🟦 Sprint 3 Plan Document

🗒️ Introduction

Sprint 3 is aimed at consolidating the foundational work completed in Sprint 2 by focusing on the integration and refinement of the Q&A platform. This includes the enhancement of the Retrieval-Augmented Generation (RAG) framework, the development of an intuitive user interface (UI), and the initiation of testing phases to ensure functionality and user satisfaction.

🗓️ Objectives

1. **Complete the LLM Training**: Finalize the training of the Language Model with the cleaned and structured data from Sprint 2.
2. **Develop and Integrate UI**: Create a user-friendly interface for interacting with the Q&A platform, including the capability for users to input URLs.
3. **Enhance RAG Framework**: Improve the RAG framework based on Sprint 2 outcomes and ensure its integration with the UI and the Language Model.
4. **Implement URL Processing Functionality**: Enable the platform to extract and use information from user-provided URLs for generating responses.
5. **Initiate Testing and Feedback Gathering**: Develop a testing plan for the platform's functionality and user interaction, and set up a system for collecting initial feedback.

🛠️ Tasks and Deliverables

1. **LLM Training Completion**
   - Finalize preprocessing and training of the domain-specific LLM.
2. **UI Development**
   - Design the UI with an emphasis on simplicity and user engagement.
   - Implement text interaction and URL input functionalities.
3. **RAG Framework Enhancement**
   - Address initial challenges and refine the RAG framework for better accuracy and integration.
   - Responsible Team: AI Development Team, Back-end Development Team
4. **URL Processing Functionality**
   - Develop functionality to process content from user-provided URLs.
   - Integrate this functionality with the RAG framework and UI.
5. **Testing and Feedback Initiation**
   - Develop and execute a testing plan covering functionality, UI, and response accuracy.
   - Implement a feedback collection mechanism for initial user and stakeholder feedback.

Requirements to Develop

Requirement	User Story	Estimation	Priority	Jira Issue
Furhat Robot interaction Must be able to filter and sort information based on user-defined criteria such as location, size, and function of the room.	U2.2: As a user, I want the robot to filter and sort information based on my provided criteria (e.g., location, size, function of the room), so that I can find what I'm looking for more efficiently.	MEDIUM	SHOULD HAVE	☑ C2QK-4: Must be able to filter and sort informati on based on user-defined criteria such as location, size, and function of the r oom. 待办

Furhat Robot interaction Must be able to describe the key information based on the website summary to visually impaired user.	U2.3: As a visually impaired user, I want the robot to describe images and visual content from websites, so that I can comprehend visual information that I cannot see.	LARGE	COULD HAVE	☑ C2QK-6: Must be able to describe the key information based on the website summary to visually impaired user. 待办
UI interactive interface Must protect user's private information and provide clear privacy policies.	U1.4: As a user, I want the Q&A platform to protect my personal information and provide clear privacy policies, so that I can ensure the security and confidentiality of my data.	MEDIUM	MUST HAVE	☑ C2QK-7: Must protect user's private information and provide clear privacy policies. 待办
UI interactive interface Support text interaction in the robot interface.	U1.1: As a user, I want to be able to support text interaction in the robot interface, so I can type and express what I need to query.	SMALL	MUST HAVE	This is a must have because it is the basic function of user interaction. It may involve only the front end and is expected to be small.
UI interactive interface Generate an livechat box when asking the question.	U1.2: As a user, I want the robot to generate an livechat box when I ask my question, so I can gain the information and answer I need directly.	SMALL	MUST HAVE	This is a must have because it is the basic function of user interaction. It may involve only the front end and is expected to be small.
Furhat Robot interaction Must summary the information quickly.	U2.1: As a user, I want to interact with the Furhat robot in a conversational manner to obtain information directly, so that I can save time by not having to search and filter information on the web myself.	LARGE	MUST HAVE	This is a must have because the robot needs to feedback the information needed by the user, and involves the model, UI and robot interaction, which is expected to be a large project.
Domain-Specific Language Model function Must provide accurate information of Melbourne Connect to user when asked.	U3.1: As a user interested in services offered at Melbourne Connect, I want the robot to provide detailed information (such as room's information, location and provided services) from websites of Melbourne Connect, so that I can clearly understand the overview of Melbourne Connect	LARGE	MUST HAVE	This is a must have because the language model needs to analyze user problems and obtain relevant information of Melbourne Connect, the project is expected to be large due to the model involved.

