SOFTWARE REQUIREMENTS SPECIFICATION

for

Learnix

Version 1.0

Prepared by Vincent Le, Bayasgalan Battogtokh, Devin Irby, Miguel Lima, Wyatt Stohr, Xander Thompson, Boscoe Lindholm

Project 1: AcademicGPT

March 12, 2024

Contents

List of Figures

Revision History

Name	Date	Reason For Changes	Version
Vincent Le	1/19/24	Initial Version	1.0
Xander Thompson	3/11/24	Updated to match Tech Demo	1.1

1 Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide an in-depth examination of the features and specifications necessary to create a mobile application intended to support academic research by using Meta's Llama 2 model, known as "Learnix". For the development team, this SRS creates a comprehensive roadmap and offers careful direction during the project's design, implementation, and testing stages.

1.2 Intended Audience

The development team behind Learnix is the main target audience for this SRS. It gives them a clear understanding of the necessary features, and functionalities that need to be implemented. Additionally, project managers, university officials, testers and clients can also refer to the SRS to comprehend the objectives, features, and scope of the app.

1.3 Project Scope

Learnix is intended to make research tasks easier by saving time and effort though quickly finding and retrieving relevant and credible scholarly articles. Through Learnix, researchers will improve their knowledge in their research topics. Learnix is the solution to increase research productivity and efficiency all while improving research quality by providing access to credible scholarly sources.

Learnix will accomplish these goals with with the following project scope:

- Will provide academic research help: Such as searching articles and answering questions.
- Interactive UX: Enabling users to interact with the application through queries and presenting results clearly and concisely.
- User profiles: Enabling users to set up profiles that keep track of their search history, login information, and number of queries. This feature lets users track their research progress and personalizes the user experience.
- Sign in/Log out feature: Users will gain access to their profiles and save their information by using the sign in feature. By logging out, users can safely exit the app and protect their privacy.

- Filter function: Will recognize and remove non-academic results will ensure that the results returned are relevant to research.
- Search Query Results: Display relevant information, and present search results in an approachable format.

The software will not be able to do:

- Provide harmful information to users.
- Be used outside of an academic setting or targeted users.

2 Overall Description

2.1 Product Perspective

Product Overview

Learnix will be an proof-of-concept android application that will allow users to ask an AI chatbot to search for academic research on nearly any subject they want. Learnix will return results from the search in a timely manner. The users will have an account that they will need to login into or create. This account will keep track of the user's previous five queries. Learnix should be able to run with up to 10 users online at any given time.

Context And Scope

Learnix will need access to the Internet and to two databases for user accounts and previous queries. The AI chatbot will interact with the textbox where the user will have their inputs and receive the outputs from the chatbot.

Stakeholders

The stakeholders of this product are the group of individuals working on the product and the mentor for the whole group.

Business and Usage

The product will not be open to the public and shall not be used outside of the stakeholders unless agreed upon by all parties.

2.2 Product Functions

The application has 3 major functions: a login mechanism when the application is started, a chatting functionality to interact with the academic AI, an easily accessible search history and statistics.

The login function will be presented to the user upon application start up if the user is launching the application for the first time or if credentials that were saved previously become invalid. The user will give an email and password, which upon pressing 'Login' will check a database for matching credentials and report back to the application if it successful or unsuccessful. Upon successful login, the screen should disappear and take

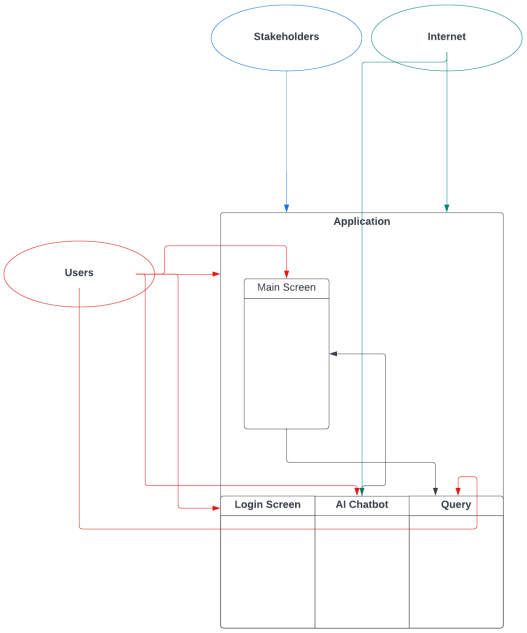

Figure 2.1: Use Cases Diagram

them to the main AI chat function. An unsuccessful login should present the user with an error detailing the error that occurred.

The core part of the application is the interact to chat with the AI to gain information on a particular academic subject. After logging in, the user will be dropped into an interface with an introduction message from the AI on screen and a box at the bottom which will allow to type in a question using an English keyboard. Upon pressing enter, or the send button on the box, the question will be sent to where the AI is being hosted at. The answer will be sent back to the user once it has been generated and processed on the server and will display underneath the question that the user asks.

The chat will keep messages displayed on screen for as long as the user interacts with the AI, with the screen scrolling when text starts to go off screen during the conversation.

Statistics and previous questions asked should be available to the user in a button towards the top of the screen. When clicked it will give information as to how many queries the user has done since using the application and will list the last five queries that the user has asked the AI in the chat function. This will also allow the user to tap one of the queries to quickly ask the AI the question that had been asked in a previous query.

