

# PRD: CampusSpot

Anagha Vasista, Utkarsh Gupta, Nayana Magadi Nagaraj, Bhakti Paithankar

Date last updated: 12/7/2025

## Vision

CampusSpot envisions transforming the way students at Northeastern University access and utilize study spaces. By providing a centralized platform that allows users to locate and book study areas in real-time, monitor noise levels, and ensure the availability of essential resources, CampusSpot empowers students to focus on what truly matters: their academics. By leveraging cutting-edge, CampusSpot also equips university administrators with actionable insights to optimize resource allocation and improve space management. We are committed to fostering a more productive, stress-free, and efficient campus environment—making CampusSpot an indispensable tool for every student and staff member.

## Motivation

The development of CampusSpot arises from the need to address critical challenges faced by university students and administrators in managing study spaces effectively. In today's post-pandemic academic environment, the demand for shared spaces has increased significantly due to the return of in-person learning and rising enrollments. However, this shift has amplified the inefficiencies of existing systems, which often fail to meet the dynamic and unique needs of campus environments. CampusSpot aims to bridge this gap by providing a centralized platform that enhances study experience for students and empowers administrators with actionable data to optimize resource utilization.

### Challenges Faced by Students

Our primary users—undergraduates, graduate students, and researchers—consistently report frustration with finding accessible and quiet study spaces equipped with the necessary resources. This problem occurs particularly during high-demand periods such as midterms and finals, when study space shortages disrupt student productivity. Based on interviews with 20 Northeastern University students and observational studies, we identified the following key pain points:

#### 1) Scarcity of Available Seats

Students experience significant stress and wasted time due to the lack of real-time information about available seating. International students, in particular, rely heavily on campus resources for extended study hours and consistently highlighted this issue. A typical scenario involves students spending 15–30 minutes searching for a space, cutting into valuable study time.

#### 2) Disruptive Study Conditions

With limited options, students are often forced to settle for less-than-ideal study environments, such as noisy or overcrowded spaces. This problem directly impacts their ability to focus and perform effectively during critical study sessions.

### 3) Inefficient Booking Solutions

Existing booking systems, such as Robin, fail to fully address student needs:

- Real-Time Noise Monitoring: This feature is absent on most platforms, leaving students unable to gauge the environment before booking a space.
- Inflexible Booking Options: Current solutions often lack cancellation or mid-session adjustment capabilities, leading to unused spaces or inefficiencies.
- Limited Customization: Students are unable to filter spaces based on their specific needs, such as equipment availability or group collaboration setups.

These shortcomings emphasize the need for a solution tailored to the unique dynamics of university life and these are evident from the pie charts below as well which are based on the customer development research performed by the team.

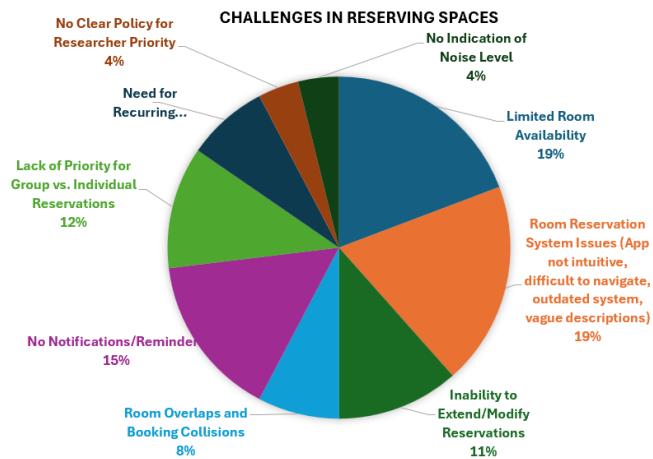


Figure 1 Challenges in reserving spaces on campus

The top challenges in reserving space included limited room availability and problems with the room reservation system, each accounting for 19% of the difficulties. Notifications and reminders are also a notable challenge, followed by issues with extending or modifying reservations.

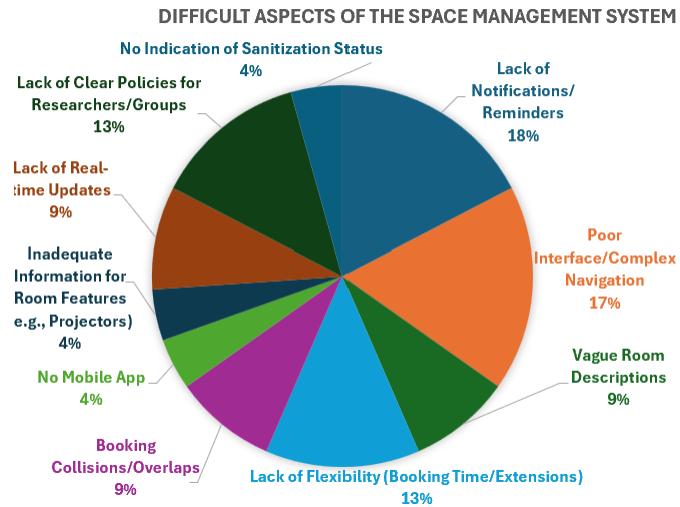


Figure 2 Difficult aspects of the existing space management system

The most prominent issue with the existing space management system is the lack of notifications or reminders and poor interface or complex navigation. Other significant challenges are the lack of flexibility in booking time or extensions and vague room descriptions.

### Needs for Administrators and Staff

While students are the primary audience, university staff and administrators are a valuable secondary user group. These individuals are responsible for ensuring optimal space utilization and addressing student complaints. However, they face several challenges due to a lack of real-time insights and data-driven tools.

#### 1. Lack of Real-Time Data

Administrators struggle to gain visibility into seating demand patterns. For example, during peak study periods, some sections become overcrowded while others remain underutilized. Without actionable data, staff cannot effectively redistribute seating resources to balance usage and reduce complaints.

#### 2. Limited Insight into Student Preferences

Administrators lack visibility into students' specific seating needs, such as preferences for quiet zones, group study areas, or additional equipment like whiteboards. This absence of data limits their ability to proactively adapt study spaces to meet evolving demands.

#### 3. Inefficient Resource Allocation

Without analytics, library staff cannot predict or manage high-demand times effectively. This results in overcrowded spaces, inefficient usage, and suboptimal resource allocation, leading to dissatisfaction among students.

## The Competitive Landscape

Existing platforms such as Robin, LibCal, and Envoy provide general booking solutions but fail to cater specifically to the needs of university campuses. Key limitations of these platforms include:

### 1. Lack of Noise Monitoring

Very few solutions offer real-time noise tracking, which is critical for students seeking focused study environments.

### 2. Limited Campus Integration

Most platforms are designed for corporate environments and primarily integrate with tools like G Suite or Outlook. They lack deep integration with university-specific systems such as student ID cards, library management platforms, or equipment booking tools.

