Taskforce

In-house Project Management Software

Sprint Two

Version 1.0

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

Taskforce

Contents

1. Sprint Planning

1.1 Important Links

Link to Project Inception: Project Inception V2.0

Link to Trello board: https://trello.com/invite/b/XmURix1g/ATTIcca68202ded99301ef54aa2fd5dc1e114A503979/project-1

Link to the meeting minutes: Meeting 4 - 16/9/2023

Link to Refined User Stories: User Stories

Link to Task Allocation: Task Allocations

1.2 Product Backlog

The product backlog consists of all the user stories that the team has decided for the project. The product backlog is logged on Trello. The link to the Trello board is provided in section 1.1. To decide on the user stories as well as its story points, a meeting was held on 16 September 2023, Saturday from 10:00 PM to 12:00 AM. The meeting minutes for the meeting are accessible through the link provided in section 1.1. It is confirmed that the product backlog follows the DEEP criteria. Additionally, the user stories are refined according to the INVEST criteria before placing it into Trello. The refined user stories can also be accessible through the spreadsheet link in section 1.1.

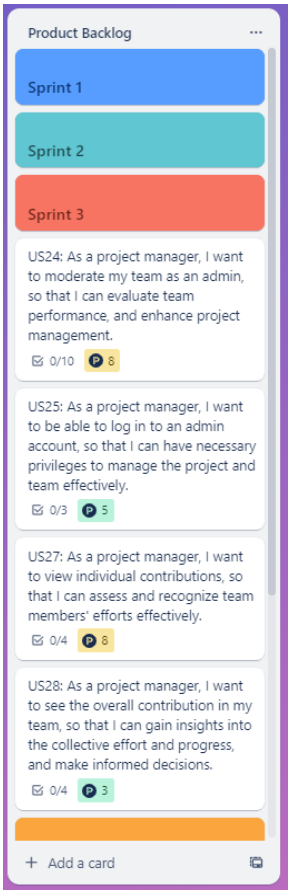
Several spike stories are also added in Trello, for team members who are not familiar with the programming languages to work on their skill sets while working on the project. Each spike story includes a link to a youtube video that is sufficient for team members who are not familiar with the programming language to gain necessary knowledge for the project. Other than that, links to online resources and courses are also included. Team members who were not familiar with any programming languages used in the project, or aspects needed to complete the project, assigned themselves with a task, and also logged their time spent during the process.

Access to essential links such as the UI/UX design decisions and the web application prototype are also attached in trello, allowing easier access to the important links for team members.

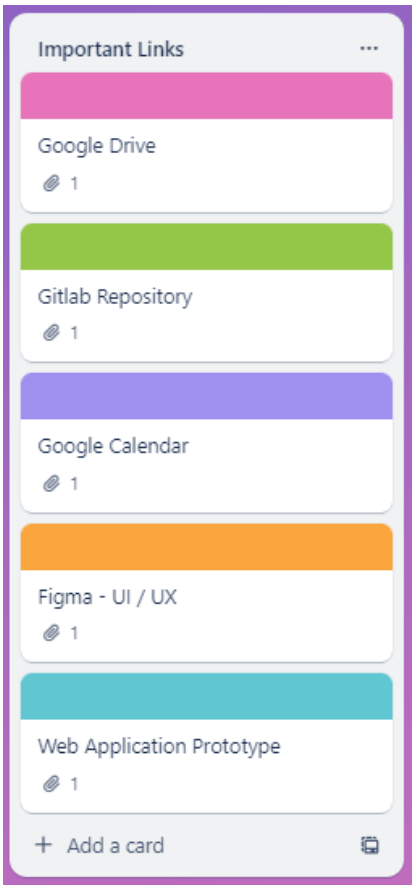
Some documentation examples for the product backlog and spike stories are attached below.

