RoboNav

SE4450: Software Engineering Design
Sprint 1 and 2 Plans

Faculty Advisor: **Dr. Yili (Kelly) Tang**

Email: ytang564@uwo.ca

Team 7:

Bryson Crook
bcrook4@uwo.ca
251217987

Christopher Higgins
chiggi24@uwo.ca
251245390

Mohamed El Dogdog
meldogdo@uwo.ca
251239204

Seth Langendoen
slangend@uwo.ca
251226610

November 3, 2024

GitHub Repo: https://github.com/meldogdo/RoboNav

Revision History

T – Team | C – Chris | M – Mohamed | B – Bryson | S - Seth

Date	Version	Description	Authors
October 31, 2024	0.1	Completed Project Backlog and User Story Requirements	T
November 1, 2024	0.1	Story Points Finished	C
November 2, 2024	0.1	Sprint 1: Defined Sprint Goal	T
November 2, 2024	0.1	Sprint 1: Established Team Working Agreement	M
November 2, 2024	0.1	Sprint 1: Defined Definition of Done	B
November 2, 2024	0.1	Sprint 1: Outlined Tasks for Each User Story	C
November 3, 2024	0.2	Sprint 1: Completed Daily Sprint Report	T
November 3, 2024	0.2	Sprint 1: Created Sprint Burndown Chart	M
November 3, 2024	0.2	Sprint 1: Conducted Sprint Review	B
November 3, 2024	0.2	Sprint 1: Finalized Sprint Backlog	C
November 3, 2024	0.2	Sprint 2: Defined Sprint Goal	T
November 3, 2024	0.2	Sprint 2: Established Team Working Agreement	M
November 3, 2024	0.2	Sprint 2: Defined Definition of Done	B
November 3, 2024	0.2	Sprint 2: Outlined Tasks for Each User Story	C
November 3, 2024	0.3	Sprint 2: Completed Daily Sprint Report	T
November 3, 2024	0.3	Sprint 2: Created Sprint Burndown Chart	M

Table of Contents

1. User Story Requirements

User Story ID	Requirements	User Story
1	Direct the robot’s movement.	As a **user**, I want to be able to control the robot's movement through the mobile app, so I can direct it to specific locations within the mapped indoor environment.
2	Monitor robot's real-time status.	As a **user**, I want to see the robot's real-time status (location, battery level, and operational mode) on the app, so I can monitor and make decisions on its deployment and usability.
3	Assign predefined routes on the indoor map.	As a **user**, I want to select specific routes or areas within the indoor map, so I can assign the robot to a predefined path for efficient navigation.
4	Receive low battery alerts.	As a **user**, I want to receive alerts when the robot's battery is low, so I can ensure it has enough power to complete tasks or return to a charging station.
5	Get error reports for robot issues.	As a **user**, I want to receive error reports if the robot encounters issues like

		connectivity loss or unexpected behavior, so I can address problems quickly.
6	Ensure secure real-time data updates in the backend.	As a **user**, I want the backend API to support secure, real-time data updates, so that all status changes are immediately reflected on the app.
7	Stay informed about task completion or issues.	As a **user**, I want to receive real-time updates when the robot completes a task or encounters an issue, so that I can stay informed about its status.
8	Save frequently used custom routes.	As a **user**, I want to save custom routes on the indoor map so that I can assign frequently used routes to robots quickly without needing to configure them each time.
9	Get notifications upon task completion.	As a **user**, I want to receive notifications when a robot has completed its assigned task so that I know when it's ready for a new assignment, improving workflow management.
10	Securely log in to the mobile app.	As a **user**, I want to securely log in to the mobile app so that I can access my personal settings and

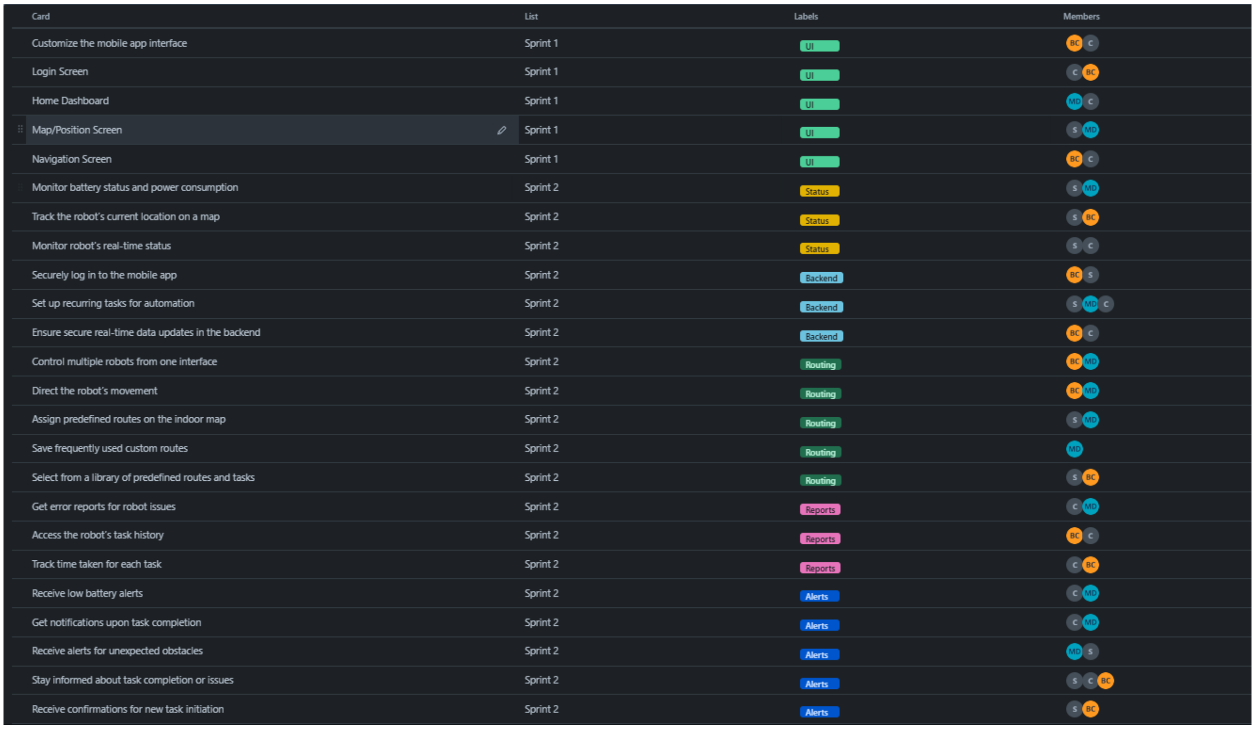
		manage my robots without unauthorized access.
11	Access the robot’s task history.	As a **user**, I want to access a history log of the robot's tasks and activities so that I can review past performance and make informed decisions for future tasks.
12	Receive confirmations for new task initiation.	As a **user**, I want to receive confirmation notifications when the robot starts a new task so that I am aware of its current operations and can manage my workflow accordingly.
13	Customize the mobile app interface (UI).	As a **user**, I want to customize the user interface of the mobile app so that I can access frequently used features quickly, improving my overall experience.
14	Select from a library of predefined routes and tasks.	As a **user**, I want to access a library of pre-defined routes and tasks for the robot so that I can quickly select common assignments without needing to create them from scratch.
15	Track the robot’s current location on a map.	As a **user**, I want to view a map of the robot’s current location in real-time so that I can track its movements and adjust tasks as needed.
16	Control multiple robots from one interface.	As a **user**, I want to be able to control multiple robots from a single app interface