### 3. Inflexibility and Poor Usability

Many existing tools lack intuitive interfaces or features like in-session adjustments, cancellations, or custom notifications, resulting in wasted resources and frustration for users.<sup>[2]</sup>

## Differentiation and Timing

As Northeastern University students, we possess firsthand knowledge of the challenges faced by both students and administrators. This personal insight drives the development of CampusSpot, a platform uniquely designed for academic environments. Key differentiators include:

- Real-Time Noise Monitoring: Helps students find quiet study zones before booking.
- Dynamic Booking Options: Allows for flexible reservations, cancellations, and mid-session adjustments to reduce wasted space.
- Data-Driven Optimization: Empowers administrators with analytics to improve space allocation and reduce student complaints.<sup>[2]</sup>
- General area seat tracking: When the study rooms are not available to book, students or faculties can track the seats available in the general study space areas and can occupy them without wasting any time.

## Why Now?

The timing is critical as campuses recover from the pandemic and adapt to a renewed demand for shared spaces:

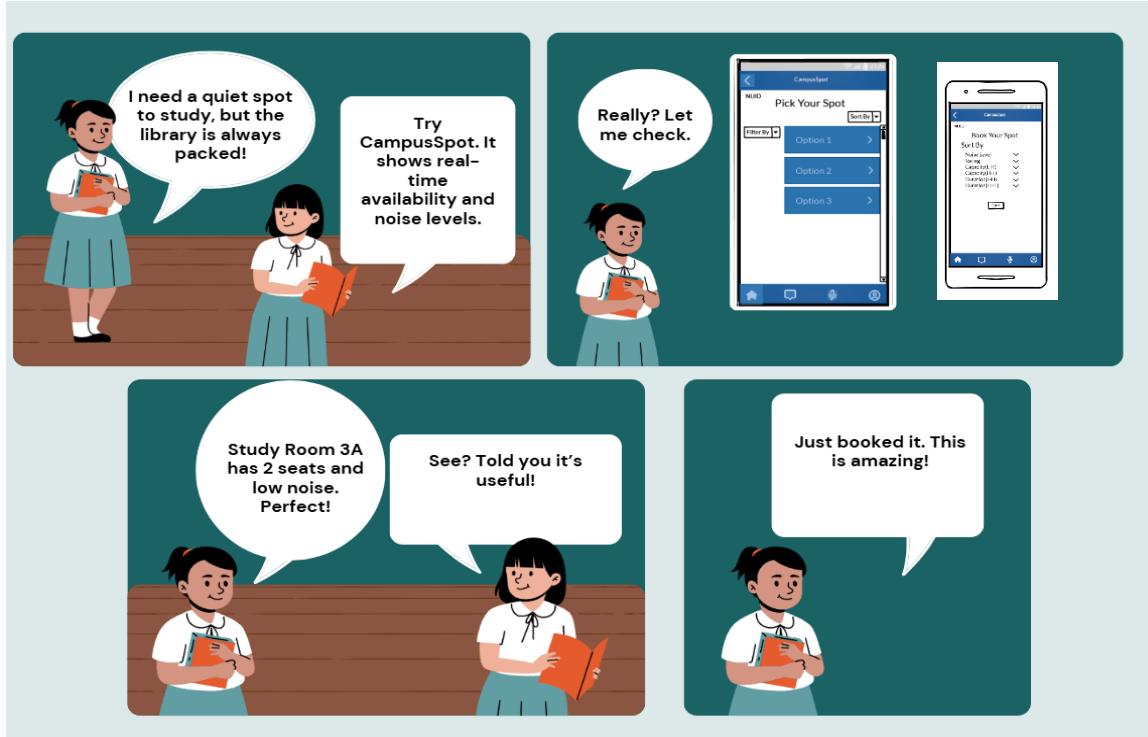
- 1) Rising Enrollments: Universities are accommodating larger student bodies, increasing competition for resources.
- 2) Growing Focus on Collaboration: Group work and in-person activities are driving the need for more adaptable study environments.
- 3) Technological Advancements: IoT sensors and real-time data tracking now enable smarter solutions that were previously impractical.

By addressing these needs, CampusSpot is poised to redefine space management and enhance the overall campus experience.

## Verbal/Visual Walkthrough of Use Cases

### 1) Use Case 1: Finding and Reserving a Study Space

Persona: Emma (Undergraduate student, always on the move, cramming for finals.)  
Emma and her friend Emma talking about the special filter available on the app.



### 2) Use Case 2: Submitting a Maintenance Request

Persona: Sophie (Graduate student, works late hours, relies on quiet, well-maintained spaces.)  
Sophie discussing with her friend about submitting a request to staff through app.



### 3) Use Case 3: Managing Staff Requests

Persona: Kiara (Staff, ensures smooth campus operations.)

A conversation between Kiara and the maintenance staff regarding handling requests.



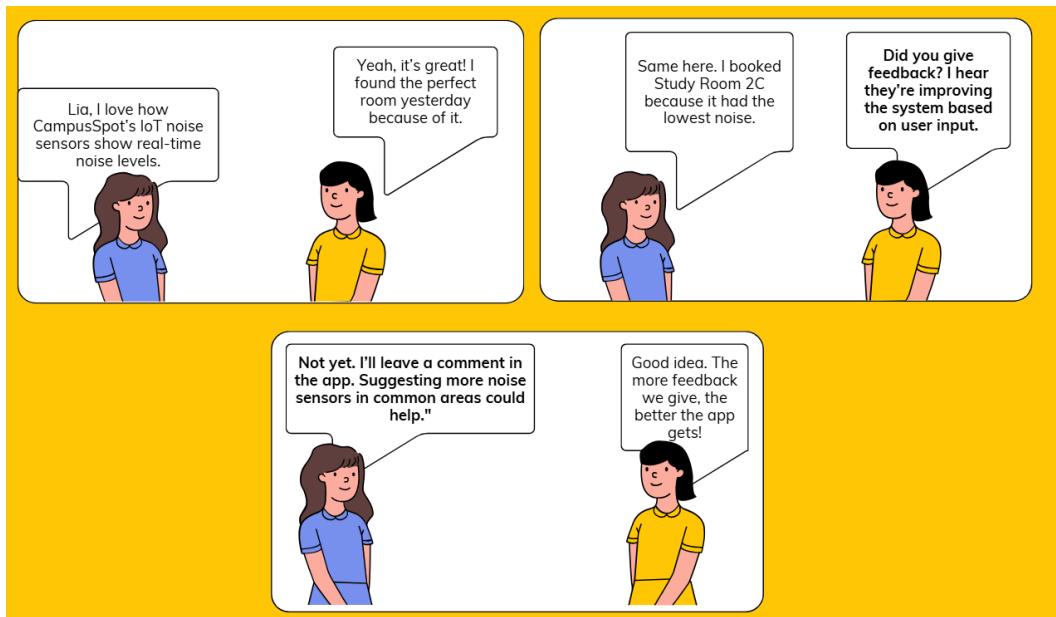
#### 4) Use Case 6: Collaborative Group Project Space

Persona: Maya and Team (A group working on a big capstone project for an online.) They are discussing having a room booked in a recurring manner for online class project.



