

# FYP Documentation

*by Ahmad Rafi*

---

**Submission date:** 22-Jul-2025 05:05AM (UTC+0500)

**Submission ID:** 2718719239

**File name:** Bixel\_Documentation.docx (943.57K)

**Word count:** 5029

**Character count:** 30589

**Brixel Living**



**1 Bachelor of Science in Computer**

**Science Session (2021-2025 Fall)**

**Submitted By:**

**Anas Mufti**

**Ahmad Rafi**

**SUPERVISOR**  
**SHOAIB SALEEM**

**1 Lecturer**  
School of Computer Science

---

**SCHOOL OF COMPUTER SCIENCE**  
**MINHAJ UNIVERSITY, LAHORE**

## **Brixel Living**

 *Submitted in Partial Fulfillment*  
of the Requirements  
for the Degree of

### **Bachelor of Computer Science**

at the

**Minhaj University, Lahore**

by

Anas Mufti 2021f-mulbscs-090

Ahmad Rafi 2021f-mulbscs 067

 <b>Moaiib Saleem</b> Supervisor School of Computer Science	<b>Dr. Gulzar Ahmad</b> Head School of Computer Science
--	---

## **DECLARATION**

It is declared that this is an original piece of my work, except where otherwise acknowledged in text and references. This work has not been submitted in any form for another degree or diploma at any university or other institution for tertiary education and shall not be submitted by me in future for obtaining any degree from this or any other University or Institution. I am the responsible if I do not meet the deadline.

Anas Mufti                              Signature: \_\_\_\_\_  
2021f-mulbscs-090

Ahmad Rafi                              Signature: \_\_\_\_\_  
2021f-mulbscs-067



<sup>1</sup>  
**DEDICATION**

**To**

***My Mother***

*Who always wished or prayed for my success, for supporting and encouraging  
me to believe in myself, who always sacrificed so much for me.*

***My Father***

*For earning an honest living for me, for supporting and encouraging me to  
believe in myself.*

***My Teacher***

*For their support and wishes.*

## **CERTIFICATE OF APPROVAL**

It is certified that the project titled "Brixel Living" carried out by Anas Mufti, 2021f-mulbscs-090, Ahmad Rafi , 2021f-mulbscs-067 under the supervision of Dr Imran Akhter, Minhaj University Lahore, is fully adequate, in scope and in quality, as a final year project for the degree of BS of Computer Science.

Supervisor:

-----  
Dr Imran Akhter  
Lecturer  
School of Computer Science  
Minhaj University Lahore

Internal Examiner 1:

-----  
School of Computer Science  
Minhaj University Lahore

Internal Examiner 2:

-----  
School of Computer Science  
Minhaj University Lahore

FYP Coordinator:

-----  
Shoaib Saleem  
Lecturer  
School of Computer Science  
Minhaj University Lahore

Head of Department:

-----  
Dr. Gulzar Ahmad  
School of Computer Science  
Minhaj University Lahore

## **ACKNOWLEDGEMENT**

First of all, I am thankful to Allah Almighty, the most beneficial and merciful who gave me the courage and guidance to complete my project. I offer my special praise for our beloved Holy Prophet Hazrat Muhammad (S.A.W) who is the symbol of knowledge and guidance of humanity as a whole. I like to express my special thanks of gratitude to my supervisor Dr Imran Akhter for his support. His steadfast support of this project was greatly needed and deeply appreciated. Last but not least, no acknowledgement could ever express our obligations to our loving parents for their love, affection, amiable, advices and prayers for my success.

## ABSTRACT

Nowadays real estate industry has numerous issues, including scamming property ads, lack of properties verification, and mismanagement of properties. **Brixel Living** is designed on modern and effective principals to enhance transparency, efficiency, and accessibility in property transactions using features like **360-degree virtual tours, verified listings, and AI-driven recommendations**. It provides a platform for seamless interaction between property owners, buyers/sellers reducing the requirements of traditional methods like physical visit of property and also prevent from fraud in property transactions.

Agile methodology besides iterative development and continuous feedback was utilized for the development of our platform. Flutter and Firebase were utilized for the frontend and back-end respectively for the development of major features consisting of **360-degree tour, digital rental agreements, investment opportunities for small-scale investors, and an online auction system** for competitive bidding.

To conclude **Brixel Living revolutionize real estate industry by providing a seamless and secure platform for interaction and property transactions with its efficient and reliable features facilitating buyers/sellers and building trust additionally being easy to access.**

## 2 Table of Contents

<b>CHAPTER 1 INTRODUCTION .....</b>	<b>1</b>
<b>1.1. Background.....</b>	<b>2</b>
<b>1.2. Goals and Objectives.....</b>	<b>2</b>
<b>1.3. Gap Analysis .....</b>	<b>3</b>
<b>1.4. Project Plan.....</b>	<b>4</b>
<b>1.4.1. Work Breakdown Structure.....</b>	<b>4</b>
<b>1.4.2. Gantt Chart.....</b>	<b>6</b>
<b>1.4.3. Team Members.....</b>	<b>7</b>
<b>1.5. Report Outline.....</b>	<b>7</b>
<b>CHAPTER 2 .....</b>	<b>9</b>
<b>2.1. Purpose .....</b>	<b>10</b>
<b>2.1.1. Document Conventions.....</b>	<b>10</b>
<b>2.1.2. Intended Audience .....</b>	<b>11</b>
<b>2.2. Overall Description.....</b>	<b>11</b>
<b>2.2.1. Service Perspective.....</b>	<b>11</b>
<b>2.2.2. Service Function.....</b>	<b>11</b>
<b>2.2.3. Product Functions.....</b>	<b>11</b>
<b>5</b>	
<b>2.2.4. Assumptions and Dependencies.....</b>	<b>13</b>
<b>2.3. External Interface Requirements .....</b>	<b>13</b>
<b>2.3.1. User Interfaces .....</b>	<b>13</b>
<b>2.4. System Features .....</b>	<b>14</b>
<b>2.4.1. User Login.....</b>	<b>14</b>
<b>2.4.2. Logout.....</b>	<b>15</b>
<b>2.4.2. Edit Settings.....</b>	<b>16</b>

