

NUS Orbital 2023

Liftoff Proposal



canvas++

Team 5583

Elgin Tan Zhi Ming

Anthony Tamzil

Content

Proposed Level of Achievement	3
Motivation	3
Aim	3
User Stories	3
Features	5
Schedule Planner	6
Integration of Canvas, NUSNextBus and NUSMods	6
Profile Creation	6
Task Manager	6
Telegram Bot Notifications	7
Social Elements	7
Teaching Utilities	7
Timetable Generator	8
Degree Requirement Checker	8
Timeline and Development Plan	9
Qualifications	11

Proposed Level of Achievement

Artemis

Motivation

Have you ever found yourself procrastinating and wasting time when there is work to be done? How about the time when you wanted to keep track of tasks but had to manually add it for each module every week? Every student has definitely dreaded the work that lies ahead of them at least once in their academic lives. As university students who are overloaded on modules and bogged down with work, we realize that the reason we procrastinate so much is due to a lack of structure and organization especially when it comes down to prioritizing and managing our workload on top of our already hectic schedule. Furthermore, there are constant updates to Canvas with new quizzes or assignments opening up every other day. The workload each student has changes drastically everyday and it is not uncommon for students to lose track of how much work they have left. This often results in overly negative perceptions of the workload. Don't you wish that there was a platform out there that kept you focused on your work and updated on the countless announcements?

This inspired us to create a web application that has all the features Canvas has, and much more! From having your own personalized task manager to a Telegram bot delivering updates straight to your devices, Canvas++ is designed to ensure that students meet their deadlines while being able to make the most out of their time in university without stressing out so much about their academics.

Aim

Canvas++ aims to be a one-stop platform where students can easily keep track of tasks and provide assistance to the users by simplifying certain tasks such as planning timetables together with friends, tracking attendance and consultation bookings for tutors. Canvas++ also guides users through the painstaking process of creating new timetables during every Module Registration with a timetable generation feature in the web application.

User Stories

1. As a student with a busy schedule, I want to be notified whenever my professor or TA uploads a new task or assignment, preferably by Telegram since it is one of the most frequently used platforms in university.
2. As a student who is trying to cope with the workload of my different modules, I want to be able to keep track of all my upcoming assignments and projects as well as their deadlines through a consolidated dashboard.

3. As a student who is constantly coming up with new timetables during every round of Module Registration (ModReg), I want to be able to generate optimized timetables according to my preferences to make the whole ModReg process more streamlined.
4. As a student who has the occasional quizzes and exams, I want to be able to keep track of important dates.
5. As a student who only has limited time to do my assignments and revision, I want to be able to prioritize my tasks according to their deadlines and weightages.
6. As a student who is trying to come up with study sessions with other students, I want to be able to see my friend's schedules so that we can easily come up with a common date and time to study together.
7. As a student who is trying to make it to class on time, I want to be able to see the bus timings on the dashboard so that I can not only see the location and time of my next class, but also estimate when I should start making my way.
8. As a student who may be shy to approach my tutor, I want to be able to give constructive feedback and ask questions about the module while remaining anonymous.
9. As a tutor who is juggling with my studies and teaching duties, I want to be able to keep track of my consultation slots and streamline the consultation booking process.
10. As a tutor who may have many students in one class, I want my students to take their own attendance when they attend my class to save time and effort on my end.

Features

This **Progressive Web-App (PWA)** will allow students to sync data from Canvas using their NUS Emails. The web-app will primarily serve as a one-stop platform for students to manage their tasks and stay updated on new assignments. Students will also be able to access their files and assignments needed straight from the platform. Furthermore, students previously viewed their timetables from NUSMods but our PWA will also support the import of timetables from NUSMods to their profiles on the PWA. When coupled with our task manager, the task manager allows them to manage and plan almost every aspect of their academic life. The dashboard containing the task manager also has NUS bus timings synced from NUSNextBus to give students a heads up on when to leave for their classes.

A **Telegram Bot** complements the **PWA** by serving as an alternate way to notify the user of new tasks and announcements on the **PWA**. It can also be used by TAs or Professors to track attendance conveniently. The bot will also serve as a platform for students to anonymously ask questions or provide feedback to tutors, and facilitate booking of consultation slots with their tutors. The bot will also assist tutors with attendance taking, allowing students to mark their own attendance with a special code given out by the tutor at every tutorial.

