Feasibility Report: **UniRide** *a Ride-Sharing Application*

Development Team: Nour El Houda EL IAMANI & Aya BENJELLOUN

October 7, 2025

Part 1: Technical Feasibility Analysis

Technology Stack Assessment

Table 1: Technology Stack Analysis

Category	Tools & Technologies	Skills Available vs Needed	Potential Technical Challenges
Frontend	React.js, JavaScript, HTML5, CSS3	Basic JS/HTML known; React learning needed	React components and state management
Backend	Python, Django, Django REST	Python familiar; Django limited	REST API setup
Database	PostgreSQL	Basic SQL known; PostgreSQL needed	Ride-sharing data design
Auth	Django auth + JWT	JWT knowledge needed	Secure token management
Notifications	Django Channels	Messaging app experience	WebSocket implementation
Maps	Google Maps API	Basic API experience	Map events, API limits
Dev Tools	Git, VS Code, Postman	Version control basics	API testing
Hardware	PCs, internet	Available	None

Feasibility Assessment

Based on the technology stack analysis, the project appears **technically feasible** with the following considerations:

- **Strengths:** Strong Python foundation, basic web development skills from CPI2, and adequate hardware resources

- **Learning Curve:** Some training and familiarization will be needed for React.js, Django, and real-time technologies

- **Risk Areas:** Authentication security, real-time notifications, map integration and route estimation require careful implementation

- **Development Tools and Version Control:** The team is equipped with appropriate code editors (VS Code, PyCharm) and uses Git for version control enabling efficient collaboration and great code management

Steps for Moving Forward

1. Allocate 1-2 weeks for team training on React.js and Django fundamentals

2. Build the app step by step, starting with the most important features first

3. Establish guidelines to ensure UniRide manages Google Maps requests efficiently and avoids exceeding usage limits

4. Make a clear plan to test our login system and backend APIs well

Part 2: Market and User Feasibility Analysis

1. Target Users and Their Needs

Target Users: Campus community including students, faculty, staff, and occasional visitors.
Needs:

- A convenient platform to request or offer rides, especially when campus shuttles are unavailable or inconvenient

- Ability to connect with others traveling off-campus to nearby destinations for shared rides

- Short trips between campus buildings (Restaurant, SaleJadida, Phase3...)

2. Competitive Analysis of Similar Products

Table 2: Comparison of Existing Transportation Solutions

Service	Strengths	Limitations
inDrive App	• Popular ride-hailing platform • Passengers and drivers can directly negotiate fares • Available throughout the entire city of Rabat	• Not designed for small areas like university campuses • Lacks campus-specific features • No optimization for short trips between buildings
Campus Shuttles	• University-provided service • Structured schedules and routes	• Fixed schedules that don't allow much flexibility • Limited number of seats available • Reduced service on weekends, holidays, and late at night
Carpooling Programs	• Reduces traffic • Cost-sharing benefits	• Relies on informal methods (social media, word of mouth) • No dedicated app for management • Difficult to organize and trust without formal system

3. Market Gaps and Opportunities

Table 3: Identified Market Gaps and Potential Solutions

Identified Gap	Opportunity for Our Solution
No single app brings together both ride offers and requests specifically for campus users	Create an integrated platform focused on the specific needs of the campus community
Missing scheduling and waitlist features for busy times like events and holidays	Implement an advanced scheduling system that includes waitlist management
No rating or feedback system to ensure user safety and build trust	Develop a comprehensive rating and feedback mechanism
Need for a flexible and affordable transport option designed specifically for the campus	Offer a tailored service that fits the campus environment and user budgets

