grateful to:

- Dr. **Mouhib Alnoukari**, Dean of the College
- Dr. **Wissam Al-Khatib**, Vice Dean

Their inspiring leadership has created an environment that fosters growth and excellence.

Special thanks go to:

- Dr. **Riad Sonbol**
- Engineer **Anas Abdulaziz**

Their mentorship and guidance have been invaluable and a constant source of motivation.

I also sincerely appreciate the esteemed faculty members who have shaped my academic experience:

- Dr. **Tharaa Aslan**
- Dr. **Rose Mohammed**
- Dr. **Akram Massouh**
- Dr. **Majida Al-Bakour**
- Dr. **Nisreen Suleiman**
- Engineer **Maher Sarem**
- Engineer **Aya Al-Aswad**
- Engineer **Marwa Al-Safadi**

Additionally, I would like to thank:

- Miss **Hala Mizhir** for her support and kindness in Physics labs.
- Miss **Hidy Eid** for her administrative hard work in the college.

Finally, My project partner, **Raghad Al-Hossny**, for your collaboration and dedication. It has been a pleasure knowing & working with you.

And to **everyone** who was once a **teacher, colleague** or an **employee** at **SPU** This journey would not have been the same without your contributions, and I am truly gra

ABSTRACT

In today's software engineering landscape, Agile methodologies are among the most widely used and adopted frameworks for developing software systems. Agile emphasizes flexibility, iterative development, and continuous collaboration, making it an essential approach for modern software teams. One of its core principles is frequent communication, which leads to numerous meetings. These meetings produce valuable artifacts that influence the project's Progression, execution, and planning.

However, managing and documenting these discussions manually is time-consuming and inefficient. In response to this challenge, this report presents an AI-powered software platform designed for Agile software teams and organizations. The system enables automated transcription, summarization, and key point extraction using AI techniques to ensure that all valuable meeting insights are accurately captured and easily accessible.

The platform processes both online and in-person meetings and provides other valuable features like role-based accessing, updating, and managing project requirements, allowing teams to track evolving needs effectively. By delivering a comprehensive project vision, the platform enhances decision-making and collaboration within Agile teams.

ملخص

في عالم هندسة البرمجيات الحديثة، تُعد المنهجيات الرشيقه (Agile) من أكثر الأطر اعتماداً في تطوير الأنظمة البرمجية نظراً لما تتوفره من مرونة تطوير وتسليم تكراري، وتعاون مسقٍ. تعقد Agile على التواصل المتكرر بين الفرق وأصحاب المصلحة، من خلال عقد العديد من الاجتماعات التي تُنتج مخرجات جوهرية تؤثر على مسار المشروع وتنفيذها وتخطيشه.

ومع ذلك، فإن إدارة هذه الاجتماعات وتوثيق ما يدور فيها يدوياً يعد مهمة مرهقة، تستنزف الوقت، وعرضة للأخطاء، مما قد يؤدي إلى فقدان معلومات مهمة أو سوء تفسيرها. ولمعالجة هذه التحديات، يقدم هذا التقرير منصة برمجية مدعاومة بالذكاء الاصطناعي، مصممة خصيصاً للمؤسسات والفرق البرمجية التي تعتمد Agile.

تمكن هذه المنصة الفرق من تحويل محتوى الاجتماعات تلقائياً إلى نصوص مكتوبة، واستخراج النقاط الأساسية، وإنشاء ملخصات دقيقة باستخدام تقنيات الذكاء الاصطناعي. يضمن هذا النظام توثيقاً فورياً وفعلاً لكافة المناقشات، مما يسمح للفرق بالوصول إلى المعلومات بسهولة، وتحليلها دون الحاجة إلى التوثيق اليدوي المرهق.

تدعم المنصة اجتماعات الفرق سواء كانت افتراضية أو حضورية، كما توفر مجموعة من الميزات المتقدمة، مثل إدارة الصالحيات بناءً على الأدوار، وعرض وتحديث متطلبات المشروع، وتوثيق القرارات، مما يساعد الفرق على متابعة التغييرات في المشروع بمرتبة وكفاءة.

من خلال تقديم رؤية شاملة للمشروع في كل مراحله، تساهم المنصة في تحسين عملية اتخاذ القرار، وتعزيز التعاون بين الفرق، وزيادة الإنتاجية داخل بيئة التطوير البرمجية.

تم بناء النظام وفق منهجيات ومارسات هندسة البرمجيات، بالاستعانة بتقنيات برمجية عدّة بالإضافة إلى بعض تقنيات الذكاء الاصطناعي وغيرها من التقنيات للحصول على النتيجة المراده.

TABLE OF CONTENT

<i>SUPERVISION CERTIFICATION.....</i>	<i>iii</i>
الاعلاء (رغد).....	iv
الاعلاء (محمد).....	v
<i>RAGHAD'S ACKNOWLEDGEMENTS.....</i>	<i>vi</i>
<i>MOHAMED'S ACKNOWLEDGEMENTS.....</i>	<i>vii</i>
<i>ABSTRACT.....</i>	<i>viii</i>
ملخص.....	ix
<i>TABLE OF CONTENT</i>	<i>x</i>
<i>List of tables.....</i>	<i>xiii</i>
<i>List of Figures</i>	<i>xiv</i>
<i>List of abbreviations.....</i>	<i>xvii</i>
<i>Chapter1 Introduction.....</i>	<i>18</i>
1. <i>Introduction:.....</i>	<i>19</i>
2. <i>Problem Definition:</i>	<i>19</i>
3. <i>Project objectives:.....</i>	<i>20</i>
4. <i>Report organization:</i>	<i>21</i>
<i>Chapter 2 project management and initialization.....</i>	<i>22</i>
1. <i>Introduction:.....</i>	<i>23</i>
2. <i>Project management documents:.....</i>	<i>23</i>
3. <i>System initial analysis and design:</i>	<i>28</i>
4. <i>Project development life cycle model:.....</i>	<i>34</i>
5. <i>Summary:</i>	<i>37</i>
<i>Chapter 3 Base Management Service.....</i>	<i>38</i>
1. <i>Introduction:.....</i>	<i>39</i>
2. <i>Service Detailed analysis:.....</i>	<i>39</i>

3. Service design and architecture:	71
4. Service practical implementation:.....	85
<i>Chapter4 Notification Service</i>	115
1. Introduction.....	116
2. Service Analysis.....	116
3. Design & Implementation Considerations	123
4. Future Considerations.....	126
<i>Chapter5 Meeting Service</i>	127
1. Introduction.....	128
2. Required theory and basic knowledge	128
3. Available Solutions.....	132
4. Actual Microservice Analysis.....	138
5. Design & Integration Considerations	145
6. Screenshots of Client App	152
7. Future Considerations.....	165
8. Conclusion	165
<i>Chapter 6 AI Service</i>	166
1. Introduction:.....	167
2. Basic Concepts and literature review:	168
3. Service Design and implementation:.....	186
4. AI Service Testing:	196
<i>Chapter 7 Deployment</i>	204
1. Introduction.....	205
2. Deployment Architecture	205
3. Cloud Providers and Services Used.....	208
4. Performance Metrics and Load Testing	210
5. Future Considerations.....	213
<i>Chapter 8 Conclusion</i>	216

1. <i>Introduction</i>	217
2. <i>Putting it All Together</i>	217
3. <i>Lessons Learnt</i>	218
4. <i>Future Recommendations</i>	218
<i>Appendices</i>	219
<i>References</i>	220

List of tables

<i>Table 1 table of abbreviations.....</i>	<i>xvii</i>
<i>Table 2 Risk management.....</i>	<i>27</i>
<i>Table 3 high level project requirements.....</i>	<i>29</i>
<i>Table 5 base management service detailed requirements</i>	<i>39</i>
<i>Table 6 sign-up flow specification.....</i>	<i>45</i>
<i>Table 7 manage members specification</i>	<i>48</i>
<i>Table 8 manage projects specification.....</i>	<i>50</i>
<i>Table 9 manage project requirements specification.....</i>	<i>53</i>
<i>Table 10 base management - test cases definition</i>	<i>59</i>
<i>Table 11 base management RTM - version 01</i>	<i>69</i>
<i>Table 12 test cases execution.....</i>	<i>102</i>
<i>Table 13 Notification Service Detailed Requirements.....</i>	<i>116</i>
<i>Table 14 Notifications - test cases definition</i>	<i>121</i>
<i>Table 15 Different Online Meeting Solutions Comparison</i>	<i>135</i>
<i>Table 16 Meeting Detailed Requirements</i>	<i>138</i>
<i>Table 17 Test Scenario Meeting Creation</i>	<i>143</i>
<i>Table 18 Test Scenario Meeting Attendance.....</i>	<i>144</i>
<i>Table 19 Test Scenario Online Meeting</i>	<i>144</i>
<i>Table 20 Test Scenario Meeting Status Transitions</i>	<i>145</i>
<i>Table 21 whisper vs. Nova 2.....</i>	<i>175</i>
<i>Table 22 STT Case 01 non-native speaker.....</i>	<i>176</i>
<i>Table 23 STT Case 02 native speakers.....</i>	<i>177</i>
<i>Table 24 STT Case 03 Arabic Native speaker.....</i>	<i>178</i>
<i>Table 25 Rouge Metrix</i>	<i>184</i>
<i>Table 26 human evaluation – summarization models.....</i>	<i>185</i>
<i>Table 27 performance testing- summarization models</i>	<i>185</i>
<i>Table 28 AI service test1 performance results</i>	<i>197</i>
<i>Table 29 AI service test1 actual results.....</i>	<i>198</i>
<i>Table 30 AI service test 02 performance results</i>	<i>201</i>
<i>Table 31 AI service test 02 actual results.....</i>	<i>202</i>

List of Figures

<i>Figure 1 high level use case diagram</i>	28
<i>Figure 2 initial system decomposition</i>	33
<i>Figure 3 used software development model</i>	35
<i>Figure 4 Gantt chart</i>	36
<i>Figure 5 use case diagram</i>	44
<i>Figure 6 sign-up flow</i>	46
<i>Figure 7 Sign-up sequence diagram</i>	47
<i>Figure 8 manage member sequence diagram</i>	49
<i>Figure 9 manage member activity diagram</i>	50
<i>Figure 10 create project sequence diagram</i>	52
<i>Figure 11 update project sequence diagram</i>	52
<i>Figure 12 requirements management sequence diagram</i>	54
<i>Figure 13 manage privileges sequence diagram</i>	56
<i>Figure 14 entity relationship diagram</i>	57
<i>Figure 15 class diagram</i>	58
<i>Figure 16 Base management service architecture</i>	73
<i>Figure 17 CQRS flow</i>	76
<i>Figure 18 strategy design pattern</i>	79
<i>Figure 19 Factory design pattern</i>	81
<i>Figure 20 privileges design</i>	82
<i>Figure 21 data modeling</i>	84
<i>Figure 22 forget password interface</i>	87
<i>Figure 23 log-in interface</i>	88
<i>Figure 24 sign up -step 01 interface</i>	88
<i>Figure 25 sign-up step 02 interfaces</i>	88
<i>Figure 26 Registration step3 screen</i>	89
<i>Figure 27 mobile app - main page</i>	90
<i>Figure 28 manage org. members interface</i>	91
<i>Figure 29 Organization overview</i>	91
<i>Figure 30 projects management interface</i>	92
<i>Figure 31 manage project members interface</i>	92

<i>Figure 32 add requirement manually- interface.....</i>	93
<i>Figure 33 view project backlog - interface.....</i>	93
<i>Figure 34 project interface.....</i>	94
<i>Figure 35 add requirements from file</i>	94
<i>Figure 36 account management interface.....</i>	95
<i>Figure 37 manage privileges interfaces</i>	95
<i>Figure 38 Notifications Class Diagram.....</i>	119
<i>Figure 39 Notification Sequence Diagram</i>	120
<i>Figure 40 Notification Service Architecture</i>	123
<i>Figure 41 WebRTC Architecture</i>	129
<i>Figure 42 RTC Implementation Example</i>	130
<i>Figure 43 Media Codec & Compression</i>	130
<i>Figure 44 Famous Codecs Comparison.....</i>	131
<i>Figure 45 Twilio Video Pricing Model</i>	133
<i>Figure 46 OpenVidu Single Node.....</i>	137
<i>Figure 47 OpenVidu internal architecture & Components.....</i>	138
<i>Figure 48 Meetings Use case Diagram.....</i>	142
<i>Figure 49 Meeting Service Overall Architecture.....</i>	146
<i>Figure 50 Meeting Service Internal Structure</i>	147
<i>Figure 51 BaseMgt Integration of Meeting Service</i>	148
<i>Figure 52 Resilient Integration between BaseMgt and Meeting Service</i>	149
<i>Figure 53 Circuit Breaker Design Pattern</i>	150
<i>Figure 54 Meetings tab</i>	152
<i>Figure 55 Past Meetings Tab</i>	153
<i>Figure 56 Quick Meeting Sheet.....</i>	154
<i>Figure 57 Full meeting Creation Screen</i>	155
<i>Figure 58 Past Meeting Upload.....</i>	156
<i>Figure 59 Audio Upload in Background with Notifications</i>	156
<i>Figure 60 Online Meeting Pre-Join screen.....</i>	157
<i>Figure 61 Online Meeting Share Screen Functionality.....</i>	158
<i>Figure 62 Online Meeting with 2 participants.....</i>	159
<i>Figure 63 In-person meeting session.....</i>	160

<i>Figure 64 Background-supported Recording with Notification</i>	161
<i>Figure 65 Ongoing Recording In-Person</i>	161
<i>Figure 66 Cached Recording to Upload Later</i>	162
<i>Figure 67 Meeting Details & Report.....</i>	163
<i>Figure 68 AI Analysis Outcome Page.....</i>	164
<i>Figure 69 Whisper model family.....</i>	173
<i>Figure 70 Nova2 architecture.....</i>	174
<i>Figure 71 BART Bidirectional Encoder & Autoregressive Decoder</i>	179
<i>Figure 72 Bart model family</i>	180
<i>Figure 73 T5 Model family.....</i>	180
<i>Figure 74 decoder-only Transformer architecture</i>	181
<i>Figure 75 AI service processing flow.....</i>	188
<i>Figure 76 service architecture</i>	189
<i>Figure 77 Strategy design pattern solution.....</i>	190
<i>Figure 78 authentication solution.....</i>	193
<i>Figure 79 MongoDB stored report.....</i>	200
<i>Figure 80 MongoDB stored report.....</i>	203
<i>Figure 81 Dockerfile to Container</i>	206
<i>Figure 82 Docker vs Docker-Compose.....</i>	206
<i>Figure 83 Deployment Diagram</i>	207
<i>Figure 84 AWS CloudFormation (IaC).....</i>	210
<i>Figure 85 Case 01 - 50 Clients load-test.....</i>	211
<i>Figure 86 Case 01 - 100 Clients load-test.....</i>	211
<i>Figure 87 Case 01 - 100 Clients load-test.....</i>	212
<i>Figure 88 Case 02 - 200 Clients load-test.....</i>	212
<i>Figure 89 Case 02 - 400 Clients load-test.....</i>	213

List of abbreviations

Table 1 table of abbreviations

Abbreviation	Definition
API	Application Program Interface
AI	Artificial Intelligence
UML	Unified Modeling Language
RTC	Real Time Communication
SQL	Structured Query Language.
SOW	Statement of Work
RTM	Requirements traceability matrix
BaseMgt	Base Management
AWS	Amazon Web Services
EKS	Elastic Kubernetes Service
LLM	Large Language Model
REST	REpresentational State Transfer

Chapter1 Introduction

1. Introduction:

In this chapter, we will introduce our project, discussing the main issues and reasons for building this system. We will also explain the objectives and goals we aim to accomplish with this system. Finally, we will provide an overview of the report organization.

2. Problem Definition:

Agile teams rely heavily on frequent meetings such as sprint planning, daily stand-ups, retrospectives, and client discussions to make key project decisions. However, managing and extracting insights from these meetings can be challenging, especially when dealing with large teams, remote work environments, and overlapping discussions. Manually taking notes, summarizing key points, and tracking decisions can be time-consuming, error-prone, and inconsistent, leading to the loss of critical information.

Additionally, Agile teams often struggle with organizing meeting outcomes effectively within their project workflows. Important discussions and action items may get buried in long meeting recordings or scattered notes, making it difficult for teams to track project requirements, priorities, and decision-making rationales. Without an automated and structured system, Agile teams may face difficulties in

maintaining alignment, optimizing workflow efficiency, and ensuring that insights from discussions are properly utilized.

There is a clear need for an AI-driven solution that can automate capturing, summarizing, and structuring meeting discussions. Agile teams can benefit from a centralized, intelligent system that organizes meetings and meetings insights, improves collaboration, and enhances project tracking.

3. Project objectives:

The primary objective of this project is to develop a comprehensive AI-powered meeting management and note taking app for Agile teams. The system will not only process meeting outcomes using AI but also provide end-to-end organization and project management features, allowing Agile teams to efficiently manage members, projects, and requirements while facilitating online and in-person meetings.

To achieve this, the system will:

- Support organization and project management, enabling companies to manage team members, projects, and associated requirements efficiently.
- Facilitate and manage both online and in-person meetings, allowing teams to schedule, conduct, and document meetings seamlessly.

- Use AI techniques to process meeting audio, transcribe speech into text, summarize discussions, and extract key takeaways for improved documentation and decision-making.

Ultimately, this system will enhance collaboration, improve organizational efficiency, and empower Agile teams to make well-informed, data-driven decisions by managing their projects, and teams and meeting outcomes in a structured and intelligent manner

4. Report organization:

The report will be organized in the following chapters:

Chapter 1: introduction.

Chapter 2: Project management and initialization.

Chapter 3: Base Management Service.

Chapter 4: Notification Service.

Chapter 5: Meetings Service

Chapter 6: AI Service.

Chapter 7: Deployment.

Chapter 8: Conclusion.

Chapter 2 project management and initialization

1. Introduction:

In this chapter, we will explore the project management phase, a crucial element in ensuring the project's success. We will focus on scope determination, initial analysis, and system structuring, laying the foundation for a well-organized and efficient development process.

2. Project management documents:

2.1 Project charter:

A project charter is a formal document that serves as an official authorization for the start of a project. It acts as a reference point throughout the project, providing a clear understanding of the project's purpose and establishing a foundation for decision-making and project governance.

<i>Project title</i>	<i>Agile-Focused AI Note-taking app.</i>
<i>Project start date</i>	<i>September – 20 – 2024</i>
<i>Project finish date</i>	<i>February – 23 – 2025</i>
<i>Project manager</i>	<i>Dr. Raid Sonbol</i>
<i>Project objectives</i>	<i>Develop an AI-powered software system to assist Agile software development teams in capturing, organizing, and managing meeting artifacts efficiently.</i>
<i>Approach</i>	<ol style="list-style-type: none"> <i>1. Define project scope and objectives.</i> <i>2. Highly Analyze system requirements and specifications.</i> <i>3. Design system architecture and decomposition.</i> <i>4. Develop the first increment, focusing on accounts, project, and organization management.</i>

5. *Test and validate functionalities of the first increment.*
6. *Develop the second increment, development of the AI components, and the meeting management components.*
7. *Perform testing on the second increment.*
8. *Document all development phases and outcomes.*

Roles and responsibilities:

<i>Name</i>	<i>Role</i>	<i>Responsibility</i>
<i>Dr. Riad Sonbol</i>	<i>Supervisor</i>	<i>Highly Project management and work monitoring.</i>
<i>Eng. Anas Abdulaziz</i>	<i>Supervisor</i>	<i>work monitoring.</i>
<i>Raghad al hossny</i>	<i>Engineer</i>	<i>Backend development – documentation – AI components – frontend development.</i>
<i>Mohamed al-Balkhi</i>	<i>Engineer</i>	<i>mobile development – backend development - deployment- documentation.</i>

2.2 The SOW document:

Statement of Work is a comprehensive document that defines the scope of work for a project. It outlines the specific tasks, deliverables, timeline, and responsibilities. The SOW document provides a clear understanding of what needs to be accomplished, the project's objectives, and the criteria for success.

<i>Project Title:</i>	<i>Agile-focused AI note-taking app</i>
<i>Project Description and Objectives:</i>	<i>The project aims to develop an AI-based software system that helps Agile teams extract and manage the key insights gained from frequent different types of meetings.</i>
<i>Project Scope:</i>	<i>This project involves developing a full-stack software system that integrates with AI technologies. The system will record and transcribe meeting discussions (both online and in-person), generate structured summaries, and highlight key discussion points, provide privileges-based access control for managing user permissions, offer an interactive dashboard for tracking projects with their requirements and related meetings.</i>
<i>Project goals:</i>	<ul style="list-style-type: none"> ▪ <i>Develop a software system that helps software companies and teams that follow agile methodology, to optimize decision-making that is driven from frequent meeting insights.</i> ▪ <i>Use AI techniques for processing meeting audio: transcript the audio and generate valuable summary and key points.</i> ▪ <i>Give high-level organization and project management features.</i>
<i>Project Deliverables:</i>	<ul style="list-style-type: none"> ○ <i>Project plan.</i> ○ <i>Working software – web app.</i>

	<ul style="list-style-type: none">○ <i>Working software – mobile app.</i>○ <i>Final project report.</i>
<i>Technology and Tools:</i>	<ul style="list-style-type: none">- <i>Programming Languages: Python, C#, JavaScript, Dart – HTML - CSS.</i>- <i>Frameworks: Django and asp.net for the backend, React and flutter for the frontend.</i>- <i>Database: PostgreSQL, MongoDB.</i>- <i>AI-Models: Nova02 – GPT 4o mini.</i>
<i>Assumptions:</i>	<ul style="list-style-type: none">▪ <i>Continuous availability of project team members.</i>▪ <i>Regular supervision and feedback from the project manager.</i>▪ <i>Continuous delivery of working software components.</i>
<i>Project Resources</i>	<p><i>Dr. Raid Sonbol - Project Manager</i></p> <p><i>Eng. Anas Abdulaziz – supervisor.</i></p> <p><i>Raghad al Hossny - SE Developer</i></p> <p><i>Mohamed al Balkhi - SE Developer</i></p>
<i>Schedules:</i>	<p><i>Project Start Date: September 20, 2024</i></p> <p><i>First project Seminar: October 18, 2024</i></p> <p><i>second project Seminar: November 30, 2024</i></p> <p><i>Project Finish Date: February 23, 2024.</i></p>

2.3 Risk Management:

the process of identifying, evaluating, and mitigating potential risks that could impact the success of the project or the team.

Table 2 Risk management

Risk_ID	Rk-01	Rk-02	Rk-03	Rk-04
Risk title	Small Team Size (Only Two Members).	Unclear Project Scope	Learning and Adopting New Technologies	Workload Separation in the Implementation Phase
Risk Description	If one student is unable to continue for any reason, the project's progress will be significantly impacted.	A misunderstanding of the project scope at the beginning could lead to major errors in development.	The project requires developing advanced software and AI components, but our initial knowledge in these areas is limited.	At a certain stage of development, our project was being worked on in parallel, with team members focusing on different aspects simultaneously.
Raised Date	20/9/2024	20/9/2024	1/12/2024	1/1/2024
Tracking Frequency	weekly	daily	weekly	daily
State	active	closed	Under Mitigation	Under Mitigation
Impact	high	medium	high	medium
Mitigation Plan	The member who completes their tasks first will assist the other.	Conduct regular meetings at the end of each phase to ensure alignment.	Engage in continuous learning and self-development.	Assign an Implementation Leader to oversee and coordinate both parts of the project.

3. System initial analysis and design:

This section presents the initial analysis and key features identified during the early stages of project planning and development.

3.1 High-level analysis:

Use Case Diagram: use-case diagrams model the behavior of a system and help to capture the requirements of the system.

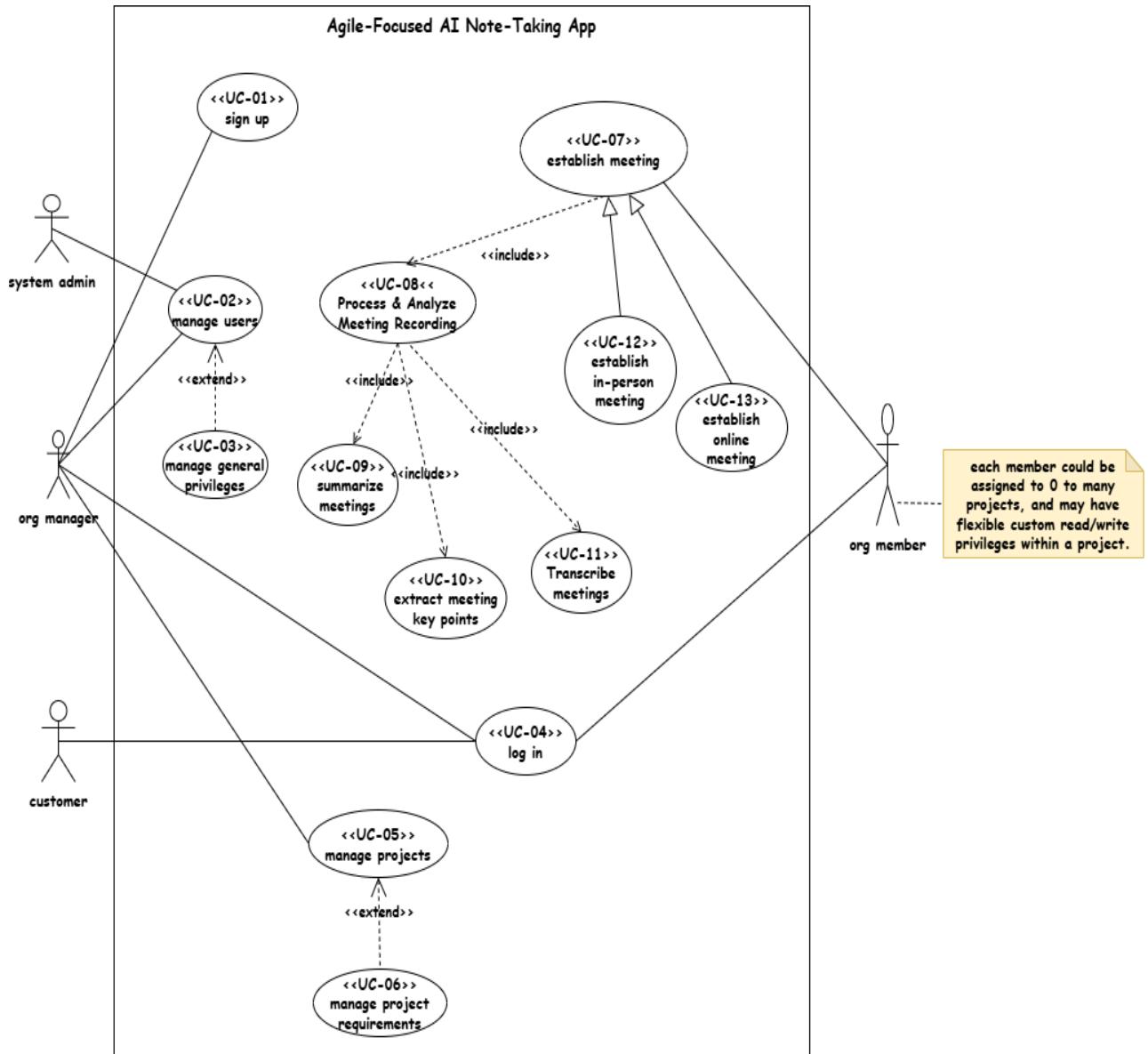


Figure 1 high level use case diagram

High-level Requirements:*Table 3 high level project requirements*

Req_ID	Requirement Title	category	description
RE-FR-AM-01	The system should provide organization managers the ability to create a new account, using their credentials	account management	The credentials needed are only email + a strong password (according to the system password limits), full name, and birthdate.
RE-FR-AM-02	The system should be able to verify users' emails by using a verification code "OTP".	account management	The OTP is a random 5-digit number.
RE-FR-AM-03	The system should allow all users to log in to their accounts using their credentials	account management	email + password
RE-FR-AM-04	The system should allow users to view and manage their account information	account management	basic information like "username", photo, email...
RE-FR-OM-05	The system should provide organization managers with the ability to register and manage their organization into the system.	Organization management	
RE-FR-OM-06	The system should provide the organization manager the ability to create and add new organization members to the organization.	Organization management	Email, default password.
RE-FR-NM-07	The system should be able to send email to the new created organization members.	Notifications management	

RE-FR-AM-08	The system should ask new members with default passwords to update their own information and change the password.	Account management	
RE-FR-OM-09	The system should provide organization managers with the ability to deactivate Organization members' accounts.	Organization management	
RE-FR-OM-10	The system should allow the organization manager or a member with the right privileges to assign and manage privileges to team members.	Organization management.	The system defines fixed privileges, read/write on every aspect of the project.
RE-FR-PM-11	The system should provide the organization manager or a member with the right privileges the ability to create a new project.	project management	
RE-FR-PM-12	The system should provide the organization manager or a member with the right privileges the ability to delete a project.	Project management	
RE-FR-PM-13	The system should be able to display all projects related to an organization.	Project management	
RE-FR-PM-14	The system should provide the organization manager or a member with the right privileges the ability to assign a member to a project.	Project management	
RE-FR-PM-15	The system should be able to view the members related to a specific project.	Project management	
RE-FR-RM-16	The system should provide project managers the ability to add project requirements manually.	Requirements management	

RE-FR-RM-17	The system should provide project managers the ability to add project requirements by uploading files.	Requirements management	The system supports CSV, Excel files.
RE-FR-RM-18	The system should be able to display the project requirements Backlog.	Requirements management	
RE-FR-MM-19	The system should provide a member with the right privileges the ability to establish new in-person meetings.	Meeting management	
RE-FR-MM-20	The system should provide a member with the right privileges the ability to establish new online meetings.	Meeting management	
RE-FR-MM-21	The system should provide a member with the right privileges the ability to add members to the meeting.	Meeting management	
RE-FR-MM-22	The system should provide a member with the right privileges the ability to create scheduled meetings.	Meeting management	
RE-FR-MM-23	The system should provide a member with the right privileges the ability to create recurrent meetings.	Meeting management	
RE-FR-MM-24	The system should be able to show the meeting history.	Project management	
RE-FR-NM-25	The system should be able to send notifications (email, in-app notifications) to the meeting members about the meeting time, and information.	Notifications management	

RE-FR-MM-26	The system should be able to record the meetings.	meeting management	for both online meetings & in-person meetings.
RE-FR-AI-27	The system should be able to process meeting records	AI processing	
RE-FR-AI-28	The system should be able to use AI techniques to transcribe the meeting record.	AI processing	
RE-FR-AI-29	The system should be able to use AI techniques to summarize the meeting transcript to a meaningful summary.	AI Processing	
RE-FR-AI-30	The system should be able to use AI techniques to extract meeting key points.	AI Processing	
RE-FR-AI-31	The system should be able to generate a meeting final record.	AI processing	
RE-NFR-AI-32	The system should be able to support many languages for the transcription process.	Nonfunctional; requirements	AI component
RE-FR-SM-33	The system should be able to generate usage reports to the system admin.	System management	No. organization. No. projects. No. users, And more.
RE-NF-34	The system should handle requests that are not related to AI processing in less than 3 seconds.	Nonfunctional requirements	
RE-NF-35	The system should handle requests that are related to the AI processing in less than 4 minutes	Nonfunctional requirements	
RE-NF-36	The system should provide strong authentication ways using access and refresh tokens to provide security.	Nonfunctional requirements	
RE-NF-37	The system should handle privileges and access control to apply security.	Nonfunctional requirements	

RE-NF-38	The system should enable components to be reused across different modules or projects without requiring significant modifications.		Reusability
----------	--	--	-------------

3.2 High-level system decomposition:

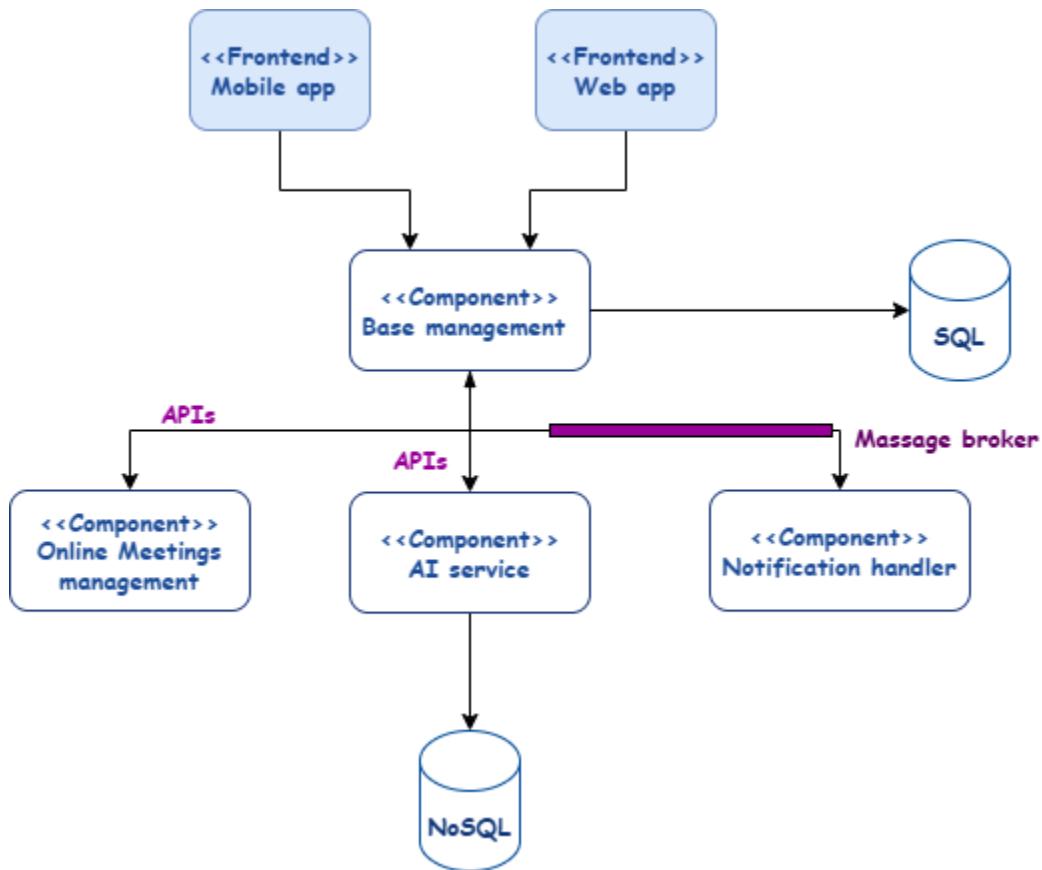


Figure 2 initial system decomposition

This High-Level System Decomposition diagram provides an overview of the system's architecture – **microservices architecture** – illustrating its main components and their interactions.

1. Frontend: The system consists of two frontend interfaces:
 - Mobile App (for users accessing the system via mobile devices).
 - Web App (for users accessing through web browsers).
2. Backend:
 - Base Management Component:

This serves as the core backend module, handling essential functionalities such as authentication, authorization, and data management.
It communicates with multiple subsystems plays as the gateway of the backend system and manages data storage.

 - Online Meetings Management Component: provide online meeting establishment service provider and handle meetings-related issues, called by the base management component via REST APIs.
 - AI Service Component: a standalone component processes meeting recordings, transcribes speech, generates summaries, extracts key points using AI techniques, and stores the result in a NoSQL database.
 - Notification Handler Component: handles sending emails and notifications, called and used by the base management component.

4. Project development life cycle model:

4.1 Determine development model based on system requirements and structure:

After conducting a high-level analysis, eliciting system requirements, and decomposing the system into its main components, the next step is to determine the most suitable software development process for our system. Given the nature

of our system and its requirements, we have chosen the Incremental Development Model as the most appropriate approach.

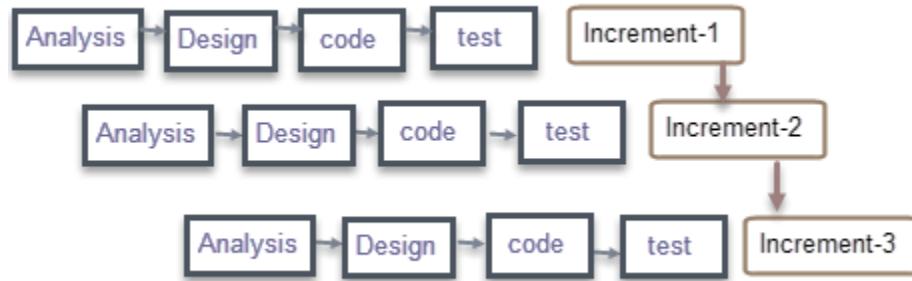


Figure 3 used software development model

The incremental approach aligns well with our system's structure since it consists of separate, independent components that can be developed individually as standalone sub-projects. This model allows us to gradually build and integrate system components, ensuring flexibility and iterative improvements.

4.2 Incremental project management:

Unlike traditional models where the entire project is planned upfront, incremental project management plans and executes in phases where the focus is on achieving key business objectives and delivering benefits to the customer in a series of small increments or stages. The **Project Charter** and **Statement of Work (SOW)**, created at the project's beginning, may evolve to accommodate updates and refinements based on system needs and development progress.

4.3 Initial Project Plan – Gantt chart:

A document outlining tasks, deadlines, and resources needed to achieve project objectives, serving as a roadmap for successful project execution, and reflecting the system development model.

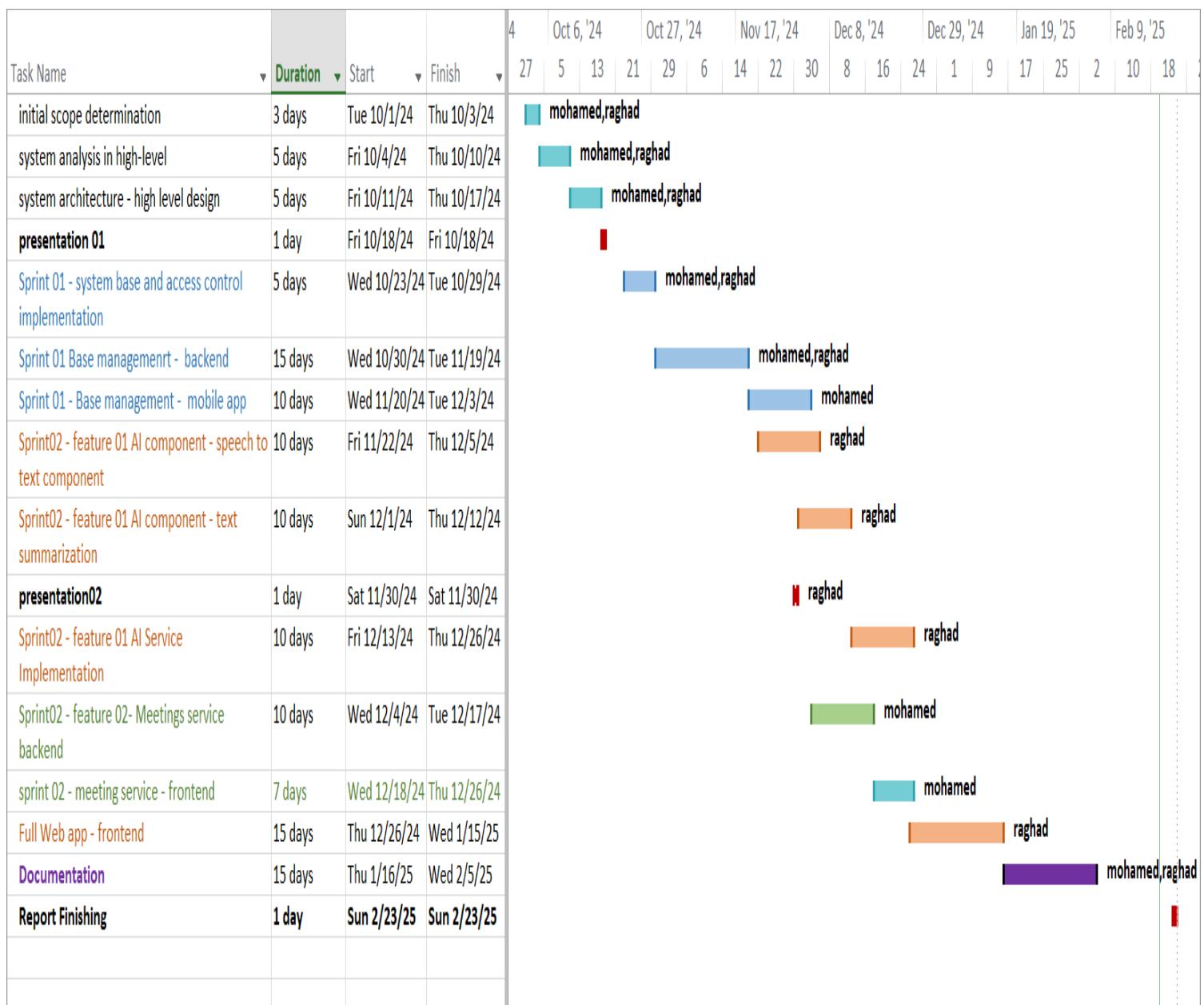


Figure 4 Gantt chart

5. Summary:

In conclusion, good project management is vital for successful software system development. It provides structure, ensuring projects are completed within scope and schedule.

Overall, strong project management enhances the delivery of high-quality software systems that meet the goals of the project.

Chapter 3 Base Management Service

1. Introduction:

The Base Management Service is the central component of the system, serving as the backend API gateway. It is responsible for handling all requests from front-end applications, ensuring a structured and controlled flow of data. This service plays a critical role in managing user accounts, organizations, projects, and requirements, acting as the backbone of the system.

In this section we provide detailed documentation of the analysis, design, and development of the Base Management Service, highlighting its functionalities, architecture, and role in the overall system.

2. Service Detailed analysis:

2.1 Introduction:

In this section, we will introduce the analytical study of the service using the needed UML diagrams for system requirements modeling.

2.2 Detailed Service Requirements:

Table 4 base management service detailed requirements

Req_ID	Requirement Title	category	description
RE-FR-AM-01	The system should provide organization managers the ability to create a new account, using their credentials as the first step of the registration flow	account management	The credentials needed are only email + a strong password (according to the system password limits), full name, and birthdate.

RE-FR-AM-02	The system should be able to verify users' emails by using a verification code "OTP" as the second step of the registration flow.	account management	The OTP is a random 5-digit number.
RE-FR-AM-03	The system should be able to resend the OTP for a user when they ask for.	account management	
RE-FR-OM-04	The system should provide organization managers with the ability to register their organization in the system as the third step of the registration flow.	Organization management	
RE-FR-AM-05	The system should allow Org. manager to complete the sign-up flow after a while from their current step.	Account management	
RE-FR-AM-06	The system should allow all users to log in to their accounts using their credentials	account management	email + password
RE-FR-AM-07	The system should allow users to view their account information	account management	basic information like "username", photo, email...
RE-FR-AM-08	The system should provide users the ability to update their account information.	account management	
RE-FR-AM-09	The system should provide users the ability to reset their account password when the password is forgotten.	account management	
RE-FR-AM-10	The system should be able to verify the email for password reset requests.	account management	
RE-FR-AM-11	The system should provide users the ability to change their account password to a new password that also meets the system requirements.	account management	

RE-FR-AM-12	The system should be able to check if any password strength meets the system's requirements	account management	<ul style="list-style-type: none"> - 8 at least. - Continue a letter uppercase. - Contain symbol.
RE-FR-OM-13	The system should provide the organization manager the ability to create a new organization member account.	Organization management	Email, default password.
RE-FR-AM-14	The system should force the newly created members to complete their account missing information.	account management	
RE-FR-AM-15	The system should ask new members with default passwords to change the password, to a strong password that follows the system requirements.	Account management	
RE-FR-OM-16	The system should provide organization managers with the ability to deactivate Organization members' accounts.	Organization management	
RE-FR-OM-17	The system should allow the organization manager or a member with the right privileges to assign privileges to team members.	Organization management.	The system defines fixed privileges, read/write on every aspect of the project.
RE-FR-OM-18	The system should allow the organization manager or to make a member as admin.		The admin full access privileges over the organization.
RE-FR-OM-19	The system should allow the organization manager or a member with the right privileges to remove a privilege from a team member.	Organization management	
RE-FR-OM-20	The system should be able to show each organization member with their related privileges.	Organization management.	