2.4.3.	View Properties and Virtual Tours .....	16
2.4.4.	Participate in Property Auctions .....	17
2.5.	Other Nonfunctional Requirements .....	18
2.5.1.	Performance Requirements.....	18
2.5.2.	Safety Requirements .....	19
2.5.3.	Software Quality Attributes .....	19
2.6.	Other Requirements.....	20
	<b>CHAPTER 3.....</b>	22
3.1.	Use Case Model.....	23
3.3.	Fully Dressed Use Case.....	24
•	User Registration.....	24
•	Edit Setting.....	27
•	Browse Property Listing.....	29
•	View 360° Virtual Tour .....	31
•	AI-Based Property Recommendations.....	33
•	Participate in Property Auctions .....	35
•	Upload Property Listing.....	38
•	Generate Digital Rental Agreements.....	41
•	Admin Moderation .....	44

## List of Tables

Table 1.1 Team Member .....	7
Table 3.1 Use Case<01> .....	26
Table 3.2 Use Case<02> .....	28
Table 3.3 Use Case<03> .....	30
Table 3.4 Use Case<04> .....	32
Table 3.5 Use Case<05> .....	34
Table 3.6 Use Case<06> .....	37
Table 3.6 Use Case<07> .....	40
Table 3.6 Use Case<08> .....	43
Table 3.6 Use Case<09> .....	45

## Table of Figures

Figure 1.1 Word Breakdown Structure .....	5
Figure 1.2 Gantt Chart.....	6
Figure 3.1 Use Case Model .....	23
Figure 3.2 Login Use Case.....	25
Figure 3.3 Setting Use Case .....	27
Figure 3.4 Browse Property Listing Use Case.....	29
Figure 3.5 360-Degree Virtual Tour Use Case .....	31
Figure 3.6 Property Recommendation Use Case .....	33
Figure 3.7 Property Auction Use Case.....	36
Figure 3.8 Upload Property Listing Use Case.....	39
Figure 3.9 Digital Rental Agreement Use Case.....	42
Figure 3.10 Admin Moderation Use Case.....	44

**CHAPTER 1**  
**INTRODUCTION**

Key points like **background, goals, gap analysis** etc. of our project are defined in this chapter of documentation. This section provides a comprehensive understanding of the project's motivation, challenges, and implementation approach.

## 1.1. Background

Real estate industry has improved a lot with the time, but issues like **fraudulent property listings, lack of verified transactions, and inefficient property search processes** still exist due to lack of interaction, transparency between dealing individuals. Buyers, sellers and investors faced and even still facing difficulties due to the old methods like physical visits of site, idol paperwork and delay in transaction which are still present in current industry.

**Brixel Living** is a **real estate platform** designed to **eliminate these issues** in property transactions by the utilization of advance and effective features like **360-degree virtual property tours, verified property listings, AI-powered recommendations, digital rental agreements, and online property auctions**. By integrating these features, Brixel Living aims to **simplify property transactions while minimizing fraud and inefficiencies**.

## 1.2. Goals and Objectives

The primary objective of Brixel Living is to streamline the real estate industry by implementing technology-driven solutions. Key objectives include:

- Eliminating the need of site visit physically with the help of 360-degree virtual tour.
- Finding desired and effective properties with the help of AI-driven property recommendation.

- Ensuring authenticity of listing through property verification system.
- Shared ownership model for the facilitation of small investors.
- Improved accessibility for buyers with the utilization of online auction bidding.
- Enhancement of security and improving efficiency with digital rental agreements instead of traditional paperwork.
- These objectives aim to improve the real estate transaction experience for buyers, sellers, and investors.

### **1.3. Gap Analysis**

Despite digital advancements, **existing real estate platforms have limitations** that hinder smooth transactions:

- **Limited virtual viewing experiences**—most platforms rely on static images rather than immersive virtual tours.
- **Lack of verified property listings**—many buyers face fraud due to unverified sellers.
- **No seamless integration for digital rental agreements**—paper-based agreements cause delays.
- **Minimal investment opportunities for small-scale investors**, making real estate investing difficult for individuals with low capital.
- **Absence of efficient property auctioning platforms**, limiting buyer access to competitive deals.
- **Brixel Living** addresses these gaps by **integrating technology-driven solutions, providing a secure and user-friendly platform for real estate transactions**.

## 1.4. Project Plan

The **development and execution of Brixel Living** involve multiple phases, ensuring a well-structured implementation.

### Key Features:

- Users can **create profiles** and set preferences for personalized property recommendations.
- Real estate agents and sellers can **upload verified property listings** with **360-degree tours**.
- Buyers and investors can **search, filter, and explore properties** through an **AI-driven recommendation engine**.
- **Digital rental agreements** enable hassle-free leasing processes.
- **Property auctions** allow buyers to place bids on listed properties.

### 8 1.4.1. Work Breakdown Structure

The **Work Breakdown Structure (WBS)** for Brixel Living involves several stages:

#### 1. Concept Phase:

- Market research, target audience analysis, problem identification.

#### 2. Design Phase:

- UI/UX design, database structuring, architectural planning.

#### 3. Development Phase:

- Frontend and backend development, API integrations, AI-based recommendations.

#### 4. Testing Phase:

9

- Functional testing, security testing, and user acceptance testing.

#### 5. Deployment Phase:

- Platform launch, user onboarding, and performance monitoring.

This structured approach ensures a **systematic and efficient development process**.

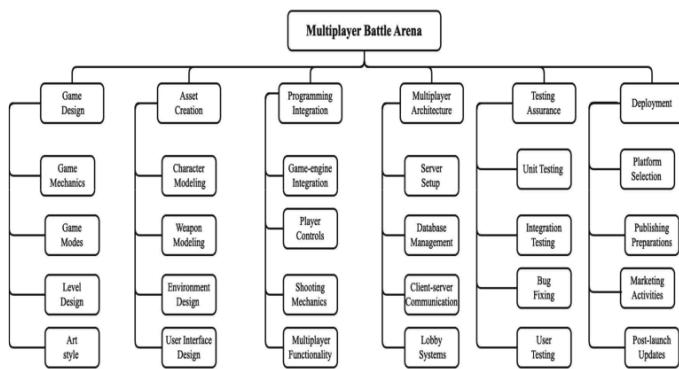


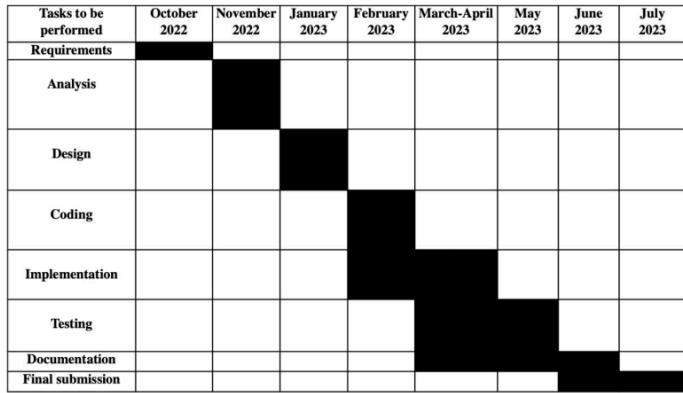
Figure 1.1 Word Breakdown Structure

#### **1.4.2. Gantt Chart**

A **Gantt Chart** is used to manage the timeline of development tasks efficiently. It outlines the duration of each phase, dependencies, and milestones in the project.

