Sprint 2 - Development Planning

This document is the plan and progress log of this project. It will be updated as the project progresses.

Driver	@Dongning Li
Approver	@xueshuo li
Contributors	@Xueting ZHAO @Nanyu Chen @Elena Li @Shulei He
Informed	@Larissa Salerno de Castro
Objective	Framework setup, All test pages and Tests
Due date	6 May 2024
Key outcomes	Workflow, GitHub Workflow, UI Design, Login Page, All Test Page, Test Content Page, Generate Keys for Answer, Tests
Status	COMPLETE

Schedule Statement

- We conducted our sprint planning meeting on Monday of the first week of this sprint. Given its four-week length, we have scheduled a "mini sprint planning" meeting every Monday. In these meetings, we add, re-estimate, and update tasks from the backlog, if any.
- With six team members, we estimate around 200 story points for the whole sprint, which allocates less than 10 user story points for each team member per week.

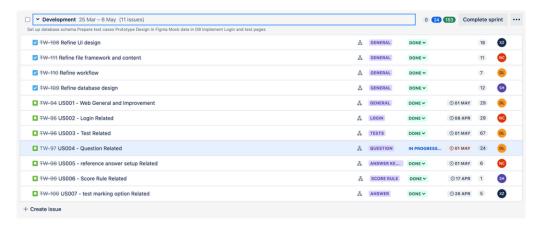
Problem Statement

- · Working together might cause extra time for peer review. However, this will guarantee code efficiency and cleanliness
- · Working remotely might cause an inconsistent understanding of requirements. We think documents and logs are good approaches
- The requirements from the client are not clear, so we have to allocate more time to requirement analysis

Scope

Must have:	 Finalize workflow Finalize Github merge workflow [US001]: Finalize UI [US002]: Implement login page [US003]: Integrate all functionalities for all test page [US004 - US007]: Test list and content: [US004 - US007] Implement all test functionalities
Nice to have:	 [US001.T1]: Create reusable components [US002.T7 - T9]: Update login page to UI (2nd version) [US004.T4- T5]: Reorder test
Not in scope:	Deployment

Sprint Overview



Sprint Backlog

U S ID	Task ID	Description	P oi nt s	Pri orit y	Acceptance Criteria	Re qui red	Progres s (Red: Delayed)	Links	Wee k	Due Day
U	5001	As a test developer, I want a user-friendly, secure, functional interface.	29	Me diu m	 A user-friendly, secure ,and functional interface Tests are applied and passed 	Yes	Done	TW-94: US001 - Web General a nd Improvement	Wee k 1-4	1 May 2024
	US0 01.T 1	Create a reusable modal component	5	Me diu m	Reusable and suited for different models	No	Done	Code review log: PR #20: Story/ tw-90	Wee k 3-4	
	US0 01.T 2	Test implemented functionalities	8	Me diu m	 Implemented functionalities have been tested Test results have been logged 	Yes	Done	Test cases: ■ Sp rint 2 - Test Case s	Wee k 1-4	
	US0 01.T 3	UI design modification to fit new requirements regarding setting multiple keys for a question	3	Hig h	Design UI with multiple keysMore options for test configuration	Yes	Done	UI: UI Design	Wee k3	
	US0 01.T 4	Database design modification to fit new requirements regarding	3	Hig h	Multiple keys for one testTest has a title	Yes	Done		Wee k 4	

