Group 7

Virtual Chat 3D
Software Development Plan
Version 1.4

Revision History

Date	Version	Description	Author
29/10/2024	1.0	Initial version of project plan.	Hoằng, Đạt
30/10/2024	1.1	Updated schedule, plan, added risk management.	Hoằng, Đạt
1/11/2024	1.2	Added schedule diagram, timeline, project organization.	Hoằng, Đạt, Quang
6/11/2024	1.3	Added sprint 0 to plan, format update	Đạt
7/11/2024	1.4	Re-format document, added revision history	Hoằng, Đạt

Table of Contents

Software Development Plan

1. Introduction

This document outlines the strategy which our team employs to implement the "Virtual Chat 3D" as stated in the proposal. The contents of this document provide an overview of our project, and details our team organization. It also explains the planning process, risk management and team monitoring with estimations. And most importantly, it gives an outline of plan phases that the team must follow to ensure the project development progress.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The project aims to create safe, secure and friendly 3D interactive chat rooms, together with an integrated 3D chatbot. The main purpose of this is to provide a way for users to remotely engage and communicate with each other in an immersive fashion.

The platform should deliver an intuitive 3D interface where users can easily navigate and interact with objects. Designs should also be customizable for each user. Meaningful activities within the chat rooms should be provided to promote the strengths and benefits of virtual 3D space.

Furthermore, chats and actions should be displayed in real-time smoothly for real-time activities. AI chatbot should be implemented to naturally respond to users requests with emotion expression and relevant wordings. Finally, the implementation process should follow security standards to ensure privacy and security.

2.2 Assumptions and Constraints

2.2.1 Assumptions
- **Copyright and legal use**: assets and functionalities used in this project should be copyright-free and our team should not violate any copyright or free to use tool terms of services.
- **Responsible users**: the development team might assume that users use our project responsibly in real-life situations, any function of our application which can be potentially used for illegal activities or non-technical exploitation of the project will be ignored.
- **Teammates ability:** team members are expected to be able to complete deadlines or finish parts of the assigned tasks. Team members should be supportive and responsible for the project.
- **Small project capability:** this project is outlined to be a small project, for that reason, some of the scalability might be ignored.

