PRODUCT REQUIREMENT DOCUMENT

Spare Hand Students

SPARE HAND STUDENTS™ × Applaudo

PRODUCT REQUIREMENT DOCUMENT

Spare Hand Students

Date: 2025, April 25th.
Version: v1.

Spare Hand Students is an innovative two-sided marketplace platform designed to connect households and individuals with reliable local college students for everyday chores and odd jobs. Through a streamlined, app-based experience, the platform promotes time savings, cost efficiency, and community support by empowering students with flexible, paid opportunities.

This community-first initiative offers an accessible solution to managing daily tasks, ranging from cleaning and errands to tutoring and event assistance, while providing students with meaningful work experience and supplemental income. All participating students are verified, ensuring a secure, professional experience for users.

The platform's phased launch began in selected college towns, allowing for validation and service refinement before broader regional expansion. Insights from these initial deployments will guide long-term growth and feature development.

This document outlines the operational and technical framework required to deliver a scalable, secure, and impactful platform that reimagines how communities and students collaborate for mutual benefit.

Change History:

Version	Published Date
v1	April 25th, 2025.

Table of Contents

1. Overview

1.1 Background

The Challenge

Spare Hand Students has successfully validated its model over 10 months of operations, serving 8,000 users and fulfilling over 3,500 jobs across 3 cities in Texas. The early growth confirms strong demand for a student-powered service marketplace, but the system's current limitations are holding back scale, automation, and customer satisfaction.

Today, over **70% of operational tasks**, including **job assignment, communication, and scheduling** are performed manually. Zach, one of the co-founders, currently handles these tasks directly, receiving phone calls from customers, assigning jobs to students (Hands), and resolving issues ad hoc. As volume increases (already 20–25 jobs per day), this model becomes unsustainable.

The platform must now evolve beyond early scrappiness into a **scalable, automated system** that improves **credibility**, supports **older users (65+)**, reduces friction, and frees the team to focus on strategic growth.

At the same time, Spare Hand Students is facing a high-opportunity moment:

- It is positioned to **expand to new universities** this year.
- There's increasing demand for **repeat jobs, upsells**, and **referral-driven growth**.
- Internal stakeholders have clearly prioritized **Product Maturity (Sam)**, **System Automation (Zac)**, and **System Stability & Scalability (Jeremy)**.

To seize this opportunity, the product must:

- Automate job fulfillment flows end-to-end
- Enable customers to self-serve
- Support large-scale regional expansion
- Reinforce trust through in-app protocols and job tracking
- Build scalable admin tools to reduce operational dependency

This initiative marks the transition from an MVP built for validation to a **platform built for scale**.

The Opportunity

Spare Hand Students is uniquely positioned to scale its mission-driven service platform across new regions, deepen customer engagement, and solidify operational excellence with significant near-term upside.

Despite the current operational limitations, the platform has already demonstrated:

- **Product-market fit** in Texas university towns
- A highly engaged customer base (8,000 users, 3,500+ jobs)
- High reliance on manual workflows, creating immediate impact potential through automation
- Consistent inbound volume (20–25 jobs/day) without scaled outbound or automation

The growth potential is clear but must be unlocked through product and platform maturity.

Key Opportunity Areas

1. **University Expansion at Scale**
   - Actively targeting 10+ new universities by end of year, with a long-term roadmap to reach 200+
   - Platform must support seamless onboarding and demand activation across multiple geographies
2. **Increased Revenue per Customer**
   - Upsell and cross-sell potential through smart job suggestions, seasonal services, and multi-Hand bookings
   - Recurring job flows and personalized campaigns can increase booking frequency
3. **Referral-Based Growth**
   - Word-of-mouth and campus-based virality are already contributing to user growth
   - In-app referral mechanics, popups for new university launches, and campaign-triggered credits can amplify this further
4. **Platform-Led Operational Efficiency**
   - 70% of current processes are handled manually by the founding team
   - Replacing manual phone-based job creation with self-serve flows directly improves scalability
   - Admin panel tools will enable operational teams to manage supply, demand, and campaigns without engineering bottlenecks
5. **Trust & Experience for Senior Demographic**
   - A growing segment of senior users (65+) demands a simplified interface, reliable matching, and app-based transparency

- Ensuring trust through job status updates, ratings, and payment protocols will reduce dependency on manual reassurance

This opportunity goes beyond growing usage, it's about transforming Spare Hand Students into a **scalable, resilient platform** that supports community-driven job fulfillment with minimal friction, while enabling leadership to focus on growth and long-term strategy.

The Vision

"Empowering students to grow through work that matters, connecting communities with reliable help while giving students meaningful opportunities to earn, learn, and lead."

1.2 Product Goals

- **Increasing Market Share Through University Expansion**
  Drive the adoption of the platform in new regions and among new users by expanding the presence of Spare Hand Students across more universities.
- **Boost Cross-Sell and Upsell Through Engagement Features**
  Improve retention and revenue per user by promoting new categories, seasonal services, and feature-driven repeat usage.
- **Transition Operational Tasks to the Platform to Support CEO-Level Focus**
  Eliminate Zach’s manual involvement in job scheduling, student outreach, and customer coordination so he can focus on strategic leadership.

1.3 Idea Validation

Qualitative Validation

Objective: Assess user needs, behavioral patterns, and systemic gaps to validate the platform's relevance and desirability.

Methods Used:

1. **Competitor Benchmarking.**
   a. **Key findings:**
      i. Value propositions are distinct, ranging from trust and scheduling (Angi) to speed (Taskrabbit) and flexibility (Thumbtack).
      ii. Digital maturity is even (Thumbtack, Taskrabbit, Angi), with one (Craigslist) lagging more modern app-based model.
      iii. Service models are diverse, direct booking (Taskrabbit), quote-based matching (Thumbtack), curated and managed (Angi), open and unmoderated (Craigslist).

iv. Each platform minimizes liability, placing responsibility for service outcomes largely on the provider and customer.

v. All platforms depend on local supply availability, making service quality and speed vary by location.

Quantitative Validation

Objective: Support the product's feasibility with market data, analytics, and potential impact metrics.

Data-Driven Insights:

a. Student Workforce Supply

According to the National Center for Education Statistics (NCES), there are over 19 million college students in the U.S., with many seeking flexible work opportunities. A 2023 Handshake survey found that 68% of students prefer short-term, flexible jobs over traditional part-time employment.

Implication: The Spare Hand Students model aligns with a growing demand for gig-style work among students, creating a large potential supply of Hands.

b. Initial Market Penetration Metrics

In the first 10 months, Spare Hand Students has onboarded 8,000 users, fulfilled over 3,500 jobs, and is seeing 20–25 jobs per day, primarily in just 3 university cities in Texas.

Implication: With minimal marketing and largely manual workflows, the current adoption confirms product-market fit and validates potential for scalable expansion.

c. Growth Opportunity Through University Expansion

Assuming a phased expansion to 10 new universities with a conservative estimate of 100–150 active users per university, the platform could exceed 10,000+ additional users and a 2–3x increase in job volume within 6–9 months.

Implication: The business can more than double its size through regional expansion alone, validating the investment in scalable architecture and self-service tooling.

d. Upsell and Repeat Engagement Potential

Analysis of returning users shows repeat bookings in similar categories. Cross-sell modules (e.g., "Suggested for You") have been proven to drive 10–30% conversion increases in comparable gig platforms like Thumbtack and TaskRabbit.

Implication: Incorporating upsell features and recurring job flows could significantly increase average jobs per customer, supporting the goal of boosting lifetime value (LTV).

e. Operational Bottlenecks

Currently, over 70% of core workflows are handled manually, including job assignment, payment tracking, and support coordination. Automating these flows is expected to reduce admin intervention by 80–90%, based on projections modeled on similar platforms after backend modernization.
Implication: Automation and admin tool enablement are essential not only for scale, but for freeing leadership from day-to-day ops to focus on CEO-level growth strategy.

2. Target Users

Spare Hand Students will serve multiple user groups, each supported through tailored features and experience-driven solutions that enhance usability, trust, and impact.

As Customers

Primary Users - Household

- **Senior Individuals**
  Older adults living independently requiring help with physically demanding or recurring tasks. They prioritize ease of use, prompt service, and trust in workers.
  The platform will address their needs through senior-friendly interfaces, automated matching, and transparent service tracking.
- **Adult Parents**
  Families managing work and household responsibilities who need dependable, vetted assistance for cleaning, errands, and childcare-related support. They value reliability, speed, and rehire options. The platform will deliver intuitive booking, verified student profiles, and loyalty-based features.

Secondary Users - Business

- **Midlife Professionals**
  Entrepreneurs and small business owners seeking short-term help for operations such as deliveries, administrative tasks, or event support. The platform will serve access to rated and reviewed student workers, and dedicated functionalities for managing recurring needs without long-term commitments.

Tertiary Users

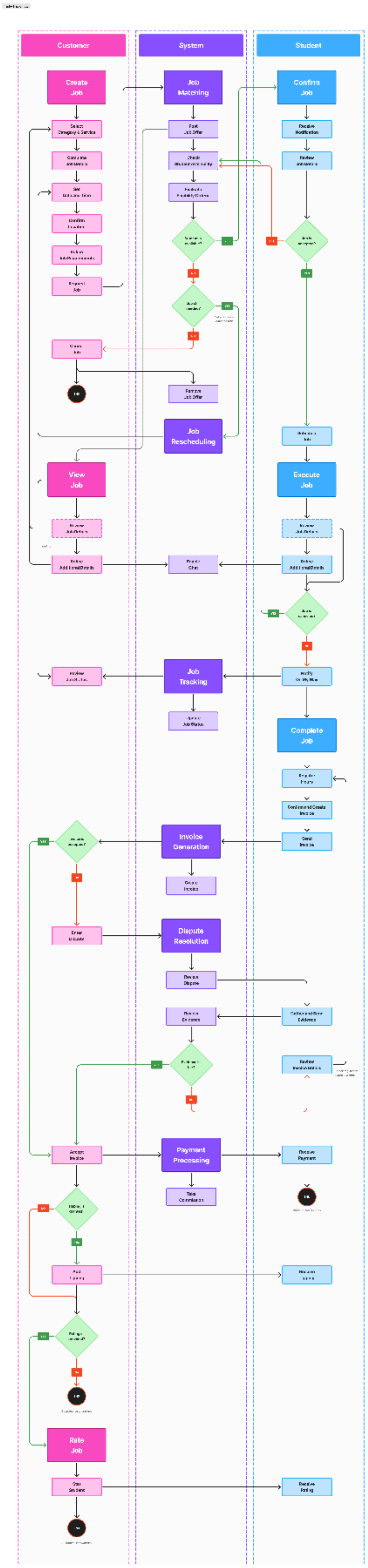
- **Young Students**
  College students require cost-effective services such as tutoring, pet care, and moving assistance. They prioritize affordability, quick access, and seamless coordination. The platform will support them with real-time availability, and in-app communication tools.

As Providers

Primary Users

- **Young Students**
  Local college students offering their time and skills to earn supplemental income. They seek flexibility, fair compensation, and clear job expectations. The platform will offer streamlined onboarding, instant job notifications, and transparent payment systems.

3. High Level User Flow



4. Success Metrics

To ensure the success of Spare-Hand-Students, we have established Objectives and Key Results (OKRs) as a structured framework to align product goals with measurable outcomes. These OKRs provide a clear direction for tracking progress, evaluating impact, and optimizing the product based on real user engagement.

The following section outlines the OKRs for the product, detailing the strategic objectives and the corresponding measurable results to evaluate success.

- **Objective 1: Increase Market Share Through University Expansion**
  - **Key Result 1:** Achieve 25% growth in job volume originating from new university regions
  - **Key Result 2:** Maintain a job fulfillment rate >85% across all active regions

- **Objective 2: Boost Cross-Sell and Upsell Through Engagement Features**
  - **Key Result 1:** Generate 25% of job bookings from promoted categories or "Suggested for You" modules

- **Objective 3: Transition of Operational Tasks to the Platform to Support CEO-Level Focus**
  - **Key Result 1:** Reduce phone-based job scheduling by 90%
  - **Key Result 2:** Automate 90% of job matching and student notifications through platform logic

5. Product Requirements

This section outlines the product requirements for the Spare-Hand-Students Platforms: Mobile Apps (both customer and student), web site and admin panel, detailing its core features, user types, and acceptance criteria. It serves as a guide for development, ensuring alignment with business objectives and user needs.

5.1. Clients Mobile App

E1. Allow Guest Navigation on Home Screen with Smart Sign-Up Trigger for Job Requests

Statement

New users can explore the app as guests, with access to the home screen and key sections, but must create an account to submit a job request.

Acceptance Criteria

1. First Launch

- On first app launch, users can choose between "Sign Up / Log In" and "Continue as Guest"

2. Guest Users Capabilities

- Home screen and job categories
- Suggested jobs or services
- Request a job (allow them to go through the entire flow and ask them to sign up right before submitting)
- Help / FAQ section

3. Guest Users Restrictions

- Submit a job (when a guest tries to create a job, a modal appears with the message: "To request a job, you'll need to create your free account first. It only takes a minute!" After accepting, they're redirected to the sign-up page. Once signed up, the app should return to the point where the guest left off.)
- Access profile and save preferences
- View student availability

E2. Enhance Customer Sign-Up Experience with Phone Verification and Smart Referral Auto-Fill

Statement

Customers can register using their email, phone number (with OTP verification), and password. If they join via a referral link, the referral code field is automatically pre-filled.

Acceptance Criteria

1. Phone Number Input & Verification

- The sign-up form includes a required phone number field with proper input mask and country code.

- After entering a valid number and tapping "Sign Up", the user receives a one-time password (OTP) via SMS.
- The user must enter the code in a verification step to complete the sign-up.
- If the code is incorrect or expired, the user can request a new one.
- Optimized Form Structure:
  - First name
  - Last name
  - Email
  - Password (with real-time validation rules shown below the field)
  - Zip code
  - Phone number (new)
  - Referral code (optional)
  - "Apply" button for referral code becomes active only if the code is inputted and valid.

2. Email Collection (Required, But No Verification at Signup)

- Email is required as a field in the same form
- Email is stored for:
  - Password recovery
  - System communication
  - Security fallback if phone number is lost
  - No email verification flow is triggered during initial sign-up
  - A post-signup prompt may later suggest email confirmation:
  - “Secure your account: confirm your email to make recovery easier.”

3. Referral Code Auto-Fill via Link

- If a customer downloads the app via a referral link (e.g., app.com/signup?ref=ABCD1234), the Referral Code field is pre-filled on the form.
- Tooltip or info icon next to the field says examples of benefits (e.g., “Sam has gifted you $5, enjoy $5 off on your first job with Spare Hand Students”)
- Customer can still edit or remove the referral code manually.

