Session: 2022-2023

Department of Computer Science

Shaheed Sukhdev College of Business Studies

University of Delhi

Travandoz : A Tour Planner Application

Software Engineering Project Report

(CHSP – 410)

Submitted By:

Aman Sachdeva (21506)

Anshu Kumar (21510)

Rishav Raj(21550

Submitted To:

Mrs. Kavita Rastogi

Associate Professor, SSCBS, DU

INDEX

Problem Statement 4

Process Model 5

Software Requirement Specification 6

1. **Introduction** 6

1.1 Introduction 6

1.2 Purpose 6

1.3 Scope 6

1.4 Acronyms, Definitions and Abbreviations 7

2. **Overall Description** 7

2.1 Description 7

2.2 Product's perspective 7

2.3 Product's function 8
- 2.3.1 Admin
- 2.3.2 User
- 2.3.3 Host

2.4 User Characteristics 11

2.5 Constraints 11

3. **Requirements and Functional description** 13

3.1 Functional Requirements 13
- 3.1.1 Use Case Diagram
- 3.1.2 Use Case Scenario

3.2 Non Functional requirements 20
- 3.2.1 Reliability
- 3.2.2 Security
- 3.2.3 Efficiency
- 3.2.4 Availability

3.2.5 Maintainability
3.2.6 Portability

3.3 Data Flow Diagram 21
3.3.1 Context Level
3.3.2 Level-1
3.3.3 Level2

3.4 Data Dictionary 24

4 External Interface Requirements 25

4.1 User Interfaces 27

5.Estimations 29

5.1 Function Points

6.Scheduling 32

7.Risk Management 33

7.1 Risk Table

7.2 RMMM Plan

8.Data Design 35

9.Coding 38

10. Testing 40

PROBLEM STATEMENT

Travandoz is a startup company that aims to provide an online platform for users to discover and book accommodations for their travel needs. The company has identified that there is a gap in the market for a platform that offers a wide range of accommodation options, including hotels, hostels, guesthouses, and vacation rentals, all in one place. Additionally, the company wants to differentiate itself by offering a seamless booking experience, personalized recommendations, and a social network for travelers to connect and share their experiences.

To achieve these goals, Travandoz requires a software development team to create an online platform that can handle the following tasks:

1. User registration and account management: Users should be able to create and manage their profiles, including personal information, payment methods, and booking history.
2. Property listing and management: Hosts should be able to list their properties on the platform, including property information, availability, and pricing.
3. Property search and booking: Users should be able to search for properties based on their preferences, view property details and photos, and make bookings.
4. Payment processing: The platform should be able to securely process payments from users and distribute payments to hosts.
5. Recommendation engine: The platform should incorporate a recommendation engine that suggests properties to users based on their search history and preferences.
6. Social networking: The platform should have social networking features that allow users to connect with each other, share experiences, and provide feedback on properties and hosts.
7. Administration and reporting: The platform should include an administration panel for managing user accounts, properties, bookings, payments, and reporting on performance metrics.

The software development team should use modern technologies and best practices to create a scalable, secure, and user-friendly platform that can handle high volumes of traffic and transactions. The team should also prioritize data privacy and security, ensuring that user data is protected at all times.

Overall, the goal of this project is to create a comprehensive and innovative online platform that revolutionizes the travel industry and provides a valuable service to users around the world.

PROCESS MODEL

In the case of Travandoz software, the V-model could be a suitable choice as the development team prioritizes testing and quality assurance throughout the development process.
The V-model is known for its emphasis on testing, with the testing phase being integrated into every stage of the development cycle.

We have used the V-model for the following reasons

- Early detection of defects: Testing activities are integrated throughout the development life cycle, which helps to identify defects early in the development process.
- Clear and well-defined process: The V-model provides a clear and well-defined process that helps to ensure that each phase of the development life cycle is completed before moving on to the next phase.
- Emphasis on testing: The V-model places a strong emphasis on testing activities, which helps to ensure that the software is thoroughly tested and reliable.
- Improved communication: The V-model encourages communication between developers, testers, and other stakeholders, which helps to ensure that everyone is on the same page and that project goals are met.
- Better quality software: By using a well-defined process that emphasizes testing, the V-model helps to ensure that the software is of high quality and meets the requirements of the stakeholders.
- Reduced costs: By identifying defects early in the development process, the V-model can help to reduce the costs associated with fixing defects later in the development life cycle.

