Contents

[First Deliverable for Object Oriented Approach 4](#_Toc70791322)

[Chapter 1: Feasibility Report 4](#_Toc70791323)

[Introduction 5](#_Toc70791324)

[Project Feasibility report: 5](#_Toc70791325)

[Technical Feasibility 5](#_Toc70791326)

[Operational Feasibility 5](#_Toc70791327)

[Economic Feasibility 5](#_Toc70791328)

[Schedule Feasibility 6](#_Toc70791329)

[Specification Feasibility 6](#_Toc70791330)

[Information Feasibility 6](#_Toc70791331)

[Motivational Feasibility 6](#_Toc70791332)

[Legal and Ethical feasibility 6](#_Toc70791333)

[Project/product scope 6](#_Toc70791334)

[Project/product costing 7](#_Toc70791335)

[Project Cost Estimation by Function Point Analysis 7](#_Toc70791336)

[**Basic COCOMO** 8](#_Toc70791337)

[Task dependency table 12](#_Toc70791338)

[Cam-critical path method 13](#_Toc70791339)

[Gantt chart 14](#_Toc70791340)

[Introduction to team members: 14](#_Toc70791341)

[Task and member assignment: 15](#_Toc70791342)

[Allocation of People to Activities: 15](#_Toc70791343)

[Tools and Technologies with reasons: 16](#_Toc70791344)

[Vision document 16](#_Toc70791345)

[Scope 16](#_Toc70791346)

[Problems in Current System: 16](#_Toc70791347)

[Solution: 17](#_Toc70791348)

[Stake Holders: 17](#_Toc70791349)

[Assumption: 17](#_Toc70791350)

[Constraints: 17](#_Toc70791351)

[Key Benefits: 17](#_Toc70791352)

[RISK LIST 18](#_Toc70791353)

[Product features and product decomposition: 18](#_Toc70791354)

[Deliverable 2 20](#_Toc70791355)

[Software requirement specification 20](#_Toc70791356)

[Introduction: 21](#_Toc70791357)

[System specification: 21](#_Toc70791358)

[Identifying External Entities: 23](#_Toc70791359)

[Context level DFD: 23](#_Toc70791360)

[Capture “shall’’ statements: 24](#_Toc70791361)

[Allocate Requirements: 25](#_Toc70791362)

[Prioritize Requirements: 26](#_Toc70791363)

[Requirement Trace-ability Matrix: 27](#_Toc70791364)

[High level use case diagram: 29](#_Toc70791365)

[Analysis level use case diagram: 30](#_Toc70791366)

[Use case Description: 30](#_Toc70791367)

[Get Application: 30](#_Toc70791368)

[Enable location: 31](#_Toc70791369)

[Location filters: 31](#_Toc70791370)

[Display distances: 32](#_Toc70791371)

[Optimal path: 32](#_Toc70791372)

[Get Hotel data: 32](#_Toc70791373)

[Calculate rating: 33](#_Toc70791374)

[Show rating: 33](#_Toc70791375)

[Budget Filter: 33](#_Toc70791376)

[Recommendation: 34](#_Toc70791377)

[Listings: 34](#_Toc70791378)

[Feedback: 34](#_Toc70791379)

[Chapter 3: Design Document 36](#_Toc70791380)

[Introduction: 37](#_Toc70791381)

[Domain Model: 37](#_Toc70791382)

[High level domain model: 37](#_Toc70791383)

[Analysis level Domain model: 38](#_Toc70791384)

[System Sequence Diagram: 38](#_Toc70791385)

[Sequence Diagram: 39](#_Toc70791386)

[Sequence diagram of budget: 40](#_Toc70791387)

[Sequence diagram of getting data(from www): 40](#_Toc70791388)

[Sequence diagram of location module: 41](#_Toc70791389)

[Sequence diagram of listings: 41](#_Toc70791390)

[Sequence diagram of feedback: 42](#_Toc70791391)

[Collaboration Diagram: 42](#_Toc70791392)

[Design Class Diagram: 43](#_Toc70791393)

[Identified classes: 43](#_Toc70791394)

[Class diagram: 44](#_Toc70791395)

[State chart diagram: 45](#_Toc70791396)

[Activity diagram: 46](#_Toc70791397)

[Activity diagram Content based filtering: 46](#_Toc70791398)

[Activity Diagram of Search process: 47](#_Toc70791399)

[Activity diagram of View hotels: 47](#_Toc70791400)

[Activity diagram of Collaborative searching: 48](#_Toc70791401)

[Activity diagram of Hybrid filter: 48](#_Toc70791402)

[Package diagram: 49](#_Toc70791403)

[Data model of the system: 49](#_Toc70791404)

[3rd Deliverable (User Interface Design) 50](#_Toc70791405)

[Introduction: 51](#_Toc70791406)

[Sitemap: 51](#_Toc70791407)

[Storyboards: 51](#_Toc70791408)

[Navigational Maps: 55](#_Toc70791409)

[Traceability matrixes: 58](#_Toc70791410)

# 

# First Deliverable for Object Oriented Approach

# Chapter 1: Feasibility Report

## Introduction

It happened every time when people visit the different cities for the very first time for different purposes and they need a place to stay but they don’t know anything about the place. Of course they will look out for some reasonable and best hotels for them. And they will check which hotel is nearest to their current location. It can be time consuming for the people to go to each hotels and deciding whether it in their budget or not. People can only know the hotels reputation once they get there. So our application” Hotel selection based on hybrid recommendation using AI” will facilitate people to find a hotel best suited to their budget and having excellent reviews and at the nearest place. Based on a user’s location it will recommend best hotel based on collaborative filtering to the user. The system will maintain all the records of the hotels on the map along with their rents. The system will calculate the optimal path to the hotel after determining the location of the user after user has selected the hotel to visit, the interface of the system can be accessed by anyone having internet connection. The aim of this application is to predict and recommend hotel clusters to a user that meets his/her requirements. Using this application best hotel based on user experience can also be found. Using the comments of users on the websites of the hotel we can find the reviews i.e. which hotel is best to visit according to user experience. Without this much physical effort is utilized and time is also wasted. There is no such application so far that help user to find the best hotel.

## Project Feasibility report:

### Technical Feasibility

React native technologies will be used, there is a lot of work already done in react native applications and technologies associated with it which makes the implementation and development very easy for us. So technically our project is feasible. Since react native develop the applications which are cross platform and is compatible across any platform. It offers slick, plane user interface which is also responsive and making the less burdened of the time. It is quite faster and less costly to develop the react native application without compromising the quality and features.

### Operational Feasibility

Selecting the hotel which is reasonable means is in person’s budget and nearest to its current location and having good reviews when the person is new in the city is one of the major and significant existing issues. So the aim is to build such an application that facilitates the users to select the hotel which he/she can afford and nearest to its current location and has good reviews without wasting too much time and with very less effort. Our project will be easy to use that even an unprofessional man can understand easily how to operate it.

### Economic Feasibility

Economically, our project is feasible according to us because we have used the most effective method for the analysis of the cost and our project budget. Which leads us to identify the profit compare to the investment anticipated from a project? Agency returns, costs and even the public benefits are increased. The net benefit will be demonstrated of the planned project in order to accept the electronic funds/benefits, also taking into account the agency benefits and costs. Minimizing the waste and ineffectiveness every resource is allocated to help each individual making the economic state of our project highly feasible.

### Schedule Feasibility

The important factor to achieve the success is time. Our deliverables will be presented by the team members and will fulfill the milestones and checkpoints on time. The probability to complete our project in time is high. In order to meet the deadlines, we can also add the alternatives for the timelines. Before the system becomes outdated or pointless we ensure that the project will be completed.

### Specification Feasibility

This application will be useful for all the people who visit different places very often. And will help them to recommend only those hotels which fall into their budget and nearest to them. User will be able to select the hotel which has good reviews so he/she can select the hotel without any doubt. The system will reduce time and effort of the user.

### Information Feasibility

Our project will end up the hustle of finding the hotel which fulfills all the requirements, the whole process will be automated and efficiency to handle also increased. It will facilitate the user to get information on time and recommendation system module will help users to choose right hotel according to their specifications.

### Motivational Feasibility

Our team for this project is highly motivated as there is no such application in Pakistan. It is an issue for the people who visit different cities very often like Lahore or Islamabad. Regarding the skills related to solving the problem our team is very good. Even our supervisor is very skill full and motivated.

### Legal and Ethical feasibility

Legally and even ethically our project is completed feasible. Confidentiality is maintained and information will not be provided to other organizations. We do not copy from others.

## Project/product scope

The scope of this project is very broad because in Pakistan there is no such an application that recommends the user the hotels which meets their particular requirements such as budget, optimal distance and review based recommendations etc. As most of the people visits different cities and in order to stay there they want to select the hotel with good reputation and according to their budget. But due to lack of knowledge about the area they are unable to find the hotel with their specified requirements. Our system saves the time and effort of them and provides efficient services that help in recommending majority to find affordable hotel and nearest to their current location by avoiding all the hotels which don’t fulfill their specifications. The hustle involved in this process will be decreased and making the process easy and effective.

The other reason to make this project is that most of the time people need to visit the hotels to decide whether they should stay there or not but with our application he/she will select the hotels which our system recommend according to the budget, checking current location and giving the nearest hotels in option and most importantly calculating the reviews to conclude whether they are bad or good. So, from standing where they are they will be able to select the hotel without wasting the time and exhausting their selves.

## Project/product costing

### Project Cost Estimation by Function Point Analysis

**Step 1**

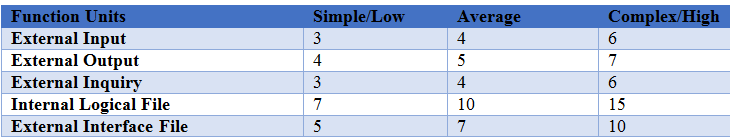


Table 1 Function type

**Step2**

Calculating the UFP (Unadjusted function point) for the average values

User input = 2

User output = 5

User Enquiries = 4

User Files = 1

External Interfaces = 2

Formula to calculate the UFP

UFP = (2\*4) + (5\*5) + (4\*4) + (1\*10) + (2\*7)

UFP = 8 + 25 + 16 + 10 + 14

UFP =73

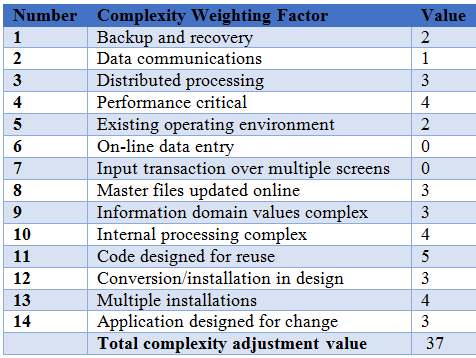


Table 2Complexity weighting

**Formula**

To calculate the CAF

CAF = [0.65 + 0.01 \* ∑ (If)]

CAF = 0.65 + 0.01 \* (37)

CAF = 0.65 + 0.37

CAF = 1.02

Step3

Find result of software

FP = UFP \* CAF

FP = 73 \* 1.02

FP = 74.46

Project Cost Estimation by using COCOMO’81 (Constructive Cost Model)