4. Form Submission Logic

- All required fields must be filled and validated before “Sign Up” button is enabled.
- After successful OTP verification, the user is redirected to the app’s home screen or onboarding flow (pending to define)

5. Error Handling

- If OTP verification fails 3 times, the app blocks further attempt for 5 minutes.
- Clear error messages shown for invalid phone numbers, expired OTPs, or referral code issues, login failure or permission denial
- Social Authentication Support
- Customers can choose to sign up using one of the following third-party accounts:
  - Google
  - Facebook

  - Apple (iOS devices only, shown conditionally)
- If social login is selected:
  - The app securely fetches name, email, and profile photo (if permission is granted)
  - User is prompted to verify their phone number via OTP, just like standard sign-up
  - Referral code field is still shown (and auto filled via link if applicable)
  - The account is linked to the social provider for future logins

E3. Job Suggestions Based on Past Activity, Seasonal Trends, and Popular Jobs to Drive Repeat Engagement

Statement

The home screen includes a dynamic “Suggested for You” section that prioritizes relevant past job categories, seasonal services, or popular jobs depending on the user’s activity history.

Acceptance Criteria

1. Dynamic Module Behavior Based on User Type

- Returning user with job history - Show past jobs (1–3) + seasonal & popular jobs (1–2)
- Returning user with no recent activity - Show seasonal & popular jobs (1–4)
- First-time or guest user - Show seasonal & popular jobs (1–4)

2. Personalized Job Suggestions

- Tiles generated based on:
  - Most recently completed jobs
  - Most frequently requested job types
- Excludes:
  - Expired, canceled, or disputed jobs
- Each tile includes:
  - Category name (e.g., “Yard Work”)
  - Icon or image
  - Subtext like “You’ve requested this 3 times”
  - Tapping a tile opens the job creation form with the category pre-selected

3. Seasonal Jobs Integration

- Jobs like “Yard Work,” “Holiday Help,” “Moving Help” are tagged as seasonal.
- Appear automatically based on date-based rules (configurable via admin).
- Visual cues: Badge: “In Season” Subtext: “Popular this time of year”.

4. Popular Jobs Fallback

- If the user has no job history, the system displays top-performing job types across the platform. Example categories: Yard Work, Moving Help, Pet Sitting, Cleaning.
- These tiles include static copy as “Most requested by your neighbors”

5. Interaction Behavior

- Tapping any tile opens the job request form with that category pre-selected. If the customer has previously completed a job in that category: Prompt appears: “Do you want to reuse details from your last [category] job?”

6. Placement and UI Guidelines

- Appears in the top half of the home screen
- Section title dynamically adapts:
  - “Suggested for You” (when personalized)
  - “Popular Jobs” (if no history, if first-time user)

E4. Simplify Home Screen with Main Job Categories and Voice-Enabled Search for Accessibility

Statement

The home screen must allow senior users to quickly start the job creation process using either large, touch-friendly job category buttons or an accessible voice-activated search interface.

Acceptance Criteria

1. Category Buttons on Home Screen

- The home screen displays 4–6 prominent job category button
- Each button must include:
  - A clear, friendly icon (e.g., shovel, moving box, pawprint)
  - A large, legible label with text no smaller than 16pt
  - High-contrast colors to meet accessibility standards
  - Buttons are placed in a grid or list layout above the fold (no scrolling needed to see at least 4)

2. Category Tap Behavior

- Tapping a job category button opens the job creation form with the category pre-selected.
- If the customer has used this category before: the system prompts: “Would you like to use your last [category] job details?” Options: Yes / No
- If the customer is new or selects “No,” a blank job form opens with only the category pre-filled.

3. Text Search as Fallback

- Users may still type into the search bar
- Autocomplete suggestions are offered as they type (based on category database)

4. Admin Configuration Options

- Admins can configure:
  - Which job categories appear as homepage tiles
  - Order of display
  - Tooltips or short descriptors for each tile
- Enable/disable voice input toggle (per platform or region)

E5. Display Dynamic Job Forms Based on Job Type Selection

Statement

Users should only see fields that are specific and necessary for the selected job category to reduce friction and improve clarity during job creation.

Acceptance Criteria

1. Job Type and Field Mapping

- Each job category (e.g., Pet Sitting, Yard Work, Moving) is linked to a configurable set of fields based on business rules.
  Example:
  Pet Sitting → Number of pets, duration, tools needed
  Yard Work → Yard size, tools provided, frequency
- Field sets are maintained via admin panel, allowing dynamic updates without code deployment.

2. Dynamic Field Display Based on Job Type

- When the user selects a job category:
  - The form immediately updates to show only the fields relevant to that job type.
  - Unrelated fields are hidden.
  - Section headers (e.g., “Job Details”, “Tools Required”) adjust to reflect relevant content.
- If the user changes the category mid-way, the form resets and:
  - Displays the new relevant field set
  - Clears any previously entered values that do not apply to the new category

3. Address Input Options

- Address section gives users two clear options:
  - “Use my saved address” (autofills from profile information)
  - “Enter a new address” (expands editable fields for: Street, Apt/Suite, City, Zip code)
- Once a selection is made:
  - The choice persists through the session, including navigation back or forth
  - Autofilled fields are editable if the user wishes to modify them before submitting

4. Form Field Validation

- All displayed fields must retain their specific validations:
  - Required vs optional
  - Format validation (e.g., zip code, numeric ranges, text limits)
- Validation must occur:
  - On input (real-time, where possible)
  - On form submission
- Error messages must be:
  - Clear and visible next to the invalid field
  - Written in accessible language suitable for all users (including seniors)

5. Persistence During Session

- Once the job type is selected and fields are filled:
  - The app retains the user's progress during the session
  - Navigating away and returning within the same session does not reset the form
  - If the app is closed or timed out, the form data is cleared

6. Fallback and Error Handling

- If dynamic form logic fails to load
  - Show a fallback message: "We're having trouble loading the form fields for this job. Please try again or select another category."
  - Prevent form submission if mandatory fields tied to the job type are missing due to a config error

7. Auto-Save Incomplete Job Requests as Drafts

- If a customer begins filling out a job form but navigates away from the app (e.g., minimizes, closes, or switches apps), the form is automatically saved as a draft.
- When the customer reopens the app:
  - A popup/modal appears with the message: "You have a job request in progress. Would you like to continue editing it?"
  - Options:
    - Continue Editing → reopens saved form with all previous inputs
    - Start New Job → discards draft and opens blank form
- Drafts persist in local storage for at least 24 hours.
- Fields saved in the draft include:
  - Selected job type
  - All completed inputs (address, tools, notes, etc.)
  - Partially filled entries (e.g., notes or tool selection)
- System clears the draft automatically if:
  - The job is successfully submitted
  - The customer explicitly chooses "Start New Job"

E6. Optimize Real-Time Job Fulfillment for Multi-Hand Requests with Smart Reassignment and Scheduling Flexibility

Statement

When a customer requests a job, the system attempts to assign the required number of Hands in real time. If the request is not fully filled, fallback options are presented in-app, including continuing with fewer Hands or switching to a nearby time slot that fits the confirmed Hands' schedules.

Acceptance Criteria

1. Assignment Logic

- The system attempts to assign exactly the number of Hands requested:
  - Uses real-time availability, proximity, and job-type match filters
  - Batches of students are notified until slots are filled

2. Bulk Notification and Retry Strategy

- When a job is submitted:
    - The platform sends push notifications to a batch of eligible Hands simultaneously
    - If not all slots are filled within X minutes (e.g., 10), system retries with a new batch
    - Notifications continue every retry cycle until: All slots are filled, job hits its expiration threshold, customer accepts fallback option

3. Partial Assignment Fallback UX

- If only some Hands accept (e.g., 4 of 10) and job hits its expiration threshold:
    - The app presents a modal: "4 of 10 Hands have accepted your request. What would you like to do?"
        - Proceed with 4 Hands
        - Switch day to [next mutual availability]
    - If the customer selects Switch day:
        - The app shows the next available day/time that fits all accepted Hands' calendars
        - The job is automatically updated to that time Hands are notified again and new eligible
        - Hands are invited to fill remaining slots
        - The date-switch option must remain lightweight, with limited suggestions (ideally one best-fit time), to avoid overcomplicating the decision on mobile and web

4. Smart Reschedule Algorithm (Backend)

- A scheduling algorithm checks:
    - Duration estimated by the customer (e.g., "about 2 hours")
    - Confirmed Hands' real availability
    - Suggests nearest mutual open slot where Hands are still available
    - Uses schedule data pulled from student app (synced in real time or within 1 min)

5. Customer Communication and Feedback Loop

- The customer is always kept informed:
    - Live tracking UI: "Confirming Hands… 2 of 3 accepted"
    - System notifies when retry or fallback actions are triggered
    - If customer does nothing by X time (e.g., 3 hrs), auto-message: "We're still finalizing your request. Would you like to adjust the time or try again later?"

6. Confirmation and Lock-In

- When all Hands are assigned:
    - Customer sees a confirmation screen showing: All assigned Hands, Time slot, Job details
    - Proceed to tracking interface

7. Admin/Support Sync

- Admins can see:
  - Which jobs are partially filled
  - Which fallback path the customer chose
  - Any system-generated time shifts or escalations

E7. Display Real-Time Job Assignment and Status Updates in a Visual Tracking Interface

Statement

After creating a job, customers are shown a real-time, visual interface that clearly reflects the status of the job assignment and progress, from searching, to assignment, to completion, with intuitive updates, icons, and timestamps.

Acceptance Criteria

1. Initial Assignment Status: Searching for a Hand

- Immediately after submitting the job, the customer is taken to a tracking screen that shows:
  - A message: “Looking for an available Hand…”
  - A visual indicator (e.g., animated search radar, loader, or pulse effect)
  - Estimated time range for assignment (e.g., “This usually takes less than 30 minutes”)
  - A fallback message is displayed after 5 minutes if no one has accepted: “We’re still searching for available Hands nearby. Hang tight!”

2. Assignment Confirmation

- When a student accepts the job, the screen automatically updates to show:
  - Student (Hand) name
  - Profile photo (if available)
  - Star rating (if applicable)
  - A label: “[Hand Name] has accepted your job!”
  - A status marker updates to: "Hand Assigned"
  - Timestamp is recorded and displayed under the label
- User can access from this point to the students profile

3. Real-Time Job Status Stages

- Each job must pass through and display the stages, with distinct icons, titles, and live timestamps
- Icons must be visually distinct and easy to interpret for seniors
- Timestamp format: “Today at 3:45pm”

4. Persistent Access

- The user can re-access the tracking screen at any time by:
  - Tapping a notification (e.g., push or SMS link - pending to confirm)
  - Navigating via Home > My Jobs > [Job In Progress]
- When re-accessed, the screen shows the latest known status without delay

5. Real-Time Sync with Backend

- The tracking interface must:
  - Pull live updates from the backend automatically
  - Update status markers, timestamps, and student actions without requiring the customer to refresh or leave the screen

6. Edge Case Handling

- If no student accepts the job after a threshold (e.g., 2.5 hours), the screen should show:
  - “We haven’t found a Hand yet. Would you like to try again or reschedule?”
  - Offer actionable buttons: Reschedule / Cancel Job / Contact Support
- If the student cancels mid-way:
  - Update the status to “Your Hand has canceled. We’re finding another match...”
  - Revert to “Searching” stage if time window still applies

E8. Enhance "My Hands" Section to Let Customers Rehire Previously Rated Students (with Availability Awareness)

Statement

Customers can view and rehire students they’ve rated 4–5 stars through a next-availability-driven flow. The system displays each Hand’s upcoming open time, allows job duplication, and handles fallbacks if the Hand becomes unavailable with easy options to exit or expand the job.

Acceptance Criteria

1. Display & Sorting of Hands

- “My Hands” section shows all students the customer has rated 4–5 stars.
- Hands are sorted by their next available time, not current online status:
  - First: Hands with upcoming availability within 7 days
  - Then: Hands with availability beyond 7 days
  - Hidden (optional): Hands with no availability set (inactive or not scheduled)
- For each Hand, display:
  - Name and photo
  - Next available time block (e.g., “Next available: Monday at 2 PM”)
  - Option: View full schedule (opens calendar viewer)

2. Availability Logic & Syncing

- Availability is:
  - Pulled from the student availability definition for the semester or break
  - Synced in real-time or within max 1-minute intervals
- “Availability for Job Requests” toggle must be ON for the student to appear
- Hands without set availability are hidden (or marked inactive, per admin config)

3. Calendar Selection & Time Slot Booking

- Tapping on a Hand opens a calendar view of their availability
- Customer selects a day/time from available blocks
- That time is locked for job request creation

- If another customer books it before submission, fallback logic is triggered

4. Job Creation Flow

- After time selection, the customer chooses the job category.
- If the same category was used before with that Hand, a prompt shows:
    - “Would you like to repeat your last [category] job with [Hand Name]?”
    - Yes → form pre-fills with:
        - Job type
        - Number of Hands
        - Tool availability
        - Notes
        - Address
    - No → opens a blank form
        - All fields are editable. Nothing is auto-submitted.

5. Confirmation & Final Screen

- The confirmation screen shows:
    - Hand name and photo
    - Scheduled time
    - Job category
    - Summary of job details and charges
    - A message is shown: "If the Hand you are selecting is not available we will find an available Hand to help out" the customer can select wheather or not he is ok with this option
    - “Confirm and Send” button

6. Fallback & Reassignment Options

- If the Hand becomes unavailable before submission or declines the job:
    - The system reassigns the job automatically and a message is shown to the client: "{Selected Hand} is not available, we are in the process of finding an available Hand to help out on {Date}"
    - In case the customer didn't accept the automatic reassignment, then a message is shown: "Zach is not available, please click to try another time" and it will prompt them to try selected Hand or another one of their My Hands.

7. Additional Options

- “Find a Hand for Me” button:
    - Instantly sends customer into auto-match flow
- “Add More Hands”:
    - Allows selection of more students from “My Hands”
    - Or lets system auto-fill remaining slots based on job type
    - Visible at all stages of the rehire flow

8. Error Handling & Sync Conflicts

- If the selected time becomes unavailable after selection:
    - Show alternate time suggestions or prompt to reselect
    - System never hard-fails fallback or reschedule options are always offered

E09. Ensure Secure and Transparent Job Payment Confirmation After Completion

Statement

Customers must receive a detailed invoice after job completion that shows the rate per student, time worked, and any tool-related charges, with the ability to confirm or dispute before payment is processed.