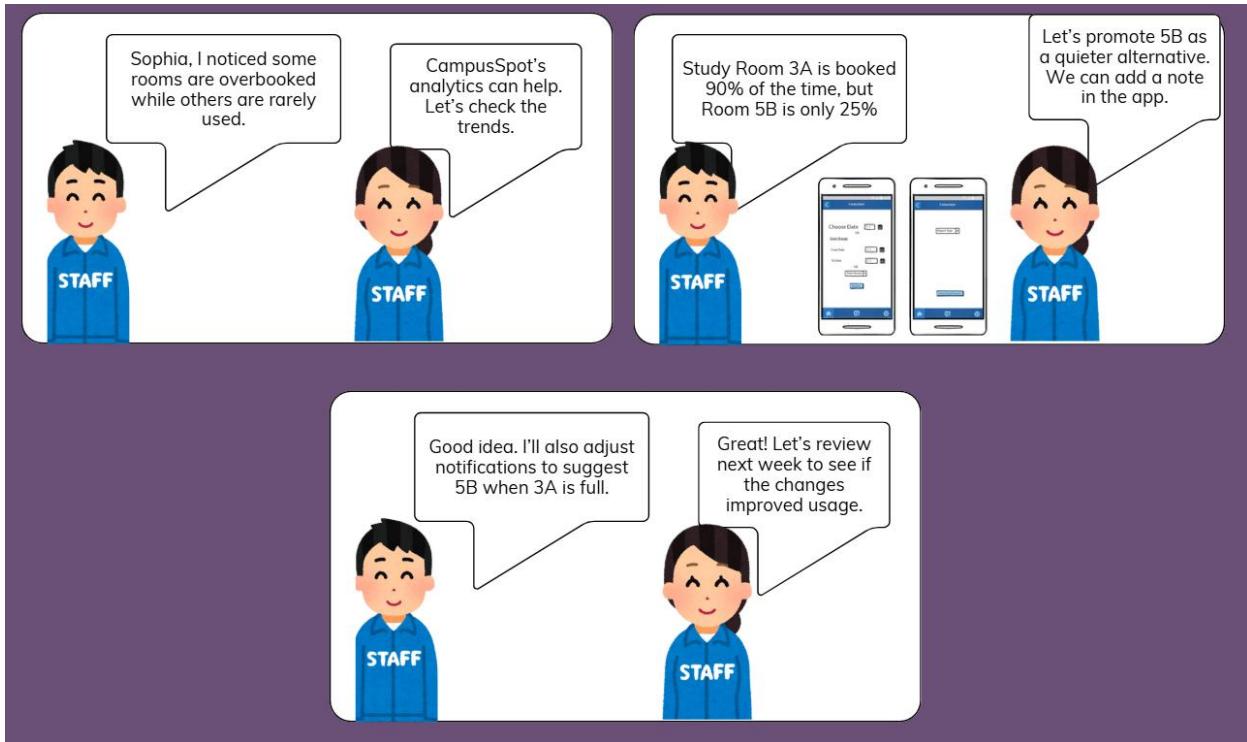
#### 5) Use Case 8: Students Using IoT Sensors and Providing Feedback

Persona: Aria (Sophomore prefers quiet study spaces) and Leo (First-year, tech-savvy and curious).



## 6) Use Case 7: Staff Reviewing Room Booking Trends

Persona: Alex (Campus Manager, analyzes trends to optimize space usage) and Sophia (Site Administrator, manages daily operations).



# Detailed Design & Features Description

## Design Principles

The design principles behind CampusSpot are driven by a commitment to delivering a simple, intuitive user experience that maximizes student productivity. Here is how we will approach these principles:

### 1) Simplicity Over Complexity

We aim to provide a clean, easy-to-navigate interface, focusing on minimalism and intuitive design, making sure users can find study spaces quickly without unnecessary steps.

### 2) User-Centered Design

We will prioritize features that directly solve user problems, like real-time seat availability and noise level management, ensuring the app evolves based on user feedback.

### 3) Backward Compatibility and Integration

For universities like Northeastern, it is important that CampusSpot integrates seamlessly with existing campus management systems. This includes integrations with student ID systems, library tools, and event management apps.

#### 4) Scalability and Futureproofing

As CampusSpot expands to multiple campuses and incorporates new features, we will focus on scalability. We will use cloud-based infrastructure and modular software design to allow easy scaling.

#### Few Sample Screens

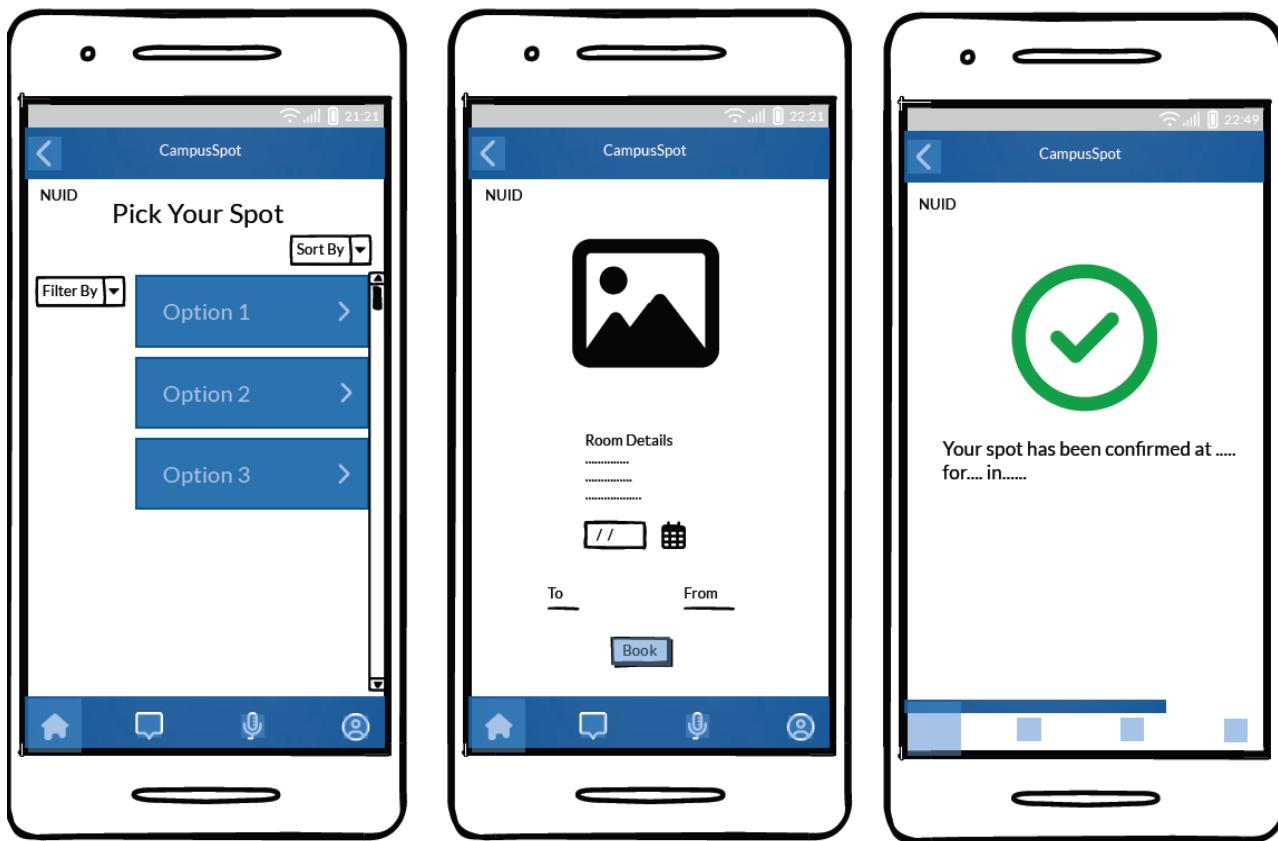


Figure 3 Room Booking Flow - User-Centered Design



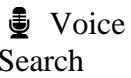
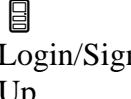
Figure 4 Screen when a staff member tries to get reports to study trends related to booking

