RadiantIQ

UNIVERSITÀ DI TRENTO
Department of Information Engineering and Computer Science

Project's acronym:	RADIANTIQ
Project's title:	RadiantIQ
Start date:	08/03/2024
Finish date:	DD/MM/YYYY

D1 - RadiantIQ Software Specification

WP1:	Software Specification
Task 1.1:	Define Software Specification
Submission date:	09/04/2024
Responsible:	RadiantIQ
Version:	1.0
Status:	Completed
Author(s):	Lorenzo Cattai, Gabriele Pernici, Anh Tu Duong, Lorenzo Negut
Deliverable type:	DOCUMENT

Version list:

Version	Authors	Date	Description
0.1	Anh Tu Duong	09/04/2024	Define the scheme of the document
1.0	Anh Tu Duong	11/04/2024	Complete domain analysis and objectives

Contents

Acronyms

Acronym	Description
FRS	Functional Requirement Specification
FSD	Functional Specification Document
LLMs	Large Language Models

1 Introduction

The project consists in a platform providing a better learning experience for scientific subjects. The idea is to include standard formal explanations of topics (with associated exercises) accompanied by a small number of interactive minigames. To better involve the students, each exercise will be put in an AI generated context (e.i. a phisics problem related to speeds and distances could be told using the story of Achilles and the turtoise). Moreover, the formal explanations can be genrated by an AI, uploaded by a professor or by a combination of the two. Lastly, AI is used to suggest which topics should be revised for the students using the platform.

2 Domain Analysis

2.1 Domain: Interactive Education

We want to create an application in the education domain and try to make it extremely interactive.

2.2 SWOT analysis

- **Strengths:**
  - (1) Use of AI to generate basic information in the articles, with the supervision of admins.
  - (1) Use of AI to generate more advanced information from user's customized necessities and weaknesses
  - (2) Use a more fun and interactive approach in the education process
  - (2) Learn by trying with minigames with immediate feedback or in a more standard way with articles
  - (3) Possibility for experts to contribute with their own articles
  - FLOSS Education Platform
- **Weaknesses:**
  - Developing effective mini-games' experiences might be time consuming from a developer perspective
  - Making fun and yet instructive experiences is hard
  - Careful management of the AI generated information is needed
  - Quality check on article uploaded must be implemented
  - Costs of using LLMs
- **Opportunities:**
  - (1) Using AI in education is an emerging idea that has not yet spread widely and can make this system unique
  - (2) Various studies show the positive effect of interaction and hands-on experiences in the learning process
  - Large demand of easy and complete ways to learn science
  - (3) Lack of effective and proficient communication/cooperation/interaction between students and teachers which creates an ample improvement margin to be enhanced
- **Threats:**
  - Possible limitations of AI technologies from government entities
  - Minigames must be entertaining to be successful
  - Brilliant is a direct competitor (tho it does not currently use AI nor personalize its exercises and lectures)
  - Hard to find funding

3 Project Objectives

3.1 Use of minigames in interactive learning

Create some minigames to make learning more interactive and more intriguing for students.

3.2 Use AI to help develop a more compelling learning experience

Use AI to generate compelling descriptions of the topics, to make learning funnier.

3.3 Make learning science more approachable and enjoyable

Make the platform an accessible starting point in the learning of science, to allow everyone to learn science using intuition and reason.

3.4 Provide single-topic focused content

Providing single topic courses means the possibility to create a learning path specific for the interests of the user.

3.5 Provide private classes as instances of a course

Create classes, from the general courses, to allow teachers/professors to integrate the platform in their standard lectures.

3.6 Allow better student-professor interaction

Using the classes the professors can understand which topics are clearer and then provide feedback. Moreover students can easily determine the level of comprehension of each topic before a test.

3.7 Provide quality guarantees on the material published

Have personnel checking the validity of the published material, while not interfering with the private resources.

3.8 Collect data and provide a progress history

For the learning user, having feedback on the level of comprehension is fundamental and will help them focus more on the less understood topics.

