

Meeting Minutes - 15

Meeting Agenda

Date and Time

Date: 14 October 2023

Time: 9:30pm - 11:30pm

Location

Discord Call

Participants

- Brian Nge Jing Hong
- Chua Xian Loong
- Diana Wijaya
- Koe Rui En
- Lucas Wee
- Muhammad Ibrahim bin Mohd Yusni

Agenda Details

Meeting Objectives

1. Conducting Sprint Review planning
2. Film Project Demonstration
3. Conducting Sprint Retrospective

Pre-meeting Preparations

-

Agenda Details

Time	Content
30 minutes	<div>Sprint Review planning<ul style="list-style-type: none">- Inspect the product- Analyse on how done the user stories for the sprint are- Write the planning for the demonstration before the actual video demonstration</div>

40 minutes	Recording for the demo video <ul style="list-style-type: none"> - Record demonstration of the final product from the first sprint via Zoom
50 minutes	Sprint Retrospective <ul style="list-style-type: none"> - What went well? - What were the problems encountered? - What could have been done better? - What will we try next? - What questions do we have?
Total meeting time: 120 minutes	

Meeting Minutes

Attendance

Group member	Present	Absent
Brian Nge Jing Hong	✓	
Chua Xian Loong	✓	
Diana Wijaya	✓	
Koe Rui En	✓	
Lucas Wee	✓	
Muhammad Ibrahim Bin Mohd Yusni	✓	

Apologies

-

Minute Taker and Time Keeper

Minute taker: Rui En

Time Keeper: Rui En

Meeting Summary from Previous Meeting

Our team organised a stand-up meeting on Saturday. During the meeting, we provided updates on each team member's work progress.

Agenda

Conducting Sprint Review planning

Inspect the product

All the user acceptance criteria for each user story are confirmed to be done on time by all team members as shown below.

☒ User Acceptance Criteria Hide checked items Delete

100%

☒ Stored in database correctly

☒ Edit button shown

☒ Show name and email correctly

☒ update username after edit

☒ Remove user and related tasks

☒ Add new user

☒ Allow user to change password

Add an item

All the tasks allocated in each user story are done on time by all team members as shown below.

☒ Task Allocations Hide checked items Delete

100%

☒ Frontend: Add button to change password in user information page
—Brian

☒ Frontend: Add button to edit in all users page —Brian

☒ Backend: Change password and edit feature —Lucas

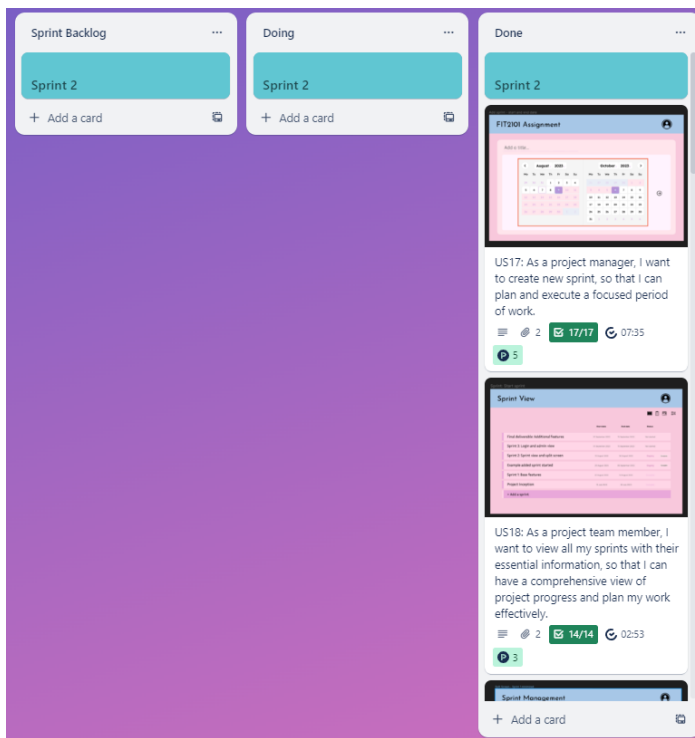
☒ Backend: Render the updated users in the task pages —Ibrahim

☒ Backend: Added buttons to add new users and remove existing
ones—Ibrahim

Add an item

Analyse on how “done” the user stories for the sprint are

It is confirmed that all the user stories for the third sprint have been completed. As shown below, all the tasks have been moved into the done section, and there are no more user stories in the sprint backlog and doing sections.



Planning

The team conducted a sprint review planning before beginning to record a video demonstration of our team's progress and working software for Sprint 3. During the discussion, our team decided on their roles for the video:

- Brian introduced the team, sprint goal and project information. He would also briefly go through the user information page.
- Ibrahim briefly showed the sprint backlog and user stories in Trello. He would also briefly go through the software features from sprint 1 and 2 as well as wrapping up the demonstration.
- Lucas demonstrated the individual contribution functionality along with the graph function as well as the edit and remove functionality.
- Diana introduced the change password feature and the individual contributions page.
- Xian Loong demonstrated the overall contribution functionality which includes the calendar page.
- Rui En demonstrated the functionality that limits the number of sprints in progress to only one.