2.2.2 Constraints
- **9 weeks schedule**: the project should be completed in this timeframe to keep up with the semester's progress.
- **Budget**: due to limited budget, costly APIs and tools should be avoided.
- **Team members' personal schedules**: each member of the team has a specific schedule with different deadlines, free time and work hours. The planning phase should ideally provide a healthy and effective schedule for the team.
- **Product quality and maintenance capability:** the final product should be acceptable to deploy and operate in real life as a functional application.
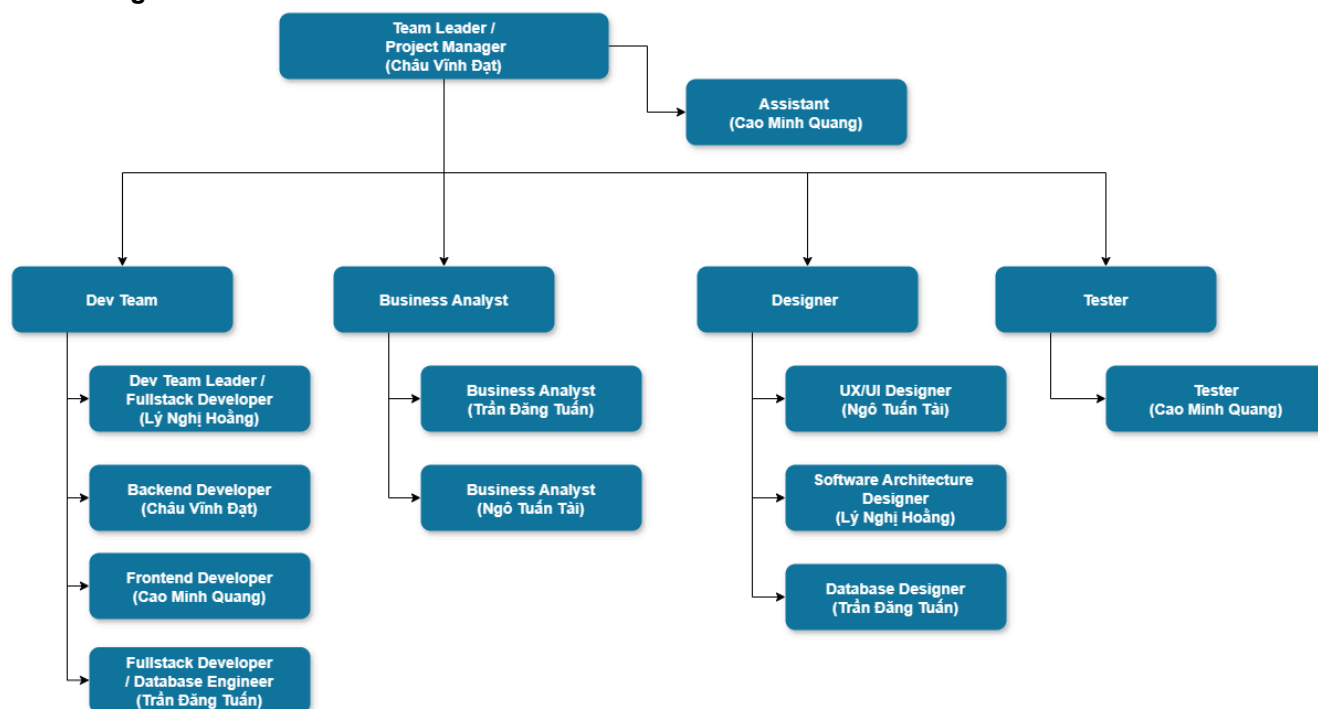
2.3 Project Deliverables

Deliverables	Delivery Date
Proposal	24/10/2024 - Completed
Software Development Plan	2/11/2024
Vision Document	2/11/2024
Use-case Specification Document	12/11/2024

Software Architecture Document	19/11/2024
Class Diagram	19/11/2024
UI Prototype	26/11/2024
Test Plan	3/12/2024
Test Cases Document	7/12/2024
Test Report Document	9/12/2024
Demo version Product	12/12/2024
Alpha version Product	19/12/2024
Beta version Product	23/12/2024
Release version Product	26/12/2024

3. Project Organization

3.1 Organizational Structure

3.2 Roles and Responsibilities:

Member	Role	Responsibilities
Châu Vĩnh Đạt	Project Manager, Backend Developer	Responsible for tracking the process, managing timelines. resources, coordination among team members. Creating the backend server for sending messages.
Cao Minh Quang	Assistant, Developer, Tester	Responsible for writing reports, help with small modules in the project function and creating test plan, test case to ensure the quality of the project
Lý Nghị Hoằng	Development Leader, Software Architecture Designer	Responsible for leading the development process, main developer of the project, designing project architecture, creating the main functions, modules and helping other members with their work. Control the quality and deploy the application to the Android environment.
Trần Đăng Tuấn	Developer, Database Engineer, Database Designer	Responsible for designing, creating and managing the database. Implement CRUD of the database. Help in development with small modules.
Ngô Tuấn Tài	Business Analyst, UI/UX designer	Gather and analyze business requirements, make sure the application meets the requirements. Design UI/UX.

4. Management Process

4.1 Project Estimates

Project duration:
- This project starts from 24/10/2024 to 26/12/2024. Duration is 9 weeks.

