System Requirement Specifications (SRS)

1 Table of Contents

2 Problem Statement

International students at Nanyang Technological University (NTU) lack a centralized digital platform for managing essential onboarding and integration tasks. Critical information and services—including immigration processing, health screenings, campus navigation, and resource discovery—are fragmented across multiple disparate university websites, physical offices, and third-party systems. This lack of system integration creates significant inefficiencies and inconvenience, leading to delayed task completion, redundant information seeking, and a high potential for error during the students' critical transition period.

3 Overview

3.1 Background

As Nanyang Technological University (NTU) continues to attract a large and diverse international student body, the decentralized and manual nature of the onboarding process has become overwhelmingly complex and inefficient. The current process of recording, retrieving, and communicating critical information is fragmented across numerous university websites, physical offices, and paper-based checklists. Routine inquiries from new international students regarding immigration, housing, and campus navigation cannot be readily answered through a single source.

Conflicts and confusion arising from disparate information sources must be manually addressed by administrative staff, leading to significant delays. During the peak arrival period each semester, students face an overload of disconnected tasks—from Student's Pass processing to learning the campus bus routes. With the current fragmented processes and the mounting anxiety and confusion expressed by students and support staff, there is an urgent need to develop a centralized, student-centric digital hub for the university's international student community.

3.2 Overall Description

In essence, the current digital infrastructure provides multiple, unintegrated interfaces to various back-end university systems. Though the individual back-end databases for student records, academic scheduling, and campus services can reliably handle their specific functions, there is no unified system that aggregates this information into a cohesive onboarding journey for international students.

The university's core student data is managed by a central Student Information System (SIS). However, other critical systems—such as the campus mobility service, the Fullerton Health clinic system, and immigration compliance databases—operate as independent platforms. These systems of record lack the integration layer or APIs necessary to provide a consolidated view to the end-user. The proposed all-in-one web application will serve as this essential integration and presentation layer, interfacing with these disparate systems to provide a single, authoritative source of information and task management for incoming international students.

4 Investigation & Analysis Methodology

4.1 System Investigation

The current NTU international student onboarding experience is fragmented, requiring students to navigate multiple disconnected systems: Student's Pass applications, campus health services, shuttle bus routes, academic portals, and local amenities. This results in inefficiency, information overload, and delays in integration.

Our proposed solution, NexusNTU, centralises these workflows into a single web platform. It aggregates verified university content (via deep links), real-time updates (e.g., bus, clinic queues), and support resources (chatbot, quick links) in a secure and user-friendly application. The system adopts a Model–View–Controller (MVC) architecture, with a React PWA frontend, Node/Express backend, and MongoDB database. Firebase Authentication with JWT ensures secure access.

This investigation confirms that while NTU provides all required services individually, students lack a consolidated hub. NexusNTU fills this gap with a reliable, scalable, and maintainable platform.

4.2 Analysis Methodology

4.2.1 Feasibility study and requirements elicitation

We will establish a cross-functional team of developers, UX designers, and international student representatives.Surveys and Interviews: Capture student pain points on onboarding, mobility, and admin processes.Focus Groups: Validate proposed features like the Information Hub, Navigation, and AI chatbot.Feasibility & Risk Assessment: Evaluate integration feasibility (e.g., deep links to NTU One-Stop, Omnibus shuttle, Fullerton Health), adoption risk (student readiness), and cost of cloud deployment.

4.2.2 System analysis and requirements specification

4.2.2.1 Object-Oriented Problem Analysis

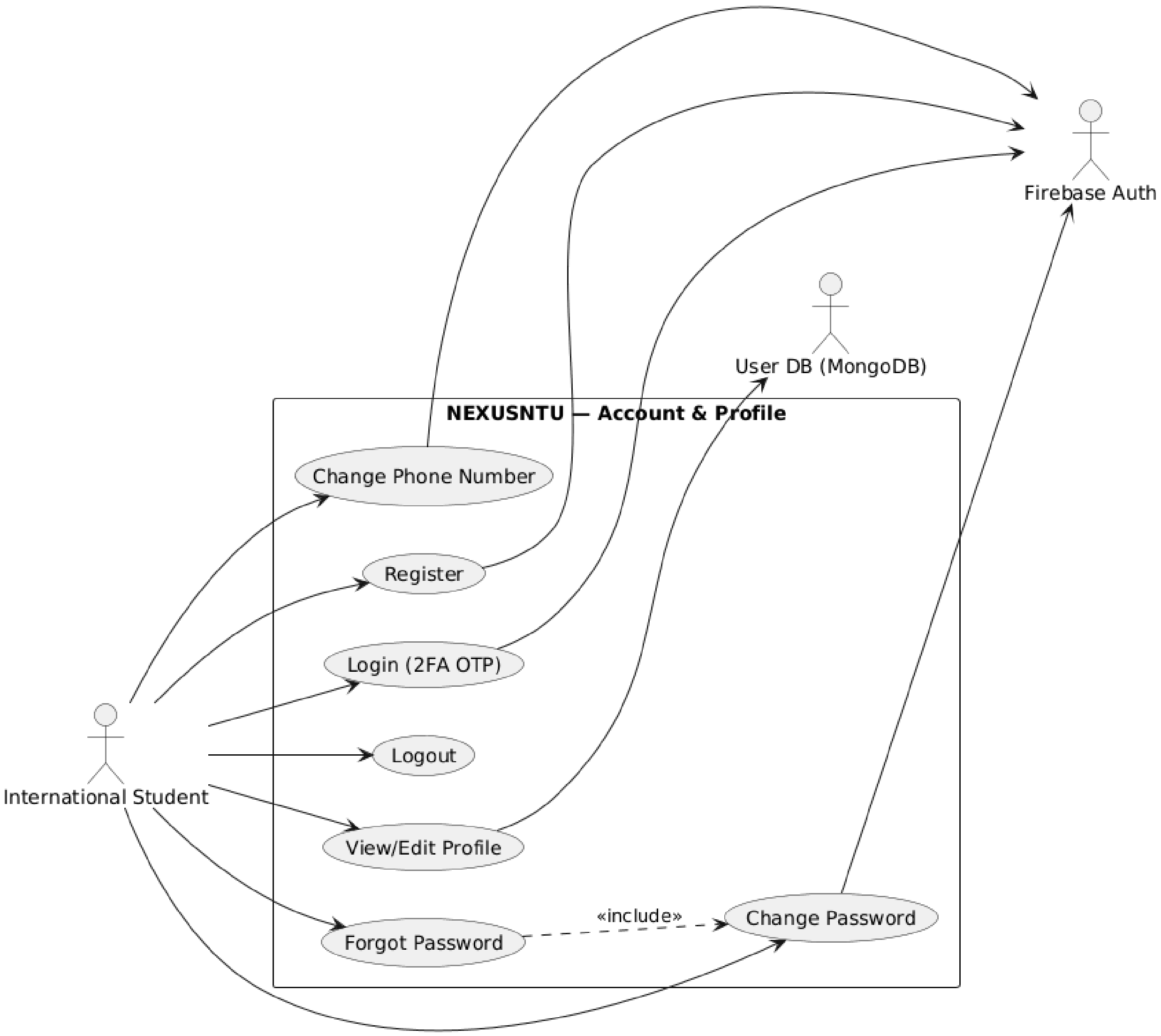
- Using UML modelling, the system is analysed around key entities:
- Student (profile, preferences, language, school hub)
- Hub/Module (NTU Hub, School Hub, Singapore Hub, Home Country Hub)
- Content Link (deep links, metadata, expiry dates)
- Chatbot Interaction (FAQ, intent, knowledge base)
- Navigation/Amenities (map, routes, POIs, queues)

