ParkingNow:

A Human-Computer Interaction Analysis and System Design Proposal for Improving Urban and University Parking Experiences

Ca'Marai Clyburn

Table of Contents

Abstract

Parking remains a persistent challenge in both urban environments and university campuses, where drivers frequently encounter limited availability, unclear regulations, inefficient enforcement systems, and prolonged search times. These conditions contribute to wasted time, increased fuel consumption, and heightened stress, negatively impacting daily mobility and overall user experience. In dense metropolitan areas, drivers often rely on guesswork when searching for parking, while university campuses experience similar difficulties due to finite parking resources, peak-hour congestion, and outdated permit and enforcement systems. The ParkingNow application was developed in response to these challenges as a human-centered and scalable solution designed to improve parking efficiency for users while supporting administrative oversight. Initially conceptualized as a citywide parking-support platform, ParkingNow later evolved into a University Edition tailored to the behavioral patterns and operational needs of campus environments.

ParkingNow is examined through the integration of human–computer interaction principles, storyboard-based user journey analysis, iterative interface design, usability testing, and system-level planning. Emphasis is placed on reducing cognitive load, improving accessibility, minimizing user error, and enhancing visibility of system status. In addition to design and usability contributions, an exploratory machine learning component is introduced using an unsupervised autoencoder model applied to survey data collected from Hofstra University students. The model analyzes commuter behavior and identifies anomalous or inconsistent parking-related responses, serving as a supplementary decision-support mechanism. Experimental results demonstrate stable model convergence and high-precision anomaly detection, outcomes that align with theoretical expectations for reconstruction-based methods applied to low-dimensional survey data. While limitations exist due to dataset size and heuristic labeling, the results indicate that anomaly detection can provide meaningful behavioral insights when used alongside rule-based logic. Overall, the findings suggest that ParkingNow represents a viable technological intervention capable of reducing user frustration, improving parking workflows, and supporting smarter mobility management in both urban and academic settings.

1. Introduction

Parking remains one of the most persistent and universal challenges faced by drivers in both urban environments and university settings. In cities such as New York City, the difficulty of locating an available, legal, and conveniently located parking space often results in wasted time, increased fuel consumption, heightened stress, and diminished overall driving satisfaction. National reports show that nearly half of American drivers experience anxiety related to parking, underscoring its psychological and practical impact on everyday mobility. University campuses mirror many of these same frustrations, with students and faculty navigating complex regulations, limited parking supply, and peak-hour congestion.

The ParkingNow system was conceptualized as a direct response to these challenges. Initially designed as a citywide parking-support application, it aims to provide drivers with real-time information about parking availability, spot navigation, and streamlined decision-making. The project was later adapted into a University Edition that addresses the specific needs of campus communities, including digital parking verification, tailored navigation systems, and more intuitive enforcement processes. This thesis synthesizes human-computer interaction principles, design documentation, usability research, and system prototyping to evaluate the evolution, feasibility, and user-centered value of the ParkingNow application.

Drawing from project documents, design studies, and preliminary user feedback, the following analysis examines the multi-stage development of ParkingNow. The project demonstrates how an human-computer interaction driven approach can transform an everyday problem into a coherent technological solution that integrates user experience, accessibility, and administrative functionality. The goal of this thesis is to articulate both the conceptual foundation and practical engineering that support ParkingNow's evolution into a scalable, human-centered parking platform.

2. The Parking Problem

Parking is a complex and multifaceted issue that extends beyond the simple availability of physical spaces. It is shaped by a combination of infrastructural limitations, regulatory complexity, user behavior, and information gaps that affect both drivers and administrators. In many environments, parking systems have failed to evolve alongside increases in vehicle ownership and mobility demands, resulting in inefficiencies that compound over time. These inefficiencies manifest not only in logistical challenges, such as congestion and underutilized resources, but also in psychological burdens, including stress, uncertainty, and frustration. Understanding the parking problem therefore requires examining how these factors interact across different contexts, particularly in urban settings and university campuses, where demand is high and decision-making often occurs under time pressure.

2.1 Urban Parking Challenges

Urban parking environments, particularly in dense metropolitan regions such as New York City, present a complex ecosystem shaped by limited physical space, fluctuating demand, and inconsistent enforcement systems. Drivers often encounter a shortage of open curbside spaces, long travel times between potential parking zones, and competing rules communicated through confusing or contradictory signage. These conditions frequently lead to “circling,” a phenomenon in which drivers loop blocks repeatedly in search of a legal spot, consuming time, gasoline, and attention.

Beyond availability, the emotional experience of urban parking is equally significant. Drivers report frustration due to unclear regulations, fear of receiving a ticket, and anxiety about towing or overstaying permitted durations. City meters and payment kiosks further introduce friction when they malfunction or require additional steps that interrupt travel. Because availability is unpredictable, drivers often rely on guesswork rather than reliable data. ParkingNow’s city-focused design arises from these conditions, offering a system intended to simplify the search process, clarify regulations, and reduce both the tangible and intangible burdens associated with urban parking.

2.2 University Parking Challenges

While universities exist on a smaller scale than cities, their parking challenges share notable similarities. Campuses such as Hofstra University accommodate thousands of commuters daily, all of whom rely on a finite number of parking lots distributed across academic and residential areas. During peak times, particularly mornings and early afternoons, these lots fill quickly, leaving students to navigate congested aisles or distant overflow zones. Many institutions also rely on outdated systems for permitting, such as physical passes or decals, which can be lost, misplaced, or counterfeited.

Enforcement on campus is similarly inefficient. Officers patrol lots manually, scanning for valid permits or violations. This process consumes staff time and leaves gaps in coverage. Additionally, campuses rarely provide real-time information about space availability, leading to unnecessary congestion as students search for open spots. ParkingNow University Edition responds to these specific problems by integrating digital permit verification, real-time mapping, and navigation tailored to campus infrastructure. By bridging these technological gaps, the system seeks to reduce confusion, improve enforcement accuracy, and support a more equitable distribution of parking resources.

3. Background and Motivation

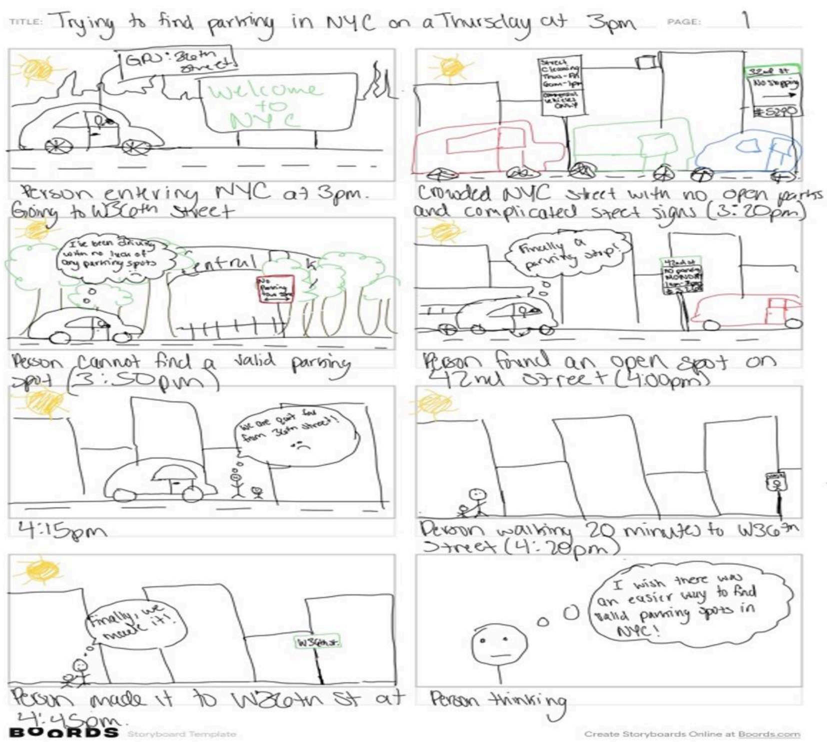
The ParkingNow project originated from an inquiry focused on addressing a real-world mobility challenge. The initial spark emerged when I personally experienced repeated frustration navigating New York City’s constrained parking system. Observations of inconsistent signage, unpredictable availability, and the emotional strain associated with parking motivated the exploration of a technology-driven solution. As research deepened, similar problems were identified on Hofstra University’s campus, where an absence of modern digital tools hindered both students and administrative personnel.