		so that I can manage multiple operations efficiently without switching apps.
17	Receive alerts for unexpected obstacles.	As a **user**, I want to receive alerts when the robot encounters an unexpected obstacle so that I can take immediate action to address any potential disruptions.
18	Track time taken for each task.	As a **user**, I want to track the time taken for each task completed by the robot so that I can evaluate its productivity and efficiency.
19	Monitor battery status and power consumption.	As a **user**, I want to monitor the robot's battery status and power consumption in real-time so that I can manage energy use efficiently.
20	Set up recurring tasks for automation.	As a **user**, I want to set up recurring tasks for the robot so that it can automate regular operations without needing to be reprogrammed each time.
21	Login Screen (UI)	As a **user**, I want to securely log in to the app using my username and password to access my dashboard and manage my robots.
22	Home Dashboard (UI)	As a **user**, I want to view the status of my robots and current tasks on a central

		dashboard to monitor their performance at a glance.
23	Map/Position Screen (UI)	As a **user**, I want to manage the robot's position through a dedicated screen to efficiently perform various location-related functions.
24	Navigation Screen (UI)	As a **user**, I want to navigate my robots to specific locations through an intuitive interface to control their movement effectively.

Table 1 - User Stories

2. Project Backlog



Card	List	Labels	Members
Customize the mobile app interface	Sprint 1	UI	BC C
Login Screen	Sprint 1	UI	C BC
Home Dashboard	Sprint 1	UI	MD C
Map/Position Screen	Sprint 1	UI	S MD
Navigation Screen	Sprint 1	UI	BC C
Monitor battery status and power consumption	Sprint 2	Status	S MD
Track the robot's current location on a map	Sprint 2	Status	S BC
Monitor robot's real-time status	Sprint 2	Status	S C
Securely log in to the mobile app	Sprint 2	Backend	BC S
Set up recurring tasks for automation	Sprint 2	Backend	S MD C
Ensure secure real-time data updates in the backend	Sprint 2	Backend	BC C
Control multiple robots from one interface	Sprint 2	Routing	BC MD
Direct the robot's movement	Sprint 2	Routing	BC MD
Assign predefined routes on the indoor map	Sprint 2	Routing	S MD
Save frequently used custom routes	Sprint 2	Routing	MD
Select from a library of predefined routes and tasks	Sprint 2	Routing	S BC
Get error reports for robot issues	Sprint 2	Reports	C MD
Access the robot's task history	Sprint 2	Reports	BC C
Track time taken for each task	Sprint 2	Reports	C BC
Receive low battery alerts	Sprint 2	Alerts	C MD
Get notifications upon task completion	Sprint 2	Alerts	C MD
Receive alerts for unexpected obstacles	Sprint 2	Alerts	MD S
Stay informed about task completion or issues	Sprint 2	Alerts	S C BC
Receive confirmations for new task initiation	Sprint 2	Alerts	S BC

Figure 1 – Project Backlog

The backlog for the *RoboNav* project was organized in Trello, featuring a project board with lists for "To Do," "Doing," "Done," "Sprint 1," and "Sprint 2." Each user story was assigned to one of these lists, with team members designated to specific stories based on their expertise. Tasks were labeled according to their purpose, and as work progresses, user stories will be moved between lists to reflect their current status, enhancing collaboration and tracking throughout the project.

3. Story Points

Story points will be assigned using a relative estimation technique, where team members evaluate the complexity and effort required for each user story compared to others in the backlog. The team will hold estimation sessions, typically using the Fibonacci sequence (1, 2, 3, 5, 8, 13, etc.) to facilitate discussion and reach a consensus. Each user story will be assigned points based on its perceived difficulty, potential risks, and the time it may take to complete. This approach will help prioritize tasks effectively and ensure a balanced workload across sprints.

In this method, each team member will individually assign story points to user stories during collaborative sessions. After voting, the team will discuss any differing estimates to reach a consensus. The final story points will be calculated by averaging the votes, ensuring that multiple perspectives are considered for a more accurate assessment of effort and complexity. This process promotes collaboration and accountability within the team.

