

Persona:

Background:

Most LaundryDash drivers are freelance or part-time gig workers who rely on delivery platforms as a main or supplementary source of income. They typically work flexible hours and handle multiple platforms simultaneously (e.g., food delivery, courier services). Many use motorcycles or small cars to manage multiple pickups and drop-offs efficiently. They are familiar with mobile delivery apps and expect smooth, real-time coordination features to support their day-to-day operations.

Goals and motivation (what they hope to achieve with laundry dash):

Drivers aim to maximize their daily earnings by completing as many jobs as possible while minimizing downtime. They want a stable flow of laundry delivery requests, accurate job information, and optimized routes that save fuel and time. They also seek fair compensation, transparency in job allocation, and clear communication with customers and laundry partners to avoid misunderstandings or delays.

Key tasks:

- Receive and accept pickup or delivery requests.
- Navigate between customer and laundry partner locations using in-app maps.
- Update delivery statuses (e.g., Picked Up, Delivered, Completed).
- Communicate with customers or support for clarifications or issues.
- Monitor daily earnings and completed jobs through the app dashboard.

Behaviours (How comfortable they are with mobile apps, when and how they'll use it):

Drivers are generally comfortable and experienced with mobile apps, often managing several platforms at once. They rely on real-time updates, notifications, and GPS navigation throughout the day. The app is used most actively during morning and evening peak hours when customer demand is highest. Drivers prefer simple, map-based interfaces, quick response times, and minimal manual input while on the move.

Epic and user story:

Epic:

A freelance gig worker who juggles multiple delivery platforms. They are motivated by maximizing earnings through efficiency, requiring a stable flow of jobs, optimized routes, and a clear, reliable app interface. They are experienced mobile app users who need real-time information and minimal distractions while on the road.

User story 1:

As a driver, I want to receive and accept nearby pickup or delivery requests so that I can reduce idle time and earn more.

User story 2:

As a driver, I want to view detailed pickup and drop-off information so that I can prepare for each delivery properly.

User story 3:

As a driver, I want to update job statuses (e.g., Picked Up, Delivered, Completed) so that the system and customers stay informed.

User story 4:

As a driver, I want the app to calculate and display an optimized route for my accepted jobs so that I can optimize my route and complete more jobs in a single trip

User story 5:

As a driver, I want to be able to communicate with the customer or laundry partner via a quick-call feature within the app so that I can resolve issues like finding the location without leaving the app

User story 6:

As a driver, I want to see a real-time dashboard with my earnings for the day, my acceptance rate, and the number of completed jobs so that I can track my performance and motivation

User story 7:

As a driver, I want to set my status to “Online” or “Offline” with a single tap so that I can control when I receive new job requests and take breaks

User story 8:

As a driver, I want to receive system alerts about high-demand "surge" areas or times so that I can strategically position myself to get more jobs.

User story 9:

As a driver, I want to rate my experience with customers or laundry partners so that feedback can improve future operations.

User story 10:

As a driver, I want to view my delivery history and job details so that I can reference past transactions if disputes occur.

Acceptance Criteria:

User story 1: Job Request and Acceptance

Scenario: A driver receives and accepts a nearby job offer

Given I am a driver logged into the app with my status set to "Online"

And I am located within the designated service area

When a new job request appears on my screen

Then I should see the key job details: pickup location, delivery location, estimated total distance, and the estimated payout.

And a 30-second countdown timer for acceptance should be displayed.

When I tap the 'Accept' button before the timer expires

Then the job should be added to my 'Active Jobs' list

And the app should display the detailed order information screen.

User story 3: Real-Time Job Status Updates

Scenario: A driver confirms a successful laundry pickup from a customer

Given I have navigated to the customer's pickup address

And I am viewing the active job's details

When I have collected the laundry from the customer

And I swipe the 'Confirm Pickup' slider in the app

Then the job's status should immediately be updated to 'Picked Up'.

And the customer should receive an automated notification that their laundry has been collected.

And my navigation should automatically update to show the route to the next stop (the laundry partner).

User story 4: Route Planning and Optimization

Scenario: A driver starts their route with multiple accepted jobs

Given I have accepted at least two jobs (e.g., a pickup and a delivery)

And I am viewing my 'Active Jobs' list

When I tap the 'Start Route' button

Then the app should display a map with a single, optimized route connecting all stop locations in the most efficient sequence.

And the map interface should provide turn-by-turn GPS navigation.

And I should be able to see a list of all stops on the route and their order.

User story 5: In-App Driver-Partner Communication

Scenario: A driver needs to call a laundry partner for drop-off instructions

Given I am a driver who has arrived at a laundry partner's location

And I am on the 'Active Job' screen for the drop-off task

When I tap the 'Contact' button for the laundry partner

And I select the 'Call Partner' option

Then the app must initiate a masked phone call, connecting me to the partner's registered phone number.

And my personal phone number must not be revealed to the laundry partner.

And after the call ends, the app should automatically return to the 'Active Job' screen.

User story 6: Driver Performance Dashboard

Scenario: A driver checks their daily performance statistics

Given I am logged into the driver app

And I have completed several jobs during the day

When I navigate to the 'Dashboard' screen

Then I should see a clear summary of my 'Total Earnings Today'.

And the dashboard should display the 'Total Jobs Completed' for the day as a whole number.

And it should also display my 'Job Acceptance Rate' as a percentage for the current work session.

Use case diagram:

https://app.diagrams.net/#G11LpOpBhE4FP-KcKBVpp55jAtrcYyed8T#%7B%22pageId%22%3A%22Kcz_HCmt1B1GLNp9VJgn%22%7D

Use case description:

Use Case	Update job status
Triggering event	The driver wants to update the progress of an assigned job (e.g., Picked Up, Delivered, Completed) to notify the system, customer, and laundry partner in real time.
Brief description	The driver opens the assigned job in the LaundryDash Driver App and selects the appropriate job status. The system validates the driver's location, records the timestamp, and updates the job record in the database. Notifications are then sent to all relevant parties. If the driver is offline, the update is stored locally and synchronized once the connection is restored.

Actors	Driver (primary), Customer, Laundry Partner	
Pre-conditions	1. Driver is logged into the LaundryDash Driver App	
	2. Driver is set to “Online” and has an active, assigned job	
	3. The job status is valid for the next transition (e.g., from “Accepted” to “Picked Up”)	
	4. Internet connection or offline sync capability is available	
Post-conditions	1. Job status is updated in the system database with location and timestamp	
	2. Customer and laundry partner receive real-time status notifications	
	3. Driver dashboard and earnings summary are updated automatically	
Flow of events	1. Opens assigned job in the app.	1.1 Displays job details including current status and available next actions (e.g., “Picked Up”).
	2. Selects the new job status (e.g., “Picked Up”).	2.1 Validates that the driver is near the pickup/drop-off location.
		2.2 Captures timestamp and GPS coordinates.
	3. Confirms the status update.	3.1 Updates the job record in the system database.
		3.2 Notifies the customer and laundry partner of the new status.
		3.3 Refreshes the driver’s dashboard and job list.
	4. Continues with the next task (e.g., navigates to delivery).	4.1 Displays a success message and transitions to the navigation screen if applicable.
Exception conditions	1. Driver too far from location	System shows “You are not at the expected location” and prompts to retry or provide reason for override.
	2. Offline or weak connection	The system saves the update locally and displays “Pending Sync.” Data is auto-synced when connection resumes.

	3. Server error during update	System shows "Unable to update status. Please retry." and logs the attempt.
	4. GPS unavailable	The system requests manual confirmation and marks the record with "No GPS data."