### Features/information architecture

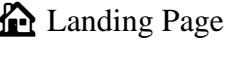
The CampusSpot app serves a central purpose: to provide students with real-time insights into study space availability, noise levels, and more. Below is a detailed breakdown of key features and the overall architecture:

#### Feature Table: Students

Sr. No.	Feature	Description	Dependencies
1	Login/Sign-Up	Students log in using their NEU ID or create a new account.	Integration with NEU authentication systems, secure password storage, and email verification.
2	Landing Page	Quick-access actions (e.g., book room, check schedule, submit requests).	Backend to retrieve personalized data, UI navigation setup, and notification system.
3	Room Booking	Students filter and sort rooms, view details, and confirm bookings.	Real-time room availability database, filter/sort algorithms, and booking confirmation notifications.

4	 Schedule Viewer	Displays a summary of booked schedules for easier planning.	Calendar integration and database for storing scheduled bookings.
5	 Request Form	Students submit requests to staff for specific needs or concerns.	Backend for form submissions, routing logic to staff, and automated acknowledgment messages.
6	 Notifications	Displays updates and reminders (e.g., booking status, alerts).	Notification service integration and user-specific update retrieval.
7	 Voice Search	Enables quick navigation or actions using voice commands.	Voice recognition API, predefined command mapping, and voice-to-text conversion.
8	 Profile Management	Students view and update their profile details, including contact information.	Secure data storage, retrieval systems for user profiles, and dynamic update capability.
9	 Emergency Support	Quick-access options for reaching out during emergencies.	Emergency contact integration and routing for escalations.
10	 Contact Staff	Allows students to directly reach out to building staff for assistance.	Database of staff details and messaging feature integration.
11	 Login/Sign-Up	Ensures app usability for students with disabilities (e.g., voice navigation).	Compliance with accessibility standards (e.g., WCAG), screen reader and keyboard navigation support.

Feature Table: Staff

Sr. No.	Feature	Description	Dependencies
1	 Login	Staff log in using their NEU credentials to manage responsibilities.	Integration with NEU authentication systems and secure password handling.
2	 Landing Page	Quick-access actions (e.g., contact students, generate reports).	Backend support to fetch required data and facilitate smooth navigation.
3	 Report Generation	Staff generate and download reports by applying filters (e.g., date range).	Database integration for report data retrieval and export functionality (e.g., PDF/Excel).

4	 Contact Students	Staff contact students for specific reasons (e.g., rule violations, emergencies).	Backend routing for staff-initiated messages and notification services.
5	 Notifications Panel	Displays incoming student requests and general updates for staff review.	Request and notification database integration for real-time updates.
6	 Profile Management	Staff view and update personal details and settings within their profile.	Secure data retrieval and update mechanisms for user profiles.
8	 Emergency Alerts	Staff receive and respond to emergency notifications from students.	Emergency escalation system and real-time alert routing.
10	 Accessibility Features	Supports inclusive access for staff with disabilities (e.g., screen readers).	Compliance with WCAG standards and accessibility tools integration.

## v1 (Minimum Viable Product)

The MVP will address immediate needs: real-time space reservations and efficient communication.

### Priority 0 (P0)

- i. Signup and Login: Secure access using NEU credentials.
- ii. Real-Time Space Availability: View available seats and noise levels.
- iii. Space Reservations: Book spaces instantly.
- iv. Notifications: Get updates on bookings and alerts.
- v. Profile Management: Store user preferences and activity history.

### Priority 1 (P1)

- i. Group Bookings: Reserve spaces for small groups.
- ii. Request Form: Submit support requests to staff.
- iii. Staff Quick Actions: Manage bookings and student inquiries.

### Priority 2 (P2)

Basic Analytics Reporting: Reports of all hours booking and occupancy.

Voice Search: Use voice commands for quick actions.

## vNext (Next Features After MVP Launch)

- i. Focus: Enhancing user experience and expanding functionality.
- ii. Advanced Booking: Recurring and extended reservations.
- iii. Campus System Integration: Sync with events, dining, and shuttles.
- iv. Request Management: Improve staff tools for handling requests.

### vLongTerm (Future Vision)

- i. Goal: Make CampusSpot the productivity hub for campus.
- ii. Complete System Integration: Connect library, ID, and academic tools.
- iii. AI Recommendations: Suggest study spaces and schedules.
- iv. Collaborative Spaces: Book rooms for group projects or workshops.
- v. Expanded Use: Add features for fitness or event space reservations.

### Conclusion: A Vision for the Future of Campus Productivity

As we move forward with CampusSpot, the development and design principles outlined above will ensure that the app becomes not only a useful tool but also an essential platform for students at Northeastern and beyond. By focusing on real-time needs like seat availability and noise management and scaling up with features like predictive analytics and AI-powered recommendations, CampusSpot will continue to evolve into a critical part of the student experience, improving productivity and campus life.

## Roadmap / Timing

### How We will Roll Out CampusSpot: Key Milestones and Vision

The CampusSpot roadmap is designed to ensure a deliberate, thoughtful rollout, balancing speed, user experience, and operational efficiency. Each phase focuses on building trust with users, refining features, and creating a solid foundation for success.



Figure 5 Product Roadmap – Steady Growth, Smart Expansion

## Quarter 1: The Foundations – Internal Demo and App Launch

What will happen?

The CampusSpot app will debut internally as an MVP (minimum viable product). This phase involves testing with internal stakeholders, addressing any bugs, and ensuring the app is ready for real-world environments.

Goals:

- i. Launch the app to achieve at least 1,000 active users by the end of Q1.
- ii. Refine the core features for 95% accuracy in real-time updates and 80% reservation success rates.

## Why It Matters:

This initial stage is critical for building confidence within the team and ironing out major pain points. Feedback collected from internal testers and a small pool of early users will directly inform our next steps.

## Reassessment Point:

After the internal demo and app launch, we will evaluate:

- Are we meeting core functionality expectations?
- What is broken or confusing for users?
- What should we prioritize for the beta phase?