Recognizing these parallels, the ParkingNow project expanded to encompass two interconnected systems: the City Edition and the University Edition. While both share core goals, reducing search time, improving clarity, and modernizing parking processes, their implementations differ based on context-specific needs. This dual-framework approach allowed the project to explore how a unified design philosophy could be adapted to distinct user groups while maintaining consistency in usability, accessibility, and system logic.

The motivation behind ParkingNow extends beyond convenience. It reflects broader human-computer interaction principles including user-centered design, error prevention, accessibility, and cognitive load reduction. By emphasizing iterative prototyping and data-informed decision-making, the project demonstrates how thoughtful system design can support more efficient mobility and reduce user stress. This background informs the development process described throughout the remainder of this thesis.

4. Storyboard and User Journey Analysis

A central component of the project's early design phase involved constructing a storyboard to visualize a typical user experience. The storyboard follows a driver arriving in Manhattan at approximately 3:00 p.m., a notoriously busy time when weekday traffic is at its peak. Upon reaching West 36th Street, the driver initially finds no available parking spaces and becomes increasingly frustrated as confusing signage and congested lanes complicate the search. After twenty minutes of circling, an experience that reflects common parking behavior in crowded districts, the driver ultimately identifies a spot far from the intended destination. Paying the meter and walking the remaining distance further extends the inconvenience, culminating in the driver's arrival nearly an hour behind schedule.



This storyboard highlights several key pain points: inefficiency, emotional frustration, cognitive overload, and the absence of reliable guidance. It also captures the user's desire for a systematic solution, one that could display available spaces, clarify rules, and assist with navigation. These insights informed the decision to develop ParkingNow with a focus on intuitive interface design, real-time updates, and features that reduce the cognitive burden on drivers. By grounding the system in real user experiences, the project remains aligned with human-computer interaction best practices that prioritize user empathy and scenario-based design.

5. Application Overview and Core Features

The ParkingNow system was designed with a strong emphasis on usability, accessibility, and real-time decision support, recognizing that parking decisions are often made under time pressure and in cognitively demanding environments. Both the City Edition and the University Edition share a common design foundation centered on intuitive interaction, clear system feedback, and minimal user effort. At the same time, each version incorporates features tailored to its specific operational context, reflecting differences in user populations, regulatory requirements, and infrastructure. Across both editions, the system prioritizes reducing cognitive load by limiting unnecessary steps, presenting information clearly, and supporting rapid decision-making. These design goals ensure that drivers can interact with the application safely and effectively, even while navigating traffic or unfamiliar surroundings.

5.1 Login and Access

In the City Edition, users encounter a flexible and low-friction login workflow designed to accommodate varying levels of engagement. Returning users may choose to sign in to access saved preferences and payment methods, while first-time or infrequent drivers can proceed using a “continue as guest” option. This approach minimizes barriers to entry and supports inclusivity, ensuring that users can quickly access essential functionality without committing to account creation. Such flexibility is particularly important in urban environments, where drivers may be visiting temporarily or may already be under stress due to traffic conditions.

In contrast, the University Edition restricts access to individuals with verified institutional credentials. Requiring login through a Hofstra University email account establishes a secure and controlled environment, enabling tighter integration with campus databases and administrative systems. This authentication method supports personalized parking verification and ensures that only authorized users interact with campus parking resources. The interface further reinforces familiarity through institutional branding and campus-specific map visuals, reducing cognitive friction and helping users orient themselves within known surroundings. This distinction illustrates how ParkingNow adapts authentication workflows to different contexts while maintaining consistency in overall design principles.

5.2 Real-Time Parking Spot Finder

A central feature of ParkingNow is its real-time parking detection and visualization system, which addresses one of the most significant pain points in both urban and campus parking environments: uncertainty. After granting location permissions, users are presented with a map displaying nearby available parking spaces, along with distance indicators and availability markers that support quick and informed decision-making. By presenting spatial information visually, the system reduces the mental effort required to evaluate multiple options simultaneously.

The real-time nature of this feature distinguishes ParkingNow from traditional parking tools that rely on static signage or rule-based information. Availability data is dynamically updated based on system inputs, allowing users to respond to changing conditions as they occur. For city users, this reduces the frustration and inefficiency associated with circling blocks in search of curbside parking. For university users, it helps alleviate peak-time congestion and reduces the wasted time typically caused by students repeatedly circling high-demand lots, particularly on the east side of campus.

5.3 Navigation and Spot Locking

Once a parking space is selected, ParkingNow initiates GPS-based navigation that guides the user directly to the chosen location. This step-by-step routing minimizes guesswork and reduces the likelihood of missed turns or unnecessary detours. During navigation, the system temporarily "locks" the selected parking spot, signaling to other nearby users that the space is no longer available. While this mechanism does not reserve the physical space, which would conflict with municipal parking regulations, it serves as a coordination tool that reduces competition among users.

By updating availability in real time, the spot-locking feature discourages multiple drivers from converging on the same space, thereby reducing frustration and improving fairness. This coordinated flow of vehicles supports a more balanced distribution of traffic across parking zones and helps prevent localized congestion, particularly in high-demand areas. The feature reflects ParkingNow's broader goal of improving not just individual user experience, but overall system efficiency.

5.4 Payment Systems (City Edition)

In the City Edition, ParkingNow integrates a streamlined and transparent digital payment workflow designed to replace traditional parking meters and kiosks. After arriving at the selected parking space, users enter their license plate information and specify the duration of their stay. The application displays hourly rates, calculates the total cost, and processes payments through widely used methods such as Visa, Mastercard, and Apple Pay. This consolidated workflow reduces transaction friction and eliminates the need for users to interact with physical infrastructure.

To further reduce the risk of citations, the system sends a notification shortly before the parking session expires, allowing users to extend their stay directly through the app. This feature not only improves convenience but also supports compliance with parking regulations by reducing accidental overstays. The payment system reflects ParkingNow's emphasis on user empowerment and aligns with contemporary expectations for seamless, integrated digital services.

5.5 License Plate Verification (University Edition)

The University Edition replaces monetary payment with automated license plate verification, reflecting the permit-based nature of campus parking systems. After parking, users enter their vehicle's license plate number, which is then verified against Hofstra University's permit database. If a valid permit is detected, the parking session is officially registered within the system. This process effectively transforms the application into a mobile parking pass, reducing dependence on physical decals that can be lost, damaged, or counterfeited.

This digital verification system benefits both users and campus safety personnel. Students gain peace of mind knowing that their parking authorization is recorded electronically, while safety officers can verify compliance quickly and accurately without relying on manual inspection. By streamlining enforcement and reducing administrative overhead, ParkingNow supports a modernized campus parking infrastructure that aligns with broader smart-campus initiatives and data-driven management practices.

6. Comparative Analysis: City vs. University Editions

The ParkingNow City and University Editions share a common design foundation rooted in user-centered principles, yet they differ significantly in implementation due to the distinct environments and user populations they serve. The City Edition prioritizes universal accessibility, minimal onboarding requirements, and seamless integration with municipal payment systems. It is designed to support a diverse range of users, including residents, commuters, tourists, and transient drivers who may be unfamiliar with local parking regulations. As a result, the interface emphasizes clarity, simplicity, and immediate usability, allowing users to interact with the system quickly in fast-paced and often stressful urban settings. Features such as guest access, real-time payment processing, and clear pricing transparency are essential to accommodating the unpredictable and heterogeneous nature of city parking behavior.

In contrast, the University Edition operates within a closed and more predictable ecosystem characterized by recurring users, consistent commuting patterns, and institution-specific policies. This version of ParkingNow incorporates digital permit verification, campus-specific navigation, and student-focused interface elements that reflect familiarity with the environment. Because users are authenticated through university credentials, the system can personalize functionality and enforce rules more precisely. The inclusion of digital license plate verification streamlines enforcement processes and reduces reliance on physical permits, directly benefiting campus safety personnel. While the City Edition emphasizes transactional convenience and financial transparency, the University Edition places greater emphasis on access control, administrative efficiency, and operational oversight.

Together, this dual-edition structure demonstrates ParkingNow's adaptability and scalability. It illustrates how a single design philosophy, grounded in human-centered design and usability research, can be flexibly applied to address two distinct sociotechnical environments. By maintaining consistent interaction patterns while tailoring features to contextual needs, ParkingNow achieves coherence across platforms without sacrificing specificity, reinforcing its potential as a scalable mobility management solution.