User Story ID	User Story Description	Bryson Crook (Story Points)	Chris Higgins (Story Points)	Mohamed El Dogdog (Story Points)	Seth Langendoen (Story Points)	Average Points
1	Direct the robot’s movement.	5	3	5	8	5.25
2	Monitor robot's real-time status.	3	2	3	3	2.75
3	Assign predefined routes on the indoor map.	8	5	8	8	7.25
4	Receive low battery alerts.	3	2	3	5	3.25
5	Get error reports for robot issues.	5	5	3	3	4.00
6	Ensure secure real-time data updates	8	5	8	8	7.25

	in the backend.					
7	Stay informed about task completion or issues.	2	3	2	3	2.50
8	Save frequently used custom routes.	3	3	5	5	4.00
9	Get notifications upon task completion.	2	2	3	3	2.50
10	Securely log in to the mobile app.	5	5	5	5	5.00
11	Access the robot’s task history.	2	3	3	2	2.50
12	Receive confirmations for new task initiation.	3	2	2	3	2.50
13	Customize the mobile app interface (UI).	2	3	2	3	2.50
14	Select from a library of predefined routes and tasks.	3	5	3	4	3.75
15	Track the robot’s current	8	8	8	8	8.00

	location on a map.					
16	Control multiple robots from one interface.	5	5	3	5	4.50
17	Receive alerts for unexpected obstacles.	3	2	3	4	3.00
18	Track time taken for each task.	5	5	5	3	4.50
19	Monitor battery status and power consumption.	3	2	2	3	2.50
20	Set up recurring tasks for automation.	5	3	5	5	4.50
21	Login Screen (UI)	3	3	2	3	2.75
22	Home Dashboard (UI)	5	6	5	4	5.00
23	Map/Position Screen (UI)	8	8	7	8	7.75
24	Navigation Screen (UI)	8	8	7	8	7.75

Table 2 – Story Points

The assumptions for story point completion include that all team members understand the user story requirements and acceptance criteria, possess the necessary skills, and adhere to a consistent estimation process like the Fibonacci sequence. It is assumed that the required resources are available, the team can complete the stories within the sprint timeframe, and effective collaboration will address any issues. Additionally, there should be no major scope changes during the sprint, and the team will recognize dependencies between user stories, following a clear definition of "done" that includes testing and documentation. Regular reviews will also help improve future estimations.

4. Sprint 1

1.1. Sprint Goal

The goal of this sprint is to complete all the user interface (UI) components of the mobile application, ensuring a cohesive and user-friendly design. By focusing on the UI, the team aims to create an intuitive experience for users to interact with the mobile app effectively. This includes designing and implementing essential elements such as buttons, menus, forms, and navigation systems that will facilitate seamless operation and control of the autonomous robots.

1.2. Team Working Agreement

Purpose

This agreement outlines our commitment to collaboration, communication, and performance standards throughout the duration of the *RoboNav* project.

Team Values

- **Respect:** We will treat each other with respect and value diverse perspectives.
- **Accountability:** Each team member is responsible for their tasks and commitments.
- **Transparency:** We will maintain open communication and share information promptly.
- **Support:** We will support each other in achieving our individual and team goals.

Communication

- **Channels:** We will use Discord for daily communication.
- **Frequency:** We will have weekly check-in meetings every Wednesday to discuss progress, challenges, and next steps.
- **Updates:** Team members will provide updates on their tasks during meetings and via the project management tool Trello.

Decision-Making

- **Consensus:** We will strive for consensus when making decisions; if consensus cannot be reached, we will take a vote.
- **Escalation:** If a decision cannot be resolved, it will be escalated to advisor Dr. Yili (Kelly) Tang for resolution.

Roles and Responsibilities

- Each team member will have clearly defined roles based on their expertise and the project needs. Roles will be documented in the project management tool.

Conflict Resolution

- **Addressing Issues:** Team members are encouraged to address conflicts directly with each other in a constructive manner.
- **Mediation:** If conflicts persist, the team will seek mediation from advisor Dr. Yili (Kelly) Tang.

Work Ethic and Standards

- **Quality:** We will maintain high standards for our work and hold each other accountable for meeting those standards.
- **Deadlines:** We will respect deadlines and communicate any potential delays as early as possible.
- **Flexibility:** We understand that changes may occur, and we will adapt to new circumstances collaboratively.

Feedback and Reflection

- **Regular Feedback:** We will provide and seek feedback regularly to foster a culture of continuous improvement.
- **Retrospectives:** At the end of each sprint, we will conduct a retrospective to reflect on our processes and identify areas for improvement.

Commitment

By signing this agreement, each team member commits to adhering to these principles and supporting each other in achieving the goals of the *RoboNav* project.

Team Member Name	**Signature**	**Date**
Bryson Crook	*Bryson Crook*	November 2, 2024
Chris Higgins	*Chris Higgins*	November 2, 2024
Mohamed El Dogdog	*Mohamed El Dogdog*	November 2, 2024
Seth Langendoen	*Seth Langendoen*	November 2, 2024

1.3. Definition of Done

- **UI Components:** All specified UI components for the mobile application are designed, implemented, and match the design mockups.
- **Testing:** Basic functionality is tested, with any critical bugs identified and logged for resolution.
- **Documentation:** User documentation for UI components is created, including instructions for navigating and utilizing the app.
- **Code Review:** All code related to UI components is reviewed by a group member, and necessary feedback has been addressed.
- **Stakeholder Approval:** Key stakeholders have reviewed the UI components and provided feedback or approval before moving on to the next sprint.

1.4. Tasks for Each User Story

User Story ID	User Story Description	Release	Task 1	Task 2	Task 3
13	Customize the mobile app interface (UI).	Sprint 1	Create design mockups that highlight the layout and arrangement of frequently used features for user customization.	Implement the frontend functionality that allows users to customize their interface by adding, removing, or rearranging features.	Test the customization feature with users to ensure it works as intended and improves their overall experience, making adjustments based on feedback.