RE-FR-PM-21	The system should provide the organization manager or a member with the right privileges the ability to create a new project.	project management	
RE-FR-PM-22	The system should provide the organization manager or a member with the right privileges the ability to deactivate a project.	Project management	
RE-FR-PM-23	The system should be able to display all projects related to an organization.	Project management	
RE-FR-PM-24	The system should provide the organization manager or a member with the right privileges the ability to assign a member to a project.	Project management	
RE-FR-PM-25	The system should provide an org manager with the right privileges to remove a member from a project.	Project management	
RE-FR-PM-26	The system should be able to view all the members related to a specific project.	Project management	
RE-FR-RM-27	The system should provide project managers with the ability to add project requirements manually.	Requirements management	
RE-FR-RM-28	The system should provide project managers the ability to add project requirements by uploading files.	Requirements management	The system supports CSV,21111 and Excel files.
RE-FR-RM-29	The system should be able to display the project requirements Backlog.	Requirements management	
RE-FR-RM-30	The system should provide the ability to update a project requirement.	Requirements management	

RE-FR-RM-31	The system should provide the ability to delete a project requirement from the requirement list.	Requirements management	
RE-FR-OM-32	The system should provide the organization manager with statistical insights to describe the organization's status effectively.	Organization management	Active projects no., inactive projects no. , members no.
RE-NF-33	The system should respond to any action in less than 3 sec.	Nonfunctional requirements	Performance
RE-NF-34	The system should provide strong authentication ways using access and refresh tokens to provide security.	Nonfunctional requirements	Security
RE-NF-35	The system should handle privileges and access control to apply security.	Nonfunctional requirements	Security
RE-NF-36	The system should provide easy-to-use interfaces for users, Users need on average “10 min” to completely navigate and understand all the parts and pages of the system.	Nonfunctional requirements	Usability.
RE-NF-37	The system should allow modifications, such as bug fixes or feature updates, to be implemented with minimal effort and without affecting other components.	Nonfunctional requirements	Maintainability
RE-NF-38	The system should allow new features or functionalities to be added with minimal changes to the existing codebase and without affecting core components – communicate with other new services.	Nonfunctional requirements	Extensibility

2.3 Requirements Modeling:

Use case diagram:

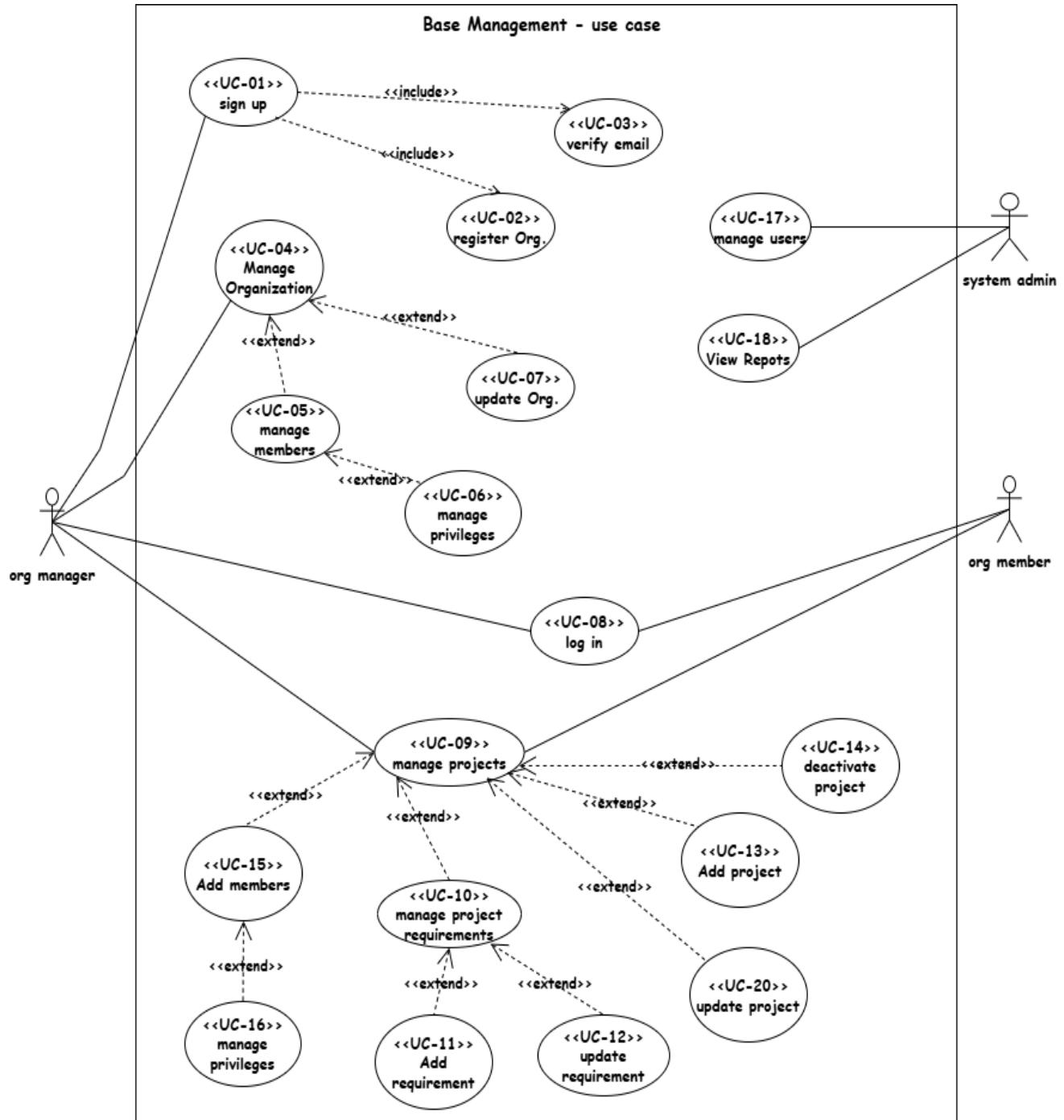


Figure 5 use case diagram

service features – Sign-up flow UC-01:

Table 5 sign-up flow specification

Use case name:	Sign-Up
Participating Actors:	Initiated by: organization manager
The flow of events:	<ol style="list-style-type: none"> 1. The organization manager enters the website on the sign-up page. 2. The system shows the first form for the base account data. 3. The Org manager completes the form and chooses “create account”. 4. The system applies verification over the form fields 5. The system will send to the Org. manager an OTP via email to verify it. 6. The Org. manager enters the received OTP. 7. The system checks the entered OTP. 8. The system routes the organization manager to the third form page to register their organization. 9. The Org. manager completes the fields needed. 10. The system applies verification over the form fields. 11. The system creates and saves the new account and organization data and routes the Org. manager to the main page in the system.
Alternative flows:	<p><u>First alternative flow A1:</u> start at step 4 or 10 in the main flow, there is a missing field: 5. the system will show an error message “There is a missing field”.</p> <p>6. the Org. manager will complete the fields, and the flow will return to step 3.</p> <p><u>Second alternative flow A2:</u> start at step 4 from the alternative flow: the email already exists.</p> <p>6. The system will ask the Org. manager to choose another email.</p> <p>7. The Org manager will reenter the email field, and the flow will return to step 4.</p> <p><u>Third alternative flow A3:</u> if the password the Org. manager chose is weak start at 4:</p>

	<p>6. The system will tell the Org. manager owner that the password must be at least 8 characters.</p> <p>7. The Org. manager will reenter the password, and the flow will return to step 4.</p> <p><u>Fourth alternative flow A4:</u> if the OTP is incorrect start at step 7 in the main flow:</p> <p>5. the system will show an error message “incorrect OTP “.</p> <p>6. the Org. manager will choose “resend OTP”.</p> <p>And the flow will return to step 6.</p> <p><u>First exception flow E1:</u> if the verify code is incorrect the system will cancel the whole process, and the use case will fail.</p>
Entry condition	The user opens the system website.
Exit conditions	The Org. manager has an account, organization is registered

Activity diagram:

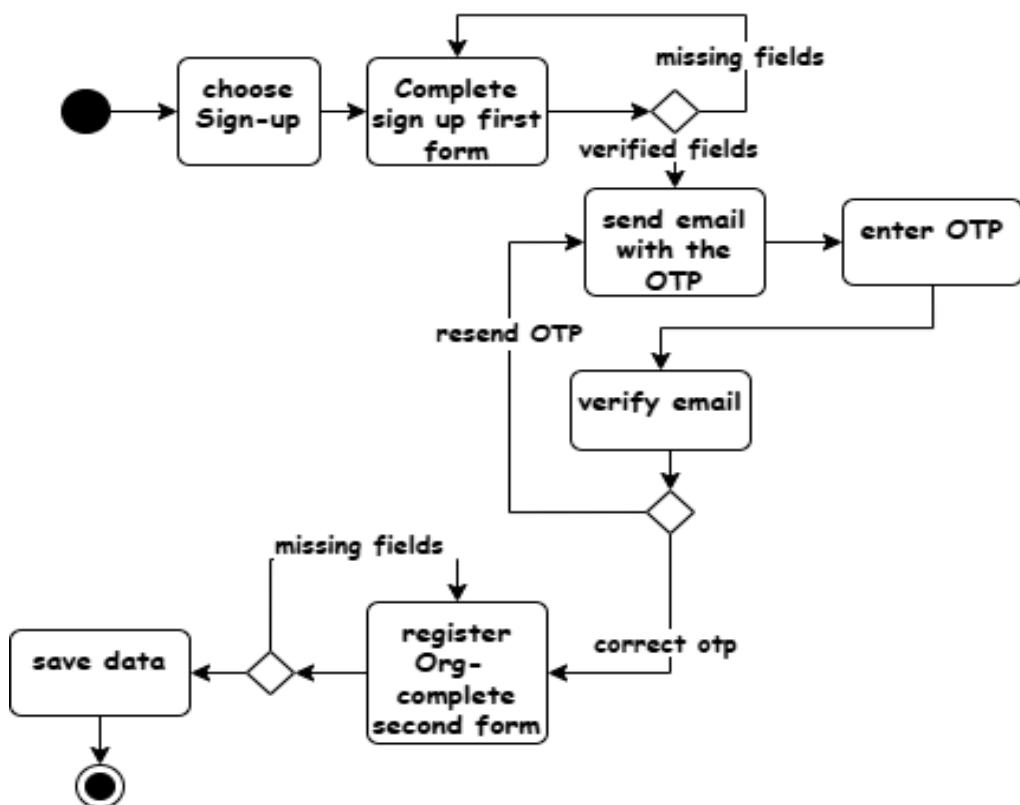


Figure 6 sign-up flow

Sequence diagram:

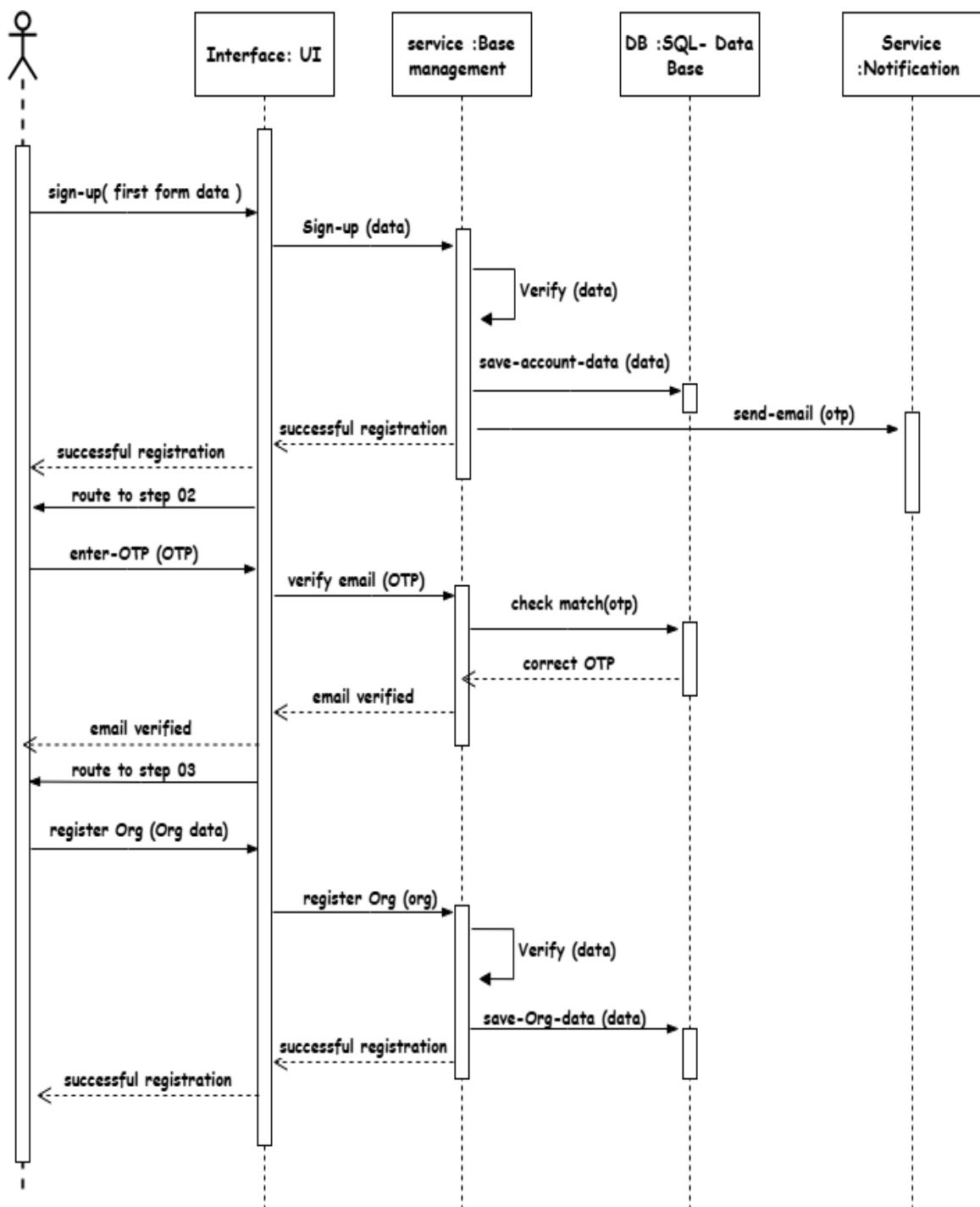


Figure 7 Sign-up sequence diagram

service features – manage members UC-05:

Table 6 manage members specification

Use case name:	Manage members
Participating Actors:	Initiated by organization manager or any organization member with the right privileges (admin).
The flow of events:	<ol style="list-style-type: none"> 1. When the organization manager chooses to add a new Org. member. 2. The system shows the “add new member” form. 3. The Org. manager enters the email for the new member. 4. The system will verify the entered email. 5. The system will generate a default password for the new account created. 6. The system sends emails to notify the members of the account creation with the password generated. 7. The organization member logs in to the system, with their default credentials. 8. The system shows the complete account form. 9. The organization member completes the account missing data and changes the password. 10. The system makes the members an active Org. member. 11. If the organization manager chooses to manage the member privileges. 12. The system shows all organization members. 13. The org. manager chooses to make a member as an Org. admin. 14. The system gives the admin privilege to the org. member.
Alternative flows:	<u>First exception flow E1:</u> start at step 4 from the alternative flow: the email already exists. 5. The system will show an error message for this member and the use case will fail.
Entry condition	The user has the right privileges.
Exit conditions	A New Org member account was created.

Sequence diagram:

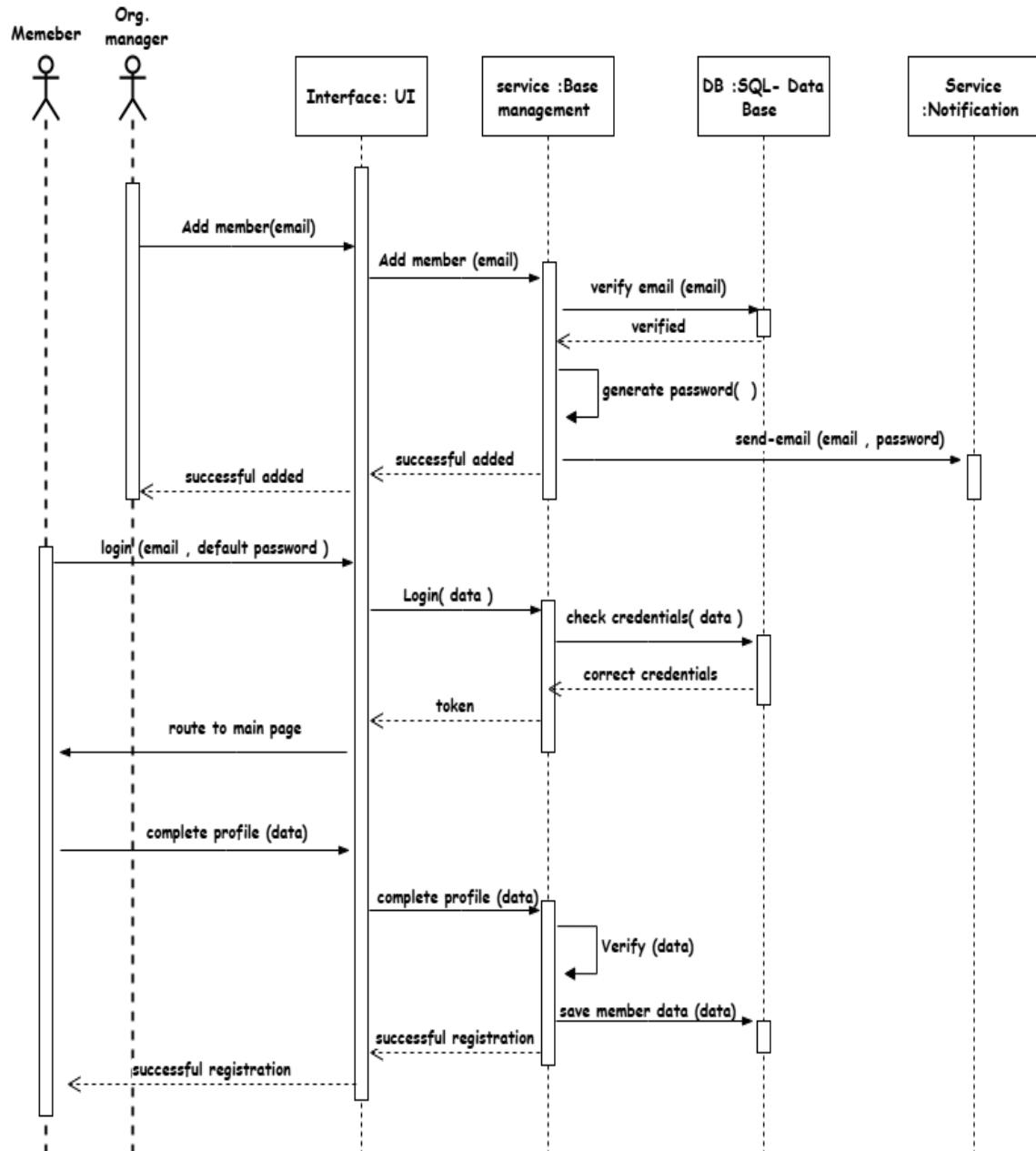


Figure 8 manage member sequence diagram

Activity diagram:

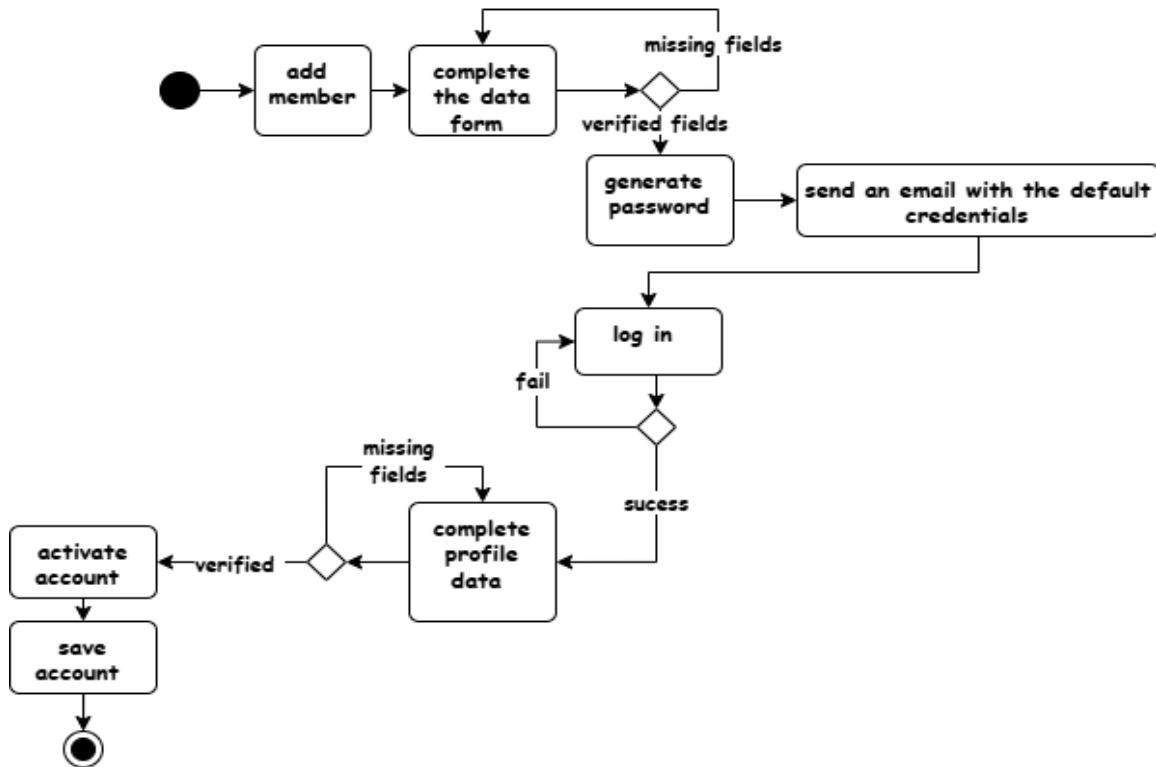


Figure 9 manage member activity diagram

service features – manage projects UC-09:

Table 7 manage projects specification

Use case name:	Manage projects
Participating Actors:	Initiated by: organization manager or any organization member with the right privileges (admin).
The flow of events:	<ol style="list-style-type: none"> 1. When the user enters the “projects” interface. 2. The system shows the Org. existing project active and inactive ones. 3. If the user asks to add a new project: <ol style="list-style-type: none"> a. The system shows the “add project” form

	<ul style="list-style-type: none"> b. The user enters the project information needed and completes the form. c. The user adds an organization member as a project manager. d. The system verifies the information entered. e. The system creates the new project and saves it to the database. f. The system gives the project manager member full access privileges over this project <p>4. If the user asks to deactivate a project:</p> <ul style="list-style-type: none"> a. The system asks to select a project. b. The user selects a project c. The system deactivates the selected project. <p>5. If the user asks to update project information:</p> <ul style="list-style-type: none"> a. The system shows the old information form. b. The user updates the needed fields. c. The system verifies and updates the project information and saves it.
Alternative flows:	<p><u>First alternative flow A1: start at step 7 from the alternative flow: “missing data”</u></p> <p>8. The system shows an error message of missing data. And the flow returns to step 5.</p>
Entry condition	The user has the right privileges.
Exit conditions	A new project was created.

Sequence diagram – create project:

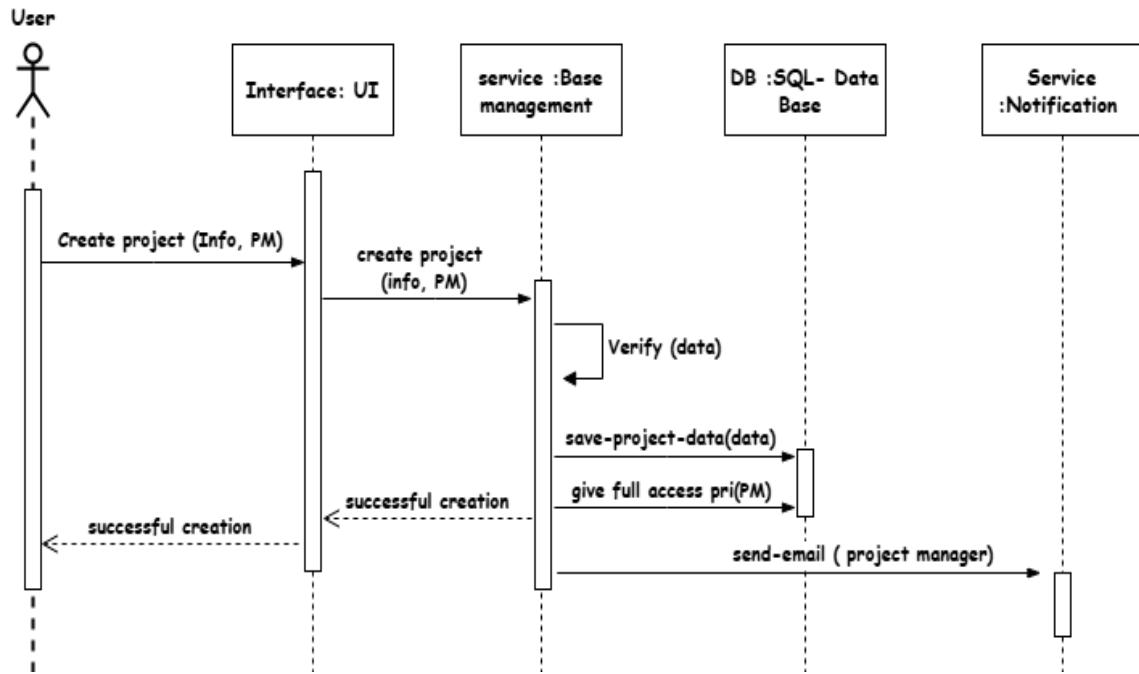


Figure 10 create project sequence diagram

Sequence diagram – update project:

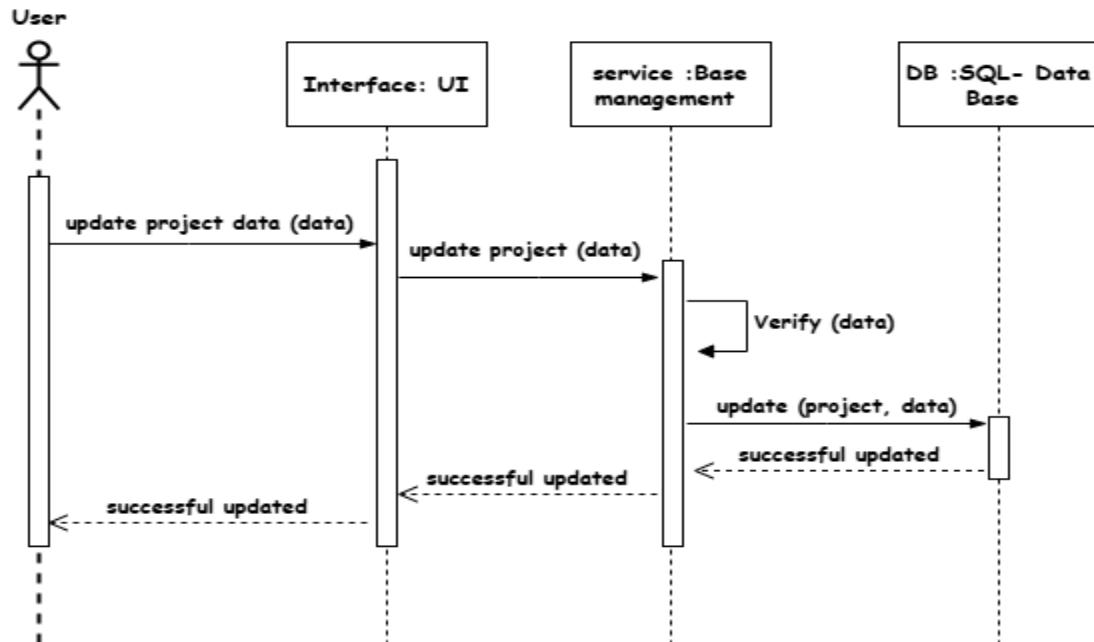


Figure 11 update project sequence diagram

service features – manage project requirements UC-10:*Table 8 manage project requirements specification*

Use case name:	Manage project requirements
Participating Actors:	Initiated by: organization manager or any organization member with the right privileges.
The flow of events:	<ol style="list-style-type: none"> 1. If the user chooses to manage requirements. 2. The system will open the requirements management interface and display the req backlog for this project. 3. If the user asks to “add a new requirement” to the table manually: <ol style="list-style-type: none"> a. The system will give a new empty row on the req. table and ask the user to enter information to complete the row. b. The user will enter the req.id, req. title, req. priority. c. The system will add the new req to the req table and display it to the user. 4. If the user asks to add the requirements from a file. <ol style="list-style-type: none"> a. The system shows the file uploading interface with the related instructions. b. The user will upload a local file. c. The system will parse and extract requirements from the file and add it to the project backlog. 5. If the user, asks to “delete a specific req”. <ol style="list-style-type: none"> a. The system will ask to confirm the process. b. The user will confirm the process. c. The system will delete the req from the table and display the edited table to the user. 6. If the user, asks to “edit a specific requirement information”. <ol style="list-style-type: none"> a. The system asks to enter new information about the requirement. b. The user will enter the information needed. c. The system will edit the req. info and display it to the user.
Entry condition	The user has the right privileges.
Exit conditions	A new project was created.

Sequence diagram:

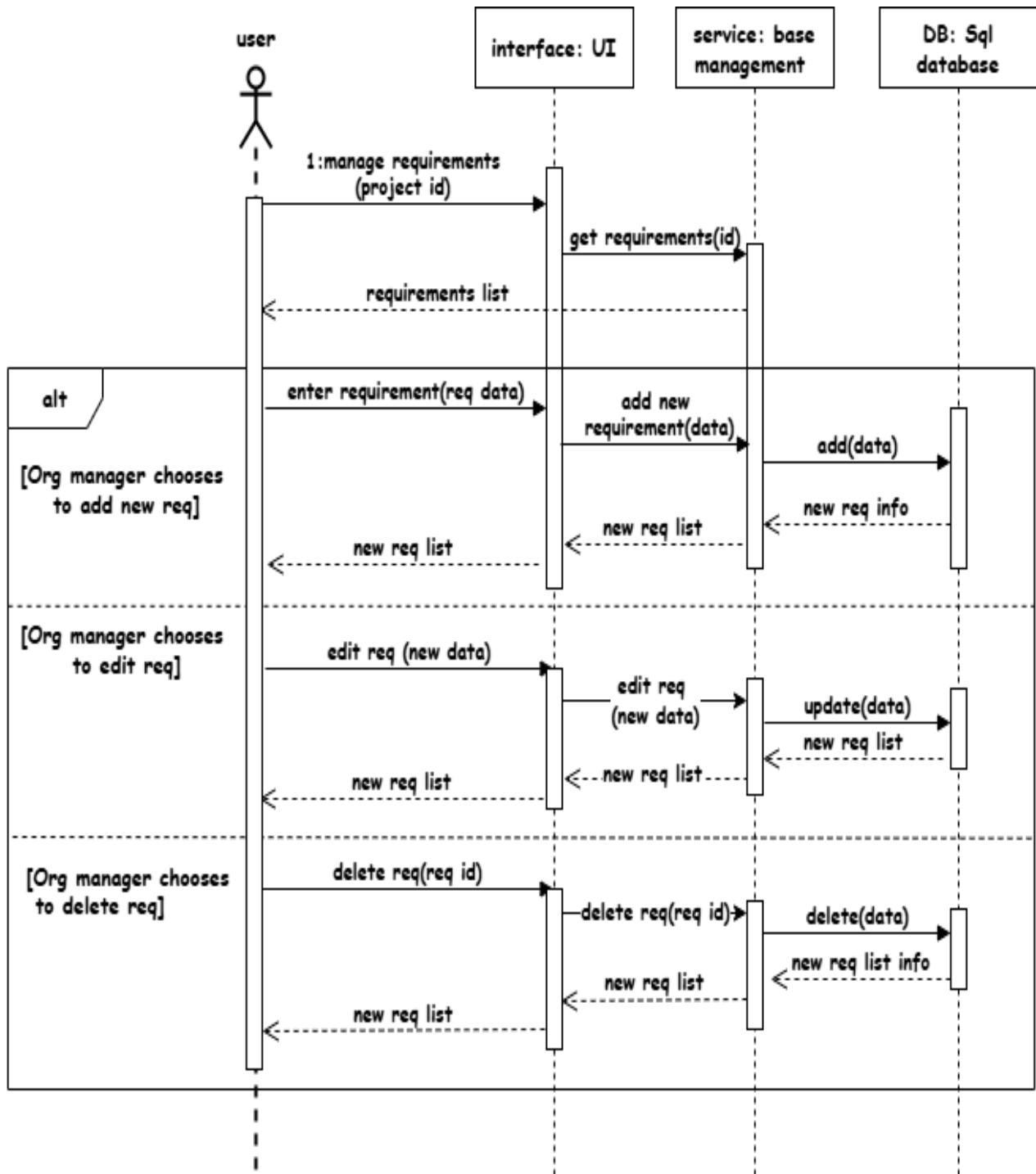


Figure 12 requirements management sequence diagram

service features – manage members' privileges UC-16:

Use case name:	Manage members privileges
Participating Actors:	Initiated by: organization manager or any organization member with the right privileges.
The flow of events:	<ol style="list-style-type: none"> 1. When the user chooses to assign a member to a project. 2. The system shows the existing members of the organization. 3. The user chooses a member to assign to this project. 4. The system assigns the member to the project and gives them the default privileges (read for (meetings, members, requirements, none for settings). 5. If the user chooses to edit the privileges 6. The user will choose the new privilege over each section (members, meetings, requirements, settings). 7. The system will update the privileges of the member. 8. The system will save changes. 9. The system notifies the member about the changes
Entry condition	The user has the right privileges.
Exit conditions	Member privilege managed.

Sequence diagram:

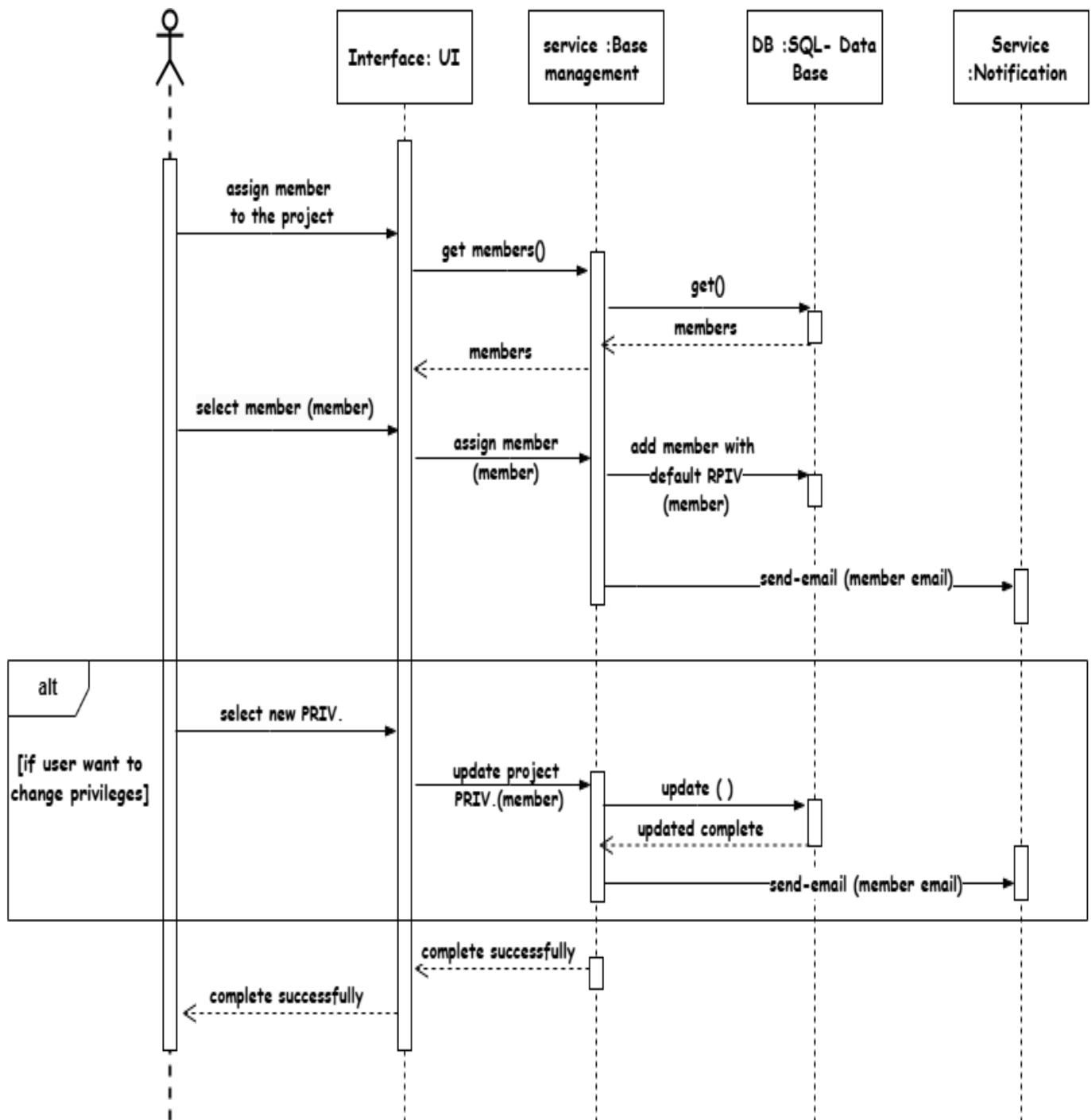


Figure 13 manage privileges sequence diagram

ERD Diagram:

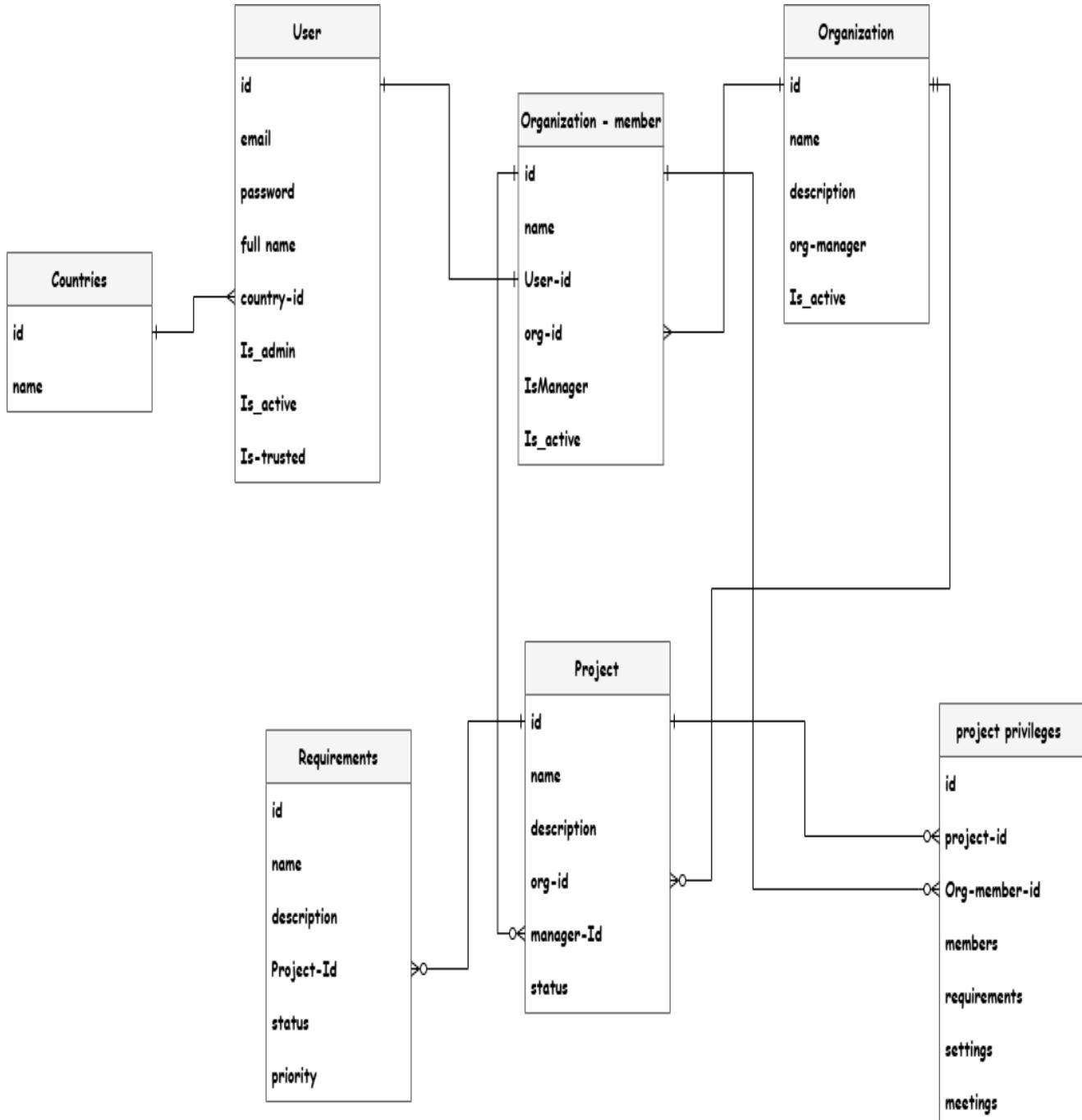


Figure 14 entity relationship diagram

Class diagram:

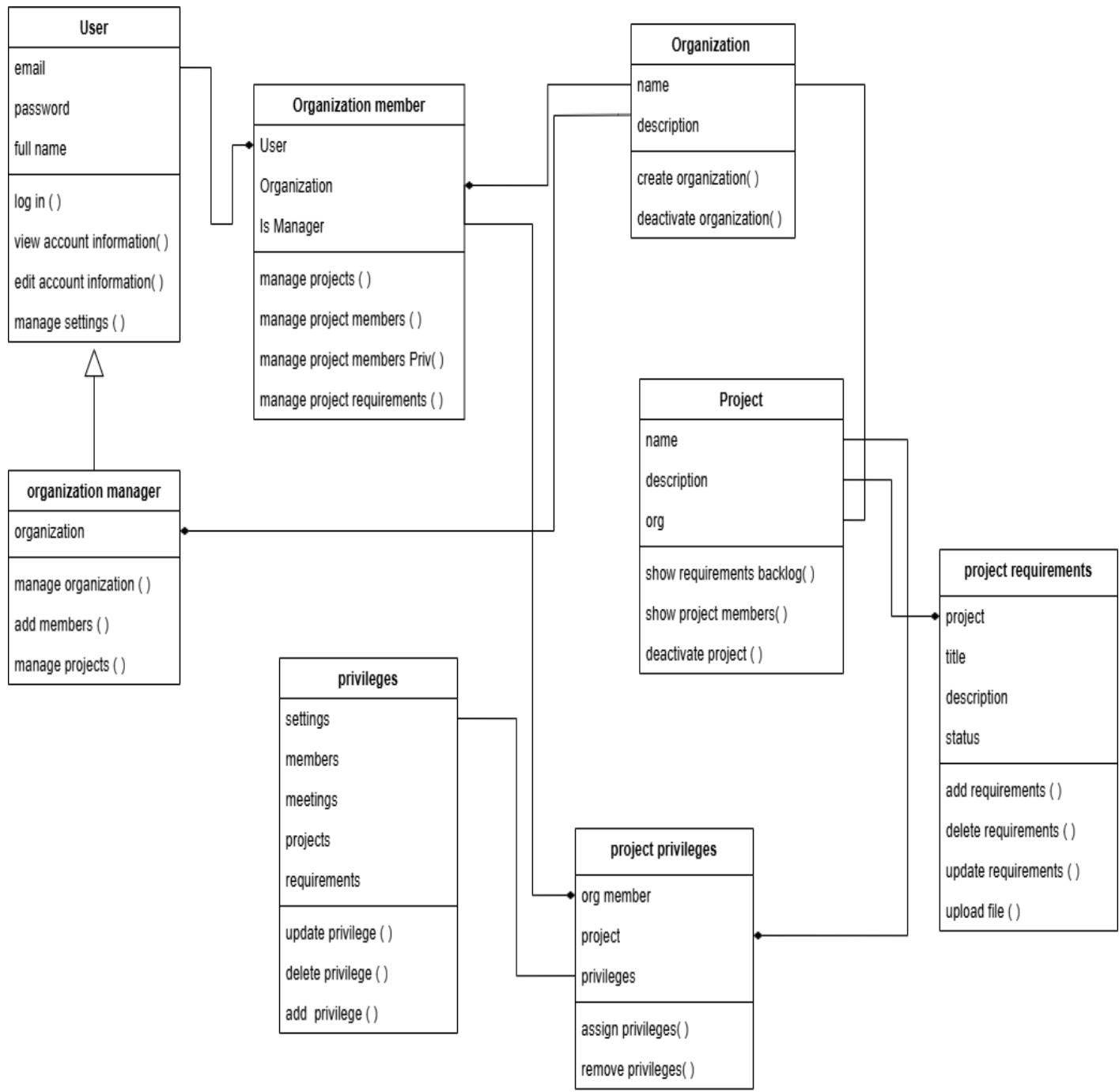


Figure 15 class diagram

2.4 Test cases definition:

Table 9 base management - test cases definition

Test Case Scenario:		Sce-01: Check Login Functionality.	
Test case id	Test case title	Test steps	Expected result
Tc-01	Check results on entering a valid email and password.	<ol style="list-style-type: none"> 1. Launch the application on the login page. 2. Enter your email and password. 3. Choose “login”. 	The login should be successful
Tc-02	Check results on entering an invalid email, or password.	<ol style="list-style-type: none"> 1. Launch the application on the login page. 2. Enter an email and password. 3. Choose “login”. 	Error message “invalid email or password.”
Tc-03	Check results when a user email or password is empty, and the “login” button is pressed.	<ol style="list-style-type: none"> 1. Launch the application on the login page. 2. Enter a password. 3. Choose “login”. 	Error message “a field is missing”
Tc-04	Check results when user chooses “forget password”.	<ol style="list-style-type: none"> 1. Launch the application on the login page. 2. Choose “forget password”. 3. Enter the new password needed. 	The system would send an OTP to verify user email, if the email verified the password will updated successfully.

Test case scenario:		Sce-02: Sign-up functionalities	
Test case id	Test case title	Test steps	Expected result
Tc-05	Check results on completing the first sign-up form correctly and select sign-up.	<ol style="list-style-type: none"> 1. Launch the application on the sign-up page. 2. Complete the form of your information. 3. Choose “sign-up” 	The account should be created successfully, and the front app will route to step 2
Tc-06	Check results on entering an existing email.	<ol style="list-style-type: none"> 1. Launch the application on the sign-up page. 2. Complete the form of your information. 3. Choose “sign-up” 	Error messages said, “The email already existed”.
Tc-07	Check results when a user chooses a weak password (according to the system requirements).	<ol style="list-style-type: none"> 1. Launch the application on the sign-up page. 2. Complete the form of your information. 3. Choose “sign-up” 	Error message said, “weak password!”
Tc-08	Check results when any field is empty, and the “Sign-up” button is pressed.	<ol style="list-style-type: none"> 1. Launch the application on the sign-up page. 2. Complete the form of your information. 3. Choose “sign-up” 	Error message “a field is missing”
Tc-09	Check results when entering the correct OTP received via the email	<ol style="list-style-type: none"> 1. Enter the received OTP. 2. Choose “verify”. 	The email will be verified successfully, and the user will become Trusted, the

			web app will route to the third step.
Tc-10	Check results when entering a wrong OTP received via the email	1. Enter the OTP 2. Choose “verify”.	The system will show an error message “Wrong OTP!”
Tc-11	Check results when the user chooses the “resend OTP”.	1. Launch the email verification page. 2. Choose “Resend”.	The system will generate a new OTP and send it to the user.
Tc-12	Check results when the user completes the organization register form as expected and presses “register”.	1. Launch the organization register page. 2. Complete the form fields. 3. Press “register”.	The system should register the organization successfully, route the user to the main account page, and make the Org. manager account active.
Tc-13	Check results when any field is empty, and the “Register” button is pressed.	1. Launch the organization register page. 2. Complete the form fields. 3. Press “register”.	Error message said “A field is missing”
Tc-14	Check the result when the user sign-up flow is interrupted and completed after a while.	1. Launch the system landing page. 2. Log in.	The system will detect the user sign-up flow current step and route them to it.

Test Case Scenario:		Sce-03: Check account management functionalities	
Test case id	Test case title	Test steps	Expected result
Tc-15	Check results when the user chooses “show account information”.	<ol style="list-style-type: none"> 1. Launch the application on the account page. 2. Choose “My Account”. 	The system will display account-related information.
Tc-16	Check results when the user chooses “update account information”	<ol style="list-style-type: none"> 1. Launch the application on the login page. 2. Choose “update”. 3. Enter the updated fields. 	The system will update the fields successfully.
Tc-17	Check results when a user chooses to “change password” with new passwords.	<ol style="list-style-type: none"> 1. Launch the application on the login page. 2. Choose “change password”. 3. Enter the old password, new password 	The system will successfully update the passwords.
Tc-18	Check results when a user chooses to “change password” with the wrong old password.	<ol style="list-style-type: none"> 1. Launch the application on the login page. 2. Choose “change password”. 3. Enter the old password and new password. 	An Error message said “ wrong old password”.
Tc-19	Check results when a user chooses to “change password” with a weak new password.	<ol style="list-style-type: none"> 1. Launch the application on the login page. 2. Choose “change password”. 3. Enter the old password and new password. 	An Error message said, “password must follow the system definitions”.

Test Case Scenario:		Sce-04: Manage organization members functionalities	
Test case id	Test case title	Test steps	Expected result
Tc-20	Check results when the user chooses to create a new organization member account with the correct email.	<ol style="list-style-type: none"> 1. Launch the application on the members' management page. 2. Choose “add members”. 3. Add members' accounts. 	The system will create the accounts successfully, generate a default password, and send it to the new member via email.
Tc-21	Check results when the user chooses to create a new organization member account with an existing email.	<ol style="list-style-type: none"> 1. Launch the application on the members' management page. 2. Choose “add members”. 3. Add members' accounts. 	Error message this email exists in the organization.
Tc-22	Check results when a new member login the system	<ol style="list-style-type: none"> 1. Launch the application on the login page. 2. Enter the default credentials. 	The system routes the user to the complete profile page.
Tc-23	Check results when a new member completes their profile, with verified data.	<ol style="list-style-type: none"> 1. Launch the application on the login page. 2. Enter the default credentials. 3. Complete the profile missing data 4. Choose “complete”. 	The member's account becomes “active” successfully

Tc-24	Check results when the Org. manager chooses to deactivate an existing member.	<ol style="list-style-type: none"> 1. Launch the application on the members' management page. 2. Choose “deactivate members”. Select the needed member. 	The member's account becomes “inactive” successfully
Tc-25	Check results when the Org. manager chooses to set member as admin.	<ol style="list-style-type: none"> 3. Launch the application on the members' management page. 4. Select a member. 5. Set the member as admin. 	The member's account becomes admin successfully means having the full access privileges over all projects.

Test Case Scenario:		Sce-05: manage the system by Sys-admin	
Test case id	Test case title	Test steps	Expected result
Tc-26	Check results when the system admin chooses to view reports	<ol style="list-style-type: none"> 1. Launch the application on the main system admin page. 2. Choose “view reports” 	The system will display the usage reports successfully

Test Case Scenario:		Sce-06: Check project management functionalities	
Test case id	Test case title	Test steps	Expected result
Tc-27	Check results when the organization manager creates a new project.	3. Launch the application on the project management page. 4. Choose "Create New Project". 5. Enter the project details.	The system will create a new project successfully.
Tc-28	Check results when the organization manager deactivates a project.	1. Launch the application on the project management page. 2. Choose an active project. 3. Select "Deactivate".	The system will deactivate the project successfully.
Tc-29	Check results when the organization manager views all projects related to an organization.	1. Launch the application on the project management page. 2. Choose "View All Projects".	The system will display all projects associated with the organization.
Tc-30	Check results when the organization manager assigns a member to a project.	1. Launch the application on the project management page. 2. Choose a project. 3. Select an org. member and assign them to the project.	The system will successfully assign the team members to the project, and give them default privileges.
Tc-31	Check results when the organization manager removes a member from a project.	1. Launch the application on the project management page. 2. Choose a project. 3. Select a team member and remove them from the project.	The system will successfully remove the team member from the project.
Tc-32	Check results when the organization manager views all members assigned to a specific	1. Launch the application on the project management page. 2. Choose a project. 3. Select "View Members".	The system will display all members associated with the project.

	project with their related privileges.		
Tc-33	Check results when the organization manager assigns privileges to a team member.	1. Launch the application on the organization management page. 2. Choose a team member. 3. Assign specific privileges over main sections in the system.	The system will update the privileges successfully.
Tc-34	Check results when the organization manager removes privileges from a team member.	1. Launch the application on the organization management page. 2. Choose a team member. 3. Remove assigned privileges.	The system will update the privileges accordingly.