Key components include:

- System architecture design
- Backend and frontend development
- Integration of AI-powered recommendations
- Testing and debugging
- Deployment and final evaluation



*Figure 1.2 Gantt Chart*

### 1.4.3. Team Members

Each member has some specific skills which they used efficiently in project. A table is given to show team members names, their skills and their task description.

**Table 1.1 Team Member**

Name	Skills	Description
Anas Mufti	Back-end Developers	Design The Back End
Ahmad Rafi	Front-end Developer	Design The User Interface

**Table 1.1 Team Members**

### 1.5. Report Outline

In the digital age, the real estate industry is evolving rapidly, yet many property seekers, investors, and agents still face challenges such as fraudulent listings, lack of verified data, time-consuming physical visits, and inefficient property management. Traditional platforms often fail to offer transparency and intelligent personalization, leaving users overwhelmed or uncertain during the property search process. **Brixel Living** is a mobile-based real estate application developed to solve these modern problems using innovative technologies. Built with **Flutter** for cross-platform functionality and **Firebase** as its real-time backend, Brixel Living combines core features such as **AI-driven property recommendations, 360-degree virtual tours, digital rental agreements, and real-time property auctions**. The goal is to streamline property transactions, reduce dependency on physical interactions, and make real estate more secure, accessible, and intelligent for

everyone whether you're a buyer, seller, tenant, or investor.

- **Chapter 1 - Introduction:** Project background, objectives, gap analysis, and project planning.
- **Chapter 2 - System Requirements & Specification:** Technical specifications, system architecture, and feature breakdown.
- **Chapter 3 - Use Case Analysis:** User interactions, use case models, and functional flow.
- **Chapter 4 - System Design:** Architecture diagrams, database schemas, and component structures.
- **Chapter 5 - Implementation:** Coding standards, technology stack, and development workflow.
- **Chapter 6 - Testing & Evaluation:** Functional, security, and performance testing details.
- **Chapter 7 - Conclusion & Future Enhancements:** Overview of whole project and possible additions for the future.

**CHAPTER 2**

**SYSTEM REQUIREMENTS AND SPECIFICATION**

## **2.1. Purpose**

Brixel Living aims to revolutionize real estate industry by providing a reliable platform for properties transactions with efficiency and complete transparency excluding the fraudulent activities creating a **user-friendly** experience for buyers, sellers, renters, and investors by offering advance modules like **360-degree virtual property tours, verified property listings, AI-based property recommendations, digital rental agreements, and an online property auction system**. System requirements and specifications to ensure smooth and seamless flow of platform are defined in this chapter.

### **2.1.1. Document Conventions**

To maintain consistency and readability, the following formatting conventions are used in this document:

- **Font Style:** Times New Roman
- **Font Size:** 12pt (Body Text), 13pt (Headings)
- **Line Spacing:** 2.0
- **Headings:** Bold
- **Tables & Figures:** Numbered and labeled for reference

This document is intended for **developers, testers, stakeholders, and end-users** who wish to understand the purpose, features, and requirements of **Brixel Living**.

### **2.1.2. Intended Audience**

The Bixel Living app is designed for property buyers, renters, sellers, agents, and small-scale investors who want a secure and efficient platform for real estate transactions. It also serves administrators managing listings and developers or students interested in mobile app technologies like Flutter and Firebase.

## **2.2. Overall Description**

### **2.2.1. Service Perspective**

Bixel Living is developed to improve the real estate experience by providing a **user-friendly interface, 360-degree virtual tours, and secure property transactions**. The application is designed using **React Native** for cross-platform support and utilizes **Firebase** for backend services.

### **2.2.2. Service Function**

The main functions of this project include **property listing, 360-degree virtual tours, post-verification by admins, rental agreements with smart contracts, and collective investments**. These features enhance the real estate experience by ensuring trust, transparency, and accessibility.

### **2.2.3. Product Functions**

#### **User Management**

- User Registration & Login
- 360-Degree Virtual Tour

- Post Verification by Admins
- Property Listings (Buy/Sell/Rent/Hostels)
- Smart Contracts for Rentals
- Collective Investments (Shareholders)
- Contractor Hiring
- Property Auctions

**User Functions**

- Register/Login
- Upload Property Listings
- View and Participate in Property Auctions
- Browse 360-Degree Virtual Tours
- Edit Profile and Settings

**• Admin Functions**

- Approve/Reject Property Listings
- Manage Users and Listings
- Oversee Investment and Rental Agreements

• Minimum **2GB RAM** required for optimal performance

• Requires **stable internet connection**

- Compatible with **Android and iOS**

#### **Tools & Technologies Used**

- Flutter (Frontend)
- Firebase (Backend & Database)
- Google Maps API (Location Services)

#### **2.2.4. Assumptions and Dependencies**

- The user must have **internet access** to use the platform.
- The admin will **verify property listings** before they go live.
- 1 • Users must provide **authentic documents** for verification.

### **2.3. External Interface Requirements**

#### **2.3.1. User Interfaces**

Bixel Living provides a modern, user-friendly interface where users can:

- Explore 360-degree property tours
- Chat with property owners or agents
- Participate in property auctions
- Browse available rental and investment opportunities

#### **Hardware Requirements**

- Processor: Minimum Quad-core
- Storage: At least 1GB free space
- Memory: 4GB RAM

## **2.4. System Features**

### **2.4.1. User Login**

#### **1. Description and Priority**

The user login system is a core functionality of Brixel Living and holds the highest priority. When a user first opens the app, they are required to log in or register an account. This ensures secure access and personalized experience throughout the platform.

