

DIGT2107: Practice of Software Development  
Project Iteration 2: Detailed User Stories, Requirements, and Initial Prototype  
OpenBid

Tyler, Mani, Yanness, Alaister

October 5, 2025

**Course:** DIGT2107 – Fall Term 2025    |    **Instructor:** Dr. May Haidar    |    **Team:** #1

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Functional and Non-Functional Requirements</b>	<b>2</b>
2.1	Functional Requirements . . . . .	2
2.2	Non-Functional Requirements . . . . .	2
<b>3</b>	<b>Use Cases</b>	<b>3</b>
3.1	Use Case Descriptions and UML diagrams . . . . .	3
3.2	UC-2: Browse & Bid (Job Bidder) . . . . .	4
3.3	UC-3: Award & Escrow (Job Poster) . . . . .	5
3.4	UC-4: Complete & Payout . . . . .	6
<b>4</b>	<b>Detailed User Stories</b>	<b>6</b>
<b>5</b>	<b>Initial Prototype Guidelines</b>	<b>9</b>
5.1	Prototype Features (Iteration 2 scope) . . . . .	9
5.2	Development Approach . . . . .	9
5.3	Task Completion & Collaboration . . . . .	9
<b>6</b>	<b>Documentation Updates</b>	<b>9</b>
6.1	System Architecture (Prototype) . . . . .	9
6.2	Planning Document and Log . . . . .	10
<b>7</b>	<b>Iteration Plan and Backlog</b>	<b>10</b>
7.1	Iteration Plan (Weeks 3–5) . . . . .	10
7.2	Updated Backlog (Top) . . . . .	10
<b>8</b>	<b>Submission Guidelines</b>	<b>11</b>

# 1 Introduction

**Project Name:** OpenBid

**Team Number:** #1

**Team Members:** Tyler, Mani, Yanness, Alaister

**Document Overview** This document outlines the deliverables for Iteration 2 of OpenBid: detailed user stories, functional and non-functional requirements, and an initial prototype plan. The aim is to establish a solid foundation for subsequent iterations by clarifying what the system should do and validating the core experience with a working prototype.

## 2 Functional and Non-Functional Requirements

### 2.1 Functional Requirements

1. **Account & Identity:** Users shall be able to sign up and sign in via OAuth (Google/Apple) and email+password. Mandatory KYC must be completed before posting or bidding.
2. **Two-Factor:** Users shall confirm sign-in with Duo 2FA for sensitive actions (login, payouts).
3. **Job Posting:** Posters shall create, update, and delete jobs with title, description, photos, budget (fixed/open), category, date, and location (map pin / address).
4. **Discovery:** Job bidders shall browse jobs on a map and list, filter by radius, category, budget range, and date.
5. **Bidding:** Job bidders shall place, edit, and withdraw bids (amount, note, ETA). Posters shall accept a winning bid.
6. **Payments (Escrow):** Posters shall fund an escrow hold on accept; on completion, funds shall capture and pay out to the job assignee (Stripe Connect).
7. **Messaging:** Posters and job bidders shall exchange messages within the job thread, with attachments and report/block controls.
8. **Reviews:** Both parties shall leave ratings and text reviews after completion.
9. **Safety Score:** The system shall compute a location safety score (0–100) and apply graduated friction (tips, daylight default, verified-only, or manual review).
10. **Admin:** Admins shall review reports, manage categories, and handle disputes via a basic dashboard.

### 2.2 Non-Functional Requirements

- **Performance:** Map/list browse and search shall return results in  $\leq 500$  ms p95 for target metro; prototype may use simplified filters.
- **Usability:** Mobile-first, accessible (WCAG 2.1 AA where practical); clear copy for KYC/2FA and escrow steps.
- **Security:** All traffic over TLS; short-lived JWTs; server-side validation; reCAPTCHA on signup/post/bid.

- **Scalability:** Stateless API (Node/Express) behind Firebase/Cloud Run; Firestore as primary data store; storage via Firebase Cloud Storage.
- **Reliability:** Error tracking via Sentry; observability with OpenTelemetry; automated CI checks on PRs.
- **Privacy:** Approximate location shown pre-accept; exact address released to the accepted job assignee only; phone masking post-accept.