1. Software Requirement Specification

1. Introduction :

1.1 Introduction :

Our mission is to "connect travelers with hosts and unique accommodations worldwide, while delivering an exceptional travel experience through innovative technology and outstanding customer service."

1.2 Purpose:

The purpose of the System Requirements Specification (SRS) document for Travandoz is to provide a comprehensive description of the system requirements and specifications for the development and implementation of the Travandoz platform. This document serves as a reference for all stakeholders involved in the development process, including developers, project managers, testers, and business analysts. The SRS document defines the functionality, performance, and design constraints of the system, and serves as a foundation for system design, development, and testing. It also ensures that all parties have a clear understanding of the system requirements and specifications before development work begins.

1.3 Scope of system

The scope of the system for Travandoz includes the development of a web-based platform and mobile application that connects budget-conscious travelers with local hosts in various destinations around the world. The system allows travelers to search and book shared accommodations, local experiences, and tours offered by hosts.

The system should include a user-friendly interface for travelers to search and filter options based on location, price, and other preferences. It should also provide a secure payment gateway for booking and payment processing.

For hosts, the system should allow them to create and manage their listings, including uploading photos, setting pricing, and managing availability. Hosts should also be able to communicate with travelers and manage bookings through the system.

Overall, the scope of the system is to provide a comprehensive platform for budget-conscious travelers to find unique and local experiences while also providing hosts with a platform to share their culture and earn income.

1.4 Acronyms, definitions and abbreviations

- **User**: The person who is using this platform for its services is a User
- **Admin**: A dedicated team who is responsible for login records and updating the database for any new information.
- **Host**: refers to all the hosts listed on our platform.

2. Overall Description

2.1 Description

To provide a platform for budget friendly travellers who want authentic travel experiences

To provide a platform to all the local villagers where they can show people around and earn by providing various services like accommodation, food etc.

2.2 Product's Perspective

Travandoz is an online platform that connects travelers with hosts who offer various types of properties for short-term rentals. The platform allows users to search for properties based on their preferences, such as location, price, property type, and availability. It also provides hosts with tools to manage their listings, including setting prices, availability, and accepting/rejecting bookings.

Travandoz aims to provide a user-friendly interface and a seamless booking experience for travelers while also offering hosts a way to monetize their unused or underutilized properties. The platform also strives to maintain a high level of trust and security by verifying hosts and users and facilitating secure payments.

Overall, the product perspective of Travandoz is to offer a convenient and reliable platform for short-term rentals that benefits both travelers and hosts.

2.3 Product’s Function

The application function varies with the user using it.

2.3.1 Admin:

As an administrator, the functions of the Travandoz product may include:

1. User and Host Management: The ability to manage user and host accounts, including creating, modifying, and deleting accounts, as well as managing user and host permissions.

2. Property Management: The ability to manage properties listed on the platform, including creating, editing, and deleting property listings, updating property information and images, and managing availability calendars.

3. Booking Management: The ability to manage booking requests, including approving or rejecting bookings, updating booking status, and managing payment and refund processing.

4. Reporting and Analytics: The ability to generate reports and analytics on various aspects of the platform, including user and host activity, booking trends, and financial data.

5. Customer Support: The ability to provide customer support to users and hosts, including responding to inquiries, resolving disputes, and handling complaints.

6. Platform Customization: The ability to customize the platform's appearance, including branding, website design, and content m anagement.

2.3.2 User:

Here are some of the product functions for the Travandoz user:

1. Search Properties: Users can search for properties based on location, property type, price range, availability, and other parameters. The search results can be filtered and sorted based on user preferences.

2. Property Details: Users can view detailed information about properties, including property features, photos, reviews, ratings, and availability calendar.

3. Booking Management: Users can manage their bookings, including viewing booking details, cancelling bookings, and modifying bookings if allowed by the host.

4. Review and Rating: Users can leave reviews and ratings for properties they have booked, which can be helpful for other users looking to book a property.

