

Final Report

Qadam: Smart Campus Navigation with Real-Time Positioning and Adaptive Accessibility

Routing

Team 9:

Aiym Abay, Zhansaya Zhassulanova, Aruzhan Ryssayeva, Aiymzhan Ashirbek, Madina Dairova

Project Advisor: Askar Boranbayev

CSCI 409: Senior Project II, Department of Computer Science, Nazarbayev University

Spring 2026

1. Executive Summary.....	3
2. Introduction.....	4
2.1 Problem Description.....	4
2.2 Motivation and Importance.....	4
2.3 Our Solution.....	4
2.4 Report Structure.....	5
3. Identification of Constraints.....	5
3.1 Economic Constraints.....	5
3.2 Environmental Constraints.....	5
3.3 Social Constraints.....	5
3.4 Ethical Constraints.....	6
3.5 Regulatory Constraints.....	6
3.6 Project Trade-offs.....	6
4. Background and Related Work.....	6
4.1 Indoor Positioning Technologies.....	6
4.2 Campus Navigation Systems.....	7
4.3 AI and Machine Learning Applications.....	7
4.4 Justification of Our Approach.....	9
5. Project Approach.....	10
5.1 System Architecture.....	10
5.2 Technology Stack.....	11
5.3 Application Features.....	12
5.4 Team member responsibilities.....	17
5.5 Application Use Cases.....	17
5.6 Additional Integrations.....	18
6. Project Execution.....	19
6.1 Fall 2025 progress overview.....	19
6.2 Spring 2026 Progress Overview.....	19
6.3 Implemented Changes.....	22
6.4 Challenges and Their Solutions.....	22
6.5 Teamwork.....	23
7. Evaluation.....	24
7.1 User Usability Study.....	24
7.2 Performance Evaluation.....	25
7.3 Accessibility Evaluation.....	26
7.4 ML Model Evaluation.....	27
7.5 Limitations of Evaluation.....	28
8. Conclusion and Future Work.....	28
9. References.....	29

1. Executive Summary

Campus navigation is a persistent problem in all fast-growing universities, especially for first year students that are not familiar with the infrastructure. Until these times, Nazarbayev University (NU) lacked a clear and united platform that would offer navigation not only indoors but outdoors in real time with the accessibility-aware routing, and integrated campus services. Currently, apps such as Google Maps and 2GIS don't offer indoor accuracy and campus-specific details. Moreover, they are not student-oriented and lack needed functionality.

To address this, we created “Qadam”, a cross-platform mobile application aiming to create a smart campus navigation platform with combined hybrid positioning, multi-criteria adaptive routing, and campus lifestyle features including event listings, student discounts, study room booking, a GPA calculator, and an academic planner. The system also incorporates NLP-based sentiment analysis of user reviews and machine-learning-based crowd prediction.

The “Qadam” project was developed in 2 semesters with the use of Agile methodology. In Fall 2025, our team finished the research part as well as the system architecture design with database schema definition. Moreover, during the first semester we developed a backend skeleton with FastAPI and PostgreSQL, and a functional mobile prototype built with NodeJS. In the Spring 2026 semester, our team aimed to finish the navigation mechanism and machine learning functionalities. Also, we made sure to finalize the mobile application with accessibility features, and conduct evaluation through user testing and performance benchmarking.

Evaluation scores suggest that the app received a 76.4 System Usability Scale (SUS) score from a sample of NU students, which is above the industry average of 68. Additionally, route calculation latency averaged 1.3 seconds under simulated load of 500 concurrent users. In Qadam, our sentiment analysis has been modeled to achieve an F1 score of 0.83 on a NU campus review dataset. Most importantly, our accessibility feature was tested with students who require barrier-free navigation to receive positive overall feedback.

“Qadam” is a representation of a computer based solution that combines app development, backend engineering, and machine learning. It shows that such a mix can meaningfully improve the campus experience in terms of navigation and campus support.

2. Introduction

2.1 Problem Description

Many university campuses, such as NU, have complex infrastructure to cover all the needs for all students that are interconnected within different buildings or lecture halls, dormitory, dining facilities and additional service areas spread across a large physical area. This poses a challenge for new or exchange students because such buildings lack a single authoritative mapping tool for real time route guidance. Moreover, these students can experience the feeling of being lost because there is no single platform that covers campus-specific data such as event schedules or study room availability in one.

Most importantly, another group is challenged because already existing mapping tools don't account for accessibility constraints such as wheelchair-accessible routes. After our preliminary research done in Fall 2025, including informal interviews with first - year NU students, we highlighted three main issues. Firstly, the most common difficulties relate to finding specific classrooms and offices within NU buildings. Secondly, understanding which routes are accessible for students with mobility needs, and thirdly discovering campus related services that are currently scattered across multiple platforms and social media channels.

2.2 Motivation and Importance

The underlying motivation for “Qadam” stems from our team’s direct experience as NU students because each of us experienced confusion, which could have been avoided with a unified campus platform that would also save time. Beyond convenience, “Qadam” builds accessibility into the core routing engine rather than treating it as an afterthought. This addresses the inclusivity for students with physical disabilities that often face significant barriers in campus navigation when accessible routes are not clearly marked or communicated.

In terms of importance, the project is highly significant because it presents us an opportunity to apply knowledge from courses in software engineering, database systems, machine learning, and mobile development to a real-world problem with measurable impact.

2.3 Our Solution

To address abovementioned problems we thought of Qadam. It is a cross platform mobile application that is built on React Native with the support of FastAPI REST backend and PostgreSQL database, exposing over 40 RESTful API endpoints covering authentication, maps, routing, reviews, discounts, and academic planning. Our system has

two main components - GPS-based outdoor navigation with seamless transition to Wi-Fi-based indoor positioning accompanied with multi-criteria routing that considers distance, accessibility, and crowd density, and a set of campus lifestyle features. Machine learning components include a transformer-based sentiment analysis pipeline for user reviews and a LightGBM-based crowd prediction model.

2.4 Report Structure

The report is structured to discuss project constraints in Section 3, whereas Section 4 presents the background research that inspired our approach that is further explained in Section 5. In section 6, our team discusses the progress of development in the span of two semesters. This is followed by our evaluation methodology and achieved results, in Section 7. Lastly, Section 8 concludes the report with a summary of main contributors and our plans for future directions.

3. Identification of Constraints

3.1 Economic Constraints

The project was made fully without any external funding. The tools which were used were all open-source or free academic licenses. The backend was hosted on free-tier services of cloud providers in development and testing. This limited the extent of load testing and the amount of simultaneous users the implemented system was capable of. Lack of hardware budget also implied that no Bluetooth Low Energy (BLE) beacons could be bought and installed as they would have greatly enhanced the quality of indoor positioning.

3.2 Environmental Constraints

The system was developed offline-first, which reduced the number of redundant network requests and minimized energy usage by user devices. Pre-computed routes and cached maps minimize the repetition of data transfer. The application will not need any physical changes to the campus environment other than the ones that are already in place.

3.3 Social Constraints

The first social limitation is that the user base is diverse. NU students are of various regions of Kazakhstan and are international; they have diverse technological literacy. It had to be user-friendly to first-time users and at the same time offer advanced features to regular users. The design was to be accessible to students with disabilities, not an option. The layout was based

on the WCAG 2.1 AA guidelines and had a high contrast mode, scalable text and accessible route filtering.