## Quarter 2: Gaining Momentum – Beta Launch and User Feedback Collection

### What's Happening?

This is where CampusSpot enters into the hands of real users. It will be a limited release to select pilot campuses, mostly the Boston area, to gather structured feedback on usability, functionality, and performance. Key enhancements will also roll out during this phase.

### Focus Areas:

- i. UI/UX Enhancements: Improve the overall design to boost ease-of-use ratings in app reviews.
- ii. Booking Flexibility: Introduce new booking features with a goal of 50% user adoption by the end of Q2.
- iii. Noise Monitoring Upgrade: Increase satisfaction scores around noise-tracking features by a minimum of 25%.

## Why It Matters:

This phase allows us to validate features in a live environment while still operating in a controlled setting. By the end of Q2, we will not only know what users love but also where they get stuck.

## Reassessment Point:

Before expanding to full launch, we will revisit the user feedback to ensure the app delivers on its promises. Which is a way of telling Are we solving the problems we set out to address?

## Quarter 3: Scaling Up – Full Launch and Campus Integrations

### What's Happening?

This is the pivotal quarter when CampusSpot goes live for a larger audience. By this point, the app will integrate with at least two additional campus systems, such as scheduling and student life services, to provide seamless user experience.

## Focus Areas:

- i. Admin Dashboard Phase 2: Deliver actionable insights for 85% of admin needs, giving administrators better tools to manage reservations and usage data.
- ii. Integration with Campus Systems: Ensure deep integration with existing systems to improve user inter-app usage by.

## Why It Matters:

Scaling up means moving from a promising prototype to a fully functional, widely adopted solution. Successful integration ensures CampusSpot becomes part of daily campus operations, increasing adoption and loyalty among users.

## Reassessment Point:

After the full launch, we will assess the health of the platform:

- Are the systems stable at scale?
- Are the features delivering value to both students and administrators?

## Quarter 4: Enhancing the Experience – Expansion and Refinement

### What's Happening?

With the foundation in place, we will focus on elevating the CampusSpot experience by introducing predictive features, testing in new environments, and growing our user base through strategic marketing.

## Focus Areas:

AI-Based Recommendations: Help users make smarter decisions with personalized suggestions, aiming for 70% adoption and 4/5 satisfaction ratings in surveys.

- i. Trend Forecasting: Reduce search times during peak hours by 25% and optimize off-peak usage by 10%.
- ii. Satellite Campus Pilot: Test CampusSpot features in a new environment, ensuring they meet the unique needs of different student populations.
- iii. Targeted Campaigns: Launch a focused marketing effort to onboard 2,000 new users and boost first-year adoption by 30%.

## Why It Matters:

This phase ensures we are not just meeting expectations but exceeding them. The introduction of predictive and AI-powered features solidifies CampusSpot as a must-have tool for campus life.

## Scenarios for Service Introduction

### Why This Rollout Plan Is the Right One

We considered alternatives, such as skipping beta testing for a faster full launch or narrowing our focus to a specific audience. However, these approaches would have introduced risks. By following this phased plan:

Alpha Testing (Internal Demo): We ensure our foundation is rock-solid before scaling up.

Beta Testing (Q2): We collect valuable feedback from real users, allowing us to fine-tune the experience.

Full Launch (Q3): A carefully calibrated release ensures stability and strong user adoption.

Expansion (Q4): We introduce advanced features and broaden our reach after proving the app's core value.

### Timing Sensitivities

i. Academic Calendar Alignment:

Launching at the start of the semester ensures high visibility and adoption among students and administrators.

ii. Competitive Landscape:

Timing our full launch before similar products enter the market gives CampusSpot a first-mover advantage.

iii. Campus IT Cycles:

Integration schedules need to align with other campus technology upgrades to avoid delays or compatibility issues.

### What This Means for Execution

This roadmap will translate into a detailed Gantt chart with specific tasks, dependencies, and deadlines. From internal demos to beta testing and full launch, we will measure success through clear KPIs like adoption rates, user satisfaction, and system performance.

## Metrics

### Key Metrics

Metric	Description	How to Compute	Example
Search-to-Booking Conversion Rate	Measures the percentage of searches that result in successful bookings.	Conversion Rate (%) = = (Successful Bookings / Total Searches) × 100	If 1,000 searches lead to 700 bookings: $(700/1,000) \times 100 = 70\%$
Real-Time Occupancy Accuracy	Measures the accuracy of real-time updates for seating availability.	Accuracy (%) = (Correct Availability Updates / Total Updates) × 100	If 950 out of 1,000 updates are accurate: $(950/1,000) \times 100 = 95\%$

User Satisfaction Score	Reflects average user ratings of booking experience and app usability.	Average of user feedback ratings (e.g., 1–5 stars)	If 50 users give a total of 220 stars: $220 \div 50 = 4.4/5$
Failure Rate of Booking	Monitors the frequency of booking failures due to technical issues or seat unavailability.	Failure Rate (%) = $(\text{Failed Bookings} / \text{Total Booking Attempts}) \times 100$	If 50 out of 1,000 booking attempts fail: $(50/1,000) \times 100 = 5\%$
Daily Active Users (DAU)	Counts the number of unique users logging in each day.	Count of unique logins each day	Count of unique users logging in per day
Monthly Active Users (MAU)	Tracks unique users over a month, indicating overall adoption and retention.	Count of unique logins over 30 days	Count of unique users over a month
Noise Level Accuracy	Assesses how accurately the app reports noise levels in study areas.	Compare app-reported noise levels with actual measurements (via sensors or manual checks)	If the app reports noise levels align with actual measurements 95% of the time, 95% accuracy
Peak Usage Insights	Detects the busiest times for study space usage to help staff manage resources.	Analyze timestamps of searches and bookings to identify peak usage hours	Use search and booking data to identify 12 PM – 2 PM as the busiest times
Churn Rate	Measures the percentage of users who stop using the app over a period.	Churn Rate (%) = $(\text{Users Lost During Period} / \text{Total Users at Start of Period}) \times 100$	If 100 out of 1,000 users stop using the app: $(100/1,000) \times 100 = 10\%$
App Crash Rate	Tracks the number of app crashes per session, indicating app stability.	Crash Rate (%) = $(\text{App Crashes} / \text{Total Sessions}) \times 100$	If the app crashes 10 times in 1,000 sessions: $(10/1,000) \times 100 = 1\%$
Search Abandonment Rate	Measures the percentage of searches that do not result in bookings.	Search Abandonment Rate (%) = $(\text{Uncompleted Searches} / \text{Total Searches}) \times 100$	If 200 out of 1,000 searches are abandoned: $(200/1,000) \times 100 = 20\%$

Reservation Cancellation Rate	Measures the percentage of canceled bookings, indicating usability issues.	Cancellation Rate (%) = (Canceled Reservations / Total Reservations) × 100	If 50 out of 1,000 reservations are canceled: $(50/1,000) \times 100 = 5\%$
-------------------------------	--	--	---

## Log Data and Statistics Indicating Success

### 1) Basic Traffic Metrics:

- Daily Active Users (DAU) and Monthly Active Users (MAU): Tracks app usage over time.
- How It is Computed: Count the number of unique users logging into the app daily and monthly. A high DAU/MAU ratio indicates frequent engagement.