Acceptance Criteria

1. Triggering the Invoice Review

- After the student marks the job as "Completed", the system:
    - Generates a detailed invoice
    - Notifies the customer and displays the confirmation screen
    - Starts a 24-hour countdown before auto-confirmation

2. Detailed Invoice Layout

- The invoice includes the following:
- Hands assigned: Name, hourly rate, total hours per Hand
- Time Summary: Start time, end time, total duration
- Tool Charges: Each tool used, quantity, individual cost
- Subtotal: Labor + tools, before tax (if applicable)
- Total: Final amount due, with tax and payment method

    Julia (Hand) - $20/hr × 2.5 hrs + $5/hr (Mower)= $62.50
    Marcus (Hand) - $22/hr × 2 hrs = $44

- An invoice is generated per hand

3. Customer Action Options

- Two buttons at bottom of screen:
    - Confirm and Pay (charges are processed)
    - Report an Issue (opens support contact form with invoice attached)
- Timer shown:
  "If no action is taken, this job will be auto-confirmed and charged at [HH:MM]."

4. Handling Multi-Hand Jobs

- An invoice is created for each hand as described in AC 1
- Disputes will be handled individually so they won't affect other Hands invoices

5. Tool Cost Tracking

- Tool charges are based on:
    - What the student indicated during job setup
    - Admin-defined tool pricing (editable via Admin Panel)
- Only tools marked as during job booking flow

6. Support Escalation Path

- If the customer taps “Report an Issue”:
- A short message box appears:
  “Tell us what’s wrong so we can review this before charging your card.”
- Auto-confirmation and payment are paused until resolved by support
- Status marked as “Under Review” in backend

7. System Logging & Payment Confirmation

- Upon confirmation (manual or automatic), payment is processed and:
    - Timestamp recorded
    - Invoice locked and archived
    - Confirmation receipt emailed to the customer

8. Fallbacks & Payment Errors

- If payment fails on confirmation:
    - Job is marked “Pending Payment”
    - Customer is prompted to retry payment
- A 48-hour window is granted to resolve the issue
- Reminders are sent every 12 hours
- If unresolved, support team is alerted to intervene

E10. Trigger Customer Rating Flow for the Hand and App After Job Payment Confirmation

Statement

After a job is confirmed and payment is processed, the customer is prompted with a friendly, lightweight two-step flow to rate both the Hand and the app experience. Feedback is stored for platform quality, ranking, and analytics.

Acceptance Criteria

1. Rating Prompt Timing and Visibility

- The rating flow appears immediately after job payment confirmation, with no additional taps or navigation required.
- The prompt appears as a modal overlay or in-app screen before navigating to the final job summary.
- If the app is closed before completing the flow:
    - The user is prompted again only once upon their next login.

2. Step 1 – Rating the Hand (Student)

- User is asked:
    - "How would you rate [Hand Name] for this job?"
    - Options:
        - Star rating (1–5 stars)
        - Optional free-text comment field: "Tell us what stood out or what could've been better." Field accepts up to 500 characters.
        - A "Skip" button is available (no forced rating)

3. Step 2 – Rating the App

- After submitting or skipping the Hand rating, the user sees:
    - "How was your experience using the app?"
    - Star rating input (1–5 stars) only
    - No comment field required
    - Option to skip this step as well

4. Skip Handling and Re-Prompt Logic

- If the user skips or closes the rating modal:
    - The app records the skip event for both the Hand and/or app feedback
    - The system will not prompt the user again for the same job
    - A separate "Rate Your Experience" entry in the app (e.g., past jobs section) may allow optional feedback later (out of scope for this Epic)

5. Feedback Storage and Association

- All submitted ratings are stored in the backend with:
    - Job ID
    - Customer ID
    - Student ID
    - Timestamp
    - Star values
    - Optional comments
    - Tagged as: "Customer Rating – Hand" or "Customer Rating – App"

6. Impact on Student Profile

- The student's public rating is updated based on an average of their last X ratings (logic defined separately).
- Feedback comments may be reviewed before being shown in student profiles (if applicable, to confirm with stakeholders).

E11. Prompt Satisfied Customers to Rate the App in the Store via Smart Push Campaigns

Statement

The platform identifies happy users and prompts them via push or in-app prompt to rate the app publicly, boosting organic reviews in the App Store or Google Play.

Acceptance Criteria

1. Target User Identification

- Eligible users include:

  - Users who rated the app or a Hand 4 or 5 stars internally
  - Users who completed at least 1 job without reporting an issue
  - Users who logged in or completed an action in the last 7 days
- User selection is randomized or batched daily to avoid over-triggering

2. Push Notification Logic

- A push notification is sent to selected users with copy such as:
  "Loving SpareHand? Rate us in the app store and help others discover the app"
- On tap, the app opens a deep link to the App Store (iOS) or Google Play (Android)

3. Store Redirect Behavior

- iOS users are redirected to the App Store review screen via native deep linking
- Android users are sent to Google Play's "Rate This App" intent
- If deep link fails, the app shows a fallback message:
  "Can't open the store right now. You can rate us later from your profile. Thank you"

4. One-Time Prompt Enforcement

- Each user can only be prompted once every 6 months or once per app version
- The system records if the user:
  - Clicked the prompt
  - Skipped it
  - Was already redirected previously

5. Fallback Entry Points

- A "Rate the App" option is permanently available in the profile or settings screen
- Visible only if the user has completed at least one job successfully

6. Exclusion Logic

- Users who gave 1–3 star ratings in internal feedback are excluded from this flow
- Users with open support tickets or flagged behavior are also excluded from the prompt

E12. Enable In-App Tipping for Students After Job Completion with Percentage-Based Suggestions and Thank-You Acknowledgment

Statement

After a job is completed and confirmed, customers must be prompted to leave a percentage-based tip, and Hands must be able to acknowledge tips with a simple, meaningful thank-you message through the app.

Acceptance Criteria

1. Tipping Flow Activation

- Triggered after the job is confirmed and paid by the customer. Appears as:
  "Would you like to leave a tip for [Hand Name]?"

2. Tip Amount Options

- Preset options are percentages of the total job cost:
  - 10%
  - 15%
  - 20%

- Also includes custom Amount (manual entry)
    - Custom entry shows currency input field
- Max custom tip allowed: $1,000 (configurable via admin panel)
- Validation shown if the amount exceeds limit
- Example UI:
    - *“Select a tip amount based on your total of $75”*
    - $7.50 (10%) $11.25 (15%) $15.00 (20%) Custom

3. Tip Confirmation and Payment

- Customer taps *“Submit Tip”*
- Tip is charged as a separate transaction to the saved card
- Confirmation message:
    - *“Thanks! Your tip has been sent to [Hand Name]”*

4. Hand Notification and Thank-You Option

- Once tip is processed, the Hand receives a push:
    - *“You received a $15 tip from [Customer First Name]”*
- The Hand sees a “Send Thanks” button in the job completion or earnings screen
- When tapped, a preset message is sent back to the customer:
    - *“[Hand name] thanked you for the tip”*
    - This appears as a push or in-app message on the customer side
- Only one thank-you can be sent per job; no chat thread is opened

5. Customer-Side Experience

- After receiving the thank-you:
    - The customer sees a brief popup or toast notification
- Optionally includes:
    - “Want to hire Zach again?” ⟶ Opens 'My Hands'

6. Edge Case Handling

- If customer skips tipping:
    - Tipping module doesn’t reappear for that job
- If customer closes the app:
    - Prompt can reappear later in notifications
- If Hand does not tap "Thank You":
    - No message is sent no follow-up or reminder is shown

7. Admin Control

- Admin dashboard can configure:
    - Percentage options
    - Custom tip limit
    - Enable/disable thank-you replies
    - View tipping metrics (job ID, amount, who tipped, thank-you sent)

E14. Enable Recurring Job Setup Prompt After Job Completion

Statement

After completing a job, the system prompts the customer to convert the job into a recurring booking by selecting a repeat frequency and preferred day. If declined, the user proceeds with the normal post-job flow.

Acceptance Criteria

1. Prompt Trigger After Job Completion

- Once the job is marked as completed and confirmed, show a prompt card or modal:
  - "Would you like to make this a recurring job?"
  - Placement: After invoice confirmation, Before rating and tipping screen
  - Options:
    - Yes
    - No, just this time

2. Recurring Job Setup Flow

- If customer selects Yes, show a short 2-step flow:
  - Frequency Selector
    - Weekly, Every 2 Weeks, Monthly, Bi-Monthly
  - Day & Time Selector
    - Pick a preferred day of the week
    - Pick a time block (e.g., Morning, Afternoon, Evening)
  - Confirmation message:
    - "Your recurring job is now scheduled for every [Day] at [Time], starting [Next Available Date]."

3. Default Data Pre-Fill

- The new recurring job:
  - Uses the same Hand, job category, address, and job setup as the original (if the Hand is available)
  - If not available, the system will auto-match each time
  - Editable later via job management section

4. Not Recurring Job

- If "No" Is Selected, the customer is taken to the standard flow:
  - Invoice
  - Tipping
  - Rate the Hand
  - Rate the App (if enabled)

5. Recurring Job Visibility

- Recurring jobs appear in:
  - "My Jobs" section under a label: "Recurring"
  - Can be edited or canceled at any time

6. Admin and System Controls

- Admin can:

    - Configure available frequencies
    - Set how far in advance the system books next occurrences
    - View adoption rate and recurring job analytics
- System logic ensures:
    - Conflicts are flagged if customer or Hand is unavailable on chosen day
    - Notifications are sent ahead of each job (e.g., 24–48 hrs)

E16. Deliver App-Wide Onboarding Walkthrough for All Users After Major Updates

Statement

All users (new and returning) must be presented with a one-time, required onboarding walkthrough after a major update that introduces new features, usage instructions, and platform protocols.

Acceptance Criteria

1. Universal Walkthrough Trigger

- Automatically shown on first login after release (regardless of user status)
- Only shown once, but must be completed or explicitly skipped
- Controlled by version flag or release marker in backend (e.g., tutorial_v2_seen)

2. Content Scope

- Walkthrough content includes:
    - New feature overviews (e.g., “Recurring Jobs,” “My Hands,” “Track Job Status”)
    - Booking protocol reminder: *“All jobs must be requested and managed through the app.”*
    - Payment policy: *“All payments must be processed through the platform no cash or offline transfers.”*
    - Support access: *“Need help? Visit our support section anytime.”*
    - Highlight benefits (e.g., referral program, promotions)

3. Walkthrough Format

- Multi-step slides or tooltips with:
    - Title + short description
    - Visual/illustration or screen highlight
    - “Next” / “Skip” / “Done” navigation
- Optional CTA on final screen:
  “Let’s explore the app!”

4. Behavior After Dismissal

- Once completed or skipped:
    - User is not shown the walkthrough again
    - Option to revisit walkthrough available in: Profile > Help
- Optional “What’s New” button or banner

5. Design & UX

- Walkthrough is accessible, mobile-optimized, and branded
- Supports localization

- Uses concise, friendly tone with focus on clarity
- Follows large tap targets and contrast guidelines

6. Analytics & Admin Tracking

- Track:
  - Tutorial shown (yes/no)
  - Completion vs. skip rate
  - Most dropped-off slide
  - Post-walkthrough behavior (e.g., jobs booked, help accessed)
- Admin can configure:
  - Walkthrough versioning
  - Enable/disable by app version
  - Control which slides appear per segment or role

E23. Enable Customers to Apply Discount Coupons During Job Request or Payment Flow

Statement

Customers must be able to enter and apply valid coupon codes while creating a job or reviewing the invoice before confirmation, and the discount must be clearly reflected in the price breakdown.

Acceptance Criteria

1. Coupon Code Input Field

- A visible field is available in the job summary or invoice screen:
  - “Have a promo code?”
  - Tap or click to open input field
  - Field accepts alphanumeric codes (e.g., SAVE10, FRIEND5)

2. Validation and Error Handling

- Upon entering the code and tapping "Apply":
  - The system validates the code against: Expiration date, Usage limits, Eligibility (e.g., new users only, job category restrictions)
- If invalid:
  - “Invalid or expired promo code. Please try again.”

3. Discount Application Logic

- If valid:
  - Discount is applied to the total job cost
  - Supports: Fixed-amount discounts (e.g., $10 off) and percentage-based discounts (e.g., 15% off)
- Updated pricing shown:
  - Original total
  - Discount applied
  - Final amount due

4. Invoice and Receipt Integration

- Applied promo is reflected in:

  - Job confirmation screen
  - Customer invoice/receipt
- Admin reports
  - Line item example: "Promo: SAVE10 – $10 discount applied"

5. One-Time Use & Tracking

- For one-time codes:
  - System marks code as used once job is confirmed
- Code usage is tracked by:
  - User ID
  - Job ID
  - Timestamp

6. Admin Configuration

- Admins can:
  - Create, edit, and deactivate promo codes
  - Set: Code name and type (fixed/%)
  - Value and usage limit
  - Expiration date
  - Restrictions (e.g., new users, job types)
  - Export usage reports

E24. Implement Scalable In-App Notification and Popup System for Customer Engagement

Statement

Customers receive relevant in-app notifications and popups based on defined platform events, behavioral triggers, or campaigns. Notifications are managed dynamically via a scalable logic layer and can be created, targeted, and tracked through the admin panel.

Acceptance Criteria

1. Notification Types Supported

- System must support the following scalable categories:
  - Product Updates (e.g., new features, UI changes)
  - Promotional Offers (e.g., limited-time discounts or credit rewards)
  - Behavioral Triggers (e.g., abandoned job request, return after inactivity)
  - Market Expansion Announcements (e.g., new university or region launch)
  - Referral Prompts (linked to active referral logic)

2. Trigger Conditions and Logic

- Notifications and popups are shown based on:
  - User attributes (new vs. returning, location, segment)
  - Behavior (e.g., started job request but didn't finish)
  - Scheduled platform events (e.g., new launch date)
  - System rules (e.g., feature rollout → show once on next login)

3. Content Management and Flexibility

- Content (title, body, CTA) must be:
  - Fully configurable in the admin panel
  - Localizable
  - Reusable with templates (e.g., promo, referral, new launch)
- Notifications can optionally include:
  - Icons or emojis
  - Redirect action (e.g., open job creation, open wallet, open referrals)

4. User Experience and Delivery

- Notifications may appear as:
  - Modal popup (on app open or after trigger)
  - Inline banner (top of screen or job request flow)
  - In-app message center (optional future feature)
- Each message includes:
  - One-time display rules (e.g., show once per user)
  - Dismiss logic (with or without action)
  - Display delay (e.g., X seconds after app open)

5. Redemption and Action Tracking

- If notification links to an action (e.g., apply credit, start job, share referral):
  - System tracks whether the action is completed
  - Suppresses future reminders for same campaign if completed
  - Allows logging of redemption metrics (e.g., promo claimed, referral shared)

6. Admin Management Capabilities

- Admins can:
  - Create and edit notification campaigns
  - Set: Target audience (segment, region, device type), trigger logic (event-based, schedule-based, behavior-based), display format (popup, banner, etc.)
  - Monitor performance (views, clicks, conversions)
  - Archive or pause campaigns without deleting them

7. Technical and Delivery Controls

- Supports:
  - Offline caching until app is opened
  - Priority queues for urgent vs. passive notifications

5.2. Students Mobile App

E1. Improve Sign-Up Experience for Hands by Expanding Input Options and Highlighting Referral Benefits

Statement

The student sign-up process must begin with a dedicated verification step using a .edu email (or approved college domain), followed by a clean registration form with personal credentials, phone number, and referral handling all within a modern, step-based mobile UX.

