**LIVEIN BUDDY – A SMART LIFESTYLE-BASED LIVE-IN PARTNER MATCHING APP**

Submitted By:  
Rajshree Tanmayee Swain, Satyakam Acharya, Arbind Mishra, Banamali Nayak  
  
Course: Java Project  
C.V. Raman Global University  
  
Date of Submission: 14th March 2025

# Table of Contents

1. 1. Abstract
2. 2. Problem Statement
3. 3. Objectives
4. 4. System Architecture
5. 5. Module-wise Explanation

* a. User Registration and Onboarding Module
* b. Lifestyle & Preferences Assessment Module
* c. Matching & Recommendation Module
* d. Community Building Module
* e. Feedback & Rating System
* f. Support & Conflict Resolution Module

1. 6. Workflow / Flowchart

# 1. Abstract

The Livein Buddy project represents an innovative solution tailored for modern urban life, where co-living has become a practical and economic necessity. It goes beyond traditional rental apps by integrating lifestyle matching features to create a comfortable and emotionally enriching shared living environment. The platform focuses on bringing together like-minded individuals who align not just in rent-sharing needs, but also in habits, values, and personality traits. This ensures a more fulfilling and peaceful co-living experience, transforming how people perceive shared housing.

In an age where people are increasingly mobile and living costs continue to rise, finding a compatible living partner is critical. The application is designed to address these evolving needs with smart algorithms and user-friendly features that reduce friction and promote harmony. By prioritizing psychological comfort alongside logistical convenience, Livein Buddy helps users build lasting bonds and foster a sense of community within their living space. The app is not just a tool—it's a lifestyle enabler.

# 2. Problem Statement

The traditional concept of renting a space with someone is often driven solely by financial considerations. Unfortunately, this often results in mismatched roommates who differ drastically in lifestyle habits, personal values, and social behavior. This mismatch often leads to conflicts, mental stress, and a diminished quality of life. People may face issues like incompatible sleep cycles, hygiene standards, or noise preferences, which, although seemingly small, have a significant impact on day-to-day life and emotional well-being.

The core problem is not just about affordability—it is about compatibility. A house is not merely a space; it’s an emotional environment. A peaceful home is essential for productivity, happiness, and overall wellness. Livein Buddy seeks to address this gap by enabling users to find co-living partners who complement their lifestyle, thereby enhancing the overall co-living experience. It redefines the way we choose roommates—not just by price or availability, but by alignment in personality and living style.

# 3. Objectives

The primary goal of Livein Buddy is to promote harmonious shared living by providing a digital solution that helps users find compatible roommates. This is achieved by leveraging user preferences, habits, and behavioral patterns to match individuals with similar lifestyles. The app is designed to ease the emotional discomfort that often arises in mismatched living arrangements, replacing it with compatibility-driven partnerships.

Moreover, the application intends to build an ecosystem where users feel more like part of a community rather than temporary tenants. It provides an opportunity for people to build trust and friendships, making shared living a positive experience. Another objective is to create a sustainable model that learns from user feedback, continuously improving its matching algorithm and community features. In essence, Livein Buddy aims to transform shared accommodation into a shared life experience.

# 4. System Architecture

The architecture of Livein Buddy follows a layered, modular approach designed for scalability and seamless user interaction. The architecture consists of three key layers: the user interface layer, application logic layer, and the database layer. The front-end interface is intuitively designed to facilitate easy navigation, profile setup, and preference input. Behind the scenes, the application logic processes user data to generate compatibility scores, match recommendations, and interaction flows.

At the core lies a robust database that holds user profiles, behavioral inputs, feedback records, and activity logs. The architecture ensures data privacy and optimized performance while allowing real-time updates and feedback loops. Furthermore, this structure is designed to accommodate future upgrades such as AI-based behavioral learning, emotion recognition, and social analytics. This layered model helps ensure high system reliability, fast response times, and a personalized user experience.