### 2) Search and Booking Metrics:

- Peak Usage Insights: Identifies the busiest times, helping staff optimize resource allocation.
- Search Abandonment Rate: Tracks users who search but do not complete bookings.  
Formula: Search Abandonment Rate =  $(\text{Searches without Bookings} / \text{Total Searches}) \times 100$ .  
A lower rate indicates successful search results.

### 3) Evidence of Action Taken:

- Metrics like Reservation Cancellation Rate and Churn Rate (percentage of users who stop using the app) can indicate user satisfaction or frustration.
- Direct feedback, such as comments or issues, can validate app impact and responsiveness.

## International

CampusSpot will be adapted to meet the unique requirements of universities globally. The focus will remain on addressing common difficulties such as finding study places, measuring noise levels, and optimizing resource use, while taking into consideration regional variations in campus layouts, cultural preferences, and operational norms.

### Timeline for International Versions

#### Phase 1 (12-18 months post-launch):

- Deploy CampusSpot to other colleges in the United States to assess scalability and develop features for a variety of campus structures.

#### Phase 2 (18-24 months post-launch):

- Expand into other countries such as Canada, the United Kingdom, and Australia that have a high need for campus space efficiency technologies.
- Tailor features to meet regional requirements, providing smooth integration with local systems.

Phase 3 (24+ months post-launch):

- Create tailored versions for smaller universities and institutions in underserved areas.
- Create lightweight CampusSpot versions for use in environments with limited IT infrastructure and internet connectivity.

Phase 4 (3+ years):

- Introduce advanced features like AI-powered predictive analytics for seat trends and resource planning.
- Partner with worldwide university networks to standardize CampusSpot across numerous campuses, ensuring a consistent experience for students studying abroad.

## Key Challenges in Internationalization

### I. Localization Needs:

Translating the software into several languages to better serve international users. UI/UX customization for varied cultural expectations, such as date formats, time zones, and app design preferences.

### II. Integration with Diverse Systems:

Universities around the world utilize various student management systems and booking platforms. CampusSpot will require APIs or interfaces to provide seamless connection with diverse infrastructure.

### III. Regulatory Compliance:

Adhering to privacy laws such as GDPR (Europe), PIPL (China), and other country-specific restrictions. This includes anonymizing user data and establishing local data centers where necessary.

### IV. Noise Level Calibration:

Study environments and cultural norms about acceptable noise levels vary by region. CampusSpot's noise tracking algorithms will need to be updated to meet regional standards.

### V. Infrastructure Dependence:

In countries with inconsistent Wi-Fi access, the software has to include offline functionality and lightweight features to work with limited connectivity.

## Strategic Plan for Addressing Challenges

- Localized User Testing: Collaborate with pilot universities in key locations to test and improve features prior to full deployment.
- Modular Design: Create a modular backend system that can accommodate a variety of university infrastructures and interface seamlessly with third-party services.
- Global partnerships: Collaborate with local tech firms on language localization and regulatory compliance.

- Focus on regional customization: Include customizable noise and seating preferences that align with cultural expectations, e.g., quiet libraries in Japan versus more lively group study places in the United States.

## Why should we internationalize?

The global market for digital space optimization is quickly expanding, with universities increasingly using smart solutions to increase resource utilization. CampusSpot's foreign expansion will allow it to tap into a \$1.5 billion+ market while also providing unique solutions to students worldwide.<sup>[6]</sup>

# Projected Costs

To successfully launch CampusSpot and scale it to meet the needs of Northeastern University and future campuses, we need to carefully plan and justify each expense, from engineering efforts to hardware and infrastructure. Our goal is to ensure that every dollar of the budget is strategically allocated to deliver maximum impact while aligning with the roadmap's objectives. Breakdown of the key components and the thought process behind each cost to bring the project to life is as below.

### Engineer Months and Project Workforce Estimation

To determine the required Engineer Months for the project, we broke down the work by quarterly milestones and project tasks, estimating the number of engineers needed for each phase based on the roadmap's complexity. Here is how we arrived at that figure:<sup>[5]</sup>

- **Core Features and MVP Launch (Q1): 28 Engineer Months**

We have estimated that delivering real-time booking accuracy and managing concurrent user loads for 1,000 active users in the first quarter would require 28 Engineer Months. The foundational tasks for Q1 are prioritized to deliver the minimum viable product (MVP). This includes:

- i. Frontend Development: Designing and implementing an intuitive user interface (UI) for seamless room booking and real-time status updates.
- ii. Backend Development: Creating robust APIs to manage real-time bookings with up to 1,000 concurrent users, ensuring the database design supports scale and low latency.
- iii. Integration Work: Initial integration with campus infrastructure such as student directories and limited campus resources.

### How 28 Engineer Months?

- i. Frontend required 2 engineers for 3 months = 6 months.
- ii. Backend tasks demanded 3 engineers for 3 months = 9 months.
- iii. System integration and database design involved 2 engineers for 3 months = 6 months.
- iv. QA testing allocated 1 engineer for 2 months = 2 months.
- v. Project management overhead contributed 5 months.

These efforts sum up to 28 Engineer Months.

- **Feature Enhancements and Admin Dashboard (Q2): 16 Engineer Months**

In Q2, the team aims to add noise-level monitoring and develop the admin dashboard to provide room usage analytics and manage availability. This phase also incorporates feedback from early users to refine the experience.

How 16 Engineer Months?

- i. Admin dashboard required 2 engineers for 2 months = 4 months.
- ii. Enhancements like noise monitoring needed 2 engineers for 3 months = 6 months.
- iii. Refactoring and feedback iteration involved 2 engineers for 2 months = 4 months.
- iv. QA and support required 2 engineer months for testing and fixing bugs.

The complexity of these features is slightly lower than Q1 but require deliberate focus on scalability and usability improvements.

- **System Integration and Scaling (Q3): 24 Engineer Months**

Q3 focuses on campus-wide integration with tools like:

- Student ID systems
- Library resource management
- Security and scheduling tools

These integrations require careful handling to ensure the app worked seamlessly with existing systems, especially in a dynamic campus environment.

Why 24 Engineer Months?

- i. Backend API extensions needed 3 engineers for 3 months = 9 months.
- ii. Security and identity integration took 2 engineers for 3 months = 6 months.
- iii. Database scaling for larger concurrent use required 2 engineers for 3 months = 6 months.
- iv. QA testing for these systems required 1 engineer for 3 months = 3 months.

- **Predictive Features and Expansion (Q4): 26 Engineer Months**

The final quarter focuses on introducing AI-driven predictive features, such as recommending the best study spaces based on historical trends and user preferences. This also laid the groundwork for expanding the product to new campuses.

Why 26 Engineer Months?

- i. Data science tasks for AI models involved 2 engineers for 4 months = 8 months.
- ii. Backend updates for predictive algorithms required 3 engineers for 3 months = 9 months.
- iii. UI updates for recommendations required 2 engineers for 3 months = 6 months.
- iv. Testing and debugging required 1 engineer for 3 months = 3 months.