7. Human-Computer Interaction Framework

The development of ParkingNow draws extensively from human-computer interaction theory, particularly in the areas of cognitive load reduction, system visibility, accessibility, and error prevention. Each iteration of the interface was informed by heuristic evaluation, direct user feedback, and iterative prototyping. These methods ensured that design decisions were grounded in established human-computer interaction principles while remaining responsive to real-world usage conditions. Given that parking often occurs under time pressure and environmental stress, the system was intentionally designed to minimize unnecessary interaction steps and reduce the mental effort required to complete key tasks.

7.1 Accessibility and Audio Integration

A primary accessibility feature of ParkingNow is its audio-guided interface, which supports users during navigation and parking selection. Interacting with a smartphone while driving presents safety concerns, and visual overload can increase cognitive strain. By reading key navigation prompts aloud, the application supports multimodal interaction and allows users to keep their visual attention focused on the road. This design choice aligns with accessibility guidelines and human-computer interaction research emphasizing the importance of alternative interaction modalities. The deliberate decision to disable audio prompts on payment and confirmation screens reflects a context-aware design approach, recognizing that users are safely parked and better able to engage with visual information at that stage of interaction.

7.2 Interface Visibility Improvements

User testing revealed several readability challenges in early prototypes, particularly related to low contrast and insufficient visual hierarchy. To address these issues, outlined input fields were replaced with solid backgrounds, and text colors were standardized to darker, high-contrast tones. These changes improved scannability and made interface elements easier to distinguish at a glance, especially under varying lighting conditions. Improved visibility is a core human-computer interaction principle, as it reduces the effort required to locate information and decreases the likelihood of user error. During usability testing, these visual refinements directly contributed to reduced task completion times and improved user confidence, reinforcing the importance of visual clarity in mobile interface design.

7.3 Button Spacing and Error Reduction

Button spacing (e.g., “Yes” and “No”) emerged as a significant usability concern during testing, particularly in scenarios where users interacted with the app quickly or under stress. Closely spaced buttons increased the likelihood of accidental taps, leading to frustration and corrective actions. In response, spacing between actionable buttons, such as confirmation choices, was increased to better accommodate natural thumb movement and reduce motor-control errors. This adjustment reflects established human-computer interaction error-prevention principles, which prioritize design strategies that prevent mistakes rather than relying on corrective feedback. By reducing the frequency of accidental inputs, the interface fosters greater user trust and smoother interaction flows.

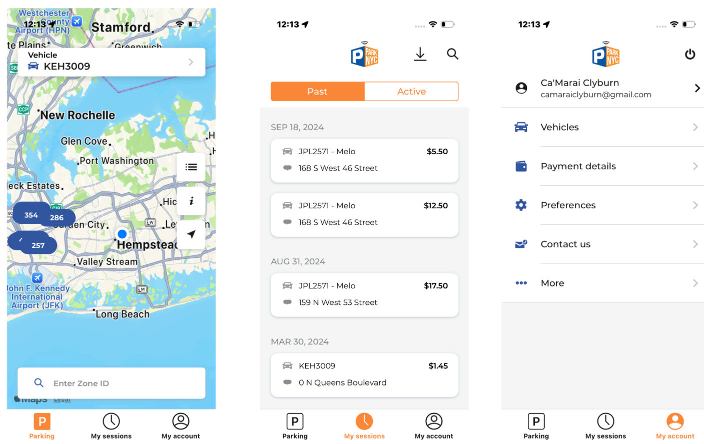
7.4 Color and Readability Enhancements

Additional color and readability enhancements were introduced to further improve usability and guide user decision-making. Button colors were adjusted to increase contrast and signal affordances more clearly, while default map views were shifted from street-level to aerial perspectives to improve spatial understanding. Sliders and interactive controls were refined to provide clearer feedback during use. These seemingly minor adjustments collectively reduce cognitive friction and improve the overall efficiency of the interface. The iterative nature of these refinements demonstrates how incremental design changes, informed by human-computer interaction principles and user feedback, can significantly enhance usability and contribute to a more intuitive and accessible user experience.

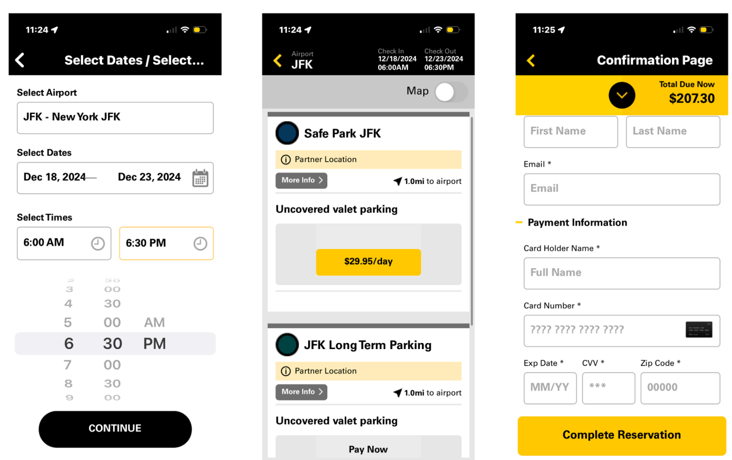
8. Research on Existing Parking Applications

An evaluation of existing parking applications reveals several functional gaps that ParkingNow is specifically designed to address. Many widely used parking apps focus on isolated components of the parking experience rather than providing a comprehensive, end-to-end solution. To identify these gaps, existing applications were analyzed through hands-on use, interface walkthroughs, and comparative feature evaluation. Particular attention was given to how each application supported parking discovery, navigation, payment, and enforcement, as well as the cognitive effort required from users during each stage of interaction. Insights from this analysis directly informed the design decisions implemented in ParkingNow.

The ParkNYC application, for example, is primarily designed to facilitate payment for municipal parking meters. As shown in the screenshots, the interface emphasizes zone-based maps, session management, and account settings, requiring users to manually enter zone identifiers after locating a parking space. While this approach improves transactional efficiency by enabling digital payments and providing access to session history, it does not offer predictive or real-time parking availability information. Through hands-on testing, it became evident that users must still search for open spaces independently and rely on physical signage to confirm zone accuracy. This limitation influenced ParkingNow's emphasis on real-time availability visualization and pre-search assistance, shifting user effort from guesswork to informed decision-making.

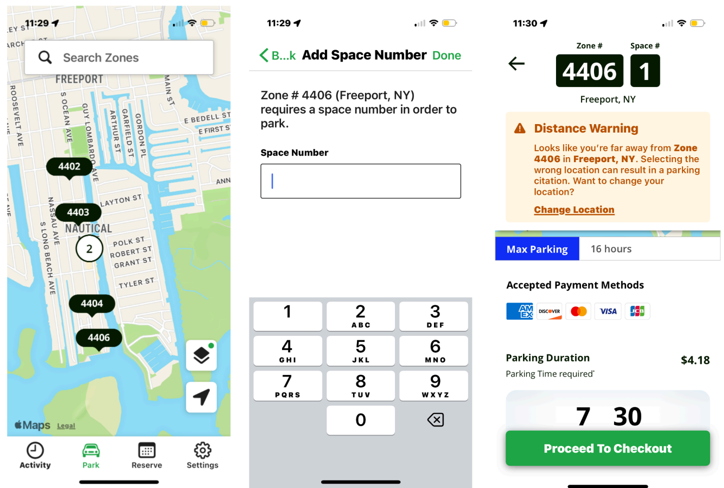


The TheParkingSpot application was examined as a representative example of reservation-based parking systems, particularly those designed for airport environments. The screenshots illustrate a structured workflow in which users select an airport location, define parking dates and times, review pricing calculated on a per-day basis, and complete payment in advance. This model is effective for long-term, planned parking scenarios where duration and location are known



ahead of time. However, hands-on evaluation demonstrated that the application assumes advance planning and fixed destinations, offering no support for spontaneous or short-term parking needs. The absence of real-time availability updates and navigation assistance makes this approach unsuitable for dynamic urban or campus environments, reinforcing ParkingNow's focus on real-time decision-making rather than pre-booked use cases.

