PI Roadmap and Non Functional Requirement

Sustainable Work through Women-in-tech Application for Older Women in Malaysia and Thailand: Integrating Action Research and Design Science Approach

The Product Management team is responsible for:

● Developing the PI Roadmap (note that it may be quite light on features given the need for a lot of enablers in this PI and you might want to revise it after the Week 4 PI planning)

● Developing preliminary solution-level nonfunctional requirements (NFR) as described in the SAFe website

● Ensuring each Agile team has their Program Backlog and iteration goals for the first iteration

● Ensuring each Agile team completes their PI Objectives/Backlog Tasks for the first PI

Team Name	**Name**	**StudentID**
Team RedCow	Daiki Kubo	30523346
	Vladislav Pikulin	30935679
Team Twenty9	Ng Guo Hao	29799392
	Lau Yi Meng	29274559
Team 404	Kelvin Kong Kai Wen	30861586
	Chia Yong Peng	29938120

RoadMap:

{Empowering women project} PI ROADMAP

	PI 1		PI 2						PI 3					
Team RedCow	**S1**		**S2**			**S3**			**S4**			**S5**		
Login	🟦	🟦												
Forum UI			🟦											
Forum posts with interests, comments and likes/dislikes				🟦	🟦									
Users choose to be anonymous or with names to interact with one another						🟦	🟦							
Forum posts and comments translation with base language of English.							🟦	🟦						
Embedding a video on the forum									🟦	🟦				
Mine data on forum users									🟦	🟦	🟦			
Admin Dashboard UI												🟦		
Represent mined data on admin dashboard													🟦	🟦

Team 404														
Chat UI	🟩	🟩												
Chatbot surveying first-time users		🟩	🟩	🟩	🟩									
Handling of user queries						🟩	🟩							
Assignment system for assigning unanswered questions to researchers							🟩	🟩	🟩					
Feedback system for users to rate responses									🟩	🟩				
Chat translation											🟩	🟩		
Voice input for responses												🟩	🟩	
FAQ Page														🟩
Team twenty9														
Recommender UI	🟨	🟨												
Preference/Topic			🟨											
Content Library				🟨	🟨									
Recommender System						🟨	🟨	🟨	🟨	🟨				
Rating System									🟨	🟨	🟨	🟨		
Voice												🟨	🟨	

Non-Functional Requirements:

Maintainability

The main aspects of maintainability we aim to focus on are extensibility, modularity, reusability, testability, and modifiability.

The main aspect of ensuring maintainability in our project is to divide our software into small modules by having most of the code enclosed in functions to ensure the functionality can be reused at any time. Not to mention that modularising the code will help keep the code modifiable and extensible, which could prove useful if requirements change. Lastly, modularising the code will greatly help us with the testability aspect as it would make it easier for the teams to test individual components of the system and prove their correctness.

Portability

Our application runs on a thin-client architecture. It is hosted on a web server and can be accessed with any web browser software. The architecture allows the client to run the application without downloading any additional software on their computing machine. In addition, the architecture also allows the application to be run on most computing platforms supporting web connectivity including Windows, Mac, Linux and a wide variety of mobile devices such as Android or IOS phones.

In addition to wide hardware compatibility, the architecture allows for updates to be shipped easily. This is due to the fact that updates can be pushed directly to the web server without communicating with the clients.

Security

The application requires users to input their mobile phone number that is used as an identifier. The application then facilitates the login process by sending One-Time Passwords (OTPs) to users' mobile phones to verify their identity. This practice prevents unauthorized access of their accounts as logging into the application would require the user to be in possession of the mobile phone number that they have specified in order to retrieve the OTP.

In addition, to keep the data of each user confidential, the users' data is only visible by the users themselves, researchers and the system administrator only. Finally, to keep the data secure, we will prevent SQL injection by sanitizing inputs and preventing malicious users from causing unexpected behavior.

Overall usability

The five main characteristics of usability: effectiveness, efficiency, engaginess, error tolerance and ease of learning, will be incorporated into our software. These can all be measured and tested for using industrial standard metrics. They are documented below:

https://www.interaction-design.org/literature/topics/usability

Effectiveness—It supports users in completing actions accurately

Efficiency—Users can perform tasks quickly through the easiest process.

Engagement—Users find it pleasant to use and appropriate for its industry/topic.

Error Tolerance—It supports a range of user actions and only shows an error in genuine erroneous situations. You achieve this by finding out the number, type and severity of common errors users make, as well as how easily users can recover from those errors.

Ease of Learning—New users can accomplish goals easily and even more easily on future visits

Name	**Scale**	**Meter**	**Target**	**Constraint**
Effectiveness	Relevance of the recommendations	Users' interest is gathered by the chatbot by using the survey for first-time logins. Users are then shown content recommendations based on the data gathered from the survey. They are then required to give an integer feedback score (out of 10) to rate the relevance of the recommendation(s) given by the chatbot.	A composite feedback score of 6 out of 10.	The composite feedback score should not be lower than 4 out of 10.
Efficiency	Response time of the application	Users navigate through the whole application and the time taken for the application to take the user to another page is recorded.	Navigation time should be less than 1000ms.	Navigation time should not be more than 4000ms.
Engagement	Satisfaction of users	Users are given random tasks to be performed on the application. They are then required to give an integer feedback score (out of 10) to rate the application's intuitiveness and ease of use.	A composite feedback score of 6 out of 10	The composite feedback score should not be lower than 4 out of 10.
Error Tolerance	Display of an error message prompt that is easy to understand for a user	A user experiences some errors during a navigation of the application. For example, when a user creates a post with title exceeding the word length, a user should be warned for the limitation.	Error pages behind the framework or database should not be redirected to the user. It should create an error message to the user instead.	The user should never be left unaware of the problem that has occurred.
Ease of Learning	Ease of use as rated by users	A new user is given an account and is asked to perform a list of tasks on the application. The time for them to complete the task is then recorded and they are asked to rate how difficult it was for them to perform the task(s)	A target composite feedback score of 7 out of 10 (has to be higher to accommodate for the elderly)	The feedback score should not fall lower than 5 out of 10

Accessibility

To start off, our application allows users to interact by using their voice. This allows users that are unfamiliar with using keyboards to interact with the application with ease. In addition, voice input also allows blind or partially sighted users to be able to interact with the application without the need for external assistance.

Next off, our application caters for the elderly with poor eyesight by having large word font options and intuitive designs for UI elements. Not only that, users are able to use the application in a variety of different languages, depending on what they’re preferences are. They include languages such as English, Malay, Chinese and Thai. Multilingual support in turn would allow a larger group of individuals to become potential users.