Acceptance Criteria

1. Step 1: Student Status Verification

- When a user taps Sign Up or opens a referral link, the app shows:
  - "Welcome to Spare Hand Students! Let's first verify you're a student."
- Fields on this screen:
  - College Email Address (must match approved domain)
  - Send Verification Code button
  - Code input field appears after email is entered
- Flow:
  - User receives a 6-digit verification code
  - Upon successful verification, proceed to Step 2
- If failed (after X attempts), user sees:
  - "Couldn't verify your student email. Please try again or contact support."

2. Step 2: Complete Your Sign-Up

- After verifying student email, the user fills out the actual sign-up form:
  - Primary Email (used for login, not necessarily the school email)
  - Phone Number (required, with SMS validation in future)
  - First Name
  - Last Name
  - Password
  - Referral Code (optional, auto-filled if from link)
- Important: School email is used for verification only and is not stored as login email unless manually re-entered.

3. Referral Code Handling

- If a user lands via a referral link (?ref=NAME123):
  - Code is auto-filled in the sign-up form
  - Tooltip below field: "You're signing up with a referral! Complete sign-up to unlock your rewards."
- Manual edits allowed

4. Tooltips & Microcopy

- Inline messages and prompts clarify:
  - Why the student email is needed

    - Why personal email is used for login
- Referral benefit messages (e.g., “Get early access to top jobs!”)

5. UI/UX Structure

- Stepper UI or progressive screens (Step 1 → Step 2)
- Clean, distraction-free screens with CTA focus
- Continue buttons disabled until inputs are valid

6. Error and Fallback Handling

- If user exits mid-way:
    - Step progress is cached locally for session recovery
    - If student email domain is not allowed: Suggest user contact support to request domain approval

7. Admin Configurable Elements

- Allowed email domains (e.g., .edu, specific university domains)

E2. Streamlined Onboarding Flow

Statement

Students complete a structured onboarding flow after sign-up, including location confirmation, job category selection, tool ownership with compensation visibility, photo upload with validation, bio entry, and payment method setup all required to activate their account for job acceptance.

Acceptance Criteria

1. Confirm Match Location via Student Email

- After email verification (handled in sign-up), the system:
    - Matches student to their service area based on email domain
    - No geolocation is used to determine location
- UI shows:
  “You’ve been matched to [City, County] based on your university email.”
- No option to override manually
- This location determines:
    - Job feed
    - Availability calendar
    - Customer-facing location label

2. Select Job Categories

- Student sees a grid of predefined job categories (e.g., Yard Work, Moving Help, Cleaning).
- Each card includes:
    - Icon
    - Short description
    - Select/deselect toggle
- Student can choose one or more based on their interest and skill
- Selections are stored for:

    - Job matching logic
    - Displaying relevant tools in the next step
    - Calendar availability setup (in a future flow)

3. Tool Selection and Compensation Setup

- Based on selected job categories, the student is shown a list of tools they may own.
- Each tool item includes:
    - Name (e.g., Lawn Mower, Vacuum)
    - Checkbox to select ownership
    - Compensation rate clearly shown (e.g., "+$5/hr")
- Tooltip or help text:
  "You'll earn this bonus if the customer requests this tool and you bring it to the job."
- Student must confirm ownership:
  "I confirm I own and can bring these tools if required"
- This data is stored and used for:
    - Job-tool matching
    - Tool-based payout calculations
    - Students can later edit tools in their profile

4.Upload Profile Picture

- Students must upload a clear, friendly photo of themselves
- Guidelines shown:
  "Your face should be clearly visible. No filters, hats, or group photos."

5. Write Personal Bio

- Prompt:
  "Tell us your superpowers!"
- Input field:
    - Max 280 characters
    - Real-time character counter
- Help text:
  "What are you great at? What makes you a reliable Hand?"
- Required to proceed

6. Add Payment Method

- Students are required to connect a payment method (e.g., via Stripe Connect)
- Message shown:
  "This is how you'll get paid for your jobs!"
- Until payment method is verified:
    - Student is inactive in the job assignment system

E3. Require Semester-Based Availability Setup to Determine Student Activation Status

Statement

Students must define and submit their weekly availability once per semester in order to be considered "active" and eligible to receive job requests. They can update it at any time, but must re-submit at the start of each new academic term.

Acceptance Criteria

1. Semester-Based Prompt

- After onboarding and at the beginning of every semester (e.g., Fall, Spring), students see a required screen:
  “Set your availability for this semester so we know when to match you with jobs.”
- The availability prompt must be completed to access job opportunities or appear as “active”

2. Weekly Schedule Setup

- Students define their availability per day of the week (e.g., Monday 2–6pm, Thursday 8am–12pm)
- Availability is recurring weekly throughout the semester unless updated by the student

3. Activation and Visibility Logic

- A student is marked as “Active” when:
  - They’ve submitted availability for the current semester
  - The current time falls within one of their selected slots
- Outside their defined availability, they appear as “Inactive” and are excluded from customer selection or auto-matching

4. Revalidation at Semester Start

- At the beginning of a new semester:
  - All previous availability data is cleared
  - Students are prompted again to set new availability
  - Until they do, they are automatically marked as Inactive
- Reminder notifications are sent (e.g., push, email):
  “It’s a new semester update your availability to keep receiving job requests!”

5. Manual Updates Allowed Anytime

- Students may manually update their availability at any time during the semester
- Changes are applied immediately and reflected in their “Active” or “Inactive” status

6. Pre-Semester Notification Campaign (6 Weeks Out)

- Starting 6 weeks before the upcoming semester, the system begins sending weekly reminders to students who:
  - Have not yet submitted their availability for that semester
  - Were active in the previous semester
- Notification cadence:
  - 6–4 weeks out:
  - Tone: Friendly reminder
  - Example: “New semester starting soon! Set your schedule now to keep receiving job requests.”
  - 3–1 weeks out:

    - Tone: Urgent
    - “Don’t forget to submit your schedule or you’ll be marked inactive next semester!”
    - Final push (48 hours before semester starts): “Last chance! Without availability, you won’t be matched to any jobs this semester.”

7. In-App Widget & Banner Alerts

- A persistent alert or banner is shown in the student’s app (on Home screen or via widget) if:
    - The semester is upcoming
    - Availability has not yet been set
- Messaging includes:
  “Semester availability missing: Set it now to stay active”
- “Update Now” button → opens availability flow
- Once the availability is submitted, the banner disappears

8. Activation Status Logic Based on Upcoming Semester

- If a student does not submit their schedule before the semester start date:
    - They are marked as inactive
    - They are excluded from job matching
- A message is shown when they open the app:
  “You’re currently inactive for [Semester Name]. Set your availability to start receiving jobs again.”
- If they later submit their schedule:
    - Status is re-evaluated and they are reactivated (if the semester is still ongoing)

9. Admin Configurability

- Admins can:
    - Define the semester and academic break start/end dates (as part of university setup)
    - Configure the reminder cadence (e.g., every 7 days)
- Adjust message templates and tone for push notifications

E4. Design an Actionable Home Screen for Students with Availability Status, Job Overview, and Referral Visibility

Statement

The student home screen must provide a clear overview of their current status, key performance indicators (e.g., earnings, upcoming jobs), and promote referrals in a visible, actionable way.

Acceptance Criteria

1. Availability Status Widget

- Displayed at the top of the home screen
- Shows one of the following:
    - “You are Active” + “Next availability: [Time]”

    - "You are Inactive" with explanation
- CTA: "Update Availability" → navigates to schedule screen
- If no semester availability has been set, show alert:
    - "You haven't set your availability for this semester. Tap to stay active and keep receiving jobs."
- Widget will persist until student updates their schedule
- Availability refreshes in real time from their availability module

2. Referral Module – With Referred Status & Popups

- Persistent referral card includes:
    - Student's referral code (e.g., HANDNAME10)
    - Copy button
    - Share button → opens native share modal
    - Description: "Earn $5 for every friend who joins and completes their first job!"
- Below the referral card, show status of invited friends:
    - Pending: "Zach (Pending – hasn't completed a job yet)"
    - Confirmed: "Jeremy Referral complete. $5 pending in wallet"
- Referral Reward Logic UI:
    - $5 shows in wallet view for both:
        - Referring Hand → locked until friend completes job
        - Referred Hand → bonus shown as "locked" until first job is completed
    - Tooltip or lock icon: "This bonus will be available once your friend completes their first job"
- Popup triggers when friend uses the code:
    - Referring Hand sees: "Congratulations! [Friend Name] used your referral code! Enjoy $5 in your wallet once they complete their first job"
    - Referred Hand sees: "You used [Hand Name]'s referral code! You'll get $5 when you complete your first job"
    - These messages must include emojis

3. Upcoming Jobs Preview

- If student has accepted jobs, show cards:
    - Job type
    - Scheduled date/time
    - Customer name
    - Distance to job location (e.g., 2.5 miles) shown in badge
    - CTA: "View Job Details"
- If no jobs:
    - "No jobs booked yet. Set your availability and be ready!"
    - Note: Address is not shown until the job is accepted, to protect privacy. Only estimated distance is visible upfront.

4. Earnings Snapshot

- Displays:

    - "You've earned $X this month"
    - Button: "View payout history"
    - Optional line: "Next payout expected: [Date]"
- Tapping opens full wallet view with locked/unlocked balances
- Referral rewards (locked) also shown here under "Pending Earnings"

5. Seasonal Availability Alerts (Pre-Semester Logic)

- If semester availability has not been updated and the new semester is approaching, show notice banner or card at the top of home:
    - ":calendar: New semester starting soon! Set your availability to stay active."
- Integrated with reminder logic from pre-semester push campaign
- Once availability is submitted, alert is dismissed

E5. Enable Students to Receive and Respond to Direct Job Requests from Customers

Statement

Students must receive, review, and respond to direct job requests from customers via push notification and in-app interface, with the ability to accept or decline within a defined time limit.

Acceptance Criteria

1. Triggering the Request

- A customer creates a job and selects a specific student (e.g., from "My Hands")
- The system sends a real-time push notification and in-app alert to the selected student
- The job enters a "Pending Student Response" state

2. Notification Content

- The push notification includes:
    - "You've been requested by Sam for a Yard Work job on Friday at 3pm. Tap to view details."
- Tapping the notification opens a job request preview screen

3. Job Request Preview Screen

- Displays the full job details:
    - Customer name (optional, if enabled)
    - Job category
    - Date & time
    - Address
    - Tools needed (if applicable)
    - Estimated duration
    - Payment info (amount or estimated range)
    - Two actions:
        - Accept
        - Decline

4. Response Window

- Student has a limited time to respond (e.g., 20 minutes)
- A countdown timer is shown on the screen

- If no response is received:
  - The system marks it as “No Response”
  - The job reverts to automatic matching or another fallback flow
- The customer is notified:
  - “[Hand Name] didn’t respond. We’re finding someone else.”

5. After Accepting

- Job status is updated to “Accepted by Hand”
- The customer is notified:
  - “[Hand Name] accepted your request!”
  - The job is now visible in the student's upcoming jobs section

6. After Declining

- The customer is notified that the Hand declined, and prompted to:
  - Choose another Hand
  - Allow automatic matching
- The system records the decline in the student’s history (for reporting, not punishment)

7. Fallback and System Behavior

- Only active students (based on availability) can be selected by customers
- The system ensures the selected student is available for the proposed time
- If not, the customer is prompted to select a new time or choose auto-matching

E6. Enable Students to Update Real-Time Job Status Through Defined Service Stages

Statement

Students must update their job status through a guided 3-step interface: “On the Way,” “Job in Progress,” and “Job Completed.” The system supports this flow with timestamped tracking and safeguards plus nudging logic to help prevent students from forgetting to update stages.

Acceptance Criteria

1. Stage 1: On the Way

- Trigger: Student taps “I’m on the way”
- Visual Feedback:
  - Status bar: Stage 1 active
  - Message: “[Hand Name] is on the way” (sent to customer)
- Logic:
  - Timestamp recorded (server + device time)
  - Button becomes disabled after tap to avoid re-trigger
- Reminder logic:
  - If the job is scheduled and status is not marked 15 minutes earlier:
  - Push Notification: “Don’t forget to tap ‘I’m on the way’ to let the customer know!”

2. Stage 2: Job in Progress

- Trigger: “Start Job” becomes active only after Stage 1 is marked
- Visual Feedback:

  - Status bar moves to Stage 2
  - Message: “Work in progress” shown to customer
- Logic:
  - Timestamp recorded
  - Cannot skip ahead without Stage 1 completed
- Reminder logic:
  - If “On the Way” was tapped, but “Start Job” is not triggered within X minutes after expected arrival time:
  - Reminder banner in app: “Start your job now to keep your timeline accurate”
  - Optional: Gentle push: “Looks like you’ve arrived ready to start the job?”

3. Stage 3: Job Completed

- Trigger: Tap “Mark as Completed”
- Visual Feedback:
  - Status bar shows Stage 3 complete
  - Message: “Job completed by [Hand Name]” appears to customer
- Logic:
  - Timestamp recorded
  - No further actions allowed after this point
- Reminder logic:
  - If job is “In Progress” for more than expected duration (e.g., 2x estimated time):
  - Push or in-app banner: “Don’t forget to mark your job as completed when you finish”

4. UI Behavior & Restrictions

- Sequential stages only
- Sticky status bar across job screen
- Each stage is visually highlighted as active
- No access to later stages unless the previous is complete

5. Backend Sync & Offline Support

- Every action is logged with:
  - Job ID
  - Hand ID
  - Stage timestamp
  - Device and server sync
- If offline:
  - Actions are queued locally and synced once reconnected

6. Customer-Side Visibility

- Each status update appears live on the customer’s tracking interface
- Includes:
  - Status message
  - Icon
  - Timestamp

- Ensures customer trust and transparency during the job

7. Fallback Logic When "On the Way" Is Not Triggered

- If the student skips Stage 1 and attempts to tap "Start Job" directly:
  - The system shows a warning modal: "You haven't marked 'I'm on the way'. Do you want to start the job now anyway?"
  - Options:
    - Yes, Start Now
- System logs Stage 1 and Stage 2 timestamps at the same moment
- Customer is shown:
  - "[Hand Name] started the job" (skips "On the Way" status)
  - Cancel → returns the user to the current job screen with Stage 1 still available
- Optional safeguard:
  - If Stage 1 hasn't been tapped 15 minutes after scheduled job time, app shows: "Looks like you may have started your job. Don't forget to update your status!"
  - With CTA: "Start Job"

E7. Allow Students to Review and Confirm Auto-Generated Invoice After Job Completion

Statement

After marking a job as completed, students are shown the system-generated invoice with a 2-hour window to review and confirm or flag it. If confirmed, it's sent to the customer. If flagged, it's routed to support. If no action is taken, it's auto-sent after the window expires.