Estimated project cost

Tool and software cost:
- **Unity:** $2040/year if over $200000 of revenue or funds raised in the prior 12 months.
- **Firebase:**
    + **Cloud Storage**

GB stored	$0.026/GB
GB downloaded	$0.12/GB
Upload operations	$0.05/10k
Download operations	$0.004/10k

    + **Authorization**

Monthly Active Users (MAU)	Price per MAU ($)
0 - 49,999	0
50,000 - 99,999	0.0055
100,000 - 999,999	0.0046
1,000,000 - 9,999,999	0.0032
10,000,000 +	0.0025

    + **Realtime Database**

GB stored	$5/GB

GB downloaded	$1/GB

- **Gemini 1.5 flash:**

Prompts up to 128k tokens	Input Pricing: $0.075 / 1 million tokens Output Pricing: $0.30 / 1 million tokens Context Caching: $0.01875 / 1 million tokens
Prompts longer than 128k	Input Pricing: $0.15 / 1 million tokens Output Pricing: $0.60 / 1 million tokens Context Caching: $0.0375 / 1 million tokens
Context caching (storage)	$1.00 / 1 million tokens per hour

4.2   Project Plan

4.2.1   Phase and Iteration Plan

Our project will be planned according to RUP and Scrum. We separate this project into 3 phases: Inception, Elaboration and Construction and 4 sprints (excluding sprint 0)
In Inception, there is 1 sprint. (excluding sprint 0) (2 weeks)
In Elaboration, there are 1 sprints. (2 weeks)
In Construction, there are 2 sprints. (5 weeks)

4.2.1.1 Inception Plan (11/10/2024 - 7/11/2024)

In this phase, the main goal is to propose the initial idea, finish the project plan, vision document, project schedule and equip knowledge about project management, firebase and unity workflow.
Sprint 0: (11/10/2024 - 24/10/2024)
This sprint's main objective is to find ideas for the project and create the project proposal for PA0.
Sprint 1: (25/10/2024 - 7/11/2024)
In this sprint objectives are:
- Create project plan
- Create vision document
- Project schedule
- Learn about project management, firebase, nodejs and unity.
- Finish PA1 reports.

4.2.1.2 Elaboration Plan (8/11/2024 - 21/11/2024)

In this phase, the main goal is to finish project designing: Use-case model, Software architecture, UI design and implement architecture in unity.
Sprint 2: (8/11/2024 - 21/11/2024)
In this sprint, objectives are:
- Create Use-case model
- Design Software architecture and class diagram
- Design UI
- Implement MVC architecture in unity
- Implement database structure, accessing.
- Finish PA2, PA3 reports.

4.2.1.3 Construction Plan (22/11/2024 - 26/12/2024)

In this phase, the main goal is to finish implementation of the project, test and prepare for presentation. We will implement all features. Then we will test each module and the whole system.
Finally, we will prepare for the presentation.
Sprint 3: (21/11/2024 - 5/12/2024)
In this sprint, objectives are:
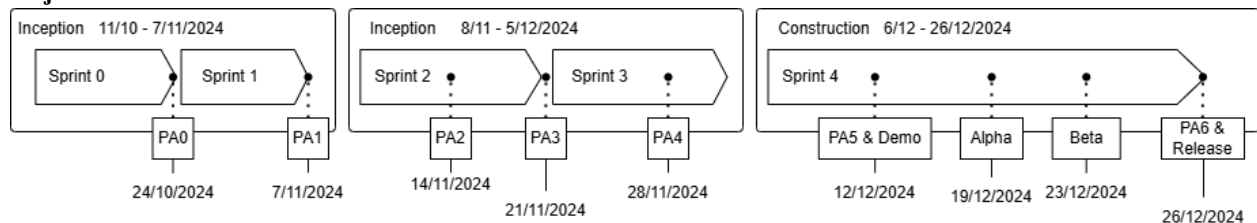- Finish PA4 reports.
- Implement:

- Auth and Account
- Friend management
- Chatting
- Chatbot
- Admin

Sprint 4: (6/12/2024 - 26/12/2024).
In this sprint, objectives are:

- Finish PA5 reports.
- Implement the online chat room.
- Testing.
- Fix bug
- Finish the uncompleted work
- Prepare presentation

Major milestones and timeline

Major Milestones	Date (mm/dd/yyyy)	Achievement criteria
Finish PA0	10/24/2024	Finish Project Proposal.
Finish PA1	11/7/2024	Finish Project Planning and schedule. Finish Vision Document. Finish Project Plan.
Finish PA2	11/14/2024	Finish Use-case model. Finish Use-case Specification Document.
Finish PA3	11/21/2024	Finish Software Architecture design Finish Software Architecture document. Finish Class diagram.
Finish PA4	11/28/2024	Finish UI design. Finish UI Prototype report.
Finish PA5	12/12/2024	Finish Testing Finish Test Plan and Test report.
Finish PA6	12/26/2024	Finish whole project
Release: demo version	12/12/2024	See 4.2.2
Release: alpha version	12/19/2024	See 4.2.2
Release: beta version	12/23/2024	See 4.2.2
Release: release version	12/26/2024	See 4.2.2

Work Breakdown Structure
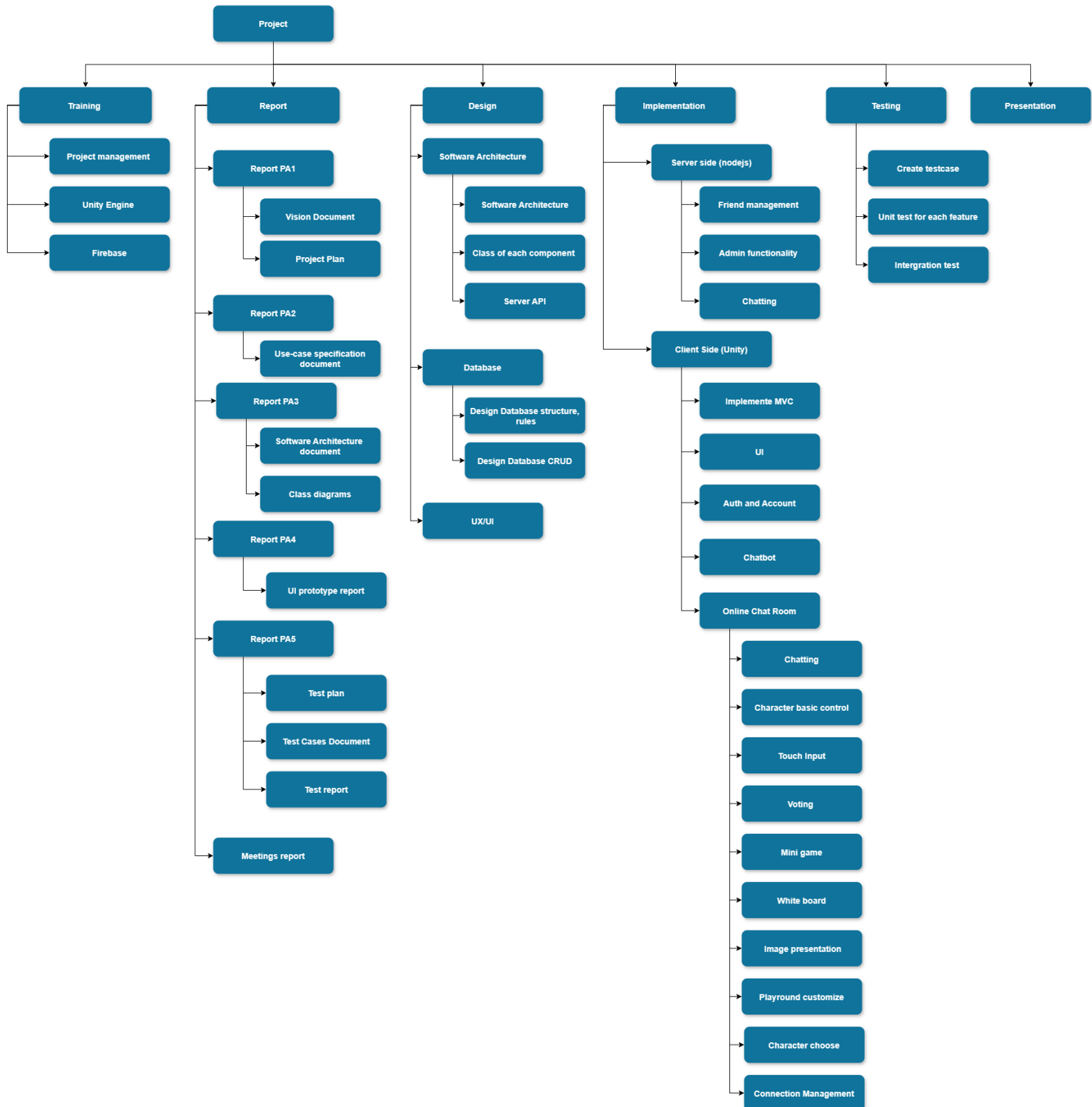
4.2.2 Releases

There are 4 version of release in our project:

12/12 demo version: include all basic features (auth, friend, chatting, chatbot,...) except online chat room.

19/12 alpha version: include feature in demo version and basic online chat room: chatting, character selection, character movement.