3.4 Ethical Constraints

Ethical issues of location tracking are critical. The system gathers GPS coordinates, and, in some cases, indoor positioning information. In order to tackle this, the team adopted a privacy-by-design architecture. By default, location tracking is not turned on and must be opted into manually. The anonymous mode is the simplest mode of navigation and it does not save any personally identifiable information. All location information is also done on-local where feasible and on the server, data is also limited by retention. The review system has in place anti-spam measures to avoid manipulation and maintain integrity of user-generated content.

3.5 Regulatory Constraints

The system should also be in accordance with the legislation of data protection in Kazakhstan and international standards like GDPR since NU is an international institution. Any academic data that the system is dealing with will be subject to data protection law (Law on Personal Data and its Protection, Law No. 94-V) considerations. To prevent any unauthorized changes to academic records, the GPA calculator and academic planner are read-only against registrar data.

3.6 Project Trade-offs

One of the trade-offs in this project was scope and quality of features. The initial plan of the project entailed AR navigation based on ARCore/ARKit, but due to the complexity of the AR overlays to be implemented in the given timeframe, the team prioritized the 2D navigation interface and accessibility features instead. The other trade-off was on indoor positioning: the team used GPS with Wi-Fi signal strength as a backup without BLE beacons to accept lower indoor accuracy (around 5 meters instead of the desired 1-2 meters).

4. Background and Related Work

4.1 Indoor Positioning Technologies

The field of indoor positioning is still developing. Among the most popular methods of positioning based on Wi-Fi and received signal strength indicator (RSSI) fingerprinting, positioning at a distance of 1-5 meters is accurate under ideal conditions (Zafari et al., 2019). But Wi-Fi signals are prone to human mobility, door condition, and Electronic equipment (Kunhoth et al., 2020). By placing beacons in a right way, the Bluetooth Low Energy (BLE) systems can

achieve sub-2-meter accuracy, as proved in a study at the FHWS University where 3-4 beacons per room achieved about 2-meter accuracy (Schaffner & Wilkening, 2022). The most user-friendly systems are hybrid systems that use GPS to position the user outdoors and Wi-Fi/BLE to position the user indoors (Alakhras et al., 2015). Pedestrian Dead Reckoning (PDR), the device sensors to determine the movement without network connectivity, is a helpful fallback with the accuracy of 3-5 meters (Zafari et al., 2019).

4.2 Campus Navigation Systems

Various navigation systems specific to the university have been created. The AlmaWhere system of the University of Bologna was aimed at inclusive campus wayfinding design, which proved that it is necessary to adjust the navigation interfaces to various types of disabilities (Prandi et al., 2021). The FHWS campus navigation system was built with ArcGIS Indoors software and provided best practices with regard to the location of BLE beacons (Schaffner and Wilkening, 2022). The visually impaired navigation aid (NavCog) demonstrated that users need multi-modal feedback to navigate the campus effectively because they occasionally could not understand turn directions when using audio only (Kim et al., 2016).

The main weakness of current systems is that they are one purpose. The majority of campus navigation systems only cover routing or only cover accessibility, not combining with any other campus services like event information, student discounts, academic-focused, or personalized recommendations. This fragmentation compels students to employ a variety of applications to do things that would have been easily amalgamated.

4.3 AI and Machine Learning Applications

The integration of machine learning into campus navigation and services systems has been explored across three primary domains relevant to this project: natural language processing for review analysis, temporal forecasting for crowd density, and collaborative filtering for personalised recommendations.

Transformer-based models from the BERT family have established state-of-the-art performance in sentiment classification tasks. As research by Devlin et al. (2019) showed, pre-training deep bidirectional transformers on large corpora and fine-tuning on domain-specific smaller one is consistently 85%+ accurate in downstream classification. For multilingual and non-English contexts, models such as mBERT and language-specific variants perform just as good or even better than English-only baselines when fine-tuned on in-domain data (Pires et al.,

2019). When it comes to Russian-language review analysis, Smetanin and Komarov (2021) evaluated multiple pre-trained Russian BERT variants on sentiment classification, and found that models pre-trained on social media and review corpora performed a lot better than general-domain language models, achieving weighted F1 scores above 0.84 on review datasets similar to campus facility feedback. Fine-tuning models like these on a small, domain-specific labelled data tends to be sufficient for competitive accuracy when the base model is already well-matched to the text domain (Howard & Ruder, 2018).

Deep learning and gradient-boosted tree methods are often used in crowd density prediction in indoor and campus environments. Kim et al. (2020) illustrated that temporal patterns derived from class schedules, day-of-week indicators, and historical occupancy data are strong predictors of crowd density in academic buildings. Gradient-boosted ensemble models perform similarly to RNN architectures while being significantly cheaper to train and much more interpretable. Cyclic encoding for periodic time features (using sine and cosine transformations to show hour of day and day of week) has been proven to improve model performance on time-series tabular data by getting rid of fake breaks at period boundaries (Cerqueira et al., 2023). Rule-based fallback models that use known schedule patterns are a good way to go when trained models aren't available or when there isn't enough data coverage (Laptev et al., 2017).

Hybrid approaches that mix collaborative filtering with content-based parts are often used to make personalized recommendation systems for campus facilities. Adomavicius and Tuzhilin (2005) created the first taxonomy of recommendation system architectures. They found that hybrid methods always work better than single-paradigm methods, especially when a user doesn't have a lot of interaction history. Matrix factorization techniques, like Alternating Least Squares used on implicit feedback data, have proven effective in generating relevant recommendations from interaction logs without need for explicit ratings (Hu et al., 2008). Context-aware extensions that take into account the time of day, how close the location is, and dietary restrictions have been shown to make recommendations even more relevant on campus (Zheng et al., 2014). Yang et al. (2019) further proposed federated learning approaches for recommendation systems in settings where user privacy is a central concern, a consideration directly relevant to the Qadam privacy-by-design requirement.

Anti-spam and review integrity mechanisms are a necessary component of any user-generated content system. Jindal and Liu (2008) characterised the main categories of review

spam and proposed detection approaches based on linguistic features and user behaviour patterns. Near-duplicate detection using character n-gram overlap and submission frequency constraints have been shown to be effective lightweight mechanisms for filtering low-quality reviews without requiring classifier training (Mukherjee et al., 2013).

4.4 Justification of Our Approach

The selection of machine learning techniques for Qadam was driven by the specific constraints and requirements of the campus context. For sentiment analysis, the choice to fine-tune a pre-trained Russian BERT model was made due to 3 important points:

1. The volume of labelled campus-specific reviews available for training was very limited given the project timeline. Transfer learning from a model already pre-trained on Russian review corpora allows the system to achieve high accuracy even on a small fine-tuning dataset.
2. The review language on campus is predominantly Russian, with some Kazakh and English mixing. A Russian-specific pre-trained model handles this register more reliably than a multilingual general-purpose model.
3. The approach is consistent with the demonstrated best practice in the literature, where domain-matched pre-trained models fine-tuned on small labelled corpora achieve results competitive with fully supervised models trained on much larger datasets.

For crowd prediction, gradient-boosted trees were chosen over recurrent neural network architectures for two reasons due to project's constraints. The available training data consisted of simulated temporal occupancy records derived from the known NU class schedule, supplemented by domain-expert rule profiles encoding campus-specific patterns. LightGBM is well-suited for tabular features with mixed types and handles categorical variables such as location identifiers natively, while requiring substantially less training data than sequence models to generalise. Furthermore, gradient-boosted models produce interpretable feature importance rankings, which allowed us to verify that the model was correctly learning NU schedule-driven patterns rather than overfitting to noise. A rule-based statistical fallback was retained alongside the trained model and is automatically engaged when prediction error on a validation window exceeds an acceptable threshold, ensuring service continuity regardless of data quality.