Acceptance Criteria

1. Trigger and Invoice Generation

- When the student taps "Mark as Completed", the platform:
  - Automatically generates a draft invoice
  - Opens a review screen before finalizing the job

2. Invoice Review Screen

- Displays all invoice details:
  - Customer name
  - Job type
  - Date and time of service
  - Start and end timestamps
  - Total duration
  - Hourly rate and total earnings
- A button is available: "Add Additional Expenses"
- Tapping opens a modal with:
  - Item name (free text, required)
  - Cost (numeric input, required)
  - Mandatory: photo of receipt
- A message bar at the top shows:

"Please review your invoice before it's sent to the customer. You have 2 hours to confirm or flag it."

4. Expense Entry Constraints

- Students can add multiple items
- Each item is:
  - Validated (max cost per item: configurable, e.g., $100)
  - Clearly marked as "Customer to reimburse"
- It supports .jpg/.png and shows preview

5. Action Buttons

- Two actions are available:
  - "Confirm and Send": Invoice is finalized and delivered to customer immediately
  - "Request Change": Student is shown a support contact form or support chat with pre-filled job details

6. Automatic Sending if No Action

- If the student does not respond within 2 hours:
  - The invoice is automatically sent to the customer
- A notification is sent to the student:
  "Your invoice was sent to the customer automatically. Thank you!"

7. Support Escalation Flow

- If student taps "Request Change":
  - The invoice is held from being sent
- A support ticket is generated with: Student comment, Invoice data, Job ID and timestamps
- Support can manually edit and resend once corrected
- Student and customer are notified when resolution is complete

8. Customer-Facing Summary

- Reimbursable expenses are shown as separate line items in the invoice:
  - "Reimbursable: Paint – $15"
  - "Reimbursable: Garbage Bags – $6.50"
- If a receipt was added, customer sees a thumbnail they can expand
- Total section includes:
  - Job labor
  - Tool fees (if applicable)
  - Reimbursable items
  - Final Total

9. Invoice Review and Confirmation by Student

- Student sees full invoice with:
  - All time- or tool-based charges (auto-filled)
  - Any added reimbursables
- Before submitting, CTA reads:
  "Confirm & Send Invoice to Customer"

- Once sent, invoice becomes read-only
- Student is reminded:
  "Make sure expenses are accurate. The customer will review this before confirming payment."

10. System & Admin Controls

- Admin can:
  - Enable/disable reimbursable feature by category
  - Set max per-expense amount
  - View submitted expenses for review if flagged
- All added expenses are logged with:
  - Job ID
  - Student ID
  - Timestamps
  - Attachments (if any)

E8. Dedicated "Availability" Section for Students to Manage When They Can Accept Jobs

Statement

Students can manage their availability from a dedicated section in the app, including semester-based weekly schedules, day-level overrides, and break-time blackout windows, all of which control their visibility for receiving job offers.

Acceptance Criteria

1. Access to Availability Section

- Available from:
  - Main navigation menu
  - Home screen card (e.g., "Set My Availability")

2. Semester-Based Weekly Schedule

- Upon entering or updating availability:
  - The app displays the current semester (pulled from university config)
- Student defines recurring weekly availability by selecting:
  - Days of the week (e.g., Mon, Wed, Sat)
  - Availability per day (e.g., 8:00 AM–12:00 PM, 2:00 PM–6:00 PM)
  - Preview calendar shows summary of set hours for the week

3. Short-Term Overrides

- Students can toggle real-time availability with:
  - "Unavailable Today"
  - "Mark Available Now" (if not previously scheduled)
- These changes apply only for the current day and reset automatically

4. Custom Breaks and Blackout Dates

- Students can add personal breaks with:
  - Start date
  - End date

    - Label (e.g., “Spring break” or “Summer”)
- During these dates, they are marked as inactive for job matching

5. University Breaks Integration

- If breaks are configured by admin (e.g., Spring Break, Summer), these appear as:
    - Read-only suggestions in the availability screen
    - Option: “Do not offer me jobs during this break” (default ON)

6. Active/Inactive Status Logic

- Student status is considered “active” if they have availability configured for the current semester
- Current time falls within one of their set blocks
- They are not marked unavailable (daily override or break)
- This status determines whether they appear in job matching and auto-assignment logic

7. Notifications and Nudges

- At the start of a new semester, students are prompted in aligment with epic: Require Semester-Based Availability Setup to Determine Student Activation Status
- If availability hasn’t been set, student remains invisible to customers

8. Audit and Logging

- Every update to availability is timestamped and stored for admin audit
- Admin can view last update date and active/inactive state from the panel

E9. Enhance View Their Job History with Status, Details, and Payment Summary

Statement

Students can access a “Job History” section from the app menu, where they can view all their past jobs categorized by status, filter by date or type, and see detailed records including payment confirmation and timestamps.

Acceptance Criteria

1. Access and Entry Point

- The “Job History” section is accessible via:
    - Navigation menu
    - Home screen CTA tile

2. Job History Overview

- Display a list of past jobs with cards showing:
    - Job type
    - Customer name or alias
    - Date/time
    - Status (e.g., Completed, Canceled, Expired)
    - Quick tag: “Paid” / “Pending” / “Resolved”
- Jobs can be sorted by:
    - Most recent
    - Oldest
    - Status

3. Job Detail View

- Tapping a job opens the job detail screen with:
    - Full job description
    - Job category
    - Scheduled date/time
    - Job duration
- Timestamps: Accepted, On the way, Started, Completed
- Final invoice amount
- Payment status: “Paid”, “Processing”, “Flagged”
- Option to contact support if something looks wrong

4. Filtering and Search

- The student can filter job history by:
    - Date range
    - Status (Completed, Canceled, etc.)
    - Job category
    - A search bar allows keyword entry (e.g., “Yard Work”, “Sam”, “Feb 10”)

5. Support Access

- Each job includes a button:
- “Need help with this job?” → Opens FAQ

6. Invoice Confirmation

- For completed and paid jobs, students can view and download the invoice (if supported)
- Invoices show:
    - Breakdown of time worked
    - Hourly rate
    - Total earnings
    - Payment method used

E10. Enhance Referral System with Shareable App Link and Automatic Code Pre-Fill

Statement

Hands can share a referral link that includes their code. When a new user downloads the app and signs up through that link, the referral code is automatically applied, and both users are tracked and rewarded after referral conditions are met.

Acceptance Criteria

1. Referral Code Assignment

- Each student is assigned a unique referral code (e.g., ZACH10) upon account creation
- This code is stored in their profile and shown in the “Referral” section of the app

2. Shareable Smart Referral Link

- The app generates a dynamic referral URL: https://sparehand.app/signup?ref=ZACH10
- The link redirects based on device:
    - iOS: App Store
    - Android: Google Play

- Upon app install and first open, the referral code from the URL is retained and auto-filled into the sign-up form

3. Referral Section UI for Students

- A dedicated "Referrals" screen in the app includes:
    - The user's referral code
    - Their smart shareable link
    - Copy / Share buttons (triggers native share modal)
    - A brief description: "Invite friends and earn $5 when they complete their first job!"
    - View of referral progress (e.g., "2 friends signed up • 1 completed job")

4. Auto-Fill Referral Code on Sign-Up

- When a new user opens the app via a referral link:
    - The sign-up screen's "Referral Code" field is pre-filled
    - The field is editable but shows a tooltip: "You've been invited by [Hand's name], enjoy your welcome bonus!"
    - If they clear the field, a warning appears: "Removing the referral code will cancel your bonus eligibility."

5. Tracking and Eligibility

- The referral is marked as "pending" until:
    - The referred user completes their sign-up
    - AND successfully completes their first job
- Once validated:
    - The referrer receives a reward (e.g., bonus payout, notification)
    - The referee (new user) also receives their welcome bonus (optional credit or payout)

6. Referral Rules and Edge Cases

- A user cannot enter multiple referral codes
- Referral codes cannot be applied retroactively after sign-up
- A user cannot refer themselves (by device fingerprint or email match)
- Admin panel can override and validate referrals manually if needed

7. Reward Notifications

- Referrers receive:
    - Push notification: "[Hand's name] completed their first job! You just earned your $5 reward!"
    - Option to view referral history
- Referees receive confirmation of their own welcome reward

E12. Enable Students to Manage Their Profile and Account Settings from the App

Statement

Students can review and edit the information they submitted during onboarding from a persistent "My Profile" section in the app, including personal, academic, and payment data.

Acceptance Criteria

1. Access to the Profile Section

- Available via:
    - Main navigation menu
    - Home screen CTA (e.g., "Edit my profile")
- Displays:
    - Student name
    - Profile photo
    - Current status: Active / Inactive
    - Quick access links: Availability, Payment, Referrals

2. Editable Fields Originating from Onboarding

- Students can edit but not skip key information originally provided:
    - University name
    - Expected graduation year
- If changed to a past year, system triggers status revalidation logic
    - Profile photo - Upload new photo (must meet quality standards)
    - Personal bio - Short, friendly description of strengths or personality
    - Phone number - Requires SMS re-verification on update
    - Email address - Requires email re-verification if changed

3. Payment Method Management

- Section: "Payout Settings"
- View, add, or update bank account or payment provider
- Display next payout date (if applicable)
- Security confirmation before changes are saved

4. Account Access and Security

- Manage:
    - Change password
    - Logout
    - Request to delete account (leads to confirmation flow or redirects to support)

5. Validation and Audit Flow

- All edits show real-time validation messages (e.g., "Invalid graduation year")
- Major changes (payment, university, graduation year) are logged and flagged for admin audit if needed
- Changes sync with backend and are logged with timestamp and user ID

6. Profile Completion Indicator (Optional)

- Show progress tracker for completion:
    - Profile photo

  - Bio
  - Graduation year
  - Payment method
- When 100% complete:
  - “Your profile is fully updated and visible to customers!”

E13. Enable Students to Contact Support via In-App FAQ and Escalation Form

Statement

Students can access an in-app support experience starting with an FAQ section. If their question remains unresolved, a support request form is enabled with a stated response time of 24 hours.

Acceptance Criteria

1. Access to Support Flow

- Students tap “Get Help” from:
  - Profile > Help & Support
  - Job details screen
  - Availability or Payment issues
  - Redirected to the FAQ screen

2. FAQ Section

- FAQ categories include:
  - Jobs & Assignments
  - Payments & Wallet
  - Availability
  - Onboarding
  - Referrals
- Features:
  - Search bar with keyword filtering
  - Expandable question-answer accordion layout
  - Scrollable and mobile-friendly design

3. Enable Support Form If No Answer Is Found

- After browsing FAQ, a sticky message or footer appears:
  - “Still need help? Tap here to contact our support team.”
- Tapping opens a support request form with:
  - Issue category (dropdown)
  - Short subject
  - Message description (required)
  - Optional screenshot or image upload
- A confirmation screen appears: “Your message has been sent! Our support team will respond within 24 hours.”

4. Response Time & User Expectations

- SLA is clearly stated in: Confirmation message after form submission

5. Admin Visibility & Workflow

- Submitted support messages are routed to:
  - Support email inbox TBD
- Message includes:
  - Student name + ID
  - Job ID (if relevant)
  - Issue category and message
  - Attached image (if provided)

6. Form & FAQ Availability

- Form is disabled by default and only shown after the student interacts with the FAQ screen
- Optional: Allow form to become visible only after:
  - At least one FAQ item is viewed
  - Selects: Still need help? Tap here to contact our support team.

E17. Enable Instant and Flexible Payout Options for Hands

Statement

Hands must be able to select between free, delayed payouts or fast, paid instant payouts after completing a job. The system must calculate availability, apply fees transparently, and process payment according to their choice.

Acceptance Criteria

1. Payout Options After Job Completion

- After a job is marked as "Completed" and accepted by the customer:
- Hand is shown a payout screen with options:
  - Standard Payout – free, 1–2 business days
  - Instant Payout – processed within minutes (fee applied)
  - Fee for instant payout is displayed transparently (This will be defined by stakeholders)

2. Default Preference Settings

- Hands can set their preferred payout method in their profile:
  - Always use standard (default)
  - Always use instant
  - Ask me each time

3. Payment Method Verification

- Instant payout only available if:
  - Bank account or debit card is linked and verified
  - Stripe (or equivalent) supports instant transfers to the selected method

4. Payout Processing Logic

- Standard payouts:
  - Triggered automatically within the business days defined by stakeholders
  - No cost to the Hand
- Instant payouts:

    - Available immediately once job is approved
    - Funds typically received within minutes
    - System deducts fee and confirms transfer

5. Receipts and Transparency

- Hand receives confirmation of:
    - Payout amount
    - Payout speed
    - Processing time and date
    - Fee deducted (if any)

6. Admin Controls & Monitoring

- Admin can:
    - Enable/disable instant payout feature
    - Set fee structure (flat rate or % of job)
    - View payout volume by type
    - Monitor payout failures or delays

E18. Enable Hands to Change Their Job Location and Work Across Regions

Statement: Hands must be able to update their working location within the app to reflect where they currently are, allowing them to receive and accept jobs outside of their originally assigned university region.

Acceptance Criteria

1. Location Update UI

- Hands can update their service location from:
- Profile > Settings > "Change My Location"
- Options:
- Search and select a new city
- Use current GPS to suggest nearby service area
- Select from predefined supported cities (linked to platform zones)

2. Location Assignment Rules

- When a new location is selected:
- Hand is now visible to job requests in that area
- Filters and availability sync with their new service zone
- Previous jobs, ratings, and history remain intact

3. Admin Panel Integration

- Admins can:
    - View all Hands active in each city
    - Track location changes per Hand
    - Filter new region data by active Hands and job fulfillment gaps

4. Seasonal Usage Considerations

- System may prompt Hands before known breaks:
  "Going home for the summer? Update your job location to keep earning in your area."

- Default return-to-university option at semester start

5. Impact on Notifications and Job Feed

- After location change:
  - Hands will receive job notifications only for their updated area
  - Job list will refresh based on new location filters
  - If availability is set, it remains active in new location unless reset

6. Data & Security

- Platform logs each location change for auditing and analytics
- Prevents frequent abuse (e.g., limit number of location changes in a short time)
- Keeps communication and matching regionally aligned for safety

E19. Enable Calendar Integration for Hands to Sync Accepted Jobs with Google or Apple Calendar

Statement

Hands must be able to sync accepted jobs to their personal digital calendar (Google or Apple), with proper time, location, and details.