### 3 Use Cases

#### 3.1 Use Case Descriptions and UML diagrams

##### UC-1: Manage Jobs (Job Poster)

**Actors:** Poster

**Description:** Create, update, and delete job postings with required details.

**Preconditions:** User is authenticated and KYC verified.

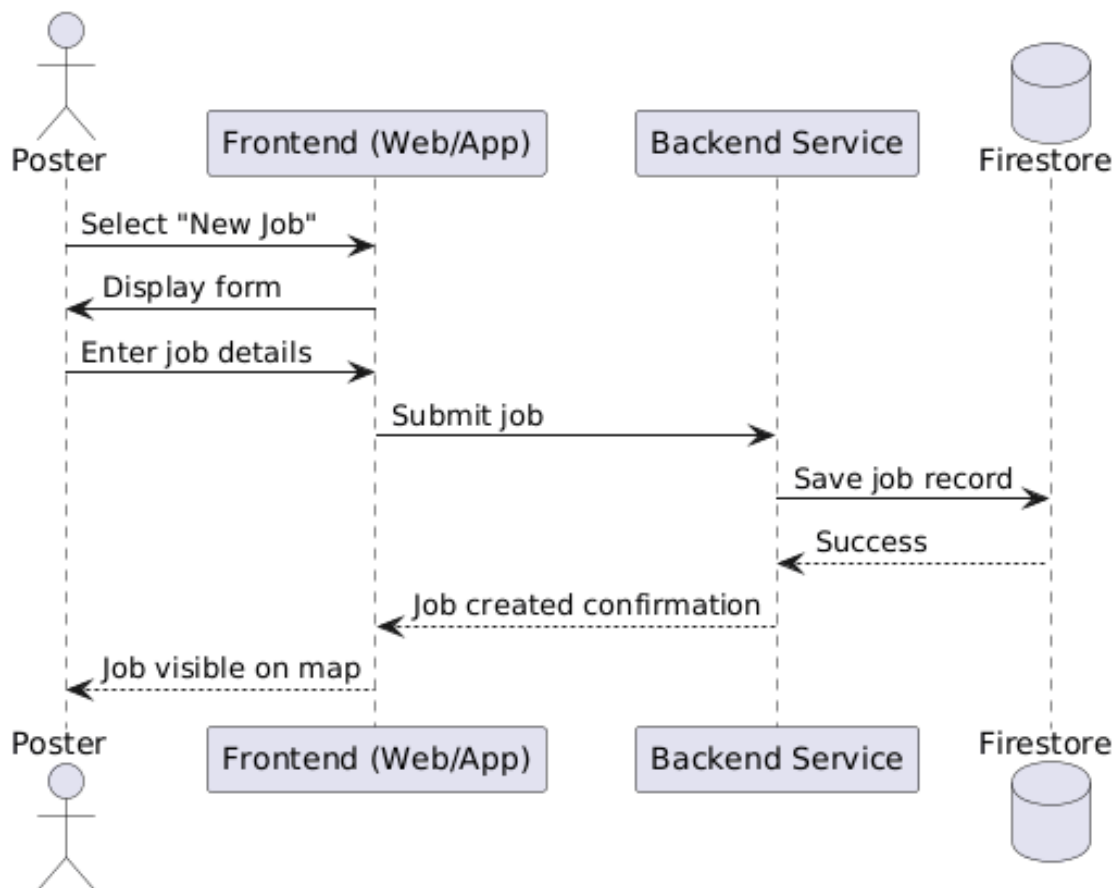


Figure 1: Use Case 1: Manage Jobs

### Main Flow:

1. Poster selects “New Job”.
2. System prompts for title, description, category, budget type/amount, date, and location.
3. Poster submits; system validates and saves to database; job appears on map/list.
4. Poster may edit or delete until a bid is accepted.