### **Basic COCOMO**

Organic

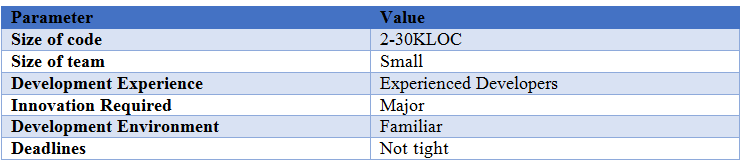


Table 3 Organic parameters

Parameters of Different Modes

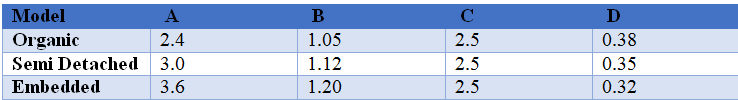


Table 4 Parameter matrix

**Effort:**

E = a (KLOC) ^b Person/Month

E = 2.4(4) ^1.05

E = 10.289 Person/Month

**Development Time:**

Dev Time = c (Effort) ^d Months

Dev Time = 2.5(10.289) ^0.38 Months

Dev Time = 6.0623 Months

**Staff Size:**

Average Staff Size = Effort/ Dev Time

Average Staff Size = 10.289 / 6.0623

Average Staff Size = 1.6972

**Productivity:**

Productivity = KLOC/Effort

Productivity = 4/10.289

Productivity = 0.3887

Activity Based Costing

**Product Attribute:**

Required Reliability: Nominal – 1.00

Database Size: Nominal – 1.00

Product Complexity: V. High – 0.95

**Personnel Attribute:**

Applications experience: Nominal – 1.00

Language Experience: V. High – 0.95

**Project Attribute:**

Software Tools: V. High – 0.91

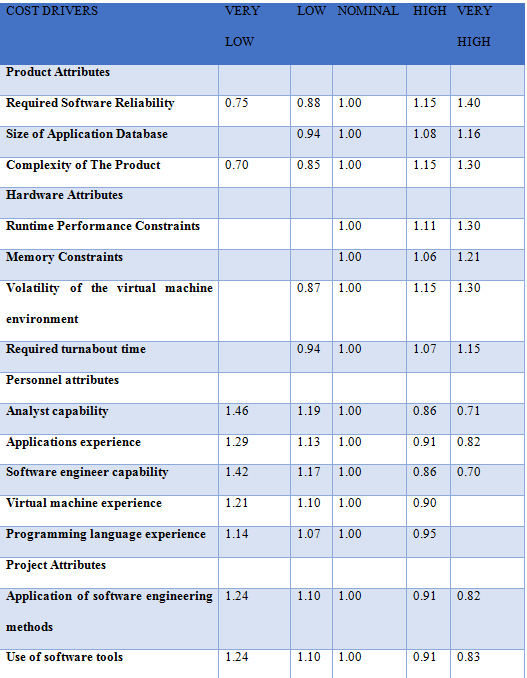


Table 5 Cost drivers

### Task dependency table

|  |  |  |  |
| --- | --- | --- | --- |
| Task  No. | Activities | Duration  (Weeks) | Dependency |
| T1 | Explore React Native | 2 | - |
| T2 | Explore existing research papers | 2 | - |
| T3 | Explore existing algorithms | 3 | - |
| T4 | Proposal Refinement | 2 | T1,T2,T3 |
| T5 | Proposal Defense | 1 | T4 |
| T6 | SRS | 2 | T5 |
| T7 | DD | 3 | T6 |
| T8 | Prototype | 2 | T7 |
| T9 | SRS & DD Defense | 1 | T8 |
| T10 | Fetching the Data for Calculating Ratings (Web Scrapping) | 3 | T9 |
| T11 | Online Hotel Information Gathering | 3 | T9 |
| T12 | Google Map Management | 2 | T9,T10 |
| T13 | Feedback Management | 2 | T9 |
| T14 | Completing Frontend | 2 | T9 |
| T15 | Testing | 2 | T12,T13,T14 |
| T16 | Final Deployment | 1 | T15 |
| T17 | Final Defense | 1 | T16 |

Table 6 Task Dependency

### Cam-critical path method

**1. Specify the Individual Activities**

Tasks are divided into broad activities. And all the activities are achieved on the weekly bases and Gantt chart is created including allocation of staff. In our case there are 17 tasks as given in Table 6 (Task Dependency).

**2. Estimate Activity Completion Time**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tasks/Activity | Duration | ES | EF | LS | LF | TS |
| T1 | 2 | 0 | 2 | 1 | 3 | 1 |
| T2 | 2 | 0 | 2 | 1 | 3 | 1 |
| T3 | 3 | 0 | 3 | 0 | 3 | 0 |
| T4 | 2 | 3 | 5 | 3 | 5 | 0 |
| T5 | 1 | 5 | 6 | 5 | 6 | 0 |
| T6 | 2 | 6 | 8 | 6 | 8 | 0 |
| T7 | 3 | 8 | 11 | 8 | 11 | 0 |
| T8 | 2 | 11 | 13 | 11 | 13 | 0 |
| T9 | 1 | 13 | 14 | 13 | 14 | 0 |
| T10 | 3 | 14 | 17 | 14 | 17 | 0 |
| T11 | 3 | 14 | 17 | 14 | 17 | 0 |
| T12 | 2 | 17 | 19 | 17 | 19 | 0 |
| T13 | 2 | 14 | 16 | 17 | 19 | 3 |
| T14 | 2 | 14 | 16 | 17 | 19 | 3 |
| T15 | 2 | 19 | 21 | 19 | 21 | 0 |
| T16 | 1 | 21 | 22 | 21 | 22 | 0 |
| T17 | 1 | 22 | 23 | 22 | 23 | 0 |

Table 7 Activity Completion Time

**3. Identify the Critical Path**

As the path is considered to be the critical path with the highest value of the duration so: Critical path of our project is

T2, T4, T5, T6, T7, T8, T9, T12, T15, T16, T17

### Gantt chart

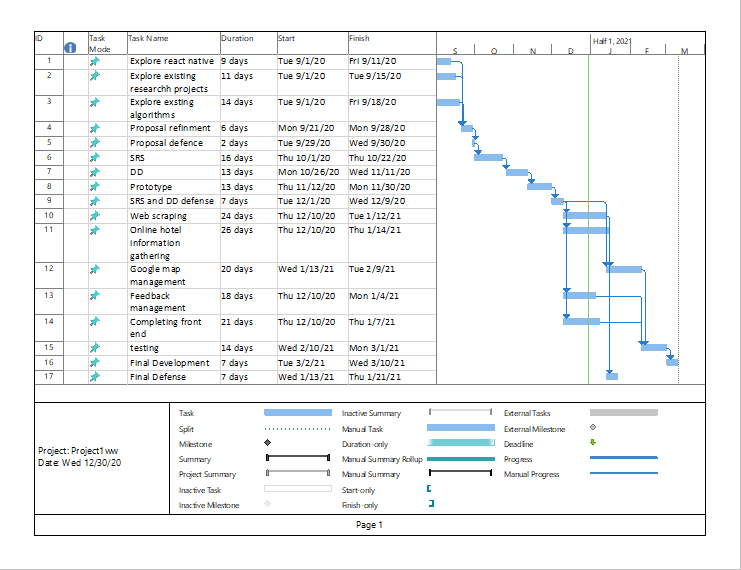


Figure 1Gantt chart

### Introduction to team members:

In our team we are three members working on the project. And main 3 modules are in our project which is divided into further sub modules. The brief introduction of each member is given as follow:

Neha Durani: Developer and designer

Saira Irtaza: Working on modeling, requirement engineer and developer

Zain Bashir: Database designer and developer

### Task and member assignment:

### Allocation of People to Activities:

|  |  |  |  |
| --- | --- | --- | --- |
| Task  No. | Activities | Duration  (Weeks) | Members |
| T1 | Explore React Native | 2 | All |
| T2 | Explore existing research papers | 2 | All |
| T3 | Explore existing algorithms | 3 | All |
| T4 | Proposal Refinement | 2 | All |
| T5 | Proposal Defense | 1 | All |
| T6 | SRS | 2 | All |
| T7 | DD | 3 | All |
| T8 | Prototype | 2 | All |
| T9 | SRS & DD Defense | 1 | All |
| T10 | Fetching the Data for Calculating Ratings (Web Scrapping) | 3 | Neha Durani |
| T11 | Online Hotel Information Gathering | 3 | Saira Irtaza |
| T12 | Google Map Management | 2 | Zain Bashir |
| T13 | Feedback Management | 2 | Zain Bashir |
| T14 | Completing Frontend | 2 | Neha Durani |
| T15 | Testing | 2 | Saira Irtaza |
| T16 | Final Deployment | 1 | Neha Durani |
| T17 | Final Defense | 1 | All |

Table 8 activity table

### Tools and Technologies with reasons:

* React Native for frontend development.
* Node Js for backend development
* Firebase database because it has better compatibility with Node js.

Since we are developing a native app. So will use visual studio, android studio, node js all are required for react native developing environment. We will use node js because it is more compatible. And will use firebase because of its compatibility and multiple features for database management.

### Vision document

This cross platform application Hotel selection based on hybrid recommendation using AI include various features, which help the end users in selection the nearest hotel which is in their budget and have good reviews. This system is an end user support. This system provides user an easy way to select the nearest hotel according to their needs without wasting time and with very little effort. This application will help the people who visit different places very often. Using this application one can find very nearest place to visit based on the budget we have. Using this application best hotel based on user experience can also be found. Using the comments of users on the websites of the hotel we can find the reviews i-e which hotel is best to visit according to user experience. Without this much physical effort is utilized and time is also wasted. There is no such application so far that help user to find the best hotel based on collaborative filtering. Basic purpose of this system is to facilitate the users by saving their time and recommending them a better hotels suggestion which meets their requirements. In short this system will only recommend those hotels which will fulfill the user’s requirements and other will be ignored means only cluster of hotels will appear that meets the requirements such as budget, nearest to current location and having good review/reputation.

Scope: This system can be used in any city of Pakistan.

### Problems in Current System:

There is no existing system of this type in our country which recommends the hotels according to various factors most importantly the budget. There are many simple hotel recommendation systems but these systems are web based and they just recommend the hotels of whole city. With the reviews of the user they tell us whether the hotel is good or bad according to different criteria’s. When user selects a specific city like Gujrat it displays all the hotels there in ascending order. On top of the list highest reviewed hotels are display. Review is calculated based on user reviews which they have given of the application. There is no module to represent nearby hotels based of location. This application displays results of the hotels that added in database by admin, so scope is very low. Unauthentic reviews as they are calculated based on review that only a single user has given to application.

### Solution:

Our system will utilize three attributes i-e current location, budget and reviews of the users and display the nearest hotels having good reputation and are in budget. As well as it will display the optimal path to all the hotels that best suits to our result. The system will use machine learning algorithm to get the comments from website available on Google and will calculate the rating. The system will use collaborative filtering and display the most appropriate results. Use web Scraping and sentimental analysis real data from Google websites will be used. Very useful for the people who visit any place for the first time. Authentic rating as the result is based on comments of many users.

### Stake Holders:

General Public: The user of our system will be the general public and no special expertise will be required and users will be able to find the best suitable hotel for themselves.

### Assumption:

* The availability of the system will be 24/7 of time.
* This system keeps the confirmed individual data for users. Hence design should consider security and data accuracy.
* Data displayed by the system will be 100% accurate.
* The hotel selection will be less than 10 seconds.

### Constraints:

* English Language GUI.
* Because of Google API usage, all the hotels will not be covered.
* No Implementation of Hotel full menu and availability of bookings.

### Key Benefits:

* Accuracy Level of this system will be advanced. All process would be done properly and it covers all information accurately.
* No repetition of information anywhere in storage.
* The system would be easy to control and it can be established within short period of time within budget.
* Avoiding the errors and tracking each single aspect.
* Data security will be improved.
* For the users the Interface will be very easy to use.
* User’s time and travel will be saved.

### RISK LIST

* In case if any of the member is unable to continue with this project there will be no substitute of it, as more time will be required to start from the beginning.
* High pressure due to less development time.
* Lack of motivational and moral support.
* More development time for reporting in detail.
* Project increased cost because of adding extra features/functionalities.
* Strengths unable to match with the assignment given to people.
* Unfamiliarity of new tools and technology require extra time.
* Conflicts in ideas among team members lead to bad performance, additional meetings and extra rework.
* Schedule expansion by adding functionalities that are not required.
* Low quality components can be delivered by the team members leading to spend extra time on quality improvement.
* No experience in project management, specifically in the testing, management of risk and management of changes in current system.
* Other work like midterm and final preparations.
* Not enough resources.

### Product features and product decomposition:

Our System contains the features which target the common community who visits different cities very often and need a good hotel to stay. Hence covering all possible aspects which a person looks for while searching for a hotel to stay. Our system provides many facilities such as current location of the user and recommends the nearest hotels. By using location services user can find nearby hotels easily. The common features of this type of software are:

* Predict the actual location of the user using Google maps API.
* Predict nearby hotels based on distance.
* Determine distance between paths.
* Determine optimal path to destination using Dijkstra algorithm and GeoJason libraries.
* Display the list of hotels with full details using web scraping.
* Recommendation based on rating using sentimental analysis using request-promise or CheerioJS.
* Recommendation based on distance by ML algorithm.
* Recommendation based on budget of user

\

# Deliverable 2

# Software requirement specification

## Introduction:

Our document is to presenting a detailed description of the Hotel selection based on hybrid model using artificial intelligence system. It is explaining the reason and features of the android application, the interfaces, what the application will do, the constraints under which it must operate.

### System specification:

We have added the following clauses in requirement specification:

**Introduction to the application:**

It might happen many times when someone visit other city and do not know much about the hotels and rest houses in that area. Off course, for the people who visit other areas very often and want to have a stay there they need to know what places are available for them and whether they can afford them or not. Further, it is very time-consuming task to visit other places and go for each hotel to check for budget and other preferences.