The ParkMobile application was also examined for its zone-and-space-number–based payment model. The screenshots show that users must search for the correct zone, manually input a space number, and confirm their physical location to avoid citations. Warning prompts emphasize the consequences of selecting an incorrect zone, highlighting the high level of responsibility placed on the user. While this design supports enforcement accuracy and payment compliance, the analysis revealed that it imposes a significant cognitive burden, particularly when users are under time pressure or navigating unfamiliar areas. These observations informed ParkingNow's decision to minimize manual inputs and reduce reliance on physical signage by integrating guided navigation and automated context awareness into the parking workflow.



Collectively, research on these applications informed ParkingNow's holistic design approach. By identifying where existing solutions succeed and where they fall short, ParkingNow was intentionally designed to integrate real-time availability, guided navigation, and digital payment or verification into a single continuous user workflow. This research-driven process ensured that ParkingNow did not merely replicate existing features, but instead addressed the most critical pain points observed across current parking applications, resulting in a more intuitive, efficient, and user-centered parking experience.

9. Usability Testing and Findings

User testing played a crucial role in refining the ParkingNow interface. To evaluate the app's functionality, I conducted usability tests with three participants: a 16-year-old female, a 24-year-old male, and a 45-year-old female. Participants completed the task of locating a parking spot under simulated real-world conditions, with background music representing common distractions. In this simplified test, participants were not required to operate a vehicle.

The initial results were as follows:

● **Participant 1**: 45 seconds
● **Participant 2**: 42 seconds
● **Participant 3**: 47 seconds

Following task completion, participants completed a survey assessing task difficulty, on-screen text readability, and areas for improvement. The average difficulty rating was 4 out of 5, while readability was rated 3 out of 5.

Based on this feedback, I implemented key design improvements:

● **Improved Visibility**: Changed input fields to solid colors to draw attention and improve usability.
● **Reduced Input Errors**: Increased spacing between "Yes" and "No" buttons to minimize accidental selections.
● **Enhanced Readability**: Standardized dark font colors for all text elements.
● **Updated Map Views**: Switched from street-level views to aerial perspectives for better navigation clarity.

Initial task completion times averaged between 42 and 47 seconds. Following interface revisions, Participants retested the app post-modifications, resulting in 3 to 7-second improvements in task completion times and increased readability ratings. Survey feedback corroborated these improvements. Participants reported reduced difficulty and increased readability after changes were implemented. The testing results empirically validate the iterative design process and illustrate how small refinements can produce meaningful increases in efficiency and user satisfaction.

10. Administrative and Safety Integration

The University Edition's integration with campus administration systems adds significant value to ParkingNow's overall design and functionality. Digital verification of license plates enables campus safety officials to confirm parking authorization quickly and accurately without relying on manual inspection of physical permits. This shift reduces the likelihood of human error, such as misreading decals or overlooking expired permits, while significantly accelerating enforcement workflows. As a result, safety personnel can allocate their time more efficiently, increasing coverage across campus and improving compliance with parking regulations. Ensuring that only authorized vehicles occupy designated parking areas also contributes to a safer campus environment by limiting unauthorized access and reducing congestion in high-demand zones.

In addition to improving enforcement efficiency, the University Edition's digital record-keeping capabilities provide administrators with valuable insights into campus parking behavior. Aggregated data on parking duration, lot utilization, peak demand periods, and recurring violations can be analyzed to inform strategic planning and operational decision-making. These insights support more effective resource allocation, such as adjusting permit distribution, redesignating parking zones, or identifying areas where infrastructure expansion may be necessary. By replacing a traditionally manual and reactive process with a centralized, data-informed system, ParkingNow enables universities to transition toward proactive parking management. This integration aligns with broader smart-campus initiatives and demonstrates how digital tools can enhance both day-to-day operations and long-term mobility planning.

11. Design Evolution

The evolution of the ParkingNow design was guided by a progressive refinement of system functionality, interaction flow, and contextual responsiveness. Rather than focusing solely on visual development, the design process emphasized aligning system behavior with real-world parking scenarios and user expectations. Each phase of development contributed to improving task efficiency, reducing user effort, and strengthening overall system reliability.

As the design matured, increasing attention was given to how users move through the system as a whole. Transitions between stages, such as searching for parking, navigating to a location, and completing confirmation, were refined to ensure continuity and logical sequencing. Design decisions were evaluated based on how effectively they supported goal completion, particularly in time-sensitive situations where users require quick understanding and minimal distraction.

Another key aspect of the design evolution was contextual differentiation. While the core system architecture remained consistent, the interface logic was adapted to address the distinct environments of city-wide parking and campus-based parking. These adaptations were informed by differences in user access requirements, spatial familiarity, and security expectations, resulting in tailored experiences that maintained consistency without sacrificing relevance.

Overall, the design evolution of ParkingNow reflects a shift from conceptual exploration toward purposeful, context-aware solutions. Through continuous evaluation and refinement, the system evolved into a cohesive set of interfaces that prioritize usability, clarity, and confidence while remaining flexible enough to support multiple parking environments.

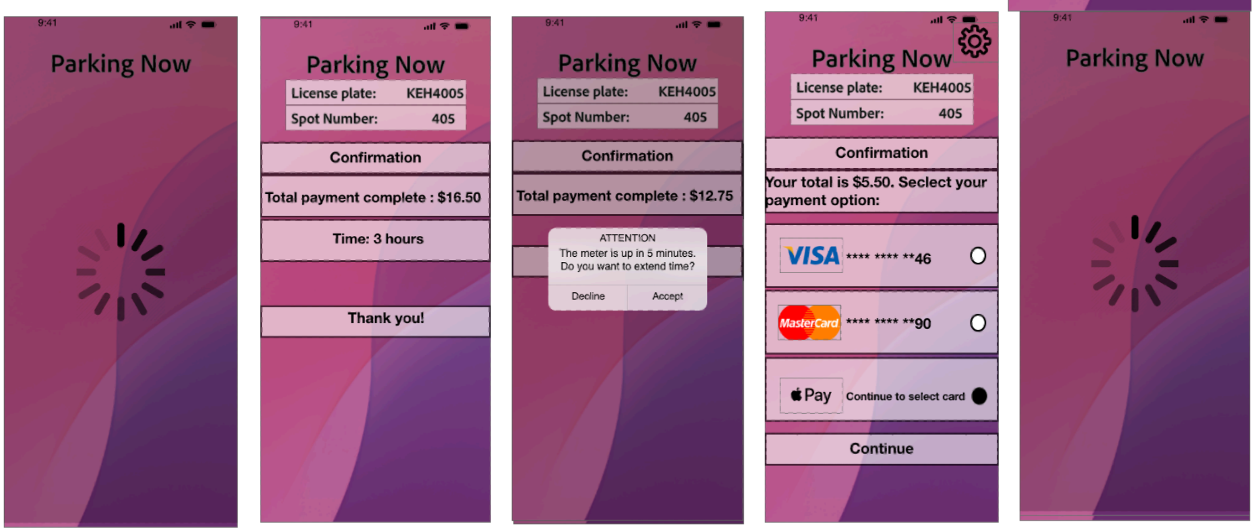
11.1 Prototyping

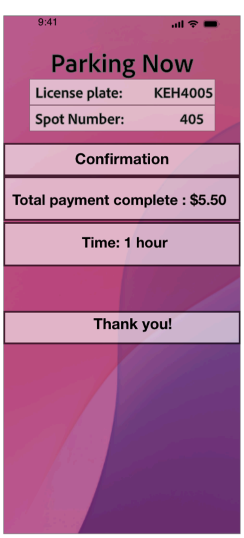
The ParkingNow system underwent several stages of prototyping as it transitioned from conceptual sketches to functional mockups. Early iterations were developed using low-fidelity paper prototypes that allowed the design team to visualize layout structure, examine navigation flows, and receive rapid feedback without committing to time-intensive development. These prototypes played a central role in establishing clarity, determining the spatial organization of elements, and ensuring that interface components followed established human-computer interaction heuristics.

Following paper prototyping, the design evolved into mid-fidelity digital mockups using Balsamiq. These prototypes introduced standardized interface components, simulated user interactions, and helped refine button placement, spacing, and typography decisions. The evolution of the prototypes demonstrated the value of iterative refinement, as each stage revealed new insights into user comprehension, usability, and visual hierarchy. The design process reflects the principles of user-centered design, underscoring the importance of frequent testing, observation, and incremental improvement.