	without navigating through those websites.			
UI interactive interface Must be able to guide user on how to using furhat robot.	U1.3: As a new user, I want the robot to offer an introduction on how to use it, so that I can quickly understand and start utilizing the platform services.	SMALL	SHOULD HAVE	This is a should have because the introduction function allows users to better understand the role of QA robots. The front-end is involved, and the engineering quantity is expected to be small.
Domain-Specific Language Model function Must be able to summary the information of the website clearly and quickly.	U3.2: As a user, I want the chatbot to automatically extract and summarize the main content of a website I provide, so that I can quickly grasp what the website is about without reading all the content.	LARGE	SHOULD HAVE	This is a should have because it is based on the language model to complement the functionality. The project is expected to be large due to the model involved.
Furhat Robot interaction Must be able to translate web content from various of languages.	U2.4: As a user learning a new language, I want the chatbot to translate content from websites in foreign languages(e.g. from Chinese to English), so that I can understand the content without being fluent in the language.	LARGE	COULD HAVE	This is a could have because it is based on the language model to complement the functionality. The project is expected to be large due to the model involved.

⚙️ Technologies to Use

- **Front-end Development**: ReactJS or Vue.js for UI development.
- **Back-end & AI Development**: Python for back-end services, using libraries such as Flask for API development, BeautifulSoup, and Scrapy for web scraping, TensorFlow or PyTorch for RAG framework enhancement.
- **Data Storage**: PostgreSQL or MongoDB for storing extracted website data and user queries.
- **Testing and Feedback Tools**: Selenium for automated UI testing, Google Forms or a custom feedback tool integrated into the UI for collecting user feedback.

📈 Expected Outcomes

- A fully functional prototype of the Q&A platform, capable of processing user-input URLs and providing accurate, relevant responses.
- A user interface that is intuitive and facilitates easy interaction with the platform.
- Initial feedback from users and stakeholders to guide further refinement.

🔓 Task Dependencies

1. **LLM Training Completion Before RAG Framework Enhancement**
   - **Justification**: Finalising the domain-specific Language Model (LLM) training provides the necessary foundation for refining the Retrieval-Augmented Generation (RAG) framework. The enhancements to the RAG framework depend on the insights and data derived from the LLM training outcomes to ensure accuracy and relevance in response generation.
2. **UI Development Dependent on Initial RAG Framework Integration**

- **Justification**: The user interface (UI) development, particularly the features for text interaction and URL input, relies on a preliminary integration of the RAG framework to ensure these functionalities are feasible and align with the back-end capabilities. This ensures that the UI design accommodates the technical requirements and limitations identified during the RAG framework's initial enhancement phase.

3. **URL Processing Functionality After UI and RAG Framework Readiness**
   - **Justification**: Developing the functionality to process content from user-provided URLs requires the UI to be at a stage where URL inputs can be tested, and the RAG framework to be ready for integration with this new functionality. This ensures a seamless flow of information from the user input through the UI to the backend processing and response generation.
4. **Testing and Feedback Gathering Post UI and Functional Implementations**
   - **Justification**: Initiating testing and feedback collection processes is contingent upon having a working prototype that includes the completed UI and functional features such as the RAG framework and URL processing capabilities. This ensures that feedback is relevant and actionable, focusing on the user experience and the accuracy of the platform's responses.

⚠️ Challenges and Mitigation Strategies

- **Data Quality and Model Accuracy**: Continuous monitoring and refinement of the LLM to improve response quality.
- **User Interface Usability**: Iterative design and testing with real users to ensure the UI meets user needs.
- **Integration Complexity**: Close coordination between the AI, back-end, and front-end teams to ensure smooth integration of components.

⏳ Next Steps

- Begin UI development and RAG framework enhancement simultaneously to accelerate integration.
- Prioritize the development of URL processing functionality to meet new client expectations.
- Establish a structured testing and feedback schedule to ensure ample time for iteration before Sprint 4.