To overcome this problem, we have developed “Hotel selection system” which help user to make selection of good hotel according to his preferences. The system provide user to look at the hotels within his range and within specific area and along with ratings. The user can also give feedback about the accuracy of the system. The functions are provided in such a way that the efficiency is increased and the workload is reduced.

**Existing System**

* Much physical effort is utilized in finding rooms.
* Precious time and money are wasted.
* Cause hustle for the users to find the suitable rooms.
* There is no such system which provides us recommendation about the system with that much high accuracy.

**Organization chart**

We are three individuals in our group. We are very little encountered that one of us drove us towards the venture assignments. We are cooperating in all assignments like dealing with the timetable, gathering prerequisites, making configuration archives, finishing documentations and building up the framework. So, we are chiefs, necessity engineers, planners, designers and analyzers simultaneously. Simultaneously we need to figure out how the hierarchical functions go. So, our hierarchical is as per the following.

**Figure (1) Organizational Chart**

Scope of the System:

The scope of our system is very high as there is no such application in Pakistan that recommend us a hotel based on our budget i-e either the hotel is in our budget or not. Moreover, this model considers how much hotel is popular in area, input locations and user’s preference. It gives the guess with 53.6% accuracy on the data we used for testing -4% improvements on pure model. Hence this system is improving the accuracy of recommendation up to a level. Till now recommendation are purely context based in which only user input is taken, and results are shown but this system is taking user preferences along with the user input and telling us the best rated hotel in an area. Our application facilitates users by providing a platform where they can find hotels easily by sitting on their couch. So, our system hopefully will help to resolve this issue in quicker way to find best hotels that makes our project more scalable.

**Summary of requirements:**

Our system is going to facilitate the user by providing them recommendation of the best hotel according to their preferences. It will give them cluster of 5 hotels that will be best suited to their input values. Using their input that is context-based filtering it will also utilize the concept of involving user preferences i-e getting rating from www and giving the hotels that the similar users have reviewed best. The basic aim of our project is to enable common person to avail comfort of searching a best hotel.

The generalized summary of our system requirements is as follow:

**Location module:**

All the user who have the app can enjoy this feature. This module is going to predict the actual location of the user. For this purpose, we will use Geolocation library of Google Map API. For this purpose, user must have to enable location services of device. After finding the exact location of user the system will determine the nearby hotels of the area based on the location. Next it will find the best cluster of the hotel using GeoJason libraries. In our system we will show cluster of the 5 hotels that are best in our range.

**Recommendation engine:**

Our system is giving us recommendation using three strategies that are using budget, ratings and location of user. It is joining two types of filtering i-e context-based filtering and preference-based filtering. Using www, the system will get the comments from the websites and then calculate the ratings of the hotels. The system will utilize the web scraping strategy to get the data from web and sentimental analysis to calculate the ratings of the hotels. For that purposes we will use cheerioJs and request-promise libraries. System will give us the separate results of the filtering as well as hybrid result.

**Hotel listing:**

The system will have ability to display the list of whole hotels in an interactive way. The pages will display all the necessary information of the hotels. All users can search for the listings in the relevant area where the user want to find hotels.

**Feedback:**

The user will be provided with the ability to give the feedback about the application that whether the system results were as they expected. These feedbacks will be utilized to improve the issues in the future. Here user can give the comments about the system so that the word matching algorithm will be used to verify whether the system is good or not.

## Identifying External Entities:

We have extracted the entities based on the information that is in our context. The external entities are same as internal entities. “Hotel selection so that everything remains comprehensible .The identification of external interfaces is done in phases.

Over Specify Entities from Abstract:

Entities from abstract are as follow:

* User
* Find hotel
* Location prediction
* Rating
* Budget match
* Feedback
* Listing
* Collaborative filtering
* Hybrid filtering
* Perform refinement:

After the refinements the entities that are identified are as follow:

* User
* Queries
* Recommendation engine
* Listings
* Feedback

## Context level DFD:

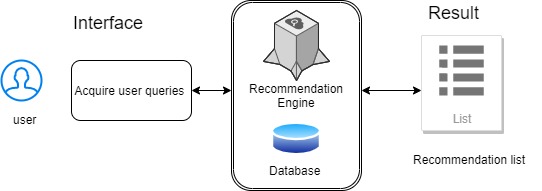


Figure 2 Context level DFD

## Capture “shall’’ statements:

The identified shall requirements of the system are:

|  |  |
| --- | --- |
| Para# | Initial Requirements |
| 1.0 | User “shall” get/download application from play store/Appstore. |
| 1.0 | User shall enable his location services. |
| 1.0 | System “shall” predict the actual location of the user  System “shall” find the nearest hotels based on distance.  System “shall” make the cluster of nearest hotels. |
| 1.0 | System “shall” find the distance between the user and relevant hotels. |
| 1.0 | System “shall” find the optimal path to reach the location. |
| 2.0 | System “shall” get the data of hotels from www. |
| 2.0 | System “shall” extract the relevant information.  System “shall” analyze the data to give them points.  System “shall” find the rating of hotel. |
| 2.0 | User “shall” enter the search criteria. |
| 2.0 | System “shall” join all filters and make a hybrid search |
| 2.0 | System “shall” recommend the best hotels. |
| 3.0 | System “shall” display the listing of all the hotels of a specific area. |
| 4.0 | User “shall ” be able to give feedback |

Table 9 shall requirements

## Allocate Requirements:

|  |  |  |
| --- | --- | --- |
| Para# | Initial Requirements | Use case |
| 1.0 | User “shall” get/download application from play store/Appstore. | UC\_GetApp |
| 1.0 | User shall enable his location services. | UC\_Enable\_location |
| 1.0 | System “shall” predict the actual location of the user  System “shall” find the nearest hotels based on distance.  System “shall” make the cluster of nearest hotels. | UC\_location\_Filter |
| 1.0 | System “shall” find the distance between the user and relevant hotels. | UC\_Find\_Distance |
| 1.0 | System “shall” find the optimal path to reach the location. | UC\_Optimal\_Path |
| 2.0 | System “shall” get the data of hotels from www. | UC\_Get\_Data |
| 2.0 | System “shall” extract the relevant information.  System “shall” analyze the data to give them points.  System “shall” find the rating of hotel. | UC\_Calculate\_Rating |
| 2.0 | User “shall” enter the search criteria. | UC\_Search |
| 2.0 | System “shall” join all filters and make a hybrid search | UC\_Hybird\_Search |
| 2.0 | System “shall” recommend the best hotels. | UC\_Recommend. |
| 3.0 | System “shall” display the listing of all the hotels of a specific area. | UC\_Listing |
| 4.0 | User “shall ” be able to give feedback | UC\_Feedback |

Table 10 Allocate requirements

## Prioritize Requirements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Para# | Rank | Initial Requirements | Use case id | Use case name |
| 1.0 | Highest | User will get/download application from play store/Appstore to get the service of the system. | UC\_1 | UC\_GetApp |
| 1.0 | Highest | User will enable his location services. The system is GPS based so location service is mandatory. | UC\_2 | UC\_Enable\_location |
| 1.0 | Highest | System will predict the actual location of the user using gps.  System will find the nearest hotels based on distance.  System will make the cluster of nearest hotels. | UC\_3 | UC\_location\_Filter |
| 1.0 | Highest | System will find the distance between the user and relevant hotels to determine which the nearest one is. | UC\_4 | UC\_Find\_Distance |
| 1.0 | Highest | System will find the optimal path to reach the location. It is shortest path to each a destination. | UC\_5 | UC\_Optimal\_Path |
| 2.0 | Highest | System will get the data of hotels from www by the process of web scraping/crawling using CheerioJS. | UC\_6 | UC\_Get\_Data |
| 2.0 | Highest | System will extract the relevant information i-e is comments on the websites. System will analyze the data to give them points i-e if the words are very good then positive number for neutral zero and bad negative points. System will combine the points to find complete rating. | UC\_7 | UC\_Calculate\_Rating |
| 2.0 | Highest | User will enter the search criteria i-e the budget to find either he can afford or not. | UC\_8 | UC\_Search |
| 2.0 | Medium | System will join all filters and make a hybrid search i-e the hotel which is nearest in budget and is having highest rating. | UC\_9 | UC\_Hybird\_Search |
|  | Medium | System will recommend the best cluster having the relevant hotels best matching to our preferences. | UC\_10 | UC\_Recommend. |
| 3.0 | Low | System “shall” display the listing of all the hotels of a specific area. | UC\_11 | UC\_Listing |
| 4.0 | Low | User “shall ” be able to give feedback | UC\_12 | UC\_Feedback |

Table 11 Prioritize requirements

## Requirement Trace-ability Matrix:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr# | Para# | System Specification Text | Build | Use Case Name | Category |
| 1 | 1.0 | User will enable his location services. The system is GPS based so location service is mandatory. | B1 | UC\_Enable\_location | Business |
| 2 | 1.0 | System will predict the actual location of the user using gps.  System will find the nearest hotels based on distance.  System will make the cluster of nearest hotels. | B2 | UC\_location\_Filter | Business |
| 3 | 1.0 | System will find the distance between the user and relevant hotels to determine which the nearest one is. | B3 | UC\_Find\_Distance | Business |
| 4 | 1.0 | System will find the optimal path to reach the location. It is shortest path to each a destination. | B4 | UC\_Optimal\_Path | Business |
| 5 | 2.0 | System will get the data of hotels from www by the process of web scraping/crawling using CheerioJS. | B5 | UC\_Get\_Data | Business |
| 6 | 2.0 | System will extract the relevant information i-e is comments on the websites. System will analyze the data to give them points i-e if the words are very good then positive number for neutral zero and bad negative points. System will combine the points to find complete rating. | B6 | UC\_Calculate\_Rating | Business |
| 7 | 2.0 | User will enter the search criteria i-e the budget to find either he can afford or not. | B7 | UC\_Search | Business |
| 8 | 2.0 | User will enter the search criteria i-e the budget to find either he can afford or not. | B8 | UC\_Hybird\_Search | Business |
| 9 | 2.0 | System will join all filters and make a hybrid search i-e the hotel which is nearest in budget and is having highest rating. | B9 | UC\_Recommend. | Business |

Table 12 Requirement Trace-ability Matrix

## High level use case diagram:

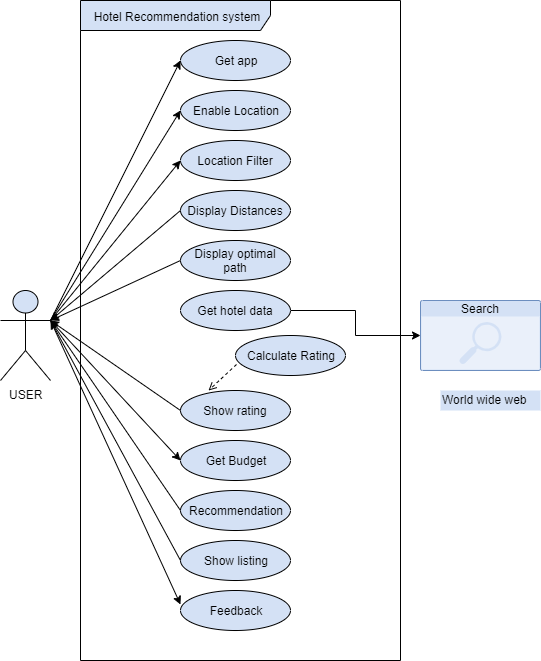


Table 13 Use case diagram

## Analysis level use case diagram:

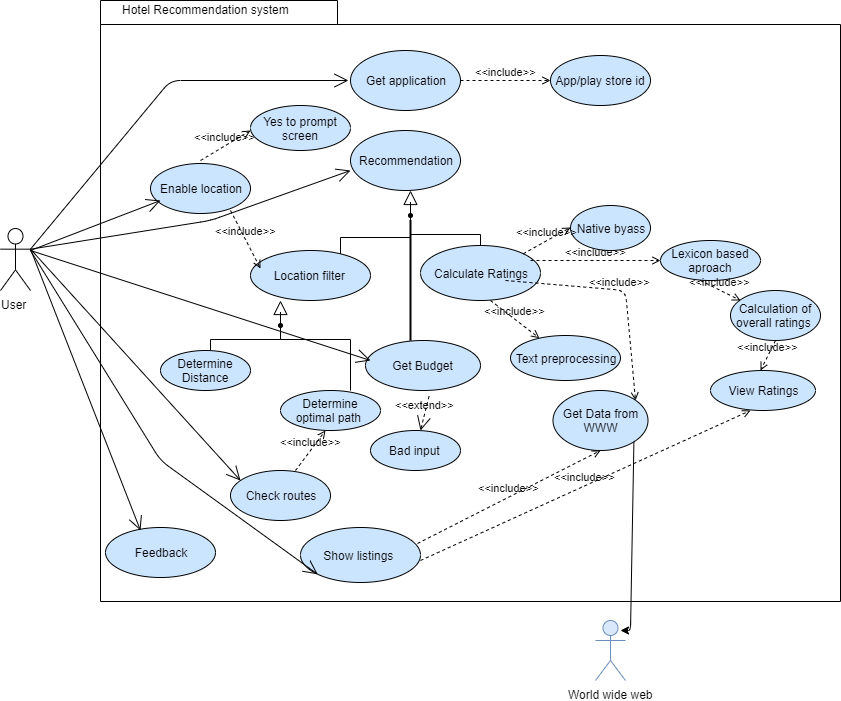


Table 14 Analysis level use case diagram

## Use case Description:

### Get Application:

|  |  |
| --- | --- |
| Use Case Id | UC\_1 |
| Use Case Name | UC\_Get\_Application |
| Description: | To get application on device |
| Primary Actor | Users |
| Pre-condition | User has an Apple id or Google account. |
| Main Success Scenario | User search for application name.  System display search results.  Select application and click install.  Prompt for id password.  Enter password.  Ok massage.  Install application |
| Alternate Process Scenario | 5.1 Enter wrong password.  6.1 Prompt to enter password again.  6.2 Forget password options show. |
| Post-Condition | User gets application on his device. |