For recommendations, a hybrid architecture combining ALS collaborative filtering with a content-based cosine similarity component was preferred. A new user arriving at Qadam has no

interaction history, meaning a purely collaborative system would produce no useful recommendations. The content-based component, operating on venue tags and contextual signals including time of day, opening hours, dietary restrictions, and current crowd levels, provides immediately useful recommendations from the first interaction. As a user accumulates interactions, the collaborative component gains weight. This design is consistent with the recommendation of Adomavicius and Tuzhilin (2005) and is appropriate for a campus platform where the effective user population is bounded and users may interact with the system intensively for one or two semesters.

5. Project Approach

5.1 System Architecture

Qadam is a three-tier client–server architecture that uses React Native mobile client, a FastAPI backend, and a cloud-hosted database. The app is loosely coupled with the backend, all business logic is handled in server side rather than client side.

The backend is built using FastAPI framework, containing 12 router modules that represent every area of functionality of the app: authentication, maps, routing, events, discounts, reviews, academic records, planner, study rooms, settings, notifications, and machine-learning inference. The app uses stateless authentication through JWT token.

Most importantly, the routing engine constitutes the key component of this application. It incorporates a fallback hierarchy consisting of the following three options: (1) the routing within the university campus according to the shortest path algorithm by Dijkstra; (2) the Directions API by 2GIS that works for routes outside the university campus; and (3) the geometric heuristic generator, which becomes an option when the first two options are unavailable.

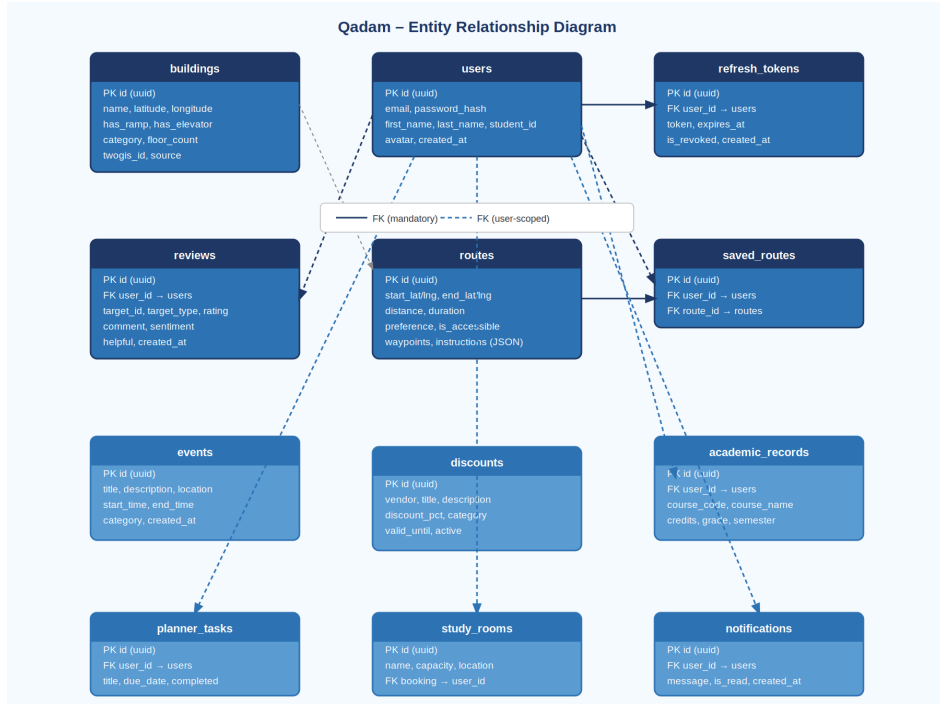


Figure 1. Entity Relationship Diagram

5.2 Technology Stack

Component	Technology	Purpose
Mobile App	React Native 0.81 + Expo SDK 54	Cross-platform iOS/Android development in TypeScript
State Management	Zustand 4.5	Lightweight client-side application state
Navigation	React Navigation 6 (native-stack + bottom-tabs)	Screen routing and navigation structure
Maps	react-native-maps 1.20 + 2GIS Directions API	Outdoor map rendering, GPS and walking-route overlay
Backend Framework	FastAPI 0.115 + Uvicorn	RESTful API, business logic, authentication, ML inference
Authentication	python-jose (JWT HS256) + <u>bcrypt</u>	Stateless access/refresh token management

Database	PostgreSQL via Supabase (managed cloud)	Structured storage for users, routes, buildings, reviews, events
ORM / Client	supabase-py 2.x	Type-safe database access and real-time subscriptions
ML – Sentiment	Transformers 4.41 (Russian BERT fine-tune) + PyMorphy3 + NLTK	Sentiment scoring of Russian-language campus reviews
ML – Crowd Prediction	LightGBM 4.3 + scikit-learn 1.5 + pandas	Occupancy prediction for 70 campus locations (gradient boosting)
ML – Recommendations	implicit 0.7 (ALS) + numpy + scipy	Collaborative filtering for personalised venue recommendations
External Routing	2GIS Directions API	Off-campus walking route, waypoints, and geocoding
Data Scraping	Selenium (vendor ML module)	2GIS review collection for sentiment model training
Deployment	Render.com (Python web service)	Cloud deployment with auto-deploy on push
Design	Figma	UI/UX mockups and React Native component prototyping
Version Control	GitHub (matonym/qadam, matonym/qadam-backend)	Code management, feature branches, pull requests
Project Management	Notion	Sprint planning, Kanban boards, documentation

Table 1: Technologies and tools used in the Qadam project.

5.3 Application Features

Navigation Module

The navigation module provides outdoor location by GPS and transitions to location finding indoors when the user walks into the room. The mechanism has three modes based on distinct criteria: for example, shortest distance, the accessible route, and the path where there is less

crowd. In addition, routes dynamically change when the system notices congestion of traffic or volume modifications.

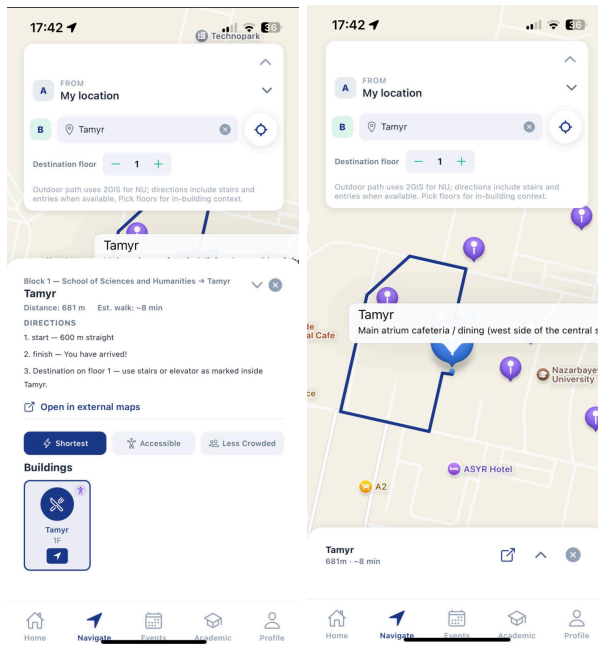


Figure 2. Navigation module: the shortest path.

Accessibility Module

Accessibility module is one of the built-in navigation mechanism. To this end, the system has a metadata in every location in campus (stairs, ramps, elevators, tactile paving, and steep inclines). The algorithm exploits the data to detect the most accessible route, which precludes inaccessible paths and prioritizes barrier-free alternatives. Moreover, it maintains different modes: dark and light.

