### 3.3 Stakeholders Onion Model

Figure 3.1: Onion Model of Apartment Recommendation App

The Onion Model above shows the stakeholders who are involved with our Application and their impact for the proposed system. There are some identified interactions between stakeholders in the diagram.

1. The Users for this Application can help developers to identify the requirements and develop the system as user friendly.
2. Apartment Details or API providers are important for our system to updates our application with Apartment details.

Apartment Owner are the financial supporters and their apartments are our resources.

1. The competitors have pressure on the developer to enhance the application and try to do with new features.

The system’s secureness should be good because of hackers may hack or damage our system.

1. The apartment dealers are help to promote are Application with new customers.

#### 3.3.1 Stakeholder Description

#### 

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Role** | **Viewpoint** |
| Application User | Normal User  Functionalities  Beneficiary | * Able to identify the favorable Apartments. * Able to sort out the reviews by other users. |
| Apartment dealer | Financial Beneficiary  Functional Beneficiary | * Can use this Application to recommend apartment to their customers. |
| Apartment Owner | Financial Beneficiary | * Able to advertise their apartments and give reputation for this Application. |
| Competitor | Negative Stakeholder | * Wants to build a better system than this application with more functionalities. |
| Hacker | Negative Stakeholder | * Able to harm our system and downing the reputation, and getting our functionalities without our copyright. |
| Supervisor | Advisory | * Provide advices and guidance to successfully complete the project in given time. |
| Developer | Designer of Application | * Build the Application and teach the machine to predict the results for user’s comfortable. |
| Domain Experts | Experts | * Having Knowledge in the domains, they can provide knowledge enhance the system. |
| Table 3.1: Stakeholder Description of Onion model | | | |

### 3.3.2 Stakeholder Analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Impact On:** | | | | |
| **Stakeholders** | **Finance** | **Business as Usual** | **Reputation** | **Thread** | **Support** |
| Apartment Owner | High |  | High |  | Normal |
| Apartment Details Provider |  |  |  |  | High |
| Application User |  | High | Normal |  | Normal |
| Apartment Dealer | Normal | Normal | Normal | low |  |
| Competitor |  |  |  | High |  |
| Supervisor |  |  |  |  | High |
| Table 3.2: Stakeholder Analysis of Onion model | | | | | |

**3.3.3 Context Diagram**

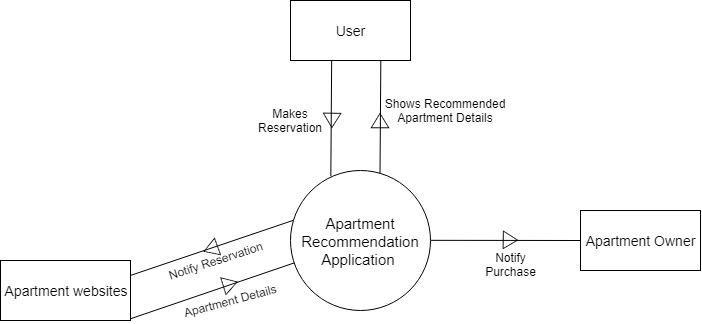


Figure 3.2: Context diagram of Apartment Recommendation System.

The above context diagrams of the Application shows the System’s boundary and the Environment Factors.

* The outgoing Events are Notifying, shows the output of resources.
* The Incoming Events are used to collecting details and make reservation.

### 3.4 [Use Case](https://mail.google.com/mail/u/1/#m_-2838936569934205253__Toc396716546) Diagram



Figure 3.3: Use case diagram of Apartment Recommendation Application.

#### 3.4.1 [Use Case Descriptions](https://mail.google.com/mail/u/1/#m_-2838936569934205253__Toc396716546)

##### 1. Use case: get User Preferences

|  |  |
| --- | --- |
| Case ID | 01 |
| Description | This use case describes how the application get the preference from user for apartment prediction. |
| Priority | High |
| Actors | User |
| Pre-Conditions | user should have access to use the application |
| Post Conditions | If the use case is successful, the user will able to view their most perfect recommended apartment. |
| Trigger Events | None |
| Flow of activities | * To get preference The forms will give to uses through the application * Check whether the preferences are correctly given. * Store the user preference to the cloud based database. * Then analysis preference and finalized recommended apartment with details. |
| Assumptions | The user is not a robot. |
| Exceptional Flow | * Missing important preferences. * Getting preferences until wanted preferences are satisfied. |
| Table 3.3: Use Case Description for getting user preferences. | |

##### 2. Use case: Calculate users similarity

|  |  |
| --- | --- |
| Case ID | 02 |
| Description | This use case describes how the system going to calculate similarity between users. That will help to predict more accurate in our recommendation app. |
| Priority | High |
| Actors | System |
| Pre-Conditions | The System should able to access all the user’s preferences. |
| Post Conditions | If this use case is successful, the system will be able calculate similarity between users and group them one for get them their perfect apartment quickly. |
| Trigger Events | None |
| Flow of activities | First system will go through the user preference |
| Assumptions | The user already have account in the Application. |
| Exceptional Flow | * Empty fields. * Incorrect Email or Password. * Email not found. |
| Table 3.4: Use Case Description for Login | |

1. **Use case: View Apartment Details**

|  |  |
| --- | --- |
| Case ID | 03 |
| Description | This use case describes how the customer View the app when the open up, and other features. |
| Priority | Medium |
| Actors | User |
| Pre-Conditions | Customer must be Logged In on this Application. |
| Post Conditions | Customer can choose any other various features. |
| Trigger Events | none |
| Flow of activities | * Customer will see recent or filtered apartments. * System will analyses user’s details. * If they want, then they can see their predicted Apartments. |
| Table 3.5: Use Case Description for View Apartment Details | |

1. **Use case : Modify Preference**

|  |  |  |
| --- | --- | --- |
| Case ID | 04 | |
| Description | This use case defines about the customer going to change their personal preferences. | |
| Priority | High | |
| Actors | User | |
| Pre-Conditions | Customer logged into the application account. | |
| Post Conditions | If the use case is successful the user can change his/her preferences and it will be update in the database (cloud based database). | |
| Trigger Events | User clicking the settings button. | |
| Flow of activities | * User will be prompt to enter their old password for authentication. * User enters his/her new preferences or their preferences to change. * Authenticate his/her new or changed preferences. * Database will update with new/changed preference. | |
| Exceptional Flow |  | * Invalid old Password for user authentication. * Validate unknown preferences. * If unknown preferences are valuable then create new category with those preferences. |
| Table 3.6: Use Case Description for Modify preferences. | | |

1. **Use case : get Apartment Details**

|  |  |
| --- | --- |
| Case ID | 05 |
| Description | This use case describes how the system going to get Apartment details from apartment websites and apartment details provides. |
| Priority | High |
| Actors | Apartment sites |
| Pre-Conditions | The system should have internet to connect with cloud.  The apartment sites should be in online. |
| Post Conditions | If this use case is successful, the System will get the details from Apartment details provides(Apartment sites) |
| Trigger Events | There is no Trigger Events. Auto generate. |
| Flow of activities | * The System will check the internet connection. * Then System will check whether the connection between cloud and system is available or not. * System will check the apartment details providing sites are online or not. * If all the flows above are true then, System will get the Apartment details from Apartment sites and store the data to the cloud. |
| Assumptions | The Apartment details providing websites allows to get their details. |
| Exceptional Flow | * If there any issues from connect then connect error message will provide to user. |
| Table 3.7: Use Case Description for getting apartment details | |