21	Login Screen (UI)		Create wireframes and UI mockups for the login screen, including fields for username and password, a "Login" button, and a "Forgot Password?" link.		
22	Home Dashboard (UI)		Create a visually appealing layout for the Home Dashboard that organizes and displays key information about each robot, including battery level, current task, and location.	Develop the functionality to dynamically display the robot overview.	Build a section beneath the robot overview that lists the active tasks being executed by the robots, allowing users to monitor ongoing activities at a glance
23	Map/Position Screen (UI)		Create an intuitive interface that allows users to easily select a robot from a list or graphical representation for position management.	Implement the various functions for managing the robot's position, including options to check, set, save, and remove locations.	Design and develop an output terminal at the bottom of the Map/Position Screen that displays results from selected functions and allows users to clear or recover outputs.

24	Navigation Screen (UI)		Create a user-friendly interface that enables users to select a robot they want to navigate, ensuring the selection process is clear and intuitive.	Develop input fields or controls that allow users to specify a destination, whether by selecting a location on a map or entering coordinates manually.	Create an output terminal at the bottom of the Navigation Screen to provide users with feedback on navigation commands and display results of their actions.

Table 3 – Task Breakdown Sprint 1

1.5. Daily Sprint Report

Total Story Points: 25.75

Team Members: Bryson, Chris, Mohamed, Seth

Story Point Goal per Member: Approximately 6.44 points each (allocated based on the complexity of assigned tasks and team expertise).

Story Point Allocation Method: Relative Estimation Method.

Sprint Duration: 27 days (06-Nov-24 to 04-Dec-24)

Projected Completion: The sprint is scheduled for 27 days but may conclude sooner if tasks proceed without major issues. Additional days are reserved to address potential deployment challenges, incomplete tasks, or unforeseen conflicts.

Story Point Allocation per Team Member:

- Bryson: 5.25 points
- Chris: 5.00 points
- Mohamed: 7.75 points
- Seth: 7.75 points

Here is a table outlining the daily story points that each team member is responsible for completing:

	Bryson Crook	Christopher Higgins	Mohamed El Dogdog	Seth Langendoen
User Story # (Points)	13, 21 (5.25)	22 (5.00)	23 (7.75)	24 (7.75)
Date				
06-Nov-24				0.75
07-Nov-24	0.75			
08-Nov-24		1		1
09-Nov-24			1	
10-Nov-24	0.75			
11-Nov-24		1	1	
12-Nov-24				
13-Nov-24	0.75			
14-Nov-24		1	1	1
15-Nov-24				1
16-Nov-24	0.75			
17-Nov-24	0.75	1		
18-Nov-24			1	
19-Nov-24				1
20-Nov-24		1		
21-Nov-24			1	
22-Nov-24				
23-Nov-24	0.75			1
24-Nov-24	0.75			
25-Nov-24			1	1
26-Nov-24				
27-Nov-24			1	

28-Nov-24		Excess		1
29-Nov-24	Excess		0.75	
30-Nov-24				
01-Dec-24				
02-Dec-24			Excess	Excess
03-Dec-24				
04-Dec-24				

Table 4 - Daily Sprint 1 Report

1.6. Sprint Burndown Chart

The burnout chart illustrates the expected progress of story points completed over 27 days, derived from the calculations in section 1.7 Sprint Review. This approach helps to track whether the team is on pace to meet their sprint goals, as the remaining work should gradually decrease. Any significant deviations from this trend can signal potential issues in the project's progress, allowing for timely adjustments.

