# NexusNTU

## Project Plan

# Version 1.0 approved

Shen Jia Cheng
Vu Thao Nguyen
Akshay Krishna
Aishwarya Anand
Nguyen Phuong Linh
Tran Son Viet

School of Computer Science and Engineering,
Nanyang Technological University

**1st Oct 2025**

# Revision History

| Revision Number | Date | Primary Author(s) | Comments |
|---|---|---|---|
| 1.0 | 1st Oct 2025 | Shen Jia Cheng<br>Vu Thao Nguyen<br>Akshay Krishna<br>Aishwarya Anand<br>Nguyen Phuong Linh<br>Tran Son Viet | Initial Project Plan Draft |
| 1.1 | 13th Oct 2025 | Shen Jia Cheng<br>Akshay Krishna<br>Nguyen Phuong Linh | Format changing |

# Table of Contents

# 1 Introduction

## 1.1 Project Overview

NexusNTU is an all-in-one web application designed to be a comprehensive digital companion for international students at Nanyang Technological University (NTU). The project aims to solve the problem of a fragmented and inefficient onboarding process by centralizing essential information and services into a single, user-friendly platform. This includes campus navigation, access to university resources, AI-powered support, and personalized content hubs to help students integrate smoothly into university life.

## 1.2 Project Description and Scope

NexusNTU targets the challenges international students face when navigating multiple disconnected university websites, physical offices, and third-party systems. This fragmentation leads to information overload, confusion, and delays during their critical transition period.

The core of the NexusNTU platform will consolidate these workflows into the following main features:

- **User Account and Authentication:** Users can register, log in securely with two-factor authentication, and manage their profiles.
- **Centralised Information Hub:** A personalized news viewer aggregates official NTU announcements and news from the user's home country.
- **Campus Navigation:** An interactive map provides walking directions and integrates with campus bus services via deep links to the official app.
- **Nearest Amenities Finder:** Users can locate nearby amenities like eateries, clinics, and transit stations, and save them to a "Favorites" list.
- **AI-Powered Support (Chatbot):** A 24/7 chatbot answers frequently asked questions about campus life, visa procedures, and administration.
- **Quick Links Dashboard:** Provides organized and verified links to essential resources, personalized for the user's school and home country.
- **Utility Suite:** Includes tools like a currency converter.

Due to the nature of the system as a course project, several constraints are in place:

- The application will be developed as a Progressive Web App (PWA), accessible on mobile and desktop browsers. Native iOS or Android development is not within the initial scope.
- The system will rely on deep-linking to many official NTU systems rather than direct API integration, as access to proprietary back-end systems is not available.
- The app will only store non-sensitive metadata and user preferences. It will not handle or store sensitive personal data such as passport numbers or FINs.

# 2 Project Organization

## 2.1 Team Structure

The following is the list of executive roles, as required by CMM level 3.

- Project Manager: Aishwarya Anand
- Release Engineer/Manager: Tran Son Viet
- Front-end Developer: Akshay Krishna
- Back-end Developer: Nguyen Phuong Linh
- QA Manager: Shen Jia Cheng
- QA Engineer: Vu Thao Nguyen

## 2.2 Roles and Responsibilities

**Project Manager: Aishwarya Anand**
The Project Manager drives the project's success by taking ownership of the plan and its execution. They are responsible for assigning work, tracking progress against the timeline and budget, and reporting on the project's status. By managing the team, facilitating meetings, and representing our work to clients, they ensure the project is delivered successfully and that its outcomes are rigorously evaluated.

**Release Engineer: Nguyen Phuong Linh**
As the central coordinator for our software releases, the Release Engineer establishes the rhythm of our release cycles and puts the processes in place to ensure they run smoothly. A critical part of their job is to anticipate and solve problems that threaten the project's scope and timeline. They keep a close watch on development to make sure we're hitting our targets and are responsible for communicating all key plans and changes to the entire team.

**Front-end Developer: Akshay Krishna**
From initial mockups to final deployment, the Front-end Developer is involved in every stage of bringing the user interface to life. Their primary responsibility is to convert design wireframes into clean, functional code, but their role extends further. They validate the technical feasibility of new designs, ensure every component aligns with established principles, and write the tests needed to guarantee their work integrates seamlessly with the rest of the application.

**Back-end Developer: Tran Son Viet**
The Back-end Developer's primary role is to connect our user interface to the server-side logic, making the application work. They achieve this by building and maintaining the APIs, designing the underlying data storage systems, and implementing essential security and data protection measures. As a full-cycle contributor, they generate all necessary documentation and tests for their code and work hand-in-hand with front-end developers to ensure a seamless integration.

**QA Manager: Shen Jia Cheng**
The Quality Assurance Lead establishes and manages the end-to-end testing process. This involves creating the master QA plan, designing the strategies for how we test, and overseeing the development of individual test plans. A key part of their role is to ensure the quality team has the right number of people to succeed

and to serve as the go-to expert for resolving difficult software quality problems as they emerge.

**QA Engineer: Vu Thao Nguyen**
This role is centered on the hands-on validation of software quality. The engineer designs targeted testing strategies to ensure thorough coverage and executes detailed test procedures to find and report bugs. A key part of their job is to conduct product assessments and verify that the final build aligns with the original requirements. By assisting the QA Manager with test planning, they ensure the entire team's efforts are organized and effective, ultimately guaranteeing an acceptable level of software quality.