Taskforce

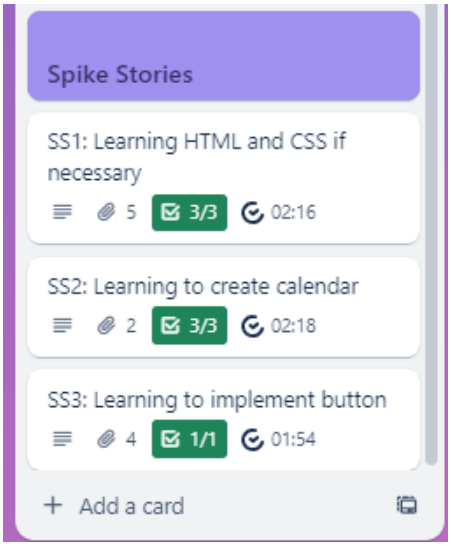
Product backlog:

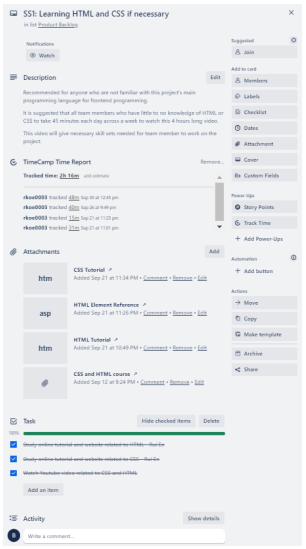


Essential links:



Spike stories:

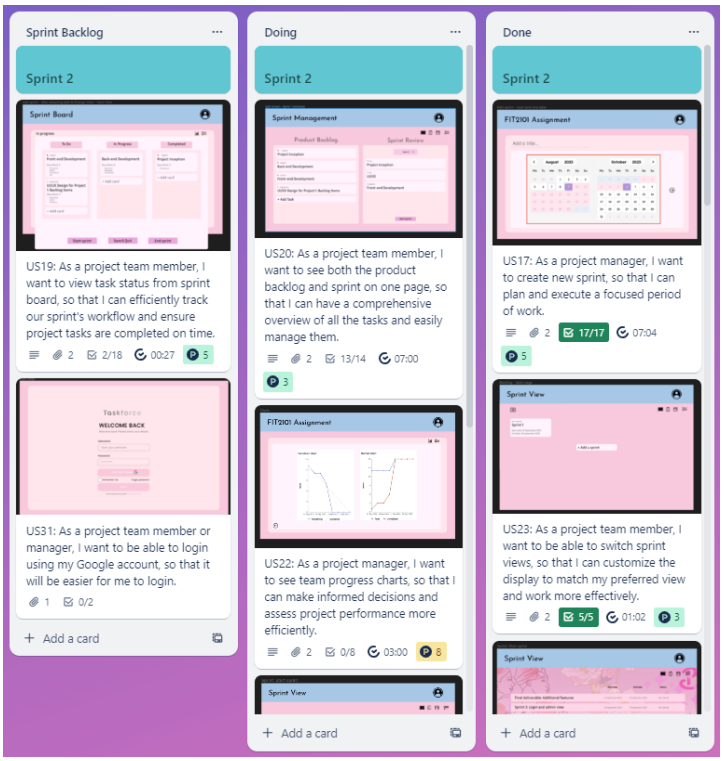




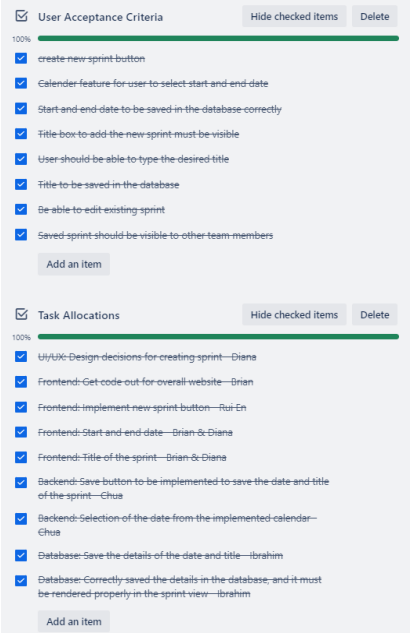
1.3 Sprint backlog

The sprint backlog is logged into Trello, provided link is in section 1.1. All the user stories for the second sprint are decided during the first meeting after the second sprint started, as shown in meeting minute 4, attached in section 1.1. The story point for each user story was decided through planning poker, where each team member will decide on the story point individually, then discussions were raised when there were wide spreads of story point choices, ultimately resulting in one agreed story point per user story. The User Acceptance Criteria is included under each individual task, showing the expectations of each user story with clear definition and standard. The User Acceptance Criterias were later used for testing. The story point is clearly indicated under each user story in the sprint backlog. We have also discussed the task allocation and each team member has assigned themselves to their respective tasks. The link to the summary of task allocations is attached in section 1.1.