	a question								
US0 01.T 5	Backend api modification to fit new requirements regarding setting multiple keys for a question	5	Hig h	Send a request with multiple keys and title of question	Yes	Done	PR: Feature/tw 141 test api modification	Wee k4	
US0 01.T 6	Test content UI modification to fit new requirements regarding setting multiple keys for a question	5	Hig h	Multiple keys could be shown with mock data	Yes	Done	■ PR#29: Story/t w-140	Wee k 4	
US002	As a test developer, I want to implement a secure login system and a reliable database, so that I can securely store and protect user data and system information.	29	Me diu m	Database have different mock data for functionalities Login UI is implemented as in Figma User login Successfully login Check user information before login, show error if invalid After login, navigate to choose-role page Fail login with a warning Navigate to different URL when choose different role	Yes	Done	UI: UI Design The state of the	Wee k 1-5	8 Apr 2024
US0 02.T 1	Make sure our databse have some mock user data for admin, test developer, marker	1	Me diu m	Data includes test, question and users	Yes	Done		Wee k1	
US0 02.T 2	Init a basic project file structure	1	Me diu m	Work with demo	Yes	Done		Wee k1	
US0 02.T 3	Add notification for an unauthorized page	1	Me diu m	Show unauthorized page	Yes	Done		Wee k1	
US0 02.T 4	login UI: input bar for username and password	5	Me diu m	Login UI is implemented as in Figma	Yes	Done	GitHub: PR: Feature/tw 63 login UI #6	Wee k 2	

					 User login Successfully login Check user information before login, show error if invalid 					
	US0 02.T 5	Send and receive message to/from backend	5	Me diu m	 Integrate with backend API Replace mock data with real data from database 	Yes	Done	Code review log: PR #7: Story/t w-65	Wee k 2	
	US0 02.T 6	Login auth	5	Me diu m	 After login, navigate to choose-role page Fail login with a warning Navigate to different URL when choose different role 	Yes	Done	GitHub: PR: Refactor/tw 73 refactor login feature	Wee k 2	
	US0 02.T 7	Refactor login feature	5	Me diu m	Maintain all functionalities working as before	No	Done		Wee k 2	
	US0 02.T 8	Login page UI beautify	3	Me diu m	Login page corresponds to new UI	No	Done	PR #34: Story/	Wee k 3-5	
	US0 02.T 9	Choose-role page beautify	3	Me diu m	Page corresponds to new UI	No	Done		Wee k 3-5	
U	6003	As a test develop, I want to construct, preserve, and eliminate tests as needed, so that I can efficiently manage the lifecycle of tests.	67	Me diu m	Test UI is implemented as in Figma`Version 2` Header, footer Test list Icons Show test list with real data Each test can be deleted/duplicated/added at any position in the list Add new test by uploading Excel with real data	Yes	Done	UI: UI Design	Wee k 2-5	1 May 2024

US0 03.T 1	Test-developer page layout	3	Me diu m	 Page is divided into different components and is suitable for test list, content, header, and footer 	Yes	Done	GitHub: https:// github.com/COM P90082-2024-S M1/TT-Scoring-F rontEnd/pull/15 Connect your Github account	k 2
US0 03.T 2	Test-developer page header	5	Me diu m	Header is implemented as Figma	Yes	Done	Code review log: PR #16: Story/ tw-79	Wee
US0 03.T 3	Test list-related UI	3	Me diu m	Test list UI is shown as Figma with mock data	Yes	Done	Code review log: PR #17: Story/ tw-91	Wee
US0 03.T 4	Copy a test api	5	Me diu m	Test can be copied when sending a request	Yes	Done	PR: Feature/create api endpoints for saving, deleting, updating and copying tests	Wee
US0 03.T 5	Delete a test api	3	Me diu m	Test can be deleted when sending a request	Yes	Done	PR: Feature/create api endpoints for saving, deleting, updating and copying tests	Week 3
US0 03.T 6	Save a test api	8	Me diu m	Test can be saved when sending a request	Yes	Done	PR: Feature/create api endpoints for saving, deleting, updating and copying tests	Week 3
US0 03.T 7	Fetch test list api	3	Me diu m	 Test list can be provided when sending a request Title, ID, key 	Yes	Done	PR: Feature/create api endpoints for saving, deleting, updating and copying tests	Week 3
US0 03.T 8	Test footer UI implementation	3	Me diu m	Footer is implemented as Figma	Yes	Done	GitHub: https:// github.com/COM P90082-2024-S M1/TT-Scoring-F rontEnd/pull/23 Connect your Github account	Wee
US0 03.T 9	Integrate fetch test list api	3	Me diu m	Test list is implemented with	Yes	Done	GitHub:	Wee