4.2.2.2 Scope and Limitations

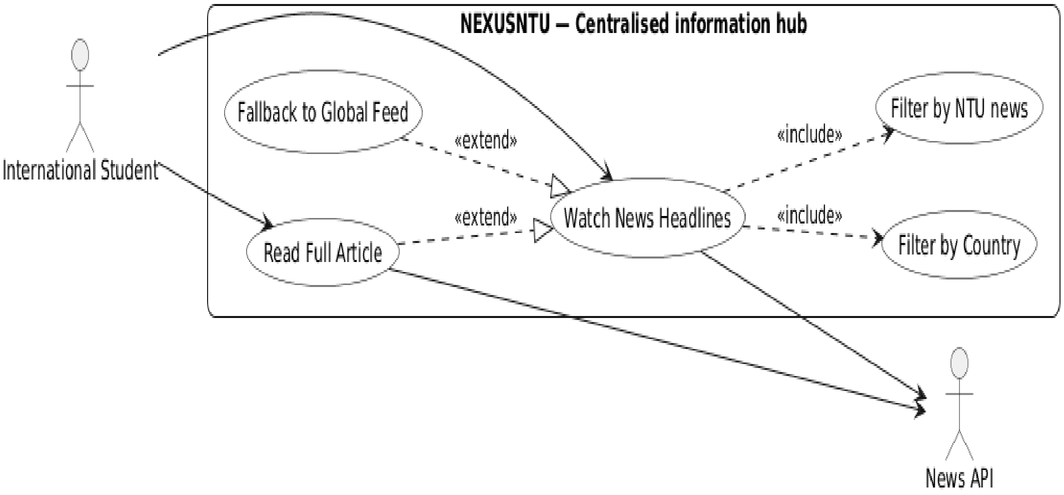
- Business Analysis: Define ownership (NTU), scope (international student onboarding), and budget (student project scale).
- Requirement Analysis: Secure logins (Firebase), functional modules (hub, chatbot, navigation), and non-functional targets (≤5s load, 98% uptime, WCAG AA compliance).
- Data Analysis: Store only non-sensitive metadata (no passport/FIN), rely on cached and verified links.
- Process Analysis: Map onboarding and daily workflows, minimise duplicated effort, ensure failover on upstream outages.
- Application Architecture: MVC layered stack (React PWA, Node/Express, MongoDB), REST API communication, feature-flagged releases.