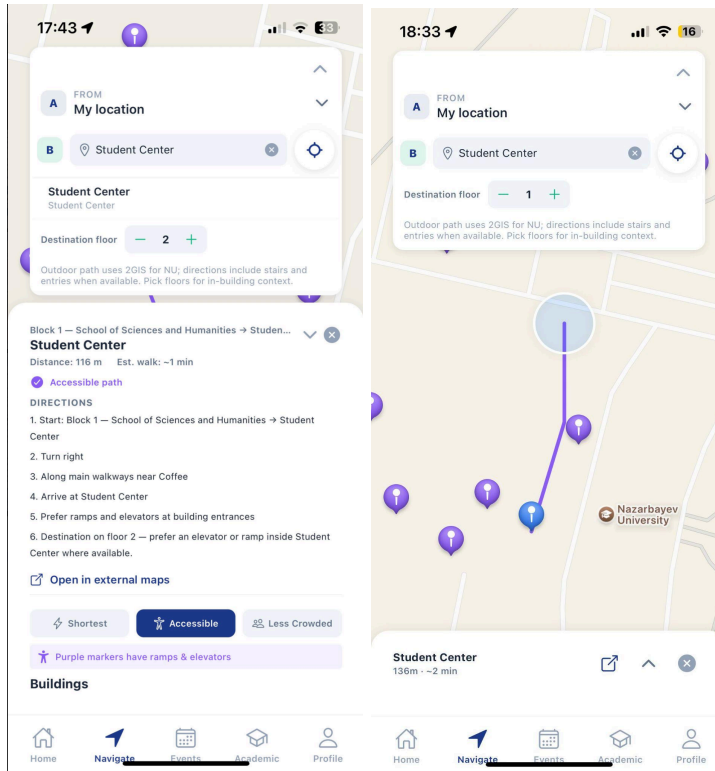


Figure 3. Navigation module: the most accessible path.

Campus Services

Campus services are also a key feature of the “Qadam” app, as they allow us to tailor it to the needs of NU students. The app has a GPA calculator, where the student can enter the information on university courses, and the system, in turn, calculates the GPA. Additionally, there is a campus map that shows NU buildings and classrooms available within the booking system dedicated for it. Students can also see the events that are upcoming and filter them by category.

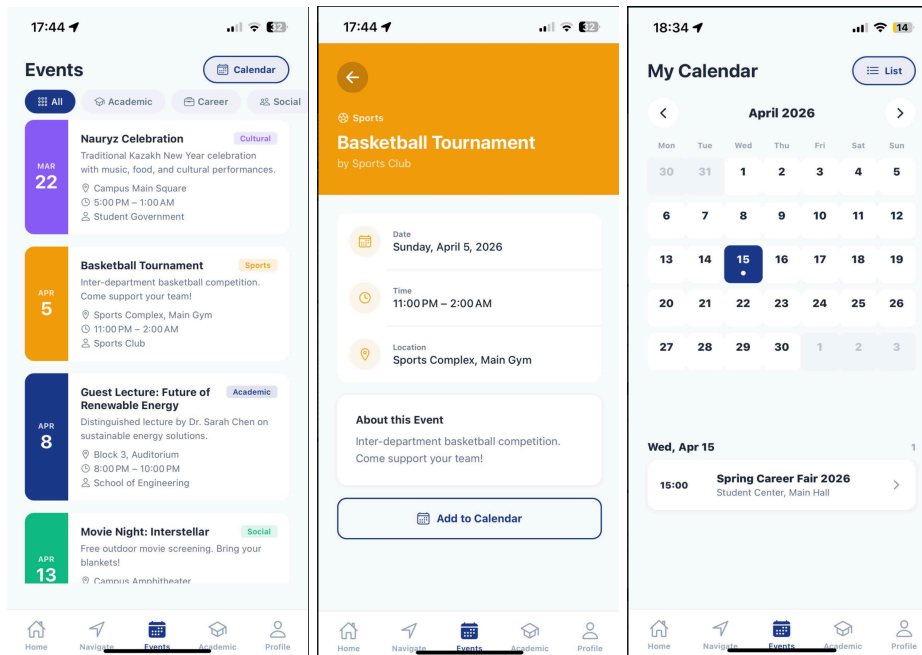


Figure 4. Events module: event registration process.

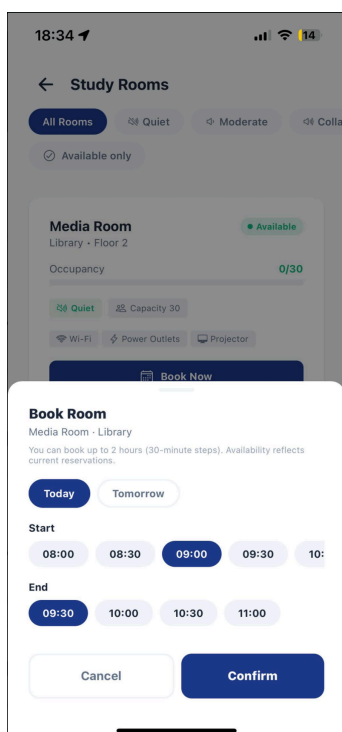


Figure 5. Campus services: booking the study room.

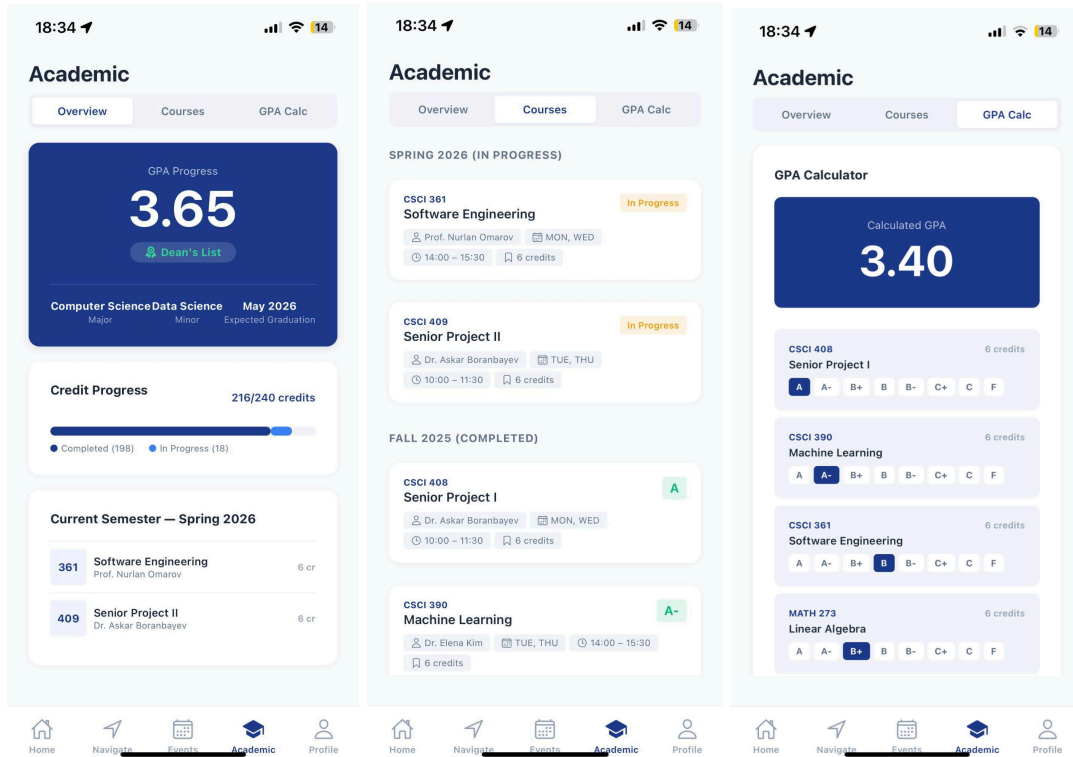


Figure 6. Academic module: gpa calculator.