Sprint Backlog:



User Acceptance Criteria and Tasks:



1.4 Sprint Goal

The sprint goal is to implement sprint view and sprint management where tasks from product backlog can be added to the sprint backlog.

This sprint goal was decided during the meeting held on 16 September 2023, Saturday from 10:00 PM to 12:00 AM. To achieve this sprint goal, we have come up with extensive planning and task allocation throughout the team. The details of the task allocation can be found on page 3 of the meeting minutes for 16 September 2023.

1.5 Project Inception

As attached in section 1.1, the project inception was updated accordingly, where the details for testing are now included. Aside from that, a tester was assigned to one of our team members. The necessary changes for testing were then implemented, as shown in section 3.

2. Daily Scrum

2.1 Important Links

Link to Trello board:
https://trello.com/invite/b/XmURix1g/ATTIcca68202ded99301ef54aa2fd5dc1e114A503979/project-1

Link to the meeting minutes: Meeting 5 - 21/9/2023 Meeting 6 - 26/9/2023 Meeting 7 - 30/09/2023

Link to the Web Application Prototype: https://taskforce.pages.dev/login
Credentials to Login:

- Email: admin@gmail.com
- Password: 123456

Link to GitLab repository:
https://git.infotech.monash.edu/FIT2101-S2-2023/MA_Thursday11am_Team1/project

2.2 Weekly Stand-Up Meeting

During the second sprint, the team conducted 3 standup meetings in total, on the 21st, 26th, and 30th of September. The meeting minutes for these meetings are attached in section 2.1.

During the standup meetings, each team member spoke on the following points:

- What has been accomplished since the last meeting?
- What will be done before the next meeting?
- What obstacles are in the way?

The team had the first stand-up meeting on 21 September 2023, from 01:00 PM - 02:00 PM. The link to this meeting minutes can be accessible via the link in section 2.1. With this, each team member is able to understand what tasks each team member is working on. During this meeting, the team has discussed the 3 non-functional requirements to be presented to the client for them to choose as per their preference. The client was then presented with the list of non-functional requirements and she has chosen usability as the selected non-functional requirement to be implemented during sprint three. She has also chosen website background as the human aspect that should be implemented during the current sprint.

On the 26th of September, the group held the second standup meeting, where each member updated their progress, what will be done before the next meeting, and the obstacles along the way. During this meeting, two more user stories were added and the story point was decided. The user acceptance criteria were also decided, and tasks were allocated accordingly to each member's suitability.

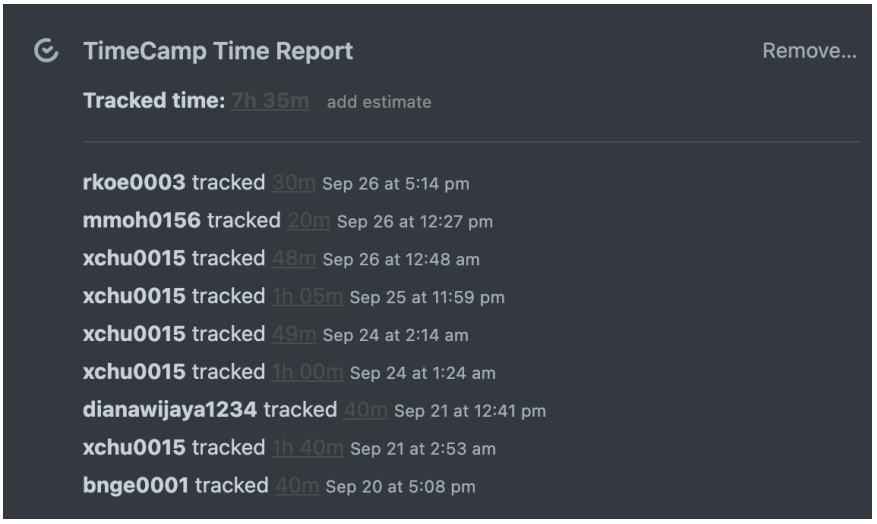
We had the last standup meeting for the sprint on the 30th of September, the meeting agenda is attached in section 2.1. During this meeting, everyone updated on their progress of work. Besides that, the team made sure that all the tasks were completed, and the project is ready to deploy.