4. Survey and User Feedback

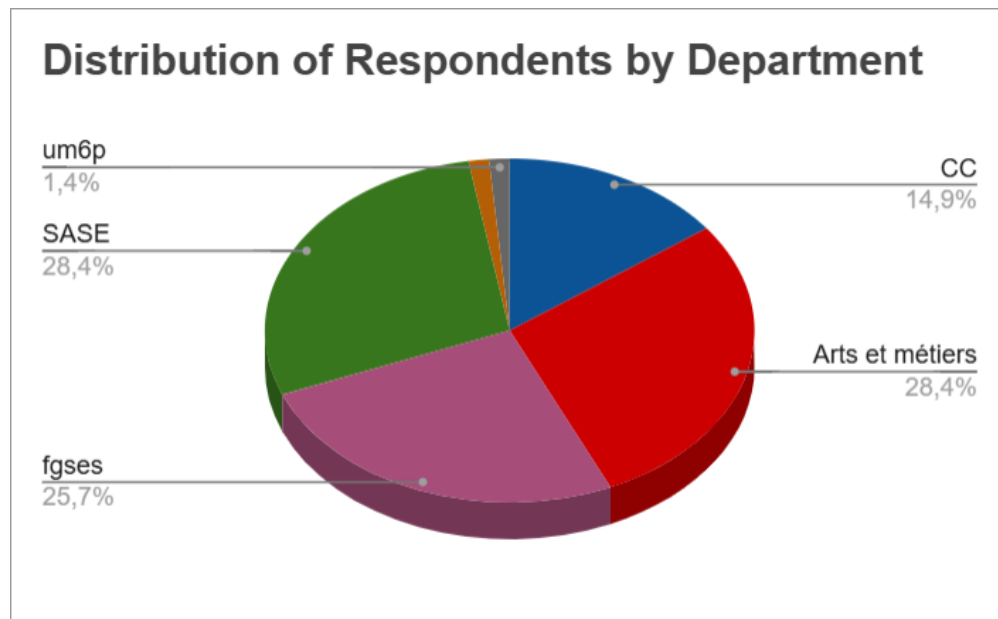**Part 1: Distribution of Respondents by Department**

Figure 1: Distribution of Respondents by Academic Department

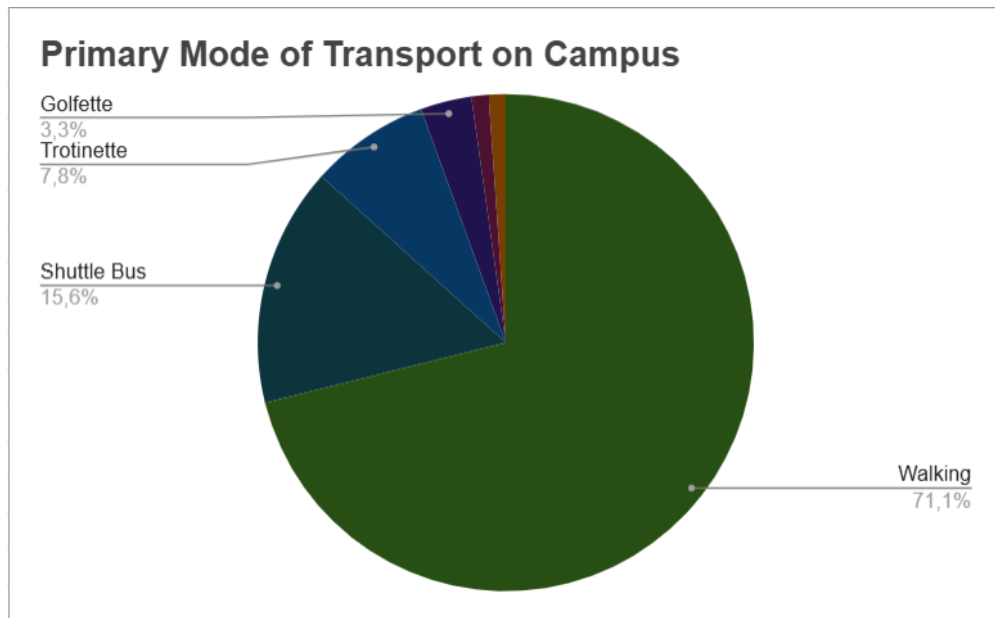**Part 2: Primary Mode of Transport on Campus**