Table 15 Get Application

### Enable location:

|  |  |
| --- | --- |
| Use Case Id | UC\_2 |
| Use Case Name | UC\_Enable\_location |
| Primary Actor | User |
| Description | User opens the application home screen. |
| Pre-condition | User sat the home screen of the application. |
| Main Success Scenario | System prompts to enable  location services   2.User clicks to enable location |
| Alternate Process Scenario | No alternate scenario |
| Post-Condition | Location service is turned on. |

Table 16 Enable location

### Location filters:

|  |  |
| --- | --- |
| Use Case Id | UC3 |
| Use Case Name | UC\_Location\_filter |
| Primary Actor | User |
| Description | To display hotels based location and distance. |
| Pre-condition | User is t the home screen and has enabled location services. |
| Main Success Scenario | User clicks at location filter  System finds the nearest cluster of the hotel.  Display cluster as a net on screen |
| Alternate Process Scenario | There is no hotel in surrounding.  System display a message “No hotel in the surrounding area” |
| Post-Condition | A hotel cluster is displayed. |

Table 17 Location filters

### Display distances:

|  |  |
| --- | --- |
| Use Case Id | UC\_4 |
| Use Case Name | UC\_Display\_Distances |
| Primary Actor | User |
| Description | To display the distance between user and hotels. |
| Pre-condition | Location filter is selected. |
| Main Success Scenario | System will find the distance between user and all the hotels in the cluster.  System will display all he paths to the hotel.  User views all the hotels along with the paths. |
| Alternate Process Scenario | No alternate flow. |
| Post-Condition | Hotels along with the distance are displayed. |

Table 18 Display distances

### Optimal path:

|  |  |
| --- | --- |
| Use Case Id | UC\_5 |
| Use Case Name | UC\_Optimal\_path |
| Primary Actor | User |
| Description | Use to find a best path to the destination. |
| Pre-condition | Location filter is enabled. |
| Main Success Scenario | User selects a specific hotel.  System will find all the paths to the hotel.  System displays the path to the hotel i-e short in distance i-e optimal path. |
| Alternate Process Scenario | No alternate flow. |
| Post-Condition | Shortest path is displayed in some other pattern. |

Table 19 Optimal path

### Get Hotel data:

|  |  |
| --- | --- |
| Use Case Id | UC\_6 |
| Use Case Name | UC\_Get\_Hotel\_data |
| Primary Actor | User |
| Description | To get the hotel data from web. |
| Pre-condition | Hotel data is available on www. |
| Main Success Scenario | Go to specified URL.  Get the xml file.  Extract the relevant data.  Store data in DB.  Enrich and display data in readable format.  User views the data |
| Alternate Process Scenario | Url not found.  Display Message “No data found”  No relevant data found.  Display error message no data found. |
| Post-Condition | Hotels data is displayed on screen. |
|  |  |

Table 20 Get Hotel data

### Calculate rating:

|  |  |
| --- | --- |
| Use Case Id | UC\_7 |
| Use Case Name | UC\_Calculate\_ratings |
| Primary Actor | User |
| Description | Calculate the rating to find the best hotel. |
| Pre-condition | Hotels data has been extracted from www. |
| Main Success Scenario | Get the comments of hotel.  Split the sentences.  Check points of the data according to trained dataset.  Assign weight to each word.  Combine all the weights to get total rank. |
| Alternate Process Scenario | No alternate flow. |
| Post-Condition | Ratings are calculated. |

Table 21 Calculate rating

### Show rating:

|  |  |
| --- | --- |
| Use Case Id | UC\_8 |
| Use Case Name | UC\_Show\_ratings |
| Primary Actor | User |
| Description | To display rating to the user. |
| Pre-condition | Ratings have been calculated. |
| Main Success Scenario | User click on view top hotels.  System display hotels in ascending order according to rank. |
| Alternate Process Scenario | No alternate flow. |
| Post-Condition | Hotels along with ratings are displayed. |

Table 22 Show ratings

### Budget Filter:

|  |  |
| --- | --- |
| Use Case Id | UC\_9 |
| Use Case Name | UC\_Budget\_filter |
| Primary Actor | User |
| Description | Display the hotel that user can afford. |
| Pre-condition | User is on budget filtering page. |
| Main Success Scenario | Prompt user to enter budget.  User enter budget.  Check hotels in range.  Display hotels according to user budget. |
| Alternate Process Scenario | User enter wrong budget.  Display Error message and prompt screen again. |
| Post-Condition | User Views hotel budget vise. |

Table 23 budget filter

### Recommendation:

|  |  |
| --- | --- |
| Use Case Id | UC\_10 |
| Use Case Name | UC\_Recommendation |
| Primary Actor | User |
| Description | Hybrid search to join all filters. |
| Pre-condition | User has selected all the filters. |
| Main Success Scenario | Join budget, rank and location filter.  Make cluster that bests suits to user preferences.  Display hotels.  View hotels. |
| Alternate Process Scenario | 1.1 Location is turned off.  1.2 Join only budget and rank.  1.3 Budget is not selected.  1.4 Use location and rank. |
| Post-Condition | Hybrid research list is displayed. |

Table 24 Recommendation

### Listings:

|  |  |
| --- | --- |
| Use Case Id | UC\_11 |
| Use Case Name | UC\_Listings |
| Primary Actor | User |
| Description | Display the list of hotels available. |
| Pre-condition | User is at hotels page. |
| Main Success Scenario | User selects a specific area.  System displays a list of hotels in area. |
| Alternate Process Scenario | No alternate flow. |
| Post-Condition | Listings are displayed. |

Table 25 Listings

### Feedback:

|  |  |
| --- | --- |
| Use Case Id | UC\_12 |
| Use Case Name | UC\_Feedback |
| Primary Actor | User |
| Description | To get user review to improve performance |
| Pre-condition | User is at feedback page. |
| Main Success Scenario | User gives feedback.  User stores them in DB. |
| Alternate Process Scenario | No alternate flow. |
| Post-Condition | Feedback has been  taken from user |

Table 26 Feed

# Chapter 3: Design Document

## Introduction:

In this document we have described the all UML diagrams that were necessary to make the system. All the data flow diagrams are ERD are also the part of this document.

The following are the UML diagrams that are described in detail:

* Domain Model
* System Sequence Diagram
* Sequence Diagram
* Collaboration Diagram
* Operation Contracts
* Design Class Diagram
* State Transition Diagram
* Data Model

Now, we will discuss them one by one:

## Domain Model:

### High level domain model:

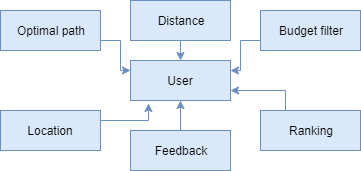


Figure 3 Domain diagram

### Analysis level Domain model:

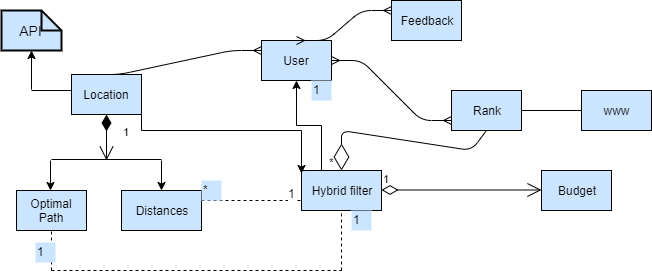


Figure 4 Analysis level domain model

## System Sequence Diagram:

Identified actors of the sequence diagrams are:

* User
* www
* database

The actors will interact the system as:

User:

* Access the application.
* Provide budget information
* Provide location.
* Provide feedback.

WWW:

* Give data to find rating of hotel.

Database:

* Store the hotel data to provide the listings.
* Store hotel data to calculate the ratings of hotel.
* Store to feedback.

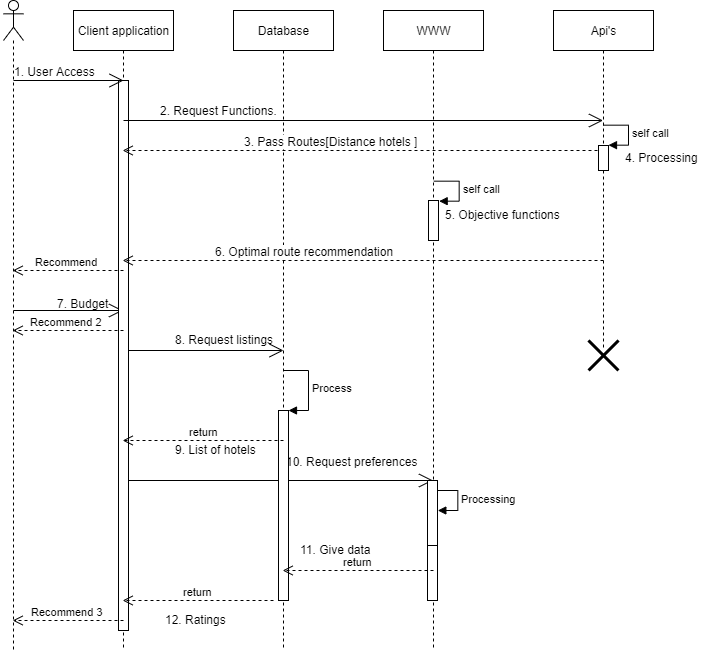


Figure 5 System sequence diagram

## Sequence Diagram:

Now, this overview diagram is divided into objects individually and the behavior of whole object is described individually.

The following are the objects of the system;

Budget

* Get data
* Location
* Feedback

### Sequence diagram of budget:

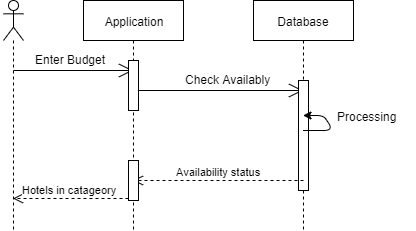


Figure 6 Sequence diagram for budget

### Sequence diagram of getting data(from www):

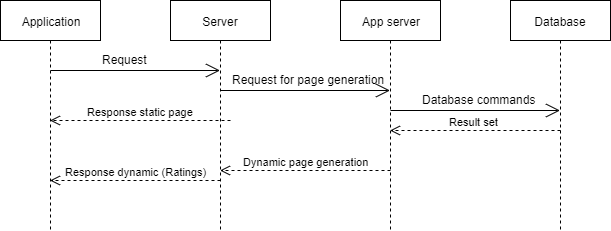


Figure 7 Sequence diagram of getting data from www

### Sequence diagram of location module:

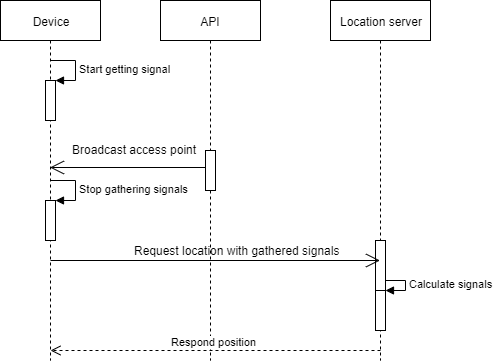


Figure 8 Sequence diagram of location

### Sequence diagram of listings:

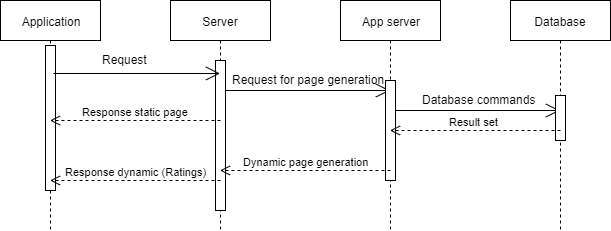


Figure 9 Sequence diagrams of listings.