## 2.3 Team Communication

Effective communication is critical for the success of the project. The primary channels of communication include:

- **Weekly Meetings:** Held to discuss progress, resolve issues, and plan upcoming tasks.
- **Shared Documents:** All project documentation (SRS, Project Plan, etc.) is maintained and regularly updated on a shared platform (Google Drive).
- **Version Control & Issue Tracking:** Source code is managed using GitHub, and project tasks and issues are tracked with a tool like Trello or Jira.
- **Instant Messaging:** A group chat on a platform like Telegram is used for daily announcements, quick updates, and informal discussions

# 3 Process Definition

## 3.1 Lifecycle Model

The Agile development methodology has been selected for the NexusNTU project. Agile is well-suited for this project because it allows for the development of the application in small, incremental releases. This iterative approach is ideal for the project's time-boxed course schedule and aligns with the chosen Object-Oriented Rapid Prototyping (OORP) method mentioned in the SRS.

The benefits of this approach for NexusNTU include:

- **Flexibility:** It accommodates changes in requirements, which is common in a project where user feedback is gathered through prototypes.
- **Incremental Delivery:** Key features like the Quick Links Dashboard and Campus Navigation can be developed and tested in separate sprints, allowing for faster delivery of a functional proof-of-concept.
- **Early Testing:** Testing is integrated throughout the lifecycle, which helps in identifying and fixing bugs early, reducing the overall effort required for quality assurance.
- **Transparency:** Constant feedback mechanisms and regular progress updates ensure all team members and stakeholders are aligned.

# 4 Schedule

## 4.1 Activity Dependencies and Schedule

The project scheduling timeline is illustrated in the Gantt Chart below.