11.2 Parking Now(City Edition) App Interface

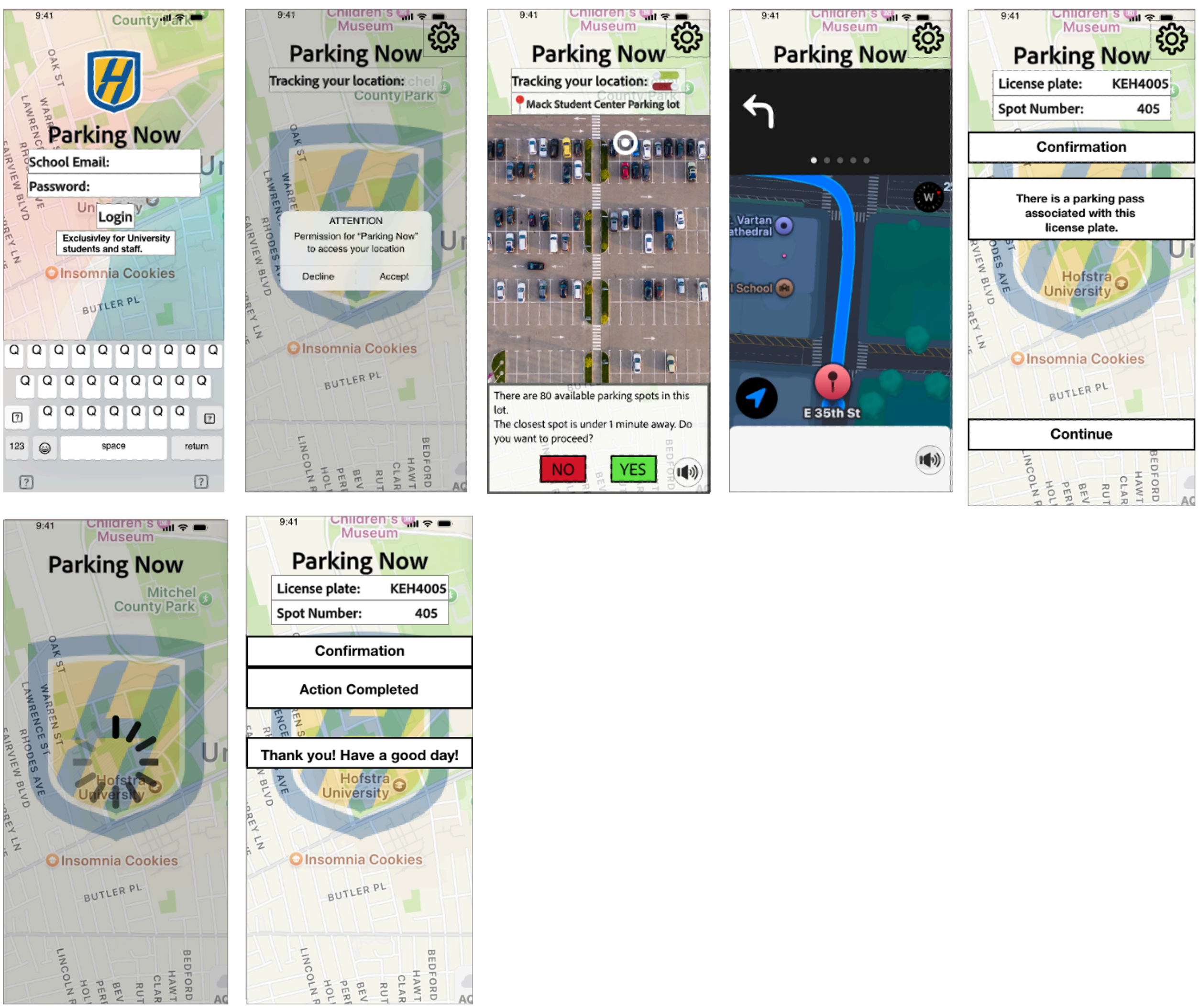
The Parking Now (City Edition) design incorporates a series of clearly defined screens that work together to enhance the overall user experience in a fast-paced urban environment. The login and guest access screens provide a simple and accessible entry point, reducing friction for first-time users. Location-tracking and loading screens offer immediate and direct system feedback by visually indicating progress as the app retrieves location data and available parking information, reinforcing visibility of system status, reducing uncertainty, and building user trust. The structure and layout of these screens were directly influenced by mid-fidelity prototypes created in Balsamiq, which helped define navigation flows, standardize interface components, and refine the placement of key interactive elements. The parking availability screen presents nearby open spots using a map-based view, allowing users to quickly assess options within their surroundings. Turn-by-turn navigation screens guide users directly to the selected parking location, minimizing cognitive load while driving. Finally, the payment and confirmation screens clearly display license plate information, parking duration, total cost, and payment options, ensuring transparency and reducing user error. Together, these screens create a streamlined, intuitive flow that supports efficiency, clarity, and confidence throughout the parking process.





11.3 Parking Now(University Edition) App Interface

The Parking Now (University Edition) design is tailored specifically to support students and staff navigating campus parking environments. Access is limited to university credentials, providing a secure and role-specific login process while remaining simple and efficient. Location tracking and loading screens communicate real-time progress as the system retrieves campus parking data, offering clear feedback that helps users understand system activity and reduces uncertainty. The parking availability screen highlights open spaces within designated campus lots, enabling users to quickly locate nearby parking in familiar surroundings. Navigation screens provide clear, step-by-step guidance through campus roads, helping users reach their selected spot with minimal distraction. Confirmation screens display license plate and assigned spot information, verifying that a valid parking pass is linked to the vehicle and reinforcing user confidence. Overall, the University Edition delivers a focused, reliable experience that emphasizes clarity, security, and ease of use within a campus setting.



12. Technical Considerations: Scalability and System Architecture

Beyond usability, the ParkingNow system required careful consideration of technical scalability and system infrastructure. The long-term vision for ParkingNow involves real-time data acquisition, cloud-supported processing, and integration with municipal or university databases. These requirements necessitate a modular and flexible system capable of supporting large numbers of concurrent users.

For the City Edition, the application must integrate with municipal parking application programming interface(API), payment processors, and dynamically updated databases of open curbside or lot spaces. This necessitates a backend architecture that is resilient, distributed, and capable of managing high-volume traffic without degradation in performance.

For the University Edition, system integration includes secure authentication through university login systems, real-time verification of student or faculty parking permits, and administrative dashboards for campus safety officers. The architecture must therefore support both user-facing and administrator-facing interfaces while ensuring security, data integrity, and compliance with privacy standards. The use of machine learning models may assist in forecasting demand and guiding users to under-utilized areas. An autoencoder neural network was used in the coding aspect for this sector of the application.

Future architectural considerations include implementing predictive analytics that estimate parking availability based on historical trends, time of day, and event scheduling. Other machine learning models may be utilized for testing and prediction periods. The scalability of the system thus hinges not only on infrastructure but also on its capacity for intelligent, adaptive decision-making.

13. Future Enhancements and Opportunities

Several enhancements have been identified to expand ParkingNow's functionality and improve the user experience. One potential feature includes third-party login options such as Google or Apple authentication, which would streamline onboarding and reduce friction for users. Another enhancement involves broadening accessibility support by incorporating audio guidance across all major app interactions, particularly for individuals who may benefit from multimodal instructions.

Administrative improvements include developing a detailed analytics dashboard for parking authorities, which could provide insights into lot utilization, peak-hour congestion, and historical demand trends. This would allow universities or city officials to make informed decisions regarding lot expansion, traffic flow design, or targeted enforcement.

Additional features under consideration include event-specific parking reservations, which would allow administrators to temporarily block or allocate sections of parking lots during major campus events, ensuring orderly traffic flow. Together, these opportunities illustrate how ParkingNow may evolve into a comprehensive mobility management platform capable of supporting both everyday use and large-scale operational planning.

14. Broader Implications for Urban Mobility and Campus Management

ParkingNow's design and functionality reflect broader transformations in urban mobility and campus infrastructure driven by advances in digital technology and smart-system integration. As smart-city initiatives continue to expand, cities are increasingly relying on real-time data, geolocation services, and connected platforms to optimize transportation networks and improve the efficiency of shared spaces. ParkingNow contributes to this evolving ecosystem by demonstrating how data-driven parking management can reduce idle driving, decrease traffic congestion, lower carbon emissions, and enhance overall public safety.

