Scholar app

Table of Contents

[**INTRODUCTION** 3](#_Toc162332832)

[**Problem statement** 3](#_Toc162332833)

[**Project Description.** 3](#_Toc162332834)

[**REQUIREMENTS** 4](#_Toc162332835)

[**Functional Requirements:** 4](#_Toc162332836)

[**Non Functional Requirement:** 4](#_Toc162332837)

[**UML DESIGNS** 5](#_Toc162332838)

**INTRODUCTION**

In the world today, there are a lot of people who face challenges in picking the right school for their college studies, programs, and opportunities to obtain funding for their education. This is where comes in.

Scholar is an AI model trained on information related to universities and colleges in the world and funding opportunities and scholarships.

It enables its users (students) make informed choices about where to study, how to get funding if applicable and keeps them informed on scholarship opportunities available.

**Problem statement**

The lack of information propels students to make lofty decisions with respect to where to study and also being less informed about scholarship and funding opportunities.

**Project Description.**

This project Scholar App aims at providing students with appropriate information on universities and colleges, and the programs they offer. It also provides students with information about funding and scholarship opportunities.

This is done with the help of a model (Scholar) built by fine tuning an LLM on information regarding colleges and universities integrated into our user interface (frontend) which permits user interactions with our model.

In this document, we will provide a detailed analysis and design of the Scholar app, including its functional and non-functional requirements, use cases, system architecture, data model, user interface, and testing requirements. This document is intended for developers, project managers, and other stakeholders involved in the development and implementation of the system.

**REQUIREMENTS ANALYSIS**

**Functional Requirements:**

* Users should be able to register login or delete his account.
* Users should be able to perform CRUD operations on chats with model.
* Users should be able to communicate with model

**Non Functional Requirement:**

* App should provide security of user and user information.
* App should be able to serve multiple users simultaneously while maintaining performance.
* App should have an easy to use user interface.
* App should be scalable.
* App should be available to users at all times.

**ACTORS:**

* Students (high school students and university/college students)

**System Constraints**

* Security: The App must ensure that all user data is stored securely and access is restricted to authorized users only. This may require implementing robust authentication and authorization mechanisms and encrypting chat history.
* Scalability: The App must be able to handle a large numbers of users and store user chats. This can be achieved with the use of multiple servers to support and store user chat history.
* Availability: The App must be available 24/7 to ensure that students get aid (information they require) anytime they please.
* Performance: The App must be able to serve users quickly and effectively. This may be done using caching techniques, and minimizing network latency.
* Compliance: The App must comply with relevant laws and regulations related to data privacy and security. This may require implementing specific data protection measures such as pseudonymization, data minimization, and access controls.

**APPLICATION DESIGN**