Acceptance Criteria

1. Calendar Sync Activation

- Students can enable calendar integration in the app under:
  Profile > Settings > “Calendar Sync”
- Options:
  - Sync each accepted job manually
  - Enable automatic calendar sync for all future jobs

2. Job Event Creation

- When enabled, each accepted job creates a calendar event with:
  - Title: “Spare Hand Job – [Category]”
  - Start/End Time
  - Location
  - Job Notes (if available)
  - Link back to the app (if supported by calendar format)

3. Event Updates and Cancellations

- If the job is canceled or rescheduled:
  - System updates or removes the calendar event (if supported)
  - OR sends a notification to the student to update their calendar manually

4. Calendar Compatibility

- Support for:
  - Google Calendar (via OAuth and calendar API)
  - Apple Calendar (via iCal event file or native integration)
  - Outlook Calendar (optional, via iCal link)

5. Privacy and Permissions

- Students must explicitly grant permission for calendar access/sync

- No access is taken to read or modify events outside Spare Hand jobs
- Events can be removed anytime from settings or directly from the calendar

E20. Implement Gamification Tiers and XP System for Hands in the Student Mobile App

Statement

Hands must be able to earn XP and unlock visible performance-based tiers with defined perks based on their activity within the app.

Acceptance Criteria

1. Badge System Structure

- Tiers:
    - Green: 0–100 XP
    - Gold: 100–1000 XP
    - Platinum: 1000–3000 XP
    - Diamond: 3000+ XP
- Each tier unlocks specific benefits:
    - Priority Job Access (Tiered: 3, 2, 1)
    - Instant Cash from Wallet (Platinum, Diamond only)
    - Commission Rate Boost (Gold: 75%, Platinum: 75%, Diamond: 80%)
    - Quarterly Meet-ups (Diamond only)
- Badge is shown in the student profile and throughout the app (e.g., next to name in job history)

2. XP Earning System

- XP Actions:
    - +50 XP per 5-star rating
    - +50 XP per successful referral (new Hand must complete 1st job)
    - +25 XP per on-time job start
    - +25 XP per recurring job signup
    - +20 XP per before/after photo set
    - +10 XP per job accepted within 5 minutes
- XP history and breakdown view available in profile
- Progress bar shown to indicate how close they are to next badge tier

3. Visual Design and In-App Visibility

- Badges displayed in:
    - Profile
    - Job completion confirmation
    - Admin panel and reporting
- Tooltip explanation for how to advance tiers
- Real-time XP updates after each relevant action

4. Admin Controls

- Admins can:
    - Adjust XP values

    - Modify tier thresholds
    - Enable/disable rewards (e.g., instant cash, meet-ups)
    - Export XP reports per user

5. Notifications and Motivation

- In-app notification triggered when a student:
    - Levels up to a new badge tier
    - Is close to earning a new benefit
    - Unlocks a perk (e.g., “You've earned instant cash access!”)

5.3. Clients Web Site

E1. Allow Guest Navigation on Home Screen with Smart Sign-Up Trigger for Job Requests

Statement

Guest users can freely browse the homepage and explore job categories. If they try to request a job, they can see the form but when they provide the full info and request the job, they are shown a friendly sign-up prompt that preserves their progress after account creation.

Acceptance Criteria

1. Guest Browsing Permissions

- Unauthenticated (guest) users can:
    - View homepage
    - Scroll through featured job categories (e.g., Gardening, Moving)
    - See the "request a job" form according to the category they selected
    - Access how-it-works sections and FAQs
    - Click on “Learn More” or “Explore Services” without being forced to log in

2. Guest Access Restrictions

- Guest users cannot:
    - Submit a job request
    - View their job history
    - Access the “My Hands” section
- Attempting to perform a restricted action triggers the smart sign-up flow

3. Smart Sign-Up Trigger

- When a guest clicks “Request Job”:
    - A modal or interstitial page appears with the message: “Let’s create your free account so we can schedule your job request!”
    - Buttons:
    - Continue to Sign Up
    - Log In if you already have an account
- After signing up, the user is redirected back to the job request form with: Job category and progress retained (if applicable)

4. Visual & UX Considerations for Web

- Page headers must clearly show “Continue as Guest” or “Sign In” options

- Sticky nav bar includes sign-up and login CTA
- Sign-up flow uses a modal or soft redirect (not a hard page jump) to reduce abandonment
- Cookies or local storage used (if allowed) to save guest selections before sign-up

5. Fallback & Error Handling

- If guest tries to proceed and loses internet, system shows:
    - "We saved your spot! Please sign up when you're back online."
- If the sign-up process is abandoned mid-way:
    - User returns to homepage but their category interaction is saved for the session

E2. Customer Sign-Up with Phone Verification and Smart Referral Auto-Fill

Statement

New users signing up on the client website must complete a structured form that includes real-time validation, phone verification via OTP, and referral code auto-fill (if applicable), with clear confirmation and fallback options.

Acceptance Criteria

1. Responsive Web Sign-Up Form Structure

- The form includes the following fields:
    - First Name
    - Last Name
    - Email Address
    - Phone Number (with country code selector and input mask)
    - Password (with strength meter and show/hide toggle)
    - ZIP Code or Address
    - Referral Code (optional input; auto-filled via URL if applicable)
- All fields have real-time validation:
    - Email format
    - Required fields
    - Minimum password requirements
    - Invalid phone format detection

2. Phone Number Verification via OTP

- After entering phone number, user clicks "Verify"
- A 6-digit OTP is sent via SMS
- User enters code in verification modal
- Timer and "Resend Code" option available
- Must complete phone verification before account can be created
- Error handling includes:
    - Incorrect or expired code
    - Retry limit reached

3. Smart Referral Auto-Fill

- If a user visits the sign-up page via a link like: https://yourplatform.com/signup?ref=ZACH10
- The referral code field auto-fills with ZACH10
- Shows a friendly message: “Referral code applied! You may be eligible for a bonus.”
- If user clears or edits the code, shows tooltip: “Removing this code will remove referral benefits.”
- Referral is tracked in backend for reward purposes

4. Confirmation and Account Creation

- Once all fields are valid and phone is verified, user clicks “Create My Account”
- Backend processes data and stores referral and verification tokens
- On success: “Welcome to SpareHand! You're ready to request your first job.”

5. Fallback and Support Prompts

- If OTP service fails, show fallback message:
  - “Having trouble? Contact support or try again later.”
- Support contact link is included on verification and form error states

6. Security and Compliance

- Password is stored securely (hashed)
- Referral codes are validated server-side
- Phone verification token has expiration and is one-time use
- CAPTCHA or bot prevention may be used (optional for web)

E3. Display Personalized Job Suggestions on Web Based on Past Activity, Seasonal Trends, and Popular Jobs

Statement

The web homepage dynamically displays up to 4 personalized job category suggestions, using a combination of user history, active seasonal trends, and fallback “Popular Jobs” for new users all driving faster job initiation.

Acceptance Criteria

1. Homepage Suggestions Module Layout

- Section title: “Suggested for You” (changes based on user state)
- Shows 2–4 clickable tiles:
  - Each tile includes: Job category name, Icon or image, Optional subtext (e.g., “Requested 3 times”)
  - Desktop layout: responsive grid
  - Tablet/mobile: horizontally scrollable or stacked layout

2. Personalized Suggestion Logic

- For returning users with job history: Top suggestions are based on:
  - Most recently completed jobs
  - Most frequently requested categories
  - Successful completions only (not expired/canceled)
  - Clicking a tile: Opens the job request form with that category pre-selected

E4. Home Screen with Main Job Categories

Statement

The customer web homepage must present a simplified layout with large, easily recognizable job category buttons and a voice-activated search input supporting fast navigation and accessible job creation.

Acceptance Criteria

1. Homepage Layout and Category Tiles

- The homepage includes a prominent section near the top displaying: 6 job categories
- Each as a large, clickable tile with: Icon or image
    - Category label (e.g., "Yard Work", "Moving Help")
    - Optional short descriptor (e.g., “Heavy lifting, boxes, or furniture”)
- Responsive layout:
    - Desktop: Grid layout (2–3 tiles per row)
    - Tablet/Mobile: Scrollable horizontal list
- Clicking a tile takes the user directly to a pre-selected job request form

2. Text Search as Fallback

- Users may still type into the search bar
- Autocomplete suggestions are offered as they type (based on category database)

3. Admin Configuration Options

- Admins can configure:
    - Which job categories appear as homepage tiles
    - Order of display
    - Tooltips or short descriptors for each tile
    - Enable/disable voice input toggle (per platform or region)

E5. Display Dynamic Job Forms on Web Based on Selected Job Type

Statement

When a customer selects a job type on the web platform, the system dynamically loads and displays only the form fields relevant to that job category, with support for saved profile data and conditional visibility logic for custom charges (e.g., tool rental).

Acceptance Criteria

1. Dynamic Form Loading by Job Type

- Job form renders based on selected category:
    - Yard Work
    - Moving Help
    - House Cleaning
    - Pet Assistance, etc.
- Each job type has:
    - A distinct set of input fields (configured via admin panel)
    - Tool-related questions (if applicable)
    - Custom notes section

    - Optional pricing-relevant inputs

2. Address Field Flexibility
    - User sees:
        - Option 1: Use saved address from profile
        - Option 2: Enter a different address manually
    - Selecting saved address autofills the field and locks it unless "Edit" is clicked
    - Manual entry supports autocomplete integration (e.g., Google Places API) or ZIP validation
3. Conditional Field Logic
    - Some fields only appear based on prior inputs:
        - Example:
          "Do you have a lawn mower?" → If "No" → Cost auto-added to summary
    - Conditional fields support:
        - Binary logic (Yes/No)
        - Numeric logic (e.g., more than X hours triggers duration notice)
4. Field-Level Validation and Accessibility
    - Each field includes:
        - Live validation (e.g., required, input format)
        - Error messages inline
        - Tab-indexed inputs and screen reader labels (ARIA compliant)
5. Form Memory and Session Persistence
    - If user navigates away or refreshes:
        - Previous form inputs remain intact during the session
        - Saved in local storage or session cache
6. Form Summary and Confirmation Preview
    - Before submission:
        - User sees summary of all inputs, tool fees, and notes
        - All fields are editable prior to confirmation
        - Confirmation button disabled until all required fields are validated
7. Admin Configuration Support
    - Admin panel defines:
        - Field list per job type
        - Field order
        - Conditional pricing rules (e.g., tool costs)
        - Required vs. optional flags

E6. Optimize Real-Time Job Fulfillment for Customer Web Requests

Same description as Clients Mobile App epic

Optimize Real-Time Job Fulfillment for Multi-Hand Requests with Smart Reassignment and Scheduling Flexibility

E7. Display Real-Time Job Assignment and Status Updates in a Visual Tracking Interface for Web Customers

Same description as Clients Mobile App epic

Display Real-Time Job Assignment and Status Updates in a Visual Tracking Interface

E8. Enhance "My Hands" Section to Let Web Customers Rehire Previously Rated Students (with Availability Awareness)

Same description as Clients Mobile App epic

Enhance "My Hands" Section to Let Customers Rehire Previously Rated Students (with Availability Awareness)

E9. Ensure Secure and Transparent Job Payment Confirmation After Completion

Same description as Clients Mobile App epic

E10. Trigger Web Customer Rating Flow for the Hand After Job Payment Confirmation

Same description as Clients Mobile App epic

Trigger Customer Rating Flow for the Hand and App After Job Payment Confirmation

E11. Enable Tipping for Students After Job Completion for Customer Web Requests

Same description as Clients Mobile App epic

Enable In-App Tipping for Students After Job Completion with Percentage-Based Suggestions and Thank-You Acknowledgment

E12. Enable Smart Cross-Selling in Web Job Request Flow with “Your Neighbors Also Requested” Suggestions

Statement

During or immediately after the job request process on the website, customers are shown a cross-sell section with 1–3 suggested additional job categories based on geographic trends, seasonal logic, or past category bundles allowing for fast second-job creation. Acceptance **Criteria**

1. Module Placement & Trigger

- After completing a job request (on confirmation page) or just before confirming (right before final submission), a section appears:
- Title: “Your Neighbors Also Requested” or “Consider Adding These Services”
- Placement:
  - Below confirmation summary
  - Sidebar (desktop) or bottom (mobile)

2. Recommendation Logic

- Suggestions are based on:
  - Other jobs booked in the same ZIP code or neighborhood
  - Past users who requested similar primary categories (e.g., Moving → Yard Work)

  - Seasonal services if active (e.g., Leaf Cleanup, Holiday Decor)
- Backend supports weighted ranking by:
  - Popularity
  - Conversion rate
  - Time of year

3. Suggestion Tile Design

- Each tile includes:
  - Icon or image of the job category
  - Job name (e.g., "Yard Work")
- Optional short tagline:
  - "Popular this week in your area"
  - "Great add-on after Moving Help"
- CTA: "Request This" → Opens new job request screen with category pre-filled

4. Customer Flow

- Clicking a suggestion opens a new job request form, pre-loaded with:
  - Suggested category
  - Default time (next available or same day, configurable)
  - Keeps existing job in confirmation state, allowing parallel job creation

5. Admin Control and Analytics

- Admins can:
  - Enable/disable module globally or by category
  - Set logic priority (e.g., seasonal > regional > general)
  - Override suggestions with promotions
- View analytics in dashboard: Suggestions viewed, clicked, converted

6. Fallback & Edge Handling

- If no relevant data is found:
  - Show curated defaults: Cleaning Help, Pet Assistance
  - Or hide section entirely (configurable fallback)

7. Mobile & Desktop Responsive Design

- Responsive layout:
  - 1–2 suggestions stacked on mobile
  - 3 horizontally on desktop
- Accessible with full screen reader support

E14. Prompt Website Users to Download the Mobile App After Completing a Job

Statement

After completing a job on the website, customers are prompted to download the mobile app with a value-focused message and direct links to the App Store and Google Play.

Acceptance Criteria

1. Trigger Logic

- The prompt appears:

    - Immediately after a job is confirmed as completed via the website
    - OR on the confirmation screen after job submission, if not already using the app

2. Content and Messaging

- Messaging includes:
  "Enjoyed Spare Hand Students? Unlock even more features like live job tracking and push updates only in our mobile app. Download today :)"
- May include a short bullet list:
    - Live status updates
    - Faster job creation
    - Instant support access
    - Rehire your favorite Hands
    - Include App Store and Google Play badges/buttons with tracking links

3. Display Format

- Shown as a:
    - Modal popup or banner (depending on UX)
    - Optional post-job "Thank You" screen with embedded message
    - Dismissible, but only after 5 seconds (optional delay to ensure visibility)

4. User Experience Considerations

- If user has already downloaded the app (based on previous visit/session), suppress the prompt
- Prompt should not block any essential action post-job
- CTA buttons open directly to the appropriate store based on device

E15. Send Re-Engagement Notification to Inactive Website Users After Period of Inactivity

Statement

Inactive users who haven't visited the website in a configurable amount of time receive a friendly notification or banner encouraging them to return and book a job.

