

# Helpers Development Roadmap

## 2025-2026

### Product Roadmap (Next Four Quarters)

#### Q1: Beta Development & Launch (MVP)

**Goal: Launch a functional beta app that enables scheduling, time tracking, communication, and workforce automation.**

- **Customer Scheduling**
  - Customer bookings will be managed through **Acuity Scheduling**.
  - Workforce scheduling will be managed through **When I Work**, allowing companies to set scheduling preferences.
  - Integration between these platforms will be handled using **N8n and Zapier**.
- **Workforce Automation**
  - Assignments will either be automated or allow workforce control over availability based on company preferences.
- **Communication**
  - Initial contact between customers and workers will be initiated through **email and SMS**.
  - No internal chat functionality will be available in the beta.
- **Time Tracking**
  - Workers will manually enter their hours.
  - GPS tracking will be available but not required.
  - No direct invoicing or syncing with payments at this stage.
- **Workforce Management**
  - Workers will be paid **hourly** with a **minimum guaranteed payment** based on time worked.
  - Payment processing will not be live yet.
- **Messaging and Notifications**

- **Real-time text-based messaging** will be enabled through **When I Work**, allowing communication between workforce members and administrators.
- **Testing and Refinements**
  - Feedback will be gathered from initial users to optimize user experience and workflows.

**Milestone:** Beta app is functional with core scheduling, time tracking, and communication features.

## **Q2: Payments and Workforce Communication**

**Goal:** Enable seamless payments and internal communication between the workforce.

- **Payments Integration**
  - Customers will be able to pay through **Stripe**, integrating with Acuity Scheduling.
  - Workers will receive payments based on logged hours in **When I Work**, synced through **Zapier and N8n**.
- **Workforce Communication**
  - **Real-time messaging** will be enabled within the beta for workforce members.
  - Communication will still not be held within the app itself but will be accessible through **When I Work**.
- **Time Tracking and Invoicing**
  - Time tracking will now sync with invoicing and payments to ensure workers are accurately compensated.
- **Testing and Optimization**
  - Payments and messaging functionality will be tested to ensure seamless workflows.

**Milestone:** Payment processing and workforce communication are fully integrated.

## **Q3: Onboarding, Reviews, and Native App Development (Phase 1)**

**Goal:** Improve workforce onboarding and reviews while beginning native app development.

- **Onboarding and Training Features**
  - Workers will complete onboarding through **Typeform**, including quizzes, certificates, and video walkthroughs.
  - Typeform data will connect to workforce profiles via **API integrations**.
  - Worker approval process will be automated upon successful completion of onboarding.
- **Reviews System**

- **Phase 1:** Reviews will be collected through **Google review links**, allowing customers to provide feedback.
- **Phase 2:** A third-party review system will be integrated for streamlined feedback.
- Reviews will be both **public and internal** for performance tracking and workforce improvement.
- **Native App Development Begins (Phase 1)**
  - Development will begin on a **React Native** app, following an **API-first approach**.
  - Initial functionality will focus on **customer and workforce scheduling, onboarding, and communication**.
- **Testing and Refinements**
  - Beta onboarding and review processes will be refined based on early user feedback.

**Milestone:** Onboarding, training, and reviews are fully integrated. Native app development is underway.

## **Q4: Native App Expansion and New Service Offerings**

**Goal:** Launch the native app with expanded service offerings and begin development of the customer-facing app.

- **Native App (Phase 2 - Full Feature Build)**
  - All beta features will be transitioned into the native app.
  - Scheduling, payments, onboarding, and workforce management will be fully integrated.
- **Expand Service Offerings**
  - Additional services will be introduced, including **landscaping, handyman services, painting, and cleaning**.
  - These offerings will be available to both existing and new customers.
- **Customer App Development Begins (Phase 3)**
  - Customers will gain access to an independent app allowing them to:
    - Create profiles
    - Book and schedule jobs
    - Write job descriptions
    - Track workers in real time
    - Participate in the **Helpers Network**

- **Testing and Beta for Customer App**
  - Early access testing will begin for select customers.
  - Iterative updates will be made based on feedback.

**Milestone:** Native app is launched with expanded services. Customer app enters beta testing.

## Beyond Q4 (Future Phases)

- **Q5: Full launch of customer app and mobile booking.**
- **Q6: Helpers Network Expansion** allowing workforce to earn volunteer points for paid work.
- **Q7: Advanced automation features**, such as AI-based scheduling and predictive analytics.
- **Q8: Geographic expansion and scaling efforts.**

## Next Steps and Open Questions

- What specific user testing strategies will be used to gather feedback for each phase?
- How will new service offerings be introduced to ensure quality control?
- What marketing strategies will be used to transition users from the beta to the native app?
- How will worker payment disputes be handled when payments are automated?

# Technical Product Requirements Document

## 1. Overview

This document outlines the technical requirements for the development of a workforce scheduling and management platform. The system will begin as a beta version utilizing third-party tools integrated via APIs and webhooks, transitioning into a fully native application. The core functionalities include customer scheduling, workforce scheduling, payments, time tracking, communication, onboarding, and review management.