Beyond its impact on urban environments, ParkingNow highlights the growing role of digital solutions in institutional settings such as university campuses. On campuses, parking is not only a logistical challenge but also an administrative and sustainability concern. By replacing physical permits and manual enforcement processes with digital verification and automated data tracking, ParkingNow supports more efficient resource management. This shift reduces paper waste, minimizes the potential for permit fraud, improves enforcement accuracy, and aligns with broader institutional goals related to sustainability and operational transparency.

More importantly, ParkingNow illustrates how everyday challenges can be addressed through thoughtful, research-informed technological interventions. The system does not merely digitize an existing process but reconsiders how users interact with parking infrastructure, emphasizing clarity, predictability, and trust. By integrating system feedback, contextual navigation, and confirmation mechanisms, ParkingNow reduces user frustration and uncertainty while improving compliance and accountability.

In this way, ParkingNow serves as both a functional application and a case study in applied human-computer interaction. It demonstrates how complex socio-technical problems, such as balancing user needs, institutional policies, and environmental considerations, can be mitigated through the integration of user experience design, systems thinking, and scalable software engineering. Ultimately, ParkingNow exemplifies how well-designed digital systems can bridge the gap between human behavior and infrastructural constraints, offering a replicable model for future mobility and campus management solutions.

15. Data Processing and Code

The technical foundation of the ParkingNow (University Edition) anomaly detection system is built on a carefully structured data-processing and modeling pipeline implemented in Python. This pipeline converts raw survey responses into a standardized numerical representation that is suitable for neural network training and subsequent anomaly detection. The design of the pipeline emphasizes reproducibility, transparency, and traceability: every major transformation and decision, from data acquisition and extraction to feature engineering, model training, and evaluation, is captured in code so that experiments can be reproduced and audited. To this end, reproducibility was enforced at the start of every experimental run by fixing global random seeds and configuring deterministic behavior for the machine-learning frameworks in use; for example, the experiment-wide seed was set as RANDOM_SEED = 42 and applied via random.seed(RANDOM_SEED), np.random.seed(RANDOM_SEED), and torch.manual_seed(RANDOM_SEED), ensuring that data splits, model initialization, and stochastic training behavior remain consistent across executions.

15.1 Data Source and Survey Structure

The dataset for this project was collected via a Google Form distributed to Hofstra University students and was intentionally designed to capture the principal behavioral and contextual variables that influence campus parking demand: commuter classification, typical arrival time windows, vehicle type, number of classes taken east of campus, preferred parking zones, and anticipated parking duration. Collectively these fields supply a behavioral fingerprint for each respondent and form the basis for modeling “typical” versus “atypical” parking requests. Because the survey results were exported as a single PDF file named Capstone Survey (Responses) rather than a machine-ready CSV, the raw data required nontrivial preprocessing before modeling could begin. In addition to the extraction challenges inherent to PDFs, privacy considerations were addressed up front: all personally identifiable information (such as names, email addresses, and consent statements) was programmatically removed so that only de-identified behavioral attributes were used in downstream analysis, aligning the pipeline with ethical research practices and institutional data-handling guidelines.

15.2 PDF Extraction and Initial Cleaning

To convert the archived PDF into a manipulable tabular form, tabula-py was employed to parse all pages of the survey output and produce a list of DataFrame objects, which were then concatenated into a single table. The critical extraction step is represented in code by dfs = tabula.read_pdf(PDF_PATH, pages="all", multiple_tables=True, lattice=True) followed by df = pd.concat(dfs, ignore_index=True). After the initial extraction pass, the tabular output commonly contains artifacts such as carriage returns and uneven whitespace; column names were therefore normalized using df.columns = [c.replace("\r", " ").strip() for c in df.columns] to produce consistent header strings. Administrative and privacy-related columns were identified via substring matching and removed automatically to yield a clean, analysis-ready table: df_clean = df.drop(columns=cols_to_drop, errors="ignore"). The cleaned DataFrame was then exported to CSV to produce a stable downstream data artifact that serves as the canonical input to feature engineering and modeling.

15.3 Feature Selection and Dimensional Constraints

Given that the subsequent neural architecture is an autoencoder requiring fixed-length vector inputs, the feature selection stage enforces a strict dimensionality constraint. For implementation simplicity and to limit model complexity in light of the modest sample size, the pipeline selects the first eight columns from the cleaned dataset as the model's feature set; that constraint is codified as INPUT_DIM = 8 and chosen_features = list(df.columns[:INPUT_DIM]). Constraining the input space in this way reduces the risk of overfitting and improves training stability by keeping parameter counts modest relative to the number of available observations. The choice of eight features represents a trade-off: it preserves a diverse mix of numerical and categorical information while enforcing the fixed vector length required for the autoencoder's fully connected layers.

15.4 Numerical Processing Pipeline

Numerical variables, examples include estimates such as the number of classes taken east of campus or the respondent's expected parking duration, were processed using standard, well-documented scikit-learn utilities. Missing numeric values were imputed using the median strategy (num_arr = SimpleImputer(strategy="median").fit_transform(df[numeric_cols])), which maintains the central tendency of each feature without being unduly influenced by extreme outliers. Following imputation, numeric columns were standardized to zero mean and unit variance using num_arr = StandardScaler().fit_transform(num_arr). This standardization step is critical for neural-network training because it brings features onto a common scale, prevents a single feature from dominating gradient updates, and improves numerical conditioning for optimization.

15.5 Categorical Processing Pipeline

Categorical features, including commuter type, arrival window, vehicle classification, and parking preference, were handled via a dedicated categorical pipeline designed to produce compact, deterministic integer encodings suitable for the autoencoder input. Each categorical feature was first coerced to string type and missing values were filled with the explicit placeholder token "__MISSING__" so that the absence of a response is treated as a distinct, model-observable state rather than as NaN. The imputation step is implemented as cat_imputed = SimpleImputer(strategy="constant", fill_value="__MISSING__").fit_transform(df[cat_cols].astype(str)). After this, the pipeline converts categories to integers using ordinal encoding. Ordinal encoding was chosen for its deterministic mapping and compact representation; the encoding step appears in code as cat_arr = OrdinalEncoder(handle_unknown="use_encoded_value", unknown_value=-1).fit_transform(cat_imputed). The encoder is configured so that previously unseen categories encountered during inference are mapped to a reserved value (-1), which prevents runtime failures and allows the model to process new input values gracefully without retraining the encoder.

15.6 Feature Matrix Construction

Once numeric and categorical components were processed independently, they were concatenated horizontally into a single feature matrix that forms the input to the autoencoder. The combined array is assembled with X_full = np.hstack(X_parts).astype(np.float32) and then truncated or zero-padded to ensure exact adherence to the specified input dimension via X = X_full[:, :INPUT_DIM]. This careful trimming or padding enforces strict dimensional consistency across all samples, an absolute requirement for fully connected models, while preserving as much derived information as possible. Because the autoencoder is trained in a self-supervised manner, the target vector is simply a copy of the input and is therefore created via y = X.copy(), making reconstruction the explicit learning objective.

15.7 Training Test Split and Tensor Conversion

For reliable assessment, the prepared feature matrix was partitioned into training and testing subsets using an 80–20 split implemented with X_train, X_test = train_test_split(X, test_size=0.2, random_state=RANDOM_SEED). Both subsets were then converted to PyTorch tensors and moved to the configured compute device to enable GPU acceleration where available, as shown by X_train_t = torch.tensor(X_train, dtype=torch.float32).to(DEVICE). Training data were wrapped into a paired TensorDataset in which each input is paired with itself as the reconstruction target and loaded into a DataLoader for minibatch stochastic optimization (train_ds = TensorDataset(X_train_t, X_train_t) and train_loader = DataLoader(train_ds, batch_size=8, shuffle=True)). This arrangement supports efficient epoch-based training while preserving the self-supervised reconstruction objective.