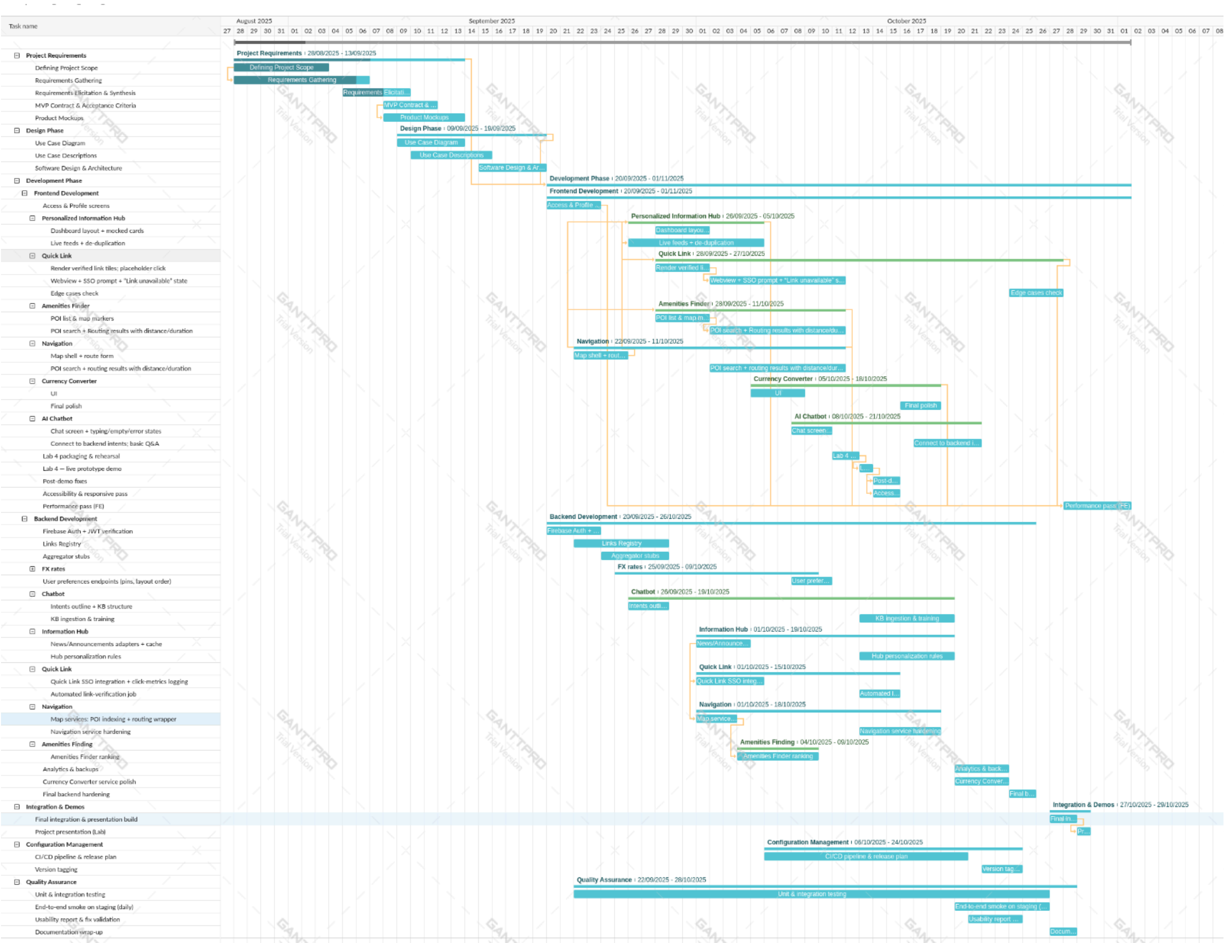


*Figure 1. Gantt Chart for Project Deliverables*

## 4.2 Work Breakdown Structure

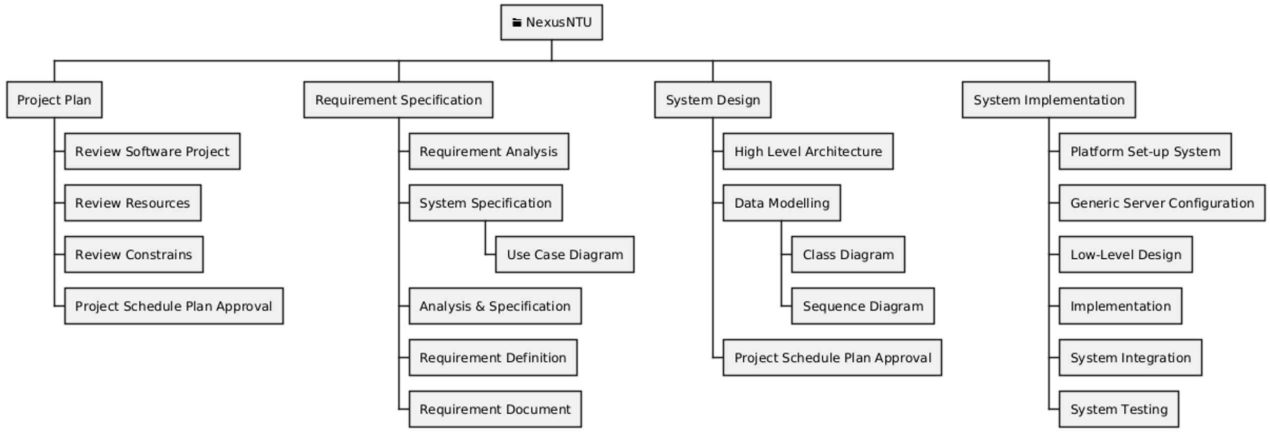


*Figure 2. Work Breakdown Structure*

## 4.3 Work Packages

The project is broken down into the following key work packages based on the software development lifecycle:

1. Project Proposal
2. Requirement Elicitation
3. Software Quality Assurance
4. System Requirement Specification
5. Risk Management
6. Project Plan
7. User Interface Design
8. Technical Architecture
9. Data Modeling
10. Coding & Unit Testing
11. Integration & System Testing

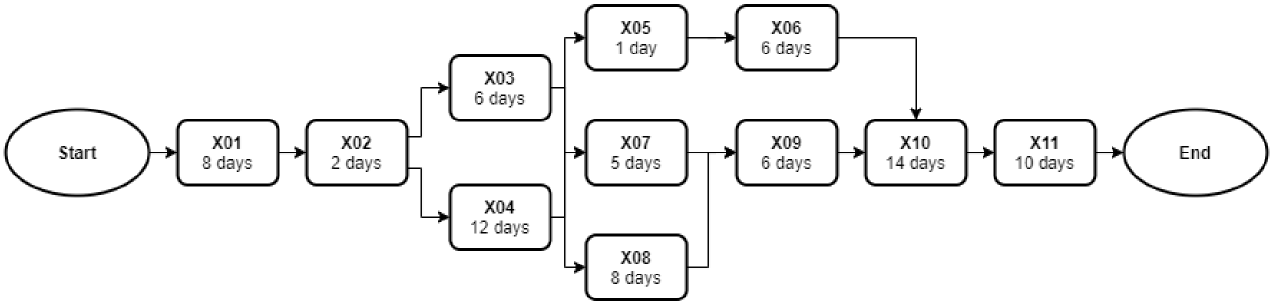
## 4.4 Activity Dependencies

The following table describes the dependencies of the deliverable work packages:

| Work Package # | Work Package Description | Duration | Dependencies |
|---|---|---|---|
| X01 | Project Proposal | 8 days | - |
| X02 | Requirements Elicitation | 2 days | X01 |
| X03 | Software Quality Assurance | 6 days | X02 |
| X04 | System Requirement | 12 days | X02 |

| | | | |
|---|---|---|---|
| | Specification | | |
| X05 | Risk Management | 1 day | X03, X04 |
| X06 | Project Plan | 6 days | X03, X04, X05 |
| X07 | User Interface Design | 5 days | X03, X04 |
| X08 | Technical Architecture | 8 days | X03, X04 |
| X09 | Data Modelling | 6 days | X08 |
| X10 | Coding & Unit Testing | 14 days | X06, X07, X09 |
| X11 | Integration & System Testing | 10 days | X10 |

The following Activity Network Diagram describes the above in more graphical detail:



## Critical Path Analysis:

| Work Package | Earliest Start | Earliest Finish | Latest Start | Latest Finish | Slack Time |
|---|---|---|---|---|---|
| X01 | Start of day 0 | End of day 7 | Start of day 0 | End of day 7 | 0 days |
| X02 | Start of day 8 | End of day 9 | Start of day 8 | End of day 9 | 0 days |
| X03 | Start of day 10 | End of day 15 | Start of day 23 | End of day 28 | 13 days |
| X04 | Start of day 10 | End of day 21 | Start of day 10 | End of day 21 | 0 days |
| X05 | Start of day 22 | End of day 22 | Start of day 29 | End of day 29 | 7 days |
| X06 | Start of day 23 | End of day 28 | Start of day 30 | End of day 35 | 7 days |
| X07 | Start of day 22 | End of day 26 | Start of day 25 | End of day 29 | 3 days |
| X08 | Start of day 22 | End of day 29 | Start of day 22 | End of day 29 | 0 days |
| X09 | Start of day 30 | End of day 35 | Start of day 30 | End of day 35 | 0 days |
| X10 | Start of day 36 | End of day 49 | Start of day 36 | End of day 49 | 0 days |
| X11 | Start of day 50 | End of day 59 | Start of day 50 | End of day 59 | 0 days |