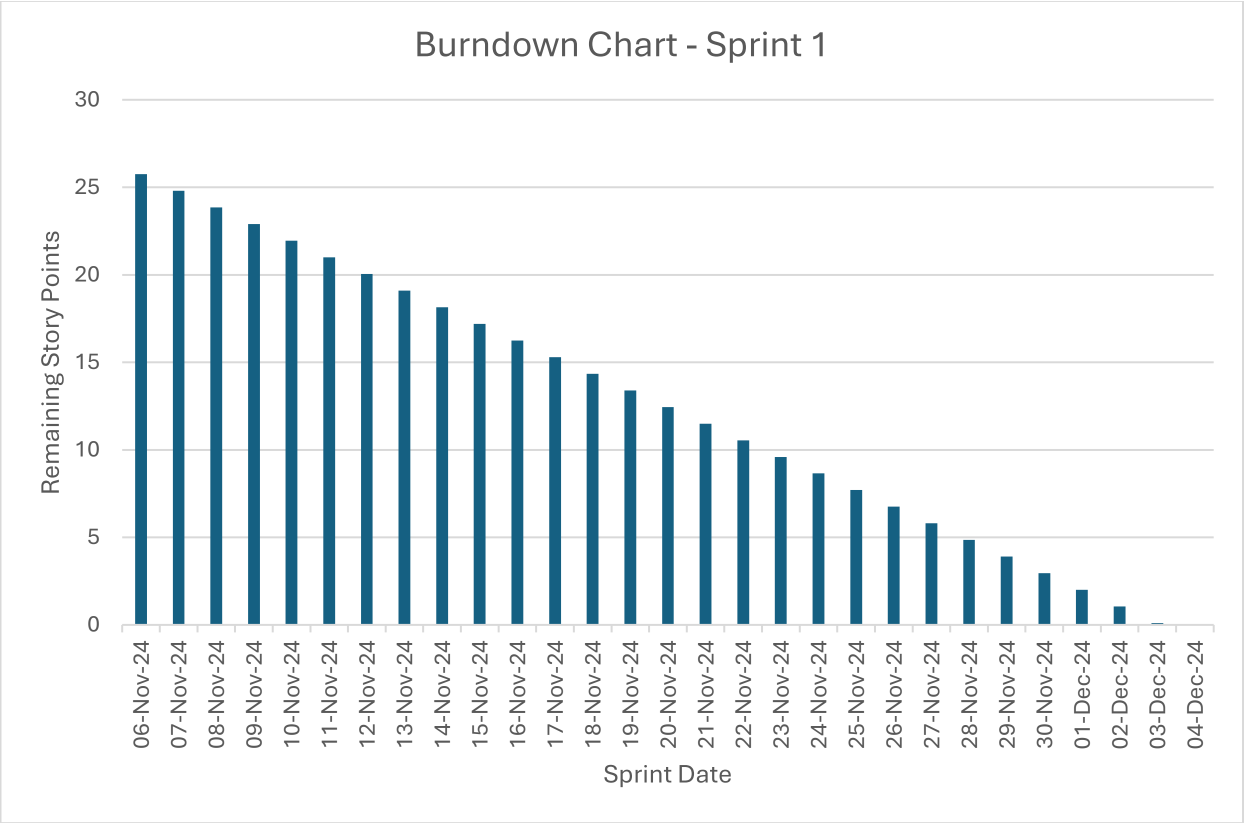


Figure 2 – Burndown Chart Sprint 1

1.7. Sprint Review

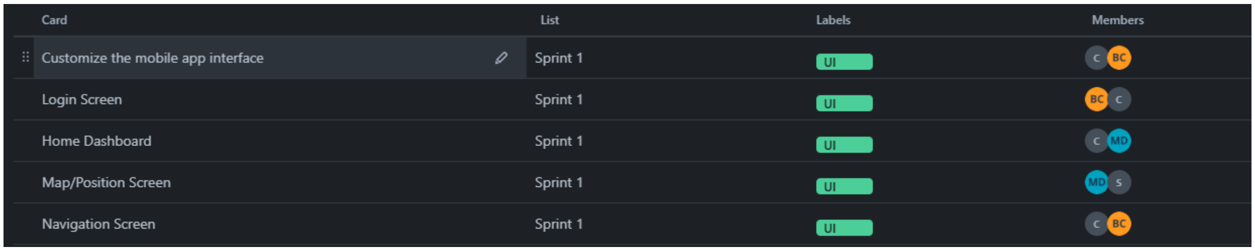
Summary of Completed User Stories for Sprint 1

- Login Screen (UI) – 2.75 points
- Home Dashboard (UI) – 5.00 points
- Map/Position Screen (UI) – 7.75 points
- Navigation Screen (UI) – 7.75 points
- Customize the user interface (UI) – 2.50 points

Total Story Points Finished = 25.75 points

Team Velocity = 25.75 (Story Points/Sprint) / 27 (Days) = **0.95 (Story Points/Day)**

1.8. Sprint Backlog



Card	List	Labels	Members
Customize the mobile app interface	Sprint 1	UI	C BC
Login Screen	Sprint 1	UI	BC C
Home Dashboard	Sprint 1	UI	C MD
Map/Position Screen	Sprint 1	UI	MD S
Navigation Screen	Sprint 1	UI	C BC

Figure 3 – Sprint 1 Backlog

5. Sprint 2

2.1. Sprint Goal

The goal of this sprint is to prepare all components necessary for the final demonstration, ensuring a fully integrated and functional system. This includes completing the backend source code and setting up cloud infrastructure to support reliable, real-time data handling. Additionally, the team will finalize the mobile-backend integration to enable seamless communication between the app and the robots. A thorough test plan and test cases will be developed to validate functionality, while a verification and validation document will ensure that all project requirements are met. Finally, a comprehensive user manual will be created for both the API and mobile application, equipping end-users with the necessary guidance to operate the system efficiently.

2.2. Team Working Agreement

Purpose

This agreement continues to outline our team's commitment to effective collaboration, consistent communication, and performance standards, with a focus on meeting the final demonstration requirements for the *RoboNav* project.

Team Values

- **Respect:** We maintain mutual respect and appreciate each member's insights, particularly as we approach project completion.
- **Accountability:** Each team member takes ownership of their tasks and delivers on commitments to meet the project goals.
- **Transparency:** We ensure open communication regarding task progress, challenges, and required resources.
- **Support:** We are committed to assisting one another in completing tasks and addressing last-minute challenges.

Communication

- **Channels:** Discord remains our primary communication tool, with additional updates in Trello for task management.
- **Frequency:** Weekly check-ins will continue every Wednesday, with additional meetings as needed for final demo preparation.
- **Updates:** Task progress updates will be shared during meetings and tracked in Trello, especially to monitor integration and testing progress.

Decision-Making

- **Consensus**: We will strive to reach a consensus on all decisions; if disagreements arise, we will discuss until a satisfactory resolution is reached or proceed with a vote.
- **Escalation:** Unresolved decisions or significant blockers will be escalated to Dr. Yili (Kelly) Tang for guidance, especially as they relate to demo requirements.

Roles and Responsibilities

- **Defined Roles:** Each team member will continue with roles based on expertise, with special attention to completing final code integration, testing, and documentation.
- **Documentation and Ownership:** All roles and task ownership will be clearly defined in Trello and referenced during meetings to ensure clarity and accountability.

Conflict Resolution

- **Direct Resolution:** Team members will address conflicts directly and constructively, with the goal of maintaining team cohesion as we near project completion.
- **Mediation:** If needed, Dr. Yili (Kelly) Tang will provide mediation for unresolved conflicts to keep the team aligned on project goals.

Work Ethic and Standards

- **Quality:** High standards of quality remain essential, especially for critical deliverables like the backend source code, test plans, and user documentation.
- **Deadlines:** All deadlines must be respected, with early communication of any challenges or potential delays.
- **Flexibility:** While adaptability remains important, we will work to minimize last-minute changes to ensure a smooth demo preparation process.