2.3 User Classes and Characteristics

The intended user-base for Learnix will be college students. The intended age range is 18 years or older. Reading level for Learnix should be at the college level. Learnix's training database consists of academic papers written by and for scholars. Therefore, the Llama 2 responses should be a summary of its findings an undergrad student could understand.

The user should be able to operate a phone and web browser with some accessibility assistance where it may be necessary. The software assumes that the user is capable of using a touch screen and can see the screen. If this is a problem for the user, then the software should still be usable with external assistance, such as OS level accessibility features.

2.4 Operating Environment

The intended operating environment for Learnix should be a device running Android 12 or later. While Learnix is powered by web frameworks, allowing for basic web browser capability, Learnix should also run as a native app on the Android device.

2.5 Design and Implementation Constraints

- Regulatory policies;
  - Llama 2 License rights and agreements.
- Hardware Limitations (e.g., signal timing requirements);
  - 64GB of memory to run a quantified 70B Llama 2 Model is the minimum requirement. For faster generation 48GB VRAM is needed to fit entire model.
  - 512 MB of RAM for mobile application to run as expected with full functionality.
- Data availability and Quality;
  - Availability of high-quality training data is crucial for the AI model's performance.
- Security Constraints;
  - User information should be stored encrypted. Sensitive information during the process should be left out of the GitHub repo.
- Language Support;
  - Software requires user to insert input in English measures within reason should be applied to ensure input is in English.
- Cost Constraints;

- The developer budget is low, targeting to stay below 28 USD. Using technologies that are low cost and open source.

2.6 User Documentation

Because this version of Learnix will serve as a proof-of-concept, there will only be limited user documentation.

2.7 Assumptions and Dependencies

Assumptions

- Phone compatibility: The phone's operating system is compatible with the required applications.
- Model functionality and availability: Assume that the AI model's functionality and availability meet our requirements.
- Ethical use: Assume that the users will not attempt to "hack" the AI model through prompt engineering
- Users knowledge: Assume that the users have basic knowledge of technology.
- AI model updates: Assume that the application's core functionality will not be affected by AI model updates.

Dependency

- Security & privacy: To comply with applicable regulations, implement security measures for user login credentials
- Network connectivity: Without a proper internet connection, the application features may be limited or not work.
- Framework compatibility with databases: Database system needs to be compatible with the application's development framework.

3 Specific Requirements

3.1 Front End Requirements

The user interface of Learnix will have:

- **Login Screen**: First screen seen upon opening the application.
  - **Username field**: Text box that can accept 20 characters of text input from users.
  - **Password field**: Text box that can accept 20 characters of text input from users.
  - **Login Button**: After the username field and password field are filled, the button can be pressed to login to the application and bring the user to the Main screen.
  - **New User Button** If a user has yet to create an account, this button can be pressed to bring the user to the New User Screen.
- **New User Screen**: Screen that loads after the "New User Button" is pressed on the Login Screen.
  - **Username field**: Text box that can accept 20 characters of text input from users.
  - **Password field**: Text box that can accept 20 characters of text input from users.
  - **Create New User Button**: Button to press after inputting username and password to create a new account. The user will be taken back to the login screen to log in again after creating the new account.
- **Main Screen**
  - **Search Bar**: Text box on the bottom of the screen that can accept 150 characters from the user. Entries can be copied and pasted into the search bar.
  - **Submit Button**: A button directly to the right of the search bar to submit the query and get results from query.
  - **Results Text Box**: Shows results of query search in scrollable text box. Results can be copied and pasted.

- **Drop Down Menu**
  - **Logout Button**: Button to log out of application. This will return a user to the login screen.
  - **Previous Queries Tab**: Shows the five most recent queries made by the user that is currently logged-in in a table format. Queries can be copied and pasted.
  - **Total Queries Counter**: A tally of the total queries made by the current user.

3.2 Back End Requirements

3.2.1 Functional Requirements

- **Llama 2 Model Integration**: The system shall seamlessly integrate the Llama 2 model from Meta, ensuring that the academic research chat AI can understand and respond to complex academic queries accurately and efficiently.
- **MySQL Database**:The system shall utilize a MySQL database, optimized for fast read/write operations, to perform queries on the model history, user authentication, and user data.
- **User Authentication**:The system shall authenticate users using a secure and robust authentication mechanism, leveraging stored credentials in the MySQL database. It should also provide features for password recovery and account lockouts after repeated failed attempts.
- **Version Control**:The system shall maintain version control using GitHub, ensuring that all changes are tracked and can be rolled back if necessary. It should also support branching strategies for parallel development.
- **Flask Server**: The system shall use the Flask server to make API calls to additional sources as needed to provide external information for the AI, access/save information to the MySQL Database, and provide User authentication. The primary goal is to route requests from the mobile application to where they should go appropriately.

3.2.2 Non-Functional Requirements

- **Performance**: The system shall respond to user queries within a reasonable period, not exceeding 20 seconds under normal load conditions. It should also implement caching strategies to improve response times.
- **Scalability**: The system shall be designed with scalability in mind, capable of supporting an increasing number of users and data volume without a significant

drop in performance. It should be able to scale horizontally and vertically as required.

- **Security**: The system shall store user credentials securely using industry-standard encryption methods. All data exchanges shall occur over secure protocols to protect against data breaches.
- **Maintainability**: The system shall be maintainable, with clear documentation and modular code to facilitate updates and bug fixes. It should follow best practices for code quality and testing.
- **Compatibility**: The system shall be compatible with React Native for front-end development and API hosting. It should follow a micro-services architecture to ensure loose coupling between the back-end and front-end.