Acceptance Criteria

1. Inactivity Detection Logic

- User qualifies for re-engagement prompt if:
- No session or login activity detected in X days
- Has booked at least one job previously
- Inactivity threshold is configurable via admin settings

2. Prompt Behavior

- When the user returns to the website after inactivity:
- Show a banner or modal at top of homepage or dashboard
- Message example:
  "We haven't seen you in a while need a hand? We're here to help whenever you need us."
- CTA: "Book a Job Now" → leads to job creation flow

3. Dismiss and Frequency Controls

- Users can dismiss the message
- Message is shown once per qualifying inactivity window
- Not shown again until another X-day gap occurs

4. Optional Personalization

- If technically feasible, personalize the message:
  "Hey [First Name], we miss you around here!"
- Optionally highlight a category they've booked before

E16. Enable Customers to Apply Discount Coupons During Job Request or Payment Flow

Same description as Clients Mobile App epic

Enable In-App Tipping for Students After Job Completion with Percentage-Based Suggestions and Thank-You Acknowledgment

6. Rollout Plan

This rollout plan is designed to ensure the successful implementation and adoption of the Spare Hand Students platform across all four key components: Customer Mobile App, Student Mobile App (Hands), Admin Panel, and Website. The rollout will proceed in four structured phases, each with specific goals, stakeholders, and success indicators.

Phase 1: Backend Re-Architecture & Core Infrastructure Redesign

Objective: Lay a scalable, maintainable foundation for the Spare Hand Students platform by rebuilding the backend infrastructure to address current limitations and enable product flexibility for future growth.

Key Activities:

- Fully redesign and implement backend services with clean, modular architecture
- Implement new data models to support:
  - Real-time job routing
  - Multi-hand job coordination
  - Dynamic form rendering
  - Ratings, tipping, and promo logic
- Set up versioned APIs for mobile apps, admin panel, and web
- Integrate Stripe (or equivalent) for dynamic payments and instant payout support
- Ensure infrastructure supports future gamification, notifications engine, and reporting modules
- Conduct load testing, security reviews, and scalability simulations
- Code best practices, maintainability and scalability
- Modular Core Rest API
- Asynchronous message communication with support services

Success Criteria:

- New backend covers 100% of planned product logic as defined in PRD
- Clean separation of services (e.g., jobs, users, payments, scheduling)
- APIs fully tested and ready for frontend consumption
- System demonstrates performance and flexibility beyond current architecture
- Backend documentation delivered for internal alignment

Phase 2: Internal Validation & Admin Panel Enablement

Objective: Prepare the foundation by activating admin-facing tools, verifying configurations, and validating operational workflows internally before exposing changes to customers or Hands.

Key Activities:

- Launch core Admin Panel functionality:
  - Job form configuration
  - Schedule and availability management
  - Notifications and reporting tools
- Validation of routing logic and fallback rules
- Internal testing of job lifecycle (creation, assignment, completion, payment flow)
- Admin and Support user training and role-based access setup

Success Criteria:

- Admins/Support users can manage jobs, Hands, and campaigns without engineering support
- All job statuses and reporting views reflect real-time data
- No blocker-level bugs on internal tools
- Internal users successfully simulate job creation and resolution flows

Phase 3: Public Launch & App Store Distribution

Objective: Open the platform to existing and new audiences and drive adoption through in-app engagement and referral tools.

Key Activities:

- Launch new version of Customer and Student apps on App Store and Google Play
- Launch customer website with app download prompts and self-serve job creation
- Activate referral flows, seasonal campaigns, and notification nudges
- Begin proactive customer support readiness via Admin Panel tools
- Enable real-time tracking, tipping, recurring jobs, and ratings
  Boost cross selling features

Success Criteria:

- Minimum 3 new universities activated
- 2,000+ app downloads in first 30 days
- Increase in multi-category usage (cross-sell metric triggered)
- App rating ≥ 4.5 stars
- Support tickets managed via platform with ≤ 24h resolution SLA

Phase 4: Scale, Stabilize, and Strengthen Platform Services

Objective: Support the continued growth of Spare Hand Students across a growing number of universities, users, and job volume while ensuring the platform remains performant, reliable, and robust. This phase also lays the groundwork for feature maturity through enhanced reporting, extended service offerings, and chatbot.

Key Activities:

- Platform Scalability & Performance Optimization
- Regional & Service Expansion
- Strengthen Admin Visibility & Control
- Feedback Loop & Product Refinement
- Introduce structured user feedback collection mechanisms:
    - In-app feedback prompts for job flow, payments, and support
    - Admin view of feedback trends and pain points
- Adjust flows, incentives, and operational logic based on usage patterns
- Identify platform-specific friction for high-volume users and introduce micro-optimizations

Success Criteria:

- Platform can reliably handle 4x current volume without performance degradation
- 90% system uptime during peak hours across all services
- 95% of jobs processed without admin intervention
- Scalable service catalog and configuration tools live across all supported regions
- Admins fully equipped to track, launch, and optimize campaigns using real-time data
- Positive user feedback on platform responsiveness and reliability during scale-up

7. Solution Strategy

8.1 Backend Architecture

- **Modular Monolith:** Modular REST API to easily separate business domains into microservices when needed.
- **Support services:** To handle specific business flows and integrations, for example invoices, notifications, etc.

8.2 Backend Design Patterns

- Use of dependency injections across services and components for modularity.
- Adapter pattern for integrating external services as dependencies.

8.3 Integrations

- **Payment Gateway Integration:** Integrate a payment gateway to tokenize payment methods and process transactions securely without storing sensitive data.
- **Notifications:** The following services are required for notifications.
  - SendGrid (Email)
  - Firebase (push notifications)
  - Twilio (SMS)
- **Files Storage:** AWS S3 Integration to save Images, profile photos and files.
- **Retry mechanism:** Third party integrations must include retry processes to handle external services' issues.

8.4 Infrastructure

- **Cloud Provider:** AWS.
- **Scalability:** Managed using Docker containers on Kubernetes.
- **Database:** PostgreSQL on RDS, ensuring managed database services.
- **Secrets Manager:** Use AWS Secret Manager to store and manage secrets for the services, applications and environments.

8.5 Mobile Development

- Flutter.

8.6 Front-end Development

- Typescript - Angular.

8.7 Back-end API

- C#, .NET

8.8 Git Manager

- GitHub

8. Building Blocks

8.1 Scope and Context

The context block describes the Spare-Hands-Students system and its integrations with third party services, check the C4-context diagram.

System Name	Responsibility
Spare-Hand-Students	- *Customers can request help for a job or chore.* - *Hands (students) will be able to accept jobs and receive compensation for it once completed.* - *The Administration/Support panel will be used to configure the customers and hands applications.*

8.1.1 Services and Integration.

Service Name	Responsibility
Stripe	*Payment gateway.*
Twilio	*Third party service to send SMS notifications to the application users.*
SendGrid	*Third party service to send email notifications to the application users.*
Firebase	*Third party service to send Push notifications to the application users.*
AWS S3	*Objects and files storage.*
RabbitMQ	*Message broker to communicate, support services and main backend API.*
AWS Observability	*AWS Tool for observability and telemetry.*
SignalR	*.NET Core open-source library for real-time functionality.*

8.2 Level 1

This level describes the “Containers” that are included in Spare-Hands-Students Software System, see the C4-container diagram.

Container Name	Responsibility
SHS Customers Hybrid Mobile Application (Flutter)	*Hybrid application where customers can:* - *Navigate as a guest.* - *Register.* - *Create jobs/chores upon registration* - *Track the job stages.* - *Manage payment methods*

	Payment gateway integration will be used to tokenize payment methods. Plain credit card information will not be handled in backend.
SHS Students Hybrid Mobile Application (flutter)	*Hybrid application where customers can:* • *Navigate as a guest.* • *Register.* • *Create jobs/chores upon registration* • *Track the job stages.* • *Manage methods to receive payments.* *Payment gateway integration will be used to tokenize payment methods. Plain credit card information will not be handled in backend.*
Administration/Support Panel (React or Angular)	*Frontend web interface to config mobile applications:* • *Job Categories.* • *Real-Time job tracking and monitoring.* • *Hands ratings.* • *Manage Universities.*
Backend REST API (.NET)	*Backend service that will provide core functionality by modules:* • *Authentication and Authorization.* • *Users' registration.* • *Message broker Integration to communicate with support services.* • *Resources management (job categories, sub-categories, etc.)*
Invoices - Support Service (.NET)	*Service that will integrate the payment gateway (Stripe) and process payments.* ** Payment methods should be tokenized directly in mobile apps, to be saved in database; plain credit card information will not be handled in backend.*
Notifications - Support Service (.NET)	*Service to process, email, SMS, Push notifications.*
Assignments - Support Service (.NET)	*Service to process automatic/support hands assignment.*
Recurrent Tasks - Support Service (.NET)	*Service to perform scheduled and recurrent tasks like:*

8.3 Level 2

This level describes the "Components" or the modules that will be included in the Containers. Check C4-Component diagram. Components do not represent a single file in the source code, but a modular logic separation for the responsibilities within the Container.

8.3.1 Backend REST API

Component Name	Responsibility
Customer Controller	*Receive customers' requests like registration, profile, payment methods, etc.*
Customer Service	*Handle customers' requests, send to support services if needed using asynchronous messages.*
Customer Repository	*Perform CRUD operations at customers' requests.*
Students Controller	*Receive students' requests like registration, profile, payment methods, schedules, etc.*
Students Service	*Handle students' requests, send to support services if needed using asynchronous messages.*
Students Repository	*Perform CRUD operations at students' requests.*
Authentication Controller	*Receive authentication and authorization requests, sing-in and user permissions, refresh tokens, etc.*
Authentication Service	*Handle authentication and authorization requests, send to support services if needed (e.g. notifications) using asynchronous messages.*
Authentication Repository	*Perform CRUD operations for authentication and authorization requests.*
Job Categories Controller	*Receive job-categories, sub-categories requests like creating categories, modifying categories, adding rates, etc.*
Job Categories Service	*Handle job-categories and sub-categories requests, send to support services if needed using asynchronous messages.*
Job Categories Repository	*Perform CRUD operations at Customers' requests.*
Job Controller	*Receive jobs' requests like creating jobs, modifying jobs, hands assignment, etc.*
Job Service	*Handle jobs' requests, send to support services if needed using asynchronous messages.*
Job Repository	*Perform CRUD operations at jobs' requests.*
Payments Controller	*Receive payments' requests like job payment, dispute, request refund, etc.*
Payments Service	*Handle payments' requests, send to support services if needed using asynchronous messages.*
Payments Repository	*Perform CRUD operations at payments' requests.*

8.4 Communication

- Communication between the mobile applications and front-end administration/support panel, and back-end API will be via HTTPS, ensuring secure data transmission.

- Asynchronous messages must be used to send requests to the support services (invoices, notifications, scheduled tasks, hands assignments), through a message broker.

9. Cross-Cutting Concepts

9.1 Security

- Input validation and output encoding to prevent common web vulnerabilities like SQL Injection, cross-site scripting (XSS), and remote code execution.
- Use an ORM, parametrized queries or prepared statements to prevent SQL Injections attacks.
- Encrypt data both in transit (HTTPS) and at rest.
- Implement strong authentication mechanisms for example: password hashing, OAuth 2.0 and JWT tokens, also Including secure session tokens, cookie attributes, and session timeouts.
- Enforce secure password policies, like password complexity and expiration.
- Implement fine-grained authorization controls to ensure users have access only to the necessary resources and functionality.
- Implement appropriate error handling mechanism to prevent Information leakage.
- Log application events and errors to facilitate Incident response, troubleshooting, and security analysis.
- Store logs securely and protect log Integrity to prevent tampering or unauthorized modifications.
- Check compliance against OWASP Top 10 web.
- Check compliance against OWASP Top 10 Mobile.
- OWASP application Security Verification Level 1, Level 2, Level 3.
- Avoid sending Credit Card information to the backend, payment transactions should be sent directly to the Payment Gateway and transaction's results must be received in a payment processing system (sent by a payment gateway webhook).
- Avoid storing Credit Card Information to the database.

9.2 Scalability and Performance

- Utilize Docker and Kubernetes for containerization to facilitate scaling the application.
- Load balancing strategies to manage high traffic.

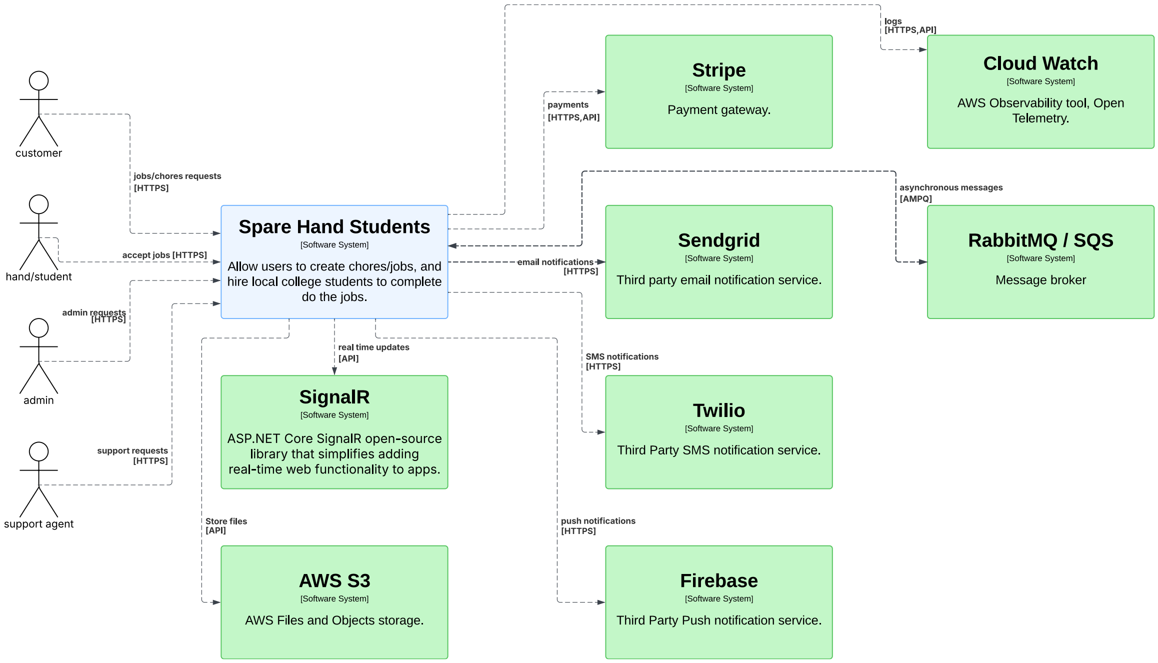
9.3 Data Management

- Implement database best practices in Postgres.
- Ensure data integrity and backups are available.