We have also utilised the group chats to keep each other updated on current task status. Besides, we have also iterated the importance of the git policy that we have outlined in your project plan and made sure to only push to the “Firebase” branch and not the main branch.This is to prevent confusion of the code and ensure that everyone is able to work on their end without much merge conflicts. It is also a good industry practice to work on a branch rather than on the main repository.

Throughout the standup meetings, the team was able to complete the following user stories. Further details on the user stories can be found in the trello link provided in section 2.1.

1. US17: As a project manager, I want to create new sprint, so that I can plan and execute a focused period of work.
2. US18: As a project team member, I want to view all my sprints with their essential information, so that I can have a comprehensive view of project progress and plan my work effectively.
3. US19: As a project team member, I want to view a sprint board, so that I can efficiently track our sprint's workflow and ensure project tasks are completed on time.
4. US20: As a project team member, I want to see both the product backlog and sprint on one page, so that I can have a comprehensive overview of all the tasks and easily manage them.
5. US21: As a project manager, I want to end sprints, for better progress tracking and project planning.
6. US22: As a project manager, I want to see team progress charts, so that I can make informed decisions and assess project performance more efficiently.
7. US23: As a project team member, I want to be able to switch sprint views, so that I can customise the display to match my preferred view and work more effectively.
8. US30: As a project manager, I want the software to look more feminine, so that we can celebrate women in project management more significantly.
9. US31: As a project team member or manager, I want to be able to login using my Google account, so that it will be easier for me to login.
10. US32: I want to use a navigation bar in the app, so that I can use the different functionalities in the software easily.

The time spent spent by each team member on each task is also logged and kept track of, via a Trello power up that we used, “TimeCamp”, a screenshot of an example of the time tracked is attached below.



This is the time spent for each team member and individual time allocated on each user story:

US17: As a project manager, I want to create new sprint, so that I can plan and execute a focused period of work.

1. Brian Nge Jing Hong - 40m
2. Muhammad Ibrahim bin Mohd Yusni - 20m
3. Chua Xian Loong - 5h 24m
4. Lucas Wee - 30m
5. Diana Wijaya - 40m

Total: 7h 35m

US18: As a project team member, I want to view all my sprints with their essential information, so that I can have a comprehensive view of project progress and plan my work effectively.

1. Brian Nge Jing Hong - 3m
2. Muhammad Ibrahim bin Mohd Yusni - 50m
3. Lucas Wee - 1h 30m
4. Diana Wijaya - 30m

Total: 2h 53m

US19: As a project team member, I want to view task status from sprint board, so that I can efficiently track our sprint's workflow and ensure project tasks are completed on time.

1. Brian Nge Jing Hong - 27m
2. Muhammad Ibrahim bin Mohd Yusni - 1h 30m
3. Diana Wijaya - 30m

Total: 3h 27m

US20: As a project team member, I want to see both the product backlog and sprint on one page, so that I can have a comprehensive overview of all the tasks and easily manage them.

1. Muhammad Ibrahim bin Mohd Yusni - 3h 30m
2. Diana Wijaya - 3h 30m

Total: 7h

US21: As a project manager, I want to end sprints, for better progress tracking and project planning.

1. Muhammad Ibrahim bin Mohd Yusni - 30m

Total: 30m

US22: As a project manager, I want to see team progress charts, so that I can make informed decisions and assess project performance more efficiently.

1. Brian Nge Jing Hong - 3h
2. Lucas Wee - 3h

Total: 6h

US23: As a project team member, I want to be able to switch sprint views, so that I can customize the display to match my preferred view and work more effectively.