### Sequence diagram of feedback:

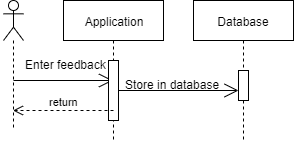


Figure 10 Sequence diagram of feedback

## Collaboration Diagram:

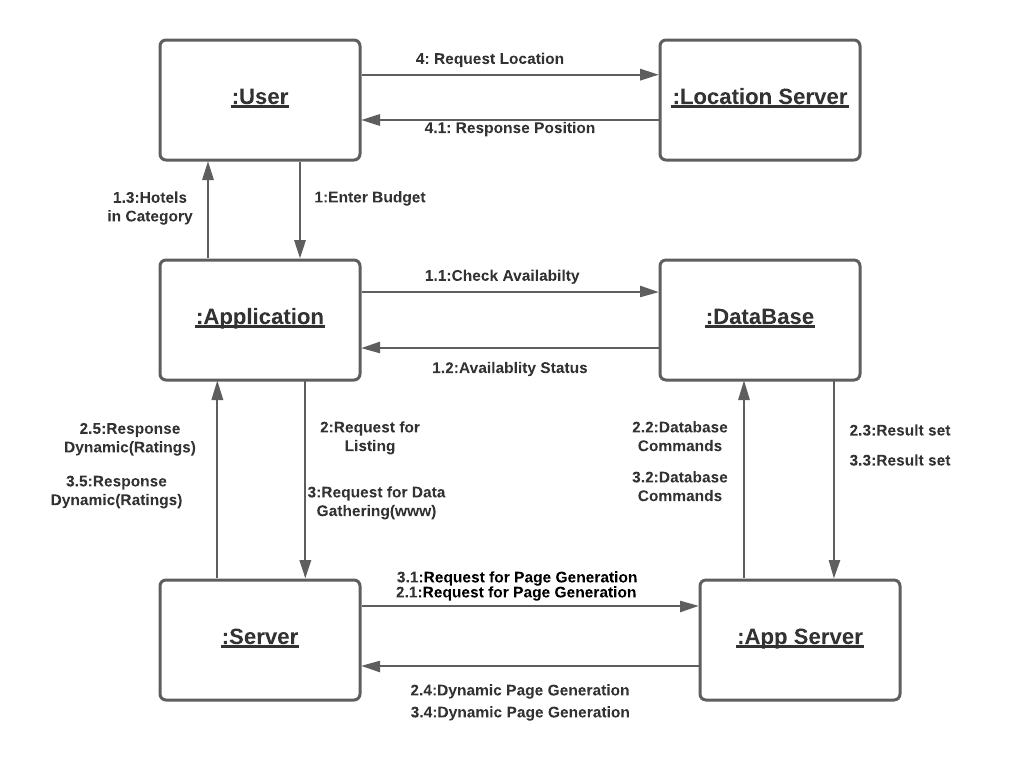


Figure 11 Collaborative diagrams

## Design Class Diagram:

### Identified classes:

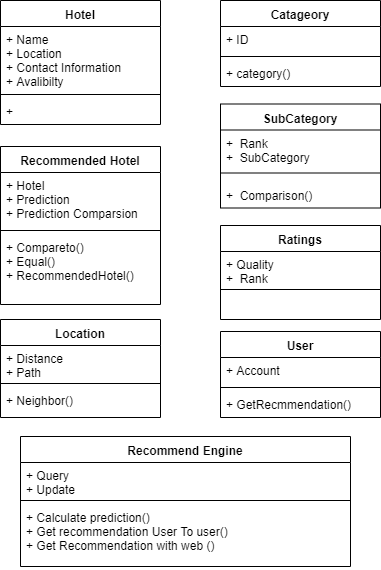


Figure 12 Classes identified

### Class diagram:

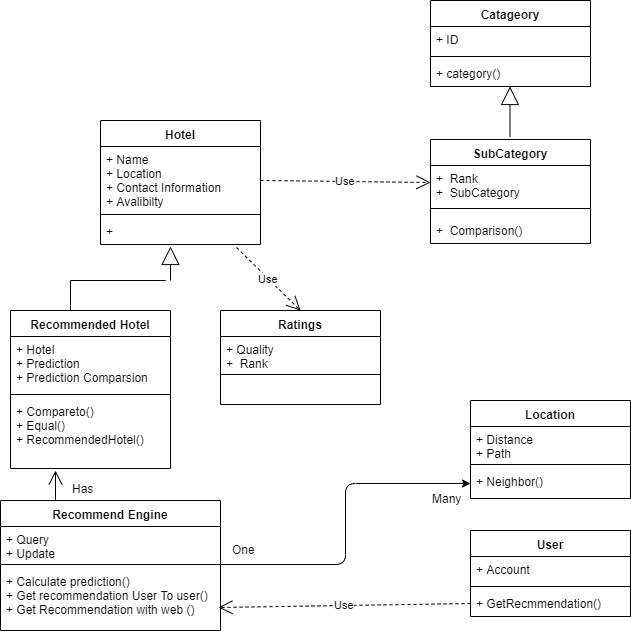


Figure 13 class diagram

## State chart diagram:

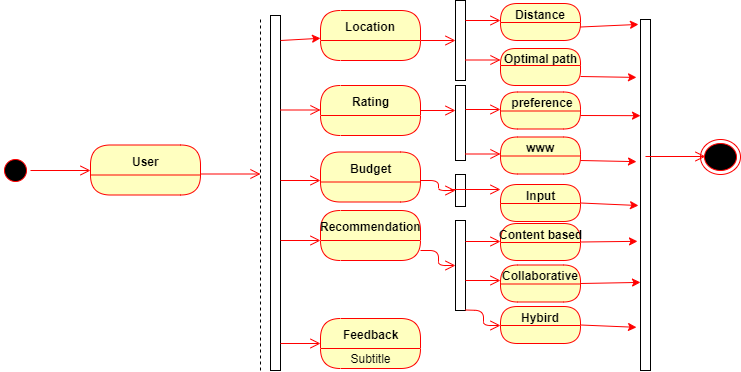


Figure 14 State chart diagram

## Activity diagram:

### Activity diagram Content based filtering:

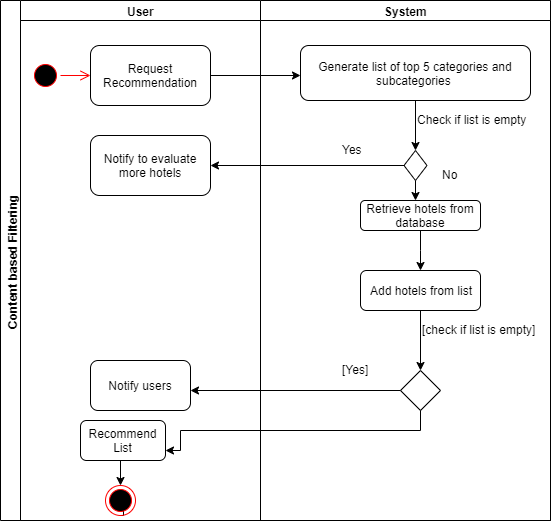


Figure 15 Activity diagram Content based filtering

### Activity Diagram of Search process:

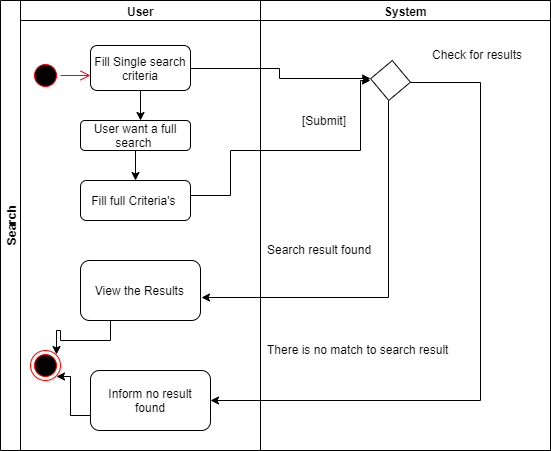


Figure 16 Activity Diagram of Search process

### Activity diagram of View hotels:

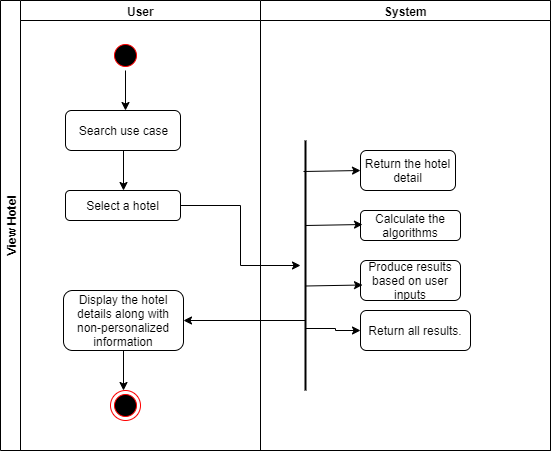


Figure 17 Activity diagram of View hotels

### Activity diagram of Collaborative searching:

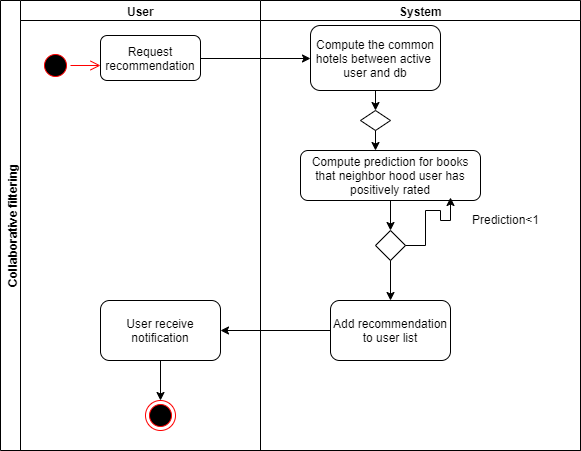


Figure 18 Activity diagram of Collaborative searching

### Activity diagram of Hybrid filter:

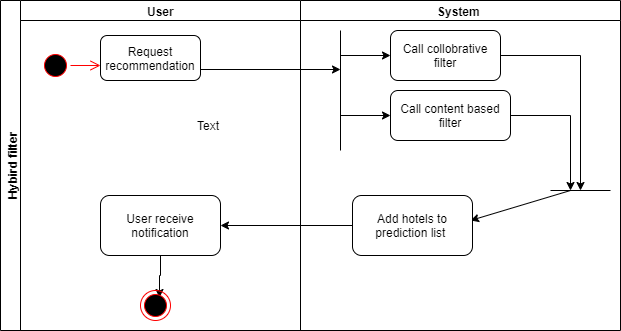


Figure 19 Activity diagram of Hybrid filter

## Package diagram:

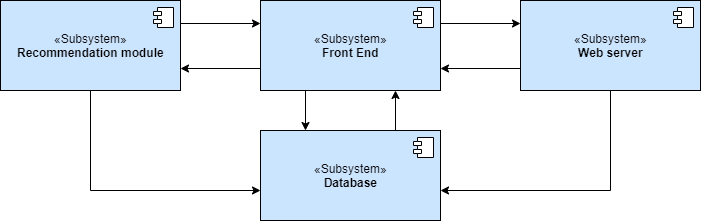


Figure 20 Package diagram

## Data model of the system:

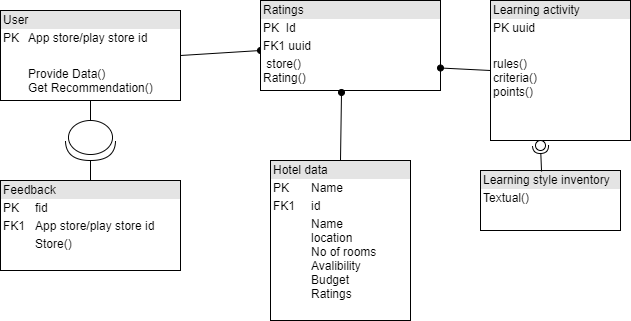


Figure 21 Data model

# 3rd Deliverable (User Interface Design)

## Introduction:

There are 3 major parts in our application. In this document we will discuss the following things:

* Site maps
* Storyboards
* Navigational maps
* Traceability Matrix

## Sitemap:

Sitemaps are representing the overall structure of the application i-e how our application is organized in a hierarchy of structure.