Test Case Scenario:		Sce-07: Requirements Management Functionality.	
Test case id	Test case title	Test steps	Expected result
Tc-35	Check results of adding requirements from files of either “csv” or “excel” type.	1. Launch the requirements management interface. 2. Choose “add from file”. 3. Upload file. 4. Press “ok”.	The requirements added successfully.
Tc-36	Check results of adding requirements from files of neither “csv” or “excel” type.	1. Launch the requirements management interface. 2. Choose “add from file”. 3. Upload file and Press “ok”.	Error message “Wrong file format!”

Tc-37	Check the results of adding requirements manually, with valid data.	<ol style="list-style-type: none"> 1. Launch the requirements management interface. 2. Choose “add”. 3. enter data. 4. Press “add” 	Requirements added successfully.
Tc-38	Check the results of adding requirements manually, with missing data.	<ol style="list-style-type: none"> 1. Launch the requirements management interface. 2. Choose “add”. 3. enter data. 4. Press “add” 	Error message “Please complete all needed data”.
Tc-39	Check results of adding requirements manually, with the wrong datatype.	<ol style="list-style-type: none"> 1. Launch the requirements management interface. 2. Choose “add”. 3. enter data. 4. Press “add” 	Error message “wrong datatype”.
Tc-40	Check results for browsing all requirements for one project.	<ol style="list-style-type: none"> 1. Launch the requirements management interface for a project. 	The system should display all requirement backlogs for this project.
Tc-41	Check results for deleting a requirement.	<ol style="list-style-type: none"> 1. Launch the requirements management interface. 2. Choose one req and press “delete”. 	The requirement should be deleted successfully.
Tc-42	Check results for editing a requirement, with valid data.	<ol style="list-style-type: none"> 1. Launch the requirements management interface. 2. Choose one req and press “edit”. 	The requirement should be edited successfully.

		3. Add new data.	
Tc-43	Check results for editing a requirement, with invalid data.	<ol style="list-style-type: none"> 1. Launch the application by the product owner. 2. Launch the requirements management interface. 3. Choose one req and press “edit”. 4. Add new data. 	Error message “wrong requirements data”.
Tc-44	Check the results of adding requirements from files in the defined structure.	<ol style="list-style-type: none"> 1. Launch the application by the product owner. 2. Launch the requirements management interface. 3. Choose “add from file”. 4. Upload file. 5. Press “ok”. 	requirements added successfully.
Tc-45	Check the results of adding requirements from files that have a wrong structure.	<ol style="list-style-type: none"> 1. Launch the application by the product owner. 2. Launch the requirements management interface. 3. Choose “add from file”. 4. Upload file. 5. Press “ok”. 	Wrong structure “Please make it as “req title”, “req priority”.

2.5 Base management RTM – version 01:

Table 10 base management RTM - version 01

Req_ID	Use cases	analysis	System design	Detailed design	coding	Test cases
RE-FR-AM-01	Uc-01	<u>BM-analysis</u>				Tc-05 Tc-06 Tc-07 Tc-08
RE-FR-AM-02	Uc-02	<u>BM-analysis</u>				Tc-09 Tc-10
RE-FR-AM-03	Uc-02	<u>BM-analysis</u>				Tc-11
RE-FR-OM-04	Uc-03	<u>BM-analysis</u>				Tc-12 Tc-13
RE-FR-AM-05	Uc-01	<u>BM-analysis</u>				Tc- 14
RE-FR-AM-06	Uc-08	<u>BM-analysis</u>				Tc-01 Tc-02 Tc-03
RE-FR-AM-07	Uc-04	<u>BM-analysis</u>				Tc-15
RE-FR-AM-08	Uc-04	<u>BM-analysis</u>				Tc-16
RE-FR-AM-09	Uc-08	<u>BM-analysis</u>				Tc-04
RE-FR-AM-10	Uc-08	<u>BM-analysis</u>				Tc-04
RE-FR-AM-11	Uc-01 Uc-04	<u>BM-analysis</u>				Tc-17 Tc-18
RE-FR-AM-12	Uc-01 Uc-04	<u>BM-analysis</u>				Tc-19
RE-FR-OM-13	Uc-05	<u>BM-analysis</u>				Tc-20 Tc-21
RE-FR-AM-14	Uc-05	<u>BM-analysis</u>				Tc-22 Tc-23
RE-FR-AM-15	Uc-05	<u>BM-analysis</u>				Tc-23

RE-FR-OM-16	Uc-05	<u>BM-analysis</u>				tc-24
RE-FR-OM-17	Uc-16	<u>BM-analysis</u>				Tc-32
RE-FR-OM-18	Uc-06	<u>BM-analysis</u>				Tc-25
RE-FR-OM-19	Uc-16	<u>BM-analysis</u>				Tc-33
RE-FR-OM-20	Uc-16	<u>BM-analysis</u>				Tc-31
RE-FR-PM-21	Uc-13	<u>BM-analysis</u>				Tc-27
RE-FR-PM-22	Uc-14	<u>BM-analysis</u>				Tc-28
RE-FR-PM-23	Uc-09	<u>BM-analysis</u>				Tc-29
RE-FR-PM-24	Uc-15	<u>BM-analysis</u>				Tc-30
RE-FR-PM-25	Uc-16	<u>BM-analysis</u>				Tc-31
RE-FR-PM-26	Uc-09	<u>BM-analysis</u>				Tc-32
RE-FR-RM-27	Uc-11	<u>BM-analysis</u>				Tc-37 Tc-38 Tc-39
RE-FR-RM-28	Uc-11	<u>BM-analysis</u>				Tc-35 Tc-36 Tc-44 Tc-45
RE-FR-RM-29	Uc-10	<u>BM-analysis</u>				Tc-40
RE-FR-RM-30	Uc-12	<u>BM-analysis</u>				Tc-42 Tc-43
RE-FR-RM-31	Uc-10	<u>BM-analysis</u>				Tc-41
RE-FR-OM-32	Uc-18	<u>BM-analysis</u>				Tc-26

3. Service design and architecture:

This section provides a comprehensive overview of the design phase, covering service design, architecture determination, detailed design, and modeling. It highlights the significance of these steps in creating a robust service capable of managing multiple functionalities efficiently.

The Base Management Service serves as the core of the system and functions as an API Gateway, facilitating communication between the frontend and backend services. It handles essential operations, processes API requests, and integrates with other system microservices (the Notification Service, AI Service, and Meeting Service.)

3.1 Service Architecture:

System design involves defining the overall structure and architecture of the software, ensuring that all components interact and integrate seamlessly. This phase establishes the blueprint for an efficient data flow, interaction model, and integration strategy within the system.

One of the first constraints to consider when designing the architecture is non-functional requirements (quality attributes). Since the Base Management Service plays a central role in the system, it must meet the following key non-functional requirements:

- Extensibility: The system should allow new features or functionalities to be added with minimal changes to the existing codebase and without affecting

core components. It should seamlessly communicate with new services that may be integrated in the future.

- Security: The system must be able to protect all sensitive user data, ensuring robust authentication, authorization, strong password policies, and a well-structured architecture to prevent vulnerabilities.
- Performance: The system should respond to any action in less than 3 seconds, ensuring a smooth and responsive user experience.
- Maintainability: The system should allow modifications, bug fixes, and feature updates with minimal effort, ensuring that changes do not disrupt other components.

- Given these strict requirements, selecting the right architectural pattern is crucial to achieving a scalable, maintainable, and extensible system.

To address these non-functional requirements while maintaining a modular, scalable, and efficient system, the **Clean Architecture pattern** has been adopted for the Base Management Service.

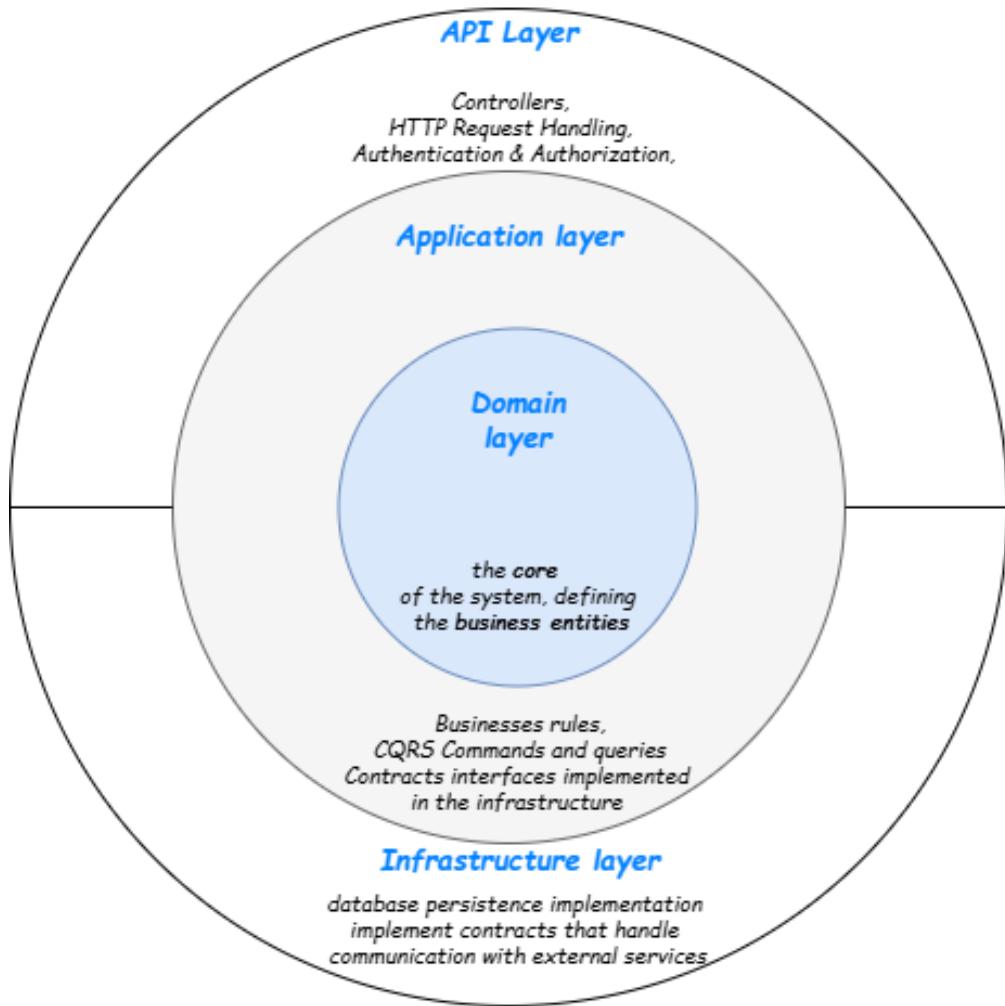


Figure 16 Base management service architecture

Benefits of Clean Architecture for This Service

1. Separation of Concerns – The system is divided into independent layers, each handling a specific concern, making the code clean, modular, and easier to maintain.
2. Scalability & Extensibility – New features can be added to the system without disrupting the existing architecture.
3. Security – Sensitive data is handled within a well-defined structure, and access control mechanisms are centralized within specific layers.
4. Testability – Since business logic is isolated from external dependencies, unit and integration testing become more effective.
5. Technology Independence – The core business logic remains decoupled from frameworks, databases, and external services, making future migrations and updates easier.

By implementing Clean Architecture, the Base Management Service maintains flexibility, robustness, and adaptability, making it ideal for microservice-based systems.

- CQRS + Mediator Pattern:

Since this service functions as an API Gateway and a core service, it handles a large number of requests and integrates with multiple external services. Using CQRS (Command Query Responsibility Segregation) alongside Mediator brings several advantages:

- Separation of Read and Write Operations – Queries (read operations) and commands (write operations) are handled separately, optimizing database performance and scalability.
- Improved Performance – By separating read models from write models, queries can be optimized independently, reducing database load and improving response time.
- Scalability – Read-heavy operations and write-heavy operations can scale separately based on system needs.
- Decoupling Components – The Mediator pattern eliminates direct dependencies between handlers and controllers, making the system more modular and testable.

By combining Clean Architecture with CQRS + Mediator, the Base Management Service ensures a highly scalable, high-performance, and maintainable system.

CQRS Flow Diagram:

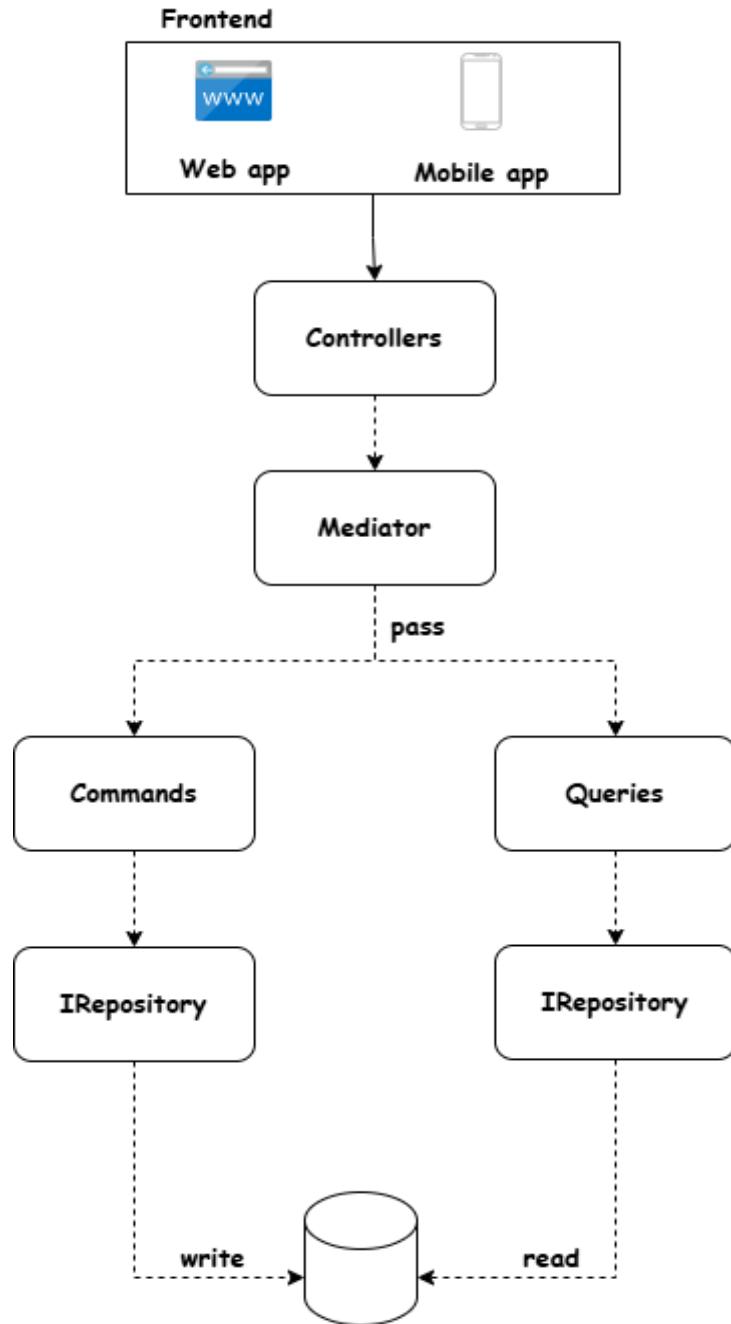


Figure 17 CQRS flow

Service Layers and Responsibilities:

The Base Management Service is structured into multiple layers, each responsible for specific tasks:

1. API Layer (Presentation Layer):

- Handles HTTP requests via controllers.
- Exposes RESTful APIs for communication between the frontend and backend.
- Implements authentication, authorization, and request validation.
- Routes API requests to the Application Layer.

2. Application Layer

- Contains business logic and implements CQRS + Mediator.
- Handles commands and queries separately, improving performance and scalability.
- Defines service contracts (interfaces) for interacting with external services.
- Ensure that business rules are decoupled from infrastructure concerns.

3. Domain Layer (Core Business Logic)

- Contains database entities, aggregates, and business rules.

4. Infrastructure Layer

- The main functionality to manage and implement the external dependencies such as databases persistence, message brokers, and third-party APIs to communicate with other system services in our system (AI service, meeting service, Notifications service).

By utilizing Clean Architecture, CQRS, and Mediator, the Base Management Service achieves:

- High maintainability through modular design
- Improved performance by optimizing read/write operations separately.
- Seamless extensibility to integrate new services without major changes.

This approach ensures that the Base Management Service remains reliable, flexible, and efficient, serving as a robust foundation for the entire system.

3.2 Used design patterns:

1. Strategy design pattern – define a family of file parser:

The Strategy Pattern is used to define a family of file parsers, encapsulate them, and make them interchangeable. Instead of handling file parsing logic in a single class, each file format (CSV, Excel, etc.) has its own dedicated strategy.

The use of the strategy design pattern improves the system:

- Encapsulation of Parsing Logic – Each file format has a separate class, making the code more modular.
- Easier Maintenance & Extensibility – New file formats (e.g., JSON, XML) can be added without modifying existing logic.
- Single Responsibility Principle (SRP) – Each class handles only one file type, improving clarity and maintainability.

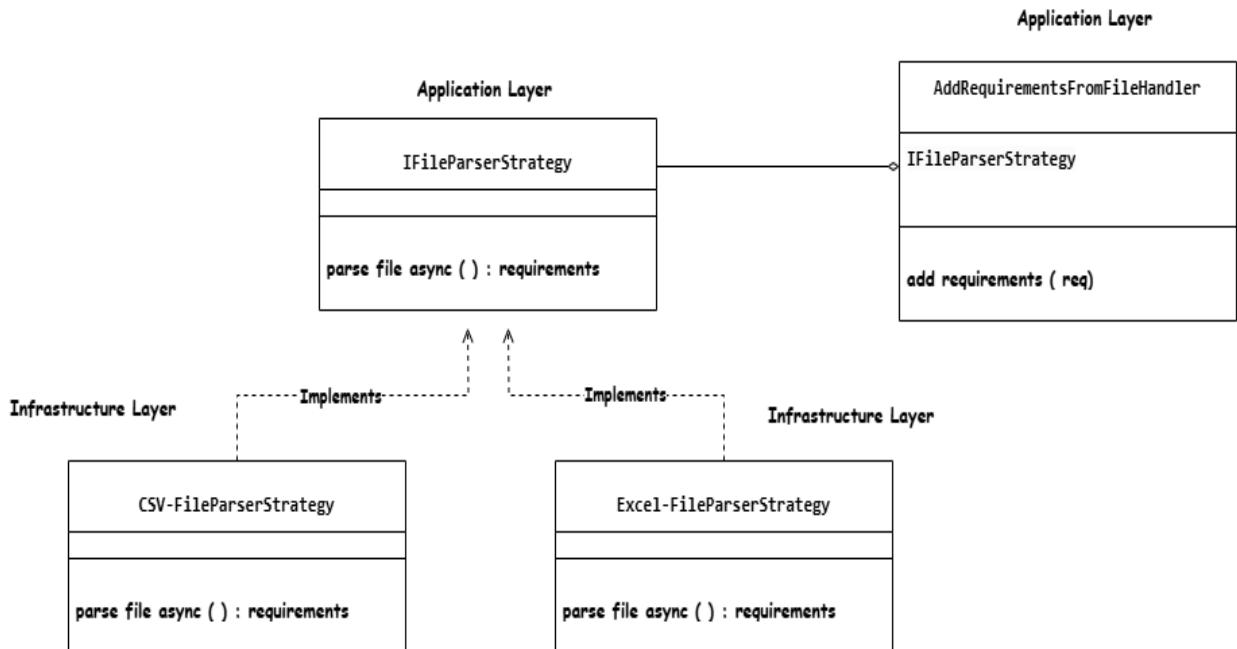


Figure 18 strategy design pattern

- IFileParserStrategy Interface defines the contract for all file parsers.
- ExcelFileParserStrategy & CsvFileParserStrategy implement the interface to parse specific file formats.
- The application layer interacts with these parsers through the interface, without worrying about file-specific logic.

2. Factory design pattern - dynamically select the appropriate file parser strategy:

The Factory Pattern is used to dynamically select the appropriate file parser strategy based on the file type. Instead of manually choosing a parser, the Factory Pattern automates the selection process.

The use of the factory design pattern improves the system:

- Centralized Decision-Making – The factory decides which parser to use, keeping the main application logic clean.
- Reduces Code Duplication – No need for if-else conditions scattered across the codebase.
- Easy to Extend – Adding support for a new file type requires only a new parser class and a small update in the factory.

- The file parsing process after using factory design pattern:

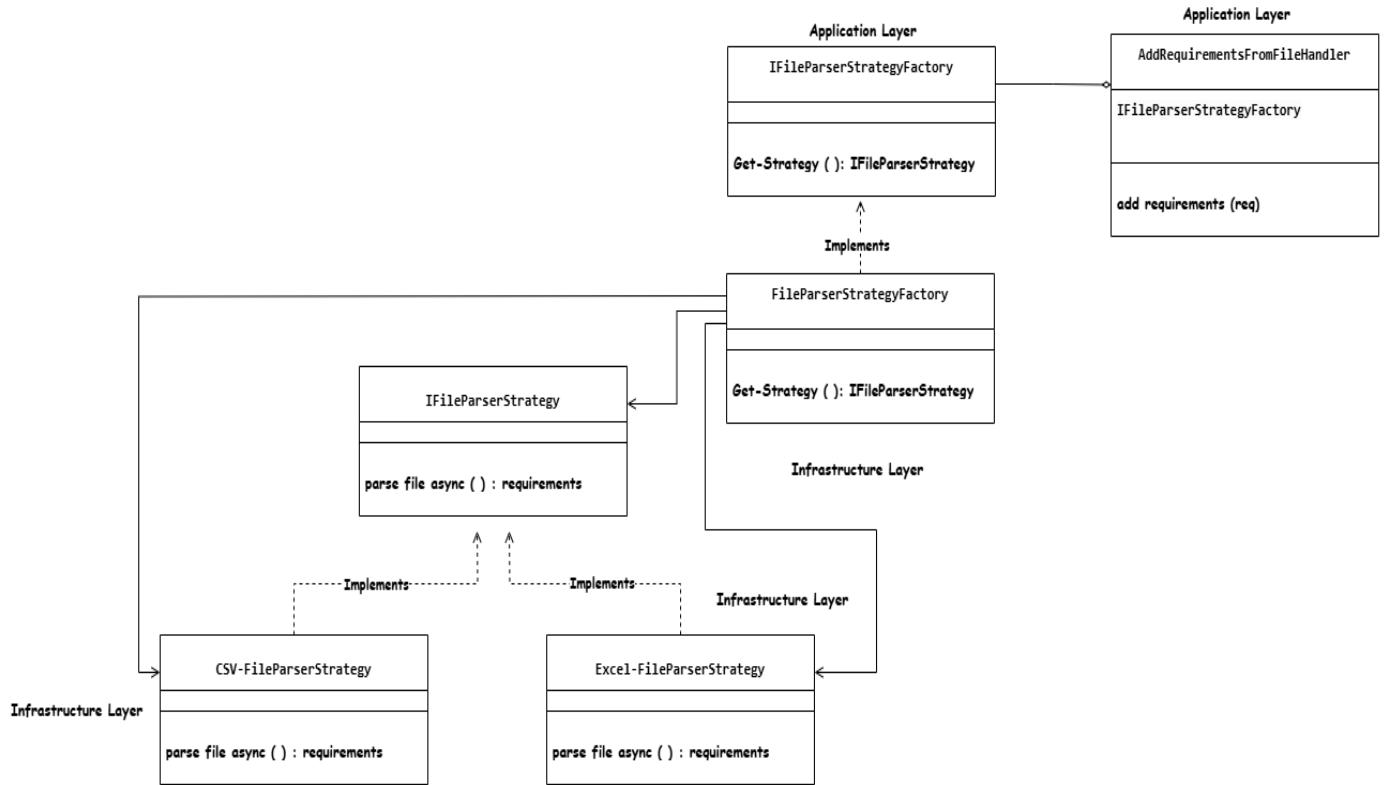


Figure 19 Factory design pattern

- `FileParserStrategyFactory` determines the correct strategy based on file extension (`.csv`, `.xlsx`).
- Returns the corresponding parser (`CsvFileParserStrategy`, `ExcelFileParserStrategy`).
- The application layer only interacts with `IFileParserStrategy`, ensuring loose coupling.

- Authorization solution - design:

The Base Management Service follows a privilege-based authorization model that allows users to have different access levels at both the organization-wide and project-specific levels. This approach ensures that users can only access resources and actions based on their assigned privileges.

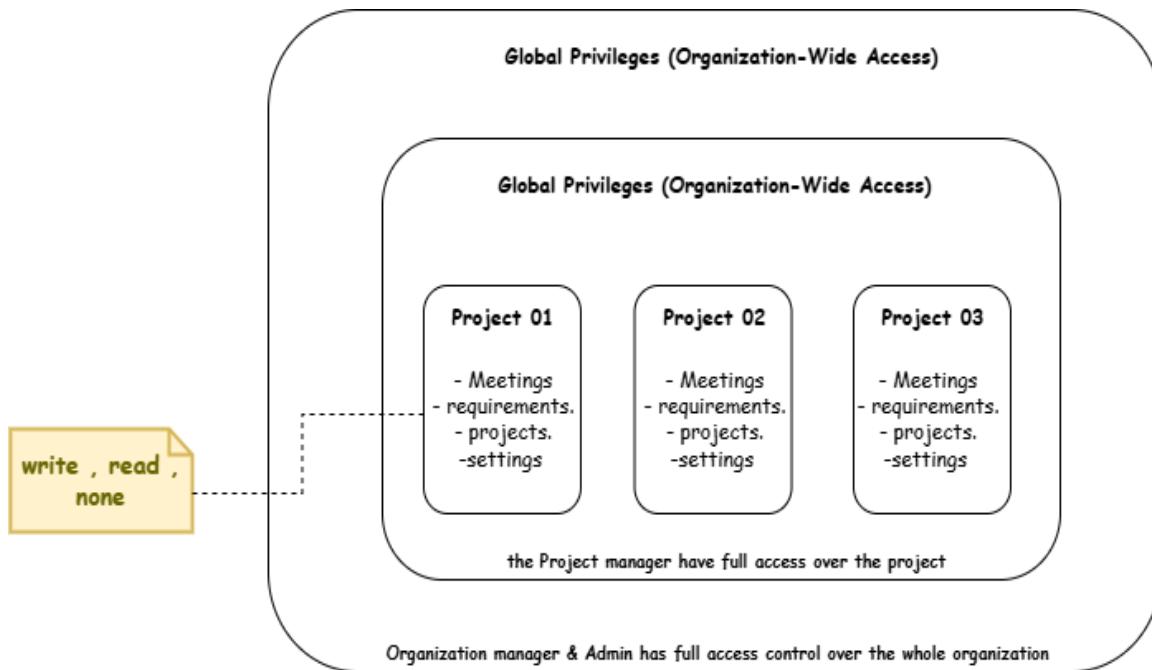


Figure 20 privileges design

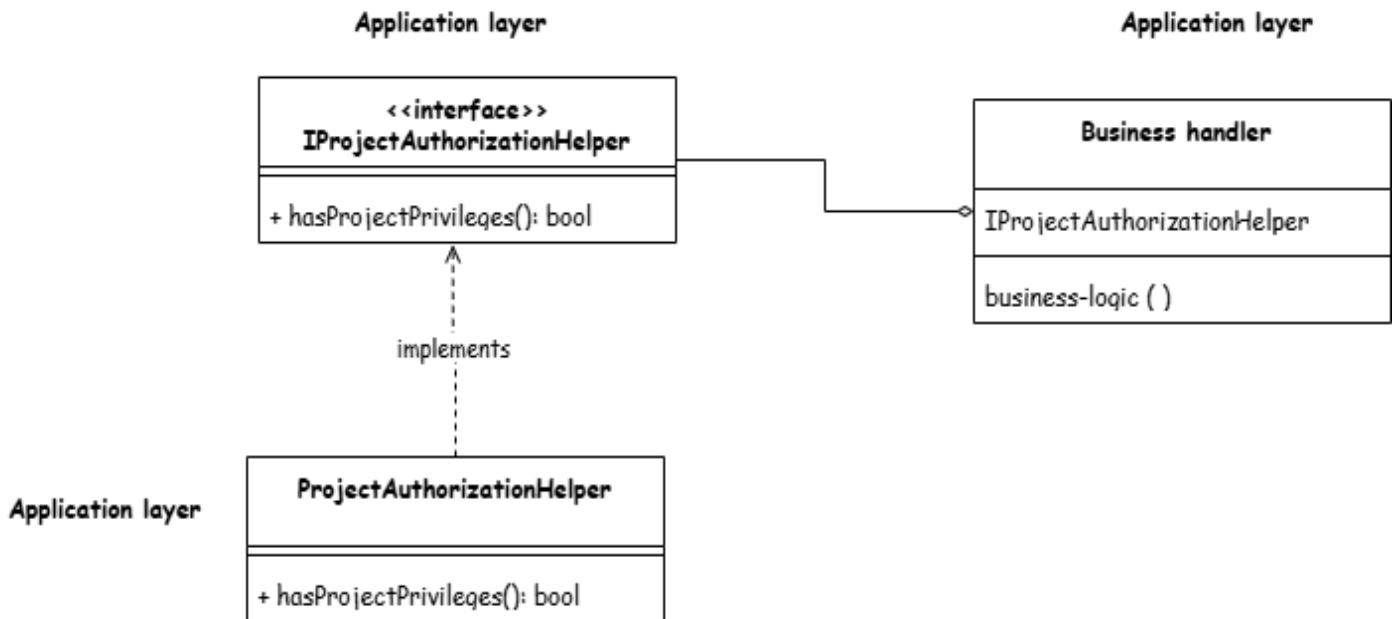
Global Privileges (Organization-Wide Access):

- Admin & Organization Manager: Have full access across all projects and organization settings.
- They do not need per-project privileges, as they have automatic access to everything.
- Members have no privileges before adding to a project.

Local Privileges (Project-Specific Access):

- Organization Members
- Their access is restricted to specific projects.
- They have different privilege levels for each aspect of a project.

In our system the authorization logic handled in the business layer more manually to achieve domain related privileges and authorization



To enforce this privilege-based authorization, the `ProjectAuthorizationHelper` class is used. It follows the Authorization Helper Pattern, ensuring that authorization logic remains centralized and reusable across the application. Since authorization is an application concern, the `ProjectAuthorizationHelper` is implemented in the Application Layer and used in Command & Query Handlers (CQRS).

Chapter 3 – BaseMgt Service

- Data base schema design – data modeling:

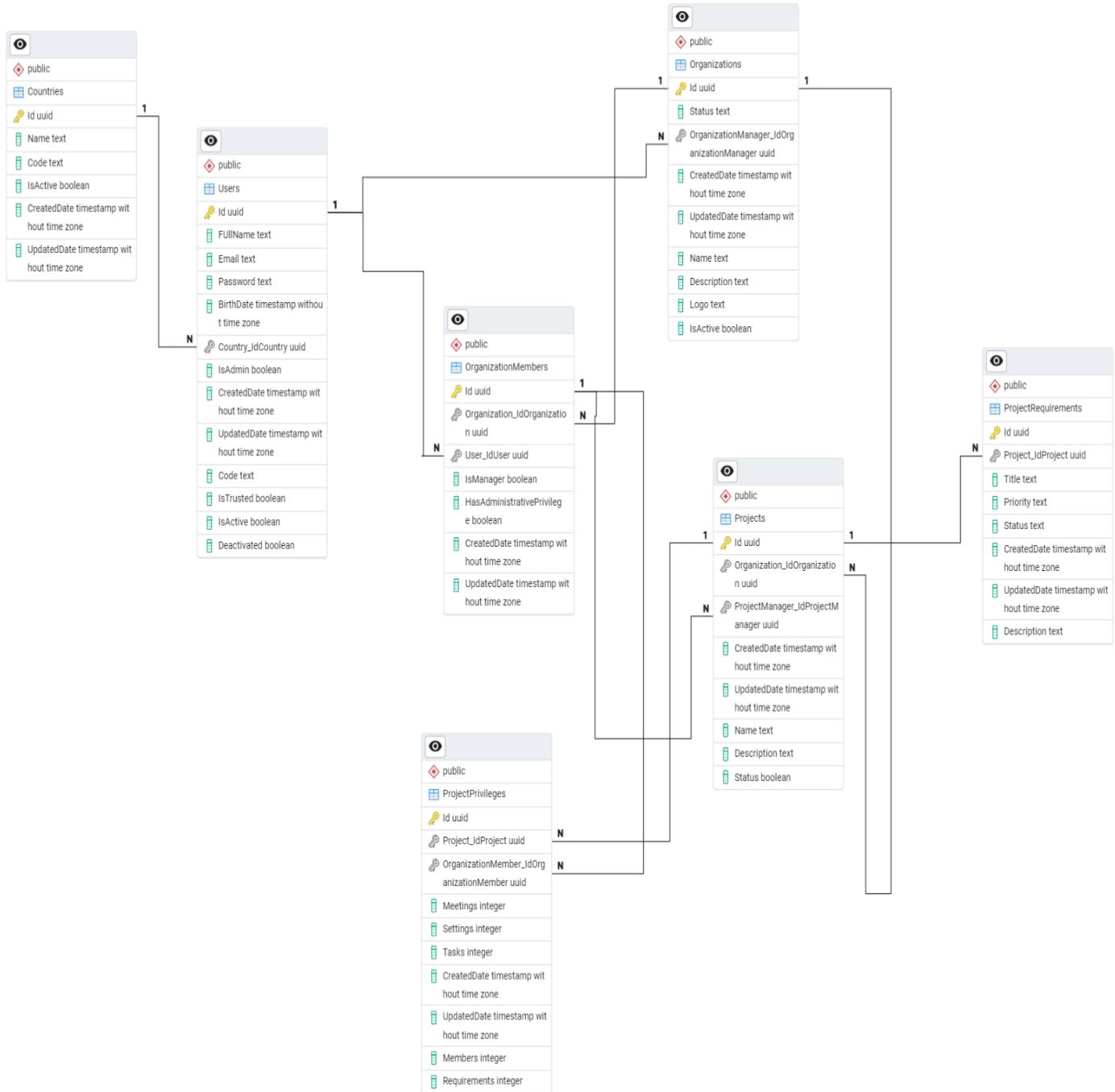


Figure 21 data modeling

4. Service practical implementation:

In this section, we will present the practical implementation of the base management service, detailing the technologies and tools employed. Additionally, we will showcase the system interfaces and conclude by executing test cases to verify the system's functionalities across different scenarios.

4.1 Used tools and techniques:

- [React:](#)

React is a popular JavaScript library for building user interfaces. It was created by Facebook and is widely used in web development. React allows developers to build reusable UI components that can efficiently update and render changes to the user interface when the underlying data changes.

React's primary focus is on building user interfaces, and it excels in creating interactive and dynamic web applications.

- [Asp.net:](#)

ASP.NET Core is used as the backend framework for developing scalable, high-performance web APIs and services. It integrates with MediatR for CQRS, FluentValidation for request validation, and Entity Framework Core for data persistence.

- [Flutter:](#)

Flutter is used to develop the mobile application for the project, providing a cross-platform solution with a single codebase. It ensures smooth user experience with high performance across both iOS and Android devices.

- **PostgreSQL Database:**

PostgreSQL is an open-source relational database management system (RDBMS) renowned for its advanced features, reliability, and scalability, making it a preferred choice for storing, managing, and retrieving data – system main database.

- **Visual studio code (VS code):**

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

We use it to develop the whole project (frontend, backend).

- **GitHub:**

GitHub is a Git-based source code hosting platform that enables version control and collaborative software development. It provides tools for code management, issue tracking, and continuous integration, streamlining the development lifecycle and ensuring efficient team collaboration [repository link](#).

- **Postman:**

is an open-source desktop application that takes the pain out of interacting with and designing, debugging, and testing APIs, we use it to test our APIs.

- **Diagrams:**

Web-based applications are used for modeling requirements and creating necessary charts, ensuring clear documentation and visualization of system components.

4.2 Service Interfaces – mobile application:

- Log-in interfaces:

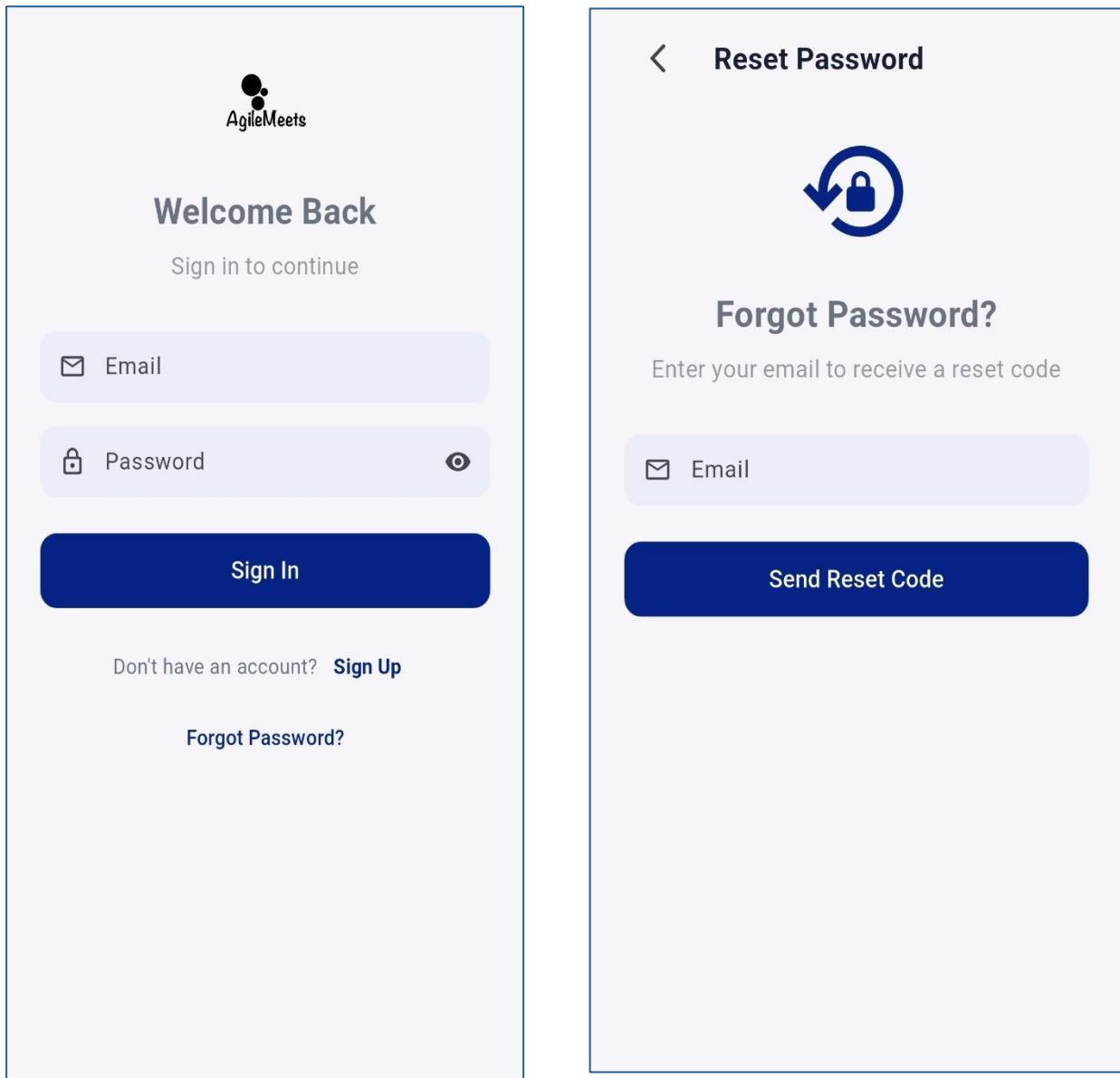
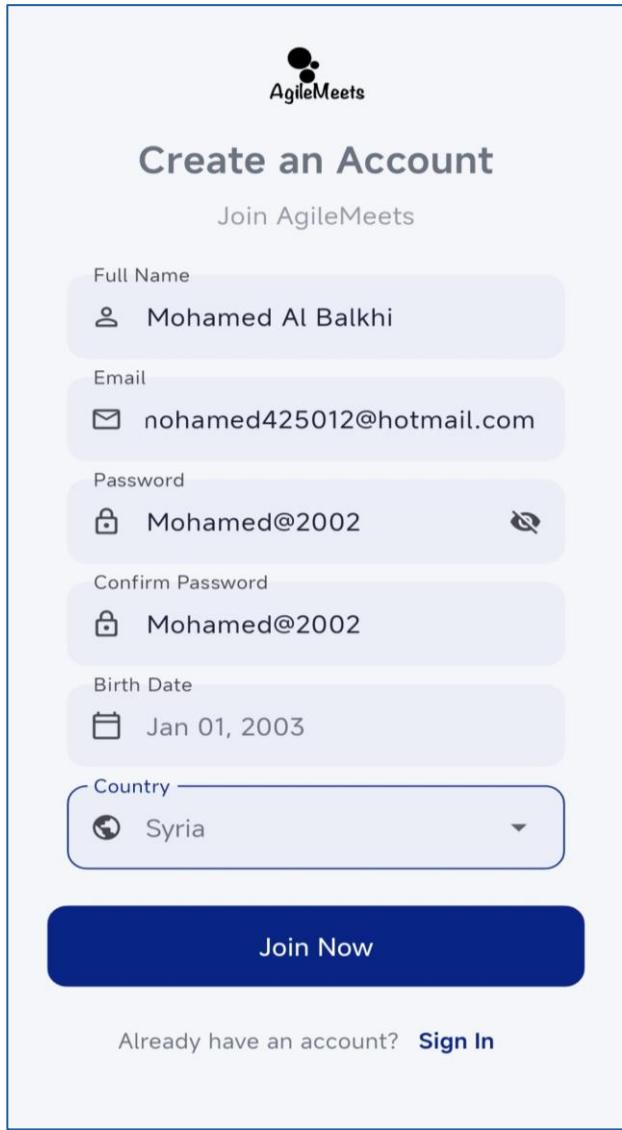


Figure 22 forget password interface

- Sign-up interfaces:



The image shows the 'Create an Account' form for AgileMeets. At the top is the AgileMeets logo. Below it is the title 'Create an Account' and a sub-instruction 'Join AgileMeets'. The form contains several input fields with rounded corners and shadows:

- Full Name: Mohamed Al Balkhi
- Email: nohamed425012@hotmail.com
- Password: Mohamed@2002
- Confirm Password: Mohamed@2002
- Birth Date: Jan 01, 2003
- Country: Syria

A large blue 'Join Now' button is at the bottom. Below it, a link says 'Already have an account? [Sign In](#)'.