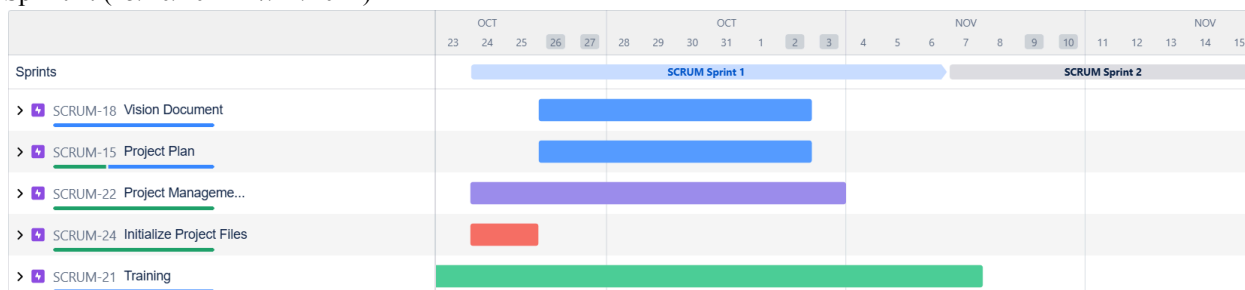23/12 beta version: include all features after testing.

26/12 release version: final version of this project.

4.2.3 Project Schedule

Sprint 0: (11/10/2024 - 24/10/2024)

Task		Task Owner	Output
Genres	Descriptions		
Report (PA0)	[Proposal] Find ideas and write the Proposal.	All members	Project Proposal

Sprint 1: (25/10/2024 - 7/11/2024)

Sprint 1: (25/10/2024 - 7/11/2024)

Task		Task Owner	Output
Genres	Descriptions		

Training	[Training]
	Learn about Unity:
	  - Framework
	  - Language C#
		All members	x

	[Training] Learn about Firebase.	Tuấn, Tài	
Report (PA1)	[Vision Document] Write the Vision Document.	Tuấn,Tài	Vision Document
	[Project Plan] Write the Project Plan.	Hoằng, Đạt, Quang	Project Plan
Project Management	[Project Management] Schedule Task in Project	Hoằng, Đạt, Quang	

	[Project Management] Weekly Scrum Meeting minutes Weekly Scrum Report x 2 Scrum planning meeting minutes Scrum review meeting report minutes	Quang	Weekly Report and meetings minutes documents
	[Project Management] Create Jira Project	Hoằng	x
Initialize Project File	[Initialize Project File] Create Github repo Create Google drive Create unity project and import firebase SDK and Vroid SDK	Hoằng	Code in github