4.2.3 Object-oriented design using UML

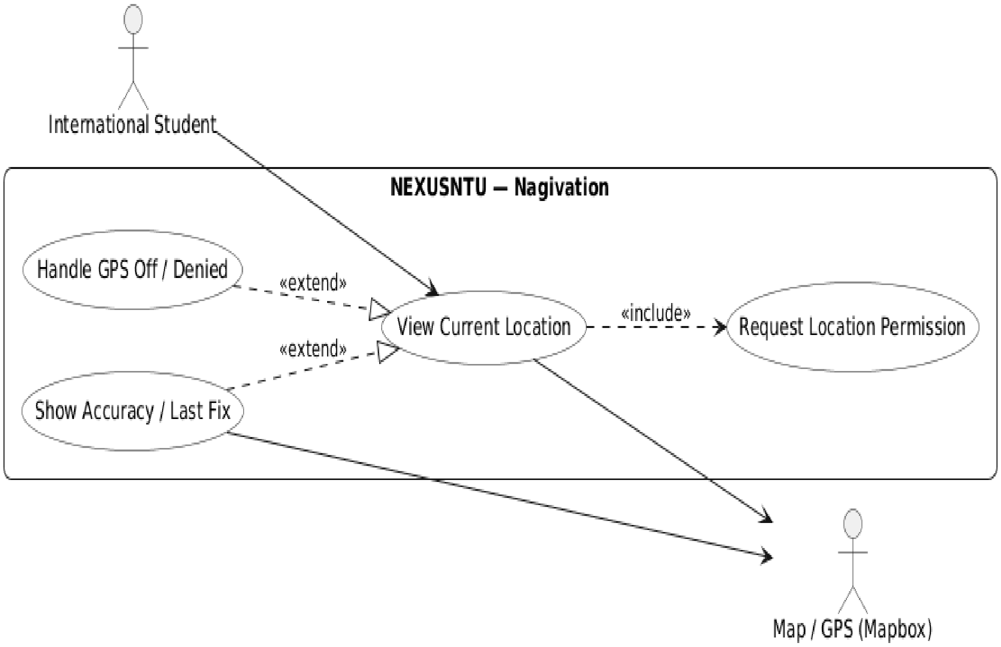
1. Account and profile



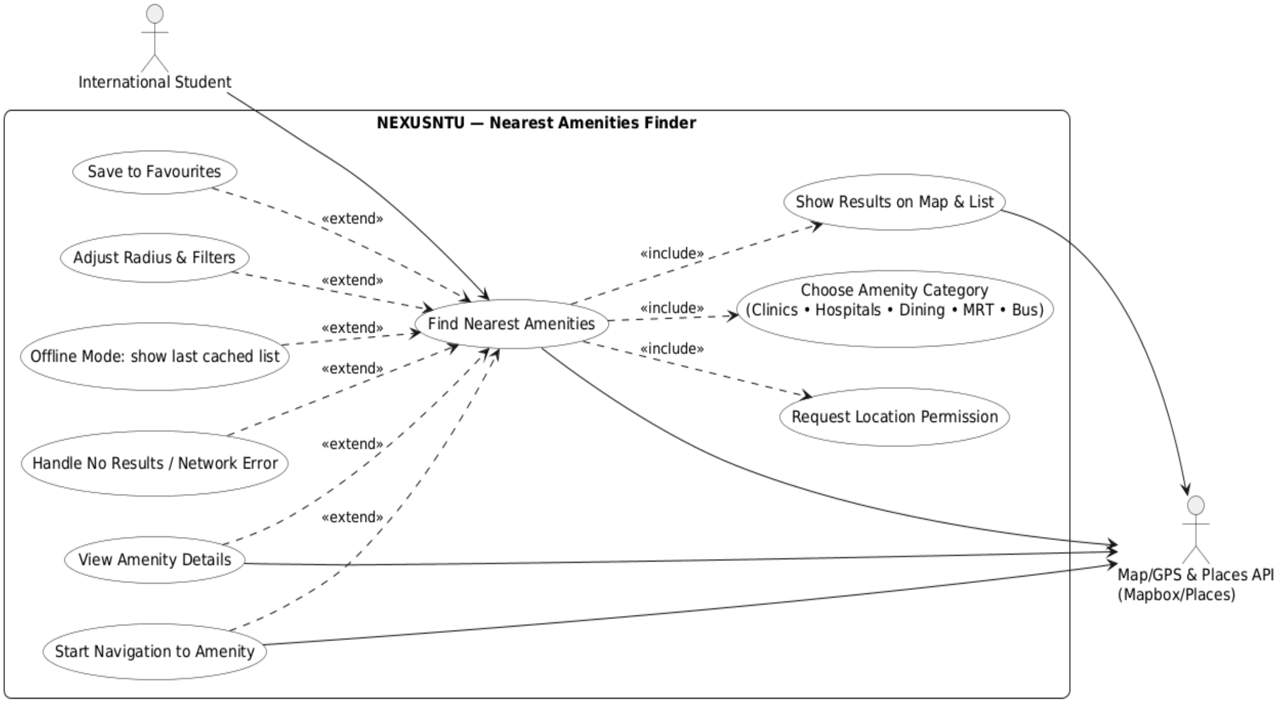
2. Centralised information hub



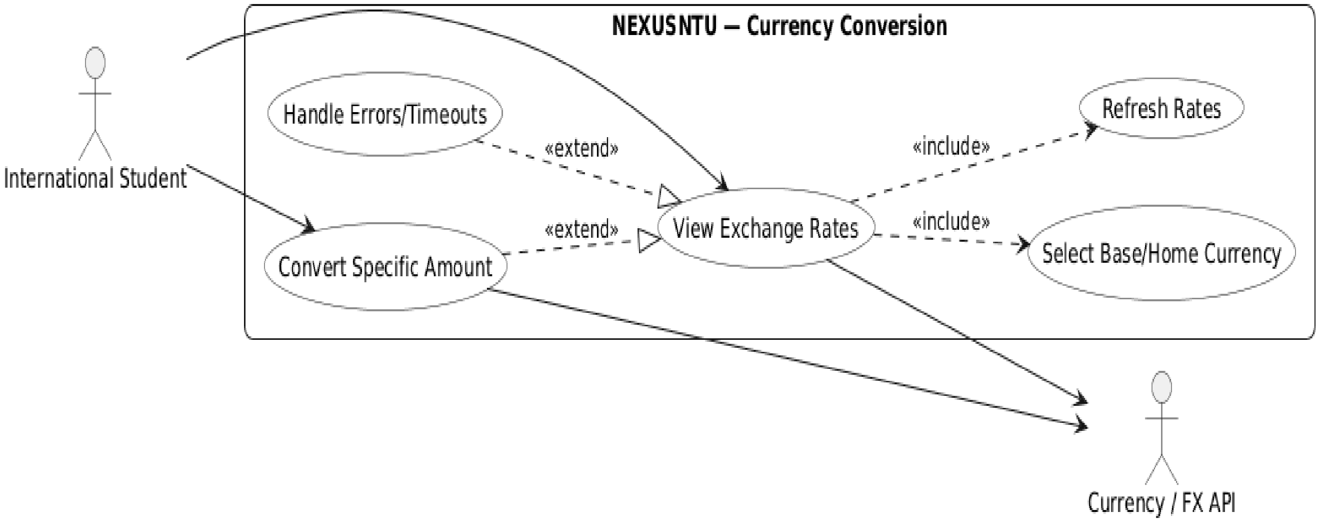
3. Navigation



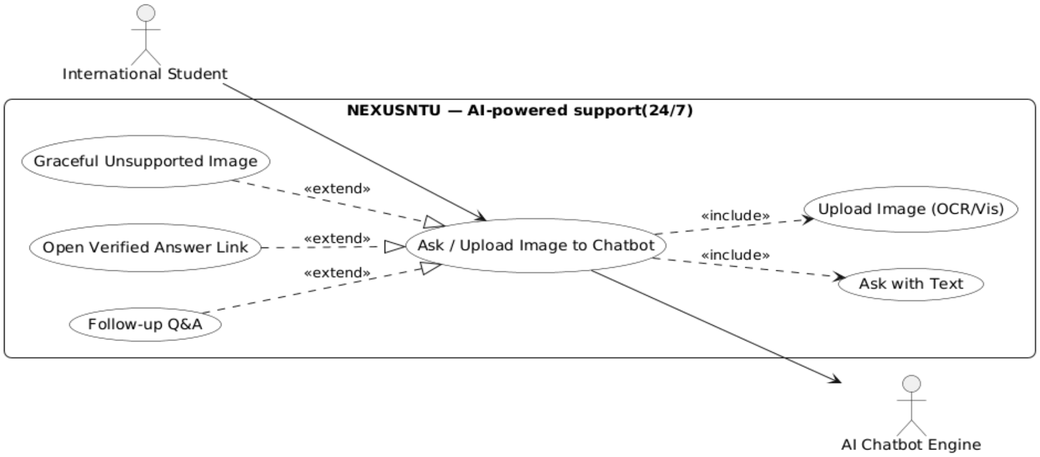
4. Nearest Amenities Finder


International Student
NEXUSNTU — Nearest Amenities Finder
Save to Favourites
Adjust Radius & Filters
Offline Mode: show last cached list
Handle No Results / Network Error
View Amenity Details
Start Navigation to Amenity
«extend»
«extend»
«extend»
«extend»
«extend»
«extend»
Find Nearest Amenities
«include»
«include»
«include»
Show Results on Map & List
Choose Amenity Category (Clinics • Hospitals • Dining • MRT • Bus)
Request Location Permission
Map/GPS & Places API (Mapbox/Places)