**Postconditions:** Job is persisted and visible to eligible bidders and browsers.

### 3.2 UC-2: Browse & Bid (Job Bidder)

**Actors:** Bidder

**Description:** Discover nearby jobs and place bids.

**Preconditions:** User is authenticated; KYC verified.

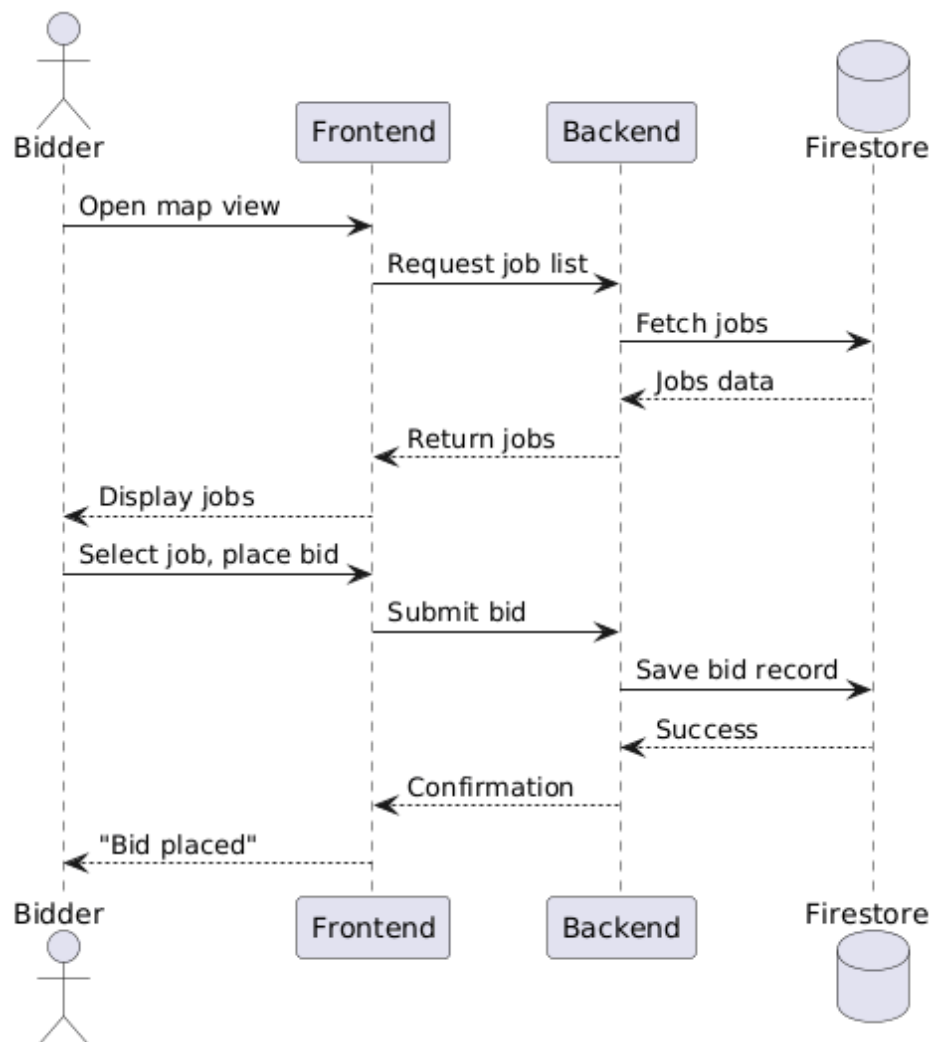


Figure 2: Use Case 2: Browse & Bid

**Main Flow:**

1. Bidder opens map view; system shows jobs within radius with filters.
2. Bidder opens a job, reviews details, and selects “Place Bid”.
3. Bidder enters amount, note, and ETA; system validates and saves bid.
4. Bidder can chat with the poster after submitting a bid.
5. Bidder can edit/withdraw bid before award.