Film Project Demonstration

Our team decided on the platform to record our demonstration. After that, we temporarily shifted our meeting to the Zoom platform for recording purposes. We had a rehearsal before commencing the actual video demonstration. After recording, Lucas shared the video in the group for all members to review before creating an unlisted Youtube video.

Sprint Retrospective

- What went well?
- What were the problems encountered?
- What could have been done better?
- What will we try next?
- What questions do we have?

What went well?

Sprint Meetings and Accountability:

- Increased the number of standup meetings in this sprint compared to the previous 2.
- Enhanced accountability and ensured task completion on time.
- Frequent updates and task awareness among team members.

Efficient Workflow:

- Proactively added new task allocations to user stories.
- Forward-thinking approach covered all aspects of user stories comprehensively.
- Efficient workflow as a result of these measures.

Task Allocation Based on Skills and Interests:

- Task allocation based on members' skills and interests enhances member satisfaction and smooth progress.
- Tasks are completed on time, leading to fast progress.

Increased Communication and Updates:

- Held more meetings, resulting in frequent task updates.
- Improved team understanding of ongoing work.
- Smoother and more efficient work processes.

Integration Testing and Quality Improvement:

- Integration testing performed at an earlier stage.
- Detection of errors in sprint 3 to improve software quality.
- Addressing bugs earlier in the development cycle.

Ahead of Schedule Task Completion:

- Team completed all sprint tasks ahead of schedule.
- Allowed for additional integration testing to find and fix bugs earlier.

What were the problems encountered?

Skill and Knowledge Gaps:

- Problem with understanding the required content to code the website.
- Lack of prior knowledge in JavaScript, HTML, and CSS, leading to high difficulty in coding and understanding the required languages.
- Improvement due to experience gained from previous sprints and online resource utilisation.

Coding and Bug Issues:

- Bugs causing 404 errors due to code and hosting platform inconsistencies.
- Hosting platform inconsistencies cause navigation problems for users trying to access specific pages.
- Timely identification and resolution of these issues by the team.

Specification and Communication Problems:

- Unclear specifications leading to confusion and time loss.
- Changes in user acceptance criteria and features due to task misunderstanding.
- Conflicts among team members during discussions.

Complex Codebase Challenges:

- The complexity of code functions are increasing and understanding becomes more time-consuming.
- There are intricate interdependencies within the codebase.
- Difficulty in pinpointing the exact source of problems.

Integration Testing Challenges:

- Lack of familiarity with integration testing.
- There was a need to understand all function files prior to testing, which can be time-consuming.
- Overcoming this challenge by seeking help from team members.

Documentation Issues:

- Incomplete or unclear documentation for completed tasks.
- There are delays when building upon existing work due to the lack of documentation.

What could have been done better?

Software Testing and Automation Testing:

- Suggests spending more time on testing to identify and discover bugs effectively.
- Advocates for multiple iterations of testing to find and resolve more bugs.
- Emphasises early bug detection and resolution for more efficient progress.
- Encourages the use of automation testing to detect and prevent sudden, undetected bugs.
- Benefits include running numerous test cases quickly and obtaining early feedback

Deployment and Database Environment:

- Improve practice of promoting understanding of deployment and database environments.
- Aims to reduce dependency on a few experts within the team.

Client Communication:

- Recommends seeking more information from the client before the sprint to understand tasks better.
- Strives for a clearer understanding of tasks to start work with fewer obstacles.

Clear Communication:

- Highlights the need for better communication about how to change or fix functions when addressing bugs or errors.
- Addresses misunderstandings within the team leading to delays and suboptimal fixes.

Documentation Responsibility:

- Suggests designating a team member each sprint to oversee and ensure thorough code documentation.
- Stresses the importance of consistent and comprehensive documentation throughout the project for better team understanding and future developer support.

What will we try next?

Preparation and Self-Learning:

- Emphasises preparing in advance through self-learning before the sprint.
- Aims to increase efficiency by having prior knowledge and preventing bugs.

Incremental Testing of Deployment:

- Proposes implementing incremental testing of deployment for early issue detection.
- Ensures compatibility with the web hosting service and smoother deployments.

Client Interviews for Information:

- Suggests conducting client interviews at the beginning of the sprint.
- Aims to clarify necessary information early, prevent redundancy, and provide clear instructions.

Coding Meetings for Issue Resolution:

- Recommends having more coding meetings to tackle issues and bugs.
- Creates a proactive forum for sharing insights and brainstorming solutions.

Automation Testing:

- Advocates for automation testing to detect more bugs and increase efficiency.
- Reduces the risk of human error common in manual testing.

Bi-Weekly Code Review Sessions:

- Proposes organising bi-weekly code review sessions.
- Aims to maintain consistency in coding practices, catch potential issues early, and promote continuous learning and skill enhancement within the team.

What questions do we have?

1. How can we be more efficient during the sprint and what additional tools can we incorporate into our practice that would improve our overall productivity?
2. What indicators should we be aware of to know if we're moving in the right direction?
3. How can we make sure that all the tasks that will be done completely suit the client's needs?
4. How can we ask more questions about the tasks that needed to be done to ensure clarity on the given task to have higher efficiency?
5. How can we deploy our software and ensure its maintenance post-deployment?
6. How can we maintain a balance between delivering features and ensuring the quality of the code?