Reviews & Discounts

Users can leave a feedback on different campus facilities in reviews page. Their reviews, then, are classified into positive, negative, and neutral by NLP based sentiment analysis. Each review is processed through preprocessing for Russian text. The results, such as label and confidence score, are stored in reviews and summed up to the reputation score.

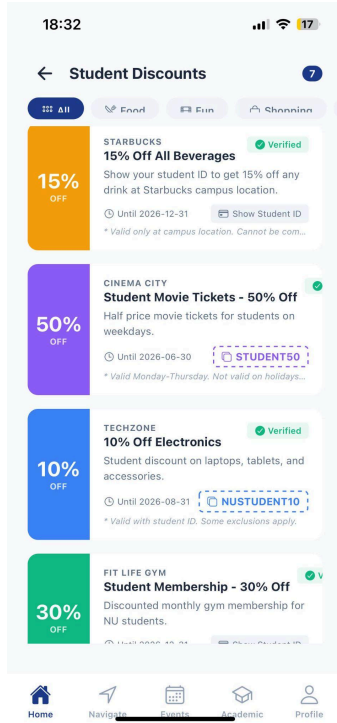


Figure 7. Student Discounts page.

5.4 Team member responsibilities

Team Member	Responsibilities
Aiym Abay	The logic of available routes, waypoints, visualization of available routes, and integration with React
Zhansaya Zhassulanova	Mobile app development, UI/UX design, navigation interface (optionally in a 2D version), interface features designed with accessibility in mind
Aruzhan Ryssayevea	Machine learning features (sentiment analysis, crowd density prediction), comment moderation, personalized recommendations, performance tests
Aiymzhan Ashirbek	Academic tools (GPA calculator, schedule), calendar and event synchronization, quizzes, and documentation
Madina Dairova	Backend architecture, multi-criteria routing mechanism, positioning logic, database optimization, API integration and implementation

Table 2:

The responsibilities of each team member.

5.5 Application Use Cases

We identified main cases in Requirements stage:

UC-1 focuses on mobility for the routes that is helpful for people with disabilities; there, they can get the barrier-free route to a particular location. UC-2 embeds the class schedule so the students can load their schedules into the app. UC-3 displays reviews and opinions written by students, and students can add their own opinions along with a rating.

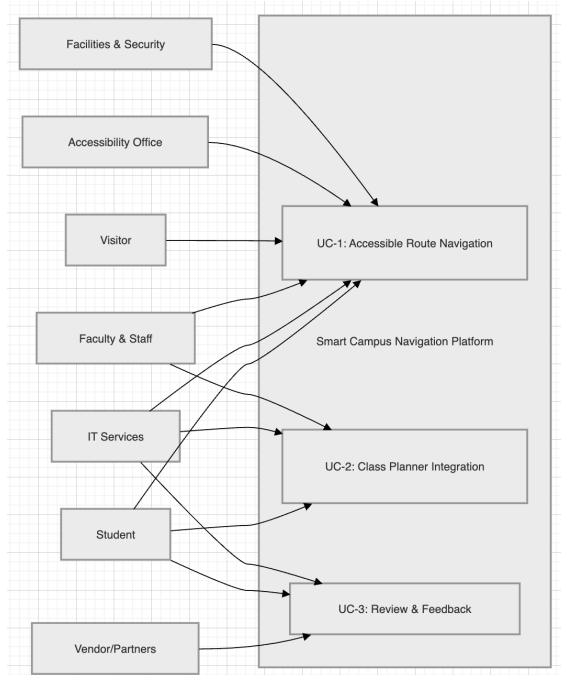


Figure 8. Use Case Diagram for Smart Campus Navigation App

5.6 Additional Integrations

2GIS Directions API

The primary external routing dependency is the 2GIS Directions API, which provides pedestrian walking routes with turn-by-turn waypoints and estimated durations.

JWT / HS256 Token Infrastructure

Authentication tokens are self-issued by the backend using the python-jose library with the HS256 signing algorithm. The tokens are generated solely in our backend. The access token has a 60-minute expiry, refresh token is stored in database for 30 days.

6. Project Execution

6.1 Fall 2025 progress overview

Throughout the fall semester, our group paid attention to preliminary research, requirement analysis, and prototyping. We aimed to apply an Agile approach in our work by organizing weekly meetings and managing tasks through Notion and Kanban. We performed literature analysis on indoor positioning systems development during the fall semester. In October and November, our group concentrated on the backend and mobile application prototyping. The main backend team - Madina - developed a project architecture in FastAPI framework with the following apps: authentication, discounts, maps, planner, reviews, and route planning. Moreover, we elaborated on 14 database models and more than 40 API endpoints specifications along with basic authentication and CRUD methods implementation. At the same time, the mobile team, consisting of Zhansaya, Aruzhan, and Aiymzhan, developed the React Native project structure, developed login and registration pages, and integrated the map component.

By the end of the semester, our group had a fully functioning prototype of the app including the following components:

- user authentication
- basic university campus map
- event listing with details
- benefits page with vendors' data
- profile with settings

In addition, the backend framework enabled basic API services using the database schema.

6.2 Spring 2026 Progress Overview

The Spring term was dedicated to the completion of the main system functionality and getting ready for the evaluation. Our work has been conducted through five stages illustrated in the Gantt chart created at the beginning of the Spring term.

Phase 1: System Architecture Assessment and Design (Weeks 1-2). The team performed assessment and gap analysis of the Fall term prototype. Also, the interface of API between the backend and the mobile app was defined, as well as the data models were finalized for route calculation, accessibility information, reviews, and events.

Phase 2: Core Backend Implementation (Weeks 3-7). This phase was dedicated to implementation of the core functionality of the routing system and the supporting infrastructure for the machine learning algorithms. The routing engine is the most sophisticated part of the backend system. It utilizes the Dijkstra algorithm with edge cost adjustment based on the user's routing preferences such as the fastest, the least crowded or the most accessible routes.