**Postconditions:** Bid saved and visible to the job poster.

### 3.3 UC-3: Award & Escrow (Job Poster)

**Actors:** Poster

**Description:** Compare bids, accept a winner, and fund escrow.

**Preconditions:** Poster is authenticated; job has active bids; poster has a payment method saved.

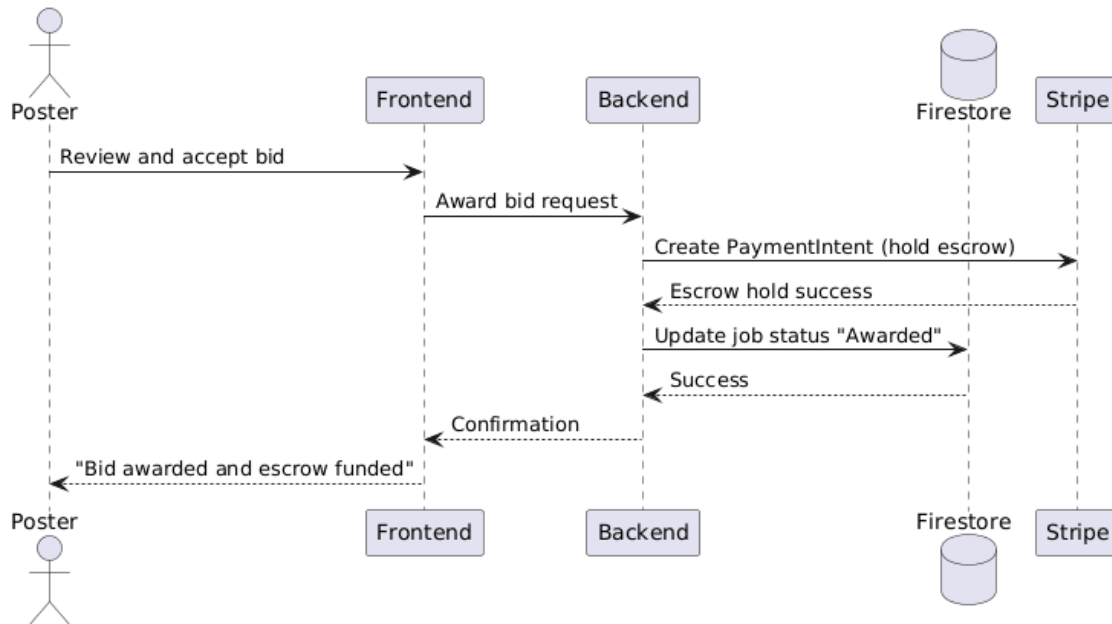


Figure 3: Use Case 3: Award & Escrow

**Main Flow:**

1. Poster reviews bids (amount, ETA, ratings).
2. Poster accepts a bid; system creates a Stripe PaymentIntent (hold).
3. On success, system marks job as awarded and opens thread with winner.