#### **2. Stimulus/Response Sequences**

- The user launches the Brixel Living application.
- The login/registration screen is displayed.
- Upon successful login, the user is redirected to the home dashboard, where personalized property recommendations and listings are shown.
- If the login fails, the user receives an error message and is prompted to retry.

#### **3. Functional Requirements**

- User must be able to register via email/password or third-party authentication (e.g., Google).

- User credentials must be securely stored using Firebase Authentication.
- The system must validate login credentials and restrict access if invalid.
- Logged-in users gain access to personalized features like virtual tours, saved listings, and auction participation.

#### **1.4.2. Logout**

##### **1. Description and Priority**

Logout functionality allows users to end their session securely. It holds medium priority but is essential for session management and privacy.

##### **2. Stimulus/Response Sequences**

- From the user dashboard or profile, the user selects the "Logout" option.
- A confirmation prompt may be displayed.
- On confirmation, the user session is terminated, and they are redirected to the login screen.

##### **3. Functional Requirements**

- Users should be able to log out at any time.
- All session tokens must be cleared.
- User data should not persist locally post-logout (except for explicitly saved preferences like language).

## **2.4.2. Edit Settings**

### **1. Description and Priority**

This feature allows users to customize app settings based on personal preferences.

While not critical to core functionality, it enhances user experience and thus holds medium priority.

### **2. Stimulus/Response Sequences**

- The user selects the “Settings” option from the navigation drawer or profile section.
- Users can modify preferences like notification settings, preferred property types, display settings, and language.
- Changes are saved and reflected immediately.

### **3. Functional Requirements**

- Users must be logged in to access and save settings.
- Settings must be stored in real-time using Firebase Firestore or Realtime Database.
- The app must retrieve saved settings on subsequent logins.

## **2.4.3. View Properties and Virtual Tours**

### **1. Description and Priority**

This feature allows users to view verified property listings, complete with 360-degree virtual tours. It is a high-priority feature as it directly supports the platform's goal of improving transparency and reducing the need for physical visits.

## 2. Stimulus/Response Sequences

- User browses property listings from the dashboard or search screen.
- Upon selecting a property, details including images, location, agent contact, and a virtual tour (if available) are displayed.
- The user can navigate the 360° view to explore the property virtually.

### Functional Requirements

- Must fetch property data from the database in real time.
- Virtual tour must be viewable using embedded panoramic viewer.
- Only verified listings should show 360° tour availability.

#### 2.4.4. Participate in Property Auctions

##### Description and Priority

The auction module allows users to bid on listed properties in real time. This is a high-priority feature for enabling competitive and accessible buying options.

#### **Stimulus/Response Sequences**

- The user navigates to the "Auctions" section.
- Available properties for bidding are displayed.
- User selects a property and views auction details and timer.
- User places a bid and receives live updates.

#### **Functional Requirements**

- Users must be logged in to participate in auctions.
- Real-time bidding must be synchronized across all users using Firebase Realtime Database or Cloud Functions.
- System must validate bids and announce winning bids after the auction ends.

1

### **2.5. Other Nonfunctional Requirements**

#### **2.5.1. Performance Requirements**

- The application must be compatible with Android and iOS platforms, as it is developed using Flutter.
- Users must have a stable internet connection to ensure smooth interaction with real-time features such as virtual tours, live auctions, and cloud-synced data.
- The app should maintain a lightweight design to ensure fast performance and low storage usage, even on low-end mobile devices.
- Firebase backend services must handle concurrent user interactions efficiently, particularly during property auctions or data retrieval.

### **2.5.2. Safety Requirements**

- Only authenticated users can access the platform features. Firebase Authentication will manage user login and session control.
- All sensitive user data (e.g., personal info, transaction records) must be stored in encrypted form using Firebase Security Rules and encryption standards.
- Role-based access control will ensure that unauthorized users cannot manipulate listings or access admin functionalities.

### **2.5.3. Software Quality Attributes**

#### **Privacy and Security**

- Privacy and user data protection are top priorities. Bixel Living ensures end-to-end encryption for sensitive operations and applies strict access control policies.

#### **Availability**

- The platform should be available 24/7. Users must be able to access their accounts and services from any location at any time with internet access.

#### **Integrity**

- Data consistency and accuracy are ensured across all modules. Any updates to listings, user profiles, or transaction statuses must reflect instantly and

reliably.

#### **Correctness**

- All modules, including virtual tours, recommendations, contracts, and auctions, must perform their intended function accurately without bugs or logic errors.

#### **Reliability**

- The app should function consistently without crashes or data loss, even under peak loads. It must handle edge cases like sudden disconnects gracefully.

#### **Security**

- User data is securely managed with encrypted storage, authenticated access, and secure communication between client and server. Security audits and Firebase rules will prevent unauthorized data access.

#### **Ease of Use**

- Bixel Living is designed with a user-friendly interface, offering intuitive navigation, smooth interactions, and guided onboarding for new users.

### **2.6. Other Requirements**

The Bixel Living mobile application must store and manage comprehensive user information to deliver a personalized and secure experience. Each user's data is saved in the Firebase database and is associated with a **unique user ID** generated at the time of registration through Firebase Authentication.

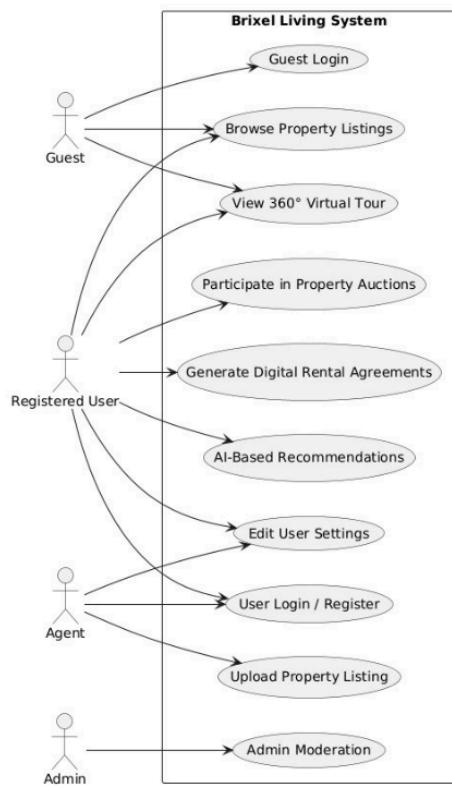
The system must:

- Store personal details such as name, email, contact number, and user role (e.g., buyer, seller, agent, admin).
- Maintain user preferences for property types, location interests, and investment profiles.
- Track user activity such as favorite properties, viewed listings, auction participation, and virtual tour interactions.
- Securely store and manage digital rental agreements and investment participation records under each user's account.
- Enable real-time syncing of user settings and preferences across multiple devices using Firebase Firestore.
- Ensure that only authenticated users can retrieve or modify their data, in compliance with security rules and access control policies.

