**Project Design Phase – Problem–Solution Fit Template**

**Date:**

25 June 2025

**Team ID:** **LTVIP2025TMID54354**

**Project Name:**

**HouseHunt: Finding Your Perfect Rental Home**

**Maximum Marks:**

2 Marks

**Problem – Solution Fit**

**Customer Segment / Target Group**

* Individuals or families looking to rent a house or apartment (Renters)
* Property owners who want to list their property (Owners)
* Admins managing user verification and platform compliance

**Problem(s) Identified**

1. **Unverified Listings**: Users often encounter fake or duplicate property listings on traditional platforms.
2. **Communication Gap**: Difficulty in reaching property owners and finalizing bookings.
3. **Scattered Information**: No centralized platform to filter based on specific needs (budget, location, size).
4. **Lack of Transparency**: Unclear booking status, hidden fees, and poor follow-up processes.
5. **No Real-Time Status Tracking**: Renters cannot track booking progress or availability in real time.

**Proposed Solution – HouseHunt**

1. **Verified Listings**: Admin-verified owners and properties reduce the risk of scams.
2. **User Roles and Communication**: Separate dashboards and in-app messaging for Renters, Owners, and Admins.
3. **Search Filters**: Filter listings based on rent, location, property type, and availability.
4. **Booking and Tracking**: Seamless booking workflow with real-time status updates.
5. **Admin Governance**: Central control to approve listings and monitor platform activity.

**Why This Solution Works**

* **Behavior Match**: Users are already using online platforms (like OLX, MagicBricks), so HouseHunt builds on this habit — but adds safety, speed, and simplicity.
* **Urgency Alignment**: Users need homes fast. HouseHunt simplifies the process with verified and available listings.
* **Trust Building**: Owner verification, admin oversight, and in-app negotiation tools help build trust between users.
* **Convenience First**: End-to-end process handled on one platform — from search to lease finalization.

**Customer Validation (Optional)**

* **Surveys** conducted showed 80% of renters faced challenges with fake listings or unresponsive owners.
* **Feedback** from early testers praised the clean UI, role-specific dashboards, and smooth booking flow.

**Conclusion**

HouseHunt addresses a clear pain point in the rental market by offering a solution that closely fits the users’ behaviors and expectations. It not only solves the technical and logistical issues but does so in a way that is user-friendly, trustworthy, and scalable.  
**Proposed Solution Template**

| **S.No.** | **Parameter** | **Description** |
| --- | --- | --- |
| 1 | **Problem Statement** (Problem to be solved) | Finding a rental property is often time-consuming, unsecure, and inconvenient due to unverified listings, lack of communication between renters and owners, and no unified platform for property booking and status tracking. |
| 2 | **Idea / Solution Description** | HouseHunt is a full-stack web application that connects renters and property owners on a single platform. It allows owners to list their properties, renters to search and book homes, and admins to manage approvals and verify data. Features include advanced filtering, booking status tracking, in-app communication, and admin moderation to ensure a secure and smooth rental experience. |
| 3 | **Novelty / Uniqueness** | Unlike traditional listing platforms, HouseHunt adds a **trust layer** through owner verification and admin approvals. The entire rental cycle — from property discovery to booking confirmation — happens on a **single platform** with **real-time updates** and **role-based dashboards**. |
| 4 | **Social Impact / Customer Satisfaction** | HouseHunt makes renting accessible, safe, and transparent for students, working professionals, and families. It reduces rental fraud, saves time, and builds trust by simplifying the search and booking process. Increased satisfaction through verified listings and clear communication helps users feel more secure. |
| 5 | **Business Model (Revenue Model)** | Revenue can be generated through: • Featured property listings for owners • Subscription plans for owners with multiple properties • Commission-based model on confirmed bookings • Potential ad partnerships with moving, legal, or furnishing services |
| 6 | **Scalability of the Solution** | The solution is scalable across cities and property types (rooms, hostels, houses, apartments). With cloud hosting and modular APIs, it can serve thousands of users. Future versions may include a mobile app, payment gateway integration, AI-based rent prediction, and map-based search functionality. |

**Solution Architecture – HouseHunt: Finding Your Perfect Rental Home**

**1. Objective**

To design a robust, scalable, and secure solution that addresses the fragmented and unreliable experience of renting properties online by integrating user management, verified property listings, booking systems, and role-based operations in a unified platform.

**2. Technology Stack**

| **Layer** | **Technologies Used** |
| --- | --- |
| **Frontend** | React.js, Material UI, Bootstrap |
| **Backend** | Node.js, Express.js |
| **Database** | MongoDB (NoSQL) |
| **APIs** | RESTful APIs (Axios-based communication) |
| **Authentication** | JWT (JSON Web Tokens), bcrypt |
| **State Management** | React Context API / Redux (optional) |
| **Deployment** | Render / Vercel (Frontend)  Railway / MongoDB Atlas (Backend & DB) |

**3. System Overview (Architecture Breakdown)**

**Client-Side (Frontend)**

* Built using **React.js** for modular component-based architecture
* Features separate dashboards for **Renter**, **Owner**, and **Admin**
* Uses **Axios** for seamless API communication with the backend
* **Material UI** & **Bootstrap** used for responsive and interactive UI design

**Server-Side (Backend)**

* Built using **Express.js** and **Node.js**
* Handles business logic: authentication, CRUD operations, booking logic, admin control
* Role-based authorization checks implemented for each API route

**Database (MongoDB)**

* NoSQL schema with collections:
  + **Users** (Renter, Owner, Admin roles)
  + **Properties** (linked to owner IDs)
  + **Bookings** (linked to properties and users)
* Indexed for fast querying (e.g., filters, availability)