

Thoth Requirements Specification Document

October 2023

1 Introduction

1.1 Document Purpose

This document is intended to provide a detailed description of system requirements and the constraints that should be satisfied. The targeted audience is the team members to provide agreement between them on what should be implemented.

1.2 Product Scope

This system will be an Interactive Programming Tutorial (C++ for a start). It will be different than other alternatives because it will provide a smoother interface. In addition, it will depend on multiple APIs that will be used to enhance the learning experience.

1.3 Definitions, Acronyms, and Abbreviations

The project's Definitions, Acronyms, and Abbreviations are currently under development

1.4 Document Overview

This document will include the product functions summarized and the product constraints. In addition, It will describe the user's Characteristics, the assumptions and dependants on the projects, and how the requirements will be transformed into function in the system(Apportioning of Requirements). Moreover, It will cover the user, hardware, and software interfaces of the system. In addition to functional and Non-functional requirements.

2 Product Overview

2.1 Product Functions

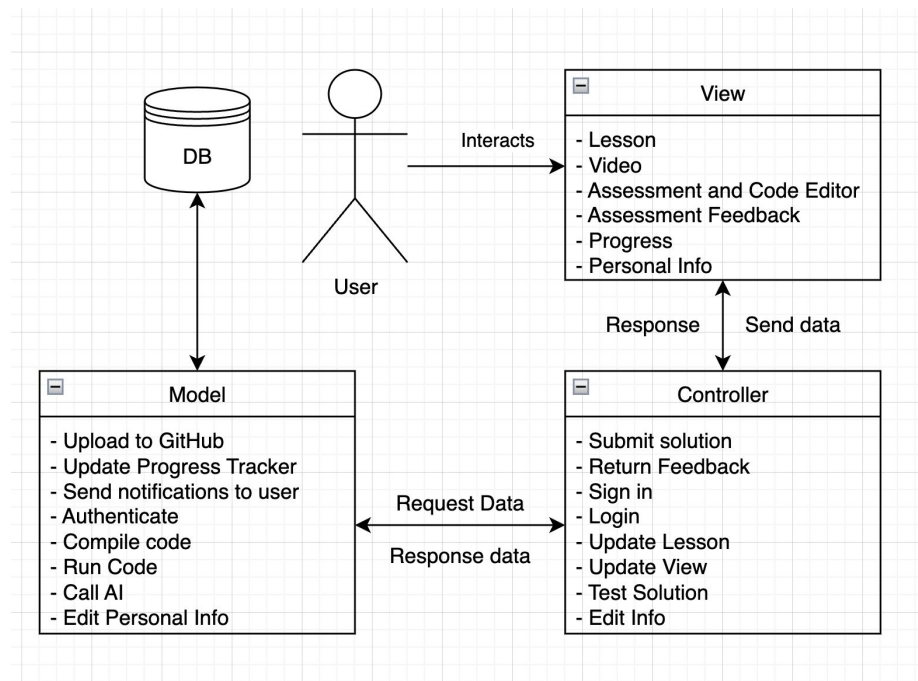
The users shall be able to sign up and track their progress across devices. In addition, the users shall be able to edit personal information and add friends to

their accounts. The main aim of the app is to enable the users to write, compile, and run C++ code; feedback on this code should be sent to the users and notify them. The users shall be able to upload this solution to their GitHub accounts.

2.2 Product Constraints

The developers will opt for a smooth user interface that may be more complex to implement. For budget constraints, a less accurate AI server will be used to generate the solution feedback. To ensure security, the accounts will be linked to GitHub accounts which will limit the signing up process if the user does not have a GitHub account.

2.3 User Characteristics



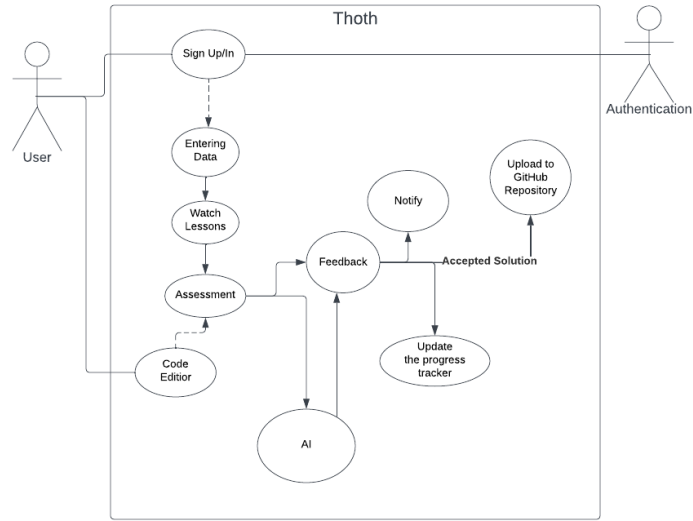
UML

2.4 Assumptions and Dependencies

The app will depend in the first place on GitHub as the accounts will be linked to it. In addition, the users' solutions should be uploaded to their GitHub account. Moreover, the solutions' feedback accuracy will depend on the AI server's accuracy. The usability of the app will be dependent on the APIs.

2.5 Apportioning of Requirements

As shown in the use case, the ability to sign up/in requirement will be represented in the (sign up/in). The ability to write, modify, and compile code represented by (code editor). The requirement of sending AI feedback is represented by (assessment) then (AI) then (feedback). And the requirement of notifying the user is represented by the (Notify) branching from (feedback). The requirement of tracking progress across devices will be done using (update the progress tracker) The requirement to upload the solution on GitHub is represented in (Upload on GitHub repository)



3 Requirements

3.1 External interfaces

3.1.1 User interfaces

The users will send data to the system through a simple GUI that will include space to write code and a button to submit their code to the system. In general, users will be only using the provided GUI to communicate with the system.

3.1.2 Hardware interfaces

The exact list of supported devices is not yet specified.

3.1.3 Software interfaces

1. The problem set and the test cases for each problem shall be stored on an external database server.
2. The users' information shall be stored in an encrypted format in an external database server.
3. Vim text editor software shall be embedded in the system allowing for typing code. Nvim API will be used to connect to this software interface.
4. The system shall connect to an external judge API to be able to run the submitted codes against the test cases. The exact judge used is not yet specified.
5. The system shall connect to an external algorithm graphical visualiser API to view some of complicated algorithms. The exact algorithm visualizer is not yet specified.

3.2 Functional requirements

1. The users shall be provided with problem sets covering the topics in the tutorial in order to help them understand these topics.
2. The users shall be able to write, compile and run **C++** code through a built-in code editor to solve problems from the provided problem set.
3. The users shall be able to sign up in using their GitHub profiles or email addresses.
4. The users shall be able to track and view their current progress across different devices.
5. The users shall be able to upload their solutions and codes to their Github profiles.
6. The users shall be able to add and edit their personal information (e.g. their username, password, and profile picture).
7. The users shall receive an AI-generated feedback on their codes and solutions to problems if they fail to solve a problem.
8. The users shall be able to add other users as friends and view their progress.
9. The system shall be able to send notifications to users through the web application.

3.3 Non-functional requirements

1. **Security :** The users' stored data should be stored in an encrypted format to achieve safety.
2. **Performance :** The system shall provide feedback for the submitted solutions in a time period ranging from **thirty seconds** to no more than **five minutes**, depending on the problem and the size of the test cases.

4 Verification

5 References

The project's list of references are currently under development.

6 Appendices