**Postconditions:** Funds on hold; winner can proceed.

### 3.4 UC-4: Complete & Payout

**Actors:** Job Winner, Poster

**Description:** Job Winner marks work done; poster confirms; funds captured and paid out.

**Preconditions:** Job awarded; escrow hold succeeded.

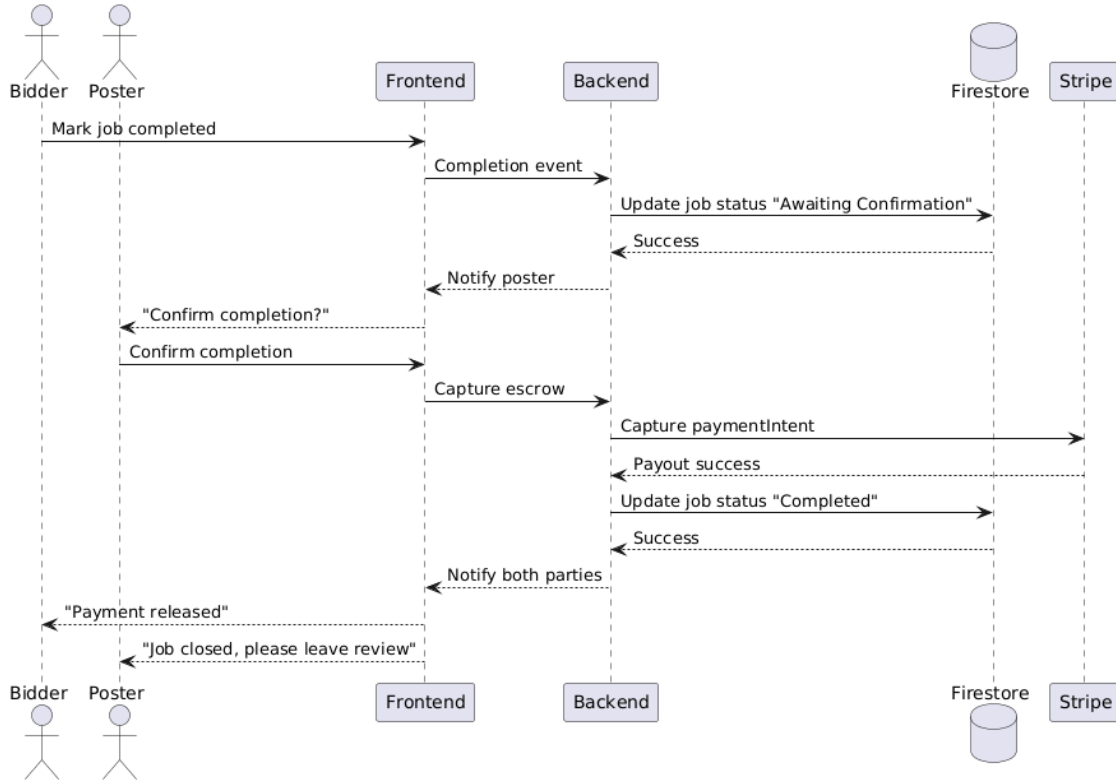


Figure 4: Use Case 4: Complete & Payout

#### Main Flow:

1. Job Winner marks job completed; system notifies poster.
2. Poster confirms completion (or opens dispute).
3. On confirmation, system captures funds and pays out via Stripe Connect.

**Postconditions:** Payment captured; both parties can review each other.

## 4 Detailed User Stories

### Job Posters

**US-1: Post a New Job (High, 3 pts)** As a **poster**, I want to add a job with title, description, photos, budget, date, and a map location so bidders can bid.

- **Acceptance:** Required fields validated; job saved; appears on map/list; poster can edit/delete until award.

**US-2: Accept a Bid & Fund Escrow (High, 5 pts)** As a **poster**, I want to accept a bid and put funds on hold so payment is guaranteed on completion.

- **Acceptance:** Stripe hold succeeds; job moves to **awarded**; chat thread opens with winner.

**US-3: Safety Score Guidance (Medium, 3 pts)** As a **poster**, I want safety guidance at post-time so I can choose safer options (e.g., daylight/public meetup).

- **Acceptance:** Score shown with tips; low-score flows add friction or require verification.

**US-4: Complete KYC (High, 3 pts)** As a **poster**, I want to verify my identity (document + selfie) so I can post jobs safely.

- **Acceptance:** Stripe Identity flow completes; app shows verified badge; posting features unlock.

**US-5: Duo 2FA (Medium, 2 pts)** As a **poster**, I want Duo phone confirmation so my account is protected during sign-in and payouts.

- **Acceptance:** 2FA challenge required; recovery path documented.

**US-6: Job Thread Messaging (Medium, 3 pts)** As a **poster**, I want a per-job chat so I can coordinate details with bidders.

- **Acceptance:** Send/receive text; attachments; report/block; notifications.

**US-7: Leave Reviews (Medium, 2 pts)** As a **poster**, I want to rate and review bidders after completion so future users can trust matches.

- **Acceptance:** 1–5 stars plus comment; appears on bidder profile; single review per job.

## Job Bidders

**US-8: Browse Nearby Jobs (High, 3 pts)** As a **bidder**, I want to see jobs near me with filters so I can quickly find relevant work.

- **Acceptance:** Map/list render; radius/category/budget/date filters; open job details view.

**US-9: Place a Bid (High, 5 pts)** As a **bidder**, I want to place a bid with amount, note, and ETA so the poster can evaluate me.

- **Acceptance:** Bid saved; poster notified; bidder can edit/withdraw prior to award.

**US-10: Complete KYC (High, 3 pts)** As a **bidder**, I want to verify my identity (document + selfie) so I can bid on jobs safely.

- **Acceptance:** Stripe Identity flow completes; app shows verified badge; bidding features unlock.

**US-11: Duo 2FA (Medium, 2 pts)** As a **bidder**, I want Duo phone confirmation so my account is protected during sign-in and payouts.

- **Acceptance:** 2FA challenge required; recovery path documented.

**US-12: Job Thread Messaging (Medium, 3 pts)** As a **bidder**, I want a per-job chat so I can coordinate details with posters.

- **Acceptance:** Send/receive text; attachments; report/block; notifications.

**US-13: Leave Reviews (Medium, 2 pts)** As a **bidder**, I want to rate and review posters after completion so future users can trust matches.

- **Acceptance:** 1–5 stars plus comment; appears on poster profile; single review per job.