**CHAPTER 3**  
**USECASE ANALYSIS**

### 3.1. Use Case Model

The **Brixel Living** use case model demonstrates how different user roles (Guest, Registered User, Agent, Admin) interact with the system to perform real estate-related actions. The app provides features like guest browsing, user registration/login, verified property listings, 360-degree virtual tours, property auctions, AI-based recommendations, and digital rental agreements.

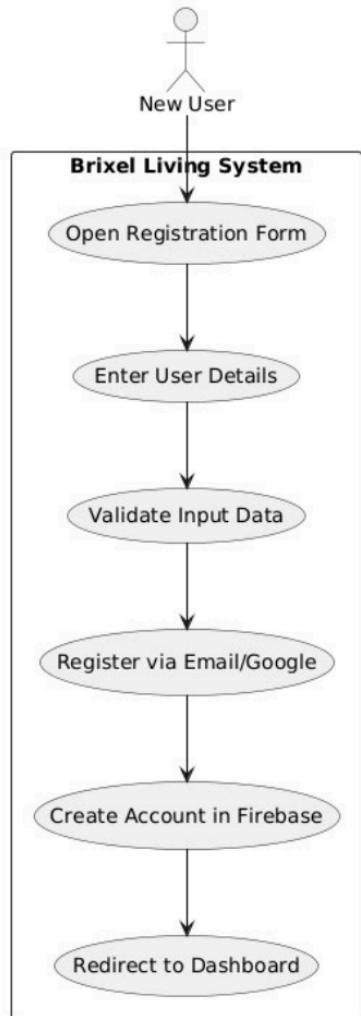


*Figure 3.1 Use Case Model*

### 3.3. Fully Dressed Use Case

- **User Registration**

The **User Registration** feature allows new users to create an account on the Brixel Living mobile platform in order to gain full access to its functionalities. Upon launching the app, users are presented with the option to register using their email and password or through third-party authentication providers such as Google. The registration process includes input fields for essential information like full name, email, phone number, and password. The system validates all inputs for correctness and uniqueness before creating an account. Once successfully registered, the user's information is securely stored in Firebase Authentication and Firestore, with a unique user ID assigned. This registration enables access to advanced features like uploading property listings, participating in auctions, receiving AI-driven recommendations, and signing digital rental agreements. To ensure security and privacy, user data is encrypted and access is protected through authenticated sessions.



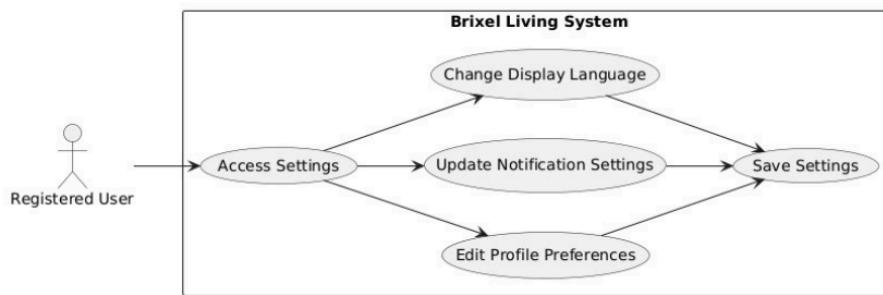
*Figure 3.2 Login Use Case*

Fully Dressed Use Case		
•	<b>Use Case Name</b>	User Login
•	<b>Scope</b>	Login
•	<b>Level</b>	User goal
•	<b>Actor</b>	User, Server
•	<b>Stakeholder</b>	User, Server
•	<b>Pre-Condition</b>	Internet Connection
•	<b>Post-Condition</b>	Access to App
•	<b>Success Scenario</b>	Successfully Login

*Table 3.1 Login Use Case*

- **Edit Setting**

The Edit Settings feature in the Brixel Living mobile app allows registered users to personalize their application experience based on their preferences. Once logged in, users can access the settings section from the side navigation menu or their profile page. Within the settings, they can update various preferences such as notification preferences, preferred property types (e.g., apartments, houses, plots), budget range, preferred locations, display language, and app theme. These settings enhance the user experience by enabling AI-based property recommendations tailored to their needs. Any changes made are stored in the Firebase database and synced in real time to ensure they persist across sessions and devices. To maintain security, this feature is only accessible to authenticated users. The system validates each input before saving and provides feedback for successful updates or errors. Overall, the Edit Settings use case enhances usability, customization, and user engagement across the platform.



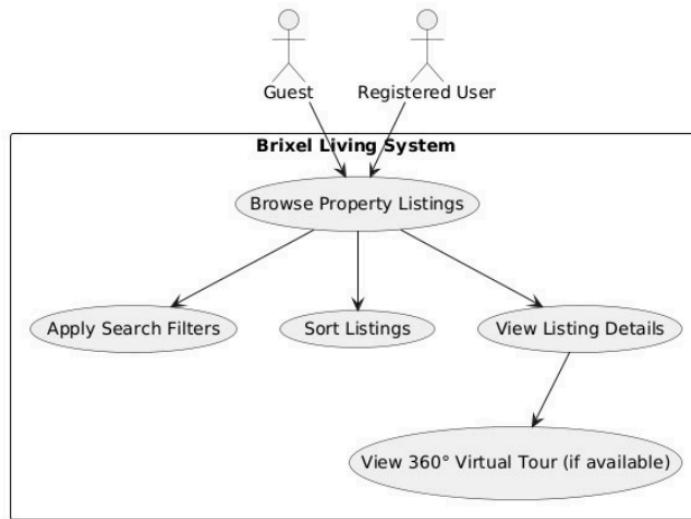
**Figure 3.3 Setting Use Case**

Fully Dressed Use Case		
•	<b>Use Case Name</b>	Edit Setting
•	<b>Scope</b>	Change Language, Update Notification, Edit Profile
•	<b>Level</b>	Change Language, Update Notification, Edit Profile
•	<b>Actor</b>	User
•	<b>Stakeholder</b>	User, Admin
•	<b>Pre-Condition</b>	Internet, Platform Access, Search Criteria, Device Compatibility,
•	<b>Post-Condition</b>	Searches & Filters can be Changed
•	<b>Success Scenario</b>	Search Results