Figure 24 sign up -step 01 interface

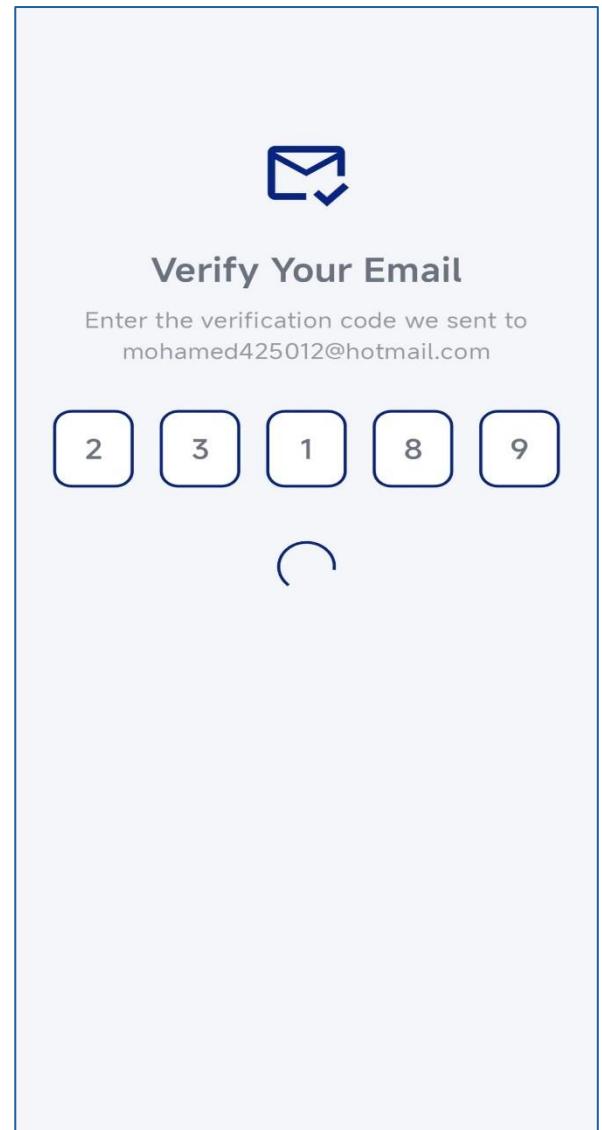


Figure 25 sign-up step 02 interfaces

- Register organization – sign up step 03:

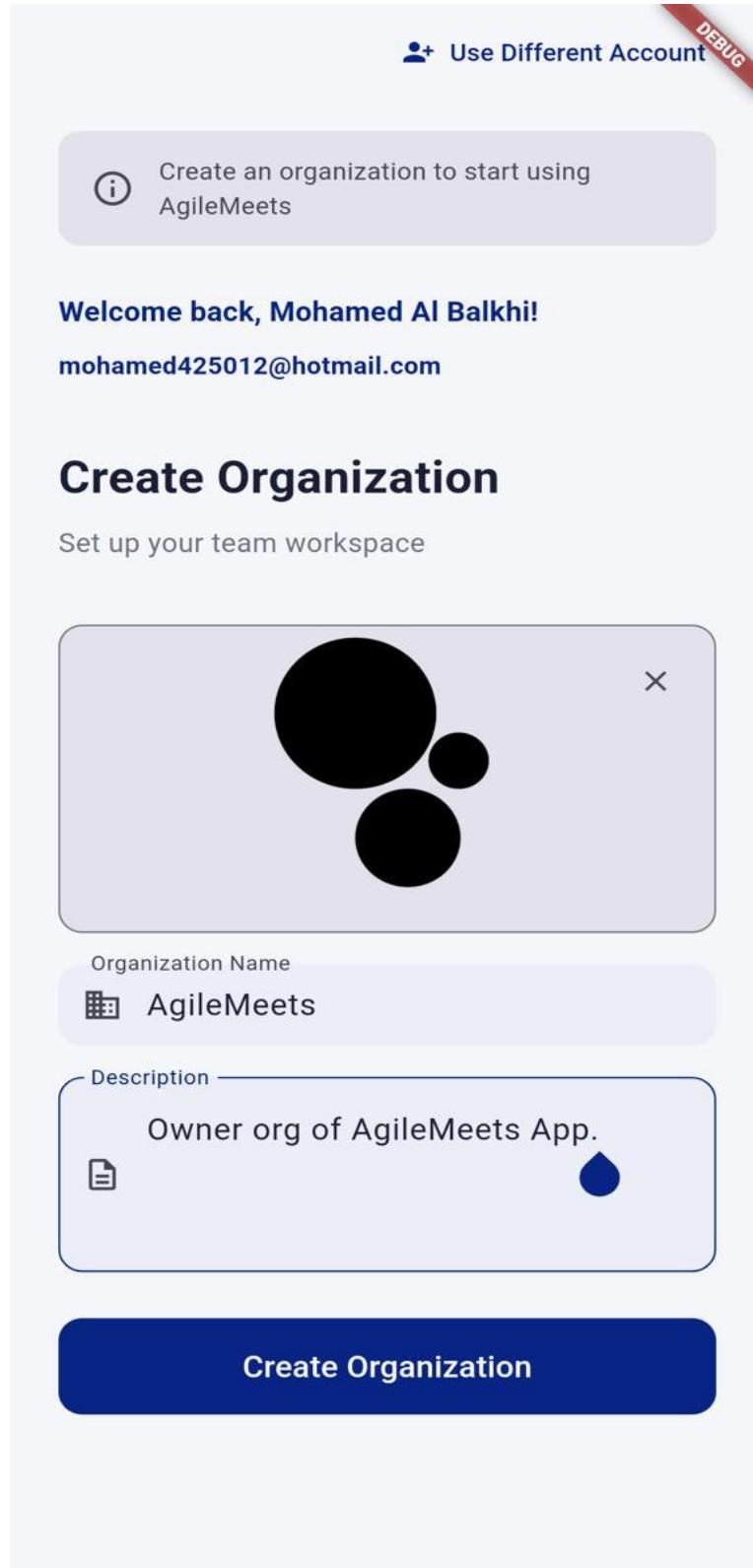


Figure 26 Registration step3 screen

- App Main page interface:

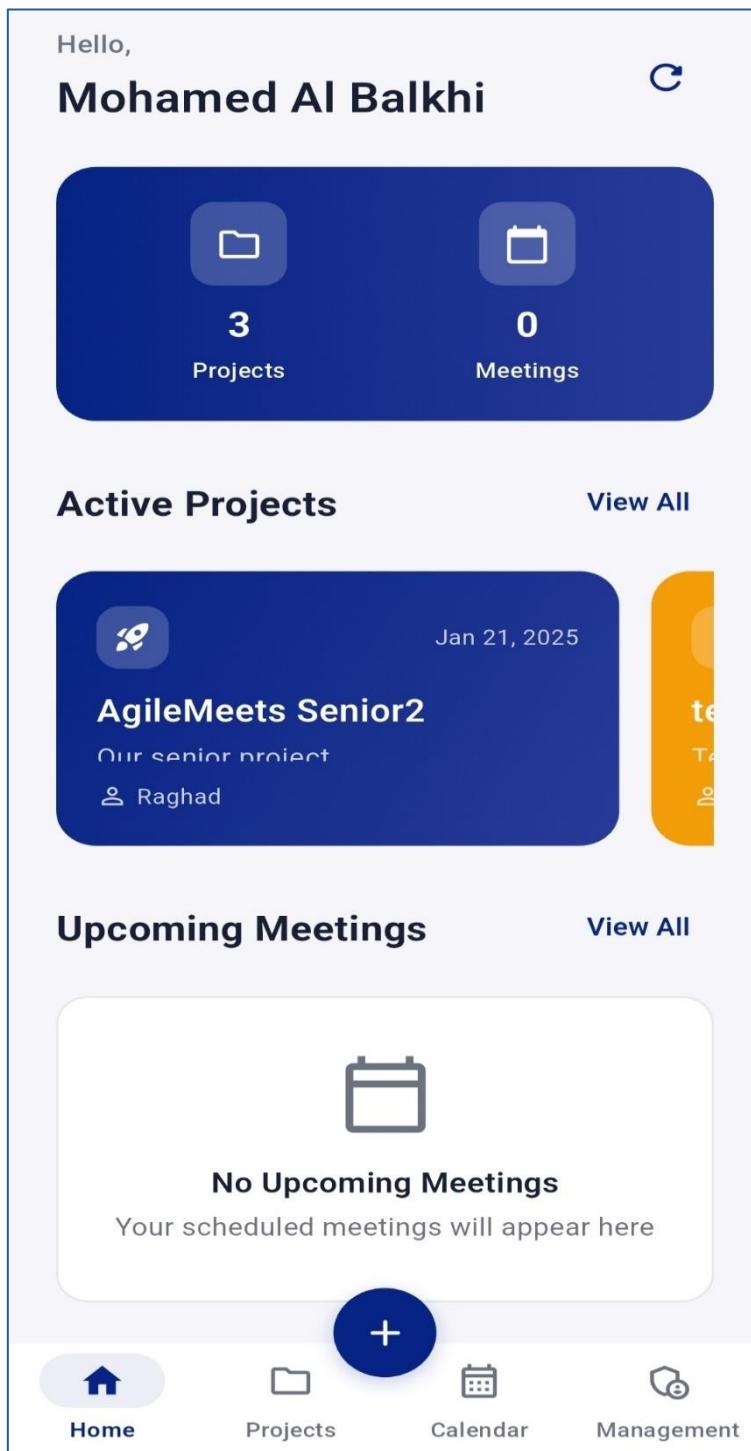


Figure 27 mobile app - main page

- Managing organization:

Management

Organization Overview

7

Total Members

3

Projects

3

Admins

1

Pending

Organization Members

[View All](#)

Member ID	Name	Email	Role
M	Mohamed Al Balkhi	mohamedbalkhi169@gmail.com	Owner
A	Abdullah Al Balkhi	mohamedx2755@gmail.com	Member
N	NewUser	ferasfawal1@gmail.com	Pending Invited
A	Abood old Tab	mohamedx2744@gmail.com	Member
A	Amani	amanibal1980@gmail.com	Admin
R	Raghad	raghodalhosny@gmail.com	Admin
A	Anas AbdulAziz	anas.anoos123456789@gmail.com	Admin

Recent Projects

[View All](#)

Project Name	Description	Created	Action
test 2	Testing	2 days ago	>
test	test	29 days ago	>
AgileMeets Senior2	+	about a month ago	>

[Home](#) [Projects](#) [Calendar](#) [Management](#)

Figure 29 Organization overview

Organization Members

Member ID	Name	Email	Status
M	Mohamed Al Balkhi	mohamedbalkhi169@gmail.com	Owner
A	Abdullah Al Balkhi	mohamedx2755@gmail.com	Member
N	NewUser	ferasfawal1@gmail.com	Pending Invited
A	Abood old Tab	mohamedx2744@gmail.com	Member
A	Amani	amanibal1980@gmail.com	Admin
R	Raghad	raghodalhosny@gmail.com	Admin
A	Anas AbdulAziz	anas.anoos123456789@gmail.com	Admin

[+ Add Member](#)

Figure 28 manage org. members interface

- Manage Projects – interfaces:

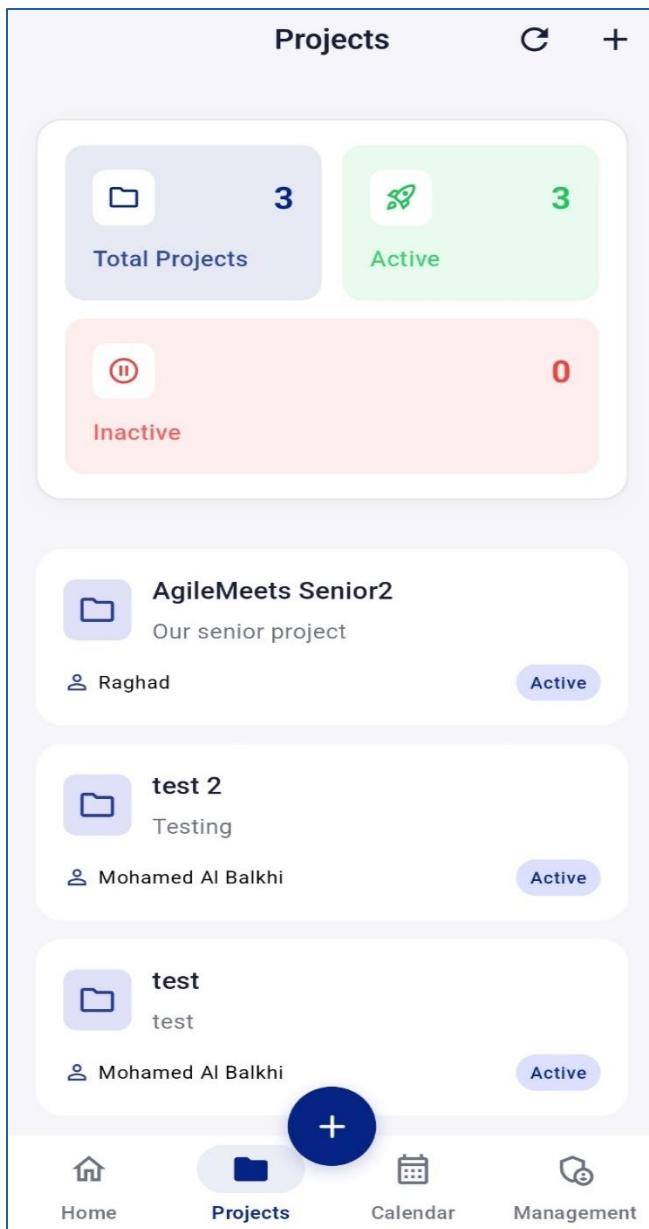


Figure 30 projects management interface

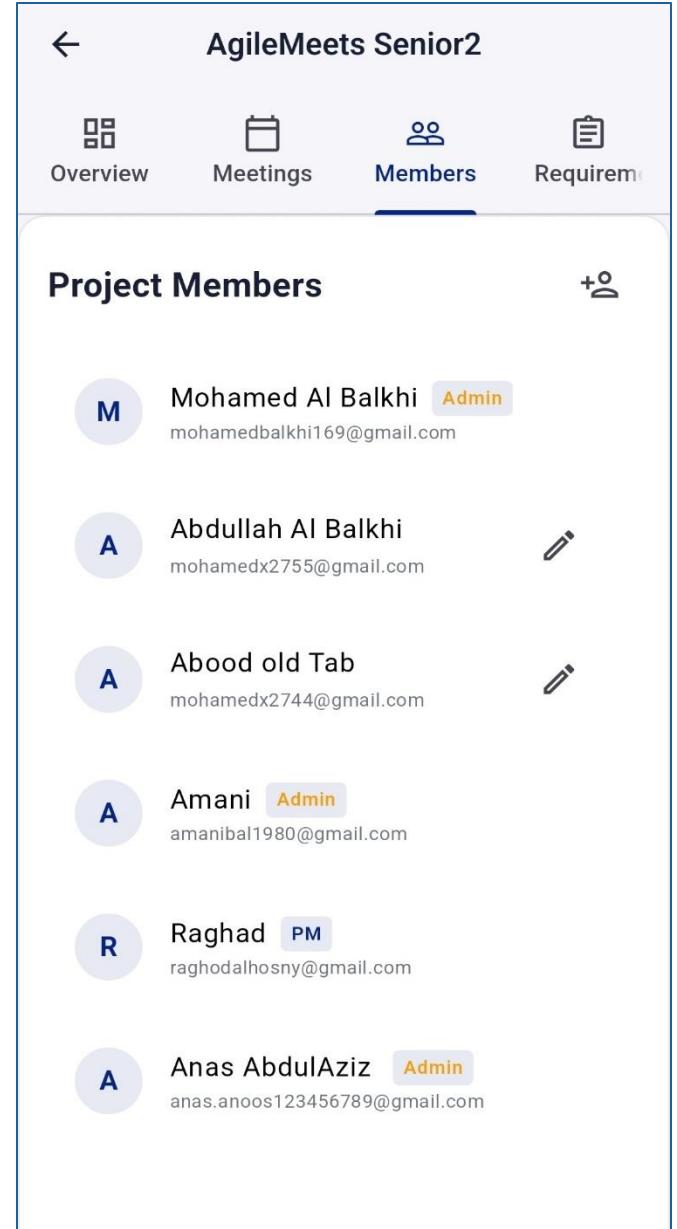


Figure 31 manage project members interface

- Manage requirements:

AgileMeets Senior2

Overview Meetings Members Requirements

Search requirements... Filters

- test3 testing Completed
- test1 testing New
- test2 with very long title bla bla bla bye In Progress
- test2 testing In Progress
- test1 testing New
- test3 testing Completed
- test1 testing New

Add Requirements

Manual File

Title

Description

Priority

Low Medium High

Status

New In Progress Completed

+ Add Requirement

Figure 33 view project backlog - interface

Figure 32 add requirement manually- interface

- Manage requirements:

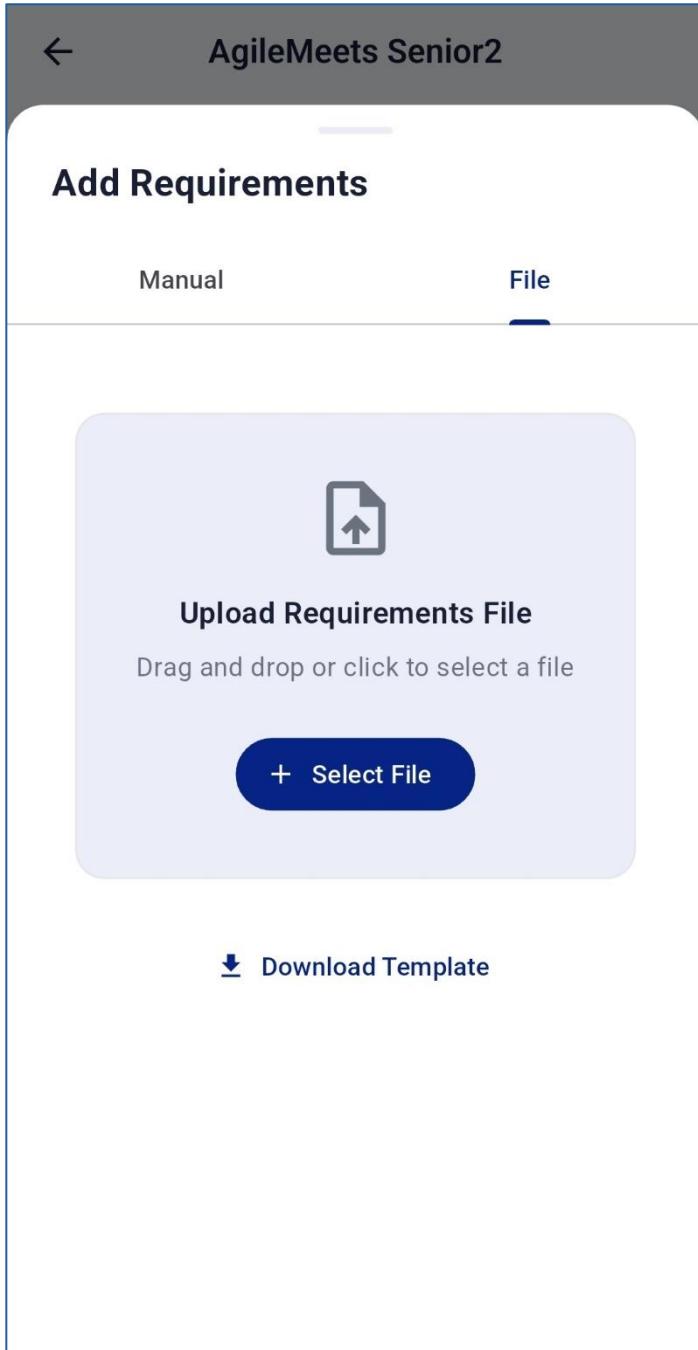


Figure 35 add requirements from file

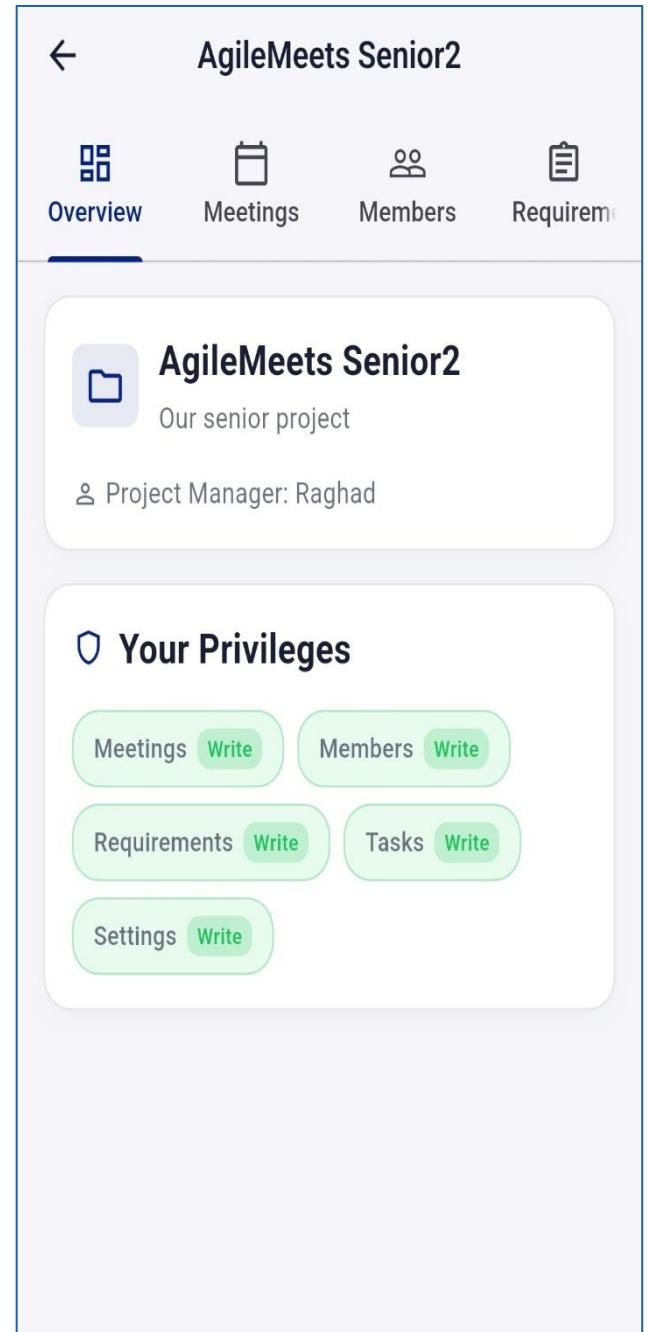


Figure 34 project interface

Manage privileges:

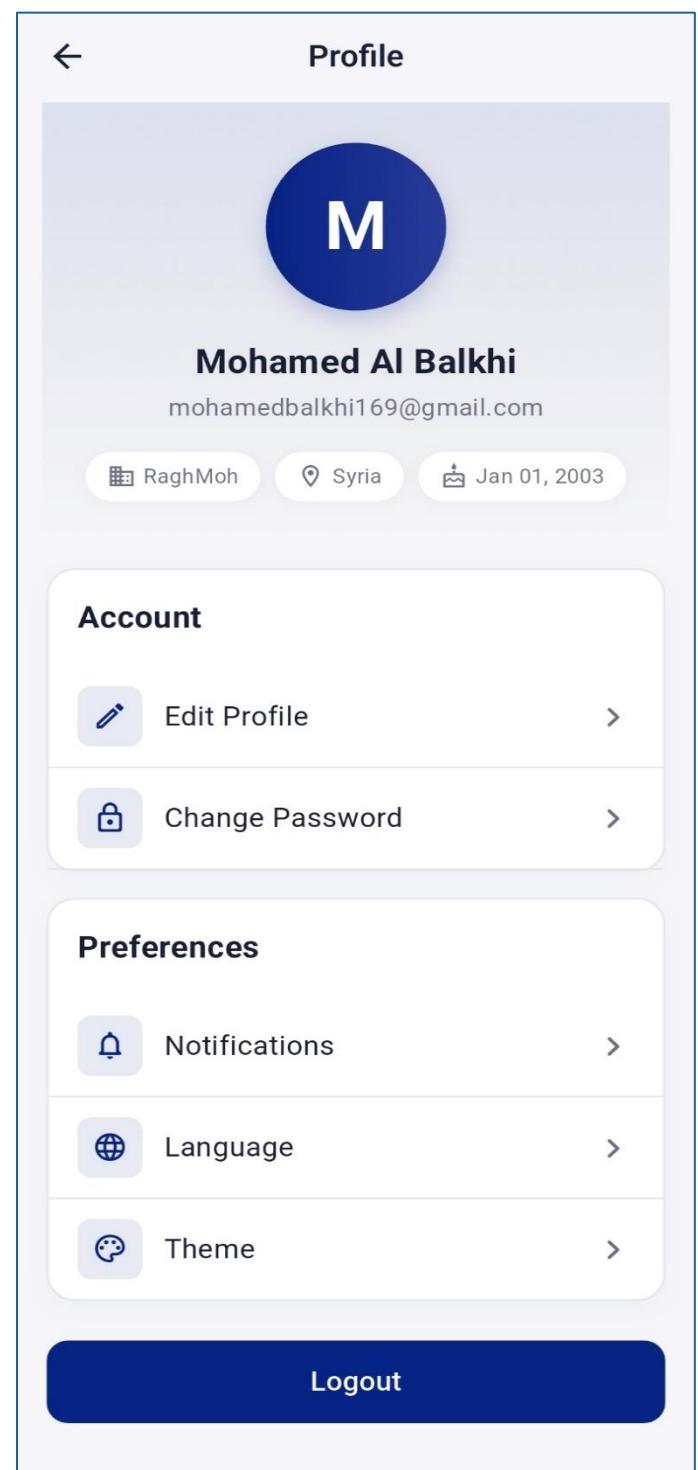
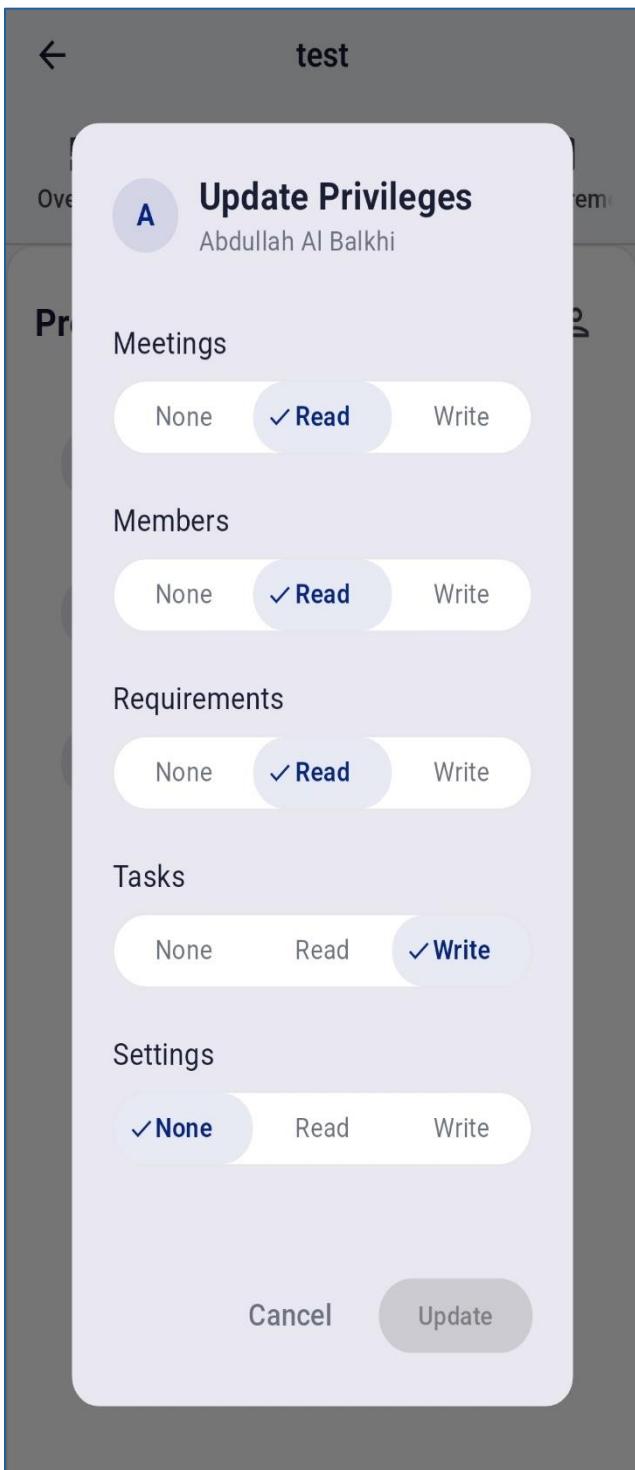


Figure 37 manage privileges interfaces

Figure 36 account management interface

4.3 Service Interfaces – web application:

System Landing page:

The screenshot shows the homepage of the AgileMeets web application. At the top, there is a navigation bar with links for "About", "Services", "Examples", "Contact", and a "Login" button. The main title "AI Agile Meet" is displayed prominently. Below the title, a brief description states: "A platform designed to streamline meeting management for companies following Agile methodologies. It provides tools to organize meetings and efficiently manage their artifacts, ensuring better collaboration and productivity". A "Join Us As A Company" button is located below this text. To the right of the text, there is an illustration of two people working at desks with laptops, surrounded by three document icons. Three callout boxes are present: one for "Service at any time" (with a clock icon), one for "high meeting quality" (with a calendar icon), and one for "Meetings artifacts management" (with a document icon). Each callout box includes a "learn more" button. Below these boxes, the section "Our AI Services" is introduced, followed by a detailed description of the "AI Meeting Processing" service, its features, and an illustration of a person interacting with a large white AI robot.

AI Agile Meet

A platform designed to streamline meeting management for companies following Agile methodologies. It provides tools to organize meetings and efficiently manage their artifacts, ensuring better collaboration and productivity

Join Us As A Company

Service at any time

our service is available 24/7 to help your team

[learn more](#)

high meeting quality

We ensure that your meetings are productive and efficient

[learn more](#)

Meetings artifacts management

We manage your meetings artifacts and ensure that they are up to date

[learn more](#)

Our AI Services

AI Meeting Processing

Our AI service is a powerful tool that processes meeting recordings to enhance productivity. It converts speech to text (STT), generates summaries, and extracts key points, making Agile meeting artifacts more accessible and actionable for teams.

- Speech-to-Text Conversion
- Automated Meeting Summary
- Smart Action Item Detection
- AI-Powered Analysis

96

Our Meeting Services

Robust Meetings

Our meeting service supports both online and in-person meetings, ensuring seamless collaboration for Agile teams. It enables efficient scheduling, recording, and management of meetings, tailored to enhance productivity.

-  In-Person Meeting Support
-  Virtual Meeting Integration
-  Smart Scheduling System
-  Real-time Collaboration



Manage your Organization



Service for your Organization

Our organization management service ensures robust role-based access control, allowing only authorized individuals to manage sensitive tasks. With clearly defined roles and privileges, it provides a secure and efficient way to organize and delegate responsibilities.

-  Role-Based Access Control
-  Team Management
-  Task Assignment
-  Progress Tracking

AgileMeets team

Expert software engineers at your service

Quick Links

- [About Us](#)
- [Services](#)
- [Contact](#)

Contact Info

-  Damascus, Syria
-  +963 958 677 568
-  raghodalhosny@gmail.com

system landing page - About-us page:

 AgileMeets

About Services Examples Contact [Login](#)

About Us

We are a team of dedicated developers committed to creating innovative software solutions.

Our Team



Mohamed al-balkhi
Full Stack Developer
Expert in building scalable web applications with modern technologies and architectures. Specialized in backend development with asp.net, mobile development with flutter, devops, meeting management.



Raghad al-hosny
Full Stack Developer
Expert in building scalable web applications with modern technologies and architectures. Specialized in backend development with asp.net, web development with react, AI Services.

AgileMeets team
Expert software engineers at your service

Quick Links
[About Us](#)
[Services](#)
[Contact](#)

Contact Info
 Damascus, Syria
 +963 958 677 568
 raghodalhosny@gmail.com

© 2025 RAG-MOH . All rights reserved.

Sign up full flow – step 01:

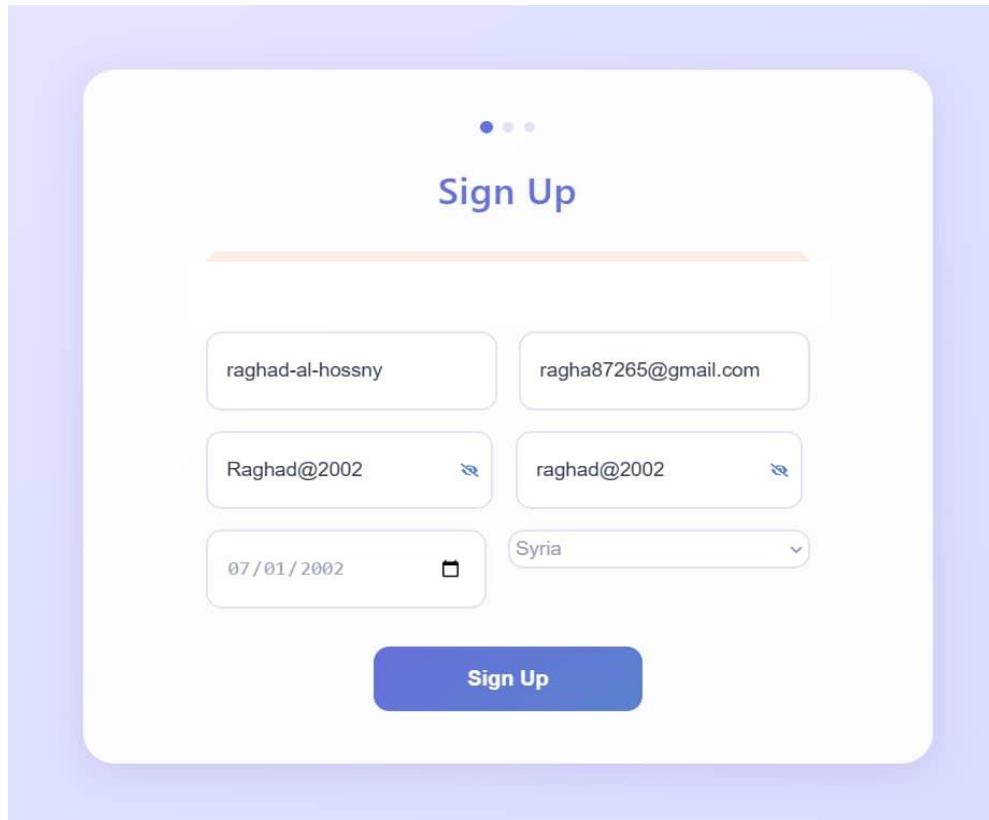


Figure 38 sign up - web interface

Sign up full flow – step 02:

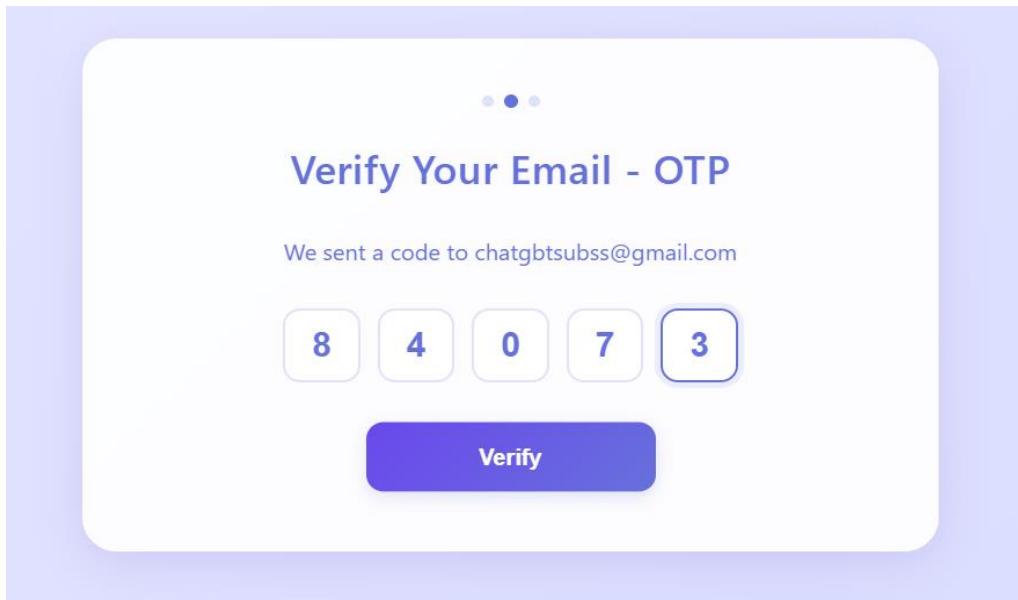


Figure 39 verify email - web interface

Sign-up full flow – step 03:

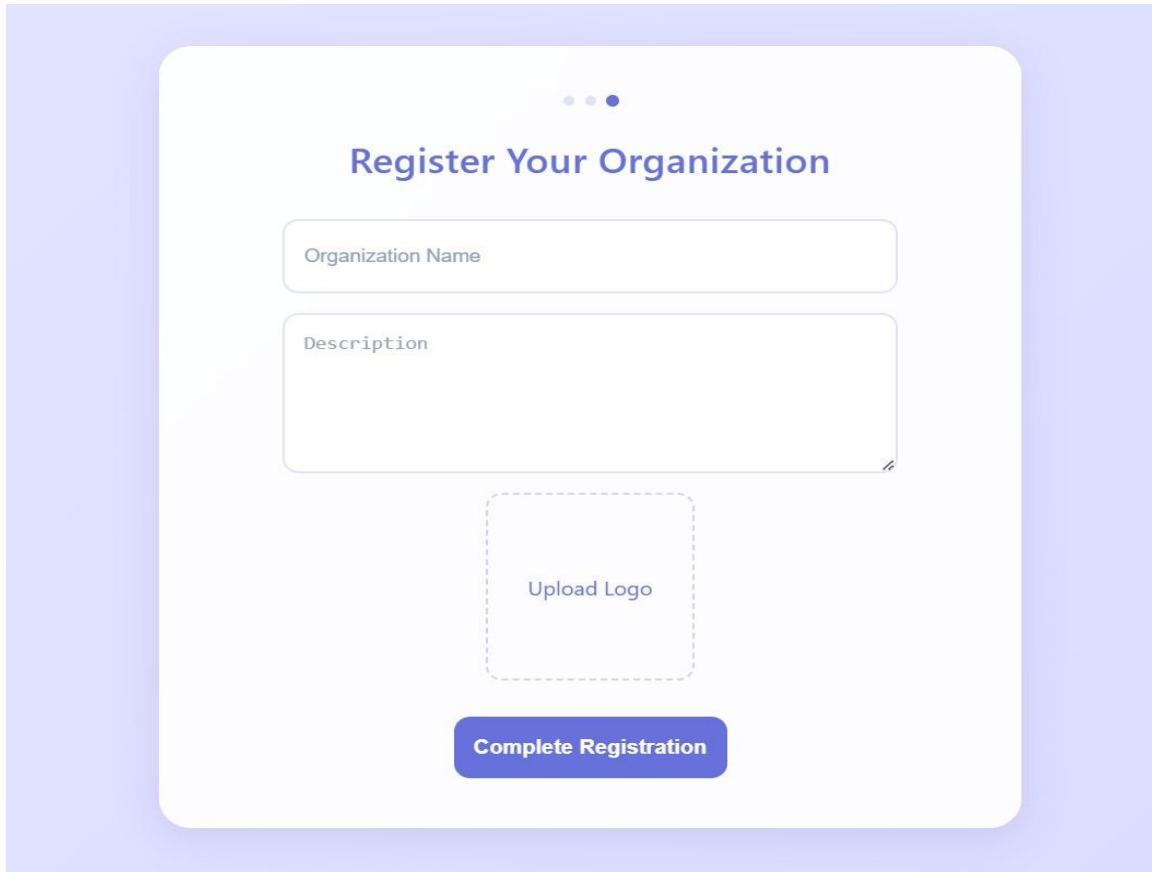


Figure 40 register organization - web interface

Login:

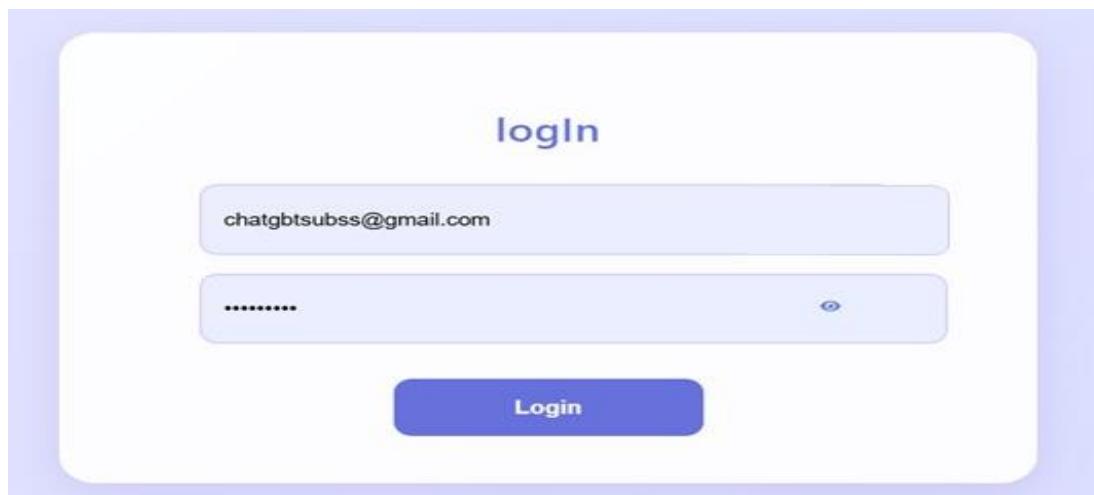
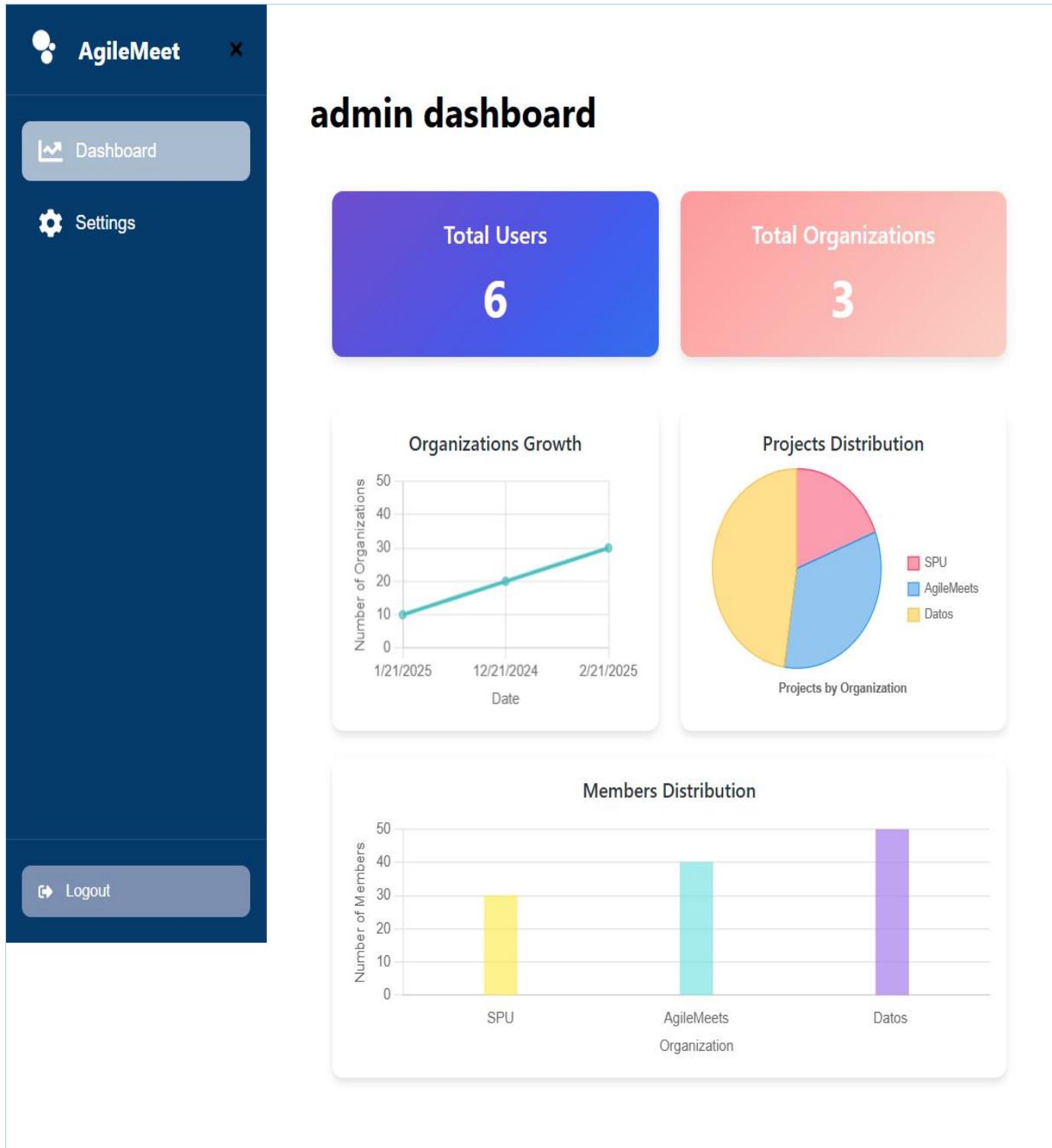


Figure 41 login - web interface

System Admin account – admin dashboard:



4.4 Test cases execution:

Table 11 test cases execution

TC id	Test case title	Tested data	Expected result	Actual result	Pass / fail
Tc-01	Check results on entering a valid email and password.	email="raghad@gmail.com" Password="Raghad124"	Login successfully.	Login successfully.	Pass
Tc-02	Check results on entering an invalid email, or password.	Email="raghad" Password="231"	Error message "invalid data"	Error message "invalid data"	Pass
Tc-03	Check results when a user email or password is empty, and the "login" button is pressed.	Email= Password=Loki1234	Error message "a field is missing"	Error message "a field is missing"	Pass
Tc-04	Check results when user chooses "forget password".	Email='raghodalhosny@gmail.com'	The system will send an OTP to verify user email, if the email verified the password will updated successfully.	The system will send an OTP to verify user email, if the email verified the password will updated successfully.	Pass

Tc-05	Check results on completing the first sign-up form correctly, and select sign-up.	Full name = “raghad-al-hossny”. Email= “raghad@gmail.com” Password= “Deda@2002” Country-id= 1.	The account should be created successfully, and the front app will route to step 2	The account should be created successfully, and the front app will route to step 2	Pass
Tc-06	Check results on entering an existing email.	Full name = “raghad-al-hossny”. Email= “raghad@gmail.com” Password= “Deda@2002” Country-id= 1.	Error messages said, “The email already existed”.	Error messages said, “The email already existed”.	Pass
Tc-07	Check results when a user chooses a weak password (according to the system requirements).	Full name = “raghad-al-hossny”. Email= “raghad@gmail.com” Password= “2002” Country-id= 1.	Error message said “weak password!”	Error message said “weak password!”	Pass
Tc-08	Check results when any field is empty, and the “Sign-up” button is pressed.	Full name = “raghad-al-hossny”. Email= “ ” Password= “2002” Country-id= 1.	Error message “a field is missing”	Error message “a field is missing”.	Pass
Tc-09	Check results when entering the correct	OTP =22456	The email will be verified	The email will be	Pass

	OTP received via the email.		successfully, and the user will become Trusted, the web app will route to the third step.	verified successfully, and the user will become Trusted, the web app will route to the third step.	
Tc-10	Check results when entering a wrong OTP received via the email.	OTP =22456	“Wrong OTP!”.	“Wrong OTP!”	Pass
Tc-11	Check results when the user chooses the “resend OTP”.		The system will generate a new OTP and send it to the user.	The system will generate a new OTP and send it to the user.	Pass
Tc-12	Check results when the user completes the organization register form as expected and presses “register”.	Organization name = “Test-SPU” Organization description= “test” Logo = /image/dd.jpg	route the user to the main account page, and make the Org. manager account active.	route the user to the main account page, and make the Org. manager account active.	Pass
Tc-13	Check results when any field is empty,	Organization name = “Test-SPU”	“A field is missing”	“A field is missing”	Pass

	and the “Register” button is pressed.	Organization description= “test”			
Tc-14	Check the result when the user sign-up flow is interrupted and completed after a while.		Successfully route to the current step	Successfully route to the current step	Pass
Tc-15	Check results when the user chooses “show account information”.		The system will display account-related information.	The system will display account-related information.	Pass
Tc-16	Check results when the user chooses “update account information”	Full name = “raghad” County-id =4	The system will update the fields successfully	The system will update the fields successfully	Pass
Tc-17	Check results when a user chooses to “change password” with new passwords.	Old Password= ”Deda@2002” New-password =Dd@2002	The system will successfully update the passwords.	The system will successfully update the passwords.	Pass
Tc-18	Check results when a user chooses to “change password” with the wrong old password.	Old Password= ”Deda@” New-password =Dd@2002	“wrong old password”.	“wrong old password”.	Pass

Tc-19	Check results when a user chooses to “change password” with a weak new password.	Old Password=“Deda@” New-password=“Dd”	“password must follow the system definitions”.	“password must follow the system definitions”.	Pass
Tc-20	Check results when the user chooses to create a new organization member account with the correct email.	Email = <u>[raghad@gmail.com]</u>	The system will create the accounts successfully, generate a default password, and send it to the new member via email.	The system will create the accounts successfully, generate a default password, and send it to the new member via email.	Pass
Tc-21	Check results when the user chooses to create a new organization member account with an existing email.	Email = <u>[raghad@gmail.com]</u>	Error message this email exists in the organization.	Error message this email exists in the organization.	Pass
Tc-22	Check results when a new member login the system.	Email = <u>“raghad@gmail.com”</u> Default-password=“Dd@345”	The system routes the user to the complete profile page.	The system routes the user to the complete profile page.	Pass

Tc-23	Check results when a new member completes their profile with verified data.	Full name=raghad hossny, Country-id =5	The member's account becomes "active" successfully	The member's account becomes "active" successfully	Pass
Tc-24	Check results when the Org. manager chooses to deactivate an existing member.	Member = raghad	The member's account becomes "inactive" successfully	The member's account becomes "inactive" successfully	Pass
Tc-25	Check results when the Org. manager chooses to set member as admin.	Member =raghad	The member's account becomes admin successfully means having the full access privileges over all projects.	The member's account becomes admin successfully means having the full access privileges over all projects.	Pass
Tc-26	Check results when the system admin chooses to view reports		The system will display the usage reports successfully	The system will display the usage reports successfully	Pass

Tc-27	Check results when the organization manager creates a new project.	Name = AgileMeets, Desc = “Greate project”	The system will create a new project successfully.	The system will create a new project successfully.	Pass
Tc-28	Check results when the organization manager deactivates a project.	Project = AgileMeets	The system will deactivate the project successfully.	The system will deactivate the project successfully.	Pass
Tc-29	Check results when the organization manager views all projects related to an organization		The system will display all projects associated with the organization.	The system will display all projects associated with the organization.	Pass
Tc-30	Check results when the organization manager assigns a member to a project.	Member=raghad	The system will successfully assign the team members to the project, and give them default privileges.	The system will successfully assign the team members to the project, and give them default privileges.	Pass
Tc-31	Check results when the organization	Member=raghad	The system will successfully	The system will	Pass

	manager removes a member from a project.		remove the team member from the project.	successfully remove the team member from the project.	
Tc-32	Check results when the organization manager views all members assigned to a specific project with their related privileges.	Project= AgileMeets	The system will display all members associated with the project.	The system will display all members associated with the project.	Pass
Tc-33	Check results when the organization manager assigns privileges to a team member.	Meetings = write, Settings = read, Projects=read, Requirements= read, Members=write.	The system will update the privileges successfully.	The system will update the privileges successfully.	Pass
Tc-34	Check results when the organization manager removes privileges from a team member.	Meetings = write, Settings = read, Projects=read, Requirements= read, Members=read.	The system will update the privileges accordingly.	The system will update the privileges accordingly.	Pass
Tc-35	Check results of adding requirements from files of either “csv” or “excel” type.	File = :/file/req.csv	The requirements added successfully.	The requirements added successfully.	Pass

Tc-36	Check results of adding requirements from files of neither “csv” or “excel” type.	File = :/file/req.pdf	Error message “Wrong file format!”	Error message “Wrong file format!”	Pass
Tc-37	Check the results of adding requirements manually, with vailed data.	Title=’add new project’, Priority =’high’, Status= ‘in-progress’	Requirements added successfully.	Requirements added successfully.	Pass
Tc-38	Check the results of adding requirements manually, with missing data.	Title=’add new project’, Priority =’high’, Status=	Error message “Please complete all needed data”.	Error message “Please complete all needed data”.	Pass
Tc-39	Check results of adding requirements manually, with the wrong datatype.	Title=’add new project’, Priority =’high’, Status= 1	Error message “wrong datatype”.	Error message “wrong datatype”.	Pass
Tc-40	Check results for browsing all requirements for one project.		The system should display all requirement backlogs for this project.	The system should display all requirement backlogs for this project.	Pass

Tc-41	Check results for deleting a requirement.	Req-id=4	The requirement should be deleted successfully.	The requirement should be deleted successfully.	Pass
Tc-42	Check results for editing a requirement, with vailed data.	Req-id=4, Title='add new req'	The requirement should be edited successfully.	The requirement should be edited successfully.	Pass
Tc-43	Check results for editing a requirement, with invalid data.	Req-id=4, Title=22	Error message "wrong requirements data".	Error message "wrong requirements data".	Pass
Tc-44	Check the results of adding requirements from files in the defined structure.	File = :/file/req.csv	requirements added successfully.	requirements added successfully.	Pass
TC-45	Check the results of adding requirements from files that have a wrong structure.	File = :/file/req.csv	Wrong structure "Please make it as "req title", "req priority".	Wrong structure "Please make it as "req title", "req priority".	

