

CITS5551 Team 1 Project Plan

Timeline & Workload

The timeline for our project starts in Week 8 of Semester 2 (2020) as our project was changed mid-semester from the *Electric Jet Ski* to the *Solar Powered Autonomous Boat* (SPAB). Project work will be undertaken during university semester weeks, excluding the summer and in-semester breaks, for a total span of 17 weeks. Note that to account for a late start to the project some work will likely occur over the summer break to ensure that 200 hours of work is done on the project by each team member. This is to reduce the time commitment team members must put in during teaching weeks.

A rough estimate of the hours required per person for the duration of the project is included below along with rationale for that time allocation. Also included is a Gantt chart to graphically illustrate our project timeline

Task	Approx. Hours Required	Rationale
Team meetings	34	The team will meet for approximately 2 hours per week. At least one meeting with the supervisor (Thomas Braunl) or project leader (Pierre-Louis Constant) will be conducted per week. 2 hours x 17 weeks = 34 hours
Requirements elicitation	20	Requirements elicitation was expediated over a two-week period, with approximately ten hours of work being undertaken per week. This includes additional meetings with the project supervisor/leader as well as brainstorming and documentation time. 10 hours x 2 weeks = 20 hours
Prototyping	30	Prototyping was undertaken over a three-week period, with an average of ten hours of work to be undertaken by each group member. 10 hours x 3 weeks = 30 hours
System development and testing	100	This time is dedicated to miscellaneous software development and testing tasks that shall occur over a ten-week period. This shall be split up into an eight-week development and testing phase, with two weeks of slack time to complete any outstanding tasks. Approximately ten hours of work per week should be undertaken. 10 hours x 8 weeks = 80 hours 10 hours x 2 weeks = 20 hours (slack time) TOTAL 100 hours
Formal documentation	20	The final two weeks of the semester are to be dedicated to formally documenting the system for handover. This will align with weeks 11 and 12 of Semester 1, 2021. 10 hours x 2 weeks = 20 hours
TOTAL	204	

Task	Semester 1 (Weeks)											
	1	2	3	4	5	6	7	8	9	10	11	12
<i>Jet Ski Project</i>												
Requirements Elicitation												
Prototype Development												

Task	Semester 2 (Weeks)											
	1	2	3	4	5	6	7	8	9	10	11	12
Development/Testing												
Dev/Testing Slack Time												
Documentation												

Details of Individual Contribution

The subsequent pages detail the individual work contributions with an approximation of the duration taken to complete work.

Note that while our team did most work separately at the prototyping stage, we met regularly to discuss our progress, organise what should be done next and allocate resources for future work. Additionally, we worked in a highly collaborative way during the requirements elicitation stage, with most work being done in person with the project supervisor/leader, collaboratively as a group or online over a video call. For this reason, we have elected to omit these hours from our individual contributions.

Lucie Cunningham (22260943)

Task	Reasoning Behind Task	Difficulties	Approx. Hours Taken to Complete Task	Cumulative Total Hours
Enable Remote Desktop on Raspberry Pi	To be able to connect and work on the pi without needing to physically connect hardware to it as at some points that would either be impossible or at least very time consuming due to where and how it is placed in the boat.	<ul style="list-style-type: none">- Did not know the password to the pi- Tried to use the pi as a wireless access point first	5	5
Install ROS Kinetic Kame	We can use ROS to connect to the ArduPilot and communicate with it. There are many other functions that ROS provides that may be beneficial for future endeavours. It was also our supervisor's preference.	<ul style="list-style-type: none">- Installing ROS is a long process (hours long) as there are 191 packages that must be built.- There were many errors during the install process due to the compiler, other dependencies and memory exhaustion. This meant that the process had to be started again many times.- The process had to be supervised as there are occasionally requirements for human input.- The boat was located in a shared lab and made noise so it was turned off during the process of installing a couple of times.	30	35

Tom Chan (23023191)

Task	Reasoning Behind Task	Difficulties	Approx. Hours Taken to Complete Task	Cumulative Total Hours
Fix and test CAN bus communication for the REV Jet Ski project	To enable the communication between the host and the Jet Ski.	<ul style="list-style-type: none">- Did not know CAN protocol- Faulty CAN bus hardware	3	3
Build the dashboard interface	To experiment with the frameworks and tools that we plan to use.	<ul style="list-style-type: none">- Recent version of Angular is very different from the version we were familiar with.- OpenLayers has a complex architecture.	30	33
Host the prototype on my personal NAS for testing	Allow the team to access the app remotely.	None	1	34

Dylan Carpenter (21982288)

Task	Reasoning Behind Task	Difficulties	Approx. Hours Taken to Complete Task	Cumulative Total Hours
Investigate existing hosting service	To investigate the current web hosting solution for the REV project and establish the feasibility of hosting a NodeJS server alongside the existing Apache/python setup.	<ul style="list-style-type: none">- Unfamiliar with implementing forward/reverse proxies with ".htaccess" files- Extra level of caution required to avoid disrupting existing web services	3	3
Implementing backend server and database	To host the API and backend infrastructure required to support the solar boat.	<ul style="list-style-type: none">- Re-learning the Express framework & asynchronous NodeJS programming- Work done with MongoDB JavaScript package had to be swapped to use Mongoose	17	20
Creating a simulator for the solar boat	To get around limited access to hardware. Allows for easier testing of the API and backend systems.	<ul style="list-style-type: none">- Graphically plotting waypoints on a map (for presentation)	6	26