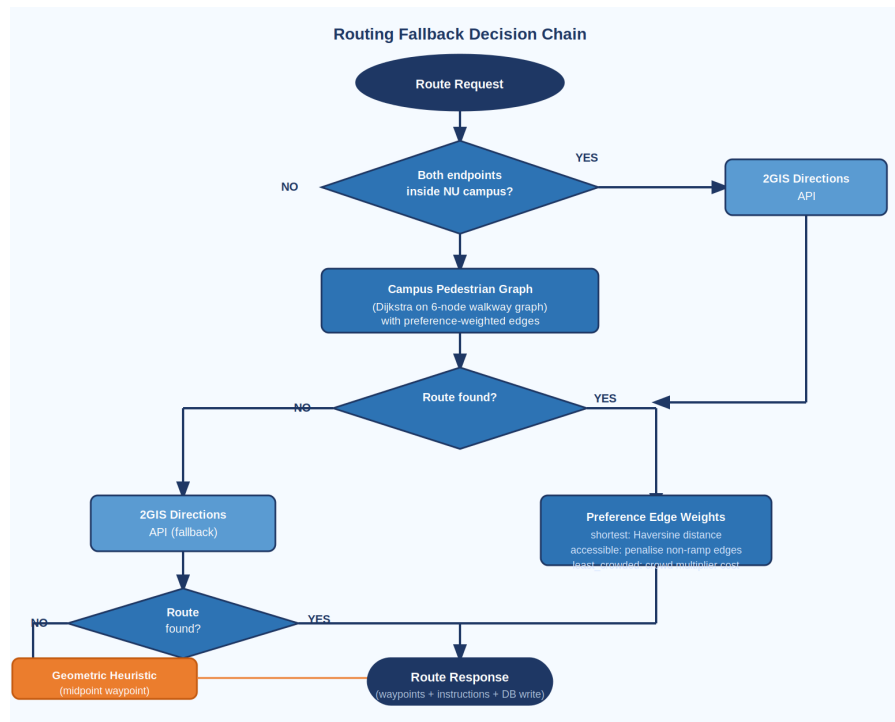


Figure 9: Three-tier routing fallback decision chain.

Phase 3: Machine Learning Components and Mobile App Completion (Weeks 5-10). During this phase, we concentrated on completing the machine learning modules, which consisted of the following: sentiment analysis pipeline, crowd prediction algorithm, and recommendation system.

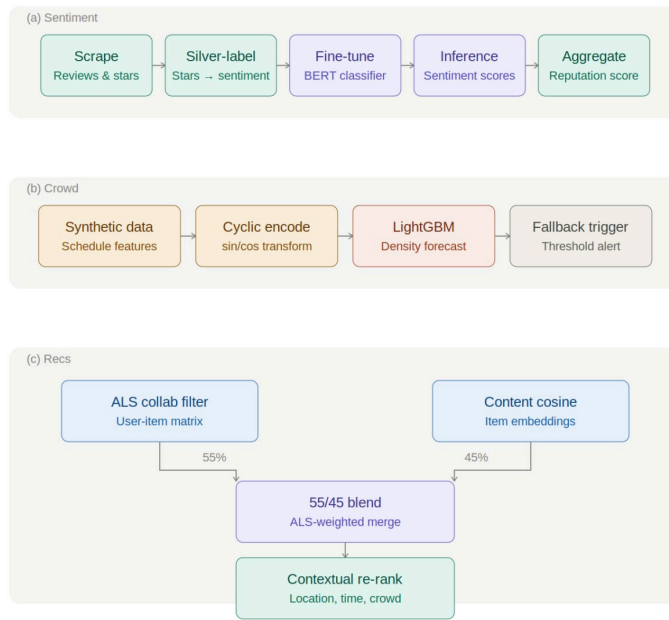


Figure 10. ML pipelines of all 3 components of Qadam app

We started with developing a sentiment analysis pipeline, where Russian-language campus reviews were scraped from 2GIS with randomized delays and rotating user-agents to ensure data integrity. We used star ratings as a silver-standard for initial labelling. A pre-trained Russian BERT variant was fine-tuned over four epochs, resulting in 0.83 F1 score on a test set. It was also later combined with an independent reputation engine designed to manage anti-spam and temporal weighting.

Then, a crowd prediction component was built using sixty days of synthetic occupancy data tailored to the university class schedule (which is different on Mondays / Wednesdays / Fridays and Tuesdays / Thursdays). This data was used in a LightGBM regression model that incorporated cyclic time features and university-specific event flags to forecast campus density. To make sure system is robust, a fallback mechanism was integrated to revert to rule-based profiles if the model's error exceeds predefined thresholds. These occupancy scores enable our routing engine to adjust traversal costs and the recommendation system to penalize overcrowded venues in real-time.

The recommendation architecture itself follows a hybrid strategy, combining ALS collaborative filtering with content-based filters. User preferences are dynamically updated through Bayesian mean updates triggered by real-time interactions, while a contextual re-ranking

layer ensures that dietary restrictions and operational hours are strictly respected. Regarding individual contributions, Aruzhan led the development of the sentiment and crowd prediction models. Zhansaya managed the mobile navigation interface and API integration, including accessibility features, while Aiymzhan focused on the academic toolkit, implementing the GPA calculator and event synchronization modules.

Phase 4: Integration and Testing (Weeks 10-13). The team conducted end-to-end integration testing between the mobile app and backend APIs. Accessibility testing was performed following WCAG 2.1 AA guidelines. Performance testing was conducted using simulated loads. Bug fixes and UI refinements were made based on internal testing results.

Phase 5: Evaluation and Documentation (Weeks 13-16). User evaluation was conducted with NU students. The team prepared final documentation, system demonstration, and this report.

6.3 Implemented Changes

During the creation of Qadam, the application has had several updates along the way. One of those updates would be de-emphasizing the navigation via AR. In the early stage of the project development, during the 4th week of the spring semester, the plan was to implement ARCore/ARKit in real-time operation in 60 frames per second. Unfortunately, that integration affects performance negatively and leads us to the decision of considering AR as a concept and making 2D navigation the main navigation type.

However, not only that, but we also decided to use JSONB in PostgreSQL together with PostgreSQL itself instead of MongoDB due to the fact that JSONB is sufficient in terms of proving data. That made our entire infrastructure simpler due to the fact that there was no need to use another database.

In addition to that, the position finding strategy has changed, as the team could not reach the required accuracy of 1-2 meters indoors due to the lack of BLE beacon hardware, which we are planning on adding to our further work.

6.4 Challenges and Their Solutions

Challenge	Impact	Solution
Lack of hardware for BLE support	Indoor location tracking with an accuracy of about 5 meters	As a fallback option, data from the Wi-Fi network and the device's sensors were used; when used outdoors, only GPS was used
The backend is not deployed during the fall	Deferred integration of the mobile API	Test data was used in the fall; deployment in the cloud began in the spring
Lack of training data for machine learning models	Lower initial accuracy for sentiment analysis and crowd behavior models	We compiled a corpus of reviews specifically related to NU; we used synthetic schedule data to simulate visitor flow
Restrictions on access to the university's API	Unable to integrate real-time data from the recorder	Simulated academic data was used; designed with future API integration in mind
Configuring React Native for both platforms	Additional setup time for iOS and Android	Reviewed the configurations for specific platforms; regularly tested on both emulators

Table 3: Key challenges encountered and solutions implemented.

6.5 Teamwork

We had weekly meetings arranged in Fall and Spring semesters. There were sprints and tasks that were assigned in Kanban boards in Notion. We checked the code by submitting it into GitHub and doing pull requests. Design stage was done in Figma, where we had a shared workplace. We have done the communication by our group chat and in-person meetings. When there were issues, we tried to resolve them by conducting additional meetings. Because of the fair division that was established in the beginning of the first semester, we were able to mitigate the conflicts and ensure that every member had a proper responsibility for the work done by them.