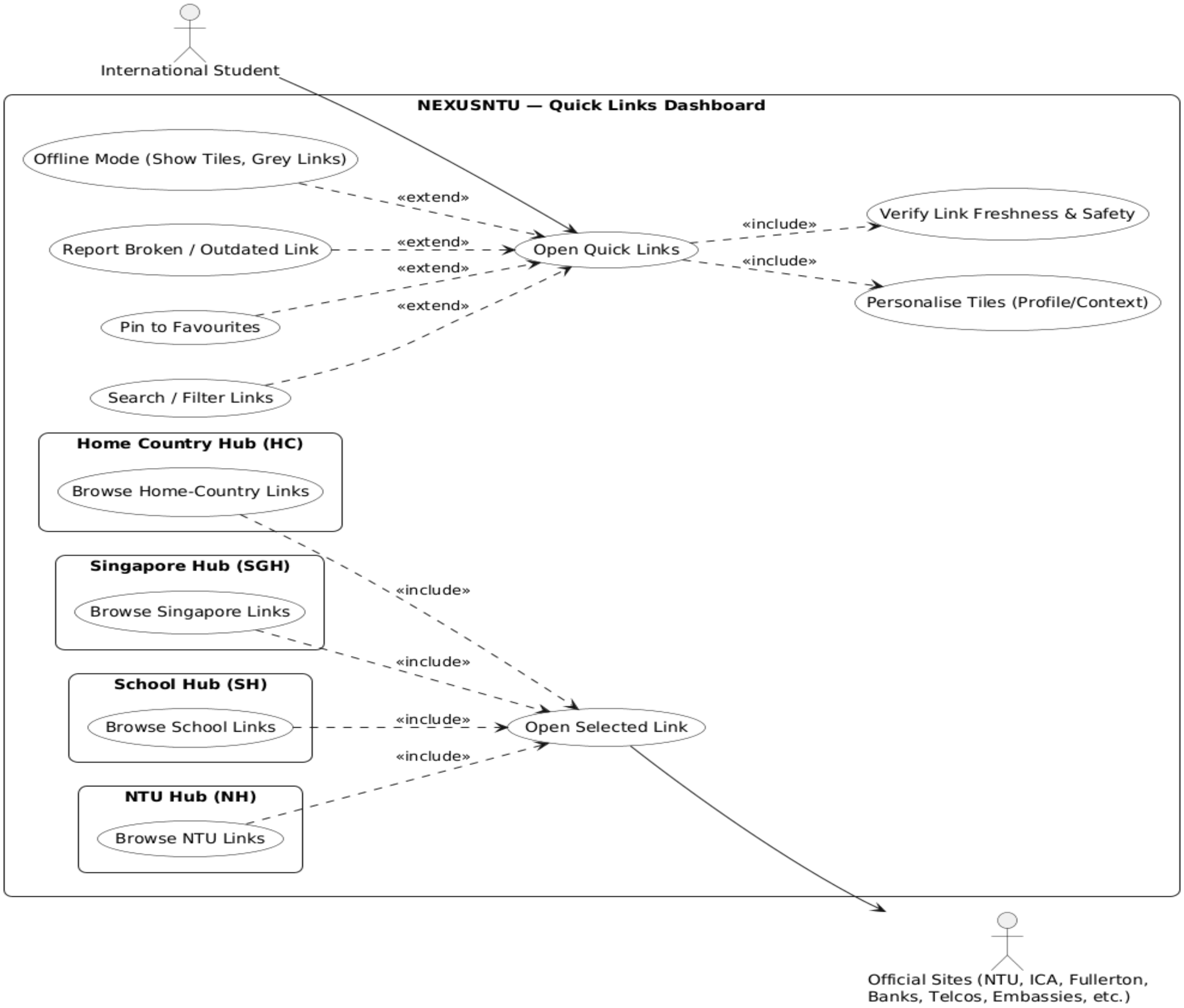
5. Currency Conversion



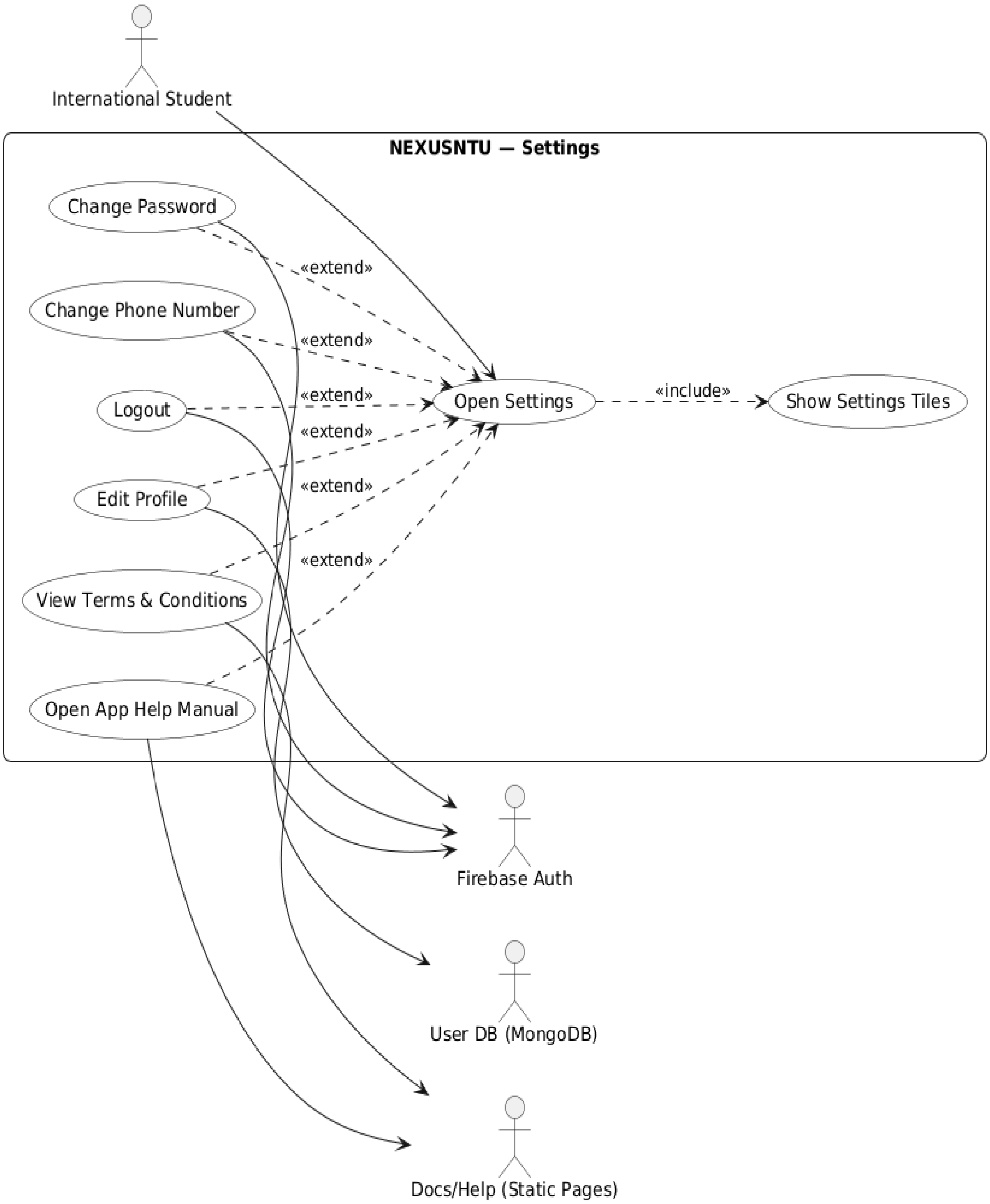
6. AI-powered support (24/7)



7. Quick links


International Student
NEXUSNTU — Quick Links Dashboard
Offline Mode (Show Tiles, Grey Links)
Report Broken / Outdated Link
Pin to Favourites
Search / Filter Links
«extend»
«extend»
«extend»
«extend»
Open Quick Links
«include»
«include»
Verify Link Freshness & Safety
Personalise Tiles (Profile/Context)
Home Country Hub (HC)
Browse Home-Country Links
Singapore Hub (SGH)
Browse Singapore Links
School Hub (SH)
Browse School Links
NTU Hub (NH)
Browse NTU Links
«include»
«include»
«include»
«include»
Open Selected Link
Official Sites (NTU, ICA, Fullerton,
Banks, Telcos, Embassies, etc.)

8. Settings


International Student
NEXUSNTU — Settings
Change Password
Change Phone Number
Logout
Edit Profile
View Terms & Conditions
Open App Help Manual
«extend»
«extend»
«extend»
«extend»
«extend»
«extend»
Open Settings
«include»
Show Settings Tiles
Firebase Auth
User DB (MongoDB)
Docs/Help (Static Pages)

4.2.4 Prototyping

Using the Object-Oriented Rapid Prototyping (OORP) method, a limited but functional NexusNTU prototype will be developed. It will demonstrate key modules such as the Quick Links Dashboard, Campus Navigation/Amenities Finder, and a basic AI Chatbot.The prototype will use a React PWA frontend, Node/Express backend, and MongoDB for data storage, with Firebase Auth for secure login. Its purpose is to serve as a proof of concept, allowing stakeholders and international students to provide feedback before full implementation.

5 Constraints

5.1 Scalability

NexusNTU is a course-project system with clear limits on what it can and can not do. We aim for a first-load time of 5 seconds or less on typical campus networks and an uptime of about 98%. Anything more than this is not part of the semester deliverable. Capacity planning is based on hosting students and small groups, not rolling out to the whole university.

5.2 Data and Function Mapping

The app only stores official resources through deep links and light metadata. We don't keep sensitive records like passports or FINs. Links may break when upstream sites change their structure or content until they are remapped. Our cache only has verified snapshots.

5.3 Proprietary hardware and software

The solution is built on a specific stack: React PWA for the front end, Node/Express for the API, MongoDB for the data, and Firebase Authentication for safe sign-in. These third-party services and licenses/quotas limit the availability and features.

5.4 Batch updates vs. (close) Real-time updates

We use periodic fetch + cache and display fallbacks when sources are slow or offline. While some information (such as transport and clinic queues) can be updated often, end-to-end "real-time" is not guaranteed when data comes from external sites.

5.5 Project Schedule

In order to illustrate important flows (Quick Links, Navigation/Amenities, basic Chatbot), this release is a proof-of-concept created using Object-Oriented Rapid Prototyping. The work is time-boxed to the course timeline; it is not a production hardening effort.