**Critical Path Analysis:**
The critical path for this project is determined to be:
Start → X01 → X02 → X04 → X08 → X09 → X10 → X11 → End
Any delays in the tasks along this path will directly impact the final project delivery date.

## 4.5 Work Package Details

Work packages are listed below. A team member, indicated in bold, has been assigned as primarily responsible for each work package and will coordinate that package.

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X01 - Project Proposal |
| **Assigned To** | All members |
| **Effort** | 8 Person Days |
| **Start Date** | 27-08-2025 |
| **Purpose** | To define project objectives, identify project scope, and determine target specifications for the NexusNTU application. |
| **Inputs** | Problem identified, some possible solutions and requirements from users. |
| **Activities** | 1. Draft problem statement.<br>2. Development of Business case.<br>3. Draft design objectives, based on the selected solution.<br>4. Define Technical approaches of the solutions.<br>5. Establish brief project management. |
| **Outputs** | Project Proposal Document |

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X02 – Requirements Elicitation |
| **Assigned To** | 2 people |
| **Effort** | 2 Person Days |
| **Start Date** | 03-09-2025 |
| **Purpose** | To collect detailed project requirements for further refinement through |

| | |
|---|---|
| | the use of various elicitation methods. |
| **Inputs** | Initial customer needs and the complete Project Proposal (X01). |
| **Activities** | 1. Conducting formal interviews with the customer to clarify software requirements.<br>2. Performing brainstorming sessions to identify potentially useful requirements.<br>3. Developing a prototype to elicit further requirements and establish a common understanding. |
| **Outputs** | A consolidated document listing all gathered requirements from interviews, brainstorming, and prototyping. |

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X03 - Software Quality Assurance |
| **Assigned To** | 2 people |
| **Effort** | 6 Person Days |
| **Start Date** | 05-09-2025 |
| **Purpose** | To define the objectives, procedures, and roles necessary for implementing effective quality assurance, ensuring the project adheres to established standards and quality benchmarks. |
| **Inputs** | All outputs from work package Requirement Elicitation (X02). |
| **Activities** | 1. Define the project's purpose and scope from a quality perspective.<br>2. Outline the management structure, roles, and responsibilities for SQA tasks.<br>3. Specify the documentation standards the project will adhere to.<br>4. Determine the tools and methods for maintaining software quality. |
| **Outputs** | A formal Software Quality Assurance (SQA) Plan document. |

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X04 - System Requirement Specifications |
| **Assigned To** | 3 people |
| **Effort** | 12 Person Days |
| **Start Date** | 05-09-2025 |
| **Purpose** | To create a shared understanding between the customer and the project team by formally documenting all information from the requirement elicitation phase. |
| **Inputs** | All outputs from work package Requirement Elicitation (X02). |
| **Activities** | 1. Formally document the problem statement, project background, and an overall description.<br>2. Define the project's technical and operational constraints.<br>3. Using the elicited requirements, specify all operational, functional, input, process, output, hardware, software, and deployment needs. |
| **Outputs** | The final System Requirements Specification (SRS) document. |

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X05 - Risk Management Plan |
| **Assigned To** | 2 people |
| **Effort** | 1 Person Days |
| **Start Date** | 17-09-2025 |
| **Purpose** | To identify, analyze, and document all potential risks (positive or negative). This plan will detail mitigation strategies and the process for performing, recording, and monitoring risk management activities. |
| **Inputs** | All outputs from work package Software Quality Assurance (X03), System Requirements Specification Details (X04). |
| **Activities** | 1. Perform risk identification and analysis covering cost, scheduling, and technical aspects.<br>2. Define mitigation strategies for all identified risks. |
| **Outputs** | A formal Risk Management Plan document. |

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X06- Project Plan |
| **Assigned To** | 3 people |
| **Effort** | 6 Person Days |
| **Start Date** | 18-09-2025 |
| **Purpose** | To provide a comprehensive overview of the project's processes and to establish a detailed project schedule. |
| **Inputs** | All outputs from work package Software Quality Assurance (X03), System Requirements Specification Details (X04), and Risk Management (X05). |

| | |
|---|---|
| **Activities** | 1. Define the project overview and scope.<br>2. Outline the project team organization.<br>3. Model the project's development processes.<br>4. Develop the detailed project schedule.<br>5. Formulate project cost estimations.<br>6. Allocate resources based on effort, duration, and team size estimates.<br>7. Create a product delivery checklist.<br>8. Identify project best practices.<br>9. Summarize risk management and quality assurance approaches.<br>10. Define project monitoring and control mechanisms. |
| **Outputs** | A comprehensive Project Plan document. |