4 Actors

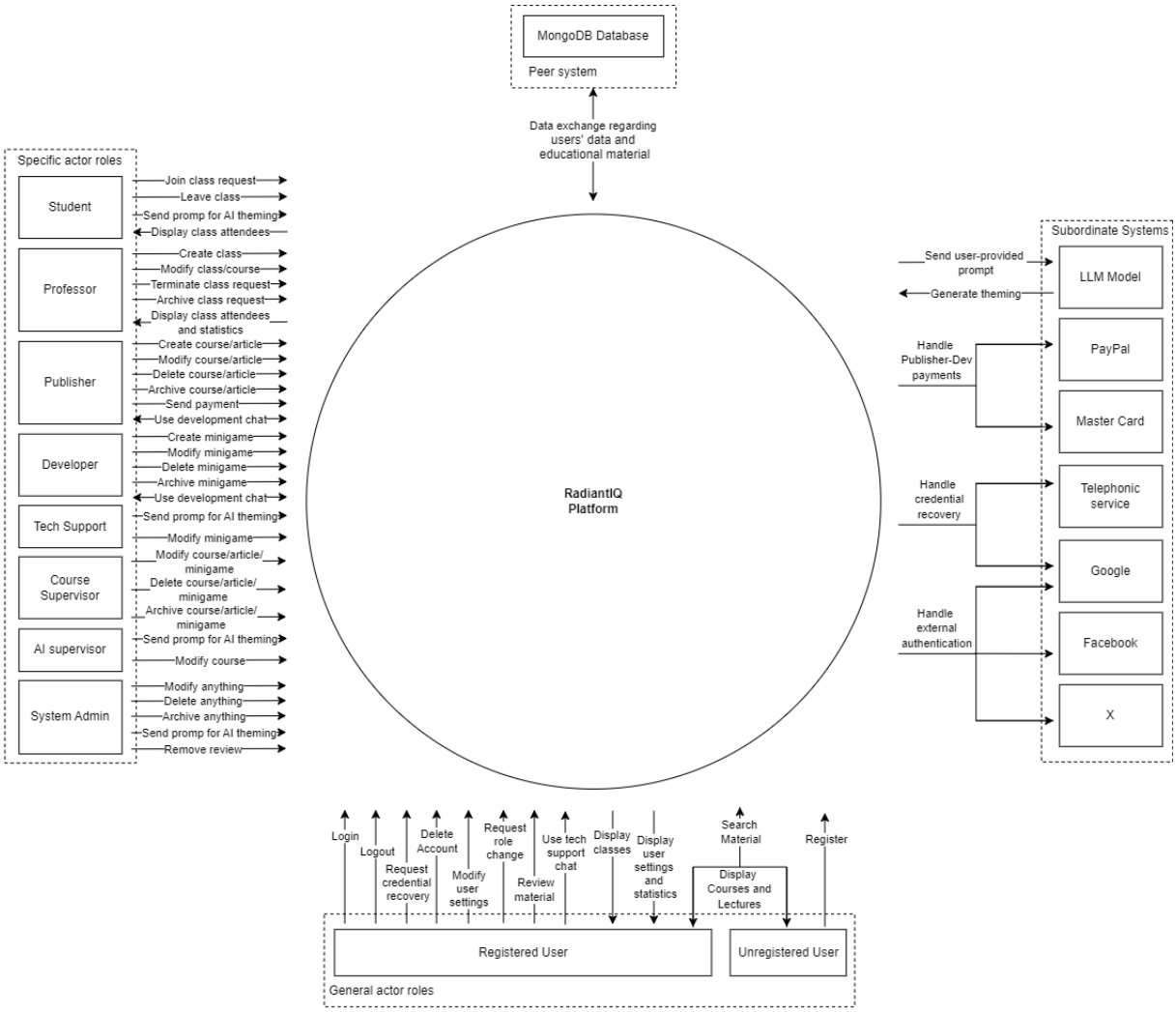
5 Functional Requirements

6 Non-functional Requirements

7 Usecases

8 Context Diagram

We created a context diagram separating the general roles (registered and unregistered users), the specific roles (all the others), the subsystems (payment, authentication, credential recovery, LLM) and the peer system (database).