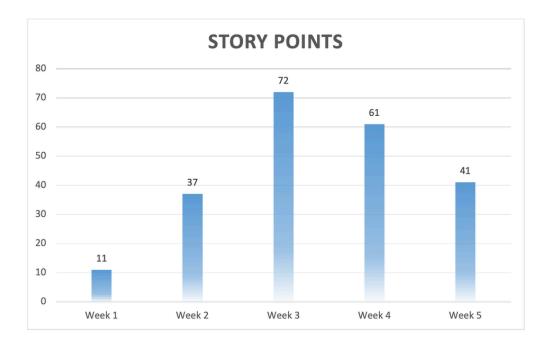
				real data from database			M1/TT-Scoring-F rontEnd/pull/24 Connect your Github account		
US0 03.T 9	Bug fix ticket for US003.T9	2	Hig h	Change auto mark UI	No	Done		Wee k 3	
US0 03.T 10	New Test Button UI implementation	5	Me diu m	Button position and style as shown in Figma	Yes	Done		Wee k 3-4	
US0 03.T 11	New Test by loading Excel UI	3	Me diu m	When clicking new test button, pop-up window should be implemented as Figma, with mock data A pop-up window Go back to the previous page after closing	Yes	Done	■ PR #32: Story/ tw-128	Wee k 3-4	
US0 03.T 12	Integrate save test api	5	Me diu m	When clicking the save test button, the test could be saved into database	Yes	Done	■ PR #37: Story/ tw-103	Wee k 4-5	
US0 03.T 13	Integrate delete test api	3	Me diu m	When click deleting button, test could be deleted into database	Yes	Done	https://github.c om/COMP90082- 2024-SM1/TT-Sc oring-FrontEnd/p ull/36 Connect your Github account	Wee k 4-5	
US0 03.T 14	Create test by loading an excel api	5	Me diu m	Click new test button, excel can be chosen and uploaded to web	Yes	Done		Wee k 4-5	
US0 03.T 15	Integrate create test by loading an Excel api	3	Me diu m	Create a new test when sending request	Yes	Done		Wee k 4-5	
US0 03.T 16	Integrate copy test api	5	Me diu m	Test with real data could be created/ copied to database Return successful/ error message	Yes	Done	■ PR #35: Story/ tw-105	Wee k 4-5	
US004	As a test developer, I want to have the ability	24	Me diu	Question could be created/ updated/	Yes	In Progres		Wee k 3-4	1 l 20

		to add and delete items (questions) within existing tests to adapt the tests as needed.		m	deleted UI is shown as designed		s 16/24	TW-97: US004 - Question Relat ed DONE		
	US0 04.T 1	Test content-related UI and interaction logic	8	Me diu m	user can add, edit, and delete questions in the currently selected test Except mark being number field, other fields are textInput	Yes	Done	Code review log: PR #19: Story/ tw 92	Wee k3	
	US0 04.T 2	Fetch a test content api	5	Me diu m	Test content could be fetched when sending a request	Yes	Done		Wee k 2-3	
	US0 04.T 3	Integrate fetch test content api	3	Me diu m	Test content could be sent when sending a request	Yes	Done	GitHub: https:// github.com/COM P90082-2024-S M1/TT-Scoring-F rontEnd/pull/26 Connect your Github account	Wee k3	
	US0 04.T 4	Enable user to reorder questions by drag and drop	5	Me diu m	Question could be reordered by dragging	No	TO DO		Wee k 5	
	US0 04.T 5	Enable user to insert a new question in the question list	3	Me diu m	Question could be inserted at any position in the list	No	TO DO		Wee k 5	
Us	S005	As a test developer, I want to have the ability to add, edit, and delete test keys (reference answers) to ensure the validity of the tests we develop.	6	Hig h	 Key for question could be added/ updated/ deleted Alternative key should be generated from backend and provided when request When clicking button, receive alternative keys and applied to page 	Yes	Done	TW-98: US005 - reference answ er setup Related	Wee k 3-4	1 May 2024