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X07 - User Interface Design |
| **Assigned To** | 2 people |
| **Effort** | 5 Person Days |
| **Start Date** | 24-09-2025 |
| **Purpose** | To design an intuitive communication medium between the system and the end-user. The initial prototype will establish the product's look and feel, helping to prevent future confusion or disagreements on design. |
| **Inputs** | All outputs from work package Software Quality Assurance (X03), System Requirements Specification Details (X04). |
| **Activities** | 1. Draft an initial high-fidelity prototype based on the SRS and customer requirements.<br>2. Gather opinions from stakeholders to refine the UI design, applying design principles and Shneiderman’s Eight Golden Rules.<br>3. Iterate on the design until it is fully coherent with all stated requirements.<br>4. Construct the user interface's screen layouts as a minimum viable product. |
| **Outputs** | A functional front-end prototype of the User Interface. |

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X08 - Technical Architecture |
| **Assigned To** | All Members |
| **Effort** | 8 Person Days |
| **Start Date** | 24-09-2025 |
| **Purpose** | To conduct a high-level architectural design, creating an overview of how system subcomponents will interact. |
| **Inputs** | All outputs from work package Software Quality Assurance (X03), System Requirements Specification Details (X04). |
| **Activities** | 1. Define the software system's architecture and its core components, including their interrelationships.<br>2. Select the hardware and software infrastructure, including the operating system, programming language, and other key technologies.<br>3. Address non-functional requirements such as maintainability, portability, and reusability in the design. |
| **Outputs** | A document detailing the high-level design and architectural specifications. |

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X09 - Data Modelling |
| **Assigned To** | All Members |
| **Effort** | 6 Person Days |
| **Start Date** | 02-10-2025 |

| | |
|---|---|
| **Purpose** | To design the data model that will serve as the foundation for the project's database. |
| **Inputs** | All outputs from the work package Technical Architecture (X08). |
| **Activities** | 1. Analyze data flows and entity relationships within the system.<br>2. Define the required data types and database tables needed to represent all stored data. |
| **Outputs** | A formal data modeling document. |

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X10 - Coding & Unit Testing |
| **Assigned To** | All Members |
| **Effort** | 14 Person Days |
| **Start Date** | 08-10-2025 |
| **Purpose** | To develop the system according to the established specifications, architecture, and UI design. This includes conducting preliminary unit testing on individual components for factors like optimality and reliability before integration. |
| **Inputs** | All outputs from the work package Project Plan (X06), User Interface Design (X07), Data Modelling (X09). |
| **Activities** | 1. Implement software modules based on the design specifications in the SRS.<br>2. Execute unit test plans for each module to ensure individual component correctness. |
| **Outputs** | All source code, header files, and the integrated system test plan with its corresponding test cases. |

| | |
|---|---|
| **Project** | NexusNTU |
| **Work Package** | X11 - Integration & System Testing |
| **Assigned To** | All Members |
| **Effort** | 10 Person Days |
| **Start Date** | 22-10-2025 |
| **Purpose** | To combine individual components into a complete system, build data flow between them, and resolve any logical or syntactical errors. All testing procedures will be documented in a Test Plan report. |
| **Inputs** | All outputs from the work package Coding & Unit Testing (X10). |
| **Activities** | 1. Simulate user interactions to test overall system behavior.<br>2. Develop necessary stubs and drivers for higher-level system testing.<br>3. Examine system-wide issues such as performance and data integrity.<br>4. Conduct heuristic assessments to evaluate the success of intelligent components. |
| **Outputs** | A final Requirement Test Coverage Report document. |

# 5 Project Estimates

## 5.1 Code Size Estimation using Function Points

We calculated unadjusted function point based on the complexity of functions provided by this system. Code size is then estimated by adjusted function point.

### 5.1.1 Unadjusted Function Points

NexusNTU supports the following proposed functions:

User:

- Register a new account (email/phone, OTP)
- Log into the application
- Reset/change password
- View personalised Information Hub (home-country news, NTU announcements)
- Open Quick Links dashboard to launch STP/health/transit pages
- Search campus map and get directions
- Find nearest amenities and save favourites
- Use the AI Chatbot for Q&A (text or image)
- View School/Course Hub
- Currency conversion
- Update profile and settings
- Submit feedback/report issue
- Log out

The measure of unadjusted function points is based on five primary component elements of these functions: Inputs, Outputs, Inquiries, Logical Files, and Interfaces. Each element ranges from Low, Medium, to High complexity. The detailed evaluation of the inputs is as follows:

**Rating Inputs:**

- Account registration: email/phone, password strength, OTP verification (Medium)
- Login: email/phone + one-time password (Low)
- Password reset/change: verification + new password entry (Low)
- Profile update: name, school, programme, year, avatar (Low)
- News preferences: country selection, source toggles (Low)
- Quick Links: hub/tile selection (Low)
- Campus navigation: building search, origin/destination or current-location permission (Medium)
- Amenities finder: search term, category filters, favourite toggle (Medium)
- AI Chatbot: text prompt and optional image upload (Medium)
- Currency converter: amount and currency pair (Low)
- School/Course Hub selector: school/programme/year (Low)
- Feedback/report issue: text, category, optional screenshot (Low)
- Settings: 2FA toggle, locale, notifications (Low)

| Files Type Referenced | Data Elements |
|---|---|

| (FTR) | | | |
|---|---|---|---|
| | 1-4 | 5-15 | Greater than 15 |
| Less than 2 | Low (3) | Low (3) | Average (4) |
| 2 | Low (3) | Average (4) | High (6) |
| Greater than 2 | Average (4) | High (6) | High (6) |