Figure 2: Primary Mode of Transportation Used on Campus

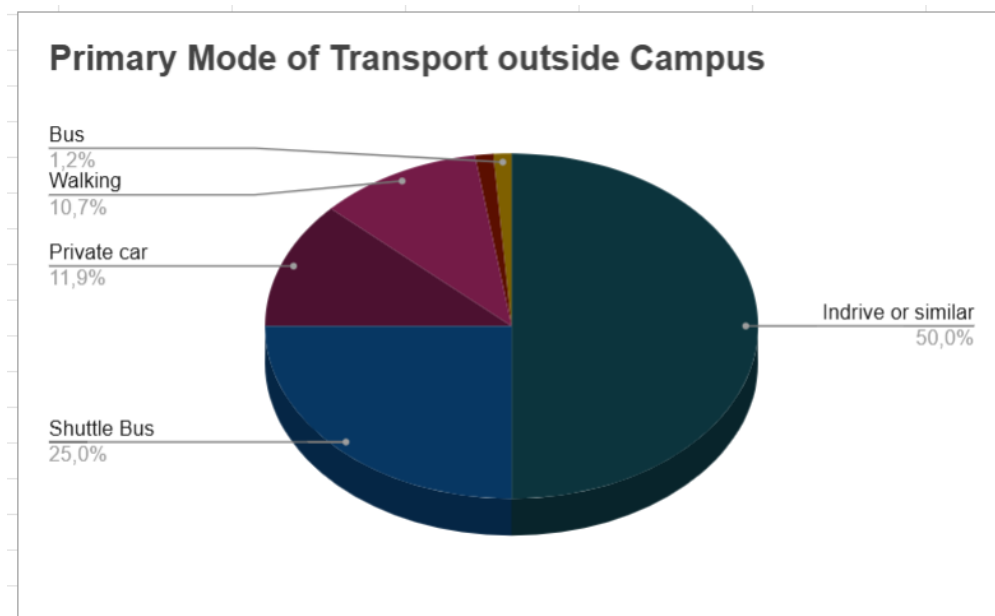**Part 3: Primary Mode of Transport outside Campus**

Figure 3: Primary Mode of Transportation Used outside Campus

Part 4: Travel Frequency on Campus

High Demand Confirmed:

- 77% of users travel frequently (7/10 or higher)

- 54% are very frequent travelers (9-10/10 rating)

- Average frequency: 7.2/10

Part 5: Travel Outside Campus Frequency

Usage Patterns:

- Average frequency: 4.1/10

- Only 19% are frequent travelers (7/10 or higher)

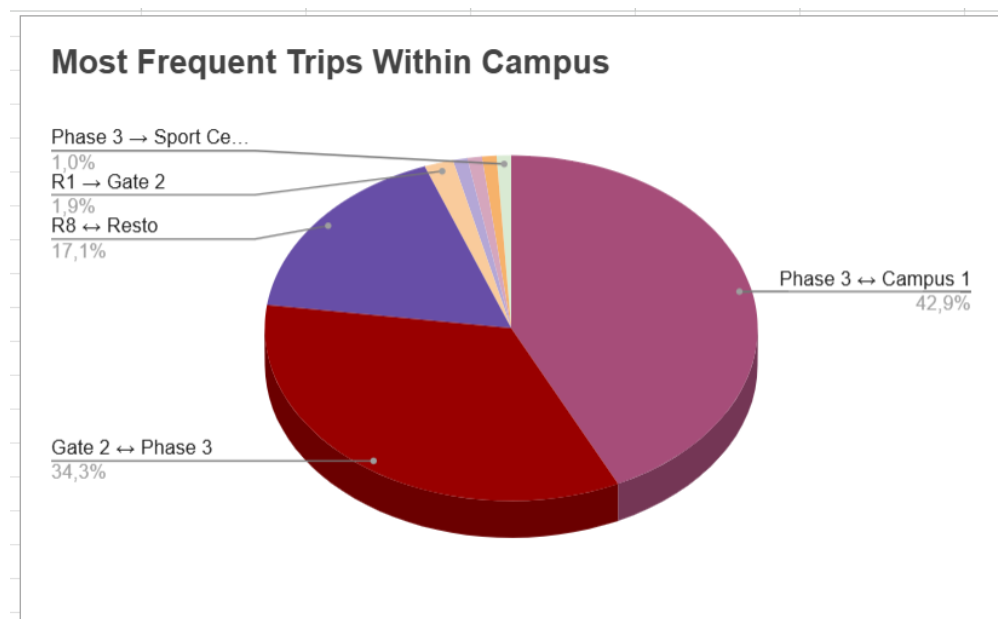**Part 6: Most Frequent Trips Within Campus**