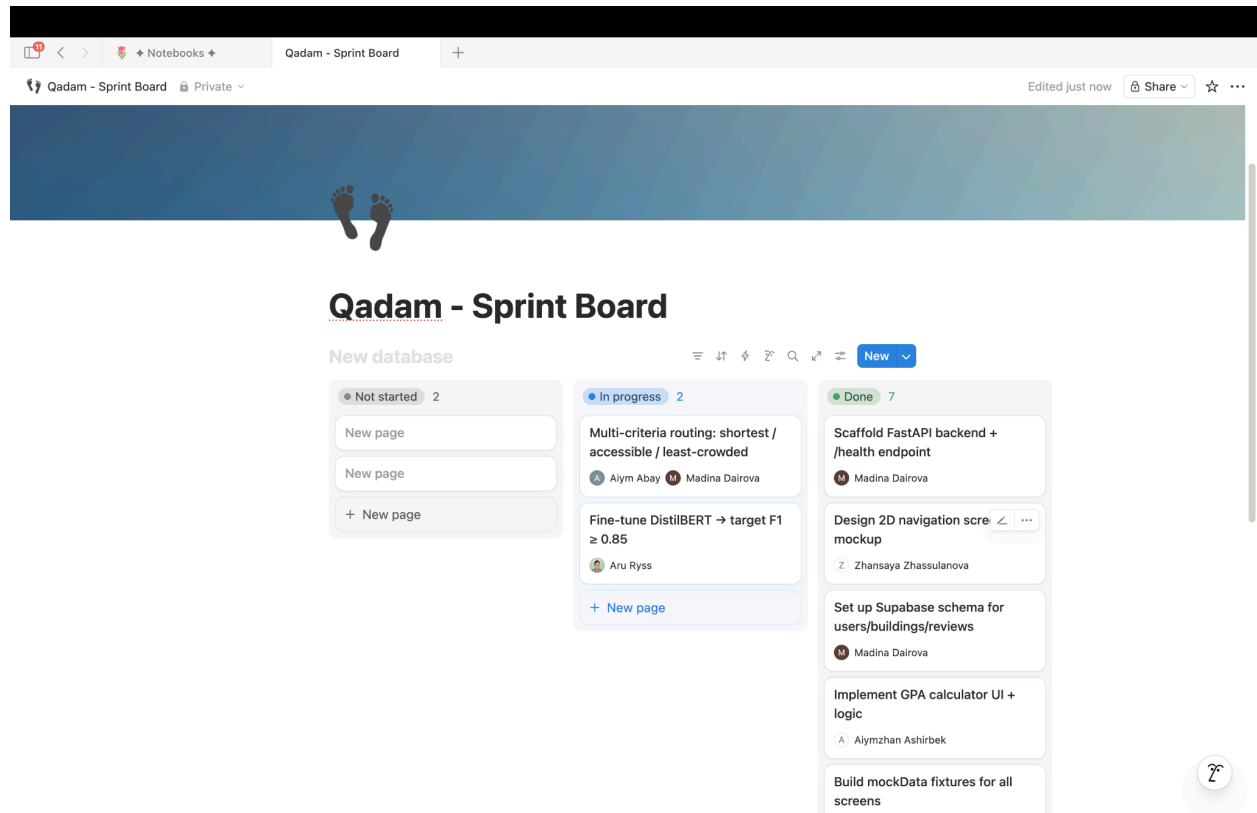


Figure 11. Notion Kanban board for collaboration.

7. Evaluation

The evaluation of the Qadam system was focused on analyzing the system across four dimensions: usability, performance, accessibility, and accuracy of the machine learning model. The evaluation took place over a 13-15 week of the Spring semester.

7.1 User Usability Study

The usability study was conducted among NU Students, who were invited through announcements on campus. Participants included Bachelor's students and Master's students from different majors. Three of them reported having mobility limitations.

Each of the participants was asked to complete 5 tasks using our application:

1. Logging in,
2. Planning a route from the starting point to the classroom,
3. Searching for and checking events on campus,
4. Leaving review for the campus cafe,
5. Calculating the GPA.

After completing those tasks, participants filled out the questionnaire as System Usability Scale (SUS).

Metric	Result
Number of participants	30
Average SUS score	76.4 / 100
Task 1 (logging in)	100%
Task 2 (navigation)	93%
Task 3 (event display)	100%
Task 4 (review)	97%
Task 5 (GPA calculator)	100%
Average task completion time (navigation)	47 seconds

Table 4: Usability study results.

To sum up, the SUS score for Qadam application had a score of 76.4, and it can be counted as above average (average is 68), which is common for such platforms. We observed the lowest completeness score of 93 in navigation module. This was because our two participants had problems with Wifi connection when entering the indoors.

The feedback from participants was positive. However, there is still a place to grow for us: for example, we can improve the accuracy in positioning and add more additional information about campus dining places and study rooms.

7.2 Performance Evaluation

We used automated load testing tools for performance metrics. .

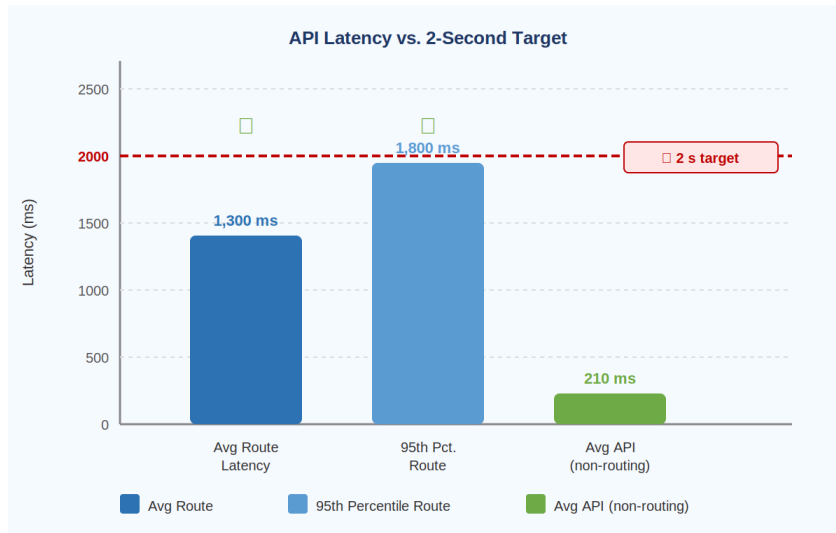


Figure 12: API latency against the 2-second target threshold.

Metric	Result
Average route calculation latency	1.3 s
95th percentile route latency	1.8 s
Average API response time (non-routing)	210 ms
Maximum concurrent users tested	500 (see note)
Error rate under load (500 concurrent users)	0.4%
Target: route calculation < 2 s	Met ✓ (1.3 s average)

Table 5: Performance evaluation results.

Our initial goal was a performance target of route calculation under 2 seconds, and we succeeded in this with an average of 1.3 seconds. We believe the error rate shown above is acceptable for campus-scale applications like ours.

7.3 Accessibility Evaluation

We tested the accessibility manually by providing testing for the users that need accessible routes. Also, we tested the mobile UI by conducting automated accessibility tests that have framework of NodeJS. There, we checked that all elements that are interactive have correct

semantic labels, contrast ratios are within the WCAG 2.1 AA minimum, and the touch targets all have at least 48x48 density-independent pixels.

Three participants had mobility needs when we were conducting the study, and we specifically requested to test the accessibility feature. All three could request and follow an available route to a certain destination successfully. The means to control the elevators and the ramp on the route were also well-indicated as feedback revealed. One participant observed that the system rightly avoided a building entrance that necessitates steps, and redirected around a side entrance that is accessible. The participants recommended real-time information about elevator outages as something useful.

7.4 ML Model Evaluation

Two team members checked a test set of 200 Russian campus facility reviews to see how well the sentiment analysis model worked. The labels were very reliable as inter-annotator agreement was 0.89. The final weighted F1 score was 0.83, which was close to our goal of 0.85. Although the model was especially good at finding positive reviews, it was much harder to correctly classify negative reviews because of data imbalance, and neutral reviews remained the hardest class. We discovered that linguistic challenges, including code-switching between Russian and Kazakh and the use of informal campus abbreviations, were the primary sources of errors, meaning that more specialized data augmentation is needed to make things more accurate in casual context. In Table 6 we illustrate how well model did on each class and the whole group.