**Rating Outputs**

- Personalised news feed cards (Information Hub)
- Quick Links dashboard tiles and deep links
- Turn-by-turn route view for campus navigation
- Nearest Amenities results and POI detail panes
- School/Course Hub pages with verified links
- AI Chatbot answers and suggested follow-ups
- Currency conversion result display
- Profile/settings update confirmations

| File Types Referenced (FTR) | Data Elements | | |
|---|---|---|---|
| | 1-5 | 6-19 | Greater than 19 |
| Less than 2 | Low (4) | Low (4) | Average (5) |
| 2 or 3 | Low (4) | Average (5) | High (7) |
| Greater than 3 | Average (5) | High (7) | High (7) |

**Rating Inquiries**

- Search buildings/rooms and request directions
- Find nearest amenities with category filters
- Search/filter within Quick Links
- Filter news by country/source
- Ask chatbot questions (text/image) and retrieve answers
- Convert currencies (amount + pair)

| File Types Referenced (FTR) | Data Elements | | |
|---|---|---|---|
| | 1-5 | 6-19 | Greater than 19 |
| less than 2 | Low (3) | Low (3) | Average (4) |
| 2 or 3 | Low (3) | Average (4) | High (6) |
| Greater than 3 | Average (4) | High (6) | High (6) |

**Rating Logical Files (ILF/EIF)**

**ILF:**

- Users: profiles, preferences, auth metadata
- Favourites: saved amenities and links
- Feedback tickets and audit logs
- School/Course Hub configuration store
- Link-Registry cache of verified deep links

**EIF (referenced, cached metadata only):**

- NTU Omnibus shuttle information
- Fullerton Health @ NTU hours/queue links
- One-Stop/SimplyGo/NTU portal pages and contacts

| Record Element Types (RET) | Data Elements | | |
|---|---|---|---|
| | 1 to 19 | 20 - 50 | 51 or More |
| 1 RET | Low (7) | Low (7) | Average (10) |
| 2 to 5 RET | Low (7) | Average (10) | High (15) |
| 6 or More RET | Average (10) | High (15) | High (15) |

**Rating Interfaces**

- Mapbox API for maps, geocoding, routing, and POI search.
- Google Dialogflow (Gemini API) for the AI chatbot.
- Firebase Authentication with JWT over HTTPS for login/OTP.
- Deep links to official NTU systems: Omnibus shuttle, Fullerton Health @ NTU, One-Stop, SimplyGo.
- External news outlets used by the Information Hub.
- Embassy/consulate contacts and visa/immigration portals linked from the Home-Country hub.

| Record Element Types (RET) | Data Elements | | |
|---|---|---|---|
| | 1 to 19 | 20 - 50 | 51 or More |
| 1 RET | Low (7) | Low (7) | Average (10) |
| 2 to 5 RET | Low (7) | Average (10) | High (15) |
| 6 or More RET | Average (10) | High (15) | High (15) |

**Summary of Above Analysis:**

| Element | Complexity | Detail |
|---|---|---|
| **Inputs** | Average | Account registration with OTP |

| | | |
|---|---|---|
| | Low | Login |
| | Low | Password reset/change |
| | Low | Profile update (name, school, year, avatar) |
| | Low | News preferences (country/source) |
| | Low | Quick Links tile selection |
| | Average | Campus navigation: origin/destination or location permission |
| | Average | Amenities finder: search + category filters + favorite toggle |
| | Average | AI Chatbot: text prompt and optional image upload |
| | Low | Currency converter: amount + pair |
| | Low | School/Course Hub selector |
| | Low | Feedback/report issue (text, category, screenshot) |
| | Low | Settings: 2FA toggle, locale, notifications |
| **Logical Files (ILF/EIF)** | Average | Users (profiles, preferences, auth metadata) |
| | Low | Favourites (amenities, links) |
| | Low | Feedback tickets and audit logs |
| | Average | School/Course Hub configuration store |
| | Average | Link-Registry cache of verified deep links |
| **Outputs** | Average | Personalised news feed cards (Information Hub) |
| | Low | Quick Links dashboard tiles and deep links |
| | Average | Turn-by-turn route view for campus navigation |
| | Average | Nearest Amenities results and POI detail panes |
| | Low | School/Course Hub pages |
| | Average | AI Chatbot answers with suggested follow-ups |

| | | |
|---|---|---|
| | Low | Currency conversion result |
| | Low | Profile/settings update confirmations |
| **Inquiries** | Average | Search buildings/rooms and request directions |
| | Average | Find nearest amenities with filters |
| | Low | Search/filter within Quick Links |
| | Low | Filter news by country/source |
| | Average | Ask chatbot questions and retrieve answers |
| | Low | Convert currencies (amount + pair) |
| **Interfaces** | Average | Mapbox API (maps, geocoding, routing, POI) |
| | Average | Google Dialogflow/Gemini for AI chatbot |
| | Average | Firebase Authentication (login/OTP) |
| | Low | Deep links to NTU systems (Omnibus, Fullerton Health, One-Stop, SimplyGo) |
| | Low | External news sources |
| | Low | Embassy/consulate and immigration portals |