Base Management -RTM Version 02:

Req_ID	Use cases	analysis	Detailed design	coding	Test cases
RE-FR-AM-01	Uc-01	BM-analysis	BM-design	source	Tc-05 Tc-06 Tc-07 Tc-08
RE-FR-AM-02	Uc-02	BM-analysis	BM-design	source	Tc-09 Tc-10
RE-FR-AM-03	Uc-02	BM-analysis	BM-design	source	Tc-11
RE-FR-OM-04	Uc-03	BM-analysis	BM-design	source	Tc-12 Tc-13
RE-FR-AM-05	Uc-01	BM-analysis	BM-design	source	Tc-14
RE-FR-AM-06	Uc-08	BM-analysis	BM-design	source	Tc-01 Tc-02 Tc-03
RE-FR-AM-07	Uc-04	BM-analysis	BM-design	source	Tc-15
RE-FR-AM-08	Uc-04	BM-analysis	BM-design	source	Tc-16
RE-FR-AM-09	Uc-08	BM-analysis	BM-design	source	Tc-04
RE-FR-AM-10	Uc-08	BM-analysis	BM-design	source	Tc-04
RE-FR-AM-11	Uc-01 Uc-04	BM-analysis	BM-design	source	Tc-17 Tc-18
RE-FR-AM-12	Uc-01 Uc-04	BM-analysis	BM-design	source	Tc-19
RE-FR-OM-13	Uc-05	BM-analysis	BM-design	source	Tc-20 Tc-21
RE-FR-AM-14	Uc-05	BM-analysis	BM-design	source	Tc-22 Tc-23
RE-FR-AM-15	Uc-05	BM-analysis	BM-design	source	Tc-23
RE-FR-OM-16	Uc-05	BM-analysis	BM-design	source	tc-24

RE-FR-OM-17	Uc-16	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-32
RE-FR-OM-18	Uc-06	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-25
RE-FR-OM-19	Uc-16	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-33
RE-FR-OM-20	Uc-16	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-31
RE-FR-PM-21	Uc-13	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-27
RE-FR-PM-22	Uc-14	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-28
RE-FR-PM-23	Uc-09	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-29
RE-FR-PM-24	Uc-15	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-30
RE-FR-PM-25	Uc-16	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-31
RE-FR-PM-26	Uc-09	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-32
RE-FR-RM-27	Uc-11	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-37 Tc-38 Tc-39
RE-FR-RM-28	Uc-11	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-35 Tc-36 Tc-44 Tc-45
RE-FR-RM-29	Uc-10	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-40
RE-FR-RM-30	Uc-12	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-42 Tc-43
RE-FR-RM-31	Uc-10	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-41
RE-FR-OM-32	Uc-18	<u>BM-analysis</u>	<u>BM-design</u>	<u>source</u>	Tc-26

Chapter4 Notification Service

1. Introduction

This chapter will specify and detail the **Notification Service** from different perspectives starting with functional reaching the actual implementation and integration into the whole system.

2. Service Analysis

2.1 Detailed Requirements

The requirements behind this service are all about sending notifications where appropriate and ensure resiliency.

Table 12 Notification Service Detailed Requirements

Req_ID	Requirement Title	category	description
RE-FR-N-01	The system must send an email containing a 5-digit verification code to newly registered users as part of the registration flow.	Notifications	-
RE-FR-N-02	The system must allow users to request a new verification code, sending it via email when requested.	Notifications	-
RE-FR-N-03	The system must send a verification code via email when users request a password reset.	Notifications	-
RE-FR-N-04	The system must notify users through both email and push notifications when their account is deactivated.	Notifications	-

RE-FR-N-05	The system must send a welcome email to new organization members containing their initial credentials and organization details.	Notifications	-
RE-FR-N-06	The system must notify all organization members through both email and push notifications when their organization is deactivated.	Notifications	-
RE-FR-N-07	The system must send both email and push notifications to users when their administrative privileges are granted or revoked.	Notifications	-
RE-FR-N-08	The system must send both email and push notifications to users when they are invited to a meeting, including meeting title, date, time, goal, and location details.	Notifications	-
RE-FR-N-09	The system must notify all meeting members through email when meeting details are updated, including changes to time, location, or other meeting parameters.	Notifications	-
RE-FR-N-10	The system must send both email and push notifications to all meeting members when a meeting is cancelled, including the meeting title and original scheduled time.	Notifications	-
RE-FR-N-11	The system must notify all meeting members via email when a meeting starts, including meeting title, time, and location.	Notifications	-
RE-FR-N-12	The system must send email notifications to all meeting members when a meeting is completed, including meeting duration details.	Notifications	-

RE-FR-N-13	The system must notify meeting creators via email when members confirm or decline meeting attendance.	Notifications	-
RE-FR-N-14	The system must send reminder notifications to meeting members before scheduled meetings.	Notifications	-
RE-FR-N-15	The system must notify assigned project managers through both email and push notifications when they are assigned to a newly created project.	Notifications	-
RE-FR-N-16	The system must send both email and push notifications to users when they are assigned to a project.	Notifications	-
RE-FR-N-17	The system must notify users through push notifications when their project access privileges are modified.	Notifications	-
RE-FR-N-18	The system must support sending push notifications to multiple devices for the same user through Firebase Cloud Messaging tokens.	Notifications	-
RE-FR-N-19	The system must provide the ability to users to subscribe and unsubscribe their devices for push notifications, managing Firebase tokens securely.	Notifications	-
RE-FR-N-20	The system must implement retry mechanisms and error handling for failed notification deliveries.	Notifications	<ul style="list-style-type: none"> - Authentication failures (no retry) - SMTP rate limiting (5-minute delay retry) - Firebase messaging errors (no retry) - General errors (immediate retry)

2.2 Class Diagram

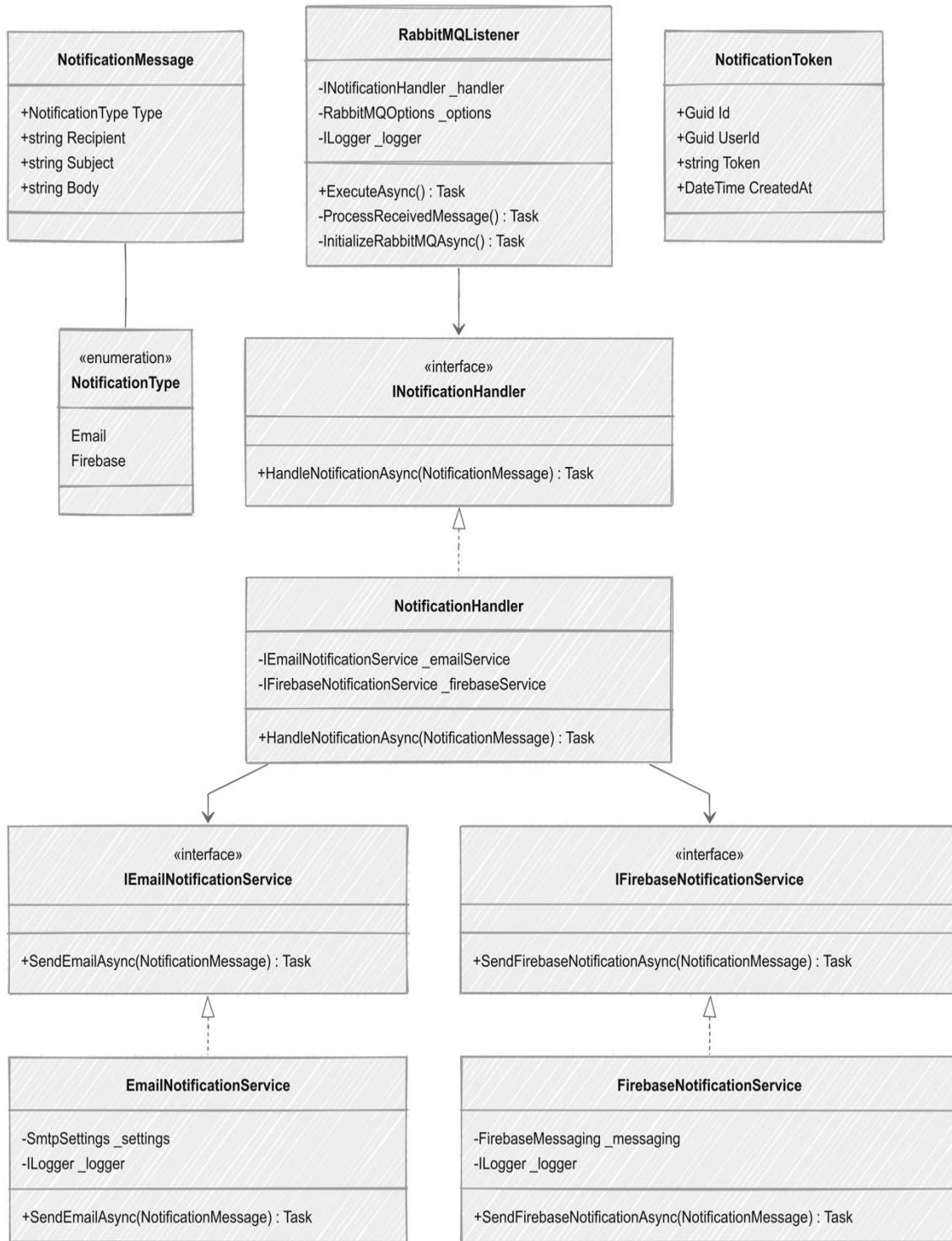


Figure 42 Notifications Class Diagram

2.3 Sequence Diagram (Comprehensive one)

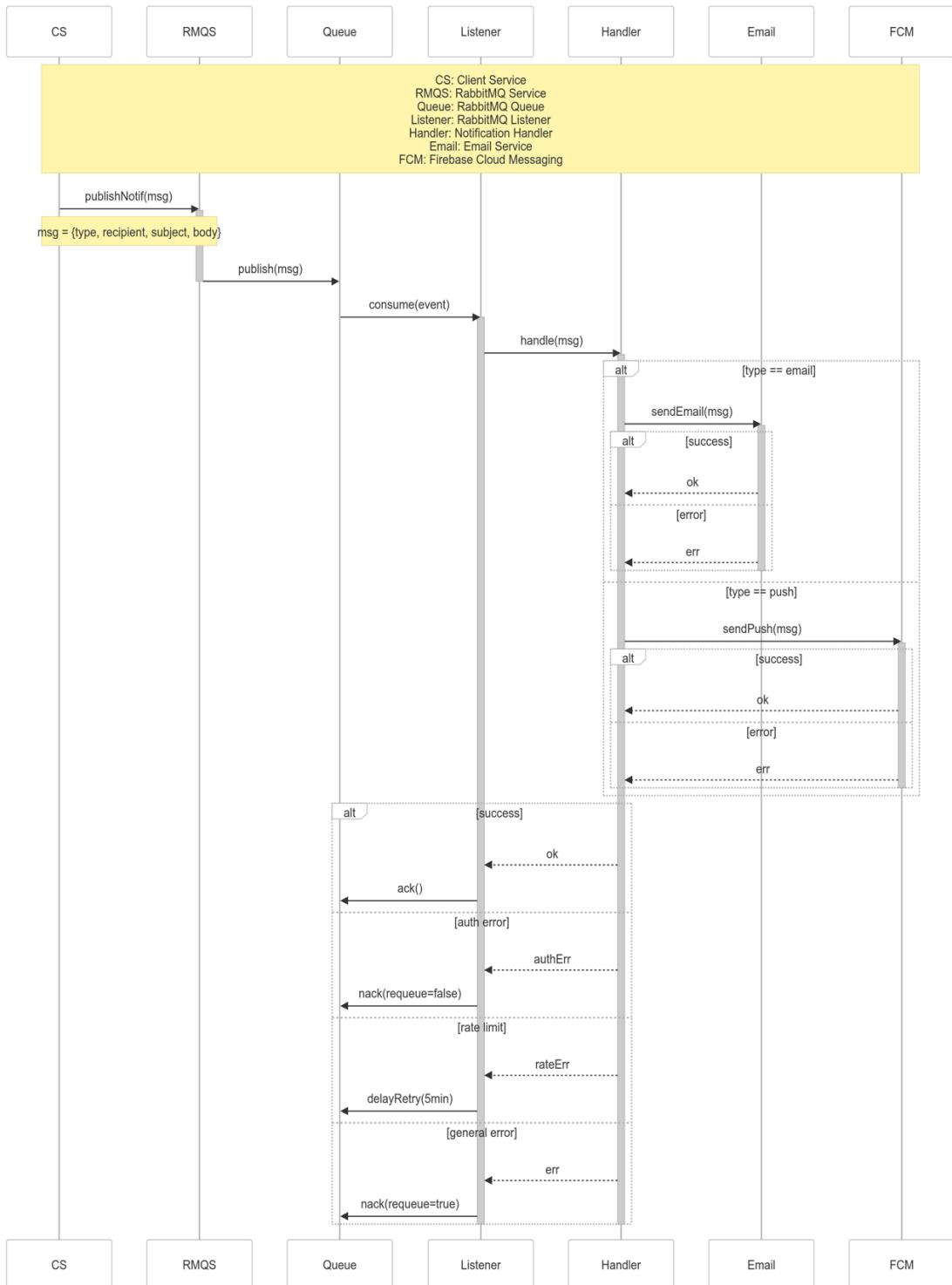


Figure 43 Notification Sequence Diagram

2.4 Test cases definition

Table 13 Notifications - test cases definition

Test Case Scenario:		Sce-01: Email Notification Delivery	
Test case id	Test case title	Test steps	Expected result
Tc-01	Check successful email notification delivery	<ol style="list-style-type: none"> 1. Create a notification message with type Email 2. Set valid recipient email, subject, and body 3. Publish notification through RabbitMQ 4. Wait for processing 	Email should be delivered successfully to recipient with correct subject and body
Tc-02	Check email notification with invalid recipient	<ol style="list-style-type: none"> 1. Create a notification message with type Email 2. Set invalid email format as recipient 3. Publish notification through RabbitMQ 	Error message indicating invalid email format, message should not be requeued
Tc-03	Check email notification during SMTP server downtime	<ol style="list-style-type: none"> 1. Create a notification message with type Email 2. Set valid recipient email 3. Simulate SMTP server unavailability 4. Publish notification 	Message should be requeued for retry with appropriate delay

Test case scenario:		Sce-02: Firebase Push Notification Delivery	
Test case id	Test case title	Test steps	Expected result
Tc-04	Check successful push notification delivery	<ol style="list-style-type: none"> 1. Create a notification message with type Firebase 2. Set valid FCM token as recipient 3. Publish notification through RabbitMQ 	Push notification should be delivered to user's device
Tc-05	Check push notification with expired token	<ol style="list-style-type: none"> 1. Create a notification message with type Firebase 2. Set expired/invalid FCM token 3. Publish notification 	Error logged, token should be marked for removal, message not requeued
Tc-06	Check push notification during FCM service disruption	<ol style="list-style-type: none"> 1. Create a notification message with type Firebase 2. Set valid FCM token 3. Simulate FCM service unavailability 4. Publish notification 	Message should be requeued for retry

3. Design & Implementation Considerations

3.1 Overall Architecture

The Notification Service is designed as a dedicated microservice within our system architecture, responsible for handling all types of notifications across the platform. It follows an event-driven architecture pattern, utilizing message queues for reliable and scalable notification delivery. The service operates heedlessly (without HTTP endpoints), processing messages from a message queue and directing them to appropriate notification channels.

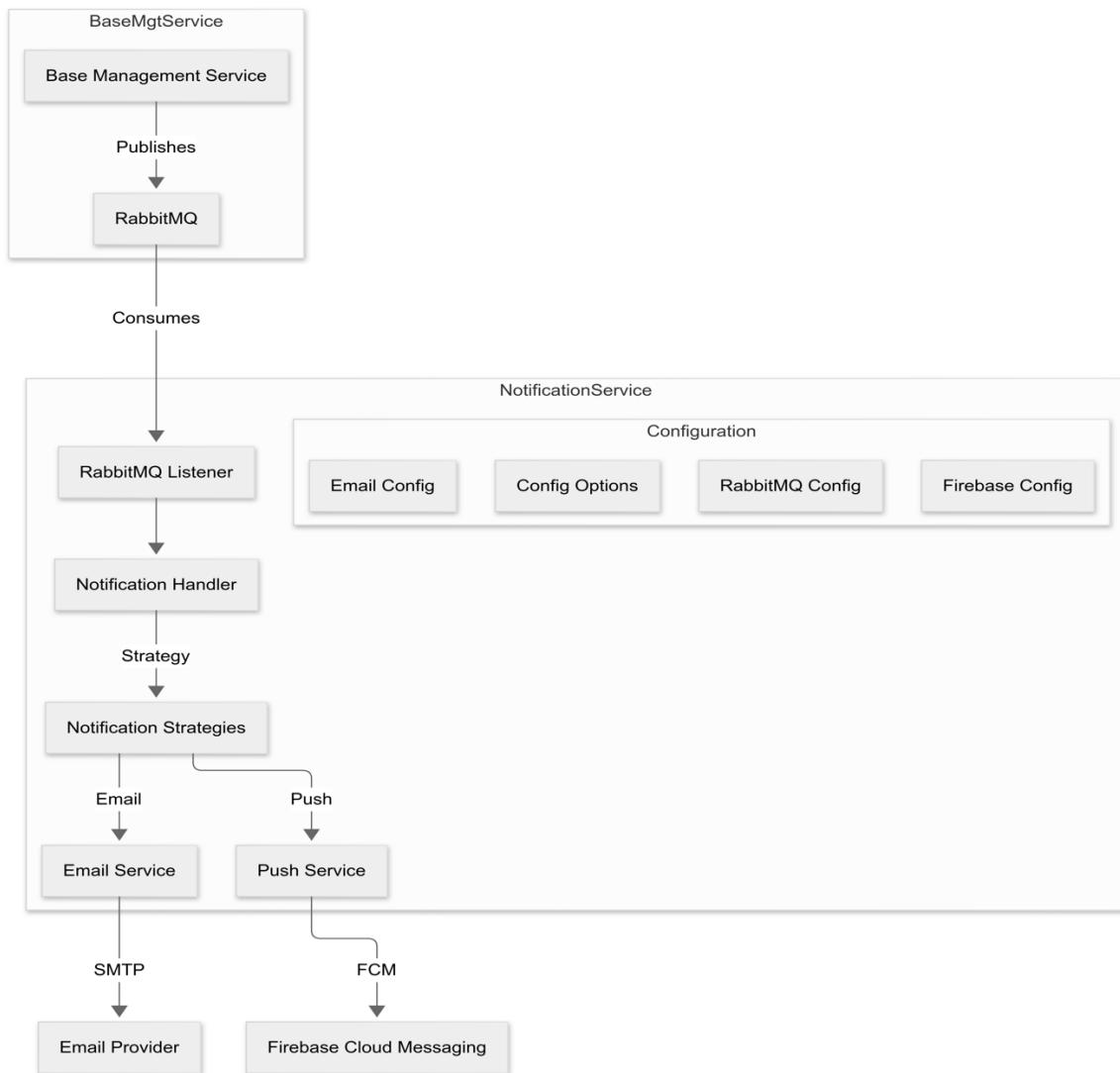


Figure 44 Notification Service Architecture

3.2 Architectural Components and Their Interactions

1- Message Queue Integration

The service utilizes RabbitMQ as its message broker, implementing a robust pub/sub pattern. This choice enables:

- Asynchronous message processing
- Reliable message delivery
- System decoupling
- Scalable message handling

2- Message Consumer Layer

The RabbitMQ Listener serves as the entry point for all notification requests. It:

- Maintains a persistent connection to RabbitMQ
- Implements connection recovery mechanisms
- Handles message acknowledgments
- Routes messages to appropriate handlers

3- Notification Handler

Acts as the orchestrator for notification processing:

- Validates incoming messages
- Determines the notification strategy
- Manages delivery attempts
- Handles errors and retries

4- Notification Strategies

Implements the Strategy pattern for different notification types:

- Email notifications via SMTP
- Push notifications via Firebase Cloud Messaging
- Extensible design for future notification channels

3.3 External Integrations

1- Email Provider

- Handles email delivery through SMTP
- Manages email templates
- Implements rate limiting
- Handles delivery status tracking

2- Firebase Cloud Messaging

- Manages device tokens
- Handles push notification delivery
- Implements FCM-specific error handling
- Manages notification priorities

3.4 Key Design Considerations

1- Reliability

- Message persistence in RabbitMQ
- Automatic retry mechanisms
- Dead letter queues for failed messages
- Circuit breaker patterns for external services

2- Scalability

- Horizontal scaling capability
- Message queue partitioning
- Stateless service design
- Configurable concurrency limits

3- Maintainability

- Clear separation of concerns
- Modular component design

4. Future Considerations

1. **Additional Channels:** Support for SMS, Slack, or other platforms
2. **Advanced Analytics:** Notification delivery tracking and analytics
3. **Template Management:** Dynamic template system for notifications
4. **Smart Routing:** Intelligent notification routing based on user preferences
5. **Enhanced Monitoring:** Advanced metrics and alerting system.

Chapter5 Meeting Service

1. Introduction

In this chapter, we present and distinguish theoretical and practical aspects of meetings—particularly online ones—along with online communication technologies/protocols, service providers, and available options. We also discuss their suitability for AgileMeets' requirements and preferences.

2. Required theory and basic knowledge

2.1 WebRTC

Definition: Web Real-Time Communication (WebRTC) is an open-source technology enabling peer-to-peer audio, video, and data exchange in web browsers without plugins.

Key Components:

We will not delve deeply into the internal components or technical details of these components. Instead, we focus solely on their key functionalities.

- Signaling: Exchanging session metadata (e.g., IP addresses) using protocols like WebSockets to establish connections.
- Media Capture: Uses **getUserMedia** API to access camera and microphone.
- Peer Connection: Managed by **RTCPeerConnection** for direct media streams between users.
- **NAT** Traversal: Uses **STUN** (Session Traversal Utilities for **NAT**) and **TURN** (Traversal Using Relays around **NAT**) to navigate firewalls and NATs.
- **Codecs and Protocols:** Supports VP8, VP9 (video), Opus (audio), and **UDP** for low-latency transmission.

Overall Architecture:

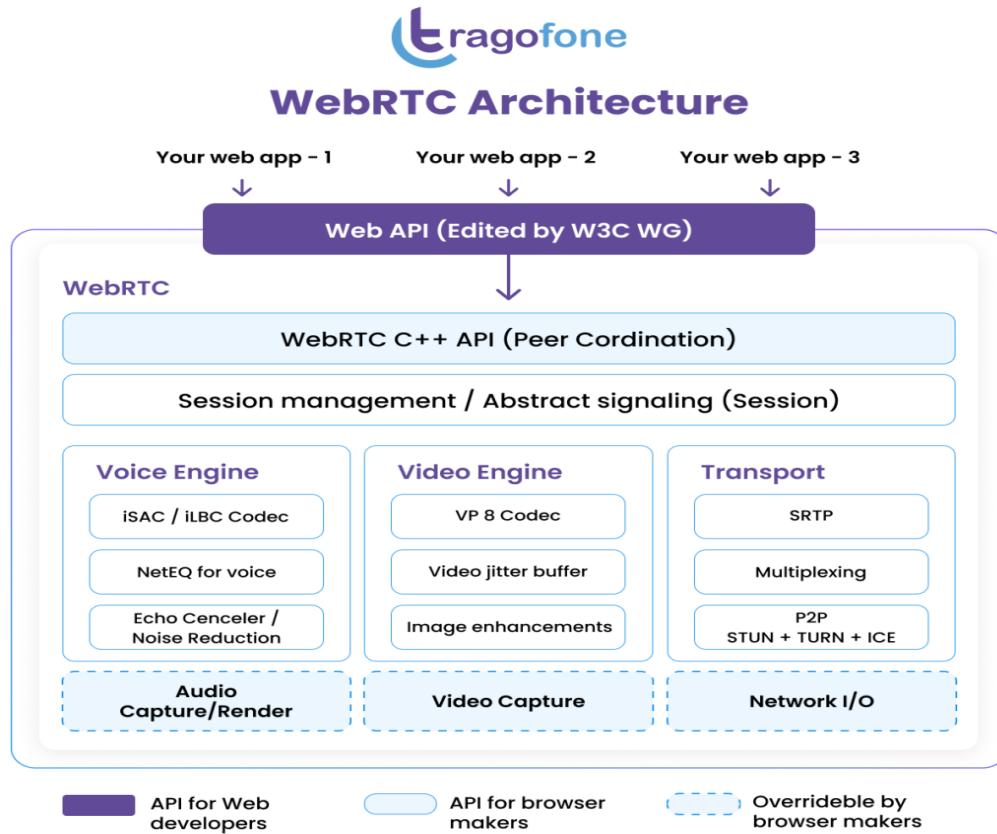


Figure 45 WebRTC Architecture

2.2 Network Protocols for Real-Time Communication

- **SIP (Session Initiation Protocol):** A signaling protocol for initiating, maintaining, and terminating real-time sessions (voice, video, messaging).
- **RTP/RTCP (Real-Time Transport Protocol/Control Protocol):** For delivering audio/video over IP networks and monitoring QoS.
- **WebSocket & HTTP/2:** Protocols enabling full-duplex communication for signaling and low-latency data exchange.
- **P2P vs. Client-Server Architectures:** Trade-offs in scalability, latency, and infrastructure complexity.

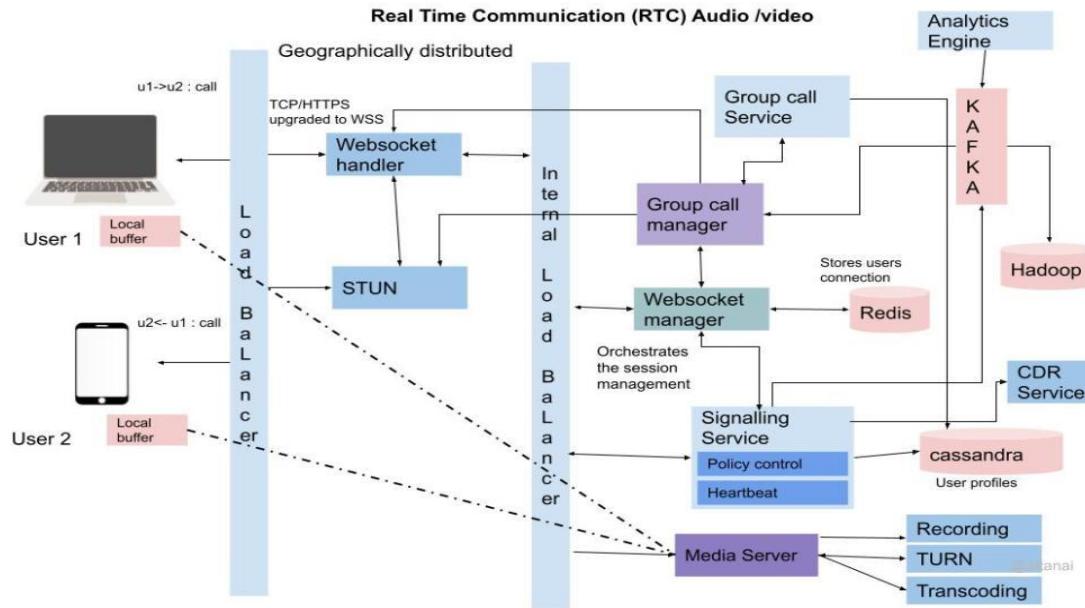


Figure 46 RTC Implementation Example

2.3 Media Codecs & Compression

- **Video Codecs:** H.264 (compatibility), VP8/VP9 (open-source), AV1 (next-gen efficiency).
- **Audio Codecs:** Opus (low latency), AAC (high quality), G.711 (legacy VoIP).
- **Bandwidth-Quality Trade-offs:** Adaptive bitrate streaming and resolution scaling for varying network conditions.

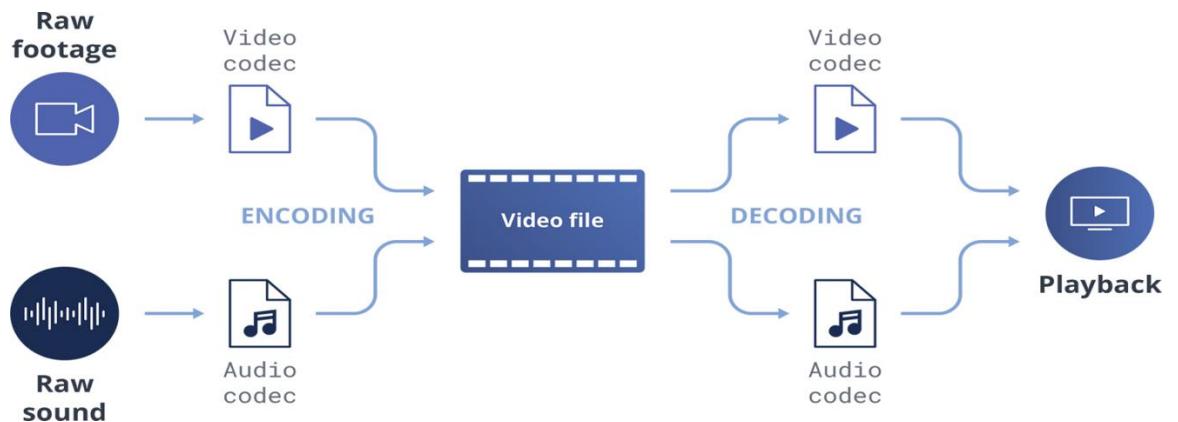


Figure 47 Media Codec & Compression

Codec	Pros	Cons
H.264 (AVC)	Widely supported	Older codec
	Good compression	Less efficient than newer codecs
	Suitable for streaming and video conferencing	Limited support for 4K and beyond
H.265 (HEVC)	Improved compression efficiency	Requires more processing power
	High-quality video	Patent and licensing issues
	Suitable for 4K and HDR content	Slower encoding
VP9	Royalty-free	Not as widely supported
	Efficient compression	Encoding can be slow
	Good for web video streaming	Hardware support may vary
AV1	Open and royalty-free	Slower encoding and decoding
	Excellent compression efficiency	Hardware support may vary
	Suitable for 4K and HDR content	Patent and licensing concerns
	Adaptive streaming support	

Figure 48 Famous Codecs Comparison

2.4 Why it matters?

In designing AgileMeets or similar online communication apps, we carefully need to balance quality, performance, and cost to meet the demands of real-time collaboration. Low latency should be prioritized to ensure seamless communication during time-sensitive Agile ceremonies like stand-ups, achieved through efficient codecs like VP8/H.264 for video and Opus for audio. Cross-platform reliability was another key consideration, guiding us toward widely supported codecs that avoid hardware limitations, particularly on mobile devices. Finally, cost predictability was ensured by leveraging scalable architectures like SFU-first WebRTC, which avoids the overhead of traditional MCU-style mixing. These decisions collectively enable a robust, accessible, and cost-effective meeting experience tailored to Agile workflows.

3. Available Solutions

In this section, we evaluate various solutions for implementing real-time online meetings, including commercial providers like **Zoom**, other cloud-based WebRTC API services, and open-source platforms. Each solution is assessed based on its features, cost, pros, and cons, with a focus on how well it aligns with our requirements for flexibility, privacy, and ease of integration.

3.1 Zoom

Zoom is a widely recognized video conferencing platform known for its reliability and scalability. It provides APIs and SDKs that allow developers to integrate video, audio, and screen-sharing features into custom applications. Despite its popularity, Zoom uses proprietary technology rather than WebRTC, which may limit integration flexibility with standard web technologies.

Cost:

Zoom's pricing starts at \$14.99 per host per month for the Pro plan, with additional costs for enterprise features or large-scale usage. API usage may incur further charges depending on the volume, but as we can see this is not like the pricing model for other cloud services and it allows limited concurrent meetings.

3.2 Other Cloud Providers (e.g., Twilio, ZEGOCLOUD)

Cloud providers like Twilio and ZEGOCLOUD offer WebRTC-based APIs for real-time communication, allowing developers to build custom solutions without managing infrastructure.

Cost:

- Twilio: \$0.004 per participant per minute for video calls.
- ZEGOCLOUD: Pay-as-you-go with 10,000 free minutes, then usage-based pricing (almost like Twilio).

Pros:

- Scalable and easy to integrate via APIs.
- No need to manage infrastructure, reducing operational overhead.
- Flexible pricing based on usage.

Cons:

- Costs can become unpredictable and high with large-scale usage.
- Limited control over infrastructure and data privacy.
- Requires ongoing dependency on third-party services.

Product breakdown	
Video calling Rate for each participant connected to a Video Group Room	\$0.004 per participant per minute
Participant recordings Audio and Video track recordings of each participant connected to a Video Group Room	\$0.004 per participant per minute
Video call compositions Rate for composing/mixing the participant recording files into a single playable MP4 file	\$0.01 per composed minute
Media storage Storage of recording and composition files on the Twilio Cloud	\$0.00167 per GB per day* * First 10 GBs is free

Figure 49 Twilio Video Pricing Model

3.3 Open-Source Solutions (e.g., OpenVidu, LiveKit)

Open-source platforms like OpenVidu and LiveKit provide self-hosted, customizable solutions for real-time communication using WebRTC. These platforms offer full control over deployment and data management.

Cost:

Free to use, with costs limited to infrastructure (e.g., cloud hosting). For example, deploying on AWS EC2 can cost **\$50–\$500** per month depending on scale.

Pros:

- No licensing fees, only infrastructure costs.
- Full customization and control over data privacy.
- Flexibility to integrate with other tools and services.
- Ideal for organizations prioritizing sovereignty and compliance.

Cons:

- Requires technical expertise for deployment and maintenance.
- Scaling and infrastructure management can be resource intensive.

3.4 OpenVidu vs LiveKit Open Source

LiveKit Open Source is probably the most advanced and feature-rich open source WebRTC stack available today. It has a simple but very versatile API design and has a large collection of SDKs to integrate into your application on both the frontend and backend. Regardless of your technology stack, there is sure to be a LiveKit Open Source SDK available for you! This is why **OpenVidu is fully compatible with LiveKit protocols**. You can use any LiveKit SDK to build your application, and it will work seamlessly with an OpenVidu deployment.

3.4.1 What does OpenVidu bring over LiveKit Open Source?

- **Egress and Ingress services already integrated with a Redis instance:** LiveKit allows you to export media from a Room (for example recording it) or import media into a Room (for example ingesting a video file), using Egress and Ingress services respectively. These modules are independent of LiveKit Server and must be correctly configured and connected via a shared Redis. When running OpenVidu you will have all these services properly integrated, so you can focus on developing your app without worrying about anything else.
- **S3 compatible storage for Egress recordings:** OpenVidu comes with an S3 compatible storage already configured to store Egress recordings (Minio).
- **Administration dashboard to monitor your Rooms:** OpenVidu comes with an administration dashboard that allows you to monitor the status of your Rooms. Not only in real time, but also historically: the number of participants, the number of published tracks, Egress and Ingress processes... This is a great tool to have when developing your app, as it can help spotting issues and debugging your application's logic.
- **Integrated Observability Modules.**

Table 14 Different Online Meeting Solutions Comparison

Aspect	OpenVidu	LiveKit	Twilio	Zoom
Technology Used	WebRTC	WebRTC	WebRTC	Proprietary (not WebRTC)
Features	Recording, screen sharing, video filters, IP camera support	Scalable SFU, simulcast, selective subscription	Customizable, build-your-own solution	Recording, screen sharing, large meetings.

Scalability	Scalable with multiple media nodes	Highly scalable, designed for performance	Very scalable, pay-per-use	Highly scalable
Cost	Free, only infrastructure costs (e.g., AWS EC2)	Free, only infrastructure costs	\$0.004/participant/minute, variable cost	Not applicable
Customization	Highly customizable, open source	Highly customizable, open source	Extremely customizable	Limited to API capabilities

For our project, **OpenVidu** stands out as the best choice due to its alignment with our likely needs for online meetings with recording, AI analysis, and STT. Here's why:

- **Comprehensive Features:** It offers recording, screen sharing, and **S3** integration for AI processing out-of-the-box, reducing development time.
- **Ease of Use:** Deployment is streamlined with **AWS CloudFormation**, making it manageable even with limited DevOps expertise.
- **Privacy:** Self-hosting gives us full control over data, crucial for sensitive meeting content.
- **Customization:** Open-source nature allows tailoring to integrate with agile tools or other systems.
- **Cost-Effectiveness:** No licensing fees—just infrastructure costs—make it budget-friendly for long-term use.

3.5 OpenVidu Single Node

Our decision was to adopt OpenVidu, specifically the single-node solution, as a starting point. This approach offers an easy path to scaling up, enabling future transitions to high-availability configurations or higher-performance setups.

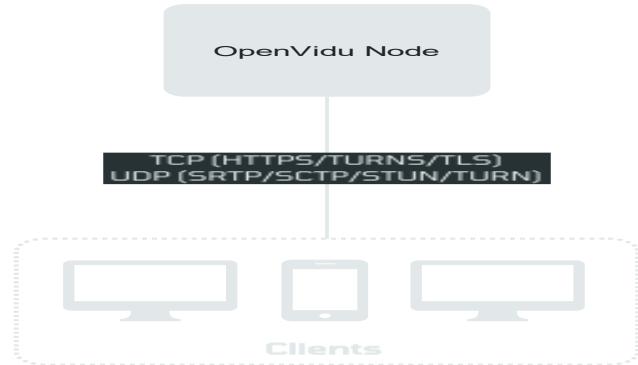


Figure 50 OpenVidu Single Node

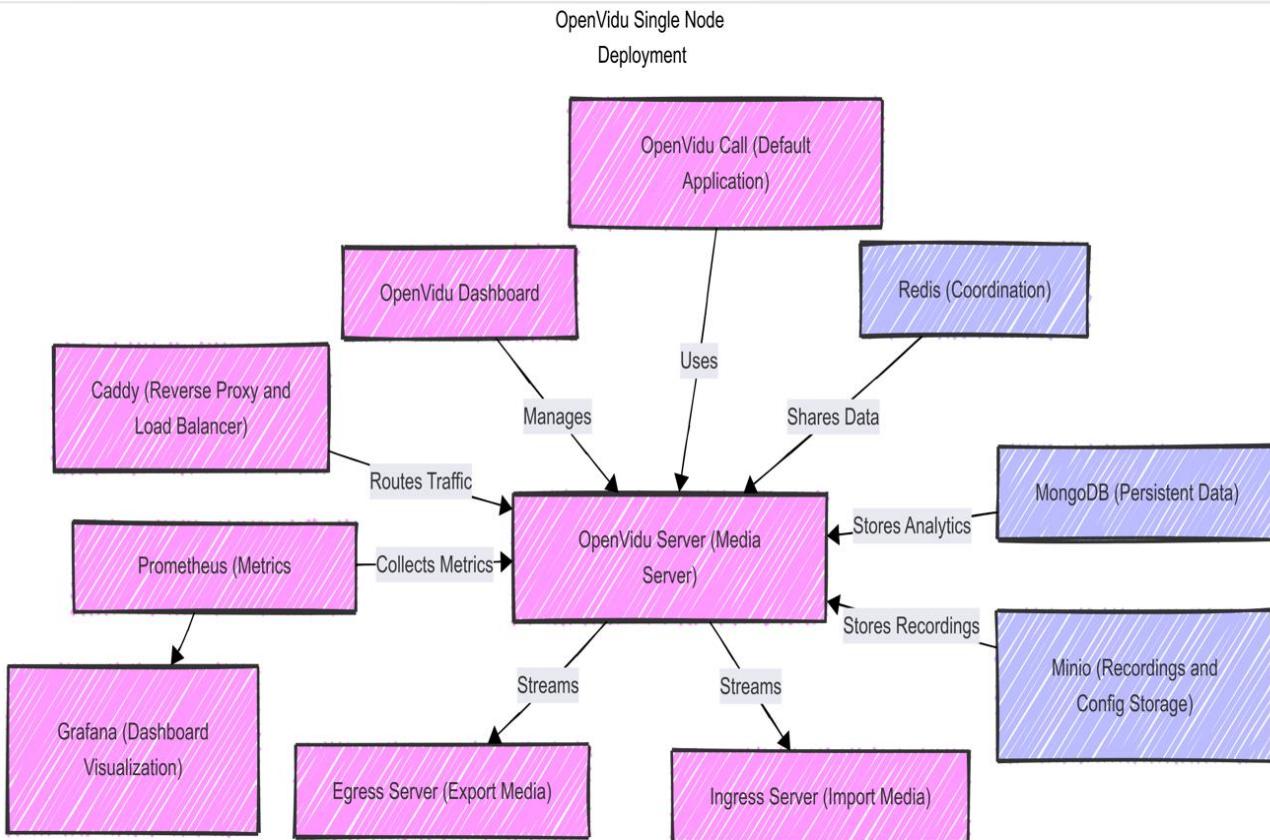


Figure 51 OpenVidu internal architecture & Components

4. Actual Microservice Analysis

While **OpenVidu** was identified as the chosen tool early in the process (for the sake of chapter organization), this selection was ultimately driven by the requirements analysis of **AgileMeets'** meeting service needs.

4.1 Detailed Requirements:

Table 15 Meeting Detailed Requirements

Req_ID	Requirement Title	category	description
RE-FR-MM-01	The system must support three types of meetings: in-person (with mandatory location), online (with LiveKit integration), and done (with audio upload).	Meeting Management	-

RE-FR-MM-02	The system must allow organization members to create meetings with title, goal, start time, and optional location (mandatory for in-person) The creator becomes the meeting organizer.	Meeting Management	-
RE-FR-MM-03	The system must validate meeting time conflicts before allowing creation or updates.	Meeting Management	-
RE-FR-MM-04	The system must allow adding and removing members to/from meetings with automatic email notifications.	Meeting Management	-
RE-FR-MM-05	The system must allow meeting members to confirm or decline their attendance.	Meeting Management	-
RE-FR-MM-06	The system must send reminder notifications to all meeting members before the meeting starts based on the configured reminder time.	Meeting Management	-
RE-FR-MM-07	The system must generate unique room names and access tokens for online meetings.	Meeting Management	-
RE-FR-MM-08	The system must automatically start and stop audio recording for online meetings.	Meeting Management	-
RE-FR-MM-09	The system must track online meeting status (Not Started, Active, Ended) and participant activity.	Meeting Management	-
RE-FR-MM-10	The system must support daily, weekly, and monthly recurring meeting patterns.	Meeting Management	-
RE-FR-MM-11	The system must maintain up to 5 future instances of recurring meetings.	Meetings Management	-
RE-FR-MM-12	The system must send email notifications to all meeting members when a meeting is completed, including meeting duration details.	Meeting Management	-

RE-FR-MM-13	The system must allow modifications to either single instances or the entire series of recurring meetings.	Meeting Management	-
RE-FR-MM-14	The system must handle exceptions in recurring patterns when individual instances are modified.	Meeting Management	-
RE-FR-MM-15	The system must automatically transition meeting status from Scheduled to InProgress when started.	Meetings Management	-
RE-FR-MM-16	The system must automatically complete meetings when they exceed their end time.	Meeting Management	-
RE-FR-MM-17	The system must automatically cancel scheduled meetings that never started.	Meeting Management	-
RE-FR-MM-18	The system must allow manual cancellation of meetings.	Meeting Management	Only Organizer or Admin users can.
RE-FR-MM-19	The system must support audio file uploads for in-person meetings.	Meeting Management	-
RE-FR-MM-20	The system must automatically record and store audio for online meetings.	Meeting Management	-
RE-FR-MM-21	The system must track audio status (Pending, Available, Failed) and source (Upload, Meeting Service).	Meeting Management	-
RE-FR-MM-22	The system must support user's different time zones and ensure time zone conversion with meetings scheduling and timing.	Meeting Management	-
RE-FR-MM-23	The system must allow the organizer to download the recording file of a done meeting.	Meeting Management	-

RE-FR-MM-24	The organizer should be able to upload in-person recordings later, even after the completion of the meeting session.	Meeting Management	The system should support caching pending recordings.
RE-FR-MM-25	The organizer should be able to leave the application while recording, recording should still be on, and system will inform the organizer with recording status.	Meeting Management	Background Recording support.
RE-FR-MM-26	The system should support background & chunk uploading, allowing the users to easily upload recordings without having an active app on foreground all the time.	Meeting management.	Background Upload support.
RE-FR-C-01	The system must support two types of calendar feed subscriptions: personal and project based.	Calendar	-
RE-FR-C-02	The system must generate iCalendar format feeds with meeting details including title, goal, location, and participant information.	Calendar	-
RE-NFR-M-01	The Online Meeting Service shall maintain an uptime of at least 99.9% over a 30-day period , excluding scheduled maintenance.	Meeting Availability	Users expect the meeting service to be consistently available for scheduling and joining meetings.
RE-NFR-M-02	For meetings up to 1 hour in length , the system shall complete audio recording, storage, and transcription within 10 minutes of the meeting's end for 95% of meetings .	Processing Latency	Users rely on quick access to transcriptions post-meeting for follow-ups or documentation.