AI-related work inherently required more resources due to the complexity of ensuring prediction accuracy and system responsiveness.

- **Quality Assurance and Testing (Throughout): 8 Engineer Months**

To ensure a reliable and user-friendly product:

- Resolute QA testing engineers reviewed each phase of development.
- They focused on performance, security, and usability.

This testing effort is spread across the project timeline and is essential for a seamless launch. The breakdown is that 2 engineers worked part-time for about 2 months per quarter, contributing 8 months over the year.

By summing up the contributions from each phase:

- Q1: 28 months
- Q2: 16 months
- Q3: 24 months
- Q4: 26 months

This adds up to 94 Engineer Months, reflecting the carefully estimated effort required for a successful project launch.

This estimation approach reflects the industry's best practices by:

- 1) Leveraging Agile Development Cycles: Breaking down work into sprints and allocating resources for specific tasks.
- 2) Drawing from Comparable Projects: Referencing similar app development cycles in tools like WebSys or successful EdTech platforms.
- 3) Buffer for Complexity and QA: Allocating sufficient QA and integration time based on lessons learned from campus-wide software projects.

This thorough methodology ensures the number is realistic and accounts for the project's unique challenges and goals.

### Justification for the \$50,000 Allocation (Core Development Focus)

Now, while the total engineering effort adds up to 94 Engineer Months, the budget for development—specifically the \$50,000 allocated to Team Power—focuses on core tasks critical for achieving key milestones.

- **Core Development Tasks**

The \$40,000 is allocated toward the development of critical features needed for the MVP launch in Q1. This includes:

- Frontend development for an intuitive user interface.
- Backend work for integrating the booking system and real-time data management.

Given the \$5,000/month average cost per engineer, this allocation covers a targeted 10 Engineer Months of the total 94, ensuring that the most crucial work is prioritized.

- Cost Savings Through Automation and Tools

By leveraging automated testing tools (\$10,000) and working in close partnership with university IT teams, we can reduce the engineering hours required for non-core tasks like backend integrations. This effectively saves about \$100,000–\$125,000, which we allocate to optimize user experience and scalability.

- Deferred Features

More advanced features (like AI-powered recommendations and multi-language support) are deferred to later phases (Q3 and Q4), which help spread the engineering costs across the project. This also allows us to reallocate resources from non-essential tasks to prioritize the MVP launch.

By strategically allocating \$50,000 toward essential engineering tasks, we can deliver a robust MVP and set the stage for further development and scalability.

## Total Hardware Costs with IoT Devices Included

To ensure CampusSpot functions efficiently both during launch and as we scale to other campuses, we must account for both hardware and cloud infrastructure costs. Here is how we justify each expense:

Total Hardware Costs: \$96,200

- Workstations for enhancements: \$20,000

We will need 10 workstations (Intel processors) for development, testing, and design teams. These machines are essential for rapid coding, prototyping, and iterative design processes.

- Cloud Infrastructure: \$43,200

We have budgeted for 3 cloud servers (AWS or Azure) to oversee the app's backend, APIs, and data storage. Each server costs \$800/month, and over 6 months, this totals \$14,400. <sup>[1][6]</sup>

- IoT Devices for Noise Monitoring: \$13,000

130 IoT devices (priced at \$100 each) will be deployed to monitor noise levels across study spaces. These devices are crucial for the real-time functionality of the app, ensuring that users can view accurate data when choosing study locations.<sup>[7]</sup>

- Network Upgrades: \$10,000

CampusSpot's success hinges on fast, reliable Wi-Fi for real-time data transmission. Therefore, an allocation of \$10,000 will go toward upgrading Northeastern's campus network, ensuring seamless device communication across all study spaces.<sup>[1]</sup>

- Integration Hardware: \$10,000

This includes server and integration costs for linking CampusSpot with existing university systems, such as student IDs and library management tools, which is necessary for a smooth user experience.<sup>[1]</sup>

## Total Projected Costs and Justification

Category	Investment (₹)	What It is For
🛠 Development	\$50,000	Powering up with a rockstar team and innovative tools to build a robust app foundation.
⌚ UX/UI Enhancements	\$20,000	Crafting an app interface so smooth and intuitive, users will love every click and swipe.
🗣 Feedback Collection	\$10,000	Setting up channels to hear user voices and fine-tune CampusSpot based on real-world needs.
⚙ Infrastructure & Integration. <sup>[5]</sup>	\$53,000	Ensuring CampusSpot plays nice with university systems and handles all the tech heavy lifting.
🔊 IoT for Noise Monitoring	\$13,000	Installing smart noise sensors to keep study spaces calm and distraction-free.
📣 Marketing & Outreach	\$15,000	Making sure everyone on campus knows about CampusSpot and cannot wait to try it out.
📡 Network Upgrades	\$10,000	Boosting Wi-Fi and ensuring smooth operation, even in low-connectivity zones.
💰 Contingency Fund	\$17,000	Preparing for the unexpected—because every great project needs a safety net!
Total	\$188,000	

- Apart from all the costs mentioned in the previous sections, marketing & outreach is estimated to cost around \$15,000 as spreading the word becomes vital.
- Also, unforeseen challenges are inevitable in any project. We will hence have a contingency fund of 10% of total value which comes up to \$17,000 to ensure we are ready to adapt, evolve, and overcome anything thrown our way, from server scaling to emergency fixes.

Each cost in this breakdown is specifically aligned with the quarterly goals of the project, ensuring maximum impact while keeping the project within budget.

## Why This Budget Works

- Strategic Allocation: Every dollar is tied directly to a roadmap goal, ensuring that the app is developed efficiently with resources allocated for core tasks and critical features.

- Long-Term Vision: This is not just about creating an app—it is about building a scalable, reliable solution that will evolve over time. The budget reflects a lean yet impactful approach, with careful prioritization of high-return features.
- Optimization and Efficiency: By leveraging automated tools and partnerships with campus IT teams, we are able to reduce costs significantly and focus engineering resources on the tasks that will drive real value.

## Conclusion

The \$188,000 budget for CampusSpot is strategically designed to deliver both short-term results and long-term scalability. Through thoughtful allocation of engineering resources, hardware investments, and strategic partnerships, we ensure that CampusSpot will empower students, increase productivity, and optimize university resources, providing an indispensable tool for academic success.

This is not just a financial plan—it is the foundation for the future of campus management!

## Revenue Model Insight

For Northeastern University, the revenue model for CampusSpot will initially focus on a university-wide subscription plan tailored to its specific needs and campus size. Northeastern would subscribe to the platform on an annual basis, starting with a base package that includes core features such as real-time room booking, noise-level monitoring, and basic administrative tools for space management. This foundational tier ensures an efficient rollout of the app while addressing the immediate pain points of students and administrators.