Calculation of Unadjusted Function Points:

| Characteristic | Low ×w | Medium ×w | High ×w |
|---|---|---|---|
| **Inputs (EI)** | 9×3 = **27** | 4×4 = **16** | 0×6 = **0** |
| **Outputs (EO)** | 4×4 = **16** | 4×5 = **20** | 0×7 = **0** |
| **Inquiries (EQ)** | 3×3 = **9** | 3×4 = **12** | 0×6 = **0** |
| **Logical Files (ILF)** | 2×7 = **14** | 3×10 = **30** | 0×15 = **0** |
| **Interfaces (EIF)** | 3×5 = **15** | 3×7 = **21** | 0×10 = **0** |
| **Unadjusted FP** | **81** | **99** | **0** |
| **Total = L + M + H** | - | - | **180** |

### 5.1.2. Adjusted Function Points

| Influence Factor | Score | Detail |
|---|---|---|
| Data communications | 5 | HTTPS APIs to Mapbox, Firebase, Dialogflow, external links |
| Distributed functions | 4 | Client–server, cloud functions, CDN, bidirectional sync |
| Performance | 4 | Low-latency navigation/chat; p95 targets during peak hours |
| Heavily used configuration | 3 | Peak morning/semester loads considered |
| Transaction rate | 2 | Predictable daily peaks (commute, class changes) |
| Online data entry | 4 | Registration, feedback, searches, filters, OTP |
| End-user efficiency | 4 | Quick Links, caching, saved favorites, shortcuts |
| Online data update | 3 | Updates to profiles, preferences, favorites |
| Complex processing | 3 | Routing + LLM chatbot + geocoding |
| Reusability | 3 | Shared UI components and service wrappers |
| Installation ease | 2 | PWA install; standard mobile/web setup |
| Operational ease | 4 | Monitoring, backup, alerting; minimal operator actions |
| Multiple sites | 1 | NTU campuses |
| Facilitate change | 3 | Config flags, modular services, CI/CD |

| Scoring (0 – 5) |
|---|
| 0 = No influence |
| 1 = Insignificant influence |
| 2 = Moderate influence |
| 3 = Average influence |
| 4 = Significant influence |
| 5 = Strong influence |

### 5.1.3 Lines of Code

NexusNTU is a React + TypeScript PWA. Using Capers Jones, **TypeScript ≈ 53 LOC/FP**.

Adjusted FP (from §5.1.2): **198**
Estimated LOC = **198 × 53 = 10,494**.

## 5.2 Efforts, Duration and Team Size Estimation

To estimate the effort and duration required for the project, we use function points as the basis to calculate Effort, Duration, Team Size and finally the schedule. The estimates are expanded to account for project management and extra contingency time to obtain the total average effort estimates. From these averages, the duration of each work package in working days is estimated based on the following calculations.

- All members are expected to work 7 days a week.
- Effort = Size / Production Rate = (3846 LOC) / (31 LOC/PD)[1] = 124.065 PD
- Duration = 3 × (Effort)$^{1/3}$ = 3 × (124.065)$^{1/3}$ = 15 Days
- Initial schedule = 15 Days / 7 days a week = 2.14 Weeks
- Team size = 124.065 PD / 15 D = 9 Persons
- Working hours include 4 hours in a working day.
- Total person-hours (PH) = 124.065 PD × 4 hours = 496.26 PH

### 5.2.1 Distribution of Effort

| 1990's Industry Data | Work Package | Distribution | Estimates |
|---|---|---|---|
| Preliminary Design 18% | Project Proposal | 4% | 19.85 |
| | Requirement Elicitation | 1% | 4.96 |
| | Software Quality Assurance | 3% | 14.89 |
| | System Requirement Specification | 6% | 29.78 |
| | Risk Management | 1% | 4.96 |
| | Project Plan | 3% | 14.89 |
| Detailed Design 25% | User Interface Design | 7% | 34.74 |
| | Technical Architecture | 11% | 54.59 |
| | Data Modelling | 7% | 34.74 |
| Code & Unit Testing 26% | Coding & Unit Testing | 26% | 129.03 |
| Integration & Test 31% | Integration & System Testing | 31% | 153.84 |
| | **Extrapolated Total Effort:** | | **496.26** |
| | 2% for project management | | 9.93 |
| | 3% for contingency | | 14.89 |
| | **Total Effort:** | | **521.08** |

These duration estimates assume that each team member works an equal amount on any given work package.

[1] Lines of code per Person Day statistics based on Industrial Benchmarks, 1997: 31 LOC/PD for United States; 62 LOC/PD for Canada

## 5.3 Cost Estimates

**Hardware:**

**Developer workstations:**

| | |
|---|---|
| **7** - Dell Precision Workstation 330 | Total<br>$0.00 |
| Pentium IV 1.4GHz single processor | |
| 256 MB RAM | |
| 20.8GB IDE drive | |

**Software:**

**Development Tools:**

| | |
|---|---|
| Firebase | $0.00 |
| Flutter | $0.00 |

**Documentation/version control tools:**

| | |
|---|---|
| GitHub | $0.00 |
| Tortoise SVN | $0.00 |
| Diagrams.net | $0.00 |
| Visual Paradigm | $0.00 |
| Figma | $0.00 |
| Android Studio | $0.00 |
| Trello | $0.00 |
| Team Gantt | $0.00 |
| MediaWiki | $0.00 |