Figure 4: Most Frequent Types of Trips Within Campus

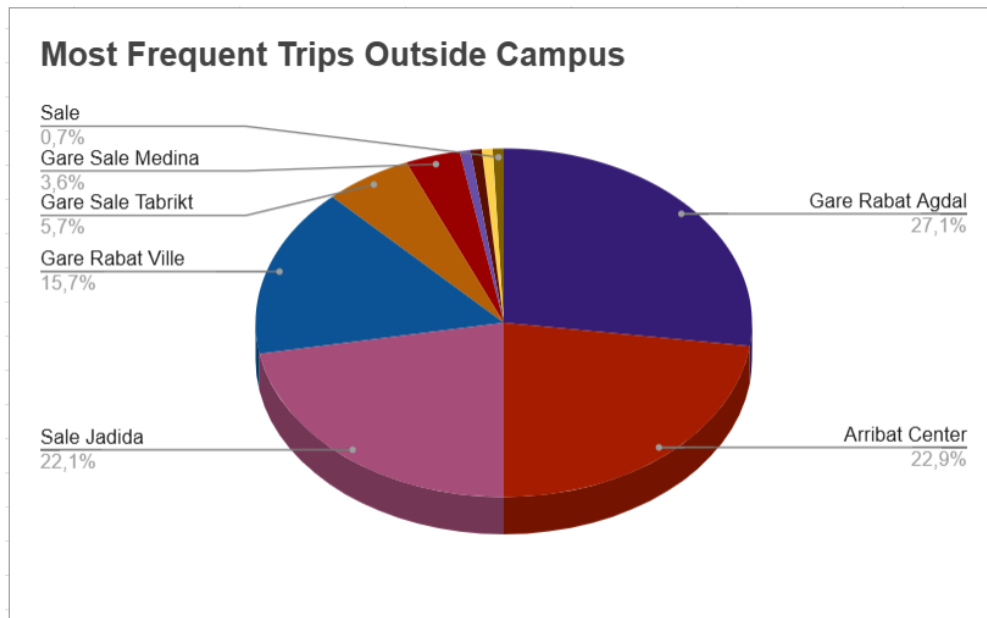**Part 7: Most Frequent Trips Outside Campus**

Figure 5: Most Frequent Types of Trips Outside Campus

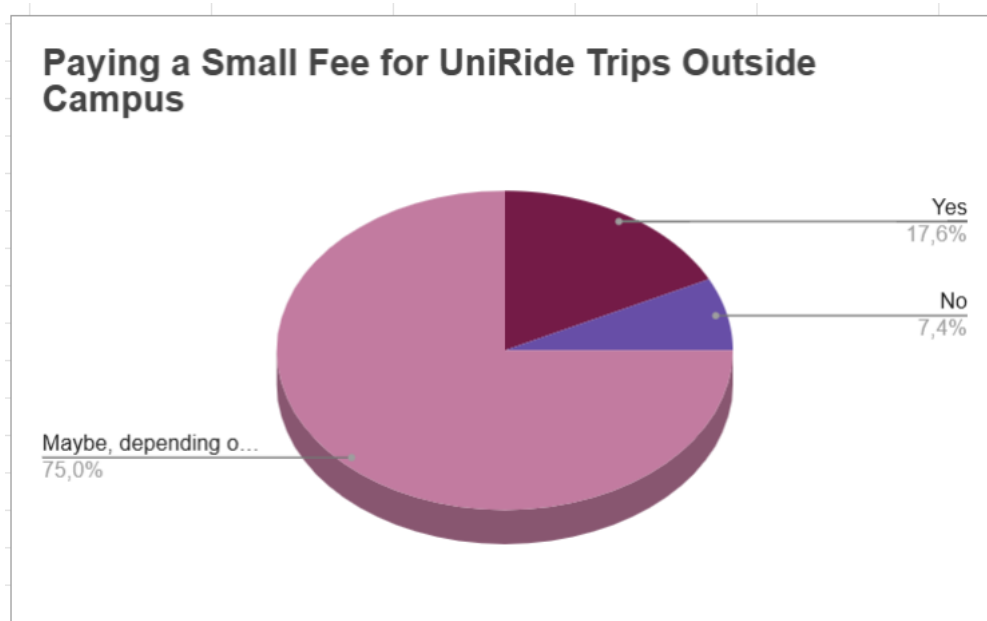**Part 8: Paying a Small Fee for UniRide Trips Outside Campus**