*Table 3.2 Setting use Case*

- **Browse Property Listing**

The **Browse Property Listings** use case enables both guests and registered users to explore real estate listings on the Bixel Living platform. This feature allows users to **search**, **filter**, and **view** verified properties available for **sale**, **rent**, or **investment**. Filters may include **location**, **property type**, **price range**, and **amenities**. Users can also sort results by **latest**, **recommended**, or **nearby** properties. Upon selecting a listing, users can view its details including high-resolution images, descriptions, agent contact info, and—if available—a 360-degree virtual tour. While guest users can browse listings, only registered users can **save favorites**, **initiate rental agreements**, or **participate in auctions**. The backend fetches property data from Firebase Firestore, and the search is powered by keyword and filter matching to ensure fast, relevant results.



*Figure 3.4 Browse Property Listing*

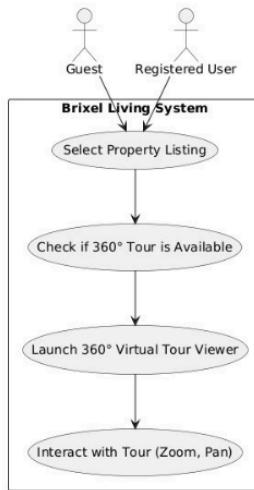
1

Fully Dressed Use Case		
• <b>Use Case Name</b>	Browse Property Listing	
• <b>Scope</b>	User Search property related to him what ever he/she needs	
• <b>Level</b>	User Search property related to him what ever he/she needs	
• <b>Actor</b>	User	
• <b>Stakeholder</b>	User, Admin	
• <b>Pre-Condition</b>	Personal Profile, Internet Connection, Authentication	
• <b>Post-Condition</b>	Only Search Available Categories	
• <b>Success Scenario</b>	Searched Successfully.	

*Table 3.3 Browse Property Listing Use Case*

- **View 360° Virtual Tour**

The View 360° Virtual Tour allows users to interactively explore properties listed on the Bixel Living platform through immersive 360-degree views. After selecting a property from the listings, users can access the virtual tour if the seller or agent has uploaded one. This feature provides a realistic representation of the property's interior and layout, improving transparency and aiding remote decision-making. The tour interface allows users to pan, zoom, and move within the virtual space using touch gestures or device sensors. This is particularly useful for long-distance buyers or renters who cannot physically visit properties. While guest users have access to view-only tours, registered users can save, bookmark, or inquire about properties after the tour. The virtual tour content is stored in Firebase Storage and rendered within the Flutter app using a compatible 360° viewer component.



**Figure 3.5 360-Degree Virtual Tour**

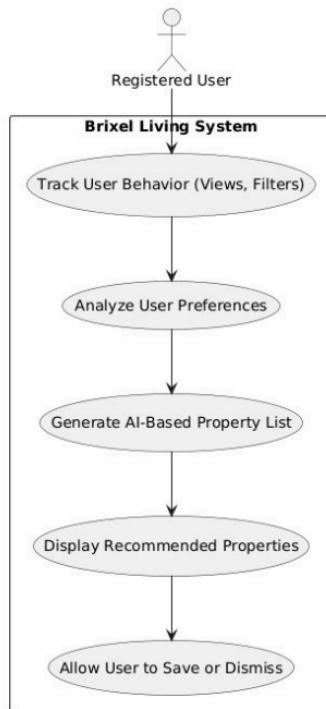
1

Fully Dressed Use Case		
•	<b>Use Case Name</b>	360° Virtual Tour
•	<b>Scope</b>	Check Property, 360° available, Launch 360°, Interact with tour
•	<b>Level</b>	Check Property, 360° available, Launch 360°, Interact with tour
•	<b>Actor</b>	User
•	<b>Stakeholder</b>	User, Admin
•	<b>Pre-Condition</b>	Authentication Internet Connection
•	<b>Post-Condition</b>	N/A
•	<b>Success Scenario</b>	360° Virtual Tour Successful

*Table 3.4 360-Degree Virtual Tour Use Case*

- **AI-Based Property Recommendations**

User behavior such listing types and search terms and filters are noted during the user interaction which leads to the suggestion of listings according to the user interests with the use of AI-Based Property Recommendations without the need of manual. Factors like location, price range, property type, and more are used as data for the recommendation of properties. It reduces the time needed to find relevant listings, and increases the chances of conversions (inquiries or transactions).



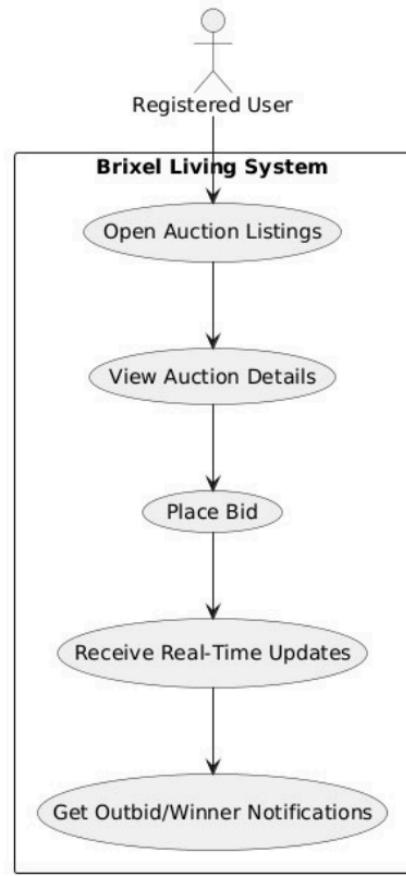
*Figure 3.6 AI Based Property Recommendations*