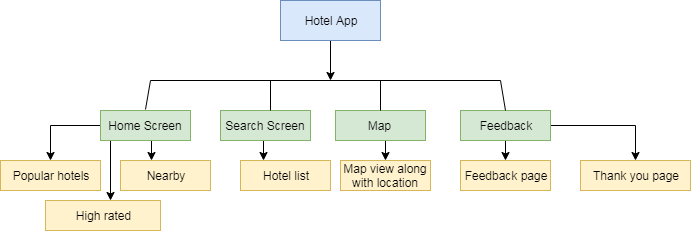


Figure 22 Site map

## Storyboards:

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | UI ID | 1 | | 1 |
| Interface |  |
| Visual Cues | This screen will pop up when the user open the app. There are four buttons in this screen to choose from. User can click on any of the option to access them. |
| Tactical Cues | Interface will display buttons for accessing top rated, nearby, budget based and filter based hotels available in nearby areas. |
| User Input | User may select any of the option above based on his will |
| System Output | System will display the screens based on the input given by the user |

Table 27 Main categories

|  |  |  |
| --- | --- | --- |
|  | | 2 |
| Interface |  | | |
| Visual Cues | The user will see the list of the hotels. User can select from the list of available restaurants. | | |
| Tactical Cues | Interface will display buttons from where user can select a hotel. Once he clicks on the hotel, further detail of the hotel will be shown on the screen. | | |
| User Input | User can select a hotel. | | |
| System Output | System will expand the details of the hotel. | | |

Table 28 Top rated

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | UI ID | 3 | | 3 |
| Interface | filter |
| Visual Cues | Three filters are available in the screen i.e. Select Price, Select Level of Popularity and select number of rooms. User will be able to select from these three options. |
| Tactical Cues | Interface will display three options from where user can select any. User can choose either one option or two even three to filter the availability of the restaurant. |
| User Input | User can select hotel based on his requirements. |
| System Output | System will search the hotels that lie in the user’s requirement. |

Table 29 Filter

|  |  |
| --- | --- |
| UI ID | 4 |
| Interface |  |
| Visual Cues | This page will display the current location of the user and the nearest hotel with the optimal path. And of course displaying the hotel which is in the budget of the user. |
| Tactical Cues | From this page user will find the nearest hotel according to their budget. |
| User Input | There is no need for the user input. |
| System Output | The system will automatically display the name of the city in which user is present at that moment and the map will display the nearest hotel to the user according to the budget. |

Table 30 Map View

|  |  |
| --- | --- |
| UI ID | 5 |
| Interface | |  | | --- | | How was your experience?  Write your review  Submit |  |  | | --- | | Thank you so much  Your Review has been submitted | |
| Visual Cues | This is the page that will allow the user to write his/her experience with the Application. After submitting their reviews, a screen will appear thanking the user for submitting the review. |
| Tactical Cues | The interface will display text box for writing their reviews. And there is one button for submitting the reviews. |
| User Input | User can enter the his/her experience with the application. |
| System Output | The system will submit the review and will display the message showing the successful submission of the experience with the thank you note. |

Table 31 Feedback

## Navigational Maps:

The next step is making navigational maps. Our application is general application and can be used without the sequence so w can use it generically as stated below:



Figure 23 Splash screen

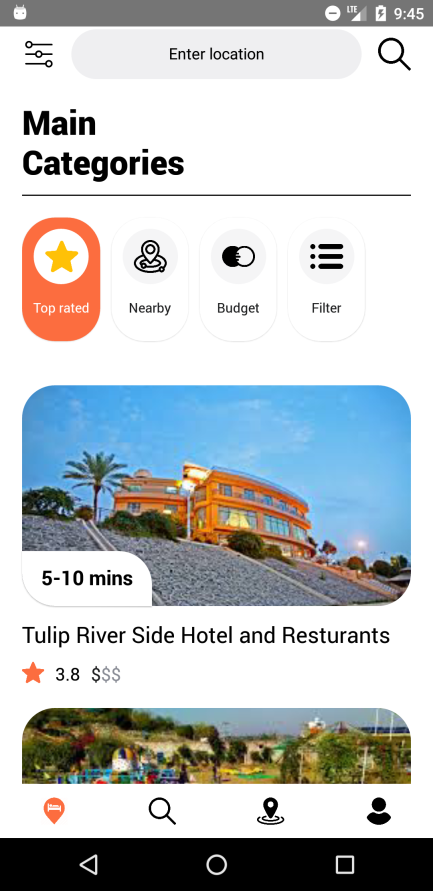


Figure 24 Home page



Figure 25 Map view

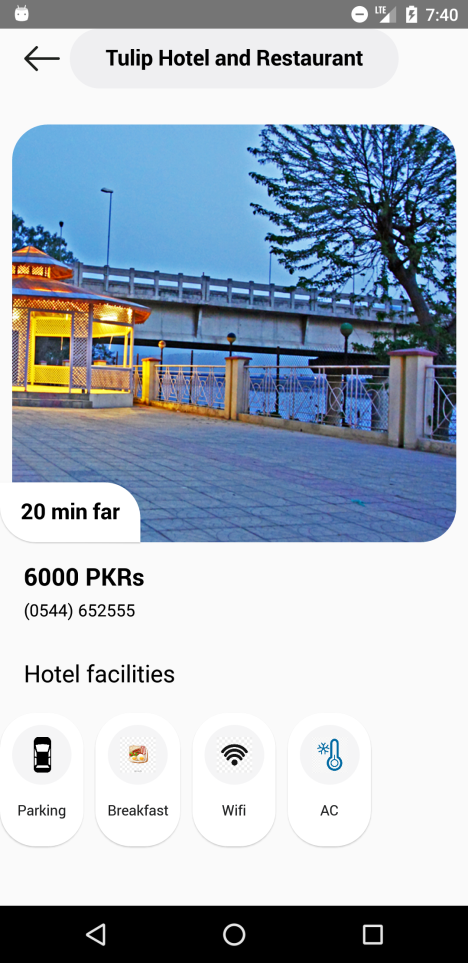


Figure 26 Hotel details page

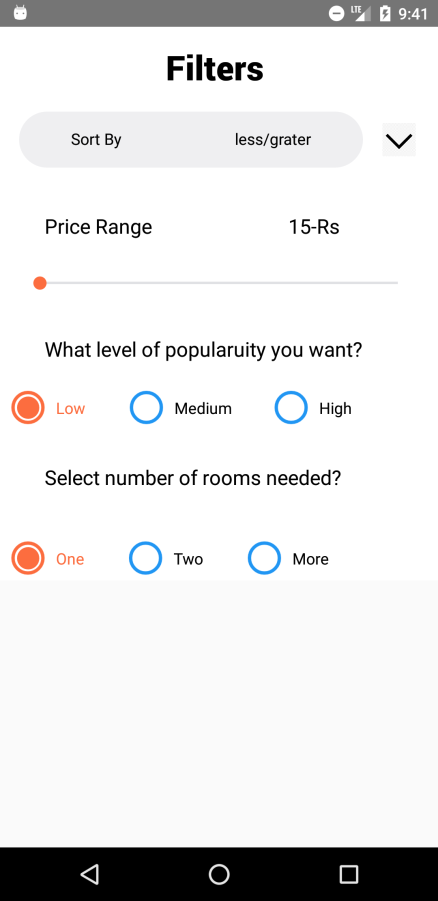


Figure 27 filter page

|  |  |
| --- | --- |
| C:\Users\Guest\Desktop\js\project\feedback.png | C:\Users\Guest\Desktop\js\project\feedback2.png |

Figure 28 Feedback pages

## Traceability matrixes:

|  |  |
| --- | --- |
| Feature | Home |
| UC\_ID | UC\_1 |
| UI ID | 1 |
| Priority | High |
| Use Case Cross Ref | UC\_GetApp |
| DB Table ID | - |
| Elaborated UC ID | User need to download the app and run it on the device. |
| Dependent Classes | User |

Table 32 Traceability Matrix of Home

|  |  |
| --- | --- |
| Feature | Home |
| UC\_ID | UC\_7 |
| UI ID | 2 |
| Priority | High |
| Use Case Cross Ref | UC\_CalculateRatings |
| DB Table ID | - |
| Elaborated UC ID | User can select the restaurant with high ratings. |
| Dependent Classes | User |

Table 33 Traceability Matrix of Home

|  |  |
| --- | --- |
| Feature | Home |
| UC\_ID | UC\_9 |
| UI ID | 3 |
| Priority | High |
| Use Case Cross Ref | UC\_HybridSearch |
| DB Table ID | - |
| Elaborated UC ID | User can select a restaurant based on his specific requirements. |
| Dependent Classes | User |

Table 34 Traceability Matrix of Home

|  |  |
| --- | --- |
| Feature | Enable Location |
| UC\_ID | UC\_2 |
| UI ID | 4 |
| Priority | High |
| Use Case Cross Ref | UC\_Enable\_location |
| DB Table ID | - |
| Elaborated UC ID | User opens the application home screen. User can enable the location to find the current location. |
| Dependent Class | User |

Table 35 Traceability Matrix of Enable Location

|  |  |
| --- | --- |
| Feature | Location Filter |
| UC\_ID | UC\_3 |
| UI ID | 4 |
| Priority | High |
| Use Case Cross Ref | UC\_Location\_filter |
| DB Table ID | - |
| Elaborated UC ID | To display hotels based on location and distance. |
| Dependent Class | Google Map |

Table 36 Traceability Matrix of Location Filter

|  |  |
| --- | --- |
| Feature | Display Distances |
| UC\_ID | UC\_4 |
| UI ID | 4 |
| Priority | High |
| Use Case Cross Ref | UC\_Display\_Distances |
| DB Table ID | - |
| Elaborated UC ID | To display the distance between user and hotels. |
| Dependent Class | Google Map |

Table 37 Traceability Matrix of Display Distances

|  |  |
| --- | --- |
| Feature | Optimal Path |
| UC\_ID | UC\_5 |
| UI ID | 4 |
| Priority | High |
| Use Case Cross Ref | UC\_Optimal\_path |
| DB Table ID | - |
| Elaborated UC ID | Use to find a best path to the destination. |
| Dependent Class | Google Map |

Table 38 Traceability Matrix of Optimal Path

|  |  |
| --- | --- |
| Feature | Feedback |
| UC\_ID | UC\_12 |
| UI ID | 5 |
| Priority | High |
| Use Case Cross Ref | UC\_Feedback |
| DB Table ID | - |
| Elaborated UC ID | To get user review to improve performance |
| Dependent Class | User |

Table 39 Traceability Matrix of Feedback

# 4th Deliverable (Testing document)

## Introduction:

In the beginning of the software engineer there are multiple testing templates and number of testing schemes has been invented. Which are used in both User Interfaces and for the Unit Testing?

Every reliable and trustworthy software houses have advanced testing documentation for making the competent systems.

For this IEEE has developed the 829 Standard for the testing purpose which are used for any kind of testing in which User Acceptance testing is included.

In our case we are going to test each module and function of the system using the template of the IEEE829.

## Test Plan:

### Purpose:

The main purpose of the test plan for the Hotel selection based on hybrid model using AI is to discuss the testing details of the use cases of the application. The test plan for the project defines the objective, even scope and approach of the testing effort. According to the features of the system this testing plan checks the input and output of the system. Hotel selection based on hybrid model using AI test plan shows the people responsibility for all the tasks and also defines the risks involved in test plan.

#### Outline:

Following structure should be added in test plan:

* Test plan identifier
* Introduction
* Test items
* Features to be tested
* Features not to be tested
* Approach
* Item pass/fail criteria
* Suspension criteria and resumption requirements
* Test deliverables
* Testing tasks
* Responsibilities
* Staffing and training needs
* Schedule
* Risks and contingencies
* Approvals

Detail of each section is given in the following sub-clauses.

**Test plan identifier:**

Test Plan Identifier: ‘Hotel selection based on hybrid model using AI (React Native Application)’.

**Introduction:**

Testing phase is the core part of the software test plan because other phases depend on it in one way or other.