Sprint 2: (8/11/2024 - 21/11/2024)
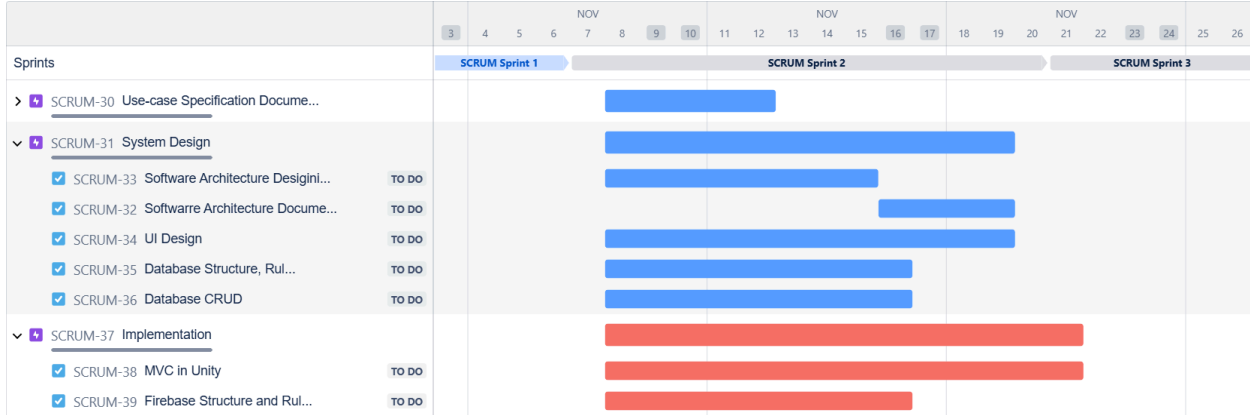
Task		Task Owner	Output
Genres	**Descriptions**		
Report (PA2)	[Use-case specification document] Write Use-case specification document	Tuấn, Tài	Use-case specification document
Report (PA3)	[Software Architecture document] Write Software Architecture document base on draft	Hoằng, Đạt	Software Architecture document
	[Software Architecture document] Draw Class diagrams base on draft	Quang	Class diagrams
Design	[Database structure design] Design Database structure, rules	Tuấn	Database design draft
	[Database CRUD design] Design Database CRUD	Tuấn	Database design draft

	[Software Architecture design] Design Project Architecture: - Define how client/server work. - Define how many components in client/server, what they do, how they connect. - How to deploy them. - How to implement each component.	Hoằng	Software Architecture document and class diagram draft
	[Software Architecture design] Design Class of each component on client	Hoằng	
	[Software Architecture design] Design Class of each component on server	Đạt	
	[Software Architecture design] Web Server API design	Đạt	
	[UX/UI design] Design UI for all page	Tài	UI design draft
Code	[Implementation] Implement MVC in unity	Hoằng	Code in github
	[Implementation] Implement firebase database rules and structure	Tuấn	