## **Job Browsers (Not KYC-Verified Users)**

**US-14: Browse Jobs Read-Only (Medium, 2 pts)** As a **browser**, I want to explore posted jobs on a map in read-only mode, so I can understand the platform's services before committing.

- **Acceptance:** Map/list view available; limited job details visible; registration prompt when trying to post/bid.

**US-15: View Limited Job Details (Medium, 1 pt)** As a **browser**, I want to view limited job details (e.g., title, category, approximate location, budget range) without registering, so I see the platform's value.

- **Acceptance:** Basic job info visible; contact details and full descriptions hidden; registration required for full access.

**US-16: Registration Prompt (Medium, 1 pt)** As a **browser**, I want to be prompted to create an account if I try to post a job or place a bid, so I know registration is required.

- **Acceptance:** Clear registration modal appears; KYC/2FA requirements explained; seamless signup flow.

**US-17: KYC/2FA Instructions (Medium, 1 pt)** As a **newly registered user**, I want clear instructions that I must complete KYC verification and enable Duo 2FA before being allowed to post jobs or bid, so I understand the trust policy.

- **Acceptance:** Step-by-step guidance; progress indicators; help resources available.

**US-18: Access Support Resources (Low, 1 pt)** As a **browser**, I want access to support or FAQ pages, so I can ask questions or learn more before verifying my account.

- **Acceptance:** Support contact form; FAQ section; help documentation accessible without full registration.

## **Admin**

**US-19: Review Reports (High, 3 pts)** As an **admin**, I want to review user reports and disputes so I can maintain platform safety and integrity.

- **Acceptance:** Report dashboard; case management tools; resolution tracking.



**US-20: Manage Categories (Medium, 2 pts)** As an **admin**, I want to manage job categories so the platform stays organized and relevant.

- **Acceptance:** Category CRUD operations; validation; audit logging.

**US-21: Handle Disputes (High, 4 pts)** As an **admin**, I want to handle payment and job disputes so conflicts are resolved fairly.

- **Acceptance:** Dispute intake; evidence review; resolution workflow; communication tools.

**US-22: System Monitoring (Medium, 2 pts)** As an **admin**, I want to monitor system health and performance so I can ensure platform reliability.

- **Acceptance:** Dashboard with metrics; alert system; performance monitoring.

## 5 Initial Prototype Guidelines

### 5.1 Prototype Features (Iteration 2 scope)

- Minimal UI (React + SCSS) with routes: Home (Map/List), Job Details, Post Job, Bids.
- Core flows: create job, browse jobs, place bid, accept bid (*escrow call can be stubbed if needed*).
- Basic navigation and client-side validation.
- Firestore setup (collections: users, jobs, bids, messages); mock/safe seed data for demo.

### 5.2 Development Approach

- **Start Small:** Implement US-1, US-2, US-3 first; wire KYC screen stub.
- **Iterate Quickly:** Short PRs; CI checks; deploy preview to Firebase Hosting.
- **Modularity:** Split UI into reusable components; keep server routes RESTful.
- **Document Progress:** Track tasks in GitHub Projects; reference commits in issues.

### 5.3 Task Completion & Collaboration

- Break stories into tasks with estimates; label by feature and owner.
- Use a shared task board; commit frequently with meaningful messages.
- Pair programming rotation (round robin); driver/navigator swap daily; 1 reviewer outside the pair.

## 6 Documentation Updates

### 6.1 System Architecture (Prototype)

**Frontend:** React + SCSS (Firebase Hosting)

**Backend:** Node.js + Express (Firebase Functions or Cloud Run)

**Database:** Firebase Firestore (NoSQL, real-time)

**Storage:** Firebase Cloud Storage

**Auth:** Firebase Auth (OAuth, email+password) + Duo 2FA; KYC via Stripe Identity

**Payments:** Stripe Connect (escrow)

**Maps:** Google Maps JS, Places, Geocoding, Distance Matrix

## 6.2 Planning Document and Log

- **Changes from Iteration 1.1:** Committed to Firebase stack; clarified escrow flow; added explicit safety score gating.
- **Plan Delta:** Pulled reviews into Iteration 2 tail; background checks remain later.