4.2 High-level Use case Diagram

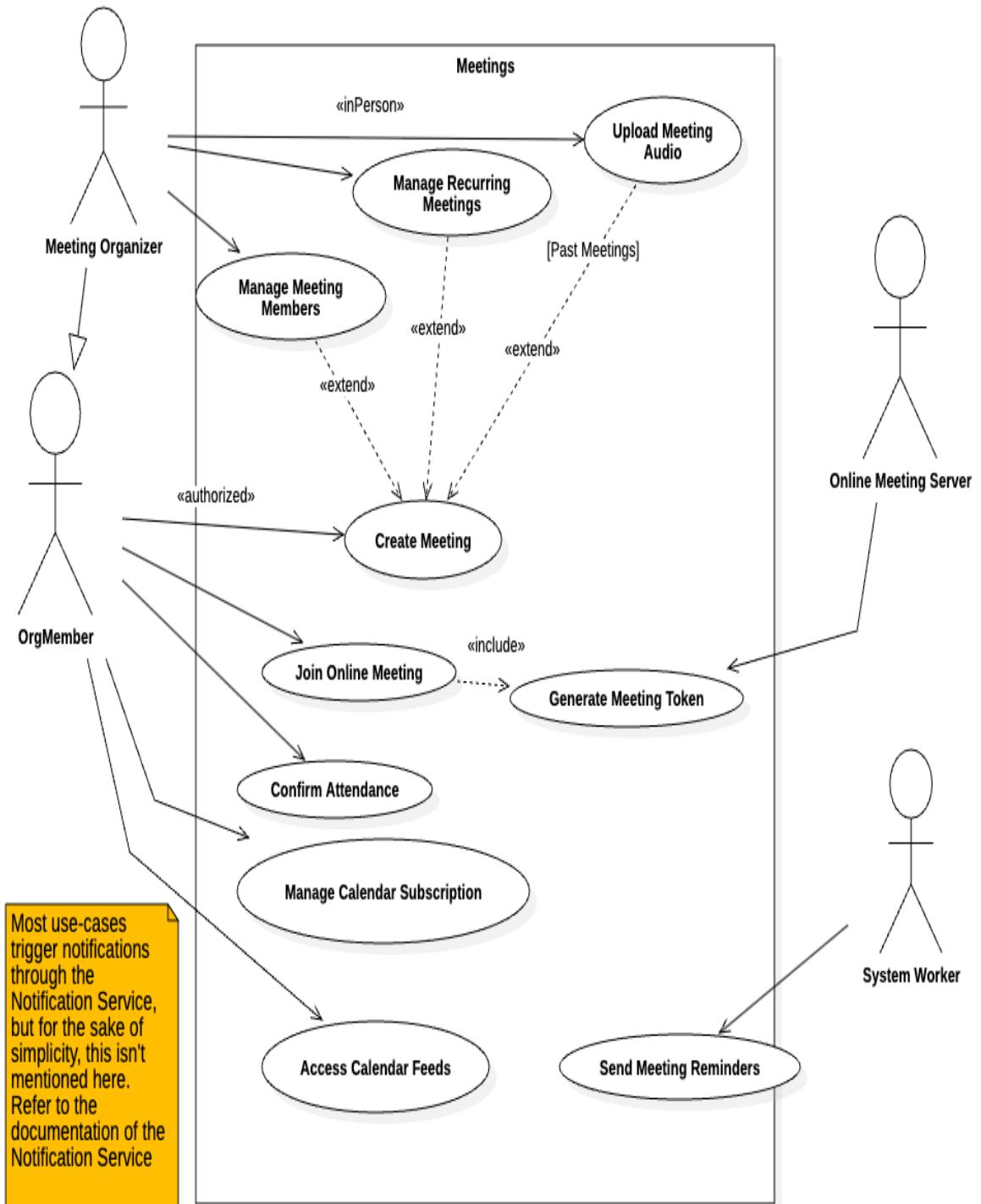


Figure 52 Meetings Use case Diagram

4.3 Testcases

Table 16 Test Scenario Meeting Creation

Test Case Scenario:		Sce-01: Meeting Creation and Basic Management	
Test case id	Test case title	Test steps	Expected result
TC-MM-01	Check successful creation of in-person meeting	<ol style="list-style-type: none"> 1. Create meeting with required fields 2. Set meeting type as in-person 3. Provide mandatory location 4. Set valid start/end time 	Meeting should be created successfully with status Scheduled
TC-MM-02	Check successful creation of online meeting	<ol style="list-style-type: none"> 1. Create meeting with required fields 2. Set meeting type as online 3. Set valid start/end time 	Meeting should be created with LiveKit room and token generated
TC-MM-03	Check meeting time conflict validation	<ol style="list-style-type: none"> 1. Create meeting with time conflicting with existing meeting 2. Submit creation request 	Error message "Time conflicts with existing meeting"
TC-MM-04	Check recurring meeting pattern generation	<ol style="list-style-type: none"> 1. Create meeting with recurring pattern 2. Set weekly recurrence 3. Set valid end date 	5 future instances should be generated automatically

Table 17 Test Scenario Meeting Attendance

Test Case Scenario:		Sce-02: Meeting Attendance and Member Management	
Test case id	Test case title	Test steps	Expected result
TC-MA-01	Check member addition with notification	<ol style="list-style-type: none"> 1. Add new member to existing meeting 2. Submit member addition 	Member should be added and receive email notification
TC-MA-02	Check attendance confirmation	<ol style="list-style-type: none"> 1. Access meeting as member 2. Confirm attendance 	Attendance should be recorded, organizer notified
TC-MA-03	Check reminder delivery	<ol style="list-style-type: none"> 1. Have scheduled meeting 2. Wait until reminder time 3. Check notification delivery 	Members should receive reminder notification before meeting

Table 18 Test Scenario Online Meeting

Test Case Scenario:		Sce-03: Online Meeting Operations	
Test case id	Test case title	Test steps	Expected result
TC-OM-01	Check online meeting start process	<ol style="list-style-type: none"> 1. Start scheduled online meeting 2. Generate access tokens 3. Join the session 	Meeting status should change to InProgress, recording should start
TC-OM-02	Check meeting auto-completion	<ol style="list-style-type: none"> 1. Have active online meeting 2. All participants leave 3. Room becomes empty 	Meeting should be marked as Completed; recording should stop
TC-OM-03	Check recording availability	<ol style="list-style-type: none"> 1. Complete online meeting 2. Wait for processing 3. Check audio status 	Audio should be available and marked as Available

Table 19 Test Scenario Meeting Status Transitions

Test Case Scenario:		Sce-04: Meeting Status Transitions	
Test case id	Test case title	Test steps	Expected result
TC-MS-01	Check auto-completion of expired meeting	<ol style="list-style-type: none"> 1. Have meeting past end time 2. Wait for status worker cycle 	Meeting should be marked as Completed automatically
TC-MS-02	Check cancellation by organizer	<ol style="list-style-type: none"> 1. Access scheduled meeting as organizer 2. Cancel meeting 3. Check member notifications 	Meeting should be marked as Cancelled; members notified
TC-MS-03	Check auto-cancellation of never-started meeting	<ol style="list-style-type: none"> 1. Have scheduled meeting 2. Let start time pass without starting 3. Wait for status worker cycle 	Meeting should be automatically marked as Cancelled

5. Design & Integration Considerations

In this section, we will focus on the following:

- **Overall Architecture:** showing how the meeting service & online meetings provider (OpenVidu) are placed in the system.
- **Meeting Service Internal Structure.**
- **How BaseMgt integrates Meeting Service.**
- **Resiliency**
- **Notable Design Patterns**

5.1 Overall Architecture

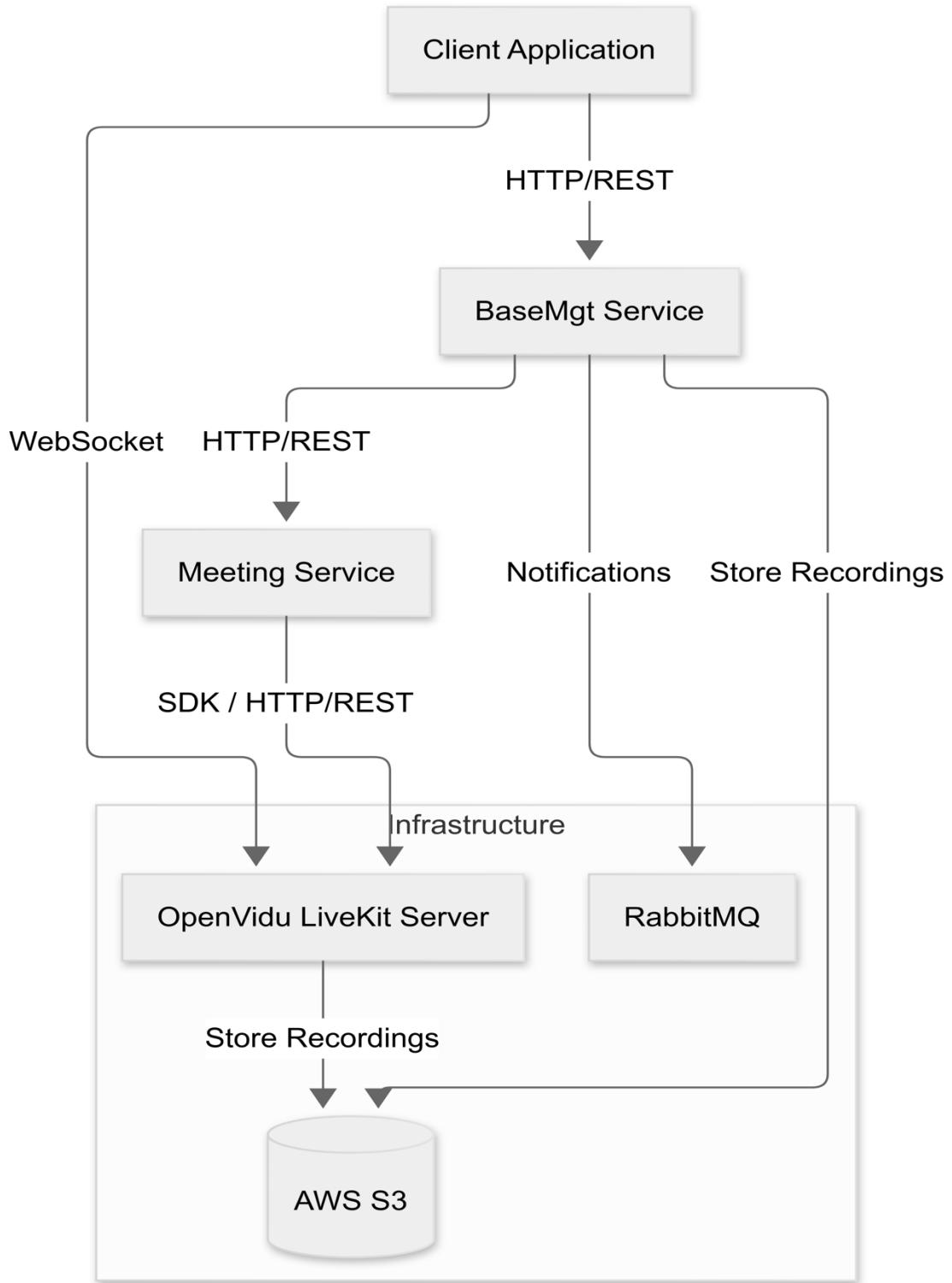


Figure 53 Meeting Service Overall Architecture

5.2 Meeting Service Internal Structure

As shown in Figure 48 below, the Meeting Service acts as a **wrapper** around the OpenVidu Server. This abstraction simplifies interactions with LiveKit, allowing us to expose only specific endpoints while enhancing functionality and incorporating customizations tailored to **AgileMeets'** unique requirements.

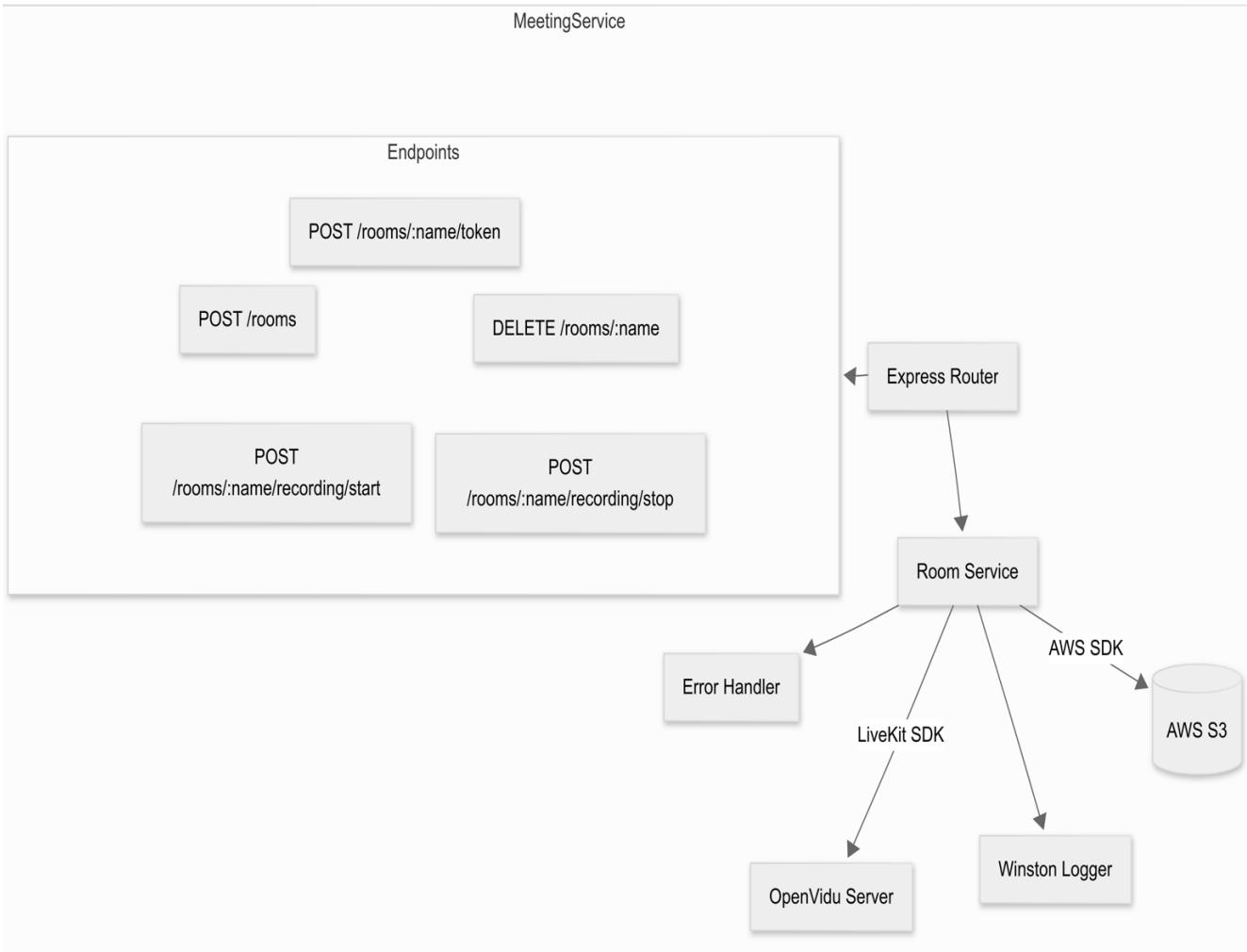


Figure 54 Meeting Service Internal Structure

5.3 BaseMgt Service Integration components

As illustrated in Figure 49, the Meeting Service has been integrated into AgileMeets' primary orchestrator (the BaseMgt Service). This integration is further reinforced with resilient patterns and fault-tolerance policies.

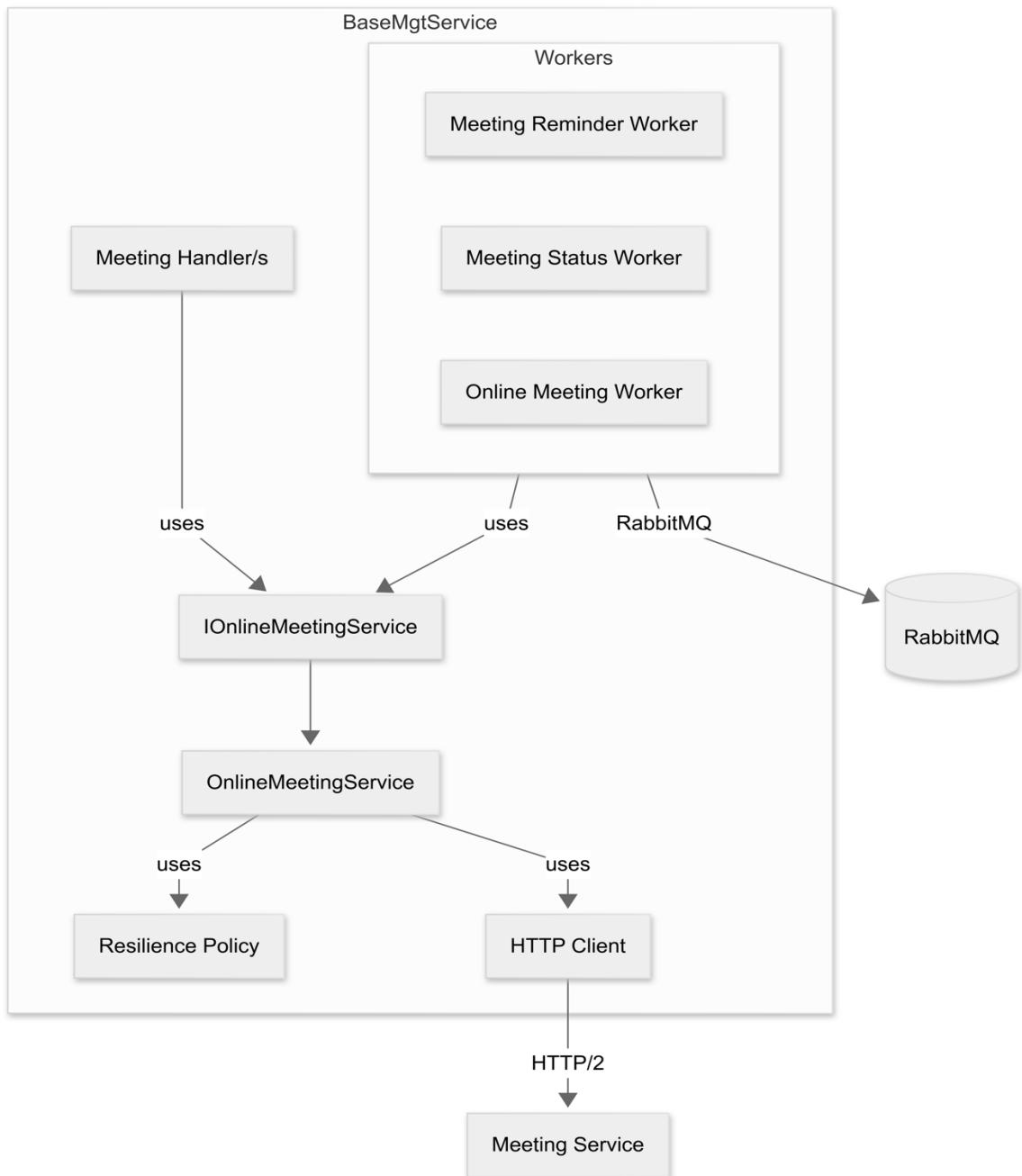


Figure 55 BaseMgt Integration of Meeting Service

5.4 Resiliency around the communication

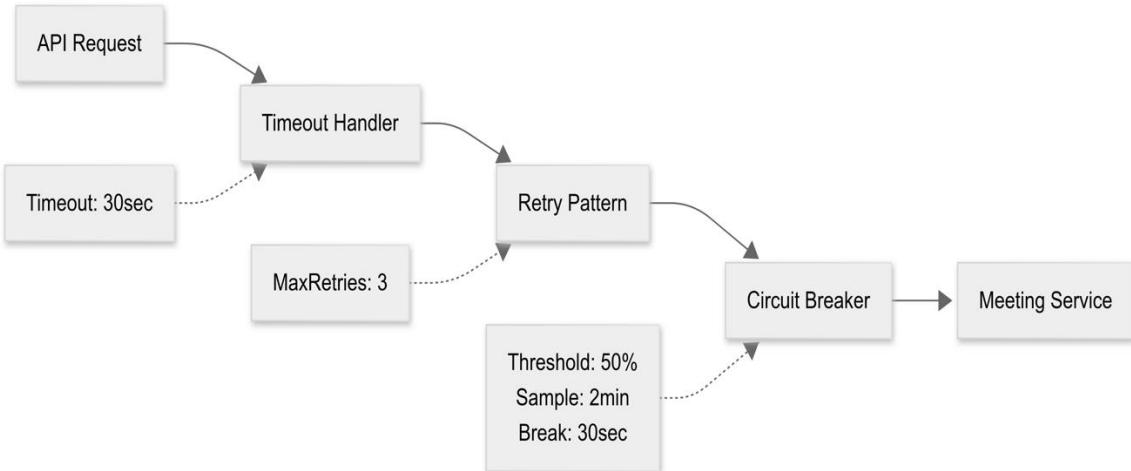


Figure 56 Resilient Integration between BaseMgt and Meeting Service

5.5 Notable Design patterns

5.5.1 Circuit Breaker:

The Circuit Breaker pattern is like an electrical circuit breaker in your home. Its main purpose is to prevent a system from repeatedly trying to execute an operation that's likely to fail.

Three States:

A- Closed (Normal Operation)

- Requests flow normally
- System monitors for failures

B- Open (Failure State)

- Requests immediately return with error
- Prevents system overload
- Gives failing service time to recover

C- Half-Open (Recovery Check)

- Allows limited requests through
- Tests if service has recovered
- Decides whether to close or reopen

Benefits:

1. Fails fast instead of waiting for timeouts
2. Prevents cascade failures
3. Allows graceful degradation
4. Enables quick recovery detection

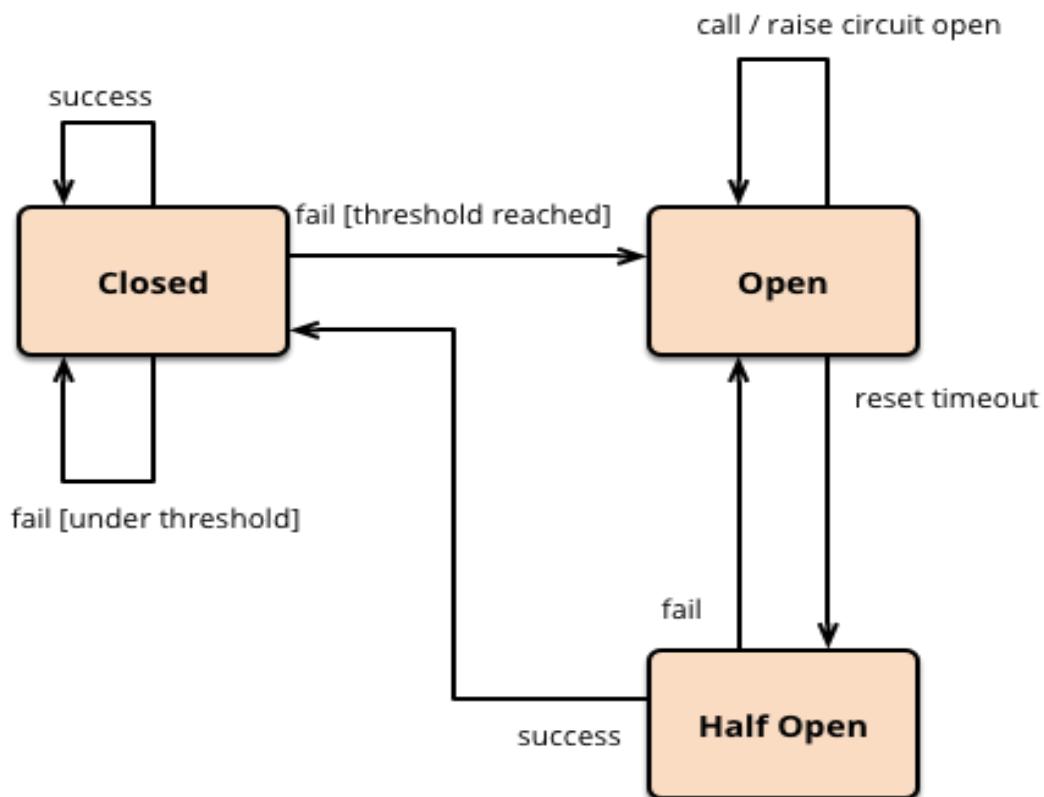


Figure 57 Circuit Breaker Design Pattern

5.5.2 Singleton:

We're Injecting a Singleton Resilience Policy instance for each HTTP Client.

But what's a Singleton?

The Singleton pattern ensures a class has only one instance throughout the entire application lifecycle and provides a global point of access to it.

Key Aspects:

1. Single Instance
2. Only one object created
3. Shared across entire application
4. Global Access
5. Controlled access point
6. Consistent state management
7. Lazy Initialization
8. Created only when needed

Resource efficient Benefits:

1. Ensures state consistency
2. Controls resource usage
3. Provides coordinated actions
4. Centralizes management

6. Screenshots of Client App

In this section we will showcase the main added screens/widgets to AgileMeets app, to provide the new meetings feature.

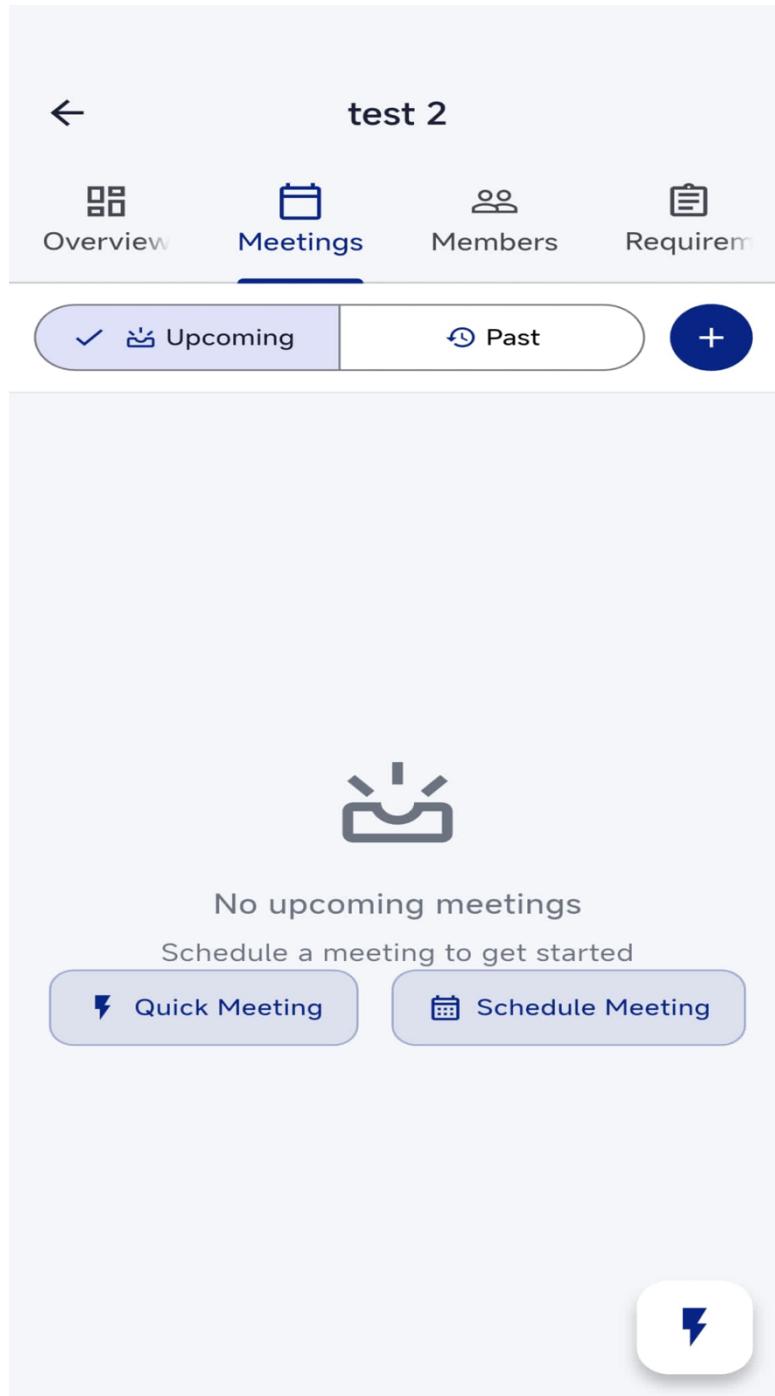


Figure 58 Meetings tab

The screenshot shows the 'Meetings' tab of the AgileMeets Senior2 application. At the top, there is a navigation bar with icons for Overview, Meetings, Members, and Requirements. Below the navigation bar, there are two tabs: 'Upcoming' (selected) and 'Past'. A large blue '+' button is located to the right of the tabs. The main content area displays four past meeting entries:

- Recurring Meeting**: Status: Cancelled. Date: Jan 27 12:42 PM - 1:42 PM. Participants: 4.
- test in spu**: Status: Completed. Date: Jan 27 12:17 PM - 1:17 PM. Participants: 1.
- Saturday, January 25**: Status: Completed. Date: Jan 25 9:49 AM - 10:49 AM. Participants: 1.
- A test**: Status: Completed. Date: (not visible). Participants: 1.

Figure 59 Past Meetings Tab

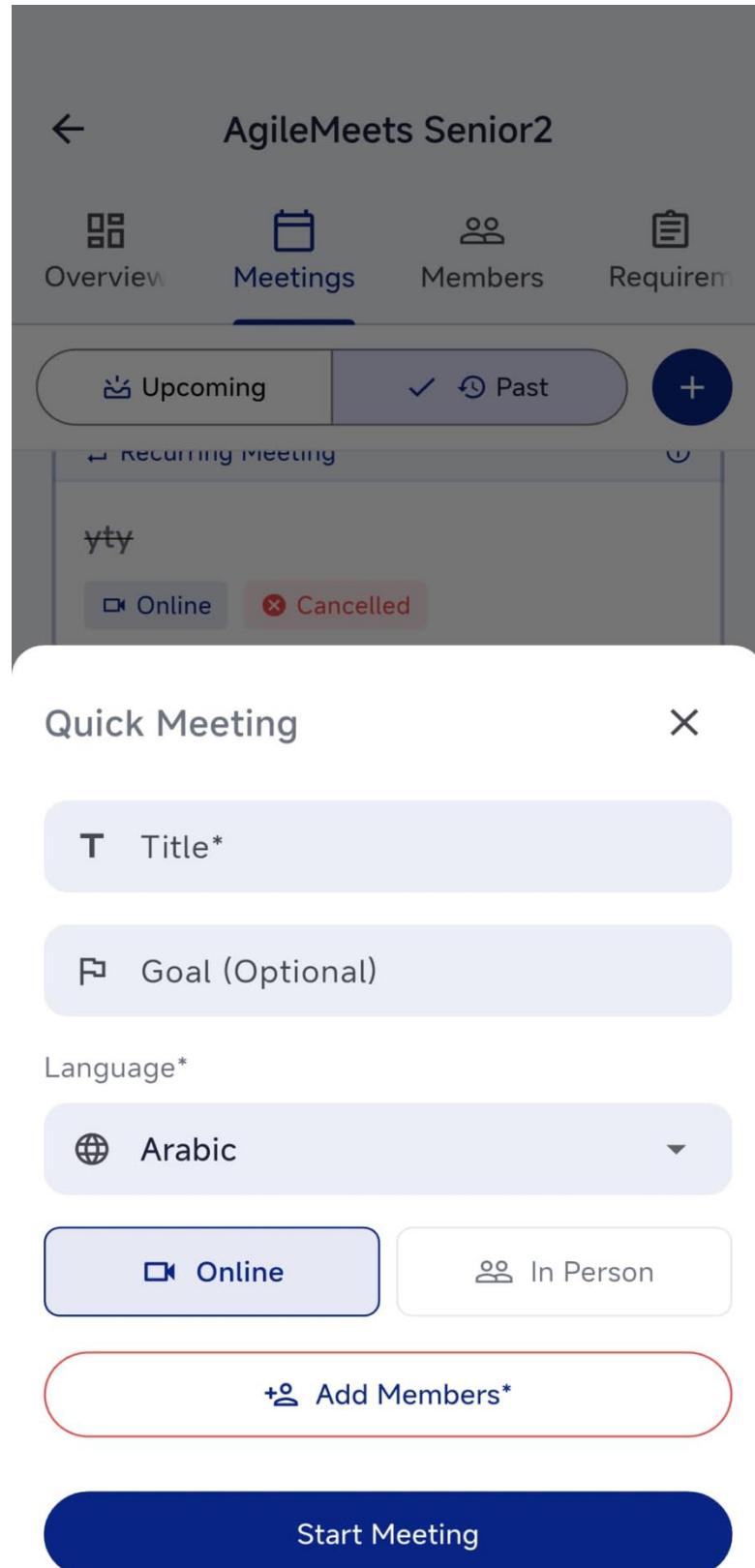


Figure 60 Quick Meeting Sheet

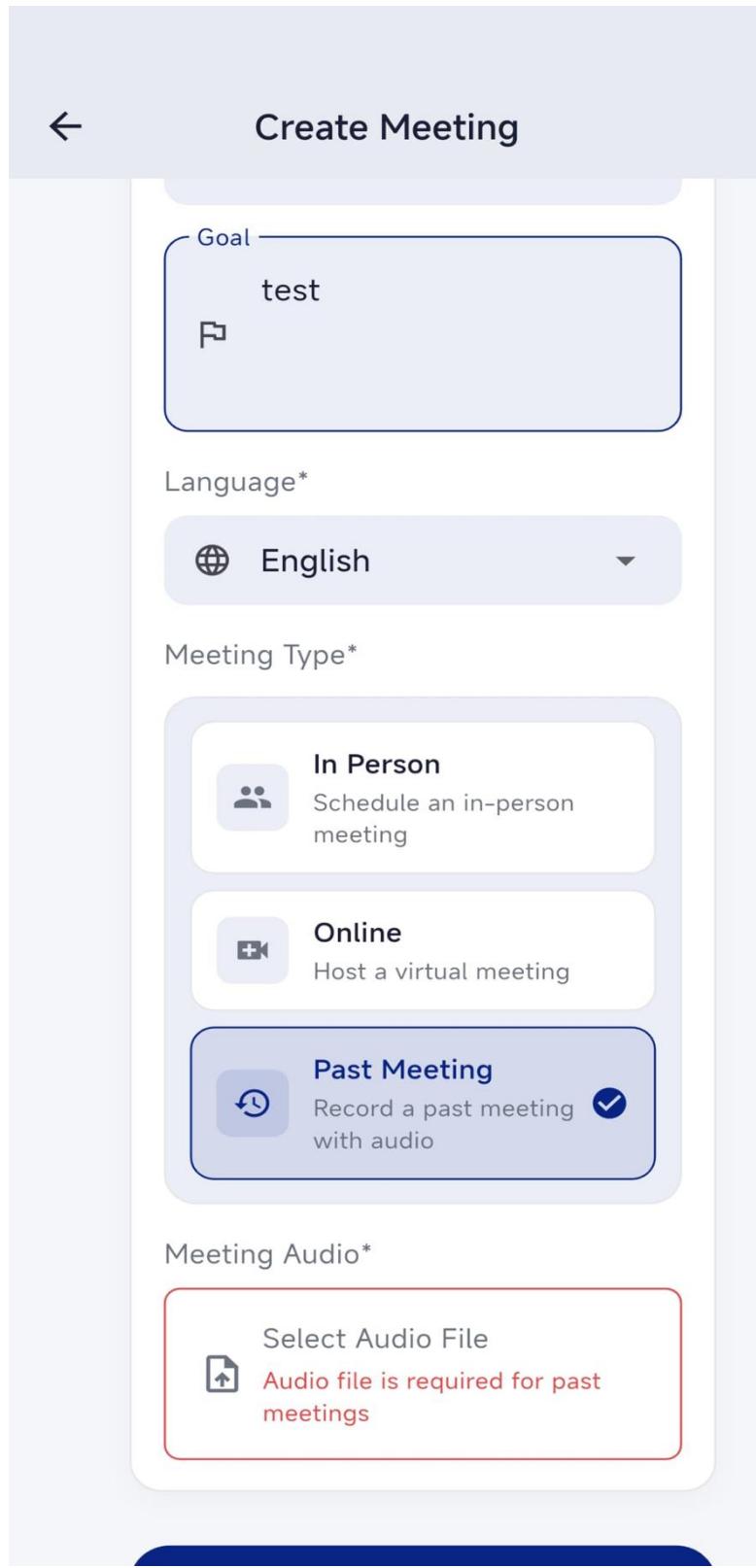


Figure 61 Full meeting Creation Screen

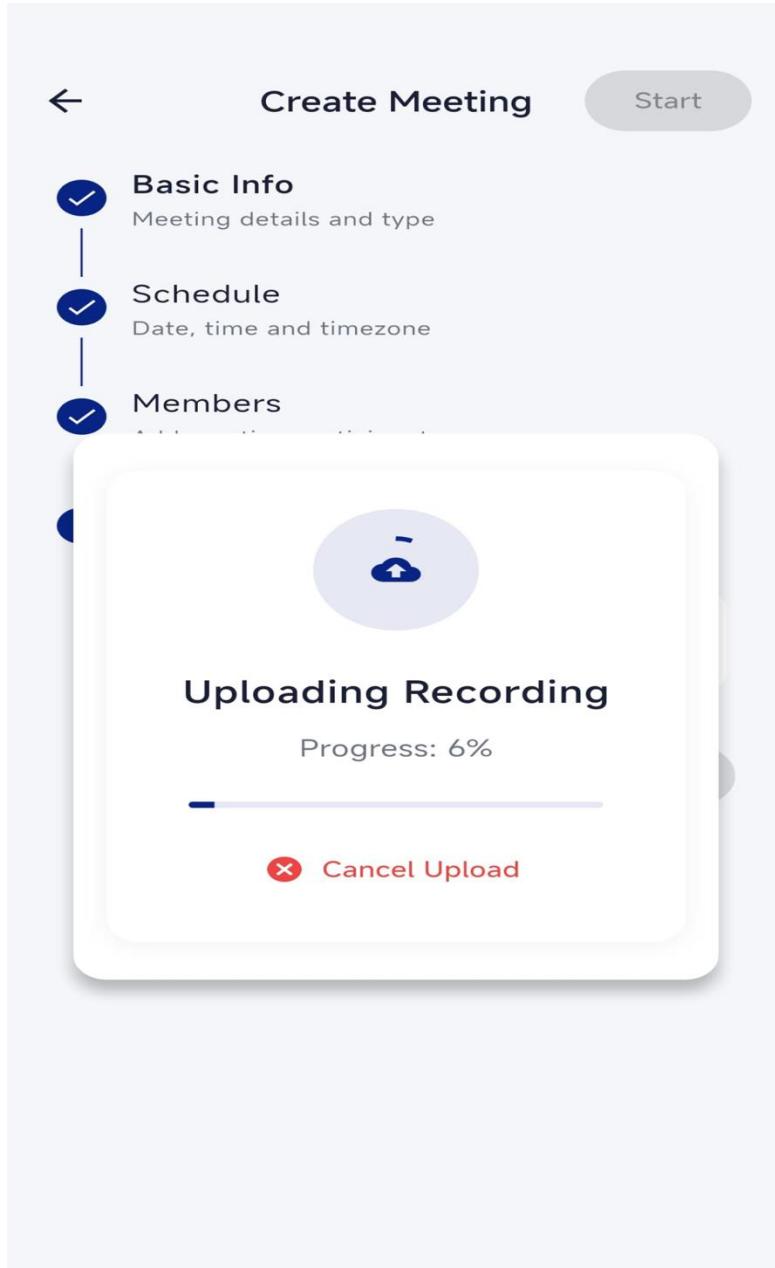


Figure 62 Past Meeting Upload

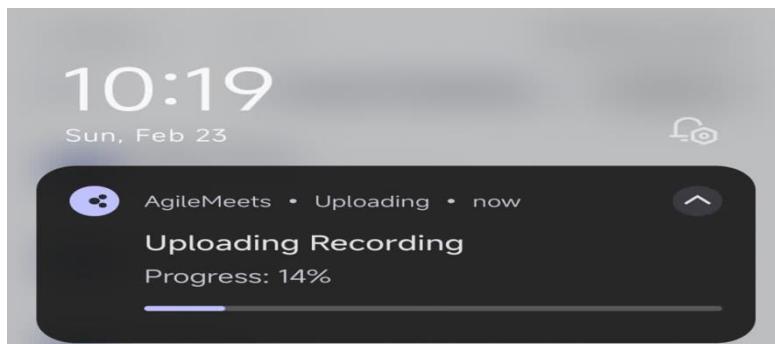


Figure 63 Audio Upload in Background with Notifications

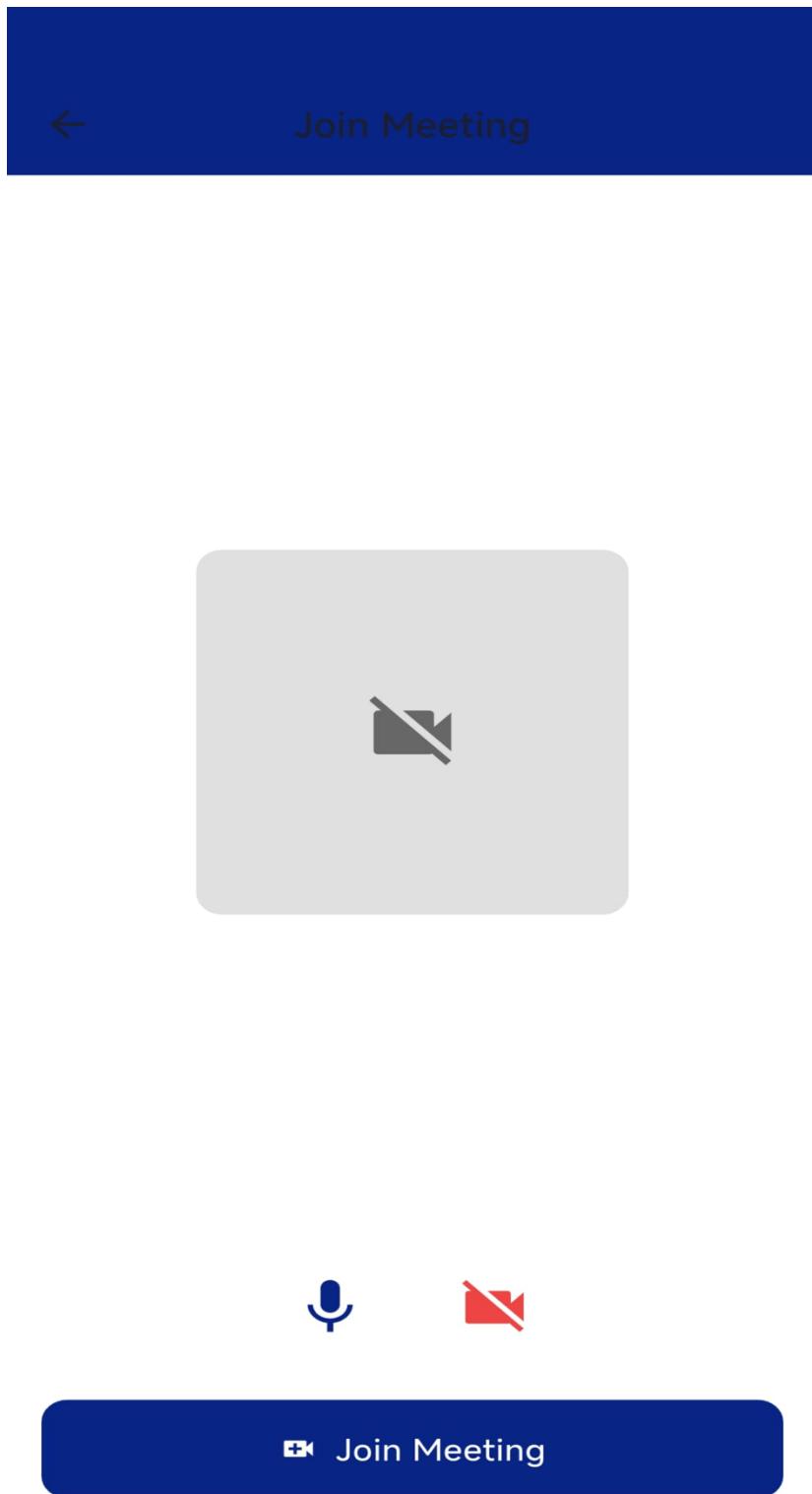


Figure 64 Online Meeting Pre-Join screen

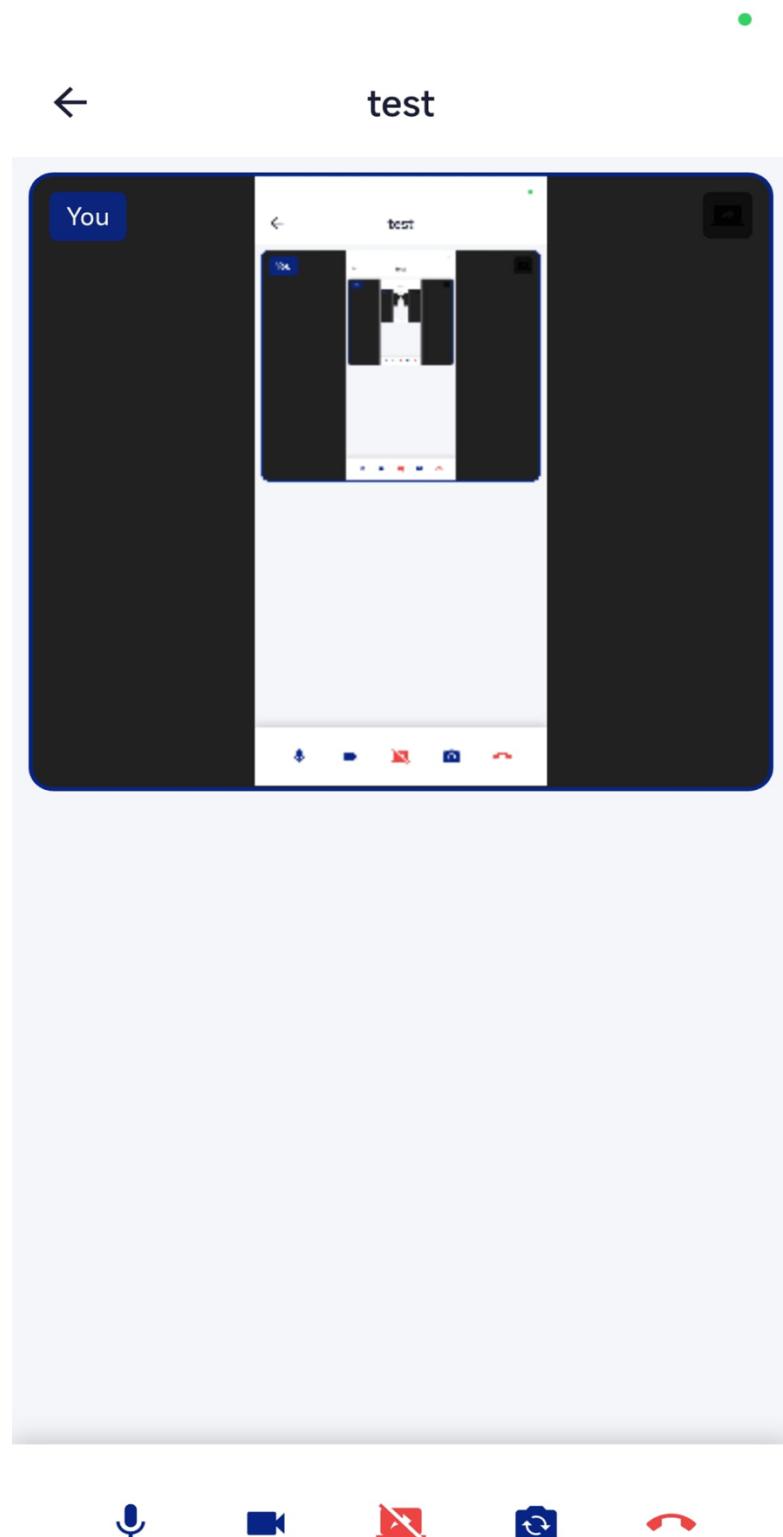


Figure 65 Online Meeting Share Screen Functionality

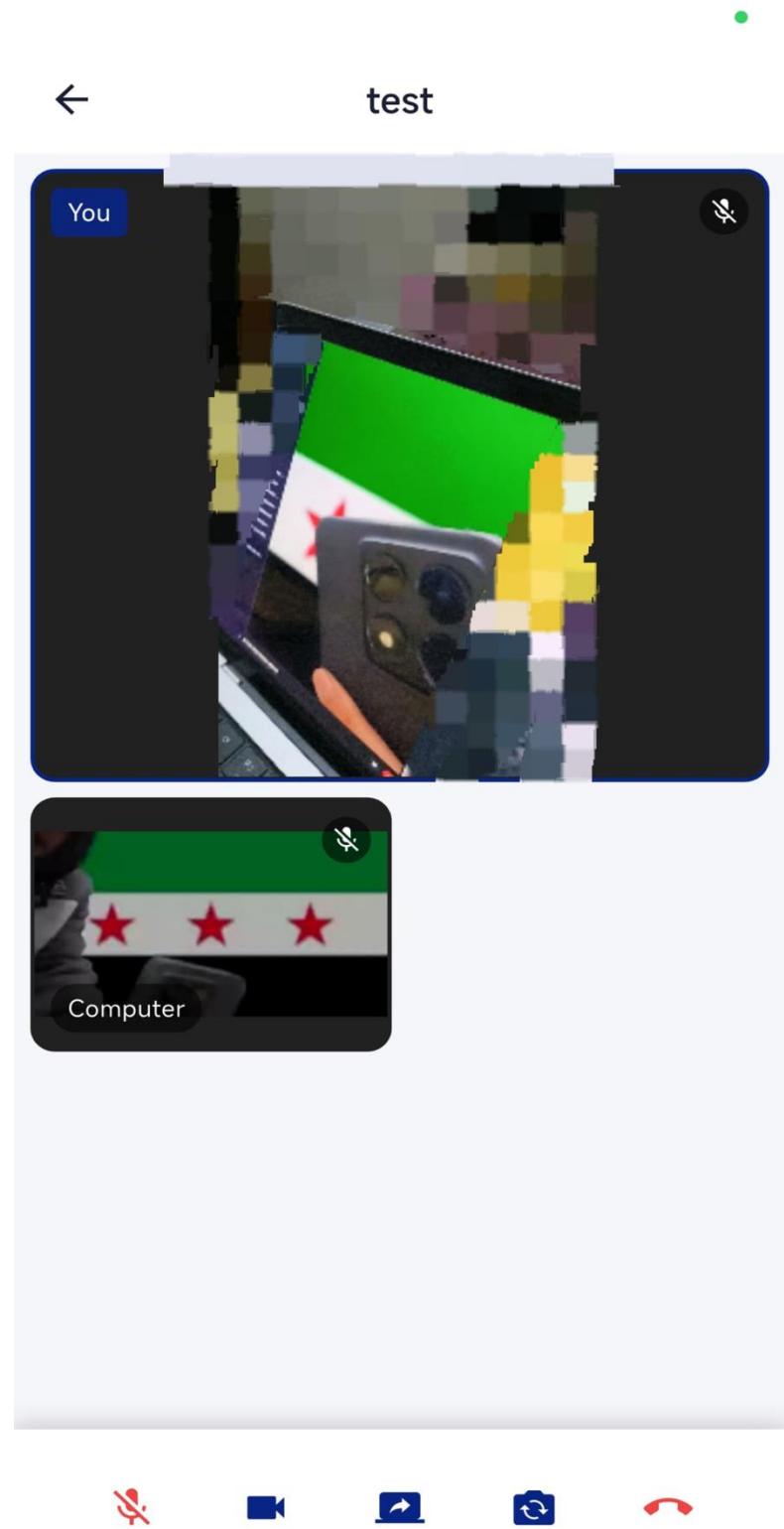


Figure 66 Online Meeting with 2 participants

This Image is Actually Blurred Intentionally (Software Editing)