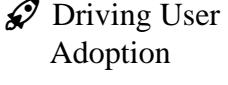
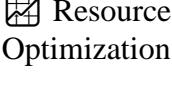
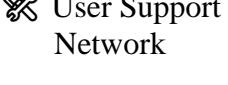
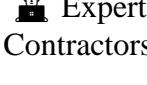
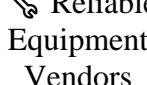
As the platform proves its value and adoption grows, Northeastern could upgrade to a premium plan, unlocking advanced functionalities like predictive analytics for optimizing space usage, AI-powered recommendations, and customized dashboards for data-driven decisions. The subscription cost would be scaled with the number of active users and additional features selected, offering flexibility to accommodate Northeastern's evolving requirements.

To drive additional revenue, add-on services like integration with existing campus systems (e.g., library resources, student ID platforms) and custom branding options can be offered. This tailored approach ensures that Northeastern benefits from a solution optimized for its infrastructure while CampusSpot builds a reliable, recurring revenue stream that sustains the platform's development and future scalability.

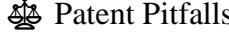
## Operational Needs

Operational Need	Description	Implementation Steps	Support Type
	Safeguard student data with anonymization and robust encryption to	Develop secure storage systems and implement encryption protocols to	Finite

 Data Privacy First	ensure privacy compliance.	protect sensitive information. <sup>[5]</sup>	
 Real-Time Insights	Deliver actionable data on space occupancy and usage patterns to empower administrators.	Integrate analytics tools for real-time and historical usage tracking.	Ongoing
 Seamless System Integration	Work collaboratively with university IT to ensure compatibility with student IDs and library systems.	Conduct rigorous compatibility testing and resolve dependencies.	Ongoing
 Always Connected	Optimize app performance for low-connectivity scenarios and ensure offline functionality when needed.	Collaborate with campus IT to enhance connectivity and implement offline features.	Ongoing
 Intuitive Interface	Design a user-friendly interface with guided onboarding tutorials to welcome new users.	Refine the design through user testing and incorporate helpful onboarding guides.	Finite
 Unified Campus Access	Integrate with existing platforms like MyNortheastern to streamline usability and enhance functionality.	Develop APIs to synchronize seamlessly with campus services and databases.	Ongoing
 Noise Level Tracking	Monitor and display real-time noise levels in study spaces to improve user experience.	Install advanced noise sensors or implement software to track and update noise data.	Ongoing
 Advanced Reservations	Empower users with features like recurring bookings, flexible cancellations, and reminders.	Develop a robust booking system with notifications and reminders for convenience.	Ongoing
 Equipment Availability	Display real-time availability of essential equipment like projectors and whiteboards.	Build an inventory management system for equipment tracking and reservation. <sup>[5]</sup>	Ongoing
 User Feedback Hub	Enable students to rate and provide feedback on spaces to drive continuous improvements.	Create an interactive feedback module with ratings and comment options.	Ongoing

 Risk Mitigation Measures	Address legal, privacy, and third-party concerns proactively for a secure platform.	Conduct patent reviews, privacy audits, and establish protocols for legal compliance.	Ongoing
 Driving User Adoption	Encourage widespread adoption with simple navigation, tutorials, and training sessions.	Launch engaging tutorials during orientation and maintain a feedback loop for iterative improvement.	Ongoing
 Resource Optimization	Provide administrators with tools to allocate spaces more effectively and reduce overcrowding.	Analyze data insights to inform better seating arrangements and resource utilization.	Ongoing
 User Support Network	Offer dedicated helpdesk and live chat support to address user concerns quickly.	Establish a responsive support team with ticket tracking and resolution tools.	Ongoing
 Expert Contractors	Engage technical specialists to streamline integration, database optimization, and overall performance.	Partner with contractors experienced in app development, database management, and analytics.	Finite
 Reliable Equipment Vendors	Source, maintain, and upgrade tools such as noise sensors and monitoring devices for uninterrupted functionality.	Collaborate with trusted vendors for quality equipment and ongoing maintenance.	Ongoing

## Addressing Caveats/Risks

Risk Category	What is the Challenge?	How We are Tackling It (Mitigants)
 Privacy Protection	How do we keep users' personal data safe while offering real-time monitoring and insights?	Establish transparent data policies, anonymize all user data, and deploy ironclad encryption to lock down sensitive info.
 Patent Pitfalls	Could our unique features like noise level tracking clash with existing patents?	Conduct deep-dive patent research, get expert legal advice, and creatively tweak functionalities to avoid conflicts. <sup>[6]</sup>

 Pirate-Proofing	What if someone tries to copy our standout features or misuse the tech?	Introduce cutting-edge encryption, enforce access controls, and safeguard our IP while rolling out frequent security updates.
 Tech Tango with University Systems	What if our integration with campus systems (like student IDs and library databases) hits snags?	Forge a tight partnership with university IT teams, conduct regular integration testing, and address dependencies swiftly.
 Third-Party Tech Trust	Relying on external vendors for noise monitoring or seat tracking tech sounds risky, doesn't it?	Partner only with proven vendors, build backup systems, and explore in-house alternatives to minimize risks. <sup>[6]</sup>
 Winning User Hearts	Will students and faculty find the app fun, intuitive, and easy to use?	Build an interface so simple it feels like second nature; offers interactive tutorials and make it a star of campus orientation.
 Operational Ups and Downs	What happens if real-time updates are disrupted by spotty campus Wi-Fi or connectivity hiccups?	Design the app to shine even under low-connectivity scenarios and add offline features to keep the show going. <sup>[6]</sup>

## References

- 1) HubStar. “Higher Education - Space Utilization.” Hubstar, May 2, 2022. <https://www.hubstar.com/blog/higher-education-space-utilization/>.
- 2) Christian-Farman, Luke. “5 Space Management Solutions for Higher Education Organizations - Kadence.” Kadence, May 23, 2024. <https://kadence.co/news/5-space-management-solutions-for-higher-education-organizations/>.
- 3) BB Agency, info@bb.agency, and Cody Slingerland. “SaaS Revenue Model: 5 Types to Consider in 2024.” CloudZero, January 10, 2024. <https://www.cloudzero.com/blog/saas-revenue-model/>.
- 4) Sasson, Maya. “SaaS Revenue Model - Meaning, Types, and Examples | Walnut.” Walnut, November 12, 2024. <https://www.walnut.io/blog/sales-tips/saas-revenue-model-types-and-examples/>.
- 5) ResearchGate. “(PDF) a Study and Overview of the Mobile App Development Industry,” n.d. [https://www.researchgate.net/publication/352490326\\_A\\_Study\\_and\\_Overview\\_of\\_the\\_Mobile\\_App\\_Development\\_Industry](https://www.researchgate.net/publication/352490326_A_Study_and_Overview_of_the_Mobile_App_Development_Industry).
- 6) LinkedIn.com. “LinkedIn,” 2024. <https://www.linkedin.com/pulse/risks-come-along-app-development-business-ways-avoid-them-moozetechn>.
- 7) Halstead, Jennifer. “The Evolution of IoT Technology Pricing | Link Labs.” Link-labs.com. Link Labs, June 2022. <https://www.link-labs.com/blog/evolution-of-iot-technology-pricing>.