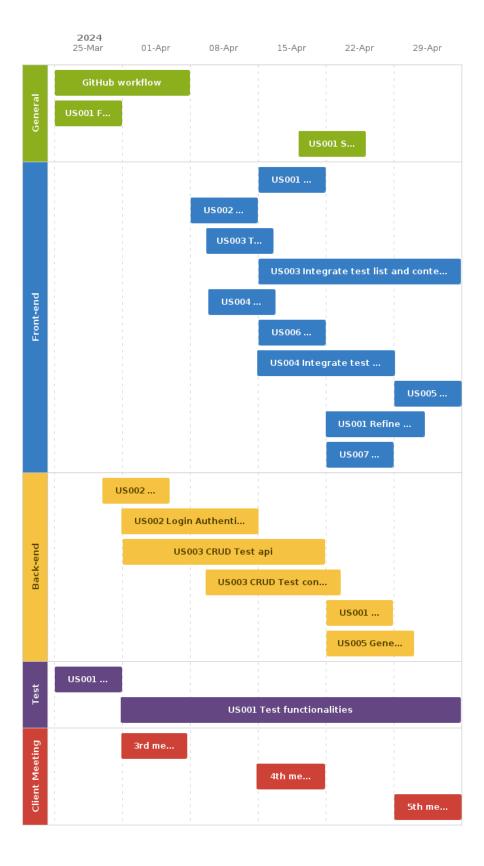
	US0 05.T 1	Generate alternative key options api	3	Me diu m	 Key for question could be added An alternative key should be generated from backend and provided when request 	Yes	Done	PR: Feature/tw 98 generate alternative key options api	Wee k 5	
	US0 05.T 2	Integrate generate alternative keys api	3	Me diu m	When clicking button, receive alternative keys and applied to page UI is implemented in a pop-up window	Yes	Done	■ PR#39: Story/t w- 98	Wee k 5	
l	JS006	As a test developer, I want to have flexibility in defining scoring rules, so that I can specify whether the scoring system is dichotomous (e.g., correct/incorrect) or allows partial scoring.	1	Me diu m	Score rules could be set as binary or partial scoring UI shown as designed Score and score rule is saved and fetch from DB	Yes	Done	TW-99: US006 - Score Rule Rel ated Done	Wee k 2-4	17 Apr 2024
	US0 06.T 1	Allow user to set score level for a question	1	Me diu m	Set a score for binary question in UI	Yes	Done	Code review log: PR #19: Story/ tw 92	Wee k 2	
l	JS007	As a test developer, I want to have options to handle different types of answers: Ignore case sensitivity, flag answers for manual marking if partially matched and handle spelling mistakes or minor grammatical errors.	5	Me diu m	 Test configuration includes title and other manual settings Configuration should be saved/ edited into DB 	Yes	Done	To a promote the second	Wee k4	26 Apr 2024
	US0 07.T 1	Implementation of a test config modal UI	5	Me diu m	Test configuration UI is shown as designed	Yes	Done	■ PR #28: Story/ tw-126	Wee k 4	

Weekly Distribution

• A distribution of user stories, taking into account that some may span multiple weeks



Timeline



Milestone	Owner	Deadline	Status
GitHub merge workflow	@xueshuo li @Dongning Li	29 Mar 2024	DONE
The initial version of UI design (under US001) is delivered	@Xueting ZHAO	1 Apr 2024	DONE
Login, authentication, authorisation (under US002) is delivered	@xueshuo li @Shulei He @Nanyu Chen @Xueting ZHAO	8 Apr 2024	DONE
Test-related UI implementation (under US003) is delivered	@Elena Li @Dongning Li @Shulei He @xueshuo li	25 Apr 2024	DONE
Question-related UI implementation, no reordering required, (under US004) is delivered	@Elena Li @Shulei He @xueshuo li	17 Apr 2024	DONE
set reference answers implementations (US005) is delivered	@Shulei He @Elena Li	1 May 2024	DONE
Scoring-setting for questions (under US006) implementation is delivered	@Elena Li	17 Apr 2024	DONE
Test option configuration implementation (under US007) is delivered	@xueshuo li @Shulei He	1 May 2024	DONE

Reference materials

Backlog Details in JIRA

TT-Wombat