**Software License Provided by Third Party:**

| | |
|---|---|
| Microsoft Office | $0.00 |

**Other Resources:**

**Staff:**

| | |
|---|---|
| 7 Employees with 1000 working hours with $18.00/hour | $18,000 |

**Stationery:**

| | |
|---|---|
| Paper, photocopying and other miscellaneous costs | $50.00 |

**Total:** $18,050

# 6 Product Checklist

The plan is that the items listed below will be delivered by the stated deadlines.

| Project Deliverable | Estimated Deadline |
|---|---|
| Meeting Minutes | Every meeting |
| Trello Backlog | Updated after every lab |
| Project Proposal | Fri, 5 Sep 2025 |
| Requirements Elicitation Summary | Thu, 11 Sep 2025 |
| Quality Plan (SQA Plan) | |
| Use Case Model & Description | Thu, 1 Sep 2025 |
| System Requirement Specification (SRS) | Fri, 19 Sep 2025. |
| Risk Management Plan | Wed, 17 Sep 2025. |
| Project Plan (this document) | Wed, 1 Oct 2025 (v1.0 approved). |
| Prototype (Lab 4 demo build) | Wed, 15 Oct 2025. |
| Configuration Management Plan | Mon, 20 Oct 2025. |
| Change Management Plan | Tue, 28 Oct 2025. |
| Release Plan | Tue, 28 Oct 2025. |
| Presentation Slides (Lab 5) | Wed, 29 Oct 2025. |
| Test Plan (Integration/System) | Wed, 22 Oct 2025. |
| Test Cases & Requirements Test Coverage Report | Tue, 28 Oct 2025. |
| CMMI Level-2 Definition Package | Tue, 28 Oct 2025. |
| Peer Review Report | Fri, 7 Nov 2025. |

# 7 Best Practices

Our project will adhere to the following best practices to ensure a high-quality outcome:

- **Standardized Documentation:** All documents will follow a consistent format.
- **Thorough Requirements Analysis:** Requirements will be checked for ambiguity, completeness, and consistency.
- **Simplicity in Design:** We will strive to minimize complexity in interfaces between modules and team members.
- **Visibility and Transparency:** Progress will be openly tracked, and code will be available for peer review.
- **Configuration Management:** A version control system (Git) will be used to manage all source code and documentation, with clear versioning and change history.
- **Accurate Estimation:** We will carefully estimate time and effort for all tasks, including testing and documentation.
- **Code Reviews:** Peer code reviews will be a standard practice to identify defects early.
- **Comprehensive Testing:** The testing strategy will include unit, integration, system, and user acceptance testing.
- **Good Coding Practices:** We will follow established design patterns, commenting conventions, and coding principles.

# 8 Risk Management

The following key risks have been identified for the NexusNTU project:

| Risk | Impact Severity | Probability | Mitigation Strategy |
|---|---|---|---|
| Unable to Meet Project Deadline | High | Medium | Follow Agile methodology with sprints to track progress closely. Have regular meetings to identify and resolve blockers early. |
| Dependency on External APIs | High | Medium | Develop fallback mechanisms (e.g., show cached data, deep links) for when an API is down. Abstract API integrations to allow for easier replacement if needed. |
| Feature Creep | High | Medium | Strictly adhere to the defined scope in the SRS. All new feature requests must go through a formal change request process and be approved by the Project Manager. |
| Team Member Unavailability | Medium | Medium | Maintain clear and updated documentation so other team members can take over tasks. Foster knowledge sharing within the team. |
| System Not Meeting Requirements | High | Low | Conduct User Acceptance Testing (UAT) with the prototype early in the process. Maintain a requirements traceability matrix to ensure all specifications are met. |
| Data Security Breach | High | Low | Use a trusted authentication service (Firebase). Avoid storing any sensitive PII. Follow security best practices for web development (e.g., input sanitization, secure API keys). |
| Broken Deep Links | Medium | High | Implement a link-checking mechanism or a user-reporting feature for broken links. Admins will regularly verify and update the links in the database. |

# 9 Quality Assurance

Quality assurance will be a continuous process throughout the project. The detailed procedures will be outlined in the Quality Plan. Our testing methodologies include:

- **Unit Testing:** To test individual functions and components in isolation.
- **Integration Testing:** To ensure that different modules (e.g., authentication and user profile) work together correctly.
- **System Testing:** To test the application as a whole, ensuring all components are properly integrated.
- **Validation Testing:** To confirm that the software meets the business requirements and user needs defined in the SRS.
- **User Acceptance Testing (UAT):** The prototype will be tested by target users (international students) to gather feedback on usability and functionality.

Testing will cover both System Function (e.g., ensuring login works) and Algorithmic/Heuristic Function (e.g., ensuring the chatbot provides relevant answers and navigation provides accurate routes). We will use a combination of realistic and extreme test cases to ensure the system is robust.

# 10 Monitoring & Control

The following procedures will be used to monitor and control the project's progress:

- **Quantitative Measurement:** We will track the effort (in person-hours) spent on each work package against the estimates in this plan to measure progress.
- **Risk Identification:** The risk register will be reviewed and updated during weekly meetings to proactively manage potential issues.
- **Regular Progress Reviews:** Weekly team meetings will be held to review the status of all tasks, discuss challenges, and plan for the next sprint.
- **Timeline and Task Management:** The Work Breakdown Structure and Gantt chart will be used to track progress against the project timeline. A project management tool like Trello will provide a fine-grained view of task status.