Feedback and Reflection

- **Continuous Improvement:** Team members will give and receive feedback, emphasizing final preparations and integration improvements.
- **Sprint Retrospective:** At the end of Sprint 2, we will conduct a retrospective to evaluate project success and reflect on our teamwork before the final demo.

Commitment

By signing this agreement, each team member reaffirms their commitment to these principles and to supporting one another in achieving the goals of Sprint 2 and the final demo for the *RoboNav* project.

Team Member Name	Signature	Date
Bryson Crook	*Bryson Crook*	November 3, 2024
Chris Higgins	*Chris Higgins*	November 3, 2024
Mohamed El Dogdog	*Mohamed El Dogdog*	November 3, 2024
Seth Langendoen	*Seth Langendoen*	November 3, 2024

2.3. Definition of Done

1. **Backend Source Code**:
   - Code is complete, tested, and deployed in the cloud environment.
   - All endpoints and backend functionalities are operational and connected to the mobile app.
2. **Test Plan and Test Cases**:
   - A comprehensive test plan is documented, covering unit, integration, and user acceptance tests.
   - Test cases are run with results recorded, ensuring functionality meets project requirements.
3. **Cloud Infrastructure Setup**:
   - Cloud resources are configured and securely handle real-time data flow between the app and robots.
   - Data storage and processing requirements are met, with scalability for multiple robots verified.
4. **Mobile-Backend Integration**:
   - Mobile app successfully interfaces with the backend API, supporting real-time data and control functions.
   - Integration is tested to ensure smooth communication, with all data points accurately synced.

5. **Verification and Validation Document**:
    - A document verifying that the project meets the outlined requirements, and validating that it is ready for final demonstration.
    - Includes a checklist of completed features and any adjustments made during testing.
6. **User Manual for API and Mobile App**:
    - A comprehensive user manual is created for both the API and mobile app, covering setup, operation, and troubleshooting.
    - Manual is reviewed for clarity and completeness, ensuring that end-users can independently operate the system.
7. **Final Demonstration Readiness**:
    - All components are successfully integrated and function as intended.
    - The project is ready for demonstration, with key team members prepared to explain, operate, and answer questions on each feature.

2.4. Tasks for Each User Story

User Story ID	User Story Description	Release	Task 1	Task 2	Task 3
19	Monitor battery status and power consumption.	Sprint 2	Create backend functionality to retrieve battery status and power data.	Display battery and power data in the mobile app UI.	Test battery monitoring accuracy.
15	Track the robot's current location on a map.		Implement location-tracking API and integrate it into the map UI.	Ensure real-time location updates in the app.	Test accuracy and responsiveness of location data.

2	Monitor robot's real-time status.		Set up backend to gather real-time robot status data.	Create UI component to display real-time status.	Test real-time updates for reliability.
10	Securely log in to the mobile app.		Implement user authentication and encryption for login.	Test login functionality for security and ease of use.	
20	Set up recurring tasks for automation.		Build backend functionality for scheduling recurring tasks.	Create UI for users to manage and adjust recurring tasks.	
6	Ensure secure real-time data updates in the backend.		Implement secure data transfer protocols.	Test real-time data handling for security vulnerabilities.	
16	Control multiple robots from one interface.		Enable multi-robot selection and control in backend.	Update UI to support multiple robot views and controls.	Test control responsiveness and accuracy across multiple robots.
1	Direct the robot's movement.		Implement movement control commands in the backend.	Add UI controls for manual movement control.	
3	Assign predefined routes on the indoor map.		Create backend function for assigning predefined routes.	Develop UI for users to select and assign routes.	
8	Save frequently		Implement save and retrieval functionality	Test custom route saving and loading.	

	used custom routes.		for custom routes.		
14	Select from a library of predefined routes and tasks.		Build a library in backend for predefined routes and tasks.	Design UI for easy selection from route/task library.	
5	Get error reports for robot issues.		Implement backend error logging system.	Configure notifications in app for error alerts.	Test error reporting to ensure timely alerts.
11	Access the robot's task history.		Set up task history logging in backend.	Display task history in the mobile app's history view.	
18	Track time taken for each task.		Log task duration in backend.	Add UI display for task time in history.	
4	Receive low battery alerts.		Configure backend to monitor battery levels and trigger alerts.	Enable notifications in the app for low battery alerts.	
9	Get notifications upon task completion.		Create backend alerts for task completion.	Design UI notifications for completion alerts.	
17	Receive alerts for unexpected obstacles.		Implement obstacle detection in backend.	Enable real-time alerts in the app for obstacles.	

7	Stay informed about task completion or issues.		Set up a task reporting system in the backend.	Display task updates and issues in the app feed.	
12	Receive confirmations for new task initiation.		Develop confirmation alerts in the backend for task initiation.	Display confirmation notifications in the app.	

Table 5 - Task Breakdown Sprint 2

2.5. Daily Sprint Report

Daily Sprint Report for Sprint 2

- **Total Story Points**: 79.5
- **Team Members**: Bryson, Chris, Mohamed, Seth
- **Story Point Goal per Member**: Allocated based on task complexity and expertise (average goal: 19.875 points per member).
- **Story Point Allocation Method**: Relative Estimation Method.
- **Sprint Duration**: 112 days (04-Dec-24 to 25-Mar-25)
- **Projected Completion**: The sprint covers a 112-day period, allowing ample time for each story point with additional buffer days for deployment, testing, and troubleshooting.