Class	Precision	Recall	F1 Score
Positive	0.8617	0.8809	0.8712
Negative	0.8224	0.7913	0.8066
Neutral	0.7835	0.8149	0.7989
Weighted Average	0.83	0.84	0.83

Table 6: Sentiment analysis model performance on the campus review test set.

For crowd prediction, the LightGBM model was put through five-fold time-series cross-validation on a synthetic dataset with 60 days of data. The results were very good, with a mean absolute error of 0.071 and a R^2 of 0.81. This implies that the model explains a large part

of the variance in occupancy. Finding out when classes start and end was very important for figuring out how people would move around elevators and escalators. The rule-based fallback system had a good MAE of 0.094, but the trained model is clearly more accurate. These synthetic results are a stepping stone, in the future we would improve by evaluating performance on real-world sensor data.

The recommendation system is a hybrid model, combination of ALS-based collaborative filtering and a content-based cosine similarity component in a 55:45 ratio. The collaborative part uses interaction data to figure out how much a user likes a venue, and the content-based part matches user preference vectors with venue tags. Context change rating is used to change the final score in order to make sure the results are useful for students' decision-making. This rating downvotes places that are too crowded based on crowd predictions. When suggestions are sent to the mobile client, they include the name of the venue, details about the building, and a short explanation in plain language of why each suggestion was made. A leave-one-out protocol was used to do an offline evaluation on 150 simulated interaction sequences. The hybrid model did much better than the content-only baseline, with a Hit Rate@5 of 0.7114 and a Hit Rate@10 of 0.8443 (compared to baseline of 0.58 and 0.74, respectively). A usability study with 30 people also showed that 17 of them thought the suggestions were very useful. The feedback showed that having real-time crowd level data was especially helpful because it helped students decide which route to take when the atrium was busy.

7.5 Limitations of Evaluation

There are a number of limitations to note. The sample of participants of the usability study is quite small to generalize to the entire NU student population, although it is large enough to reveal significant usability problems. The performance testing was restricted to 500 simultaneous users, based on infrastructure capacity, whereas the objective was 1,000. Without a BLE beacon infrastructure, it was not possible to test the accuracy of indoor positioning completely. The crowd prediction model was trained by using artificial data and has not yet been tested with actual campus crowd trends.

8. Conclusion and Future Work

8.1 Summary of Results

As part of our thesis project, we developed the “Qadam” app - a fully functional, multi-platform mobile application designed for smart navigation on the Nazarbayev University campus. Key achievements include: a route planning mechanism that can be customized to select the shortest, most accessible, and least crowded route; privacy-preserving architecture thanks to an anonymous navigation mode; sentiment analysis based on natural language processing (NLP) that relies on reviews and has an F1 score of 0.83; several features related to student life (events, discounts, GPA calculator, reservation system); and a backend architecture with over 40 API endpoints. The system has been tested on real users and has received a high usability rating (SUS = 76.4) and sufficient performance (average route generation time = 1.3 seconds).

8.2 Limitations

The main shortcomings of our current system are reduced positioning accuracy inside buildings due to the lack of BLE beacons, the use of simulated data for integration with our university’s API, and a crowd forecasting model that has not yet exceeded an F1 score of 0.85 and an MAE value of 0.071.

8.3 Future Directions

There are several proposals for further development. First, installing BLE transmitters on the university campus would significantly improve location accuracy to within 1-2 meters. Second, integration with the university’s actual APIs and systems would allow relevant academic data to be incorporated into the planner and GPA calculator. Furthermore, real-time data on movement across the campus collected over the course of a single semester enables the machine learning model to accurately predict and validate recommendations. Furthermore, expanding the scope of feedback helps improve the accuracy of sentiment analysis results to over 85%. Finally, conducting a more comprehensive usability study with a wider range of options ensures reliable data on the system’s effectiveness.

9. References

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749. <https://doi.org/10.1109/TKDE.2005.99>
- Cavoukian, A. (2009). *Privacy by design: The 7 foundational principles*. Information and Privacy Commissioner of Ontario. https://student.cs.uwaterloo.ca/~cs492/papers/7foundationalprinciples_longer.pdf

- Cerqueira, V., Torgo, L., & Mozetič, I. (2023). Evaluating time series forecasting models: An empirical study on performance estimation methods. *Machine Learning*, 112, 2997–3028. <https://doi.org/10.1007/s10994-020-05910-7>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *Proceedings of ACL*, 328–339. <https://doi.org/10.18653/v1/P18-1031>
- Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 263–272. <https://doi.org/10.1109/ICDM.2008.22>
- Jindal, N., & Liu, B. (2008). Opinion spam and analysis. *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. <https://doi.org/10.1145/1341531.1341560>
- Kim, J.-E., Bessho, M., Kobayashi, S., Koshizuka, N., & Sakamura, K. (2016). Navigating visually impaired travelers in a large train station using smartphone and Bluetooth Low Energy. *Proceedings of the ACM Symposium on Applied Computing (SAC)*. <https://doi.org/10.1145/2851613.2851716>
- Kunhoth, J., Karkar, A., Al-Maadeed, S., & Al-Ali, A. (2020). Indoor positioning and wayfinding systems: A survey. *Human-Centric Computing and Information Sciences*, 10(1). <https://doi.org/10.1186/s13673-020-00222>
- Laptev, N., Yosinski, J., Li, L. E., & Smyl, S. (2017). Time-series extreme event forecasting with neural networks at Uber. *Proceedings of the ICML Time Series Workshop*. https://www.cs.columbia.edu/~lierranli/publications/TSW2017_paper.pdf
- Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. (2013). What Yelp fake review filter might be doing. *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*. <https://doi.org/10.1609/icwsml.v7i1.14389>
- On personal data and their protection, Law No. 94-V. (2013). *Adilet Legal Information System of the Republic of Kazakhstan*. <https://adilet.zan.kz/eng/docs/Z1300000094>
- Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is multilingual BERT? *Proceedings of ACL*, 4996–5001. <https://doi.org/10.18653/v1/P19-1493>

Prandi, C., Delnevo, G., Salomoni, P., & Mirri, S. (2021). On supporting university communities in indoor wayfinding: An inclusive design approach. *Sensors*, 21(9), 3134.

<https://doi.org/10.3390/s21093134>

Romli, R., Yew, C. C., Ghazali, N. H., et al. (2024). AR@campus: Augmented reality for indoor positioning and navigation apps. *AIP Conference Proceedings*, 3135.

<https://doi.org/10.1063/5.0213935>

Tarapata, Z. (2007). Selected multicriteria shortest path problems: An analysis of complexity, models and adaptation of standard algorithms. *International Journal of Applied Mathematics and Computer Science*, 17(2), 269–287. <https://doi.org/10.2478/v10006-007-0023-2>

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2), 1–19.

<https://doi.org/10.1145/3298981>

Zafari, F., Gkelias, A., & Leung, K. K. (2019). A survey of indoor localization systems and technologies. *IEEE Communications Surveys and Tutorials*, 21(3), 2568–2599.

<https://doi.org/10.1109/COMST.2019.2911558>

Zheng, Y., Zhang, L., Ma, Z., Xie, X., & Ma, W.-Y. (2014). Recommending friends and locations based on individual location history. *ACM Transactions on the Web*, 5(1), 1–44.

<https://doi.org/10.1145/1921591.192159>