15.8 Autoencoder Architecture and Training

The neural network consisted of:

- Input layer: 8 features
- Hidden layer 1: 32 units (ReLU activation function)
- Hidden layer 2 / latent space: 16 units

- Decoder hidden layer: 8 units
- Output layer: 8 features

The autoencoder architecture implemented for this study is a compact fully connected network chosen to balance expressive power with the small dataset size. The encoder compresses the eight-dimensional input through a 32-unit hidden layer into a 16-unit latent bottleneck, while the decoder mirrors this compression to reconstruct the original input. The main linear transformations are represented by self.fc1 = nn.Linear(INPUT_DIM, 32), self.fc2 = nn.Linear(32, 16), self.fc3 = nn.Linear(16, 8), and self.out = nn.Linear(8, INPUT_DIM). Nonlinear ReLU activations are applied between layers to enable the network to model complex, non-linear relationships across features. Training was performed using the Adam optimizer and Mean Squared Error (MSE) loss; these choices are codified in criterion = nn.MSELoss() and optimizer = optim.Adam(model.parameters(), lr=1e-3). Dropout layers were included in the model definition for completeness but were assigned zero probability in practice to avoid harming reconstruction fidelity given the dataset size. The training procedure iterates over minibatches, computes reconstruction loss, performs backpropagation, and updates model weights until convergence is reached.

15.9 Anomaly Detection Using Reconstruction Error

After the autoencoder is trained, anomaly detection is performed by computing per-sample reconstruction error on held-out test data. Reconstruction error is computed as the mean squared difference between input vectors and their reconstructions; samples whose error exceeds a high quantile of the training error distribution are considered anomalous. The thresholding step is captured succinctly in code with threshold = np.percentile(train_errors, 95) and anomaly_flags = test_errors > threshold. To evaluate the system's practical utility, the reconstruction-based anomaly scores were compared against a set of rule-based labels derived from logical inconsistencies found in survey responses (for example, respondents who claimed they do not drive but requested all-day parking). Receiver Operating Characteristic and Precision–Recall curves were computed to examine how well statistical novelty aligns with these semantic heuristics, as implemented by fpr, tpr, _ = roc_curve(y_true, scores) and roc_auc = auc(fpr, tpr). The observed ROC AUC of approximately 0.40 denotes limited separability between the autoencoder's continuous anomaly scores and the rule-derived binary labels; rather than indicating failure, this outcome highlights an important conceptual distinction: unsupervised reconstruction-based models detect statistical outliers relative to the learned distribution, while rule-based labels capture semantic or logical inconsistencies that may not be statistically rare in the dataset. This divergence suggests that reconstruction-error detection and rule-based validation are complementary tools for dataset quality assurance and that combining both methods yields a more nuanced anomaly-analysis strategy.

15.10 Heuristic Occupancy Estimation Layer

While reconstruction-based anomaly detection identifies atypical survey responses, the ParkingNow (University Edition) system additionally incorporates a lightweight heuristic layer that translates anomaly signals into a real-time estimate of parking availability. This component bridges the gap between statistical anomaly detection and actionable system output by interpreting detected anomalies as indicators of short-term parking flux rather than as definitive ground-truth events. The heuristic operates entirely downstream of the autoencoder and consumes only the outputs of the anomaly detection stage, ensuring a clean separation between learned representations and decision logic.

In implementation, the heuristic evaluates a sliding window of recent anomaly flags to approximate short-term changes in lot occupancy. Specifically, the system selects the most recent WINDOW_SIZE samples using recent_flags = anomaly_flags[-WINDOW_SIZE:] and computes the number of detected anomalies within that interval via recent_anomalies = int(np.sum(recent_flags)). These anomalies are then partitioned into assumed arrivals and departures using a fixed proportional parameter (ASSUMED_ARRIVAL_RATE = 0.6), producing an estimated net change in occupied spaces defined as net_new_occupied = assumed_arrivals - assumed_departures. This proportional split reflects the assumption that unusual parking behavior is more likely to correspond to incoming vehicles than to departures during periods of elevated campus activity.

The estimated anomaly-driven change is applied to a time-aware baseline occupancy model that reflects expected parking demand throughout the day. The baseline is computed using the system clock (server_hour = datetime.now().hour) and a set of predefined rates that distinguish between daytime peak hours, evening periods, and overnight low-activity windows. For example, daytime operation applies a higher occupancy fraction (dynamic_rate = 0.70), while overnight periods apply a substantially lower rate (dynamic_rate = 0.10). The baseline occupied count is calculated as baseline_occupied = int(round(TOTAL_SPOTS * dynamic_rate)) and then adjusted by the estimated anomaly flux to yield the current occupancy estimate.

In this instance, the heuristic produces an estimated occupancy of 5 out of 50 parking spaces, corresponding to 45 available spots. The Google Colab execution environment reports a server time of approximately 10:43 PM, which the system classifies as an overnight, low-activity period. During this time window, the heuristic assigns a reduced baseline occupancy rate (dynamic_rate = 0.10), reflecting the

```
...
==============================================================================
Server date: January 25, 2026  Current time: 10:43:21 PM (Overnight (Low))
Estimating Open Spots in SIC Parking Lot (Dynamic Baseline Heuristic)
------------------------------------------------------------------------------
Estimated OCCUPIED spots now: 5 / 50
Estimated OPEN   spots now: 45 / 50
==============================================================================
```

expectation that most commuter activity has ceased and campus parking demand is minimal. Consequently, the baseline occupied count is low, and only a small adjustment is applied based on recent anomaly signals. Because the sliding window contains few detected anomalies and the assumed arrival–departure split yields a negligible net change in occupancy, the final estimate remains close to the overnight baseline. This behavior demonstrates that the heuristic responds appropriately to temporal context and avoids overreacting to sparse anomaly signals during low-activity hours.

To ensure physical plausibility and robustness, the final occupancy estimate is explicitly constrained to the known capacity of the parking lot. This constraint is enforced in code by clamping the estimate using est_occupied_now = max(0, min(TOTAL_SPOTS, est_occupied_now)), thereby preventing negative occupancy values or estimates exceeding total available spaces. The estimated number of open spots is then computed as est_open_now = TOTAL_SPOTS - est_occupied_now and presented alongside a human-readable timestamp and time-period label.

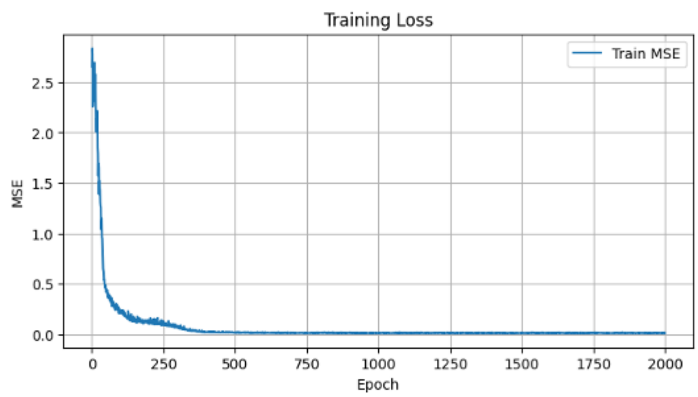
Importantly, this heuristic layer is intentionally conservative and fully interpretable: all parameters, including window size, arrival proportion, baseline rates, and total lot capacity, are explicitly defined and easily adjustable. Although these heuristics do not constitute a predictive model in the traditional machine-learning sense, they provide a pragmatic and transparent mechanism for converting abstract anomaly scores into a user-facing estimate of open parking spaces. Future work may replace or augment this component with real-time sensor data or supervised arrival–departure models; however, in its current form, the heuristic demonstrates how unsupervised anomaly detection can be integrated into a functional decision-support system without sacrificing interpretability or deployment simplicity.

16. Model Training and Performance Visualization

This section presents the visual results produced by the ParkingNow anomaly detection model. The figures illustrate the learning behavior of the autoencoder during training and evaluate its effectiveness in identifying anomalous survey responses when compared against rule-based inconsistency labels. Each subsection evaluates the results, explains whether the observed behavior is expected, and discusses limitations and possible improvements.

16.1 Autoencoder Training Loss Curve

The training loss curve displays the Mean Squared Error between the original input vectors and their reconstructed outputs across 2,000 training epochs. The curve shows a rapid reduction in loss during early epochs, followed by smooth convergence toward near-zero values. This behavior indicates that the autoencoder successfully learned to reconstruct the dominant patterns present in the survey dataset.