Fully Dressed Use Case		
• <b>Use Case Name</b>	AI Based Property Recommendations	
• <b>Scope</b>	Track user behavior, Analyze user preferences, Generate AI based property list, Allow user to save or dismiss	
• <b>Level</b> 1	Track user behavior, Analyze user preferences, Generate AI based property list, Allow user to save or dismiss	
• <b>Actor</b>	User, System	
• <b>Stakeholder</b>	User, System	
• <b>Pre-Condition</b>	Must be Login Property Searching Internet Connection	
• <b>Post-Condition</b>	Property Searching	
• <b>Success Scenario</b>	Recommended successfully	

*Table 3.5 Property Recommendation Use Case*

- **Participate in Property Auctions**

The Participate in Property Auctions use case allows registered users to engage in real-time bidding for selected properties on the Bixel Living platform. This feature provides a dynamic alternative to traditional property transactions, offering users the chance to bid on listings and secure properties at competitive prices. Users can access the “Auctions” section from the main menu, where all ongoing and upcoming auctions are displayed. Once a user selects an auction, they can view the property details, current highest bid, time remaining, and bidding history. Users place bids in real time, with the system instantly updating the auction interface across all active participants. Bidding rules ensure fairness, such as bid increments, time extensions on last-minute bids, and bid confirmations. The system also notifies users when they are outbid or win the auction. This feature is only available to authenticated users, and auction data is managed securely using Firebase Realtime Database or Firestore with Cloud Functions for live updates.



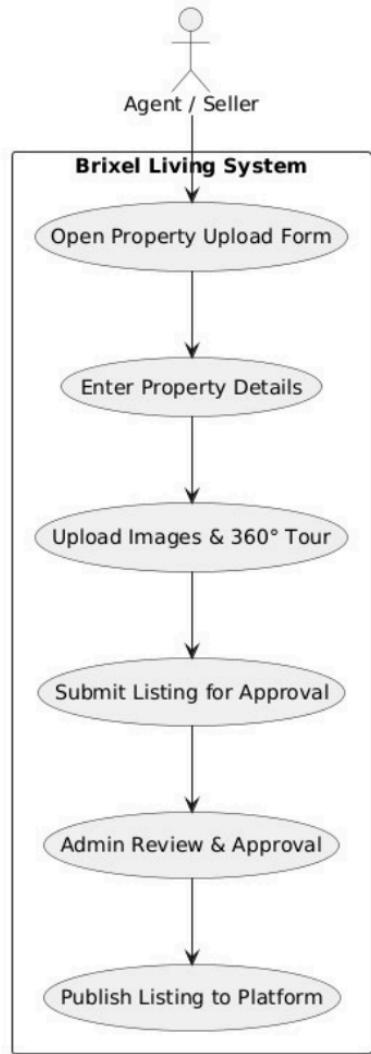
*Figure 3.7 Property Auction*

Fully Dressed Use Case		
• <b>Use Case Name</b>	Property Auction	
• <b>Scope</b>	Auction listing, view auction, place bid, receive real time update, winner notification	
• <b>Level</b>	Auction listing, view auction, place bid, receive real time update, winner notification	
• <b>Actor</b>	Admin, User	
• <b>Stakeholder</b>	Admin, User	
• <b>Pre-Condition</b>	Must be Login Internet Connection Place Bid	
• <b>Post-Condition</b>	After winning bid payment for the property	
• <b>Success Scenario</b>	Auction Successfully	

*Table 3.6 Property Auction Use Case*

- **Upload Property Listing**

The Upload Property Listing use case enables agents or property owners to publish new property listings on the Brixel Living platform. This feature allows registered users with appropriate roles to fill out a form with property details such as title, description, price, location (with Google Maps integration), number of rooms, property type, and amenities. Additionally, the user can upload high-resolution images and an optional 360-degree virtual tour. Once submitted, the listing goes into a pending approval state where an admin reviews the content to ensure it complies with platform policies and is not fraudulent. Upon approval, the listing becomes publicly visible to all users. This use case ensures that only verified, accurate, and high-quality property listings appear on the platform. Listing data is stored in Firebase Firestore, while media content (images and tours) is uploaded to Firebase Storage.



*Figure 3.8 Upload Property Listing*

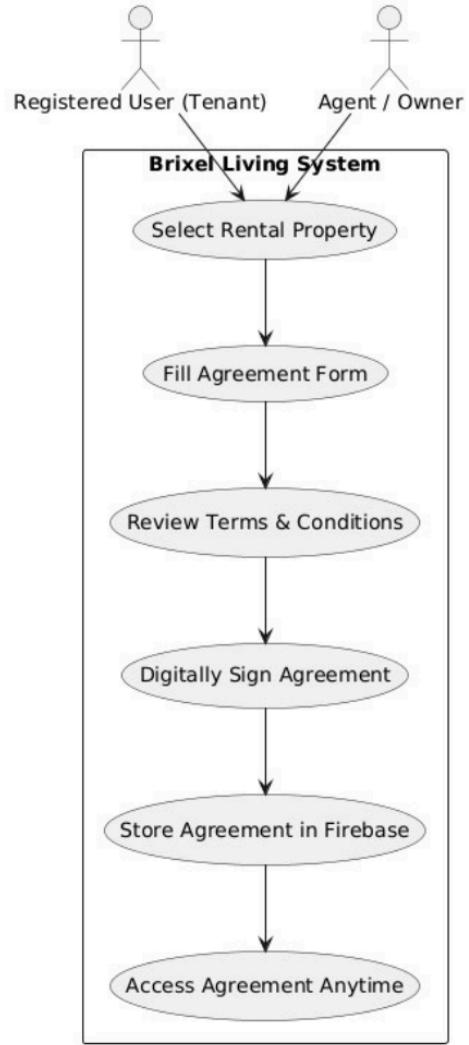
1

<b>Fully Dressed Use Case</b>		
	<b>Use Case Name</b>	Property Listing
	<b>Scope</b>	Open property uploading form, enter details, upload images, submit for approval
	<b>Level</b>	Open property uploading form, enter details, upload images, submit for approval 1
	<b>Actor</b>	Admin, User
	<b>Stakeholder</b>	Admin, User
	<b>Pre-Condition</b>	Must be Login Internet Connection
	<b>Post-Condition</b>	Submitted Property for approval
	<b>Success Scenario</b>	Property Listed Successfully

*Table 3.7 Upload Property use Case*

- **Generate Digital Rental Agreements**

The Generate Digital Rental Agreements use case allows registered users (tenants and property owners/agents) to create and sign legally valid rental agreements directly within the Bixel Living mobile application. Once a user selects a property to rent, they can initiate the agreement process. The app presents a structured form where users can fill in rental terms such as rent amount, duration, start date, tenant/owner details, and clauses. After both parties review and agree to the terms, they can digitally sign the agreement using a secure e-signature interface. Once signed, the agreement is securely stored in the Firebase database and is accessible to both parties for future reference. The system ensures data integrity and tamper-proof storage. This feature eliminates the need for paperwork, speeds up the leasing process, and adds legal accountability. Only verified users can generate or sign agreements, ensuring authenticity and preventing misuse.