This test plan for Hotel selection based on hybrid model using AI testing supports the following objectives:

The identification of the functions of the system that need to be tested.

The identification of the functions of the system that need not to be tested.

Testing items Pass and fail conditions.

In order to communication with the responsible parties, items need to be tested, and fix prospects and describe the environmental needs.

In order to define how the tests will be accompanied.

**Test items:**

The items to be tested include upload Hotels listing, view Hotels listing, check recommendation engine, give feedback, storing feedbacks, send notification, received notifications, enable location, get current location using map, show property description, get list of nearest Hotels, calculate ratings, budget match, finding hotels, hybrid filtering, collaborative filtering.

**Features to be tested:**

* Upload Hotels listing
* View Hotels listing
* Check recommendation engine
* Give feedback
* Storing feedbacks
* Send notification
* Received notifications
* Enable location
* Get current location using map
* Show property description
* Get list of nearest Hotels
* Calculate ratings
* Budget match
* Finding hotels
* Hybrid filtering

**Features not to be tested:**

In this test plan identifier, the following features will not be tested.

Logo and background animation

**Approach:**

As per Test cases defined in the document the tests will be accompanied. Since the system we developed is for the common people and is used by the users (who visit different cities more often) and such people filled the questionnaire and after that each of the member of the group will test individual feature and stain the case as Pass/Fail. People involved in testing will wrote down the actual result and all the related information. Lastly when all the tests will be completed, test report will be reviewed by the test manager.

**Item pass/fail criteria:**

The test cases implemented on the Hotel selection based on hybrid model using AI System will pass/permit only if the specified requirements meet mentioned in the project. A test case is supposed to fail/flop, if the functionality desired by the system is not satisfied. The system should work as expected and as drawn in each test case. There must be 95% passed test cases and the user’s ability to use the system should not be affected by the failed test cases.

**Item Pass Criteria**

* iOS 11.0 + and Android 5.0 +
* Ram 4GB
* Valid Inputs
* Fulfill Define conditions

**Item Fail Criteria**

* Not iOS 11.0 + and Android 5.0 +
* Not Ram 4GB
* Not Valid Inputs
* Not Fulfill Define conditions

**Suspension criteria and resumption requirements:**

In case any testing process of the system is not working according to the specified requirements then testing should stopped instantly.

**Suspension Criteria**: in case of single test case failure the testing for the dependent functions must stopped. The failed case will be recorded in test log which covers the errors description.

**Resumption Requirement**: independent test cases where the error is testified will be performed in parallel with the error fixing. Once test case which failed is recorded and after the identification and fixation the testing for that test case will resume.

**Test deliverables:**

* The following deliverable will be providing after the testing.
* Test plan;
* Test design specifications;
* Test case specifications;
* Test procedure specifications;
* Test item transmittal reports;
* Test logs;
* Test incident reports;
* Test summary reports

**Testing tasks:**

* The given activities must be accomplished:
* Test plan organized.
* Specified function inscribed and sent to the testing team.
* For testing the environment must be prepared.
* Complete the Tests.
* Make Test summary report.

**Responsibilities:**

Gathering information, designing document, preparing document, implementing the system correctly are the responsibilities of each member. Testing, handling and solving the problems of the project are responsibility of the test manager.

**Staffing and training needs**

Two or more testers must test the system. And testers must have applied the testing individually. The tester must have the general knowledge about the given expressions.

* Firebase
* React Native
* Js

**Schedule:**

The testing acquired about one week in order to complete. Some modules and their functions had been tested in a day.

**Risks and contingencies:**

In case the initial module is not completed in a day, it will be delayed. And after the error fixation the testing will be executed. If tester does not have the general understanding related to android studio, firebase and java language then it will lead to the delay and poor conduction of testing.

**Approvals**

Overall test strategy must be approved by each member.

Neha Durani

Saira Irtaza

Zain Bashir

## Test design specification:

### Purpose:

This section includes the designing and the prediction about the module which we are going to test. In this stage the prioritization takes place in test cases in order of the significance. It is important but is neglected by the most of the organization. This is the major mistake that organization implement the tests without their designing.

### Outline:

Following structure should be added in test design specification:

* Test plan identifier
* Introduction
* Test items
* Features to be tested
* Features not to be tested
* Approach
* Item pass/fail criteria
* Suspension criteria and resumption requirements
* Test deliverables
* Testing tasks
* Environmental needs
* Responsibilities
* Staffing and training needs
* Schedule
* Risks and contingencies
* Approvals

**Test design specification identifier**

The identifier for the test design specification is **Hotel selection based on hybrid model using AI**

**Introduction:**

The main purpose of the test plan for the Hotel selection based on hybrid model using AI is to discuss the testing details of the use cases of the application. The test plan for the project defines the objective, even scope and approach of the testing effort. According to the features of the system this testing plan checks the input and output of the system. Hotel selection based on hybrid model using AI test plan shows the people responsibility for all the tasks and also defines the risks involved in test plan.

This test plan for Hotel selection based on hybrid model using AI testing supports the following objectives:

The identification of the functions of the system that need to be tested.

The identification of the functions of the system that need not to be tested.

Testing items Pass and fail conditions.

In order to communication with the responsible parties, items need to be tested, and fix prospects and describe the environmental needs.

In order to define how the tests will be accompanied.

**Test items:**

The items to be tested include upload Hotels listing, view Hotels listing, check recommendation engine, give feedback, storing feedbacks, send notification, received notifications, enable location, get current location using map, show property description, get list of nearest Hotels, calculate ratings, budget match, finding hotels, hybrid filtering, collaborative filtering.

**Features to be tested:**

* Upload Hotels listing
* View Hotels listing
* Check recommendation engine
* Give feedback
* Storing feedbacks
* Send notification
* Received notifications
* Enable location
* Get current location using map
* Show property description
* Get list of nearest Hotels
* Calculate ratings
* Budget match
* Finding hotels
* Hybrid filtering
* Collaborative filtering.

**Features not to be tested:**

In this test plan identifier, the following features will not be tested.

Logo and background animation

**Approach:**

As per Test cases defined in the document the tests will be accompanied. Since the system we developed is for the common people and is used by the users (who visit different cities more often) and such people filled the questionnaire and after that each of the member of the group will test individual feature and stain the case as Pass/Fail. People involved in testing will wrote down the actual result and all the related information. Lastly when all the tests will be completed, test report will be reviewed by the test manager.

**Item pass/fail criteria:**

The test cases implemented on the Hotel selection based on hybrid model using AI System will pass/permit only if the specified requirements meet mentioned in the project. A test case is supposed to fail/flop, if the functionality desired by the system is not satisfied. The system should work as expected and as drawn in each test case. There must be 95% passed test cases and the user’s ability to use the system should not be affected by the failed test cases.

**Item Pass Criteria**

* iOS 11.0 + and Android 5.0 +
* Ram 4GB
* Valid Inputs
* Fulfill Define conditions

**Item Fail Criteria**

* Not iOS 11.0 + and Android 5.0 +
* Not Ram 4GB
* Not Valid Inputs
* Not Fulfill Define conditions

**Suspension criteria and resumption requirements:**

In case any testing process of the system is not working according to the specified requirements then testing should stopped instantly.

**Suspension Criteria**: in case of single test case failure the testing for the dependent functions must stopped. The failed case will be recorded in test log which covers the errors description.

**Resumption Requirement**: independent test cases where the error is testified will be performed in parallel with the error fixing. Once test case which failed is recorded and after the identification and fixation the testing for that test case will resume.

**Test deliverables:**

The following deliverable will be providing after the testing.

* Test plan;
* Test design specifications;
* Test case specifications;
* Test procedure specifications;
* Test item transmittal reports;
* Test logs;
* Test incident reports;
* Test summary reports

**Testing tasks:**

* The given activities must be accomplished:
* Test plan organized.
* Specified function inscribed and sent to the testing team.
* For testing the environment must be prepared.
* Complete the Tests.
* Make Test summary report.

**Environmental needs:**

As stated prior we are going to execute testing in websites using unit test and instrumentation test. For this purpose, both these types of tests need changed boards which are as:

**Unit Testing**

For implementation of unit test, we must have Jest to be installed in our computer. Jest is erected into React, and Jest is the simple solution for unit testing in React native.

**UI Testing**

We need Appium Framework to smoothly run UI tests.

**Responsibilities:**

Gathering information, designing document, preparing document, implementing the system correctly are the responsibilities of each member. Testing, handling and solving the problems of the project are responsibility of the test manager.

**Staffing and training needs**

Two or more testers must test the system. And testers must have applied the testing individually. The tester must have the general knowledge about the given expressions.

* Firebase
* React Native
* Js

**Schedule:**

The testing acquired about one week in order to complete. Some modules and their functions had been tested in a day.

**Risks and contingencies:**

In case the initial module is not completed in a day, it will be delayed. And after the error fixation the testing will be executed. If tester does not have the general understanding related to android studio, firebase and java language then it will lead to the delay and poor conduction of testing.

**Approvals**

All the team members need to approve the overall test strategy.

Neha Durani

Saira Irtaza

Zain Bashir

## Test design specifications:

### Purpose:

This section will describe the list of the features that we are going to test and also we will prioritize the test cases based on their importance.

### Outlines:

The contents that will describe the test case specification will include the following:

* Test case specification identifier
* Test items
* Input specifications
* Output specifications
* Environmental needs
* Special procedural requirements
* Inter case dependencies

### Test cases:

#### Test case specification for Upload hotel listings:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 1 |
| Test Item | Add and save hotels in database |
| Input specification | 1. Enter the hotel data in database. 2. Enter save button. |
| Output specification | The hotel data get saved. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | No |

Table 40 TC of Upload hotel listings

#### Test case specification for view hotel listing:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 2 |
| Test Item | Hotels show or not |
| Input specification | 1. Open the hotel listing page. |
| Output specification | Display list of hotels |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | TC1 |

Table 41 TC of view hotel listing

#### Test case specification for budget filter:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 3 |
| Test Item | Hotels are displayed according to budget. |
| Input specification | Select budget button. |
| Output specification | Display the hotels in ascending order according to Budget. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | From collaborative filter page enter starting and ending budget.  Click okay.  System will display the results accordingly. |
| Inter case dependencies | TC6 |

Table 42 TC of budget filter

#### Test case specification for rating filter:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 4 |
| Test Item | Hotels are displayed according to rating or not |
| Input specification | Select the rating filter button. |
| Output specification | Display the hotels in ascending order according to rating. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | No |

Table 43 rating filter

#### Test case specification for nearest filter:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 5 |
| Test Item | Nearest hotels according to area are displayed or not. |
| Input specification | Click the nearest filter button. |
| Output specification | Display the Hotel that is close to user location. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | TC7 |

Table 44 TC for nearest filter

#### Test case specification for collaborative filtering:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 6 |
| Test Item | All filters are working together correctly or not. |
| Input specification | 1. Enter the preference I-e budget and location. 2. Click search button. |
| Output specification | Display the result matching preferences. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | No |

Table 45 TC for collaborative filtering

#### Test case specification for enables location:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 7 |
| Test Item | Get the user current location. |
| Input specification | 1. From the task bar open map view. 2. Select enable location button from prompt screen. 3. Click ok. |
| Output specification | Display the map on the screen. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | No |

Table 46 TC for enables location

#### Test case specification for display map view:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 8 |
| Test Item | Display hotels in database on map according to filters. |
| Input specification | 1. From the task bar open map view. |
| Output specification | 1. Display the map view to user. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | TC7 |

Table 47 TC for display map view

#### Test case specification for hotel cluster:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 9 |
| Test Item | Minimum 5 hotel cluster is displaying or not. |
| Input specification | 1. From the task bar open map view. |
| Output specification | 1. Display the cluster of 5 nearest hotels. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | TC7 , TC8 |

Table 48 TC for hotel cluster

#### Test case specification for shortest path:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 10 |
| Test Item | Distance from user to hotel is displayed or not. |
| Input specification | 1. From the task bar open map view. |
| Output specification | 1. For cluster hotel path towards the hotel is displayed. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | TC9 |

Table 49 TC for shortest path

#### Test case specification for hotels detail:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 11 |
| Test Item | Hotel detail page is displaying data correctly. |
| Input specification | 1. Select a specific hotel either from hotel listing or from map view. |
| Output specification | 1. Hotel details page with the full data is displayed. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | TC7, TC 2 |

Table 50 TC for hotels detail

#### Test case specification for feedback:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test case ID | TC 12 |
| Test Item | User can give feedback and feedback gets saved. |
| Input specification | 1. Select star rating. 2. Enter some comments. 3. Click on submit button. |
| Output specification | 1. Display appropriate thank you message. 2. Save the feedback in the database. |
| Test case environment | Hardware: Android/ IOS  Software: Android 4+ / IOS 7+ |
| Special procedural requirements | No |
| Inter case dependencies | No |

Table 51 TC for feedback

## Test procedure specifications:

### Purpose:

The purpose of this section is to describe the procedure of the tests that we are going to perform. The major purpose is to identify the features are working according to specified by the user.