6 Operational Requirements

6.1 Help Desk Support

Users can reach us via in-app FAQ and an email contact. We reply during weekdays 09:00–21:00 SGT on a best-effort basis. For outages, we will add a notice in-app.

6.2 Application Services and Technical support

The team maintains source code in a shared repo and fixes bugs during weekly sprints. Production logs are reviewed after incidents. Routine maintenance (library updates, small fixes) happens outside class peak hours.

6.3 Administration Features

Role-based access: Student (own profile) and Admin (manage links/content and settings). Sign-in is handled by the project's auth service; each API call verifies the session. Admin pages are access-restricted.

6.4 System Interface independent of VRU

This project does not use a VRU. The online system works independently of external info sources by caching key pages and deep-linking to official sites when online. If an external site is down, cached info is shown with a "check official site" prompt.

6.5 System hardware fail over and routine back up

We rely on managed cloud hosting. A simple automated backup/export runs once per day (or at the end of each sprint). If needed, we can redeploy and restore within 1 business day on a best-effort basis.

6.6 Audit Trail

Admin actions that change data (e.g., updating links or settings) are recorded with who, what, and when in an audit log. Access to the log is limited to admins. Logs are retained for 60 days.

7 Functional Requirements

The online registration system is a "self-service style" system that shall initially address the student registration needs.

7.1 User Account and Authentication

This module covers user registration, login, and profile management.

7.1.1 User Registration

- The system shall allow new users to create an account using a unique username or email address.
- The system shall validate that the chosen username/email is not already in use.
- The password entry field shall mask the characters entered.
- The system shall enforce password strength requirements and provide real-time feedback to the user via a strength meter.
- The system shall require phone number verification via a One-Time Password (OTP) sent to the user's mobile device, with support for international country codes.

7.1.2 User Login

- The system shall allow registered users to log in using their credentials and a 6-digit OTP for two-factor authentication.
- The OTP shall be valid for a 5-minute window.
- The system shall provide an option to resend the OTP if it is not received.
- The system shall implement an account lockout mechanism after a configurable number of failed login attempts.

7.1.3 Profile and Session Management

- The system shall allow a logged-in user to explicitly log out, terminating their session.
- The system shall provide a "Forgot Password" workflow to allow users to securely reset their password.
- A logged-in user shall be able to change their password from within their account settings.
- Users shall be able to view and edit their profile information, including their display name and profile avatar.
- A user changing their registered phone number must complete a re-verification process via OTP.

7.2 Centralised Information Hub (News Viewer)

This module covers the aggregation and display of personalized content.

7.2.1 Content Aggregation

- The system shall aggregate and display news articles from top media outlets of the user's specified home country.
- The system shall fetch and display official university-wide announcements from NTU.
- The system shall feature curated guides and articles on local Singaporean culture and life at NTU.

7.2.2 Content Display

- The content feed shall be personalized based on the user's home country, as defined in their profile settings.
- All aggregated content (news, announcements) shall be displayed in reverse chronological order.

7.3 Campus Navigation

This module provides tools to help users navigate the NTU campus via public transport and on foot.

7.3.1 Campus Bus Integration

- The system shall provide a clear and accessible deep link to download the official NTU Omnibus

application for both iOS and Android platforms.
- The system shall briefly describe the purpose of the NTU Omnibus app (e.g., for real-time bus tracking and route information).

7.3.2 Navigation

- The system shall display an interactive, zoomable map of the NTU campus.
- The system shall request and utilize the user's device GPS to show their current location on the campus map.
- The system shall generate walking routes from the user's current location (or a user-selected starting point) to their chosen destination.
- The system shall display the generated route visually on the map and provide simple, turn-by-turn walking directions.

7.4 Nearest Amenities Finder

This module helps users locate essential services near them.

7.4.1 Amenity Discovery

- The system shall allow users to find nearby amenities, categorized by type (e.g., MRT stations, bus stops, eateries, hospitals, clinics).
- The search results shall be displayed on a map and in a list, sorted by proximity to the user's current location.

7.4.2 Personalization

- Users shall be able to save specific amenities to a "Favorites" list for quick and easy access.

7.5 AI-Powered Support (Chatbot)

This module provides 24/7 automated assistance.

7.5.1 Query Processing

- The system shall provide a conversational interface for users to ask questions.
- The chatbot shall answer queries based on a knowledge base of frequently asked questions covering campus life, visa procedures, and administration.
- The chatbot shall accept both text-based and image-based inputs from the user.
- The chatbot shall be capable of handling multi-turn conversations and follow-up questions.

7.5.2 Error and Escalation Handling

- When the chatbot cannot understand or answer a query, it shall respond with a graceful message indicating its limitation.
- The chatbot shall provide suggestions for alternative resources (e.g., links to relevant offices, contact information) when a query is out of scope.

7.6 Quick Links Dashboard

This module provides a centralized repository of verified, essential links.

7.6.1 Dashboard Structure

- The system shall present a dashboard of links organized into four distinct hubs: NTU Hub, School Hub, Singapore Hub, and Home Country Hub.

7.6.2 NTU Hub

- The NTU Hub shall provide categorized links to university-wide services, including Academic & Administrative Tools, Student Services & Support, Campus & Facilities, IT & Digital Setup, and Communication & Community resources.

7.6.3 School Hub

- The School Hub's content shall be automatically personalized based on the user's school and programme information in their profile.
- The system shall allow users to manually switch to view the hubs for other schools.
- The system shall implement role-based visibility, displaying certain links only to eligible users (e.g., Final Year Project links for Year 3 and 4 students).

7.6.4 Singapore Hub

- The Singapore Hub shall provide categorized links essential for living in Singapore, including Immigration & Entry, Housing, Healthcare, Transport, and Wellbeing & Safety.

7.6.5 Home Country Hub

- The Home Country Hub shall provide links specific to the user's nationality, including embassy contacts, home country visa portals, and local student community groups.

7.7 Utility Suite

This module includes supplementary tools for user convenience.

7.7.1 Currency Converter

- The system shall provide a tool for currency conversion between the Singapore Dollar (SGD) and the user's home country currency.