The **PWA** has the potential to include other additional features as listed below. This would benefit students as it includes social elements, Module Registration assistance and quality of life features which would allow students to manage their packed academic schedule with greater ease and convenience.

Main Features of PWA + Telegram Bot

1. Integrate Canvas, NextBUS and NUSMods into a single platform
2. Update students on any new tasks through Telegram
3. Allow students to add, delete, edit and view tasks

Additional features are elaborated in the list of features below

Tech Stack (Overall)

1. HTML / CSS / Javascript (Web-app development)
2. Web API
3. Next.js
4. TailwindCSS

Schedule Planner

Description

The dashboard also has a schedule or event planner for users to utilize. It consists of tutorials and lessons that the users need to attend, along with the location and timing of these events. Exams, quizzes or graded sessions that are particularly important for users will be highlighted on the schedule to ensure that the user would not forget or be late for the event. Students can add their own events to the schedule planner too.

Integration of Canvas, NUSNextBus and NUSMods

Description

There would be basic porting of Canvas features to allow users to view and access tasks from their own NUS account through Canvas API where most features in Canvas should be accessible via the PWA with the exception of Panopto (which is frequently used to rewatch lecture videos). Panopto can be accessed through a Sessionless Launch URL since it is an external tool outside Canvas.

From the dashboard, users would also be able to see the bus timings of the bus stop prioritized based on their current location as well as their upcoming schedule. These timings are provided by an unofficial NUSNextBus API. NUSNextBus is an app frequently used by NUS students to see the shuttle bus timings within the NUS transportation network. This feature provides convenience for students to immediately see when the next relevant bus is coming.

Tech Stack

- [Canvas API](#)
- [Unofficial NUSNextBus API](#)
- [NUSMods API](#)

Profile Creation

Description

Users would be able to input information such as their name, degree, year of study, as well as link their other accounts such as Google, Microsoft and also their telegram handle to enable tighter integration with our PWA. Students would also be able to store their NUSMods timetable in their profile through a link that would bring them to the NUSMods official page with their timetable.

Tech Stack

- Firebase Authentication

Task Manager

Description

Users would be able to create tasks to manage their school work, and also group the tasks with different tags. To simplify generating repetitive tasks for courses every semester, this would be able to generate tasks in their timetable automatically and tag them with course code or lesson time. In addition to basic task creation, users would be able to specify a weightage for graded assignments and tasks will be sorted by their priority determined by the due date and weightage.

Tasks would be synced either through:

1. Google Tasks
2. Microsoft To Do
3. Canvas++ Account

This will allow users to organise their tasks more easily and allow them to plan and manage their time more effectively.

Tech Stack

- [Google Tasks API](#)
- [Microsoft To Do API](#)
- Realtime Database by Firebase

Telegram Bot Notifications

Description

When users link their telegram handle to our web app, it would allow them to receive updates with regards to different updates from Canvas such as:

1. Announcements
2. File uploads
3. Messages
4. Reminders on assignments

Due to the popularity of Telegram in university, students naturally open Telegram multiple times per day, making it convenient for them to check and inform themselves of any outstanding or new tasks that require their attention, even on the desktop.

Tech Stack

- [Telegram API](#) (python-telegram-bot)
- [Canvas API](#)
- Heroku for hosting

Social Elements

Description

Users can add other users as friends and compare timetables. When other users are added as friends, basic information like the name, degree and timetable can be seen. For both users to become friends, it has to be mutually agreed on by both sides. The purpose of this feature is to allow students to easily find common periods of time and venues to study together for projects.

Teaching Utilities

Description

As part of teaching utilities, there are 3 subfeatures:

1. Attendance Taking
2. Anonymous Questions
3. Teaching Feedback

Through the telegram bot, tutors would be able to generate a unique code, be it a numerical code or QR code, to allow their class to scan or input to take attendance easily. Furthermore, tutors would be able to keep track of consultations through the telegram bot too, minimizing additional effort required for these menial tasks. The telegram bot can be used by students to ask the tutors any anonymous questions if they are too shy to approach them directly or provide teaching feedback to the tutors to improve their learning experience.

Tech Stack

- Similar to the tech stack used in “Telegram Bot Notifications”

Timetable Generator

Description

Students would be able to utilize this tool for several purposes. With an algorithm, students would be able to generate a timetable with their own constraints, such as preferred start and end time, or even generate one together with their friends to enable them to easily plan which slots to bid together for. This will allow students to better plan and make it easy for friends to bid for the same slots by simplifying the timetable planning process.