5. User Profile: Users can create and manage their profile, which includes their personal details, booking history, and saved properties.

6. Wish List: Users can add properties to their wish list, which allows them to easily access and compare properties they are interested in.

7. Communication: Users can communicate with hosts through the platform's messaging system, which allows them to ask questions about the property, discuss booking details, and request additional services if needed.

8. Payment: Users can make secure online payments for their bookings through the platform. The platform should support multiple payment options such as credit/debit cards, net banking, and digital wallets.

9. Notifications: Users should receive notifications about booking confirmations, cancellations, modifications, and other important updates related to their bookings.

10. Customer Support: The platform should provide customer support through multiple channels, such as email, phone, and chat, to assist users with any issues or queries they may have.

2.3.3 Host:

Here are some possible product functions for Travandoz hosts:

1. Property listing: Hosts can create and manage listings for their properties, including details such as property type, location, amenities, photos, and pricing.

2. Availability management: Hosts can set availability calendars for their properties and adjust availability based on bookings and other factors.

3. Booking management: Hosts can view and manage bookings for their properties, including approving or declining booking requests, adjusting booking details, and communicating with guests.

4. Pricing management: Hosts can set and adjust prices for their properties, including setting base rates, seasonal rates, and discounts.

5. Payment management: Hosts can manage payment information, view transaction history, and receive payouts for bookings.

6. Messaging: Hosts can communicate with guests through the platform's messaging system, both before and during their stay.

7. Reviews: Hosts can view and respond to reviews left by guests on their properties.

8. Dashboard: Hosts can view key metrics and data related to their properties, such as occupancy rates, revenue, and guest reviews.

9. Support: Hosts can access customer support and resources to help them manage their listings and bookings effectively.

2.4 User Characteristics:

1. Employees: The employees of Travandoz who work in various departments such as technology, marketing, customer support, and operations.

2. Local Hosts: The local hosts who offer their homes, services, and experiences to travelers through the Travandoz platform. They are the backbone of Travandoz's business model.

3. Travelers: The customers who use Travandoz to book their travel experiences and interact with the local hosts.

4. Investors: The investors who provide funding to Travandoz in exchange for equity in the company. They are interested in the growth and profitability of Travandoz.

5. Partners: The business partners who collaborate with Travandoz to provide additional services and offerings to travelers, such as transportation providers, tour companies, and local businesses.

2.5 General Constraints

1. Technical constraints: These include the hardware and software requirements that must be met for the system to function effectively, such as server capabilities, database requirements, and programming languages.

2. Security constraints: These include the measures that must be taken to ensure the security of the system and its users, such as encryption, authentication, and access control.

3. Performance constraints: These refer to the performance requirements of the system, such as response time, throughput, and scalability.

4. Legal constraints: These include the legal requirements that the system must comply with, such as data privacy laws and regulations.

5. Budget constraints: These refer to the financial constraints that the system must operate within, such as the available funding for development and maintenance.

Assumptions & Dependencies

Assumptions

1. Users have access to the internet and mobile devices to use the Travandoz platform.
2. Users are willing to share accommodations with other travelers.
3. Local hosts are available and willing to offer accommodations and experiences to travelers.
4. Payment gateway systems are available and functioning properly for secure transactions.
5. Users will comply with Travandoz's policies and guidelines for responsible and sustainable travel.

Dependencies:

1. The availability and reliability of internet and mobile network services.

2. The availability and willingness of local hosts to provide accommodations and experiences.

3. The availability and reliability of payment gateway systems.

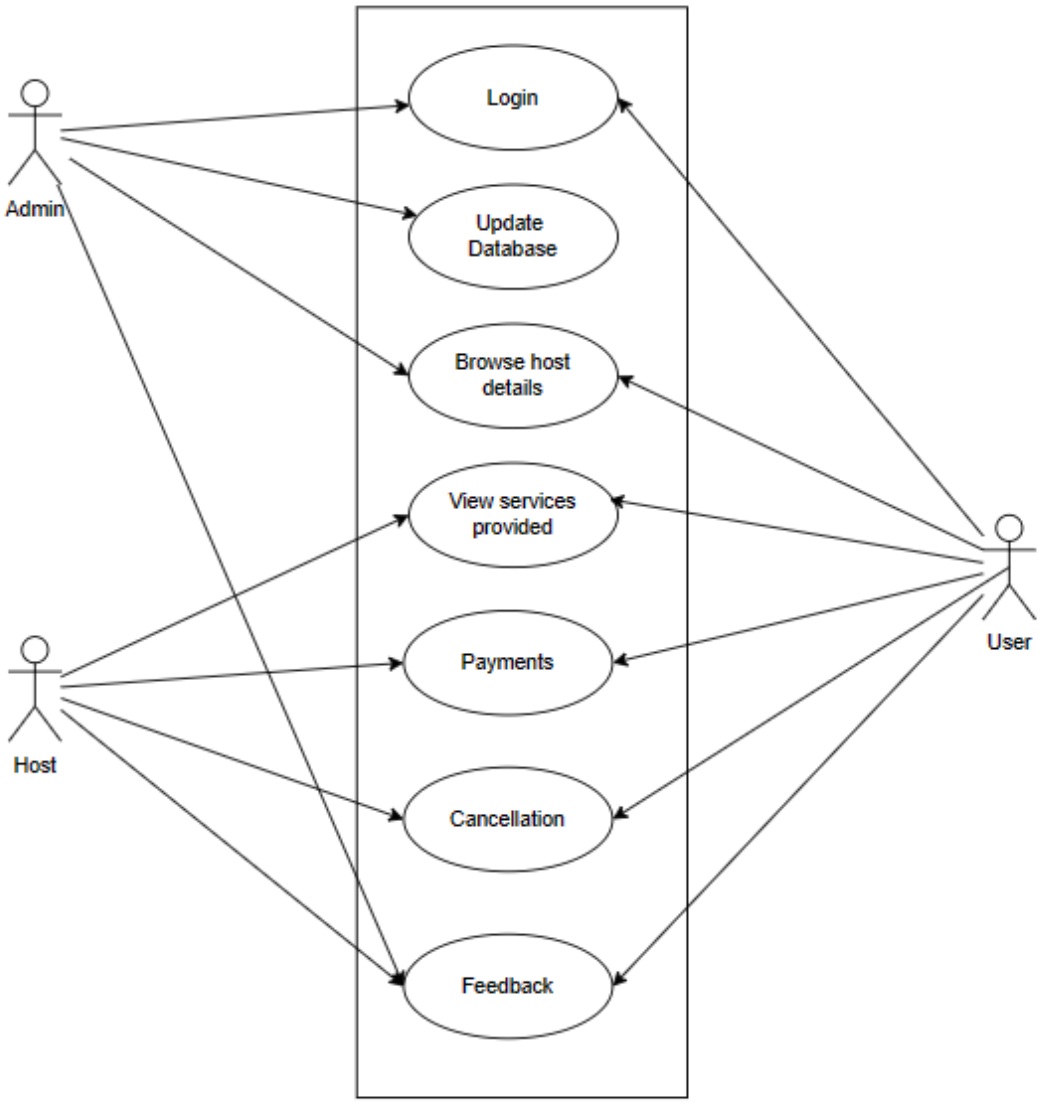
4. Compliance with local laws and regulations related to tourism and travel.

3. Requirements and Functional description

3.1 Functional requirements

3.1.1 Use Case Scenario

Use Case Diagram



3.1.2 Use Case Scenario

1. Login

1.1 Brief Description

This use case defines how a user logins into our website portal.

1.2 Actor

Admin, User and host actors interact in this use case

1.3 Flow of events

1.3.1 Basic Flow

- The system will ask the user to enter Name, Email id, Phone number and generate a new password and verify the entered information using OTP on the registered Email id/ Phone number.
- The actor will enter the above details asked by system.
- The system will store the data into the login details user database.

1.3.2 Alternative flow

- If in the basic flow, the actor enters an invalid email, phone number, or do not verify the entered details vis OTP the system displays and error message and the actor will need to enter those values again.

1.4 Pre-condition

None

1.5 Post condition

The user shall be able to log into the system.

Update Database

2.1 Brief description

This use case describes how the Admin will update different databases through Travandoz Portal.