Sprint 3: (21/11/2024 - 5/12/2024)
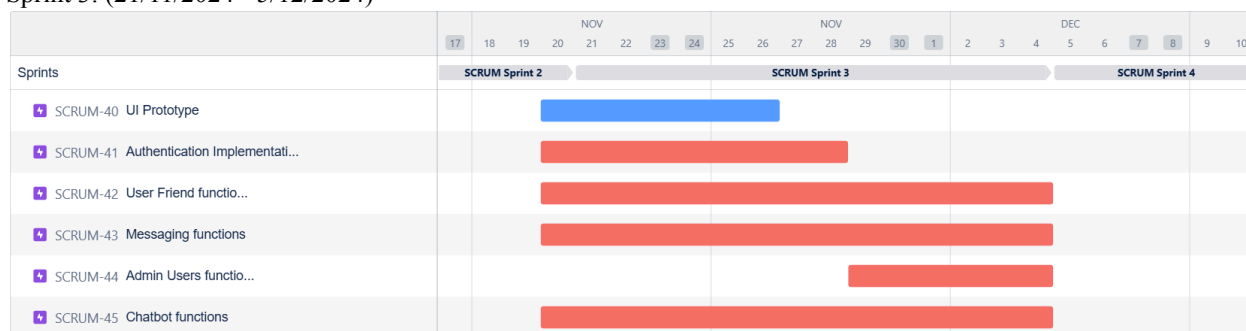
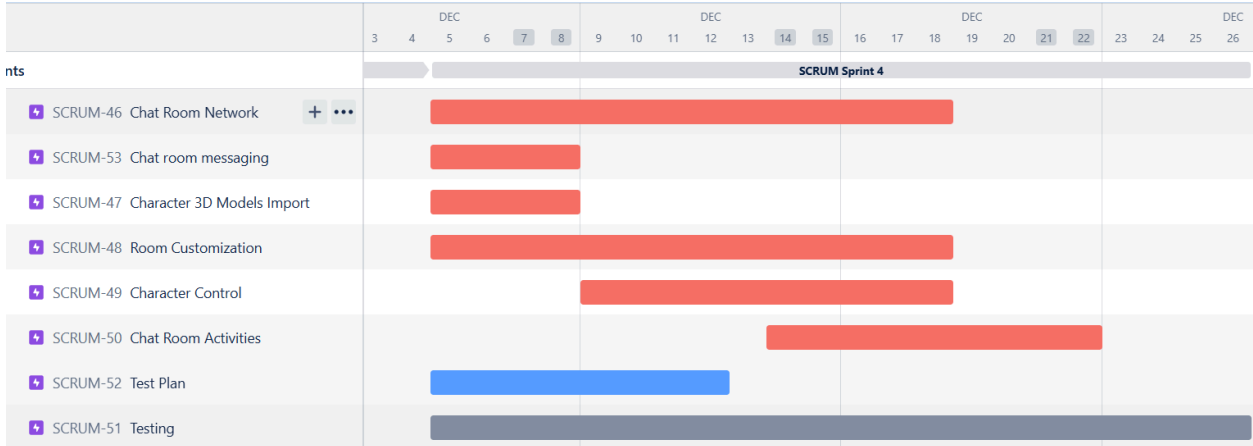| Task		Task Owner	Output
Genres	Descriptions		
Code	[UI] Implement UI in unity	Tài	Code in github
	[Client/Authentication] Implement Auth functionality based on Firebase Implement password reset based on Firebase Implement user profile view/update based on	Quang, Tuấn	

	Firebase		
	[Server/User Friend functionality] Implement add/remove friend functionality. Implement view friend list functionality.	Đạt	
	[Server/Messaging] Implement message send/receive functionality. Implement message history reading functionality.	Hoằng, Đạt	
	[Client/Chatbot] Implement Chatbot functionality based on gemini Implement Chatbot avatar changing functionality based on Vroid SDK. Implement Chatbot chatting UI	Hoằng	
	[Client/Admin] Implement Admin check chat history functionality Implement Admin lock user account functionality Implement Admin View user account list and lock account UI Implement Admin View chat history UI	Hoằng, Quang	
Report (PA4)	[UI prototype report] Write UI prototype report	Tài	UI prototype report

Sprint 4: (6/12/2024 - 26/12/2024)