## 7 Iteration Plan and Backlog

### 7.1 Iteration Plan (Weeks 3–5)

- **Week 3:** Finalize functional requirements; implement US-1 (post job) and US-8 (browse jobs); Firestore rules draft.
- **Week 4:** Implement US-9 (place bid) and US-2 (accept bid & escrow stub); basic messaging skeleton.
- **Week 5:** Review feedback; refine UX; stabilize prototype; add US-6/US-12 (chat MVP) and US-7/US-13 (reviews, stub).

### 7.2 Updated Backlog (Top)

ID	Story	Priority	Pts
US-1	Post a new job	High	3
US-2	Accept bid & fund escrow	High	5
US-8	Browse nearby jobs	High	3
US-9	Place a bid	High	5
US-4	Complete KYC (Poster)	High	3
US-10	Complete KYC (Bidder)	High	3
US-5	Duo 2FA (Poster)	Medium	2
US-11	Duo 2FA (Bidder)	Medium	2
US-3	Safety score guidance	Medium	3
US-6	Job thread messaging (Poster)	Medium	3
US-12	Job thread messaging (Bidder)	Medium	3
US-7	Leave reviews (Poster)	Medium	2
US-13	Leave reviews (Bidder)	Medium	2
US-14	Browse jobs read-only (Browser)	Medium	2
US-15	View limited job details (Browser)	Medium	1
US-16	Registration prompt (Browser)	Medium	1
US-17	KYC/2FA instructions (Browser)	Medium	1
US-18	Access support resources (Browser)	Low	1
US-19	Review reports (Admin)	High	3
US-20	Manage categories (Admin)	Medium	2
US-21	Handle disputes (Admin)	High	4

## 8 Submission Guidelines

- **GitHub Repository:** Push prototype code and documentation; tag release as ITR1.2.
- **eClass Submission:** Submit one PDF containing: Requirements; Use Case Diagram & Descriptions; Detailed User Stories; Initial Prototype Overview; Updated Plan and Backlog.

## Appendix A: API Inventory (Condensed)

Area	APIs / Notes
Auth/Identity	Firestore Auth (OAuth, email+password); Duo 2FA; Stripe Identity (KYC)
Payments	Stripe Connect (escrow), Webhooks, Radar
Maps/Geo	Google Maps JS; Places Autocomplete/Details; Geocoding; Distance Matrix
Notifications	Firestore Cloud Messaging; Twilio SMS + Proxy
Storage	Firestore Cloud Storage
Analytics/Ops	PostHog; Sentry; OpenTelemetry

## Appendix B: Firestore Schema (Indicative)

Listing 1: Firestore Collections & Example Documents

```

1 /users/{userId}
2 {
3   "name": "string",
4   "email": "string",
5   "phone": "string",
6   "avatarUrl": "string",
7   "isBidder": true,
8   "kycStatus": "pending" | "verified" | "failed",
9   "duoEnabled": true,
10  "createdAt": "timestamp",
11  "updatedAt": "timestamp"
12 }
13
14 /jobs/{jobId}
15 {
16   "posterId": "users/{userId}",
17   "title": "string",
18   "description": "string",
19   "category": "string",
20   "budgetType": "fixed" | "open",
21   "budgetAmount": 120.50,
22   "location": {"lat": 43.6426, "lng": -79.3871, "address": "string" },
23   "desiredDate": "timestamp",
24   "status": "open" | "awarded" | "in_progress" | "completed" | "cancelled",

```

```

25   "createdAt": "timestamp",
26   "updatedAt": "timestamp"
27 }
28
29 /bids/{bidId}
30 {
31   "jobId": "jobs/{jobId}",
32   "bidderId": "users/{userId}",
33   "amount": 95.00,
34   "note": "string",
35   "etaHours": 6,
36   "status": "active" | "declined" | "accepted" | "cancelled",
37   "createdAt": "timestamp"
38 }
39
40 /messages/{messageId}
41 {
42   "jobId": "jobs/{jobId}",
43   "senderId": "users/{userId}",
44   "body": "string",
45   "attachmentUrl": "string",
46   "createdAt": "timestamp"
47 }
48
49 /reviews/{reviewId}
50 {
51   "jobId": "jobs/{jobId}",
52   "raterId": "users/{userId}",
53   "rateeId": "users/{userId}",
54   "rating": 5,
55   "comment": "string",
56   "createdAt": "timestamp"
57 }

```