## 2. System Architecture

- **Backend:** Node.js
- **Database:** MongoDB
- **Hosting:** AWS
- **API Structure:** GraphQL for efficient queries
- **Authentication:** OAuth 2.0 (Google, Apple ID, Email)

- **Infrastructure Management: Kubernetes**
- **Error Handling & Logging: Sentry**

## **3. Feature Breakdown**

### **3.1 Scheduling System**

**Objective:** Enable customers to schedule services while synchronizing workforce scheduling automatically.

**Third-Party Tools Used:**

- **Acuity Scheduling (Customer-Facing)**
- **When I Work (Workforce-Facing)**
- **Zapier & N8n (Integration & Automation)**

**Data Mapping Between Acuity and When I Work**

<b>Acuity Field</b>	<b>When I Work Field</b>
Date	Date
Time	Time
Crew Size	Assigned Workers
Additional Notes	Job Notes

### **Event Triggers for Syncing**

- **New Appointment:** Triggers a corresponding entry in When I Work
- **Rescheduled Appointment:** Updates existing entry in When I Work
- **Canceled Appointment:** Removes the entry from When I Work
- **Updated Appointment Info:** Sends an SMS notification to assigned workers

### **Technical Considerations**

- **API & Webhooks** will be used for real-time updates.
- **Scheduling Conflicts** are not an issue since multiple workers are available.
- **Zapier** will handle authentication and error handling.

- No queue system is required.

### **3.2 Payment Processing**

**Objective:** Ensure automated, real-time payments based on hours worked.

#### **Implementation**

- **Payment Gateway:** Stripe Connect API
- **Deposit Requirement:** Customers pay a deposit to initiate an appointment
- **Final Payment:** Customers are automatically charged via Stripe after the job is completed based on workers clocking out

#### **Payment Calculation**

- **Payments are determined based on:**
  - Pay rate assigned in When I Work
  - Total hours logged per shift

#### **Dispute & Error Handling**

- **Disputed Payments:** Routed to an admin control panel for review
- **Failed Transactions & Chargebacks:** Trigger an admin notification

### **3.3 Time Tracking**

**Objective:** Accurately track work hours and provide GPS verification.

#### **Implementation**

- **Beta:** Manual time entry with mandatory GPS verification
- **Native App:** GPS-based clock-in/out

#### **Technical Stack**

- **GPS Framework:** Mapbox + Google Location API
- **Time Logging Events:**
  - When a job is created
  - When workers clock in
  - When workers clock out

#### **Technical Considerations**

- Manual time entries do not require admin approval.
- GPS tracking is mandatory for logging time.
- Battery life is not a limiting factor.

### **3.4 Workforce Communication**

**Objective:** Enable real-time communication among workforce members and admins.

#### **Implementation**

- **Third-Party Provider:** Twilio
- **Message Storage:** Messages stored for 90 days
- **Notification Handling:**
  - Unread messages trigger push notifications
  - Group chats are admin-created
  - Direct messages can be initiated by users

### **3.5 Onboarding & Training**

**Objective:** Automate onboarding and activate workers after successful completion.

#### **Implementation**

- **Onboarding Workflow:**
  - Workers complete Typeform-based training (quizzes, certificates, video walkthroughs)
  - Upon completion, the system triggers a background check via Checkr
  - If the background check is cleared, the worker is invited to Stripe Connect
  - Once Stripe registration is complete, the worker is fully activated

#### **Technical Considerations**

- Training data is stored for 24 hours.
- Training must be completed within 24 hours to remain valid.
- Authentication for onboarding and profile management uses OAuth 2.0.

### **3.6 Review System**

**Objective:** Enable customer feedback and track workforce performance.

#### **Implementation**

- **Beta Phase:** Google Review links sent to customers
- **Native App Phase:** Third-party review system integrated

#### **Technical Considerations**

- Reviews are attached to the customer's Google profile.

- Spam and fake reviews will be addressed in later development phases.
- Review data is structured for analytics.
- Workers receive performance scores based on reviews.

### 3.7 Native App Development

**Objective:** Transition from the beta system to a fully functional React Native app.

#### Technical Considerations

- **User Authentication**
  - Users log in once and stay authenticated (similar to Uber).
  - Authentication options: Google, Apple ID, Email.
- **Data Management**
  - Backend database: MongoDB.
  - API query optimization: GraphQL.
- **Notifications & Alerts**
  - Push notifications via Firebase Cloud Messaging.
  - Users can customize notification preferences.

## 4. Final Technical Decisions

Feature	Technology Stack / Service
Backend API	Node.js
Database	MongoDB
Hosting	AWS
API Structure	GraphQL
Authentication	OAuth 2.0 (Google, Apple ID, Email)
Infrastructure Management	Kubernetes
Error Logging	Sentry
Customer Scheduling	Acuity Scheduling
Workforce Scheduling	When I Work



Scheduling Integration	Zapier + N8N
Payments	Stripe Connect API
GPS Tracking	Mapbox + Google Location API
Workforce Messaging	Twilio
Onboarding Automation	Typeform + Checkr API
Push Notifications	Firebase Cloud Messaging

## 5. Development Roadmap

### Q1: Beta Development

- Develop API integrations between Acuity, When I Work, Zapier, and N8n.
- Implement Stripe Connect API for real-time payments.
- Set up Twilio for SMS notifications and workforce messaging.
- Deploy beta system with manual time tracking and basic scheduling.

### Q2: Payments & Workforce Communication

- Automate worker payouts via Stripe Connect.
- Enable real-time workforce communication in When I Work.
- Implement background check automation via Checkr.
- Integrate structured review system.

### Q3: Onboarding, Reviews, and Native App Development

- Finalize structured worker reviews.
- Build React Native app with scheduling and payment automation.
- Implement GraphQL API and MongoDB backend.

### Q4: Native App Expansion & Customer App

- Launch native app with full functionality.
- Introduce expanded service offerings (landscaping, handyman services, painting, cleaning).

- Release customer-facing app with booking, worker tracking, and profile management.

## **6. Next Steps**

- Confirm all API documentation for Acuity, When I Work, Stripe, and Checkr.
- Define database schema for MongoDB to store worker profiles, jobs, and payments.
- Build out a CI/CD pipeline for deployment on AWS and Kubernetes.
- Begin unit and integration testing for each feature module.