1. Brian Nge Jing Hong - 17m
2. Muhammad Ibrahim bin Mohd Yusni - 10m
3. Diana Wijaya - 30m

Total: 57m

US30: As a project manager, I want the software to look more feminine, so that we can celebrate women in project management more significantly.

1. Brian Nge Jing Hong - 30m
2. Muhammad Ibrahim bin Mohd Yusni - 10m

Total: 40m

US31: As a project team member or manager, I want to be able to login using my Google account, so that it will be easier for me to login.

1. Lucas Wee - 30m
2. Diana Wijaya - 15m

Total: 45m

US32: I want to use a navigation bar in the app, so that I can use the different functionalities in the software easily.

1. Chua Xian Loong - 1h 31m
2. Koe Rui En - 20m

Total: 1h 51m

Summary:

1. Brian Nge Jing Hong - 5h 30m
2. Muhammad Ibrahim bin Mohd Yusni - 7h 01m
3. Chua Xian Loong - 7h 55m
4. Lucas Wee - 7h
5. Koe Rui En - 5h 01m
6. Diana Wijaya - 6h 43m

3. Sprint Review

3.1 Important Links

Link to Trello board:
https://trello.com/invite/b/XmURix1g/ATTIcca68202ded99301ef54aa2fd5dc1e114A503979/project-1

Link to the meeting minutes: Meeting 8 - 30/09/2023

Link to the Web Application Prototype: https://taskforce.pages.dev/login
Credentials to Login:

- Email: admin@gmail.com
- Password: 123456

Link to the Demo Video: https://youtu.be/YgwI3MkWq6o

Link to Full Live and Updated Risk Register: Risk Register Updates - Sprint 2

Link to GitLab repository:
https://git.infotech.monash.edu/FIT2101-S2-2023/MA_Thursday11am_Team1/project

Link to the testing report: Testing Report

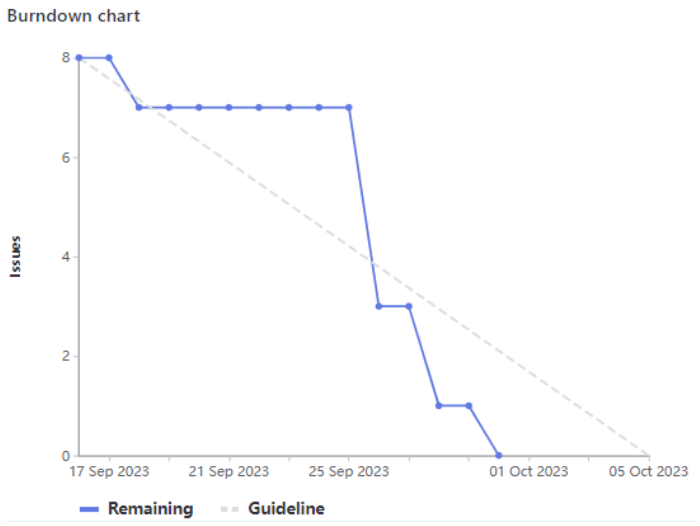
3.2 Burndown Chart

A burndown chart is a visual tool used in project management, specifically in Agile and Scrum methodologies, to track the progress of work over time. It helps teams and stakeholders monitor how tasks or user stories are being completed and whether the project is on track to meet its goals and deadlines. The team has utilised GitLab for its burndown chart feature. The user stories from Trello were placed into GitLab and issues were closed when we implemented that feature in the web application. The burndown chart and burnup chart can be found here:

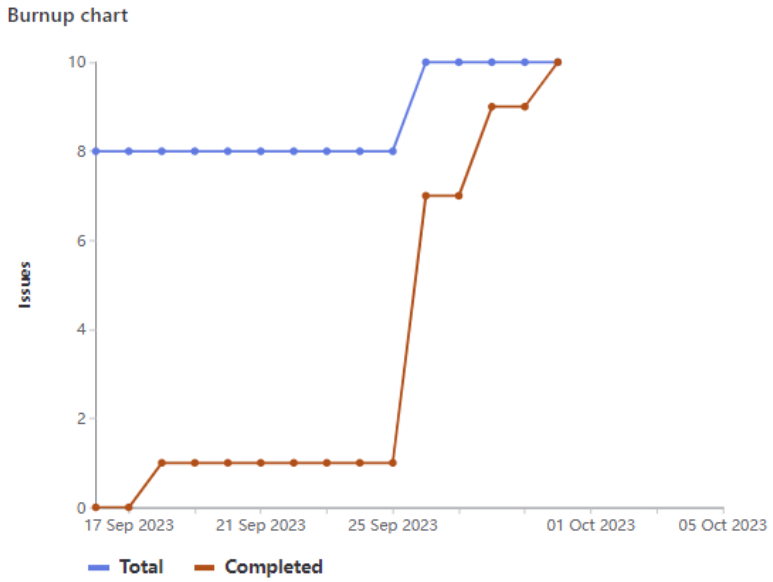
Link to GitLab Milestone:
https://git.infotech.monash.edu/FIT2101-S2-2023/MA_Thursday11am_Team1/project/-/milestones/3#tab-issues

An overview of the burndown chart and burnup chart by the end of the sprint is shown below. As shown in the charts, all the tasks are successfully completed on time with no issue.

Burndown Chart:



Burnup Chart:



3.3 Risk Register

In Scrum, a risk register is a document or tool used to identify, assess, and manage risks that may impact the successful delivery of a project or a sprint. It is a proactive approach to risk management and it is beneficial in a project management setting. The team has updated the risk register accordingly. A full overview of the live and updated risk register, along with the existing risk register is included in a link provided in section 3.1. The new risks discovered are shown in table 3.3.1, while the live and updated risk register is shown in table 3.3.2.

ID	Date	Risk	Likelih	Impact	Severit	Owner	Monitoring	Mitigation Plan

	raised	Description	ood of the risk occurring	if the risk occurs	y		Strategy	
27	18/9/2023	Hosting Security Risk	Low	High	Low	Product Owner	Ensure that software hosted is completely secured. Implement Real-time monitoring and regular testing,	1. Prioritise and implement timely software updates and security patches. 2. Enforce strong authentication measures, like multi-factor authentication 3. Regularly review and restrict user access based on roles and responsibilities.

Table 3.3.1: New Risks Discovered

ID	Occuration Date	Risk Description	Severity	Details	How the issue was resolved	Future Mitigation
03	14/9/2023	Team member burnout risk	High	1. By the end of the first sprint, all team members felt quite burnout from the work done. 2. This resulted in the team not feeling motivated to continue onto the second sprint. 3. A lot of time was wasted on no progress at all by all team members.	Our first meeting was held a few days after the first sprint ended, and a sprint goal was set, along with task allocations. Everyone was back on track, and started working on the second sprint.	1. Meetings should be held not long after each sprint end, to re-motivate team members. 2. Make sure that all team members are feeling fine to continuously work on the next sprint.
27	18/9/2023	Hosting Security Risk	Low	1. On this date, the website hosted was	We re-enabled the website, with a more secure hosting	1. Prioritise and implement timely software updates

				flagged for phishing and violating the terms of services. 2. Hosted website was taken down, and users are not able to use it anymore.	system. No data was lost, and the website was back online in a short time.	and security patches. 2. Enforce strong authentication measures, like multi-factor authentication. 3. Regularly review and restrict user access based on roles and responsibilities.
05	22/9/2023	Lack of communication risk	Medium	1. A few team members went out of town or country due to the mid-semester break. 2. This resulted in them being uncontactable for a period of time. 3. Some tasks are left undone, and the rest of the team members are not able to proceed with the project.	We eventually managed to contact them, and issues were resolved in a short time. Meetings were continuously held without absent team members. Some tasks are re-allocated and done on time.	1. Communicate with the team for any plans in advance. 2. If unable to do an allocated task on time, let the team know immediately. 3. Meetings must still be held even if there is any member absent.
12	22/9/2023	Lack of skills resources risk	Medium	1. A lot of tasks cannot be done smoothly because some team members did not have necessary skills needed for the project. 2. Progress of the project was greatly delayed.	Spike stories, including youtube videos, websites, and courses were added. Team members that are aware that they are lacking in the fields assigned themselves to the spike stories.	1. Spike stories can be added for members to learn from. 2. If there are any delays in progress, other members can help out.
15	25/9/2023	Scheduling conflicts and clashes risk	Medium	1. There were scheduling conflicts due to the mid-semester break.	Everyone had a discussion in the group chat to find out the availability of each member. Found a suitable	1. Ensure transparency between team members on scheduling matters. 2. Discussions must