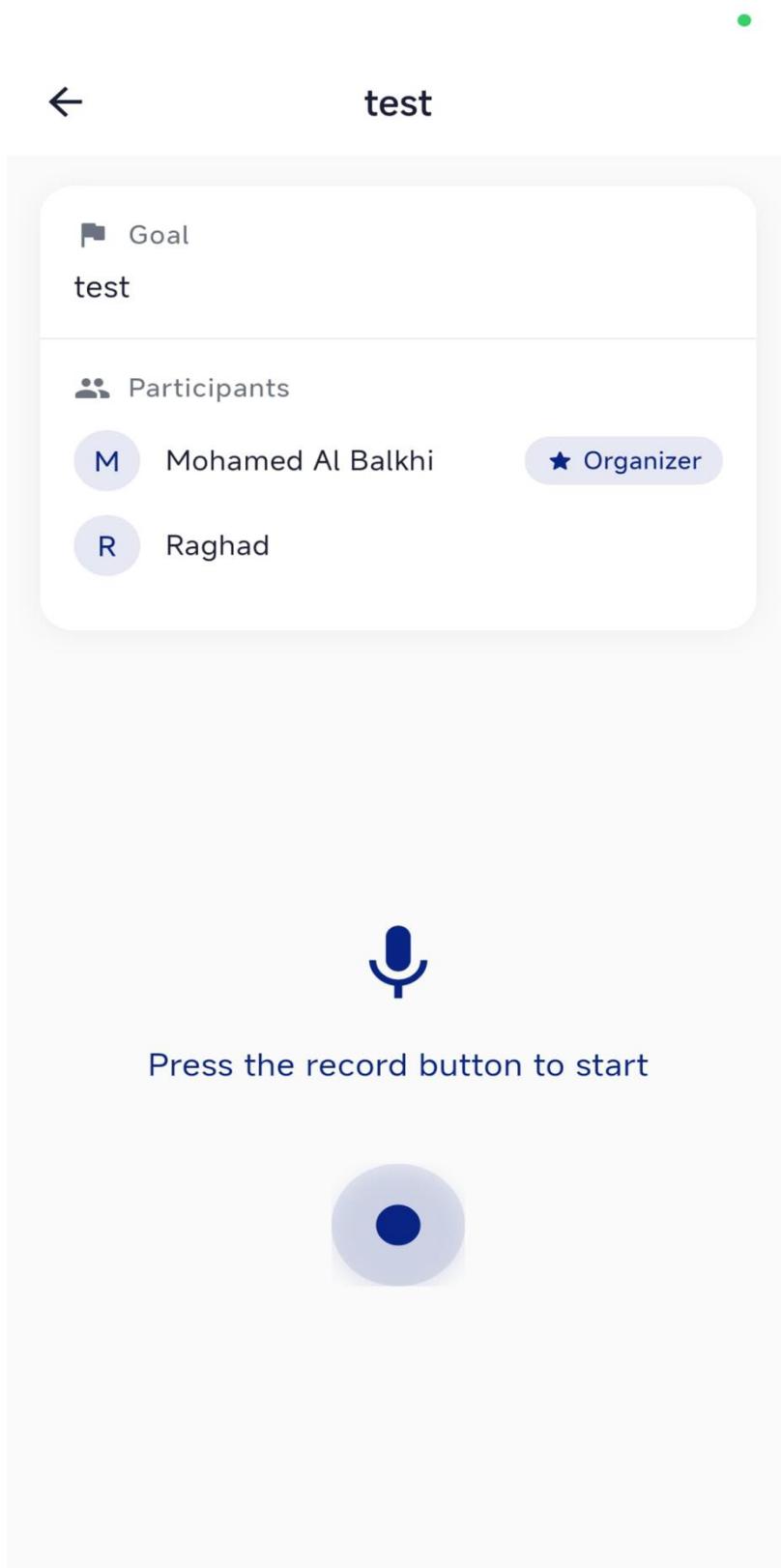


Figure 67 In-person meeting session

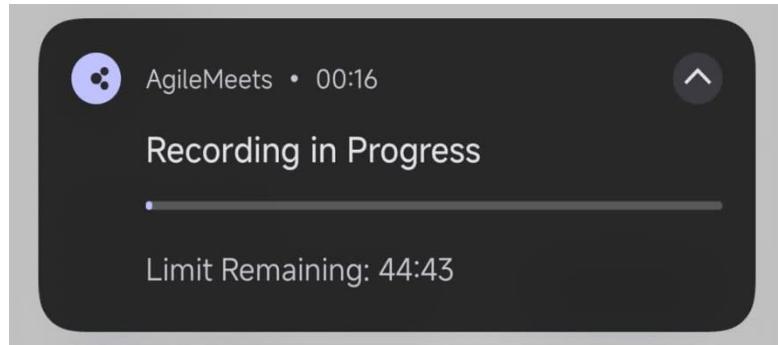


Figure 68 Background-supported Recording with Notification

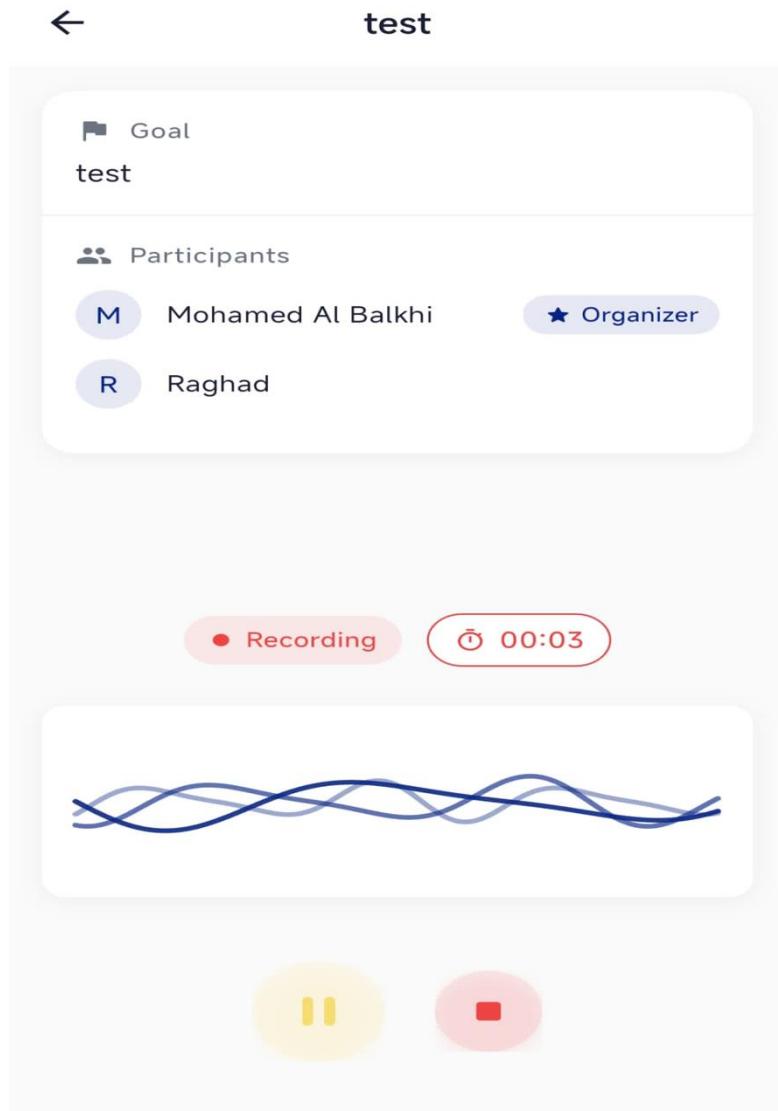


Figure 69 Ongoing Recording In-Person

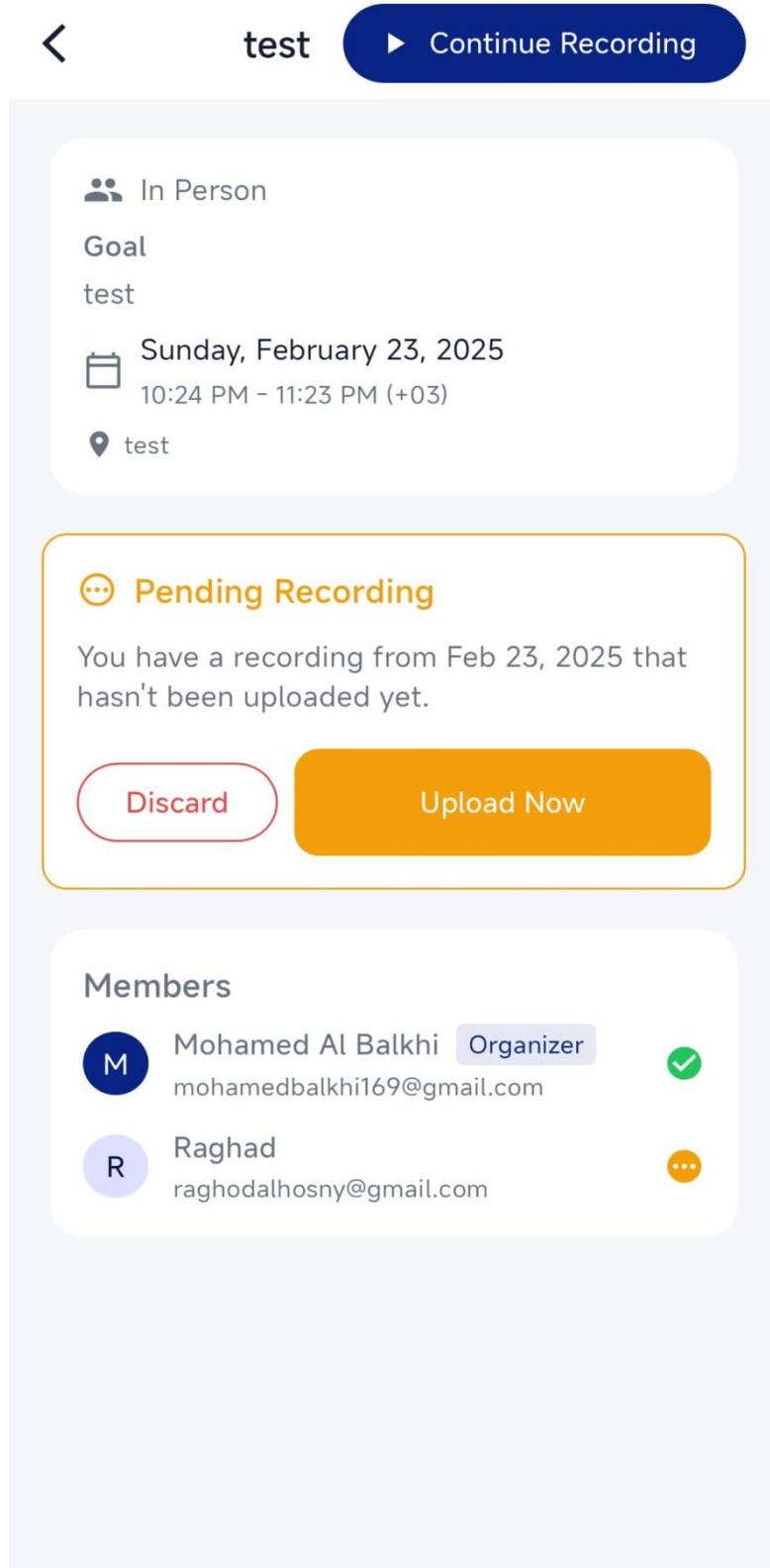


Figure 70 Cached Recording to Upload Later

Arabic Final

Online

Goal
test

Wednesday, January 22, 2025
2:39 AM - 3:38 AM (+03)

Members

M	Mohamed Al Balkhi Organizer mohamedbalkhi169@gmail.com ✓
R	Raghad raghodalhosny@gmail.com ...

Meeting AI Report

Summary

تمت مناقشة تقديم المشروع وتفاصيل الميزات الجديدة، مع التركيز على تحسين اللغة العربية في استخدام تقنيات OpenAI. تم التأكيد على أهمية تنظيم التقارير بطريقة واضحة، مع اقتراح إنشاء فصول لكل خدمة. تم التطرق إلى مشاكل في مستندات المشروع وضرورة تحسينها لتكون مختصرة وواضحة. الاجتماع

0:00 8:06

▶ ⏴ ⏵

Figure 71 Meeting Details & Report

< **test bg past**

Summary

In the transcript, a father expresses his deep love and concern for his daughter, Haley. He acknowledges her sadness and the challenges they face, emphasizing his desire to provide her with a better life despite their struggles. He reflects on their past, including financial difficulties and family issues, while reassuring her that he is always there for her. He seeks to comfort her amid confusion and pain, promising to do anything for her and to keep their bond strong.

Key Points

- A father expresses deep love and concern for his daughter, Haley.
- He acknowledges Haley's sadness and their challenges.
- The father emphasizes his desire to provide her with a better life despite struggles.
- He reflects on past financial difficulties and family issues.
- He reassures Haley that he is always

0:00 4:18

▶ ⏴ ⏵

Figure 72 AI Analysis Outcome Page

7. Future Considerations

- We may customize and benefit from OpenVidu-app which is an out of the box fully functional web app.
- Provide Real time meeting updates & Chatting capabilities.
- Allow taking notes along the meeting.

8. Conclusion

This chapter has demonstrated the thorough analysis and implementation of the Meetings functionality in AgileMeets, highlighting several key achievements:

8.1 Technology Selection

- Justified selection of OpenVidu as the WebRTC solution
- Successfully balanced features, cost, and control requirements
- Implemented scalable single-node architecture with future growth potential

8.2 Implementation Success

- Robust meeting management system supporting multiple meeting types
- Comprehensive calendar integration
- Resilient integration patterns ensuring system reliability
- Effective handling of time zone and language requirements

implementation provides a solid foundation for future enhancements while meeting current business needs effectively.

Chapter 6 AI Service

1. Introduction:

AI is transforming industries by automating processes, enhancing decision-making, and improving efficiency. In software companies, especially agile-driven ones, frequent meetings generate valuable insights, but extracting key takeaways manually is time-consuming and inefficient.

The AI Processing Service in AgileMeets automates this process by leveraging advanced speech-to-text, summarization, and key points extraction capabilities. Designed as an independent microservice, it streamlines meeting documentation, making Agile workflows more efficient.

- Key Functionalities of AI Service:

The AI service processes meeting recordings through three main components:

- Speech-to-Text (STT): Converts spoken conversations into highly accurate text, supporting both Arabic and English.
- Summarization: Extracts key discussions and generates concise, structured summaries for quick review.
- Key Points Extraction: Identifies critical decisions, action items, and essential insights, ensuring teams can focus on what matters.

In this chapter, we will explore the key tools and technologies used, review relevant literature on similar models, and detail the design and testing of the AI Processing Service.

2. Basic Concepts and literature review:

2.1 Introduction:

This section provides an overview of the key concepts, technologies, and tools essential for this AI service. It also includes a literature review of similar models. By understanding these foundational elements, we establish the relevance of our service and how it improves upon current solutions.

2.2 Basic concepts:

- AI pre-trained models:

AI pre-trained models have revolutionized the way artificial intelligence is developed and deployed across various industries. These models are trained on massive datasets, enabling them to understand patterns, recognize structures, and generate intelligent responses without the need for training from scratch. Instead of spending significant time, computational resources, and large amounts of labeled data to develop models from the ground up, pre-trained models offer a ready-to-use foundation that can be fine-tuned for specific tasks.

Leveraging pre-trained models improves efficiency, accuracy, and scalability while significantly reducing the cost of AI development. This approach allows developers to focus on customization and optimization rather than starting with raw data.

- **Transformers:**

Transformers are a deep learning neural network architecture designed to efficiently process sequential data using self-attention mechanisms. Unlike traditional RNNs and LSTMs, which process data sequentially, transformers handle entire inputs in parallel, enabling faster computation and better context understanding. This innovation has become the foundation for modern AI models, including large language models (LLMs) and pre-trained AI models. Pre-trained models, such as GPT and BERT, leverage transformers to learn from massive datasets, allowing them to generalize across various tasks with minimal fine-tuning. The ability of transformers to capture long-range dependencies and relationships within data makes them highly effective for NLP, vision, and other AI applications. By combining pre-training with transformer architectures, AI systems achieve greater efficiency, accuracy, and scalability, making them essential in today's AI advancements.

- **Self-Attention:**

Self-attention is a key mechanism in transformer neural networks that allows models to focus on different parts of the input simultaneously, rather than processing data sequentially. Unlike traditional models like RNNs, which process words in order, self-attention enables the model to assign different importance (weights) to different words or elements in a sequence, regardless of their position. This makes it highly effective for tasks that require understanding long-range dependencies, such as language translation, text generation, and speech recognition.

Self-attention is the foundation of transformers, powering models like GPT, BERT, and T5, which use multiple layers of self-attention to process vast amounts of data efficiently. By leveraging pre-trained models, self-attention enables AI systems to capture context better, improve accuracy, and enhance scalability across various applications, including NLP, STT, and image processing.

- **Large Language Models (LLMs):**

Large Language Models (LLMs) are AI models trained on vast amounts of text data to understand, generate, and manipulate human language. These models, built on transformer architectures, leverage self-attention, and deep contextual understanding to process complex language patterns. LLMs are typically pre-trained on massive datasets and then fine-tuned for specific tasks such as text generation, summarization, and question-answering. Their reliance on pre-training and transformers allows them to generalize across various applications with minimal additional training. Popular models like GPT and BERT exemplify how LLMs use transformers to achieve state-of-the-art performance in natural language understanding and generation. Today, LLMs are revolutionizing industries by enhancing AI-powered chatbots, coding assistants, research tools, and enterprise automation.

- **Speech to text (STT):**

Speech-to-Text (STT), also known as Automatic Speech Recognition (ASR), is an AI technology that converts spoken language into written text. Modern STT systems leverage deep learning models, particularly transformers and pre-trained AI models, to improve accuracy and handle various accents, dialects, and noisy environments. Unlike traditional rule-based methods, STT now benefits from self-attention mechanisms in transformers, allowing it to capture long-range dependencies in speech. Many pre-trained models, such as Whisper and Wav2Vec, are built on these architectures, making STT more efficient and scalable.

- **AI summarization:**

AI summarization is a technique that automatically condenses large amounts of text into shorter, meaningful summaries using advanced natural language processing (NLP) models. There are two main approaches to AI summarization:

Extractive Summarization: This method selects the most important sentences or phrases directly from the original text without modifying the wording. It works by identifying key sentences that best represent the core content.

Abstractive Summarization: Unlike extractive methods, abstractive summarization generates new sentences to create a more natural, human-like summary. It rephrases and restructures the content while preserving the key ideas. Transformer-based models excel at abstractive summarization, producing summaries that are more concise and coherent.

2.3 Literature review:

This section presents an analysis of pre-trained models and existing AI solutions, comparing their performance and suitability for the key functionalities of the AI service. It examines various STT, summarization models, evaluating their strengths, limitations, and how they align with the service requirements.

- **Speech to text models:**

for the speech-to-text component, we will compare **Whisper** from **OpenAI** and **NOVA-02** from **Deepgram**.

Whisper:

Whisper is an automatic speech recognition (ASR) system trained on 680,000 hours of multilingual and multitask supervised data collected from the web. It shows that the use of such a large and diverse dataset leads to improved robustness to accents, background noise, and technical language. Moreover, it enables transcription in multiple languages, as well as translation from those languages into English.

Whisper architecture is a simple end-to-end approach, implemented as an encoder-decoder Transformer. Input audio is split into 30-second chunks, converted into a log-Mel spectrogram (a numerical representation of the sound), and then passed into an encoder. A decoder is trained to predict the corresponding text caption, intermixed with special tokens that direct the single model to perform tasks such as language identification, phrase-

level timestamps, multilingual speech transcription, and to-English speech translation.

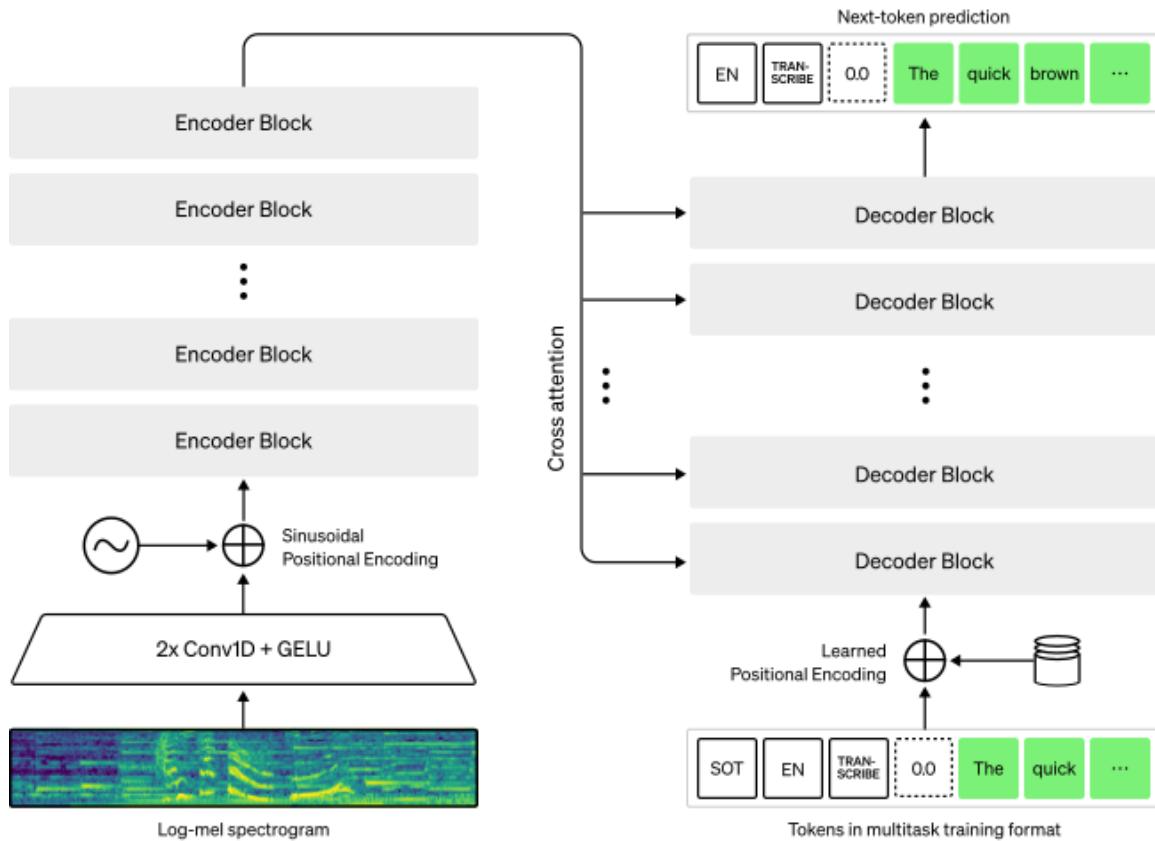


Figure 6.1 whisper model architecture

Architectural details of the whisper model family:

Model	Layers	Width	Heads	Parameters
Tiny	4	384	6	39M
Base	6	512	8	74M
Small	12	768	12	244M
Medium	24	1024	16	769M
Large	32	1280	20	1550M

Figure 73 Whisper model family

Nova-02:

Nova-2 is an advanced Transformer-based speech-to-text model designed to enhance accuracy, efficiency, and real-time transcription performance. It builds upon the original Nova-1 model, introducing significant optimizations that result in an 18.4% reduction in word error rate (WER).

The architecture is structured into two key components:

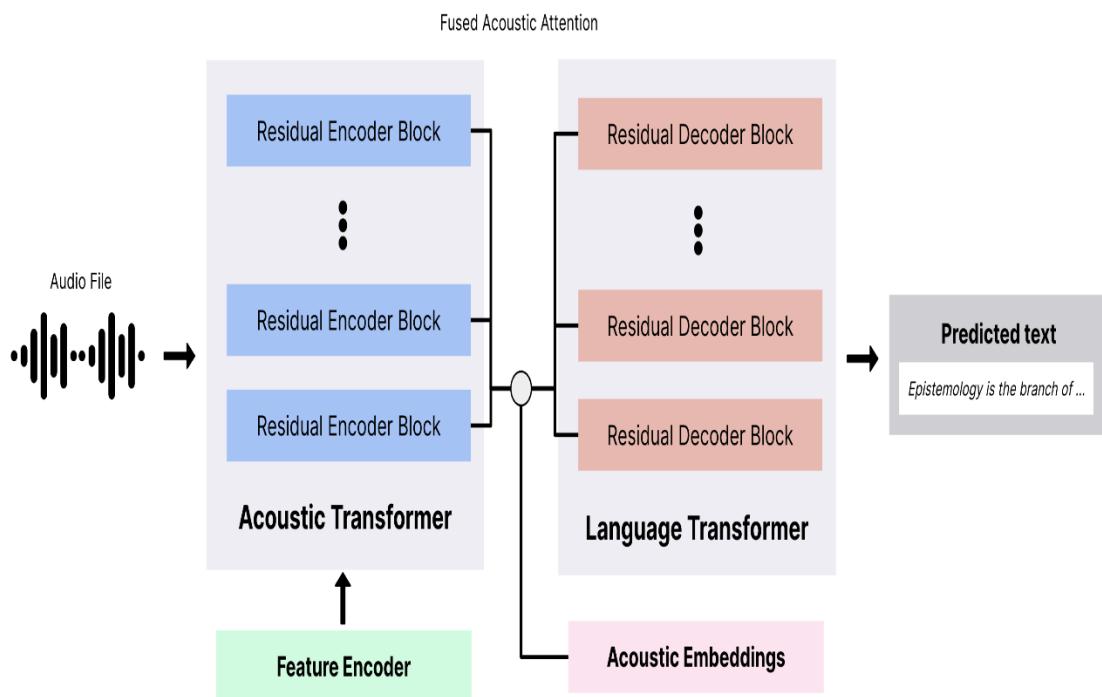


Figure 74 Nova2 architecture

- Acoustic Transformer – Encodes raw audio waveforms into a sequence of audio embeddings, capturing complex sound patterns.
- Language Transformer – Decodes these embeddings into text, leveraging an input prompt for context-aware transcription.

Nova-2 introduces a universal attention mechanism, enabling efficient information flow between these two networks, and improving both speed and transcription accuracy. Architectural enhancements also enhance punctuation, capitalization, and smart entity recognition (e.g., names, account numbers, dates).

Comparison Tests:

Table 20 whisper vs. Nova 2

	Whisper - locally	Nova-02
Usage way	Locally – offline	API communication.
Usage limits	Free	Pay-as-you-go plan that includes \$200 in free credits.
Size	tiny: ~150MB base: ~290MB small: ~970MB medium: ~3GB large: ~6GB	
Multilingual model	True	True (except Arabic)
Process many languages for the same audio	True, with translation.	True (except Arabic) “If you are submitting multichannel audio, Language Detection identifies one language per channel.”
Scalability	Limited by local machine power.	Scales easily on the cloud.

Test cases:

- **Case 01:** non-native speaker – one person – English language.

Table 21 STT Case 01 non-native speaker

Audio duration	00:02:16		
Words no.	156		
	Whisper – medium 3GB	Whisper – small	Nova-02
Processing duration	<ul style="list-style-type: none"> - Without language specification: 2.25 m - With language specification EN 2.23 m 	<ul style="list-style-type: none"> - Without language specification: 34.9 s - with language specification EN 32.1s 	2.40s
RAM usage	3GB	970 MB	0
Accuracy criteria	<ul style="list-style-type: none"> - Wrong Words: 2 - Correct Words: 154 - Original Word Count: 156 - Output Word Count: 156 - Accuracy Percentage: 98.72% 	similar to the whisper medium.	<ul style="list-style-type: none"> - Wrong Words: 1 - Correct Words: 155 - Original Word Count: 156 - Output Word Count: 156 - Accuracy Percentage: 99.30%
Wrong words:	<ul style="list-style-type: none"> - "Throughout" (origin) vs. "Through" (model) - "Financial modules" (origin) vs. "financial models" (model) 		<ul style="list-style-type: none"> - financial models" (model) vs. "financial modules" (origin).
Differences:	real-time user-friendly	real-time user-friendly	Real time User friendly

- **Case 02:** 5 persons English conversation – native speakers.

Table 22 STT Case 02 native speakers

Audio duration	00:01:19		
Words no.	111 words		
	Whisper – medium 3GB	Nova-02	Whisper – small
Processing duration	1.4 m	2.47s	36s
RAM usage	3GB	0	970MB
Accuracy criteria	<ul style="list-style-type: none"> - Wrong Words: 0 - Correct Words: 111 - Original Word Count: 111 - Output Word Count: 111 - Accuracy Percentage: 100% 	<ul style="list-style-type: none"> - Wrong Words: 1 - Correct Words: 110 - Original Word Count: 111 - Output Word Count: 111 - Accuracy Percentage: 99.9% 	<ul style="list-style-type: none"> - Wrong words: 1 - Correct 110 - Original count: 111 - Output: 111 - Accuracy 99.9%
Wrong words:		<ul style="list-style-type: none"> - Oh, ear. No (model) - Oh! Eww! No! (Origin) 	<ul style="list-style-type: none"> - Oh, ear. No (model) - Oh! Eww! No! (Origin)
Differences	Every number is a number here “By 11! 11. Unbelievable.”	Every number is a word: “Eleven? Eleven. Unbelievable. Eleven is correct”	Every number is a word: “Eleven? Eleven. Unbelievable. Eleven is correct”

- **Case 03:** Arabic Language – native speaker.

Table 23 STT Case 03 Arabic Native speaker

Audio duration	00:01:38		
Words no.	200 words		
	Whisper – medium 3GB	Whisper –small	Nova-02
Processing duration	3:76 m specifying Arabic lang. 3:97 m without specifying Arabic lang.	1.4 min	Doesn't support the Arabic language.
RAM usage	3GB	970MB	
Accuracy criteria	<ul style="list-style-type: none"> - Wrong Words: 5 - Correct Words: 195 - Original Word Count: 200 - Output Word Count: 200 - Accuracy Percentage: 97.5% 	<ul style="list-style-type: none"> - Wrong words: 12 - Correct words: 188 - Original count: 200 - Output count: 201 - Accuracy: 90% 	
Wrong words:	<ul style="list-style-type: none"> - طلبتم: تطلبتم - شركتكم: شركة تكم - متاحاً: متاحن - قصارة جهتنا: القصارة جهتنا - لتعاونك: لتعالنك 	<ul style="list-style-type: none"> - اليكـنـك : هل يمكنـكـ - الحمد للـلهـ لاـيـ خـيرـ : الحمد للـلهـ بـخـيرـ. - متاحـةـ : متاحـاـ - معـ برـامـشـ : نـعـ برـامـجـ. 	

- We can also use whisper over API communication which saves resources and increases performance.

- **Text Summarization Models:**

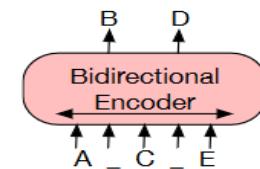
For the speech-to-text component, we will compare **BART** from **Meta**, **Deepgram** summarization, and **LLM chat GPT 4o mini**.

BART:

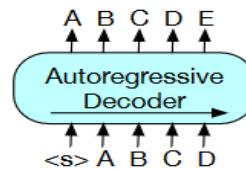
Is a Transformer-based model that combines bidirectional and auto-regressive learning, making it highly effective for both text comprehension and generation. It functions as a denoising autoencoder, following a two-phase pretraining process:

1. corrupting input text using a noising function.
2. training a sequence-to-sequence model to reconstruct the original text.

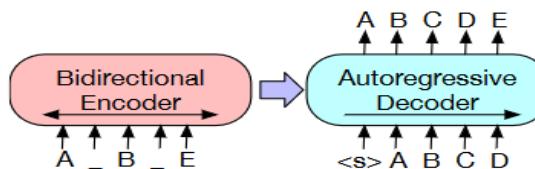
BART's architecture generalizes existing models its bidirectional encoder resembles BERT, while its left-to-right autoregressive decoder is similar to GPT. This hybrid structure enables it to excel in natural language understanding (NLU) and text generation tasks.



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitrary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Figure 75 BART Bidirectional Encoder & Autoregressive Decoder

Bart – family:

Model	Description	# params
<code>bart.base</code>	BART model with 6 encoder and decoder layers	140M
<code>bart.large</code>	BART model with 12 encoder and decoder layers	400M
<code>bart.large.mnli</code>	<code>bart.large</code> finetuned on MNLI	400M
<code>bart.large.cnn</code>	<code>bart.large</code> finetuned on CNN-DM	400M
<code>bart.large.xsum</code>	<code>bart.large</code> finetuned on Xsum	400M

Figure 76 Bart model family

T5 Text-to-Text Transfer Transformer:

T5 is a Transformer-based model developed by Google Research, designed to unify various natural language processing (NLP) tasks into a text-to-text framework. Unlike traditional models that handle tasks differently, T5 converts every NLP problem into a text generation task, allowing it to perform translation, summarization, question answering, and text classification within the same architecture.

T5 is pre-trained using a denoising objective, where parts of the input text are masked or corrupted, and the model learns to reconstruct the original text. This approach enhances its understanding of language structure and improves generalization across tasks.

T5 – family:

- **T5-Small:** 60 million parameters
- **T5-Base:** 220 million parameters
- **T5-Large:** 770 million parameters
- **T5-3B:** 3 billion parameters
- **T5-11B:** 11 billion parameters

Figure 77 T5 Model family

Chat GPT 4o mini:

ChatGPT-4o Mini is a large language model (LLM), a lighter and optimized version of OpenAI's GPT-4o. It contains approximately 8 billion parameters, making it more lightweight compared to larger models like GPT-4o, which boasts over 200 billion parameters. It is designed for efficiency, speed, and cost-effectiveness while maintaining strong NLP performance. As an LLM based on Transformer architecture, it is capable of handling a wide range of natural language understanding and generation tasks, including conversation, **text summarization**, translation, and coding assistance.

ChatGPT-4o Mini follows the **decoder-only Transformer architecture**.

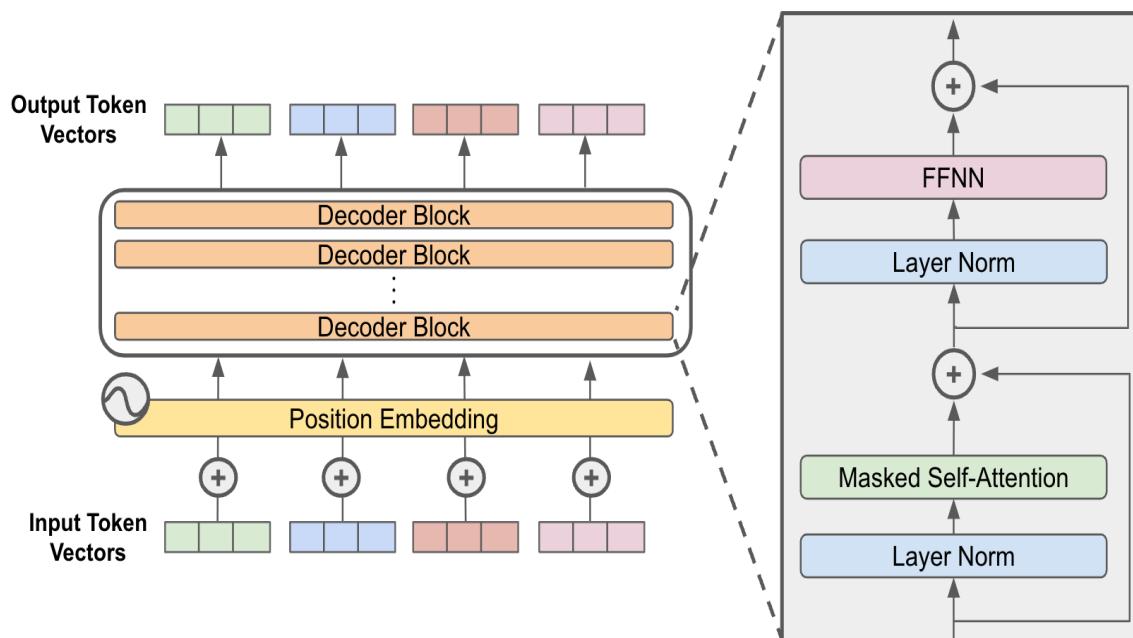


Figure 78 decoder-only Transformer architecture

meaning it generates responses autoregressively, predicting one token at a time based on previous inputs. Key features of its structure include:

- Multi-layer Transformer blocks for deep contextual understanding.
- Self-attention mechanisms to capture relationships between words across long contexts.
- Positional encodings to retain word order information in sequences.
- Fine-tuning on dialogue datasets to enhance conversational coherence and instruction-following.

Test case: meeting transcript about school management system.

The text transcript:

“thank you so much for meeting with us today. at xyz school we’re looking for a system to help us manage things more efficiently. hmm right now we’re using a lot of paper and spreadsheets and honestly it’s becoming a headache. oh absolutely. that makes sense. thank you for sharing that with us. so let’s start by understanding the core areas where you’re facing challenges.

sure. the biggest issue is managing student data. we’ve got over five hundred students. and keeping track of grades attendance and fees. well it’s a huge task you know.

ah i see. hmm so having a centralized system for student data is a top priority right. could you tell us a bit more about what you’d want that system to do.

yes of course. first. we need a detailed student profile section. something that includes you know their personal details grades and maybe even emergency contact info. then we’d like to have automated grade tracking. and a way to generate reports quickly.

got it. that makes sense. so a centralized profile automated grade tracking. what about attendance. is that also part of what you’d like in the system.

oh yes definitely. right now. it’s all manual. we’d need a way to track attendance and generate monthly reports for both teachers and parents.

hmm i wonder. would you also like to notify parents automatically if a student is absent.

oh. that would be wonderful actually. yes. automatic notifications would be a great addition.

great idea. okay so we’ll include that in the attendance module as a notification feature. hmm now what about fees. is that also an area where you face challenges.

oh yes. fees are a major issue. we’d like an online payment system that sends automatic reminders to parents about pending payments.

okay and just to confirm. would you need detailed reports for fee collection. like late payments monthly totals that sort of thing.

hmm yes definitely. detailed reports are a must for us.

Chapter 6 – AI Service

alright. so an online fee payment system with reminders and financial reporting. are there any other areas like teacher management or scheduling that you'd want the system to handle.

actually. hmm yes. we'd need a way to manage teacher profiles their schedules and assigning tasks or classes.

okay. a teacher management module with scheduling and task assignments. got it. and just checking. do you have any existing systems or tools that you'd like this to integrate with.

hmm no not really. we're starting fresh.

oh okay. that's good. it keeps things simpler.

hmm before we wrap up. let's schedule our next meeting to finalize the details and review the proposal. would next Wednesday at 10 a.m. work for you.

yes that sounds good. next Wednesday at 10 works perfectly.

perfect. i'll also include a draft of the financial agreement for review before then. once we finalize that we can move forward.

sounds great. thank you for all your help today.

thank you too. it was a pleasure speaking with you. we'll see you next week. take care and have a great day.

you too. goodbye."

To evaluate the accuracy and effectiveness of the models, we will use a combination of automated metrics (ROUGE) and Human Evaluation to ensure both quantitative and qualitative assessment of the generated outputs.

- ROUGE Metrics (Recall-Oriented Understudy for Gisting Evaluation):

ROUGE is a widely used evaluation metric for text summarization, designed to measure word overlap between generated summaries and reference summaries.

Reference summary:

"The meeting discussed transitioning to a centralized school management system to replace manual processes. Key requirements include student management with profiles containing personal details, grades, emergency contacts, automated grade tracking, and attendance monitoring with notifications for absences. Fee management requires online payments, automated reminders for pending payments, and financial reporting for late payments and monthly totals. Teacher management includes scheduling, task assignments, and profiles. The team confirmed no integration with existing systems is needed. A follow-up meeting is scheduled for next Wednesday at 10 a.m. to finalize details and review the financial agreement."

Since direct comparison of text outputs as binary classification is not feasible, ROUGE calculates the overlapping words and their weights to assess the quality of the generated content. The variations used are:

ROUGE-1: Measures overlap of unigrams (single words) between the generated and reference text, assessing basic word coverage.

ROUGE-2: Measures overlap of bigrams (two-word sequences) to evaluate local contextual accuracy.

ROUGE-L: Uses the longest common subsequence (LCS) to assess sentence fluency and coherence beyond simple word overlap.

Table 24 Rouge Metrix

Model	Rouge-1	Rouge-2	Rouge-L
BART	0.3309	0.0876	0.2302
T5 01 small	0.2824	0.0714	0.1647
T5 02 small	0.2824	0.0714	0.2824
T5 03 large	0.2914	0.0403	0.1457
Deepgram summarization	0.466	0.124	0.294
GPT 4o mini	0.55	0.21	0.42

- Human Evaluation

Since automated metrics alone do not fully capture semantic accuracy and readability, we will conduct a manual evaluation based on key quality criteria.

Each criterion is scored on a 1-5 scale, assessing:

Relevance: Does the summary or extracted key points accurately reflect the main topics discussed?

Coherence: Are the outputs grammatically correct, logically structured, and easy

to understand?

Conciseness: Does the summary avoid unnecessary repetition while effectively conveying key information?

Coverage: Does the output include all major points without omitting critical details?

Table 25 human evaluation – summarization models

Model	Relevance (1–5)	coherence (1–5)	conciseness (1–5)	coverage (1–5)
BART	4	4	4	3
T5 small	3	3	2	1
T5 large	4	3	2	2
Deepgram summarization	4	4	4	4
GPT 4o mini	5	5	5	3

- Performance Testing: response time per model

Table 26 performance testing- summarization models

model	BART (locally)	Deepgram summarization (API)	T5 large (locally)	T5 small (locally)	GPT 4o-mini (API)
Response time (sec)	16s	3s	46s	13s	3s

3. Service Design and implementation:

3.1 Introduction:

In this section, we will present the design of our AI service and the architecture it follows to efficiently process audio inputs. The service is built to transform audio into a structured report, including transcripts, summaries, and key points, through a scalable and modular pipeline. Leveraging cutting-edge tools and models.

3.2 AI Service Design:

- Service Functional requirements:

- The service must support receiving audio files for processing.
- The service must be capable of downloading and retrieving audio from external storage sources, such as Amazon S3, for processing.
- The service must be able to convert received audio files into text transcripts with high accuracy.
- The service must generate a concise and coherent summary from the transcribed text produced by the STT component.
- The service must extract key points from the summarized text, ensuring clarity and relevance.
- The service must extract key points and action items from the summarized text, ensuring clarity and relevance.

- The service must provide an API interface for external services or users to utilize its pipeline processing capabilities.
 - The service must store and track the status of audio processing at each stage (STT, Summarization, Key Extraction).
 - The service must provide an API endpoint to retrieve the real-time status of audio processing.
 - The service must provide an API endpoint to fetch the final processing report, including the transcript, summary, and extracted key points.
- Non-functional requirements:
- **Usability:** The service must support processing audio in both Arabic and English, ensuring multilingual capability.
 - **Security:** all API endpoints must be protected by an API secret key, ensuring that only authorized services can access the AI processing pipeline.
 - **Performance:** The service must process audio within a short response time, ensuring minimal latency and optimal user experience max 30 sec.
 - **Scalability:** The service must handle multiple processing requests efficiently, using queue-based processing and task management, ensuring seamless workload distribution and system scalability.

- Processing Flow:

We designed our AI service to seamlessly process audio inputs, transforming them into structured and meaningful outputs (final report).

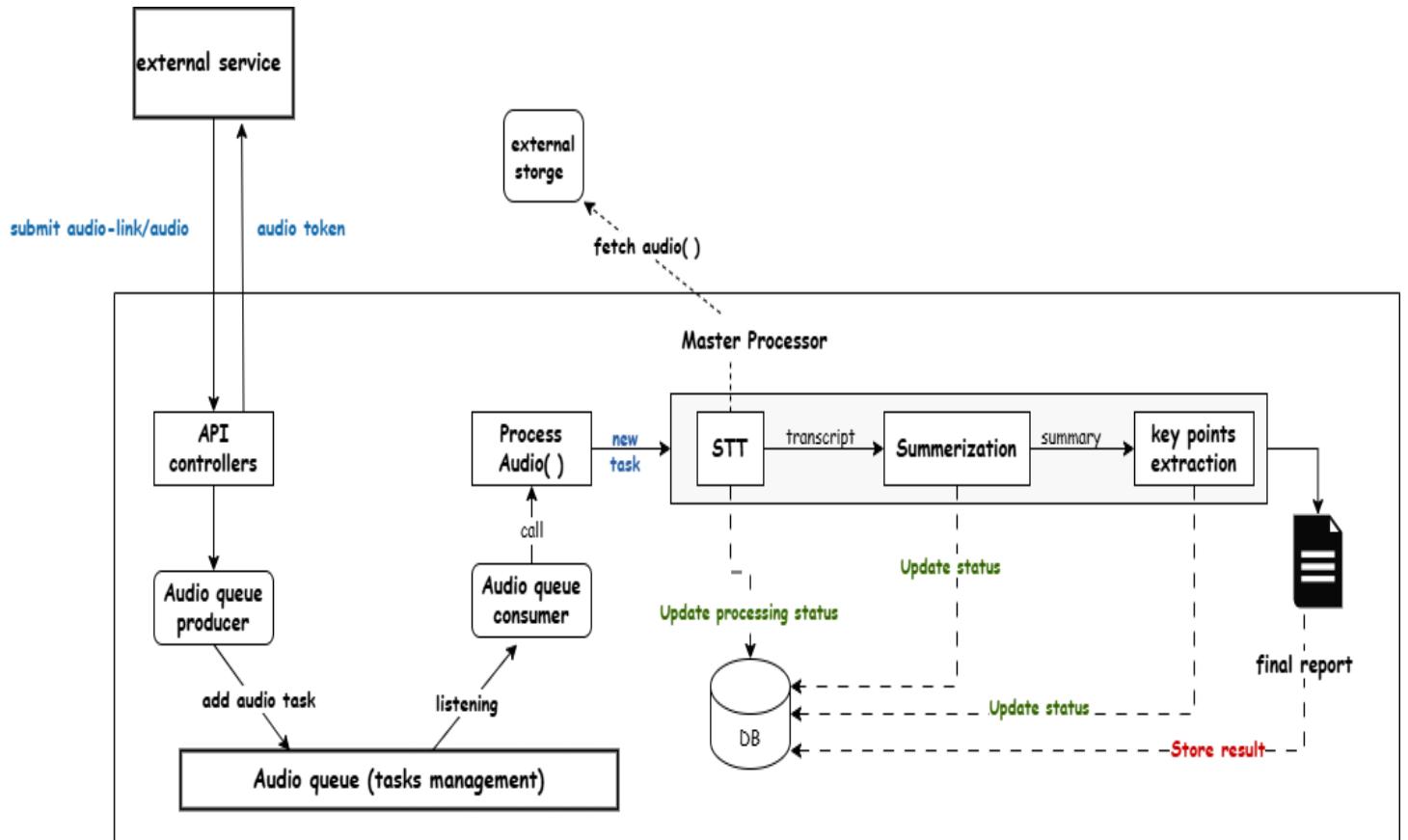


Figure 79 AI service processing flow

The AI service processes audio data through a structured flow, ensuring efficiency and scalability. When an external service submits an audio file or link, the system assigns a unique token for tracking. The request is placed in a queue, which manages and distributes tasks efficiently.

The queued task is processed in sequential stages: Speech-to-Text (STT) converts audio into text, followed by Summarization generating a concise

summary, and key points extraction component. Finally, the processed data is compiled into a final report. The system updates the status throughout this process to track progress and completion.

The importance of queue:

Using a queue system is essential for handling multiple requests efficiently. It prevents system overload, ensures fair task distribution, and enables asynchronous processing, allowing multiple audio files to be processed concurrently without blocking new requests. This improves scalability, optimizes resource utilization, and ensures smooth handling of large workloads.

- The service architecture:

The service follows **Pipe-and-Filter** architecture, where audio data flows through a sequential processing pipeline. Each stage (STT, Summarization, and Key Points Extraction) acts as an independent filter, ensuring modularity, scalability, and efficient task execution.