8 Input Requirements

8.1 User Credentials and Profile Data

8.1.1 User Identifiers

- The system shall accept a unique email address or username from a new user during account registration.

8.1.2 Authentication Factors

- The system shall accept a user-defined password. The input field shall be masked.
- The system shall accept a user-provided mobile phone number, including support for international country codes, for verification purposes.
- The system shall accept a 6-digit numerical OTP entered by the user for phone number verification and two-factor authentication during login.

8.1.3 Profile Information

- The system shall accept user-entered text for their display name.

- The system shall accept an image file upload (e.g., JPEG, PNG) for the user's profile avatar.
- The system shall require the user to select their home country, school, and programme from pre-defined lists to enable personalization.

8.2 Search Queries and Location Data

8.2.1 Location Coordinates

- Upon user consent, the system shall accept GPS coordinates from the user's mobile device to determine their current location for navigation and amenity finder functions.

8.2.2 Destination/Amenity Queries

- The system shall accept user-entered text strings for searching campus locations (e.g., "LT2A," "The Hive") and nearby amenities (e.g., "clinic," "canteen").

8.3 Chatbot and Conversational Inputs

8.3.1 User Queries

- The system shall accept user inputs of natural language questions typed by the user into the chatbot interface.

8.4 User Selections and Configuration

8.4.1 Amenity Categories

- The system shall accept user selections from a pre-defined list of amenity types (e.g., MRT stations, bus stops, eateries) to filter search results.

8.4.2 Currency Conversion Values

- The system shall accept a numerical value representing the monetary amount to be converted and user selections for the source and target currencies.

8.4.3 Favorite Selections

- The system shall accept a user's command (e.g., a button click) to save a specific amenity to their "Favorites" list.

8.4.4 Content Navigation

The system shall accept a user's command to manually switch between different school hubs within the Quick Links Dashboard.

9 Process Requirements

The online registration system must support the following inherent process requirements to ensure reliability, accuracy, and performance.

9.1 DB2 transaction

The system must be able to send, receive and trigger transactions to the DB2 registration database system.

9.2 Data integrity

The system shall commit transactions that are successfully completed and roll back unfinished or timed-out transactions to maintain data consistency.

9.3 Data validation

The system shall implement validation and error-handling routines to gracefully manage data errors originating from both the user interface and the back-end database processing.

9.4 Performance

The system shall resolve potential locking issues and support concurrent use on a 24×7 basis. It shall also send, receive, and display user messages to enhance the overall user experience.

9.5 Data repository

The system shall maintain the existing DB2 registration database as the main repository of data.

10 Output Requirements

10.1 Transaction Summary and Confirmation

What end-users must see from the system following any action that modifies their state:

- On-screen confirmation (toast + detail row). To modify a profile, change a password or email, save or remove favorites, report a broken link, create a chatbot ticket, or save an amenity or route, display:
  - Action
  - Object (name/ID)
  - Result (success/failure)
  - Timestamp (SGT)
  - optional Ref ID.

  Provide an Undo for up to 10 minutes where it is safe (such as unfavourite or revert profile field).
- End-of-flow summary pages. A readable checklist of inputs and results with Save as PDF and Copy link is the end of multi-step flows (such as first-time setup or bulk link import for administrators).
- Out-of-band confirmations (critical only). Send a brief email confirmation with the action, timestamp, device/platform fingerprint, and support link for security-sensitive changes (password, email, or 2FA updates).

- Activity log (per user). A chronological view of the last 30 days' worth of user-visible actions (filterable by type) can be found in Settings → "Recent Activity". Keeping track of read-only IDs that help support ticket resolution.

Acceptance criteria:

- P95 takes ≤ 2 seconds after the action is finished to provide a confirmation.
- All confirmations use plain language and meet WCAG AA contrast requirements; icons never replace text.
- Each item has a "Need help?" link that opens the appropriate Contact or FAQ.

10.2 Exception Reports

How we surface problems to users and admins, without exposing internals:

- Validation errors (forms). Inline, field-specific messages plus a compact banner summary at the top. Each error has a stable code (such as ACC-E102) and a short fix hint.
- Upstream outage/fallback banners. When an external source is slow/offline (such as a bus app, clinic queues, or school site):
  - Show a module-scoped banner with: current status, last successful refresh, cache age, and next retry window.
  - Fall back to cached content (labelled "cached") or a verified deep link to the official site
- Broken/dead link report. For any link that returns 4xx/5xx or is user-reported:
  - Show the last verified date, the observed error, and suggested alternatives if available.
  - Offer a "Report issue" action that files a ticket with page context and user agent.
- Security/Session notices. Clear outputs for lockouts, OTP failures, expired sessions, or suspicious login—include the reason and next action (resend OTP, reset password, or contact support).
- Data discrepancy warnings. If two official sources conflict (such as different office hours), show a neutral notice and surface both sources with dates.

Acceptance criteria.

- All error banners provide a one-line cause and one next step; no stack traces.
- Fallback/cached items always display "Last updated" and Source.
- User-submitted "broken link" tickets auto-acknowledge within 10s and appear in Admin → Issues.

10.3 Administrative Reports and Summaries

Periodic/ad hoc outputs for admins and stakeholders. Accessible from Admin → Reports with CSV export.

- Usage overview. DAU/WAU/MAU; top modules (Quick Links, Navigation, Amenities, Chatbot); top-clicked links; top zero-result searches; new vs returning users.
- Content governance.
  - Links nearing verification expiry (such as 30/7/1 day), links flagged broken, and user reports
  - by category and source domain.
  - Change history snapshot (who changed what link/label and when).
- Support & feedback. Ticket volumes, categories, P50/P90 time-to-first-response, and time-to-resolution; FAQ deflection rate.
- System health (non-sensitive). Cache refresh job success rate, average response times for key APIs (app layer only), and uptime window summary appropriate to the course scale.
- Compliance snapshot. Counts of accounts with 2FA enabled; accessibility spot-check status for the latest release (AA checks done/failed).