				2. Some members had travel plans and commitments ahead of time. 3. Meeting had to be rescheduled multiple times	time to meet online. Made sure that everyone was available at the agreed time.	be done between team members to find a suitable time for everyone to be available.
17	25/9/2023	Team members' health risk	Low	1. A team member fell sick in the middle of the sprint. 2. Tasks allocated were not done in the expected time frame.	Task was transferred to another team member, and done in time. No delay in progress was met.	1. Make sure to report to the group immediately if any team member falls sick. 2. Tasks can be reallocated to another team member. 3. Aim to not delay the progress of the sprint.
02	26/9/2023	Conflict of ideas between team members risk	High	1. During the standup meeting conducted, there were conflicts of ideas between two members. 2. This resulted in both sides being very confused, and having different goals.	Discussions were held, and the conflict was resolved in a short time. Both sides were compromising each other, with the idea to reach the same project goal. Other team members were involved in the decision-making.	1. It is important for both sides to communicate and listen to each other. 2. Any conflict must be solved immediately, to prevent any confusion in the future that may affect the process of the project. 3. Involve other team members if conflict cannot be solved.
01	26/9/2023	Software bugs in the system risk	High	1. In the standup meeting conducted, we found out that there were many functionalities that had bugs, such as the buttons not linking to the correct page. 2. This would have been a big	During the meeting, a tester was assigned to test the software thoroughly. Changes were made to fix the bugs in the software afterwards. Made sure there were no more bugs before submission.	1. Have an assigned tester to do the chosen testing method on the software occasionally. 2. If there is any bug found, let the team know, and fix it immediately. 3. Often check for any bugs in the systems, during and outside of

				issue if not fixed.		meetings.

Table 3.3.2: Live and Updated Risk

3.4 Sprint Review Planning

This is the sequence for the demo:

1. Brian, the Scrum Master and Assistant Programmer, will briefly introduce the team, sprint goal and project information.
2. Ibrahim, the Product Owner and Front End Developer, will briefly show the sprint backlog and user stories in Trello.
3. Brian, the Scrum Master and Assistant Programmer, will demonstrate the login page and the background of the software.
4. Diana, the UI/UX Designer, will introduce the sprint views and how to switch between sprint views.
5. Ibrahim, the Product Owner and Front End Developer, will demonstrate how to add a sprint and how the sprint management page looks.
6. Chua, the Head Programmer, will introduce how to pull the tasks from the product backlog to the sprint and how to start the sprint.
7. Lucas, the Database Programmer, will introduce the sprint board and how to pull tasks within the sprint board. He will also demonstrate the burnup and burndown charts.
8. Rui En, the Technical Writer and Tester, will introduce the end sprint functionality and the navigation bar.
9. Ibrahim, the Product Owner and Front End Developer, will wrap up the demo.

3.5 Product Examination

To ensure that the product is complete and shippable after the first sprint, the tester in the team has performed an intensive review of the product and cross-checked it with the user stories recorded in the sprint backlog of sprint one and sprint two. The testing method used is user acceptance testing. The tester had produced a report after the testing (see the Testing Report linked in Section 3.1). The tester has also reviewed the user acceptance criteria and tasks checklist for each user story to ensure that the user stories in the sprint backlog are actually completed. The designated user stories were confirmed as “done”.

4. Sprint Retrospective

4.1 Main Criteria

Full discussions on the sprint retrospective can be found in the meeting agenda for meeting 3, attached in section 3.1.

Based on the template given, the team has performed sprint retrospective on sprint two. We then, evaluate the sprint based on the following points:

1. What went well?
2. What were the problems encountered?
3. What could have been done better?
4. What will we try next?
5. What questions do we have?