Task		Task Owner	Output
Genres	Descriptions		

Code	[Online chat room/Network Management] Implement Network connecting management Implement joiner management functionality	Hoằng, Đạt	Code in github
	[Online chat room/Character Model Import] Implement custom model import/select functionality	Hoằng	
	[Online chat room/Room Customization] Implement customization playground functionality	Hoằng, Đạt	
	[Online chat room/Messaging] Implement message sending.	Quang	
	[Online chat room/Character control] Implement touch input module Implement player movement Implement player emotion Implement player animation	Hoằng	
	[Online chat room/Chat Room Activities] Implement mini games	Quang, Tuấn	
	[Online chat room/Chat Room Activities] Implement hosting voting	Quang, Tuấn	
	[Online chat room/Chat Room Activities] Implement image presentation	Hoằng, Quang	
	[Online chat room/Chat Room Activities] Implement white board	Hoằng, Đạt	
Testing	[Testing] Create test case	Quang	Test case
	[Testing] Testing	Quang	Test report
	[Testing] Find and report bug	Đạt, Quang	
Report (PA5)	[Test report] Write test report	Quang, Đạt	
	[Test plan] Write test plan	Quang, Đạt	Test plan
Presentation	Prepare presentation	All members	x
	Create PPT	Tài	PPT

4.3 Project Monitoring and Control

4.3.1 Reporting

To foster collaboration, ensure all team members are aligned and prevent miscommunication, members should follow to the following protocols when reporting project status or issues they met:

- **Weekly meeting**: report completed and uncompleted tasks to keep track of weekly progress. Task allocation or delay must be approved by members in these meetings. Major requirement clarification should also be considered and discussed.
- **Informal chats**: for coding or minor problem solving problems discussion.

4.3.2 Risk Management

(!) Lower priority number implicates higher important:

- *0: Very important - all activities must be frozen to handle this risk when it happens.*
- *1: Important - solution to this problem should be approved by all members.*
- *2: Medium - a number of related team members should help each other solve this problem.*
- *3: Low - each member should be able to solve this on their own, but they can ask for help if needed.*

Risk ID	Risk Description	Probability	Impact	Risk Exposure	Priority	Mitigation Strategy or Contingency Plan
1	Insufficient cost estimation	0.01	30	0.3	0	(1) Cut down costs, look for ineffective, costly processes. (2) Change APIs, tools to open-sourced or free to use ones, change the requirements to fit new technology accordingly.
2	Copyright issues	0.001	100	0.1	0	Change the violated functionalities, look for alternative assets.
3	Missed deadlines, scheduling issues and health issues	0.4	80	32	1	Rescheduling, members should fill in when someone is unable to continue their task in the current sprint. Members' task delay should be reported early for the team to react.
4	Requirement updates	0.3	90	27	1	Meetings and plan updates are required. Earlier affected implemented functionalities should be patched before building new ones.
5	Potential dangerous bugs, defects, and architecture rebuilds.	0.1	100	10	1	Code updates should be reviewed gradually. Testing and experiment should be done on different branches of git.

Risk ID	Risk Description	Probability	Impact	Risk Exposure	Priority	Mitigation Strategy or Contingency Plan
6	Ineffective task allocation	0.2	60	12	2	Members should have 1 main role and another extra role to help other members who are in trouble.
7	Quality assurance failure	0.3	60	18	2	Features should be tested and fixed before submission.
8	Miscommunication	0.8	60	48	2	Important announcements must be clearly stated, pinned and easily accessible to all members. Impactful action should be approved by all members before being done.
9	Lack of knowledge about Unity, network, etc.	0.8	70	56	3	Self-studying, sharing related documents, finding help from teammates, etc. Breakdown hard tasks into smaller, more manageable ones.

4.3.3 Configuration Management

Our team used a combination of tools: google drive, github and jira to manage the project.
- We used Jira to manage the work of each member and the project schedule.
- Our Report and Document is saved in google drive folder: https://drive.google.com/drive/folders/1OM5elqL0YIOQpqceus6kYW2DLgmJs-wY?usp=drive_link
- Our project source code, assets and releases is published in github: https://github.com/cutehusky/VirtualChat3D

(please send your email/username of your github account to email lnhoang22@clc.fitus.edu.vn for permission).