Figure 6: Willingness to Pay Small Fees for Off-Campus Trips

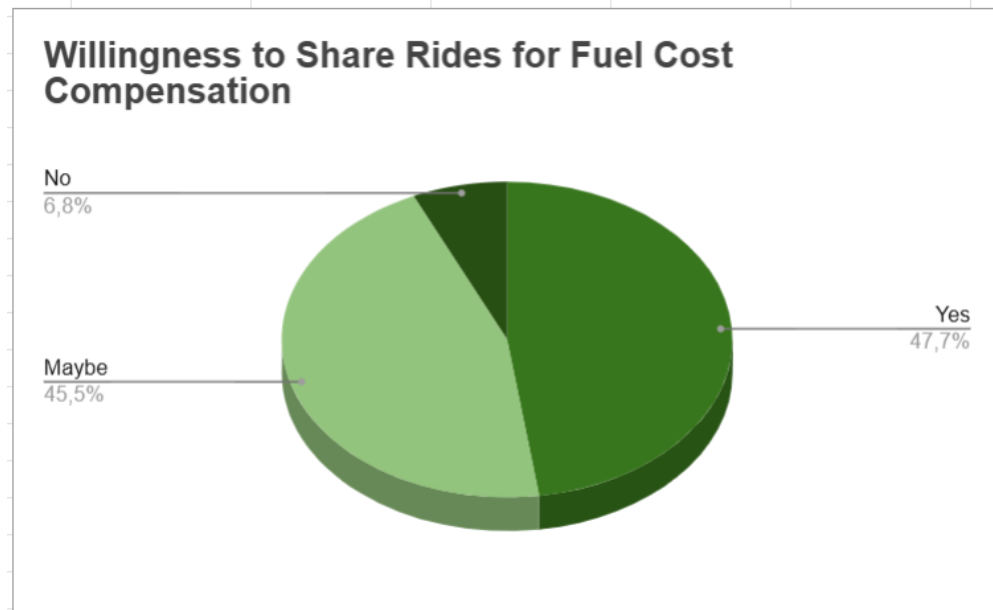**Part 9: Willingness to Share Rides for Fuel Cost Compensation**

Willingness to Share Rides for Fuel Cost Compensation

No
6,8%

Maybe
45,5%

Yes
47,7%

Figure 7: Willingness to Share Rides for Fuel Cost Compensation

Part 10: App Usefulness Rating by Respondents

Key Findings:

- High perceived usefulness - 72.2% rate 8/10 or higher

- Strong positive reception - 36.1% give maximum rating (10/10)

Part 11: Likelihood of Using UniRide

How likely are you to use UniRide if it were available?