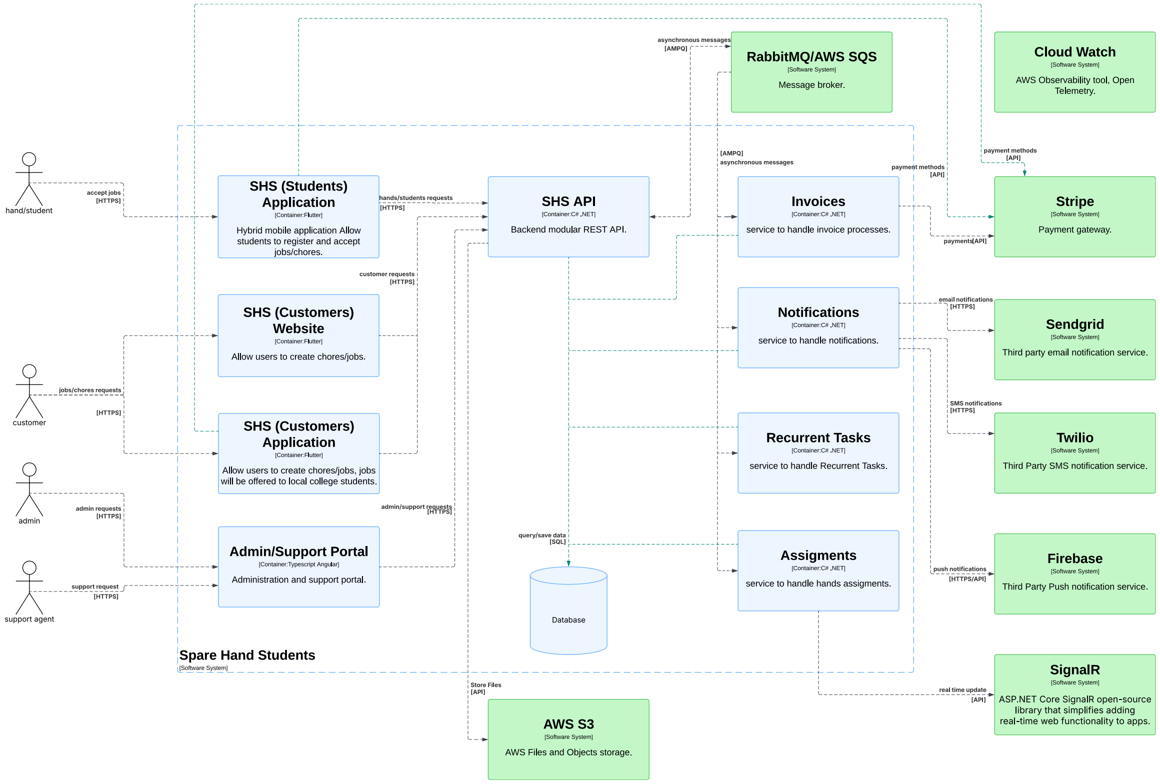
10. Architecture Design

10.1 Diagrams illustrating system architecture design.

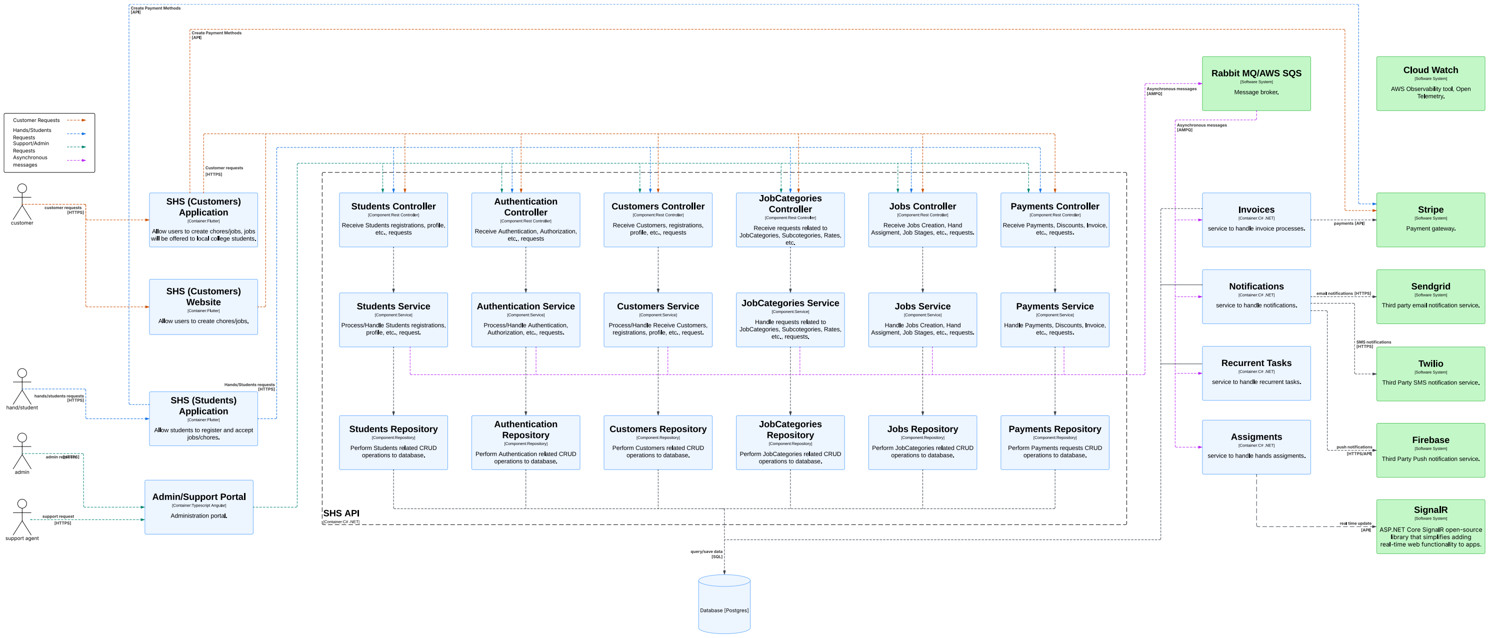
10.1.1 C4 – Context



10.1.2 C4 – Container



10.1.3 C4 – Component


Customer Requests
Hands/Students Requests
Support/Admin Requests
Asynchronous messages
Create Payment Methods [API]
Create Payment Methods [API]
Customer requests [HTTPS]
customer
customer requests [HTTPS]
SHS (Customers) Application
[Container:Flutter]
Allow users to create chores/jobs, jobs will be offered to local college students.
SHS (Customers) Website
[Container:Flutter]
Allow users to create chores/jobs.
hand/student
hands/students requests [HTTPS]
Hands/Students requests [HTTPS]
SHS (Students) Application
[Container:Flutter]
Allow students to register and accept jobs/chores.
admin
admin requests [HTTPS]
support agent
support request [HTTPS]
Admin/Support Portal
[Container:Typescript Angular]
Administration portal.
Students Controller
[Component:Rest Controller]
Receive Students registrations, profile, etc., requests
Authentication Controller
[Component:Rest Controller]
Receive Authentication, Authorization, etc., requests
Customers Controller
[Component:Rest Controller]
Receive Customers, registrations, profile, etc., requests
JobCategories Controller
[Component:Rest Controller]
Receive requests related to JobCategories, Subcotegories, Rates, etc.
Jobs Controller
[Component:Rest Controller]
Receive Jobs Creation, Hand Assigment, Job Stages, etc., requests.
Payments Controller
[Component:Rest Controller]
Receive Payments, Discounts, Invoice, etc., requests.
Students Service
[Component:Service]
Process/Handle Students registrations, profile, etc., request.
Authentication Service
[Component:Service]
Process/Handle Authentication, Authorization, etc., requests.
Customers Service
[Component:Service]
Process/Handle Receive Customers, registrations, profile, etc., request.
JobCategories Service
[Component:Service]
Handle requests related to JobCategories, Subcotegories, Rates, etc., requests.
Jobs Service
[Component:Service]
Handle Jobs Creation, Hand Assigment, Job Stages, etc., requests.
Payments Service
[Component:Service]
Handle Payments, Discounts, Invoice, etc., requests.
Students Repository
[Component:Repository]
Perform Students related CRUD operations to database.
Authentication Repository
[Component:Repository]
Perform Authentication related CRUD operations to database.
Customers Repository
[Component:Repository]
Perform Customers related CRUD operations to database.
JobCategories Repository
[Component:Repository]
Perform JobCategories related CRUD operations to database.
Jobs Repository
[Component:Repository]
Perform JobCategories related CRUD operations to database.
Payments Repository
[Component:Repository]
Perform Payments requests CRUD operations to database.
SHS API
[Container:C# .NET]
query/save data [SQL]
Database [Postgres]
Rabbit MQ/AWS SQS
[Software System]
Message broker.
Asynchronous messages [AMPQ]
Asynchronous messages [AMPQ]
Cloud Watch
[Software System]
AWS Observability tool, Open Telemetry.
Invoices
[Container:C# .NET]
service to handle invoice processes.
payments [API]
Stripe
[Software System]
Payment gateway.
Notifications
[Container:C# .NET]
service to handle notifications.
email notifications [HTTPS]
Sendgrid
[Software System]
Third party email notification service.
SMS notifications [HTTPS]
Twilio
[Software System]
Third Party SMS notification service.
Recurrent Tasks
[Container:C# .NET]
service to handle recurrent tasks.
Assigments
[Container:C# .NET]
service to handle hands assigments.
push notifications [HTTPS/API]
Firebase
[Software System]
Third Party Push notification service.
real time update [API]
SignalR
[Software System]
ASP.NET Core SignalR open-source library that simplifies adding real-time web functionality to apps.

11. Out of Scope

11.1 Social Sharing.

- **Priority:** Non Critical.
- **Feature Description:** Share your experience with your friends" which will allow the homeowner to leverage before/after photos that Hand took if allowed by customer and they can use them on a social media post showcasing the job done. Icons like Facebook, Nextdoor, Instagram, etc. will be presented so customers can with one click share their experience with us.

11.3 Chatbot Support.

- **Priority:** Non Critical.
- **Feature Description:** Chatbot as first contact on app support and able to generate job requests flows.

11.3 Subscription for Customers.

- **Priority:** Non Critical.
- **Feature Description:** Launch a diamond membership to access exclusive discounts, cashback, etc.

12. Non-Functional Requirements

These non-functional requirements (NFRs) ensure that the platform remains secure, scalable, and user-friendly, meeting the needs of customers, and businesses efficiently.

12.1 Reliability and Availability

- **System Scalability:** The platform should handle increasing numbers of users, and transactions without degradation in performance.
- **Fast Response Time:** All key interactions (job creations, profile updates, job category search, etc.) should respond in under 2 seconds.
- **High Availability:** Ensure 99.9% uptime to maintain customer engagement.

11.2 Security & Compliance

- **Data Encryption:** All sensitive data, including personal and payment details, must be encrypted in transit and at rest.
- **Secure Payments:** Tokenized Payment methods should be created from mobile applications directly, using the Payment Gateway Integration. Payment transactions should be done directly in link payment provided by de Payment Gateway or though the API using tokenized payment methods information only.
- **Role-Based Access Control (RBAC):** Ensure different user roles (customers, students, administrators, support agents) have appropriate data access.

12.3 Usability & Accessibility

- **Mobile-First Design:** The applications must be optimized for mobile devices while ensuring usability for desktop administration/support users.
- **Intuitive User Interface:** The interface must follow UX best practices to ensure easy navigation for customers.
- **Accessibility Compliance:** The app must be accessible to all users.

12.4 Reliability & Maintainability

- **Automated Backups:** The platform must perform daily automated backups to prevent data loss.
- **Error Logging & Monitoring:** Implement real-time monitoring and logging for errors, failures, and performance bottlenecks.

12.5 Integration & Compatibility

- **API-First Approach:** A core REST API will receive the users' requests; some processes or flows will be sent to the support backend services through a message broker, using asynchronous messages.
- **Third-Party Payment Integration:** The system must Integrate a payment gateway to ensure payment processing flexibility.

12.7 Legal & Compliance

- **Terms & Conditions:** Customers and students must explicitly accept terms & conditions. The details of terms and conditions will be defined by the stakeholders.

12.8 Reporting Capabilities

The Admin Panel must support access to actionable reports that help the operations, support, and product teams monitor platform performance, track key behaviors, and make data-driven decisions in real time.

Hands & Student Activity

- Report on student availability (semester-based or weekly)
- List of inactive students or students without availability set
- Report on tools owned vs. tools used during jobs
- Location change log for students (to track movement between cities or regions)
- Earnings per student by week or month

Jobs & Fulfillment

- Job status report:
  - Assigned
  - In Progress
  - Completed
  - Canceled
  - Expired (not accepted within the time limit)
- Jobs by location (city, university)
- Percentage of jobs completed on time
- Average time from job creation to acceptance
- Group job report:
  - Number of Hands requested vs. confirmed
  - Fallback logic triggered (yes/no)

Payments, Invoicing, and Tips

- Payment history by job and by student
- Payment status per job (paid, pending, disputed)
- Report on tips received by students
- Discount code usage and promotional credit application
- Uncollected revenue from jobs pending confirmation

Ratings & Feedback

- Ratings report for Hands:
  - Average rating per student
  - Highlighted comments
  - Flagged feedback for quality review
- Customer feedback on app experience
- Tip-related feedback if "thank you for the tip" logic is active

Promotions, Discounts, and Referrals

- Report on promo code redemptions by campaign
- Redemption rate of promotional credits (e.g., "app update credit")
- Referral performance:
  - Referred users pending job completion
  - Confirmed referrals
- In-app popup performance:
  - Views
  - Clicks
  - Conversions (e.g., app downloads, job creation, referral shares)

Notifications and Engagement

- Log of automated notifications sent (e.g., job reminder, reactivation prompt)
- Delivery and open status of push notifications
- App usage after key campaigns (e.g., update release or market expansion)
- Interaction with personalized modules like “Suggested for You” or seasonal campaigns

Platform Configuration and Admin Actions

- Log of actions performed by admin users (e.g., form edits, promotion creation)
- Access and role report for Admin Panel users
- Report on university creation/edit history and semester configuration activity

13. Risk and Compliance

This section outlines potential risks associated with the customers' applications and the necessary compliance considerations to ensure business continuity, regulatory adherence, and brand protection.

13.1 Key Risks:

- **Fraud Prevention:** Implementing secure payment processing and fraud detection measures with the Payment Gateway integration.
- **Platform Downtime:** Minimizing service disruptions through robust infrastructure and monitoring.
- **User Authentication Risks:** Strengthening login and password recovery mechanisms to prevent unauthorized access.
- **Security Risks:** Risks related to customers data protection, payment security, and unauthorized access must be mitigated with encryption and authentication mechanisms.
- **Integration:** Integration complexities with external SDKs.
- **Third Party Services:** Dependency on third-party services may introduce points of failure.