### Outlines:

The following items are going to be the part of our testing specifications:

* 1. Test procedure specification identifier
  2. Purpose
  3. Special requirements
  4. Procedure steps

### Test procedures:

#### Test procedure specification for Upload hotel listings:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP1 |
| Purpose | This test case check if the entered data get stored in database or not. |
| Special requirements | No |
| Procedure | 1. Open the firebase console. 2. Enter full detail of the hotel or hotels. 3. Click okay button. 4. Save changes 5. Data get stored. |
| Test identification number | TC 1 |
| Status | Passed |

Table 52 TP for Upload hotel listings

#### Test procedure specification for View hotel listings:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP2 |
| Purpose | To check whether the entered hotels data get displayed on application. |
| Special requirements | No |
| Procedure | 1. Open the application. 2. Click on home page. 3. On search bar enter name of entered hotel. 4. Data get displayed on screen. |
| Test identification number | TC 2 |
| Status | Passed |

Table 53 TP for View hotel listings

#### Test procedure specification for budget filter:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP3 |
| Purpose | To authenticate that budget filter is working properly. |
| Special requirements | No |
| Procedure | 1. Open the application. 2. Go to home page if is on other page. 3. Click the budget filter button. 4. Data get displayed. |
| Test identification number | TC3 |
| Status | Passed |

Table 54 TP for budget filter

#### Test procedure specification for rating filter:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP4 |
| Purpose | To authenticate that rating filter is working properly. |
| Special requirements | No |
| Procedure | 1. Open the application. 2. Go to home page if is on other page. 3. Click the rating filter button.   Data get displayed. |
| Test identification number | TC4 |
| Status | Passed |

Table 55 TP for rating filter

#### Test procedure specification for nearest filter:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP5 |
| Purpose | To authenticate that nearest hotel button is working properly. |
| Special requirements | No |
| Procedure | 1. Open the application. 2. Go to home page if is on other page. 3. Click the nearest hotel button. 4. Data get displayed. |
| Test identification number | TC 5 |
| Status | Passed |

Table 56 for nearest filter

#### Test procedure specification for collaborative filtering:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP6 |
| Purpose | To authenticate that collaborative filtering is working properly |
| Special requirements | No |
| Procedure | 1. Open the application. 2. Go to collaborative filter page if is on other page. 3. Enter start and end budget. 4. Enter other preferences 5. Click search button. 6. Data get displayed. |
| Test identification number | TC 6 |
| Status | Passed |

Table 57 TP for collaborative filtering

#### Test procedure specification for enables location:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP7 |
| Purpose | To check if the location is enabled on selecting the location module. |
| Special requirements | No |
| Procedure | 1. Open application. 2. Go to location page. 3. Allow enable location. 4. Result gets displayed. |
| Test identification number | TC7 |
| Status | Passed |

Table 58 TP for enables location

#### Test procedure specification for display map view:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP8 |
| Purpose | To check maps are working properly. |
| Special requirements | No |
| Procedure | 1. Open application. 2. Go to location page. 3. Allow enable location. 4. Result gets displayed. |
| Test identification number | TC 8 |
| Status | Passed |

Table 59 TP for display map view

#### Test procedure specification for hotel cluster:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP9 |
| Purpose | To check clustering algorithm is working properly. |
| Special requirements | No |
| Procedure | 1. Open application. 2. Open map page. 3. System display cluster. |
| Test identification number | TC 9 |
| Status | Passed |

Table 60 for hotel cluster

#### Test procedure specification for shortest path:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP 10 |
| Purpose | To check whether system is representing the shortest path towards the hotel or not. |
| Special requirements | No |
| Procedure | 1. Open application. 2. Open map page. 3. System display cluster. 4. System display location towards the hotel. |
| Test identification number | TC 10 |
| Status | Passed |

Table 61 TP for shortest path

#### Test procedure specification for hotels detail:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP 11 |
| Purpose | To check system display details in proper way and of respective hotel. |
| Special requirements | No |
| Procedure | 1. Select hotel from either map or list view. 2. System displays the respective results. |
| Test identification number | TC 11 |
| Status | Passed |

Table 62 TP for hotels detail

#### Test procedure specification for feedback:

|  |  |
| --- | --- |
| Test engineer | Neha Durani , Saira Irtaza , Zain Bashir |
| Test procedure ID | TP 12 |
| Purpose | To validate either the system takes and store feedback in proper format. |
| Special requirements | No |
| Procedure | 1. Open application. 2. Go to feedback corner. 3. Select number of stars you want to give. 4. Enter comments. 5. Select enter button. 6. System display thank you message to user and store in firebase. |
| Test identification number | TC 12 |
| Status | Passed |

Table 63 TP for feedback:

## Test item transmittal report:

### Purpose:

This section is describing the person who is responsible for transmitting the test items. It includes the person responsible for each item, its physical location, and its status. Any changes from the present item requirements and designs are noted in this report.

### Outlines:

A test item transmittal report shall have the following structure:

1. Transmittal report identifier
2. Transmitted items
3. Location
4. Status
5. Approvals

Table 64 Transmittal report

### Transmittal report

|  |  |
| --- | --- |
| Transmittal report identifier | The identifier of the transmittal report is “Hotel recommendation system using AI” |
| Transmitted items | All the features of our system are working properly and we (team members) are responsible for any error that will occur. |
| Location | The system is native application and will run on both android and IOS device. |
| Status | After testing all the tests are passed and system is working as expected. |
| Approvals | This report is made after the approval of whole testing team |

## *Test Log*

### Purpose

The purpose of this is to provide a chronological record of the details about execution of tests.

### Outline

Following is the content of Test Log

* Test Log Identifier
* Description
* Activity and event entries

#### Test Log Identifier

The identifier for our Test Log is HS\_UAI

#### Description

Following is the detail of all the test items which are planned to be tested and will also be submitted to test manager.

|  |  |  |
| --- | --- | --- |
| Serial No | Test Features | Test Case Identifier |
| 1 | Add and save hotels in database | TC1 |
| 2 | Hotels show or not | TC2 |
| 3 | Hotels are displayed according to budget. | TC3 |
| 4 | Hotels are displayed according to rating or not | TC4 |
| 5 | Nearest hotels according to area are displayed or not | TC5 |
| 6 | All filters are working together correctly or not | TC6 |
| 7 | Get the user current location. | TC7 |
| 8 | Display hotels in database on map according to filters | TC8 |
| 9 | Minimum 5 hotel cluster is displaying or not. | TC9 |
| 10 | Distance from user to hotel is displayed or not. | TC10 |
| 11 | Hotel detail page is displaying data correctly. | TC11 |
| 12 | User can give feedback and feedback gets saved. | TC12 |

Table Test Features

**Software Requirements**

The detail of software requirement is given below:

* Emulator (Android)
* Simulator (IOS)
* Hardware Requirements
* The detail of hardware requirement is given below:
* Emulator/Simulator
* Android Device
* IOS Device

#### Execution Description

Due to ongoing pandemic, we executed the tests using online platform i.e. Microsoft Teams so that all members of the team can get through the understanding of this testing process.

The tests were executed on 26 April 2021 on all available options i.e. Emulator, Simulator, Android device and IOS device.

#### Procedure Results

Following is the complete detail of execution of test based on fail/pass criteria already mentioned in planning phase.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No | Test Case Identifier | Test Revisions | Test Case Result | Test Status |
| 1 | TC1 | 1st | Add and save hotels in database | Pass |
| 2 | TC2 | 1st | Hotels show or not | Pass |
| 3 | TC3 | 1st | Hotels are displayed according to budget. | Pass |
| 4 | TC4 | 1st | Hotels are displayed according to rating or not | Pass |
| 5 | TC5 | 1st | Nearest hotels according to area are displayed or not | Pass |
| 6 | TC6 | 1st | All filters are working together correctly or not | Pass |
| 7 | TC7 | 1st | Get the user current location. | Pass |
| 8 | TC8 | 1st | Display hotels in database on map according to filters | Pass |
| 9 | TC9 | 1st | Minimum 5 hotel cluster is displaying or not. | Pass |
| 10 | TC10 | 1st | Distance from user to hotel is displayed or not. | Pass |
| 11 | TC11 | 1st | Hotel detail page is displaying data correctly. | Pass |
| 12 | TC12 | 1st | User can give feedback and feedback gets saved. | Pass |

Table Procedure Results

#### Environmental Information

* Samsung Galaxy J6
* I Phone 6S Plus
* Emulator and Simulator

#### Anomalous Event

We successfully conducted the tests and we did not find any noticeable anomaly except one. The Google maps were not loading on our android emulator as well as IOS simulator.

#### Incident Report Identifier

Record identifier of every test incident report, when one is generated.

## *Test Incident Report*

### Purpose

The purpose of test incident report is to record any event that has occurred during testing process and it requires further investigation.

### Outline

A test incident report shall have the following structure:

* Test incident report identifier
* Summary
* Incident description
* Impact

#### Test Incident Report Identifier

The identifier for incident report is HS\_UAI

#### Summary

During the execution of the tests, we found an anomaly that Google maps were not working properly. Our app was unable to load Google maps on android emulator. We also faced the same problem when we tested our application on IOS simulator. In the following section we will investigate the reason behind this failure.

#### Incident Description

Here is the complete description of the anomaly that occurred during the testing of the application.

|  |  |  |  |
| --- | --- | --- | --- |
| Inputs | Expected Result | Actual Result | Anomalies |
| Check hotels based on maps | Data shown successfully | Unable to load Google maps | After investigation we found that we found that we were using older versions of emulator as well simulator. |

Table Anomaly

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date and  Time | Environment | Attempt to Repeat | Tester | Observer |
| 26, April,  2021 | Android Emulator  Simulator | After correcting the anomaly, test case is dispatched for refactoring | Neha Durani | Zain Bashir |

Table Solution to anomaly

#### Impact

After testing the system, we are able to find out bugs and fixes and their solution.

## *Test Summary Report*

### Purpose

The purpose of this is to summarize the results of the testing activities and to provide evaluation based on the results obtained from testing.

#### Outline

A test summary report shall have the following structure:

* Test summary report identifier
* Summary
* Variances
* Comprehensive assessment
* Summary of results
* Evaluation
* Summary of activities
* Approvals

#### Test Summary report identifier

Test summary report identifier is as HS\_UAI.

#### Summary

All the items described in the test plan were tested and after testing we found no anomaly. We used pass/fail criteria of test cases according to the criteria mentioned above in test planning phase. After noticing that one of the test case had failed, so we prepared its incidental report so that we can refractor by the developer and send again. After refactoring test cases which had defect, we again tested it and it passed our testing criteria.

#### Variances

Every test is executed according to the test plan, test specification and test case specification. We follow the testing approach and strategy as stated in the test planning phase. Test was executed according to test procedure specification and they are deemed passed and failed according to testing criteria mentioned.

#### Comprehensive assessment

All the features described in the test planning phase are tested and no test case is left out of testing due to any sort of reason. Our test coverage report is more than 80 percent.

#### Summary of results

We encountered a bug as mentioned earlier. The Google maps were not working properly with emulator. But we figured out the solution to this problem and now every feature in the application is working properly.

#### Evaluation

We have covered almost 80 percent code with unit and UI test. Moreover, we used test driven approach in our code. Out of twelve test cases only one test case gets failed which were again refectories. So, our team including test planner, test designer and test manager work in great harmony to execute all phases of testing.

#### Summary of activities

There are three development phases in testing. The first one was test planning followed by test specification and designing and test execution. All these test activities and phases are planned and executed according to Gantt chart. Here is the breakdown of all the activities with their time and staff who was responsible

|  |  |  |
| --- | --- | --- |
| Testing Activities | Person Responsible | Time Elapsed |
| Test Planning | Zain Bashir | 5 days |
| Test Designing | Neha Durani | 7 days |
| Test Execution | Saira Irtaza | 2 days |

#### Approvals

We need approval of test planner, test designer and test manager.

**Zain Bashir\_\_\_\_\_\_\_\_\_\_\_**

**Neha Durani\_\_\_\_\_\_\_\_\_\_**