*Figure 3.9 Digital Rental Agreement*

1

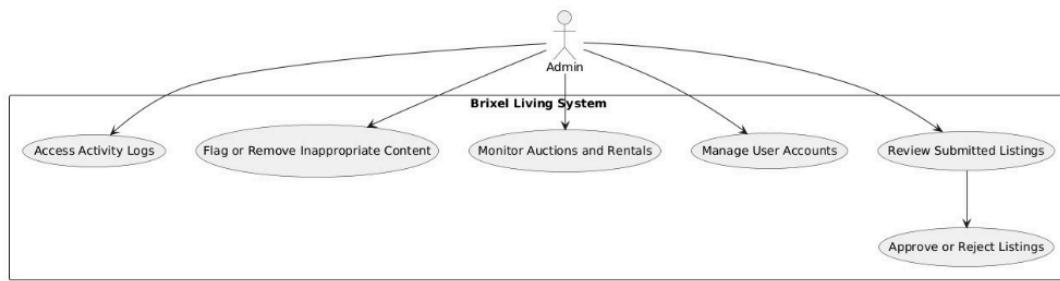
### Fully Dressed Use Case

<b>Use Case Name</b>	Digital Rental Agreement
<b>Scope</b>	Select rental agreement, fill form, terms & Conditions, Access Agreement anytime
<b>Level</b>	Select rental agreement, fill form, terms & Conditions, Access Agreement anytime
<b>Actor</b>	Admin, User
<b>Stakeholder</b>	Admin, User
<b>Pre-Condition</b>	Must be Login Internet Connection
<b>Post-Condition</b>	Must Filled Rental Application
<b>Success Scenario</b>	Property Rented Successfully

Table 3.8 Digital Rental Agreement Use Case

- **Admin Moderation**

The Admin Moderation use case emphasize the management and monitoring of activities by admin on the platform. Admins are responsible for reviewing property listings submitted by agents or sellers, approval of registration applications, ensuring that all content is accurate, non-fraudulent, and in line with platform policies..



*Figure 3.10 Admin Moderation*

1

<b>Fully Dressed Use Case</b>		
	<b>Use Case Name</b>	Admin Moderation
	<b>Scope</b>	Access Activity logs, Flag or remove inappropriate content, Monitor auction or rental process, manage accounts, review submitted listings, approve or reject listings
	<b>Level</b>	Access Activity logs, Flag or remove inappropriate content, Monitor auction or rental process, manage accounts, review submitted listings, approve or reject listings 1
	<b>Actor</b>	Admin, User
	<b>Stakeholder</b>	Admin, User
	<b>Pre-Condition</b>	Must be Login Internet Connection
	<b>Post-Condition</b>	Must have admin access
	<b>Success Scenario</b>	Modified Successfully

*Table 3.9 Admin Moderation Use Case*

# FYP Documentation

## ORIGINALITY REPORT

<b>16%</b> SIMILARITY INDEX	<b>14%</b> INTERNET SOURCES	<b>2%</b> PUBLICATIONS	<b>8%</b> STUDENT PAPERS
--------------------------------	--------------------------------	---------------------------	-----------------------------

## PRIMARY SOURCES

- |   |   |      |
|---|---|------|
| 1 | <a href="http://www.coursehero.com">www.coursehero.com</a><br>Internet Source   | 12%  |
| 2 | Submitted to University of Westminster<br>Student Paper   | 1 %  |
| 3 | Submitted to Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur<br>Student Paper   | 1 %  |
| 4 | Chen, Jessica Alice. "Medsupply Network: An Integrated Logistical Marketplace System for Buyers and Sellers in the Medical Supply Industry", Purdue University, 2024<br>Publication | 1 %  |
| 5 | Submitted to Higher Education Commission Pakistan<br>Student Paper  | <1 % |
| 6 | <a href="http://scholarworks.lib.csusb.edu">scholarworks.lib.csusb.edu</a><br>Internet Source   | <1 % |
| 7 | Submitted to Saxion Brightspace<br>Student Paper  | <1 % |
| 8 | <a href="http://eprints.utar.edu.my">eprints.utar.edu.my</a><br>Internet Source   | <1 % |
| 9 | <a href="http://fictivestudios.com">fictivestudios.com</a><br>Internet Source   | <1 % |

10 ir.unimas.my <1 %  
Internet Source

---

11 ir.uitm.edu.my <1 %  
Internet Source

---

12 www.honeymooncruiseshopper.com <1 %  
Internet Source

---

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off

# FYP Documentation

---

## GRADEMARK REPORT

---

FINAL GRADE

GENERAL COMMENTS

/0

---

PAGE 1

---

PAGE 2

---

PAGE 3

---

PAGE 4

---

PAGE 5

---

PAGE 6

---

PAGE 7

---

PAGE 8

---

PAGE 9

---

PAGE 10

---

PAGE 11

---

PAGE 12

---

PAGE 13

---

PAGE 14

---

PAGE 15

---

PAGE 16

---

PAGE 17

---

PAGE 18

---

PAGE 19

---

PAGE 20

---

PAGE 21

---

PAGE 22

---

PAGE 23

---

PAGE 24

---

PAGE 25

---

PAGE 26

---

PAGE 27

---

PAGE 28

---

PAGE 29

---

PAGE 30

---

PAGE 31

---

PAGE 32

---

PAGE 33

---

PAGE 34

---

PAGE 35

---

PAGE 36

---

PAGE 37

---

PAGE 38

---

PAGE 39

---

PAGE 40

---

PAGE 41

---

PAGE 42

---

PAGE 43

---

PAGE 44

---

PAGE 45

---

PAGE 46

---

PAGE 47

---

PAGE 48

---

PAGE 49

---

PAGE 50

---

PAGE 51

---

PAGE 52

---

PAGE 53

---

PAGE 54

---

PAGE 55

---

PAGE 56

---