Tech Stack

- [Google CP-SAT Solver](#) for timetable generator

Degree Requirement Checker

Description

Users can use this feature to track their degree progress by keeping track of which modules have been taken thus far, how many Modular Credits (MC) they have attained and what requirements the taken modules have fulfilled so far. Specifically, this feature will detect if the module is a core module for the user’s degree. It will then automatically count the MCs from the core module towards the Programme Requirement category for the user. If the module is not a core module, the user is able to manually enter the module’s detail, MCs that the module has and which category it should be under (Unrestricted Elective or Common Curriculum).

Timeline and Development Plan

Week	Tasks	Description	In Charge	Date
Pre Week 1	Prepare for liftoff [Completed]	Finalise proposal Familiarize with tech stack Do up project poster and video	Elgin Anthony	Before 08 May
1	Finalize Frontend GUI Design Set Up PWA Skeleton and Telegram Bot Do Up Login Screens [Completed]	Finalize designs of user interface using Figma	Elgin Anthony	08 May - 15 May
		Set up Github repository and workflow Set up PWA and Telegram Bot Set up firebase authentication	Elgin	
		Create Login, Signup and Forget Password Screens UI	Anthony	
2	Profile Creation Account Linking	Set up linking with other services (Google, Microsoft, Telegram, Canvas) Code up PWA Navigation Elements	Elgin	15 May - 22 May
		Code up Profile Creation UI and Home Tab	Anthony	
3	Main Screens Feedback System	Implement Feedback System Code up Planner, Settings Tab UI Create linking and permissions UI	Elgin	22 May - 29 May
		Ensure Basic Task Manager Functionality (Includes prioritization of Tasks) Code up Home and Modules Tab UI	Anthony	
Evaluation Milestone 1: Ideation				29 May
4	Task Manager Syncing Notifications Exam Tracking	Integration with Google Tasks and Microsoft ToDo	Elgin	29 May - 5 June
		Sharing of Project Tasks among Users Create Push Notification for Expiring Tasks	Anthony	
5	Integration with Canvas	Port over Quizzes, Assignments, Announcements, Grades	Elgin	5 June - 12 June
		Port over Files, External Tools and Conversations	Anthony	
6	Teaching Utilities NUSNextBus Integration	Ensure that the bus timings on the dashboard is linked to NUSNextBus	Elgin	12 June - 19 June
		Implementation of Teaching Utilities	Anthony	
7	Testing, Refining and Debugging	Refine user experience and UI Test the web-app extensively for bugs	Elgin Anthony	19 June - 26 June

Evaluation Milestone 2: Prototyping				26 June
8	Timetable Generation Degree Requirement	Implement personal timetable generation	Elgin	26 June - 3 July
		Implement automated degree requirement checking for core modules	Anthony	
9	Timetable Generation (cont.) Degree Requirement (cont.)	Implement group timetable generation	Elgin	3 July - 10 July
		Allow users to manually input non-core modules to the degree requirement checker	Anthony	
10	Social Elements	Implementation of Social Elements	Elgin Anthony	10 July - 17 July
11	Testing, Refining and Debugging	Refine user experience and UI Test the web-app extensively for bugs Gather user feedback	Elgin Anthony	17 July - 24 July
Evaluation Milestone 3: Extension				24 July

Qualifications

Anthony Tamzil

Y1 Computer Science Student

Telegram Handle: @tonyantstark

NUS Email: e0958492@u.nus.edu

Linkedin: <https://www.linkedin.com/in/anthony-tamzil/>

Github: <https://github.com/anthonytamzil>

- Taken Programming Methodology 1 and 2 (CS1101S and CS2030S), Data Structures and Algorithms (CS2040S)
- Decent proficiency in Java, JavaScript and Python
- Basic proficiency in CSS, HTML
- Prior experience in website development (mainly frontend) through hackathons (NUS LifeHack 2022 & IDEATE 2022)
- Substantial project experience with Machine Learning and Tensorflow through internships

Elgin Tan Zhi Ming

Y1 Computer Science Student

Telegram Handle: @ElginTzM

NUS Email: ElginTzM@u.nus.edu

Github: <https://github.com/ElginTzM>

- Taken Programming Methodology 1 and 2 (CS1101S and CS2030S), Data Structures and Algorithms (CS2040S), Computer Organisation (CS2100)
- Decent proficiency in Java, JavaScript and Python
- Basic proficiency in CSS, HTML