# 5. Module-wise Explanation

## a. User Registration and Onboarding Module

The registration module initiates the user journey, capturing essential user details such as name, contact information, age, occupation, and city. This is followed by a lifestyle questionnaire that captures deeper behavioral data like daily routines, social habits, noise tolerance, sleeping schedules, and cleanliness preferences. This data forms the backbone for compatibility calculations and match recommendations.

The onboarding process is designed to be smooth, interactive, and non-intrusive, helping users feel at ease while inputting personal lifestyle information. Visual cues and intuitive prompts guide the user through the process. By gamifying the onboarding experience slightly, it ensures a high user completion rate and accuracy in lifestyle assessment. This module also includes optional verification tools to enhance trust and safety among users.

## b. Lifestyle & Preferences Assessment Module

This module serves as the brain of the application. It processes detailed inputs gathered during onboarding and translates them into behavioral metrics. These metrics include sociability scores, introvert-extrovert balance, cleanliness ratings, noise sensitivity index, and time-of-day activity levels. These factors are crucial in ensuring a harmonious living arrangement.

The data is processed through a matching engine that generates a personalized Compatibility Index Score for each user. This score is dynamic and adjusts with user behavior over time. The assessment module also allows users to update their preferences periodically, ensuring continued relevance and accuracy. It is through this module that Livein Buddy differentiates itself from traditional rental platforms by focusing on emotional and psychological compatibility.

## c. Matching & Recommendation Module

This module employs a smart recommendation engine that uses weighted scoring algorithms to compare users and generate match suggestions. Users receive a curated list of potential roommates along with a percentage-based compatibility score. The recommendations are prioritized not just by preference match, but also by shared values and mutual interests.

The suggestions come with detailed compatibility summaries highlighting common interests, lifestyle overlaps, and potential areas of conflict. This transparency helps users make informed decisions and reduces the chances of unpleasant surprises post-move-in. The system also supports filtering options based on city, budget, profession, and availability dates, making it a versatile roommate search tool.

## d. Community Building Module

The community module is designed to foster interaction even before co-living begins. It includes features like group discussions, interest-based communities, and social bonding activities. Users can join discussion forums or groups based on hobbies, professions, or lifestyle niches such as pet lovers, early risers, or minimalists.

This module encourages early bonding and builds a sense of trust among users. It plays a critical role in transitioning the app from being just a matching platform to becoming a community-driven ecosystem. The shared community experience often leads to better co-living outcomes, as it creates a foundation of familiarity and mutual respect.

## e. Feedback & Rating System

Post-move-in, users are encouraged to rate their living experiences and provide detailed feedback on their roommates. These ratings help future users assess potential partners more accurately. It also helps identify recurring issues or red flags that may not be evident during initial matching.

The system continuously learns from user feedback, refining its matching algorithms and increasing overall reliability. Moreover, it creates a reputation score for users, motivating everyone to maintain positive behavior. Transparency in feedback builds trust, and the presence of a structured rating mechanism empowers users to make safe and informed choices.

## f. Support & Conflict Resolution Module

Recognizing that disagreements can still arise, this module offers proactive and reactive tools for conflict resolution. Users can raise issues confidentially and seek mediation through the app’s support system. The platform offers automated tips, community moderators, and even access to professional counselors if needed.

This ensures that users feel safe and supported, even during difficult co-living situations. By offering a conflict resolution mechanism, Livein Buddy takes full responsibility for the user experience—not just during matching, but throughout the co-living journey. This approach increases user retention and overall satisfaction.

# 6. Workflow / Flowchart

The flow of the application begins with user registration and moves through data processing, matching, interaction, and feedback. Every phase is designed for minimal friction and maximum personalization.

After onboarding, the matching engine analyzes lifestyle metrics and provides curated recommendations. Users interact with potential roommates through messaging features and shared interest communities. Once they decide to co-live, the post-move-in modules like feedback, ratings, and conflict resolution come into play. The flowchart provides a visual overview of this entire journey, reflecting a seamless and user-friendly process.