9 Components Diagram

A components diagram was create for better visualization about all the main components of the system.

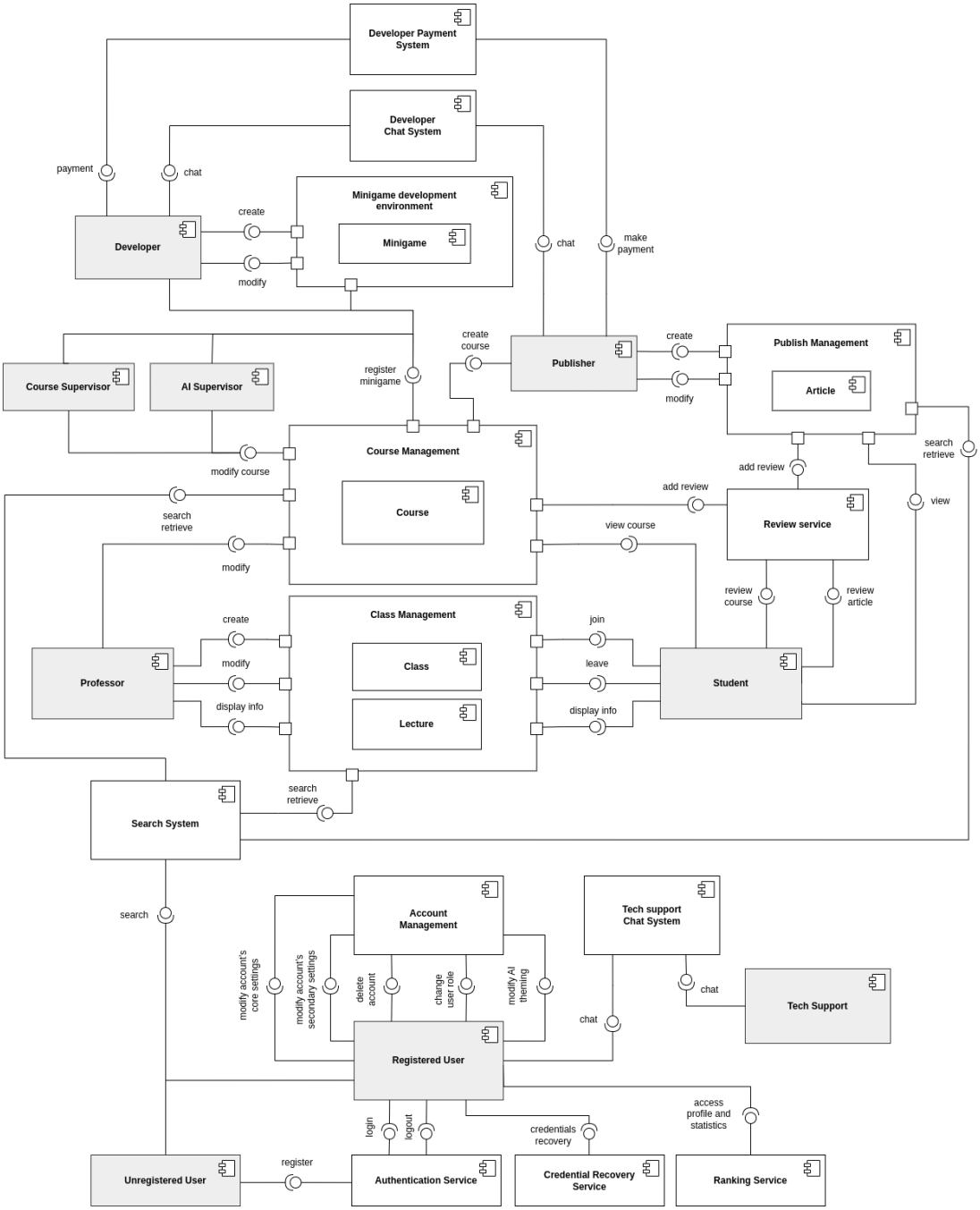


Figure 2: RadiantIQ - Components diagram

10 Class Diagram

11 Conclusion