Acceptance criteria

- All admin reports export to CSV and render under 5s for the default 30-day window.
- Each report page contains a methodology blurb (date range, filters applied, data source), so outputs are self-describing if shared.

11 Hardware Requirements

11.1 Network

Public and University network infrastructure (wired and wireless).

11.2 Client Computers

Modern desktops, laptops, tablets, and smartphones (e.g., Mac, Windows, iOS, and Android devices).

11.3 Server Infrastructure

Cloud-based platform (e.g., AWS, Heroku) to host the web application, APIs, and database.

11.4 Production Support System

Managed cloud services for routine backups, redundant storage, and failover support.

12 Software Requirements

12.1 Client Operating Systems

macOS, Windows, Android, iOS.

12.2 Client Application

A modern, standards-compliant web browser, such as Google Chrome, Apple Safari, Mozilla Firefox, or Microsoft Edge.

12.3 Network System

Network software and protocols for system communication such as TCP/IP, HTTP, HTTPS.

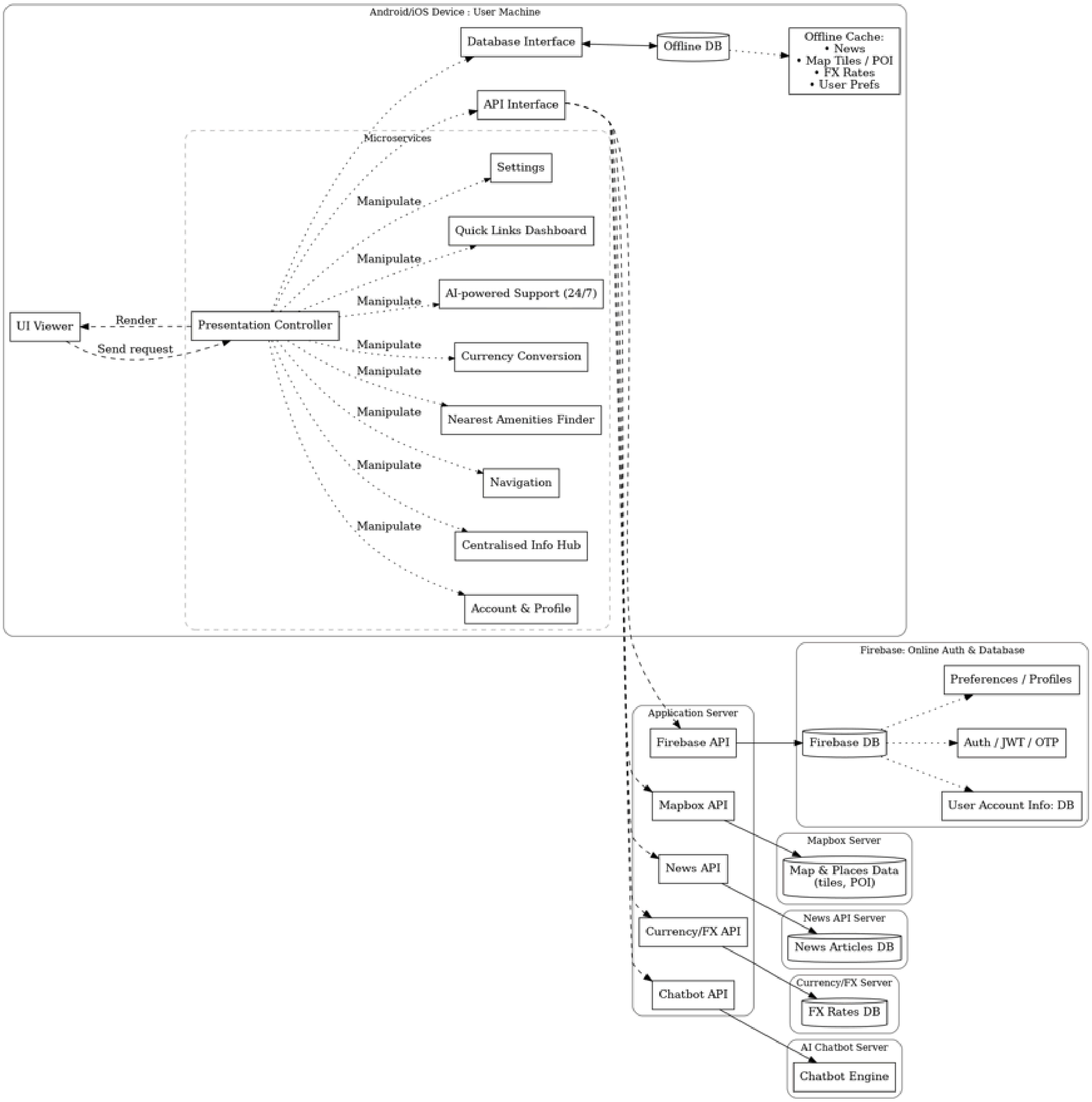
12.4 Server System

Node.js Runtime Environment, Express.js Framework, MongoDB Database.

12.5 Licenses and Third-Party Services

Valid licenses and service agreements are required for Cloud hosting platform (AWS/Heroku), Firebase Authentication service, Mapbox API for navigation, Google Dialogflow API for the chatbot.

13 Deployment Requirements


Android/iOS Device : User Machine
Database Interface
Offline DB
Offline Cache:
• News
• Map Tiles / POI
• FX Rates
• User Prefs
API Interface
Microservices
Settings
Quick Links Dashboard
AI-powered Support (24/7)
Currency Conversion
Nearest Amenities Finder
Navigation
Centralised Info Hub
Account & Profile
Manipulate
Manipulate
Manipulate
Manipulate
Manipulate
Manipulate
Manipulate
Manipulate
UI Viewer
Render
Send request
Presentation Controller
Firebase: Online Auth & Database
Preferences / Profiles
Firebase DB
Auth / JWT / OTP
User Account Info: DB
Application Server
Firebase API
Mapbox API
News API
Currency/FX API
Chatbot API
Mapbox Server
Map & Places Data
(tiles, POI)
News API Server
News Articles DB
Currency/FX Server
FX Rates DB
AI Chatbot Server
Chatbot Engine