What went well?	1. We had more standup meetings during this sprint so we were able to clarify more in detail in terms of the task allocation so that everyone is aware about each other's task. This also ensured that the team is able to progress as a team as the task allocation splits the task evenly. 2. Pair programming was utilised among team members to complete the task allocations, which improved communication and knowledge sharing and produced more efficient solutions due to collaborative problem solving. 3. The tasks are allocated more evenly during this sprint, and everyone was able to contribute to the project more fairly. The progress therefore was faster during this sprint, as everyone was able to contribute equally. 4. The standup meetings during the sprint were done more efficiently and more goal oriented. Therefore, the meetings were shorter and more efficient. 5. Every team member was able to complete their tasks on time and with quality as the tasks were delegated more specifically during this sprint. 6. The tasks were distributed well to the team members in accordance with skill based delegation. Therefore, the team members who are familiar with a particular topic knew what they had to do. This ensured that the tasks were completed quickly and on time.
What were the problems encountered?	1. The problem encountered was some instances of miscommunication which led to one instance where two members had done and performed similar tasks on a feature. 2. The problem was that in the middle of the sprint, several user stories had to be added according to the stakeholder, extending the scope of the sprint which increased the workload required of the team. 3. The problem was that there were many bugs in the system, which were not initially discovered by the team. This affected the project quite severely, as there were many times where we could not proceed with the project due to the bugs. 4. The problem we encountered was the immense challenge when trying to fix a bug. This is because the person fixing it will often have to sloth through the code which is really time consuming. 5. The problem encountered in this sprint is that we lack knowledge on implementing the sprint graph. This resulted in many errors occurring when trying to implement the graphs feature. 6. The problem encountered was the lack of documentation in the code. Therefore, it became difficult for other team members to understand the code written by other team members.
What could have been done better?	1. We could have spent more time doing tutorials and learning about a particular coding language instead of solving problems on the spot when we encounter them as it would allow us to understand the coding language better and decrease the time in searching online for solutions or debugging which is time consuming at times. 2. The quality of code could have been better as there was a lot of refactoring required and several known issues in the code were addressed at a later time

	leading to accumulated technical debt. 3. We should start implementing the tests, even earlier during the sprint, to prevent any errors and bugs in the project that may affect the overall progress and overall duration of the project in the later days. 4. We could have written more documentation for the code. This would ensure that future developers or the other team members would be able to understand the code more clearly. 5. We can give more feedback to the code that our peers work on. This ensures that there is code review and would be able to improve the quality of the work. 6. There should be a clear guideline on how we should write and structure our code so that it would be easier for future developers or other team members to understand in the future.
What will we try next?	1. We would try to prepare more effectively for the standup meeting by first drafting out the updates we would like to give during the meeting individually instead of thinking of the updates on the spot during the meeting. This is because it would take up a lot of time if we were to reflect on the tasks we were able to complete in between standup meetings and this usually would make the standup meetings lengthy in duration. 2. To improve the quality of code, we would try to conduct regular code reviews where team members can identify and rectify many code quality issues, reducing the team's technical debt in the long run. 3. To prevent miscommunication, we would try to update the group chat every time someone is done with a task, so that there are no conflicts in the work done by multiple people, and less time would be wasted. During this sprint, there were some instances where miscommunication occurred, which resulted in conflicts of ideas. This caused the team to incur some wasted time off the project. 4. We would try to document our codes better so that others understand the codes written easier. 5. We would try to help each other review and debug codes if they encounter difficulties in implementing certain features. This would ensure that we are able to complete the tasks on time. 6. We would try to write more comments in the code or use enterprise solutions like Jira & Confluence to document our code. This would ensure quality in the code.
What questions do we have?	1. How can we make sure that team members are being productive and held accountable on a daily basis to ensure progress? 2. Were the sprint's goal and deliverables aligned with the client's expectations? 3. How do we test the overall software more efficiently? 4. How do we make sure the software is scalable? 5. How can we improve our code quality to produce a high quality product? 6. How can we document our code better?