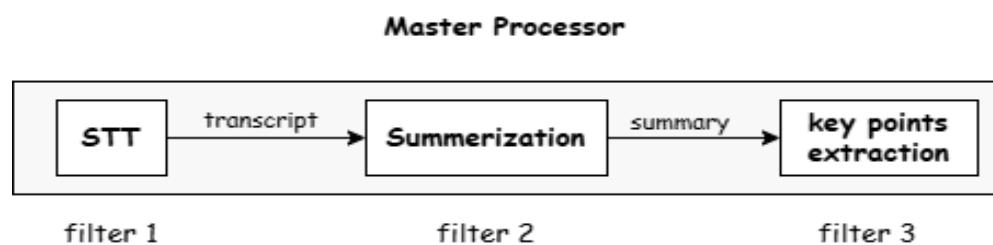


Figure 80 service architecture

- Good software design: strategy design pattern

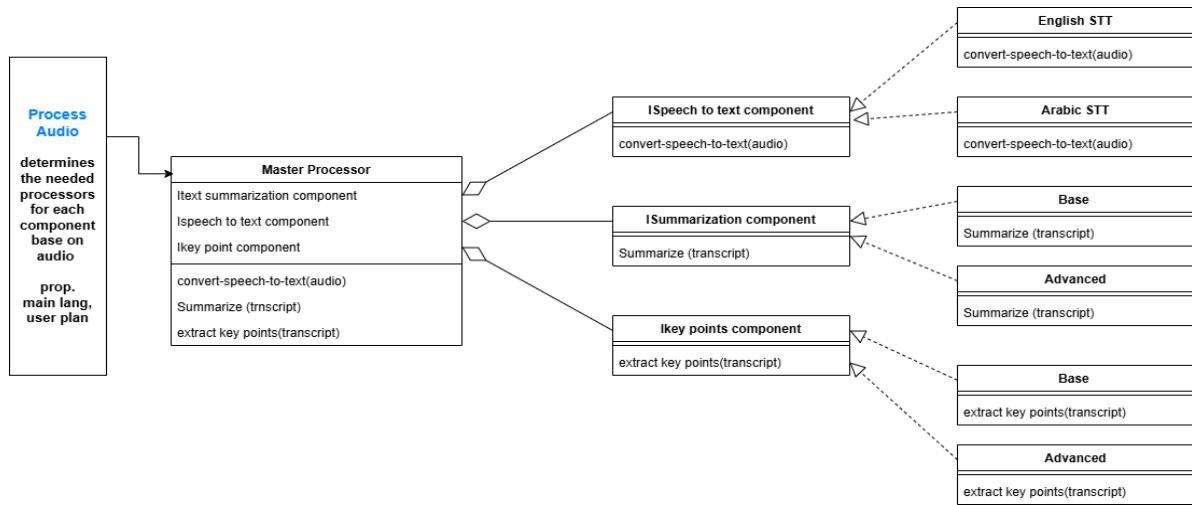


Figure 81 Strategy design pattern solution

In designing the solution, we follow the **Strategy Design Pattern**, enabling dynamic selection of processing components based on factors like language and user plan. Each processing stage Speech-to-Text, Summarization, and Key Points Extraction is implemented as an independent strategy, ensuring modularity and scalability.

This approach allows seamless integration of new models without modifying core logic, enhances code maintainability, and optimizes resource allocation. By separating concerns and promoting reusability.

3.3 Used tools and models:

Based on the findings from our literature review, we carefully selected high-performance, resource-efficient models, well-integrated software solutions, and the most suitable frameworks to develop this service. This ensures a balance between optimal performance and cost-effectiveness while maintaining reliability and scalability.

- **Django – python:**

is a high-level Python web framework that encourages rapid development and clean, pragmatic design. especially we use DRF Django Rest Framework, which is a widely used, full-featured API framework designed for building RESTful APIs with Django.

- **MongoDB**

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents, making it ideal for handling unstructured data and large volumes of information. It offers high scalability and performance.

We use it to store the final report generated for each audio.

- **Insomnia**

Insomnia is an open-source desktop application that takes the pain out of interacting with and designing, debugging, and testing APIs, we use it to test our APIs.

- **Whisper AI:**

We utilize the Whisper AI API from OpenAI to handle Arabic speech-to-text conversion due to its exceptional performance, high accuracy, and efficient resource usage. The API allows us to seamlessly use any model from the Whisper family without the need for local deployment.

- **Nova-02:**

For English speech-to-text, we leverage the Nova-02 model from the Deepgram API, known for its accuracy and efficiency in processing English audio.

- **Gpt-4o min:**

We use the GPT-4o Mini model for text summarization and key point extraction via an API. The process involves sending the audio transcript or the summary along with a specific prompt to the model and receiving a summarized response.

The prompt used – summarization is:

*“I will provide you a meeting transcript, please summarize it in {max_words} words, in {language} language only.
The transcript is:”*

The *{max_words}* value is calculated based on the number of words in the transcript, and *{language}* specifies the desired response language, which matches the transcript language.

- **RabbitMQ:**

We use RabbitMQ as the messaging broker in our AI service to ensure efficient communication and task management across the service. It plays a crucial role in handling the queuing and distribution of tasks - audio processing requests.

3.4 APIs documentation:

API documentation is crucial for enabling seamless integration, clear communication, and efficient utilization of the service by developers.

- **Authentication:**

First, we have a common authorization way – service-to-service authentication using API key, the API key ensures that only authorized external services can access the AI processing service, preventing unauthorized usage, by Including the API key in the request headers.



Figure 82 authentication solution

- AI Service communication with the rest of the system:

The AI Service provides secure endpoints for communication. The base management service calls these endpoints when a meeting audio has arrived and is ready to process.

API Endpoints Documentation

Submit audio via file:

Endpoint:	/ai_processor/submit_audio_file/
HTTP Method:	Post
Description:	Submits an audio file for processing and returns a unique audio_token for tracking.
Parameters:	<ul style="list-style-type: none"> - main_language (string, required): The primary language of the audio (ar for Arabic, en for English). - audio_file (file, required): The audio file to process.
Response:	<ul style="list-style-type: none"> - message: Success message. - audio_token: Unique identifier for the task.

Submit audio via URL:

Endpoint:	/ai_processor/submit_audio/
HTTP Method:	Post
Description:	Submits an audio URL for processing and returns a unique audio_token for tracking.
Parameters:	<ul style="list-style-type: none"> - main_language (string, required, in the body): The primary language of the audio (ar for Arabic, en for English). - audio_url (string, required, in the body): The URL of the audio file for processing.
Response:	<ul style="list-style-type: none"> - message: Success message. - audio_token: Unique identifier for the task.

Tracing the audio status:

Endpoint:	/ai_processor/get_status/
HTTP Method:	Get
Description:	Retrieves the processing status of a submitted audio file.
Parameters:	<ul style="list-style-type: none"> - audio_token (string, required, request parameter): The unique identifier token for tracking the status of the audio processing task.
Response:	<ul style="list-style-type: none"> - done: Whether the processing is complete (true or false). - status: Current status (on_queue, stt_processed, summary_processed, key_points_processed, completed, failed).

Get the result of the processing whenever the done = true:

Endpoint:	/ai_processor/get_report/
HTTP Method:	Get
Description:	Fetches the final report for a processed audio file, including the transcript.
Parameters:	- audio_token (string, required, request parameter): The unique identifier token.
Response:	- audio_id: Unique identifier for the task. - transcript: Full transcription of the audio. - summary: A concise summary of the transcript. - key_points: Array of the extracted key points.

4. AI Service Testing:

This section evaluates the AI service's core functionalities, focusing on their performance, accuracy, and overall effectiveness.

4.1 English Language Case:

Test scenario: We need to upload a meeting audio file in English and get the report generated after processing the audio. The process should include converting the audio into text, summarizing this text, and extracting the key points.

Test data: We will use the online GitLab Engineering Key Review Meeting ([meeting link](#)). The meeting duration is 24 minutes, conducted in native English, with 11 participants.

Test expected result: The system should generate the final report after completing the processing flow. The report must include the following:

1. Audio Transcript: A detailed and accurate transcription of the meeting audio.
 2. Summary: A concise overview of the meeting's discussions and key topics.
 3. Key Points: A list of actionable and relevant points derived from the meeting content.
- Performance results:

Table 27 AI service test1 performance results

Meeting's audio token	5516c7c5-28c9-4a54-92a8-53a131a26b42
Total processing duration	40 sec
Speech-to-text response duration (Nova-02)	20 sec
Summarization model response (Gpt-4o mini)	10 sec
Key points extraction model response (GPT-4o mini)	10 sec

- Test actual results:

Table 28 AI service test1 actual results

Transcript	3,843 Word with high accuracy
Summary	<p><i>"The meeting, led by Eric Johnson on February 18, 2021, focused on proposals for restructuring engineering key reviews at GitLab. Eric proposed breaking the meeting into four departmental reviews: development, quality, security, and UX. This change aims to enhance visibility, allow deeper discussions, and provide more time for strategic focus. A two-month rotation was suggested to minimize the number of new meetings, with development and quality meeting one month, followed by security and UX the next. n nThe discussion progressed to the research and development (R&D) metrics, particularly the "wider MR rate, " which includes community contributions. Concerns were raised about the current metrics potentially being confusing and not fully capturing contributions from both internal and external sources. A proposal was made to focus on the percentage of total MRs from the community over time, which was well-received. n nFurther topics included addressing replication issues within the infrastructure, with plans to separate database workloads to improve performance. The team discussed ongoing improvements in defect tracking, emphasizing the importance of measuring open bugs instead of just closed ones to get a clearer picture of the backlog. n nFinally, the meeting addressed the narrow MR rate, which had fallen below</i></p>

target in January due to increased vacation days, but there was optimism for recovery in February. The overall sentiment was one of collaboration and a focus on refining metrics to better reflect the team's performance and contributions while ensuring community engagement is maintained as the company grows."

Key points

- Meeting led by Eric Johnson on February 18, 2021
- Focus on proposals for restructuring engineering key reviews at GitLab
 - Proposal to break meeting into four departmental reviews: development, quality, security, and UX
 - Aim to enhance visibility, allow deeper discussions, and provide more time for strategic focus
 - Suggested two-month rotation to minimize new meetings: development and quality one month, security and UX the next
 - Discussion on R&D metrics, particularly | "wider MR rate| " including community contributions
 - Concerns raised about current metrics being confusing and not fully capturing contributions
 - Proposal to focus on percentage of total MRs from the community over time, which was well-received
 - Addressing replication issues within infrastructure, plans to separate database workloads to improve performance
 - Discussion on ongoing improvements in defect tracking, emphasizing measurement of open bugs for clearer backlog picture
 - Narrow MR rate fell below target in January due to increased vacation days, optimism for recovery in February
 - Overall sentiment of collaboration and focus on refining metrics to better reflect performance and contributions

-Ensuring community engagement is maintained as the company grows.

The resulting report of the meeting's audio is stored in the MongoDB database:

```
_id: ObjectId('67a7da1459bd10560736e4ef')
audio_token : Binary.createFromBase64('VRbHxSjJStSSqFOhMaJrQg==', 3)
audio_url : null
transcript : "hi this is eric johnson it's february eighteenth twenty twenty one and..."
summarization : "The meeting, led by Eric Johnson on February 18, 2021, focused on prop..."
processing_status : "COMPLETED"
key_points : Array (13)
  0: "- Meeting led by Eric Johnson on February 18, 2021"
  1: "- Focus on proposals for restructuring engineering key reviews at GitL..."
  2: "- Proposal to break meeting into four departmental reviews: developmen..."
  3: "- Aim to enhance visibility, allow deeper discussions, and provide mor..."
  4: "- Suggested two-month rotation to minimize new meetings: development a..."
  5: "- Discussion on R&D metrics, particularly "wider MR rate" including co..."
  6: "- Concerns raised about current metrics being confusing and not fully ..."
  7: "- Proposal to focus on percentage of total MRs from the community over..."
  8: "- Addressing replication issues within infrastructure, plans to separa..."
  9: "- Discussion on ongoing improvements in defect tracking, emphasizing m..."
  10: "- Narrow MR rate fell below target in January due to increased vacatio..."
  11: "- Overall sentiment of collaboration and focus on refining metrics to ..."
  12: "- Ensuring community engagement is maintained as the company grows."
main_language : "en"
created_at : 2025-02-08T22:26:28.443+00:00
updated_at : 2025-02-08T22:26:58.581+00:00
user_plan : "premium"
```

Figure 83 MongoDB stored report

test result: passed.

4.2 Arabic Language Case:

Test scenario: We need to upload a meeting audio file in Arabic and get the report generated after processing the audio. The process should include converting the audio into text, summarizing this text, and extracting the key points.

Test data: We will use a simulate first meeting over school management system new project – 11 min.

Test expected result: The system should generate the final report after completing the processing flow. The report must include the following:

Audio Transcript, Summary, Key Points.

- Performance results:

Table 29 AI service test 02 performance results

Meeting's audio token	e323daa7-6e56-45f4-a630-ca1946248847
Total processing duration	1:10 min
Speech-to-text response duration (whisper)	1 min
Summarization model response (Gpt-4o mini)	5 sec
Key points extraction model response (GPT-4o mini)	5 sec

- Test actual results:

Table 30 AI service test 02 actual results

Transcript	1073 Word with high accuracy
Summary	"تمت مناقشة نظام تعليمي جديد يهدف لتسهيل العملية التعليمية للطلاب والمعلمين وإدارة المدرسة. تم التركيز على احتياجات جميع المستخدمين، حيث تقدم النظام خدمات مثل متابعة الجدول الدراسي للطلاب، وإدارة الحصص للمعلمين، وإعداد التقارير للإدارة. كما تم التطرق لموضوع الميزانية وضرورة جمع المتطلبات من جميع الأطراف المعنية قبل بدء التنفيذ. تم اقتراح خطة شاملة تشمل مراحل جمع المتطلبات، التصميم، والتنفيذ مع إمكانية التعديلات السريعة بناءً على تغذية راجعة. الاجتماع اختتم بالاتفاق على تحديد موعد لاحق لمناقشة الميزانية وتفاصيل إضافية".
Key points	<ul style="list-style-type: none"> -"مناقشة نظام تعليمي جديد يهدف لتسهيل العملية التعليمية للطلاب والمعلمين وإدارة المدرسة،" -"التركيز على احتياجات جميع المستخدمين،" تقديم خدمات مثل متابعة الجدول الدراسي للطلاب وإدارة الحصص للمعلمين وإعداد التقارير للإدارة،" -"التطرق لموضوع الميزانية وضرورة جمع المتطلبات من جميع الأطراف المعنية قبل بدء التنفيذ،" -"اقتراح خطة شاملة تشمل مراحل جمع المتطلبات، التصميم، والتنفيذ مع إمكانية التعديلات السريعة بناءً على تغذية راجعة،" -"الاتفاق على تحديد موعد لاحق لمناقشة الميزانية وتفاصيل إضافية"

The resulting report of the meeting's audio is stored in the MongoDB database:

```

_id: ObjectId('67a7eabc29d644a4ddf2bd3e')
audio_token : Binary.createFromBase64('4yPap25WRfSmMMoZRiSIRw==', 3)
audio_url : null
transcript : "مرحباً، شكراً على ترتيب هذا الاجتماع. أود معرفة المزيد عن رؤياتكم لهذا"
"ستمت مناقشة نظام تعليمي جديد يهدف لتسهيل العملية التعليمية للطلاب والمع"
summarization : "الى"
processing_status : "COMPLETED"
key_points : Array (6)
  0: "...مناقشة نظام تعليمي جديد يهدف لتسهيل العملية التعليمية للطلاب والمعلم"
    1: "... التركيز على احتياجات جميع المستخدمين"
    2: "...تقديم خدمات مثل متابعة الجدول الدراسي للطلاب وإدارة الحصص للمعلمين و"
    3: "...التطرق لموضوع الميزانية وضرورة جمع المتطلبات من جميع الأطراف المعنية"
    4: "...اقتراح خطة شاملة تشمل مراحل جمع المتطلبات، التصميم، والتنفيذ مع إمكا"
    5: "...الاتفاق على تحديد موعد لاحق لمناقشة الميزانية وتفاصيل إضافية"
main_language : "ar"
created_at : 2025-02-08T23:37:32.889+00:00
updated_at : 2025-02-08T23:38:30.904+00:00
user_plan : "premium"

```

Figure 84 MongoDB stored report

test result: passed.

Chapter 7 Deployment

1. Introduction

In this chapter, we explore the deployment architecture of **AgileMeets** system, also we will mention the cloud providers that were chose for deployment, and finally we will talk about some measured quality attributes.

2. Deployment Architecture

2.1 Docker Overview

Docker is a critical enabling technology for the deployment of the **AgileMeets** system, serving as the foundation for containerizing and orchestrating all microservices, databases, and supporting components. As an open-source platform, **Docker** packages applications and their dependencies into lightweight, portable containers, ensuring consistency across development, testing, and production environments. This containerization approach eliminates "it works on my machine" issues by providing isolated, reproducible runtime environments.

In the **AgileMeets** system, each microservice—such as the Base Management (**BaseMgt**), Notification, Meeting, and AI Services—along with supporting tools like **RabbitMQ**, **Postgres**, **MongoDB**, and **OpenVidu**, is defined by a Dockerfile. These files specify the base images, dependencies, and configurations needed to build containers, ensuring that each service runs identically regardless of the underlying infrastructure. For instance, **BaseMgtService.Dockerfile** uses .NET 8.0, while **MeetingService.Dockerfile** leverages **Node.js 20**, demonstrating Docker's flexibility across different technologies.

Docker Compose further enhances the deployment process by orchestrating multiple containers, as seen in services like **OpenVidu** and the **AI Service**, where **docker-compose.yml** manages interconnected containers (e.g., AI Service, RabbitMQ, and MongoDB). This modularity supports scalability, as containers

can be easily replicated or scaled across **Fly.io VMs** and **AWS EC2** instances, while maintaining isolation and resource efficiency.

By leveraging **Docker**, **AgileMeets** achieves a **robust, scalable, and portable** deployment architecture. **Containers** enable **rapid deployment**, consistent performance, and simplified management, making **Docker** an integral part of the system's infrastructure and a key factor in its success.

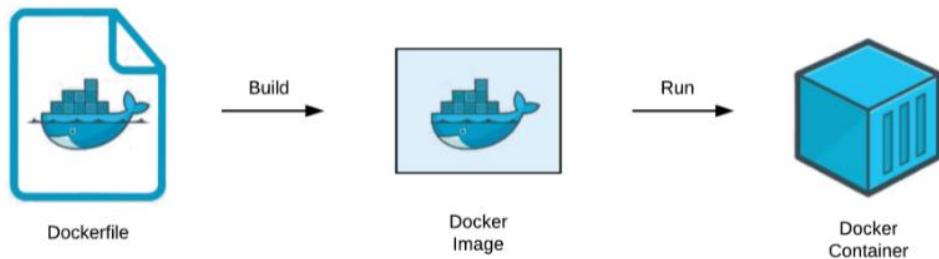


Figure 85 Dockerfile to Container

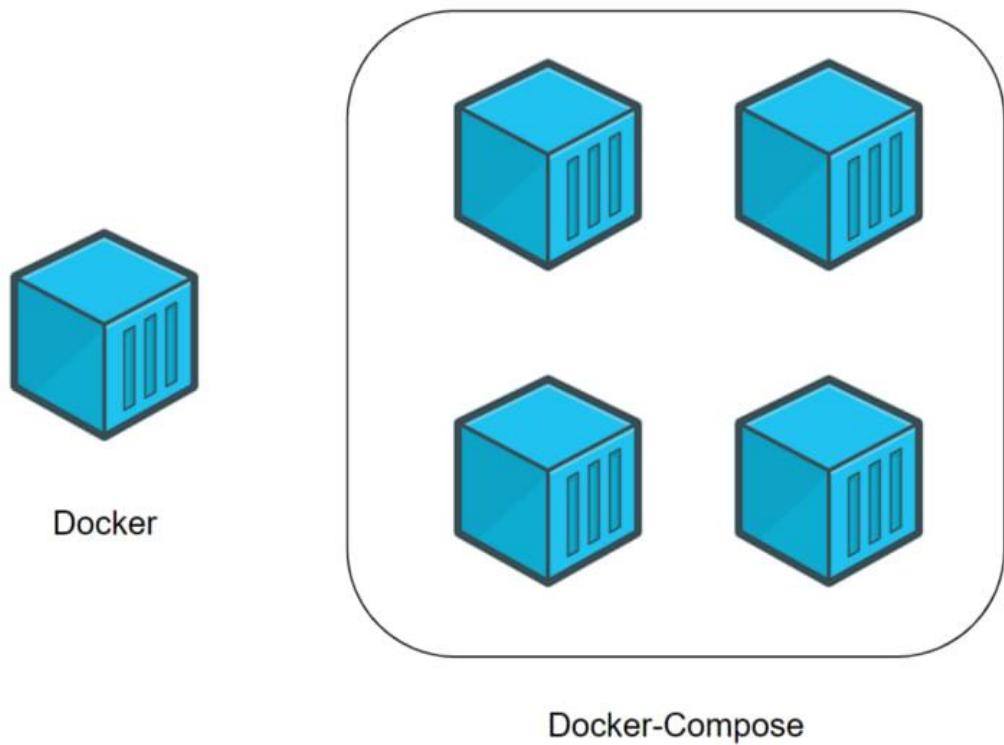


Figure 86 Docker vs Docker-Compose

2.2 Deployment Diagram

As shown in the Deployment Diagram below, the system was deployed on cloud utilizing docker containers all the way, since AgileMeets is on Development, so we aimed to reduce costs and deploy on an easy to configure/update providers with the lowest costs.

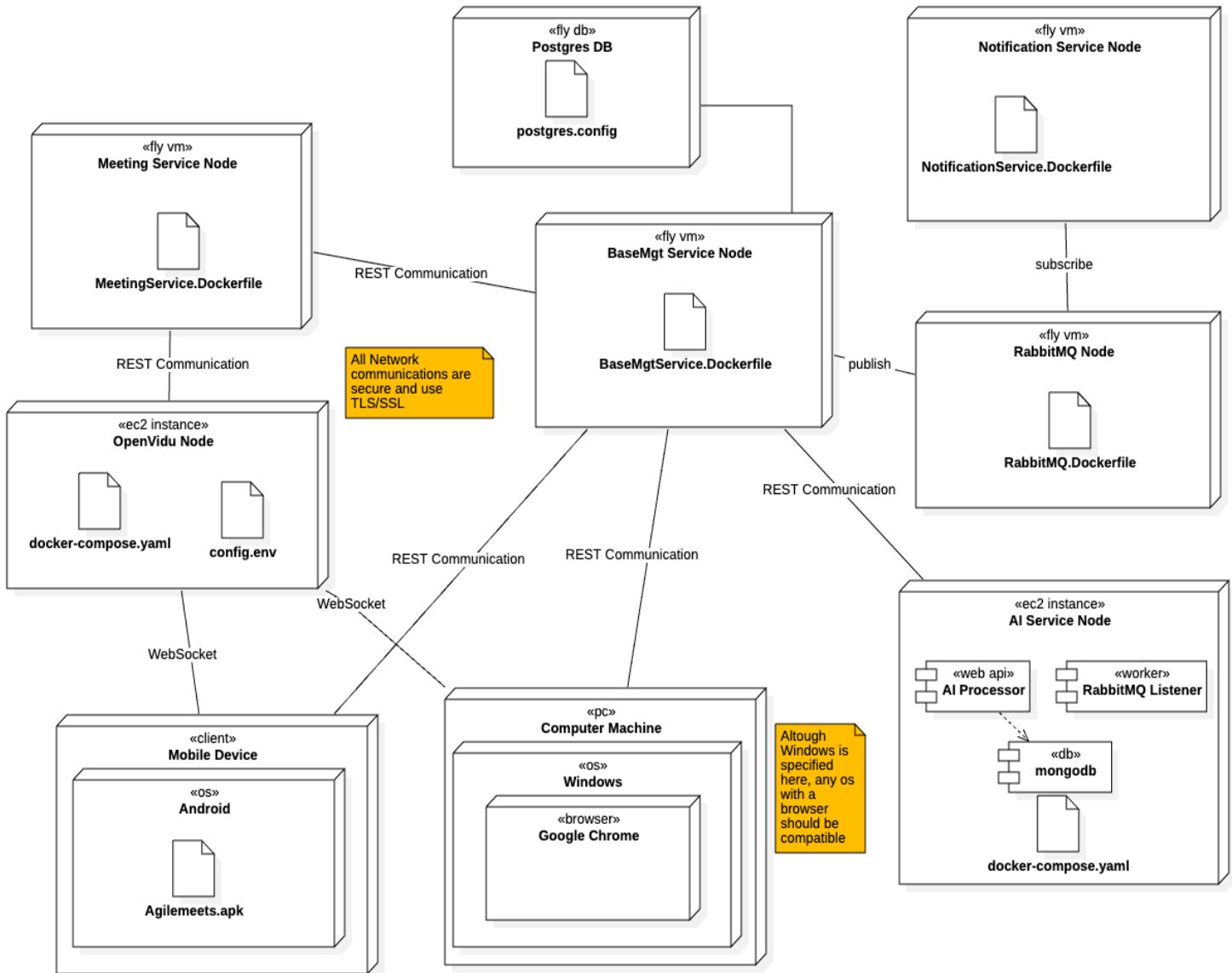


Figure 87 Deployment Diagram

3. Cloud Providers and Services Used

The AgileMeets system leverages two primary cloud providers—Fly.io and Amazon Web Services (AWS)—to ensure scalability, reliability, and cost-efficiency in its deployment. These platforms host the microservices, databases, and supporting infrastructure, each offering unique features tailored to the system's requirements.

3.1 Fly.io

Fly.io is a global application hosting platform selected for its simplicity, cost-effectiveness, and ability to deploy containerized applications worldwide. It hosts key components of AgileMeets, including:

- **Base Management (BaseMgt) Service:** Manages core functionalities like user authentication and project coordination.
- **Notification Service:** Handles asynchronous notifications and integrates with RabbitMQ for event-driven communication.
- **Meeting Service:** Facilitates meeting scheduling and real-time communication, integrating with OpenVidu and AWS S3.
- **RabbitMQ:** Provides message queuing for asynchronous interactions across services.
- **PostgreSQL:** Serves as the relational database for structured data storage, such as user profiles and project details.

Fly.io deploys these services as virtual machines (VMs) using Docker containers, defined in Dockerfile and fly.toml configurations. Its global distribution ensures low-latency access for users, while features like auto-scaling and auto-shutdown optimize resource usage and reduce costs significantly.

3.2 Amazon Web Services (AWS)

AWS provides robust and scalable infrastructure for the more resource-intensive components of AgileMeets. The following AWS services are utilized:

- **EC2 (Elastic Compute Cloud):**

- Hosts the **AI Service**, which handles AI-driven analytics and processing, along with its supporting databases (MongoDB) and RabbitMQ.
- Also hosts the **OpenVidu Single Node**, enabling real-time video conferencing via WebSocket.
- EC2 instances offer high performance, scalability, and flexibility, with customizable virtual servers tailored to the system's workload.

- **CloudFormation:**

- Used to automate the deployment and management of the OpenVidu Single Node on EC2.
- Provides Infrastructure as Code (IaC) templates, ensuring consistent and repeatable deployments with minimal manual intervention.

- **S3 (Simple Storage Service):**

- Employed by the Meeting Service & BaseMgt Service to store large files, such as meeting recordings, ensuring durable and scalable storage.

AWS's extensive ecosystem, including EC2's compute capabilities and CloudFormation's automation, supports the system's scalability and resilience, particularly for real-time and data-intensive operations.

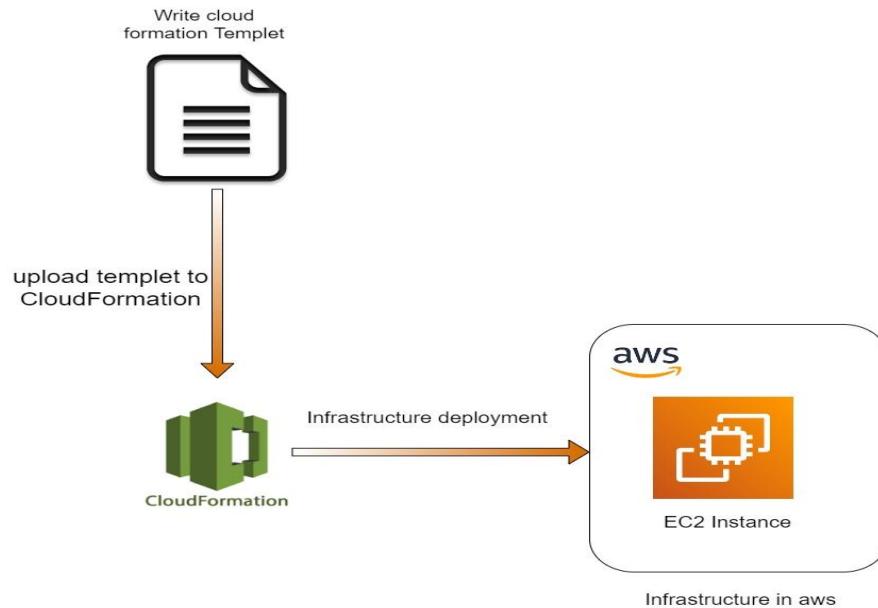


Figure 88 AWS CloudFormation (IaC)

4. Performance Metrics and Load Testing

In this section, we examine the performance metrics and load testing results for the AgileMeets system, highlighting its scalability and ability to meet business needs through the autoscaling capabilities of Fly.io and AWS EC2. By utilizing [Loader.io](#), a cloud-based load testing tool, we ensure a fair and reliable evaluation, minimizing interference from local environments and providing consistent results across different scaling configurations.

4.1 Overview

Our load testing targets the AgileMeets homepage endpoint, a key entry point integrating multiple data needs and personalized processing for users. This endpoint serves as a reliable indicator of the system's maximum concurrent user capacity, allowing us to evaluate how Fly.io and AWS EC2 autoscaling supports scalability and responsiveness under varying loads.

4.2 Case 01 (Without Scaling)

- Instances: 1 | DB: only 1 no replicas.

1. 50 Clients/s => 3000 Requests/m | Success rate = 3000/3000 => 100%

Average latency: 99ms (Accepted)

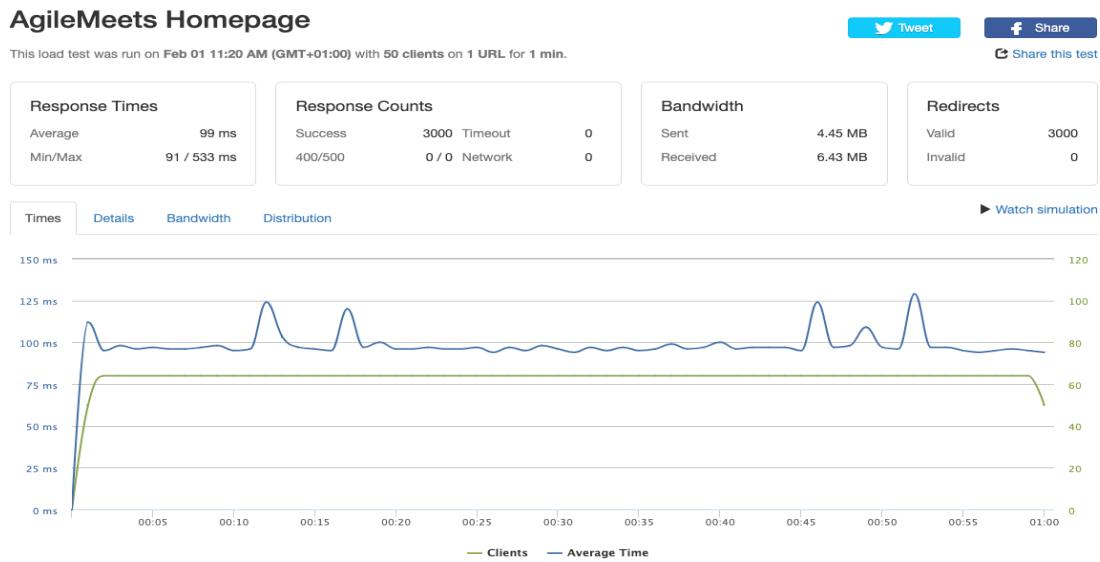


Figure 89 Case 01 - 50 Clients load-test

2. 50 Clients/s => 6000 Requests/m | Success rate = 6000/6000 => 100%

Average latency: 99ms (Accepted)

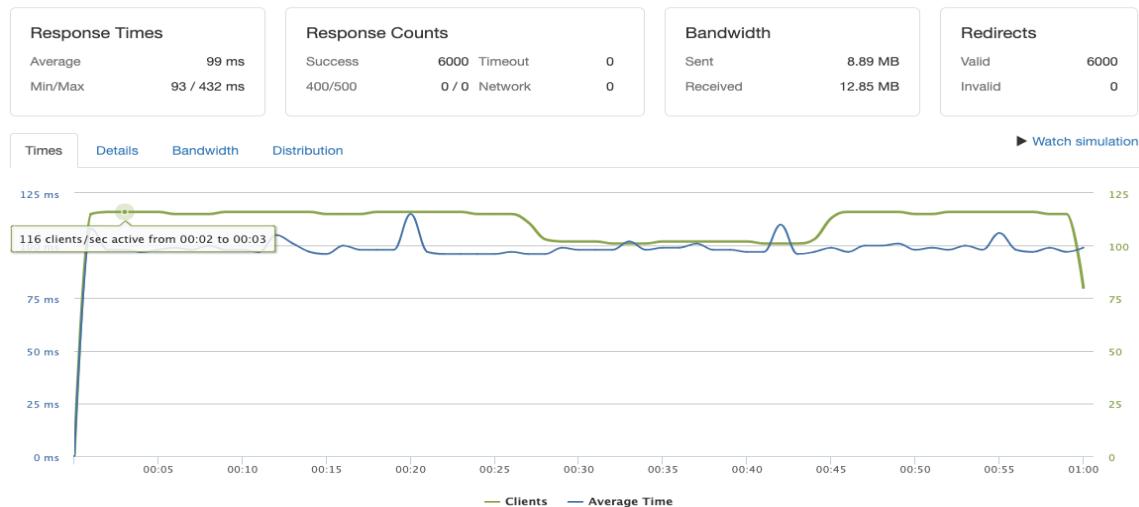


Figure 90 Case 01 - 100 Clients load-test

3. 200 Clients/s => 12000 Requests/m | Success rate: Early failure.

Average latency: 1257ms (Rejected)

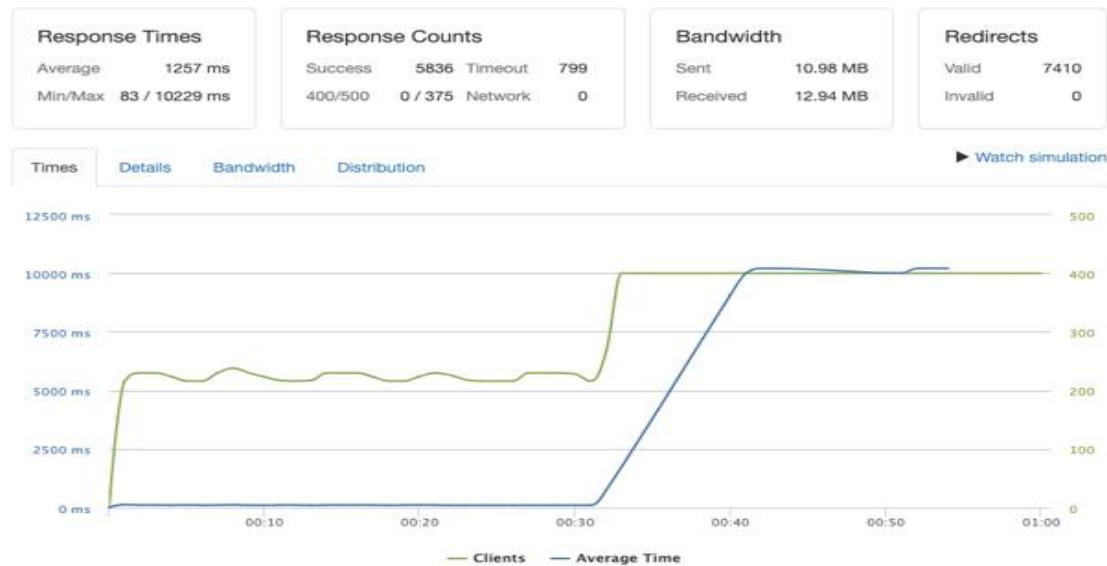


Figure 91 Case 01 - 100 Clients load-test

4.3 Case 02 (Both horizontal and vertical scaling)

- Instances: 4 with increased power | DB: 2 instance & 1 replicas.

1. 200 Clients/s => 12000 Requests/m | Success rate = 12000/12000 => 100%

Average latency: 107 (Accepted)

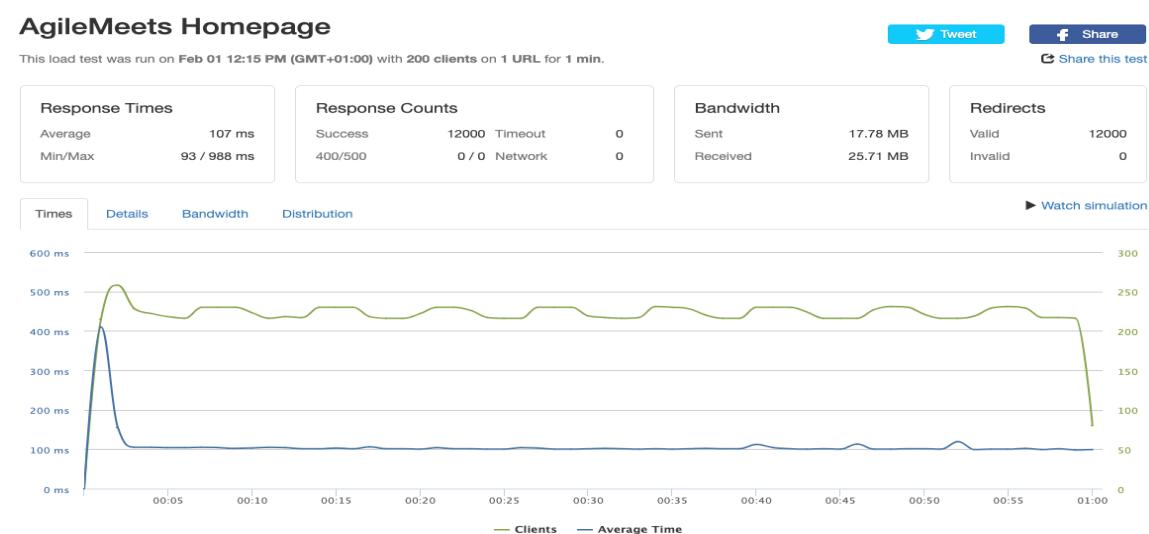


Figure 92 Case 02 - 200 Clients load-test

2. 400 Clients/s => 24000 Requests/m | Success rate = 24000/24000 => 100%

Average latency: 104ms (Accepted)

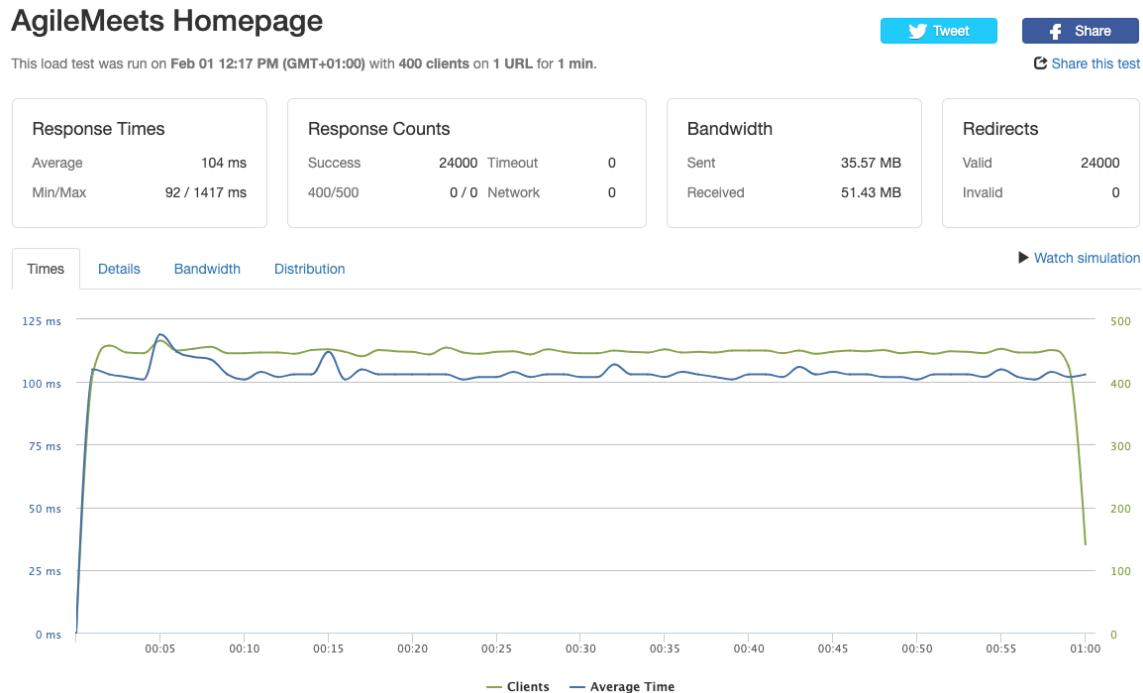


Figure 93 Case 02 - 400 Clients load-test

5. Future Considerations

As the AgileMeets system continues to evolve, several strategic enhancements can be considered to improve scalability, cost-efficiency, and reliability. These future considerations build on the current deployment architecture while addressing long-term operational goals.

5.1 Transition to a Single Cloud Provider with AWS EKS

One potential evolution is consolidating the deployment onto a single cloud provider by migrating all microservices and supporting infrastructure to **Amazon Web Services (AWS) Elastic Kubernetes Service (EKS)**. **AWS EKS**, a managed Kubernetes service, offers a unified platform for orchestrating containerized applications, providing advanced auto-scaling capabilities through **Kubernetes**.

auto-scaling groups. This transition would simplify management, enhance scalability, and ensure consistent performance across services like **BaseMgt**, **Notification**, **Meeting**, **AI**, and **OpenVidu**. Additionally, implementing cost limits within **AWS EKS**—using tools like **AWS Budgets** and **Kubernetes** resource quotas—would help control operational expenses, ensuring cost-efficiency while maintaining high availability and performance.

5.2 Migration to Owned Infrastructure

Another long-term consideration is studying a scale plan for migrating **AgileMeets** to **owned infrastructure**, such as on-premises servers or private data centers. This approach would involve assessing the feasibility, cost, and technical requirements of transitioning from cloud-based providers (**Fly.io** and **AWS**) to a self-managed environment. Such a migration could yield significant savings over time by eliminating recurring cloud subscription costs, particularly for high-traffic or long-term deployments. Moreover, owned infrastructure offers reusability for other projects or systems, enhancing resource flexibility and reducing dependency on third-party providers. A detailed cost-benefit analysis, including initial investment, maintenance, and scalability, would be necessary to validate this strategy.

5.3 Enhanced Testing for Performance and Non-Functional Requirements

To further strengthen the system, we propose conducting more comprehensive performance and non-functional testing. These tests would focus on identifying potential weak points in the deployment, such as latency under peak loads, system resilience during failures, and security vulnerabilities. By leveraging tools like **Loader.io** for additional load testing, as well as stress testing and chaos engineering practices, we can evaluate the system's behavior under extreme conditions. Non-functional tests, including reliability, availability, and

maintainability assessments, would ensure that **AgileMeets** meets business and user expectations, paving the way for continuous improvement and robust scaling.

These future considerations aim to optimize the **AgileMeets** deployment for scalability, cost-efficiency, and reliability, ensuring its readiness for long-term growth and evolving business needs.

Chapter 8 Conclusion

1. Introduction:

As a result of our project, we have developed a software system designed to support software companies that follow the Agile methodology. This system streamlines the management of frequent and recurring meetings, ensuring that critical meeting artifacts are efficiently captured, organized, and utilized to enhance planning and decision-making for subsequent project phases.

In this chapter we will conclude it all and put it all together, showing the final architecture, lessons learnt and the future possible enhancements.

2. Putting it All Together

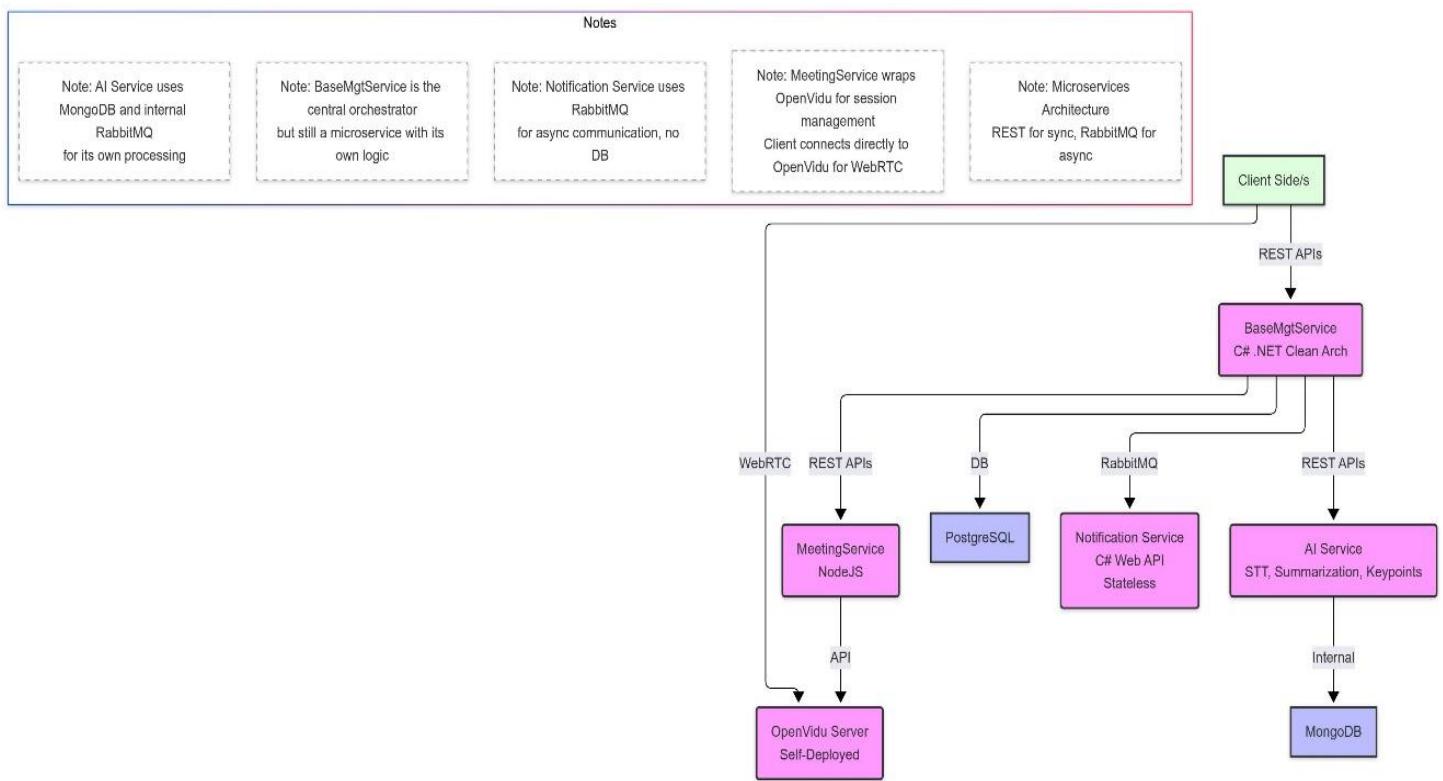


Figure 8.1 system final architecture

3. Lessons Learnt

Throughout the development and deployment of **AgileMeets**, several valuable lessons emerged, shaping our understanding of microservices and cloud-based systems:

- **Importance of Containerization:** Docker proved indispensable for maintaining consistency across environments, but managing Dockerfile configurations and Docker Compose orchestration required careful planning to avoid compatibility issues or resource conflicts.
- **Autoscaling Trade-offs:** Fly.io's auto-scaling and auto-shutdown features significantly reduced costs demanded precise load metrics and monitoring to balance performance and expense effectively.
- **Inter-Service Communication:** Implementing REST, WebSocket, and RabbitMQ required robust error handling and retry mechanisms to ensure resilience, particularly under high loads or network disruptions.

4. Future Recommendations

- Integrating or developing an Issue Management Service to create a comprehensive, all-in-one solution for projects and organizations.
- Implementing smart requirements extraction from meeting transcripts and key points to streamline project planning and documentation.
- Introducing a rich search experience, enabling semantic searches and requirement-based queries linked to meeting transcripts for improved accessibility.
- Enhancing the Meeting and AI Services to capture speaker identities within voice transcripts, distinguishing talkers for greater accuracy and context.

Appendices

- [GitHub repository](#)

References

- Deepgram. (2025, 02 23). *Deepgram*. Retrieved from <https://deepgram.com/learn/nova-2-speech-to-text-api>
- Django. (2025, 2 23). *djang Docs*. Retrieved from <https://docs.djangoproject.com/en/5.1/>
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., . . . Zettlemoyer, L. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv*.
- Microsoft. (2025, 02 23). *Microsoft Learn*. Retrieved from <https://learn.microsoft.com/en-us/dotnet/>
- OpenAI. (2025, 2 23). *openAI Docs*. Retrieved from <https://platform.openai.com/docs/api-reference/introduction>
- OpenVidu. (2025, 02 24). *OpenVidu Docs*. Retrieved from <https://openvidu.io/latest/docs>
- rabbitMQ. (2025, 02 23). *RabbitMQ Docs*. Retrieved from <https://www.rabbitmq.com/docs>