Story Point Allocation per Team Member (based on complexity and role):

- **Bryson**: 20.75 points
- **Chris**: 20 points
- **Mohamed**: 20.5 points
- **Seth**: 18.25 points

Here is a table outlining the daily story points that each team member is responsible for completing:

	Bryson Crook	Christopher Higgins	Mohamed El Dogdog	Seth Langendoen
User Story # (Points)	19, 15, 2, 10, 11 (20.75)	20, 6, 16, 14 (20)	1, 3, 8, 5 (20.5)	18, 4, 9, 17, 7, 12 (18.25)
Date				
04-Dec-24		1		
05-Dec-24	1			
06-Dec-24				0.25
07-Dec-24			0.5	
08-Dec-24		1		
09-Dec-24				
10-Dec-24	.75		1	
11-Dec-24				1
12-Dec-24		1		
13-Dec-24				
14-Dec-24				
15-Dec-24	1		1	
16-Dec-24				
17-Dec-24		1		
18-Dec-24				1
19-Dec-24	1			
20-Dec-24			1	
21-Dec-24				
22-Dec-24		1		1
23-Dec-24	1			

24-Dec-24				
25-Dec-24			1	
26-Dec-24				1
27-Dec-24				
28-Dec-24	1	1		
29-Dec-24			1	
30-Dec-24				
31-Dec-24				
01-Jan-25		1		
02-Jan-25				1
03-Jan-25			1	
04-Jan-25				
05-Jan-25				
06-Jan-25	1			
07-Jan-25			1	
08-Jan-25				
09-Jan-25		1		1
10-Jan-25				
11-Jan-25	1		1	
12-Jan-25				
13-Jan-25		1		
14-Jan-25			1	
15-Jan-25				
16-Jan-25				
17-Jan-25				
18-Jan-25	1	1		1

19-Jan-25	1			
20-Jan-25			1	
21-Jan-25				
22-Jan-25				
23-Jan-25	1			
24-Jan-25		1		
25-Jan-25			1	
26-Jan-25				
27-Jan-25				
28-Jan-25	1			1
29-Jan-25		1		
30-Jan-25				
31-Jan-25				
01-Feb-25				
02-Feb-25		1	1	
03-Feb-25				
04-Feb-25				1
05-Feb-25				
06-Feb-25				
07-Feb-25	1			
08-Feb-25		1		
09-Feb-25				1
10-Feb-25				
11-Feb-25		1		
12-Feb-25				
13-Feb-25	1		1	1

14-Feb-25	1			
15-Feb-25				
16-Feb-25				
17-Feb-25		1		1
18-Feb-25			1	
19-Feb-25	1			
20-Feb-25				
21-Feb-25				1
22-Feb-25				
23-Feb-25				
24-Feb-25		1	1	
25-Feb-25	1			
26-Feb-25				1
27-Feb-25				
28-Feb-25				
01-Mar-25	1			
02-Mar-25				
03-Mar-25		1		
04-Mar-25			1	
05-Mar-25				1
06-Mar-25				
07-Mar-25				
08-Mar-25			1	
09-Mar-25	1			1
10-Mar-25				
11-Mar-25				

12-Mar-25		1		1
13-Mar-25	1			
14-Mar-25			1	
15-Mar-25				1
16-Mar-25				
17-Mar-25	1	1	1	
18-Mar-25				
19-Mar-25				Excess
20-Mar-25				
21-Mar-25		Excess		
22-Mar-25	Excess		Excess	
23-Mar-25				
24-Mar-25				
25-Mar-25				

Table 6 – Daily Sprint 2 Report

2.6. Sprint Burndown Chart

The burnout chart illustrates the expected progress of story points completed over 112 days, derived from the calculations in section 2.7 Sprint Review. This approach helps to track whether the team is on pace to meet their sprint goals, as the remaining work should gradually decrease. Any significant deviations from this trend can signal potential issues in the project's progress, allowing for timely adjustments.