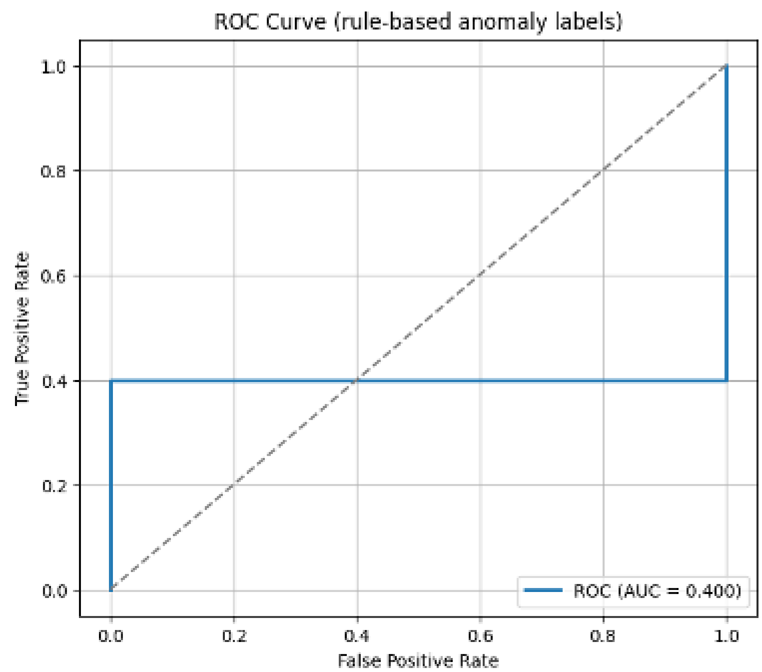
These results are considered good and expected given the structure of the model and the nature of the data. The autoencoder was trained on a relatively small, low-dimensional dataset with only eight input features, many of which exhibit consistent patterns across respondents. Under these conditions, a fully connected neural network with sufficient capacity is expected to achieve low reconstruction error on the training set. The absence of significant noise and the use of standardized numerical features further contribute to stable and efficient learning.

Despite this strong performance, low training loss alone does not guarantee effective anomaly detection on unseen data. The model may learn to reconstruct even unusual patterns if they are present in the training set. To improve robustness, future work could include validation-based early stopping, mild regularization, or training on a larger and more diverse dataset to better separate normal behavior from rare or inconsistent responses.

16.2 Receiver Operating Characteristic (ROC) Curve Analysis

The Receiver Operating Characteristic curve evaluates the model's ability to distinguish anomalous and non-anomalous survey responses across varying reconstruction-error thresholds. The resulting Area Under the Curve of approximately 0.40 indicates weak separability between reconstruction-error scores and rule-based anomaly labels.

This outcome is not unexpected given the fundamental differences between the model's learning objective and the evaluation labels. The autoencoder is trained in an unsupervised manner to minimize reconstruction error across all samples, without knowledge of semantic rules or logical constraints. In contrast, the rule-based labels are derived from human-defined inconsistencies that may not correspond to statistical rarity. As a result, reconstruction error is not expected to align strongly with these labels, leading to limited ROC performance.

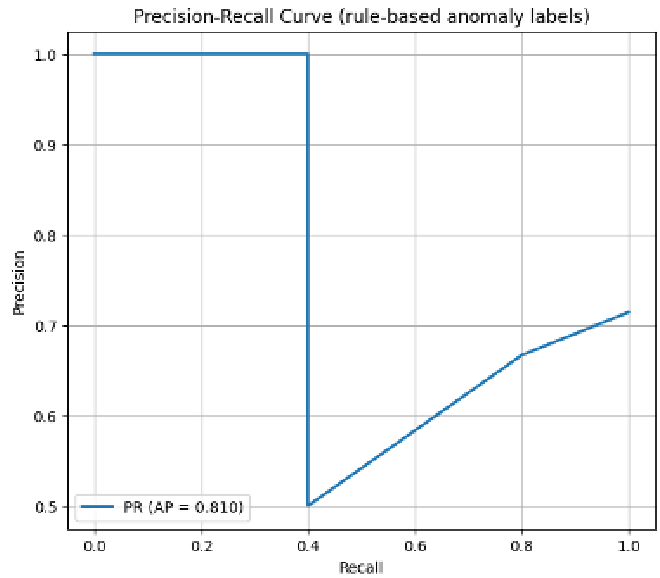


Improving ROC performance would likely require a shift in modeling approach. Incorporating supervised learning signals, constraint-aware feature engineering, or hybrid models that combine learned representations with explicit rules could better align detection scores with semantic definitions of anomalies. Alternatively, ROC analysis could be de-emphasized in favor of evaluation metrics more appropriate for unsupervised anomaly detection.

16.3 Precision–Recall Curve Analysis

The Precision–Recall curve evaluates model performance under class imbalance by measuring the tradeoff between precision and recall. The model achieves an Average Precision score of approximately 0.81, indicating that a large proportion of detected anomalies correspond to rule-based inconsistencies.

This result is both good and expected given the conservative nature of reconstruction-error–based anomaly detection. Autoencoders tend to assign high anomaly scores only to samples that deviate substantially from learned normal patterns. As a result, while not all anomalies are detected, those that are flagged are more likely to represent genuinely unusual responses. This behavior naturally leads to high precision and moderate recall, particularly in datasets with strong class imbalance.



Further improvements could be achieved by adjusting the anomaly threshold to favor higher recall when necessary or by incorporating additional contextual features that increase sensitivity to subtle inconsistencies. Ensemble methods and hybrid detection strategies may also help improve coverage while maintaining the high precision observed in the current results.

17. Conclusion

ParkingNow is a conceptually sound and practically viable system for addressing one of the most persistent challenges in everyday mobility: parking. Through the application of human-centered design principles, iterative prototyping, and usability-driven refinement, the project illustrates how thoughtful interface design can significantly reduce frustration, cognitive load, and inefficiency in high-stress environments. The dual development of City and University Editions highlights the adaptability of a unified design framework across distinct sociotechnical contexts, showing how core usability principles can remain consistent while system functionality is tailored to different user populations and operational constraints.

The usability testing results validate the effectiveness of iterative design, demonstrating measurable improvements in task completion time, readability, and perceived ease of use following targeted interface refinements. These findings reinforce the importance of grounding design decisions in empirical user feedback, particularly when developing systems intended for use under time pressure or distraction. Beyond interface design, the integration of an unsupervised autoencoder model extends ParkingNow's analytical capabilities by introducing a data-driven approach to understanding commuter behavior. Although the machine learning component is exploratory and constrained by limited survey data, its performance aligns with theoretical expectations and demonstrates high precision in identifying meaningful anomalies. This confirms its value as a screening and monitoring tool rather than a standalone decision-maker.

Importantly, the project maintains methodological transparency by acknowledging limitations related to dataset size, heuristic labeling, and the absence of real-time sensor data. Rather than detracting from the contribution, these limitations clarify scope and identify clear pathways for future development. Potential extensions include integrating live parking availability data, expanding behavioral datasets, and incorporating predictive analytics to support proactive parking management. Taken as a whole, ParkingNow serves not only as a functional application prototype but also as a broader case study in applied human-computer interaction. It demonstrates how interdisciplinary approaches that combine design research, usability evaluation, and machine learning can produce scalable, user-centered solutions to complex real-world problems, ultimately contributing to more efficient, equitable, and less stressful mobility experiences.

17. References

Norman, D. A. (2013). The design of everyday things (Revised and expanded ed.). Basic Books.

Nielsen, J. (1994). Usability engineering. Morgan Kaufmann.

Nielsen, J. (1995). 10 usability heuristics for user interface design. Nielsen Norman Group. https://www.nngroup.com/articles/ten-usability-heuristics/

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: Strategies for effective human–computer interaction (6th ed.). Pearson.

ParkNYC. (2024). ParkNYC mobile application. New York City Department of Transportation. https://www.nyc.gov/site/dot/infrastructure/parknyc.page

ParkMobile. (2024). ParkMobile mobile application. ParkMobile, LLC. https://parkmobile.io/

The Parking Spot. (2024). The Parking Spot mobile application. The Parking Spot Holdings, LLC. https://www.theparkingspot.com/

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1–58. https://doi.org/10.1145/1541880.1541882

Aggarwal, C. C. (2017). Outlier analysis (2nd ed.). Springer.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 32, 8024–8035.

United States Department of Transportation. (2023). Smart city and mobility initiatives. https://www.transportation.gov/