- Very high perceived usefulness - 72% rate 8/10 or higher

- Strong maximum rating - 38% give perfect 10/10 score

- Overwhelmingly positive - 85% rate 7/10 or above
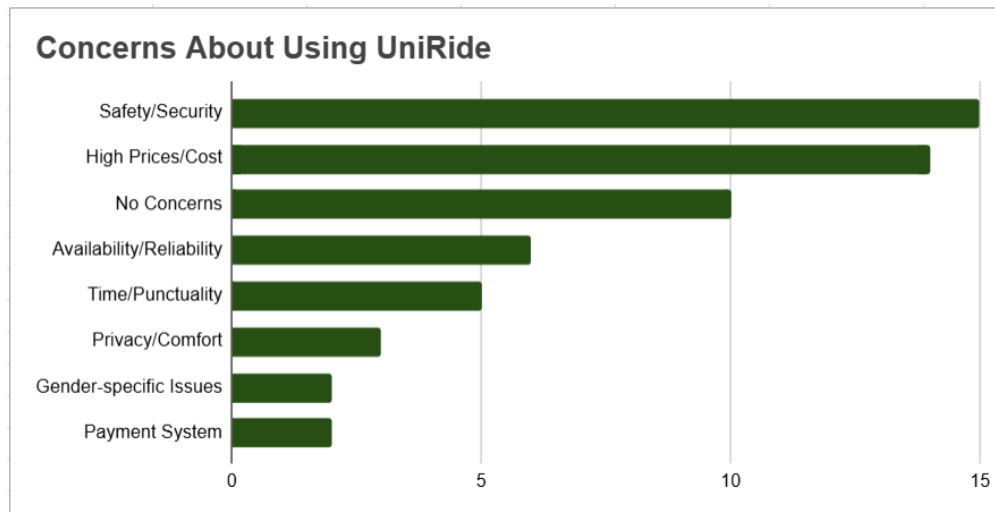
Part 12: Concerns About Using UniRide

Figure 8: Primary User Concerns About Using UniRide

Part 13: Suggestions and Expectations for UniRide

- Implement subscription-based pricing model

- Ensure 24/7 availability with frequent service intervals

- Include gender-specific safety features

- Focus on reliability and punctuality

- Plan for multi-university expansion

Part 3: Schedule Feasibility

Project Timeline and Task Breakdown

To evaluate how practical the UniRide app is over time, the project has been broken down into clear development phases with estimated timelines, creating a realistic and achievable plan for implementation

Table 4: Project Development Timeline

Task	Week 1	Week 2	Week 3	Week 4	Week 5
Requirements Gathering & Analysis	■■■■■■				
Learning Django & React	■■■■■■				
Backend Development (Django)		■■■■■■	■■■■■■		
Frontend Development (React)		■■■■■■	■■■■■■		
API Integration & Map Setup			■■■■■■	■■■■■■	
Notifications & Real-Time Features				■■■■■■	■■■■■■
Testing & Bug Fixing					■■■■■■
Final Deployment & Documentation					■■■■■■

Table 5: Project Task Breakdown

Task Category	Task Description	Estimated Duration
Requirements & Planning	Gather and finalize functional & non-functional requirements	1 week (Week 1)
Learning & Setup	Familiarize with Django, React.js, PostgreSQL, Google Maps API	1 week (Week 1)
Backend Development	• Set up Django project and PostgreSQL database • Design database schema and models (users, rides...) • Implement user authentication with Django's built-in system and JWT • Develop APIs for ride requests/offers, scheduling, waiting lists • Create endpoints for map and route data	2 weeks (Weeks 2 & 3)
Frontend Development	• Set up React.js project • Create user interfaces for login/signup, ride requests/offers, scheduling • Integrate Google Maps API to display routes and location info • Implement state management for user data and ride status • Connect frontend with backend APIs	2 weeks (Weeks 2 & 3)

Note on Timeline   While the project schedule is designed to cover all essential features within the 5-week timeframe, the deadline is somewhat tight. As a result, some additional or advanced functionalities—such as enhanced user profiles, complex ride scheduling options...—may not be fully developed by the initial release. These features can be planned for future updates once the core application is stable and operational.

Conclusion

The project schedule has been carefully broken down into clear, manageable tasks aligned with the available 5-week timeline. By prioritizing core functionalities such as user authentication, ride requests, and route integration early in the development phase, the team can ensure a working prototype within the semester. Overlapping frontend and backend development will optimize progress, while dedicating the final weeks to integration, testing, and deployment will help deliver a stable and functional application. Although the timeline is tight, with focused effort and good time management, completing UniRide within the given period is feasible.