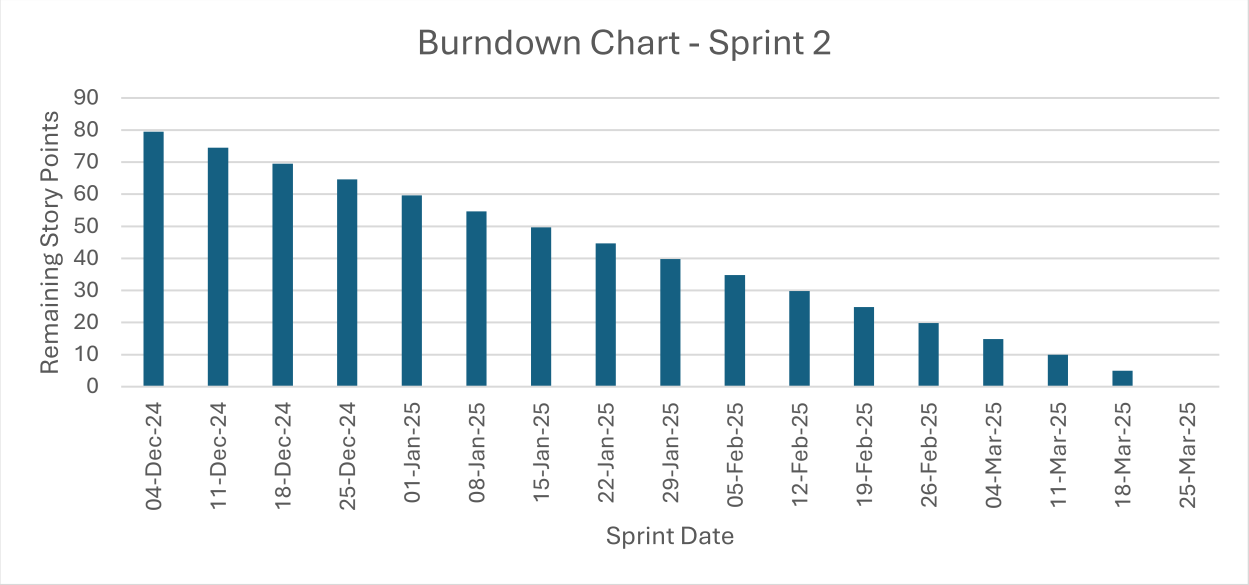


Figure 4 - Burndown Chart Sprint 2

2.7. Sprint Review

Summary of Completed User Stories for Sprint 2

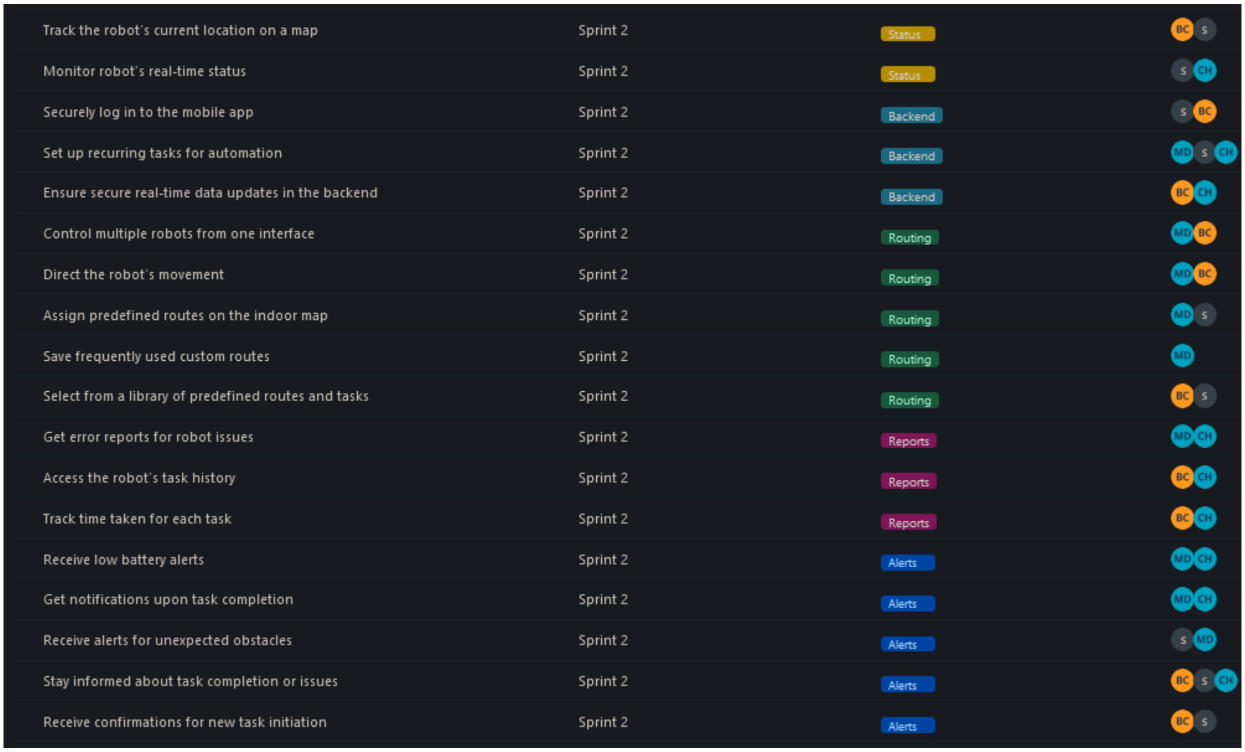
- Direct the robot’s movement – 5.25 points
- Monitor robot's real-time status – 2.75 points
- Assign predefined routes on the indoor map – 7.25 points
- Receive low battery alerts – 3.25 points
- Get error reports for robot issues – 4.00 points
- Ensure secure real-time data updates in the backend – 7.25 points
- Stay informed about task completion or issues – 2.50 points
- Save frequently used custom routes – 4.00 points
- Get notifications upon task completion – 2.50 points
- Securely log in to the mobile app – 5.00 points
- Access the robot’s task history – 2.50 points
- Receive confirmations for new task initiation – 2.50 points
- Select from a library of predefined routes and tasks – 3.75 points
- Track the robot’s current location on a map – 8.00 points

- Control multiple robots from one interface – 4.50 points
- Receive alerts for unexpected obstacles – 3.00 points
- Track time taken for each task – 4.50 points
- Monitor battery status and power consumption – 2.50 points
- Set up recurring tasks for automation – 4.50 points

Total Story Points Finished = 79.50 points

Team Velocity = 79.50 (Story Points/Sprint) / 112 (Days) = **0.71 (Story Points/Day)**

2.8. Sprint Backlog



Track the robot's current location on a map	Sprint 2	Status	BC S
Monitor robot's real-time status	Sprint 2	Status	S CH
Securely log in to the mobile app	Sprint 2	Backend	S BC
Set up recurring tasks for automation	Sprint 2	Backend	MD S CH
Ensure secure real-time data updates in the backend	Sprint 2	Backend	BC CH
Control multiple robots from one interface	Sprint 2	Routing	MD BC
Direct the robot's movement	Sprint 2	Routing	MD BC
Assign predefined routes on the indoor map	Sprint 2	Routing	MD S
Save frequently used custom routes	Sprint 2	Routing	MD
Select from a library of predefined routes and tasks	Sprint 2	Routing	BC S
Get error reports for robot issues	Sprint 2	Reports	MD CH
Access the robot's task history	Sprint 2	Reports	BC CH
Track time taken for each task	Sprint 2	Reports	BC CH
Receive low battery alerts	Sprint 2	Alerts	MD CH
Get notifications upon task completion	Sprint 2	Alerts	MD CH
Receive alerts for unexpected obstacles	Sprint 2	Alerts	S MD
Stay informed about task completion or issues	Sprint 2	Alerts	BC S CH
Receive confirmations for new task initiation	Sprint 2	Alerts	BC S

Figure 5 – Sprint 2 Backlog