2.2 Actor

Admin can interact in this use case.

2.3 Flow of events

2.3.1 Basic flow

- Admin will need to select the desired database
- After selecting a plan Admin will update the data in the chosen database
- After a successful updation, the changes will be saved in the database.

2.3.2 Alternative Flows

- If in the basic Flow, the details updated by the user are not valid then an error message will be shown to the user and the user can enter the details again.

2.4 Pre-condition

None

2.5 Post-Condition

The new data will be updated in the database and the user will be able to see the changes if the updation is done successfully.

3. Browse host details

3.1 Brief description

This use case describes how the user can choose host in this user interface on the website of Travandoz.

3.2 Actor

User can act in this use case.

3.3 Flow of events

3.3.1 Basic flow

- User will choose from the list of hosts available
- User will confirm host according to their convenience.

- Admin will confirm whether the host can offer the service on desired date or not.

3.3.2 Alternative flows

- If in the basic Flow, the details updated by the user are not valid then an error message will be shown to the user and the user can check for other available hosts.

3.4 Pre-condition

None

3.5 Post-Conditions

The user will be sent to the interface where all the services given by the host are listed down.

4. View services provided

4.1 Brief description

This use case describes how user can select all the services provided by a host.

4.2 Actor

User can act in this use case

4.3 Flow of events

4.3.1 Basic flow

- User will choose from the list of services available
- User will confirm services according to their convenience.
- Admin will confirm whether the host can offer the service on desired date or not.

4.3.2 Alternative flows

- If in the basic Flow, the details updated by the user are not valid then an error message will be shown to the user and the user can check for other available hosts/services.

4.4 Pre-Condition

User must have selected the host who will be providing these.

4.5 Post-Condition

The user will be sent to the interface where they can make payment and confirm reservations.

5. Payments

5.1 Brief description

This use describes how the user can make payments.

5.2 Actor

Host and users can interact in this use case

5.3 Flow of events

5.3.1 Basic Flow

- The costumer will need to choose the payment method first, then enter the required details i.e., Credit card no./ Debit card No./ UPI ID.
- Users can now verify the payment using a registered mail about payment or transaction.
- Host will now receive the payment in its account.

6.3.2 Alternative Flow

- If the payment details entered by the user are not valid, then an error message will be shown and the costumer can enter the payment details again.

5.4 Pre-Condition

There has to exist a bank account linked with the registered phone no.

5.5 Post-Condition

Costumer will receive the payment receipt on the registered main ID and the Host also will be able to see the transaction done in its account, if payment is done successfully.

6. Cancellation

6.1 Brief description

This use case will describe how the user can cancel the order on Travandoz portal.

6.2 Actor

Host and User can interact in this case.

6.3 Flow of events

6.3.1 Basic Flow

- The user will have to select the host which was booked.
- User will then confirm cancellation of the reservation.
- User will now receive a confirmation mail about cancellation of reservation.
- Travandoz will now cancel the reservation and will make the refund within 24 hrs.

6.4 Pre-Condition

There has to exist a database of reservations to keep track of all the host bookings.

6.5 Post-Condition

User will receive a confirmation of cancellation of reservation on the registered Email Id and the Host will cancel the booking.

7. Feedback

7.1 Brief description

This use case will register feedback of the users through Travandoz portal.

7.2 Actor

User and host can interact in this use case.

7.3 Flow of events

7.3.1 Basic flow

- Users will register the feedback after the usage of Travandoz website.

7.4 Pre-Condition

Users will first have to use the portal to give the feedback.

7.5 Post-Condition

The concerned users can see the received feedback and improve their services accordingly.

3.2 Non-Functional requirements

3.2.1 Reliability

- The stored data should be backed up every 7 days to avoid the loss of important information.

3.2.2 Security

- Login Details, Passwords, and History of customers should be encrypted and not shared over the Internet.
- Transactions should be provided using a secure interface.

3.2.3 Efficiency

- The platform should be able to handle a minimum of 10,000 users on the platform at the same time.
- The platform should be able to handle vast amount of data; might be in the millions of entries as everything is supported using data.

3.2.4 Availability

- The platform must be active 24/7..

3.2.5 Maintainability

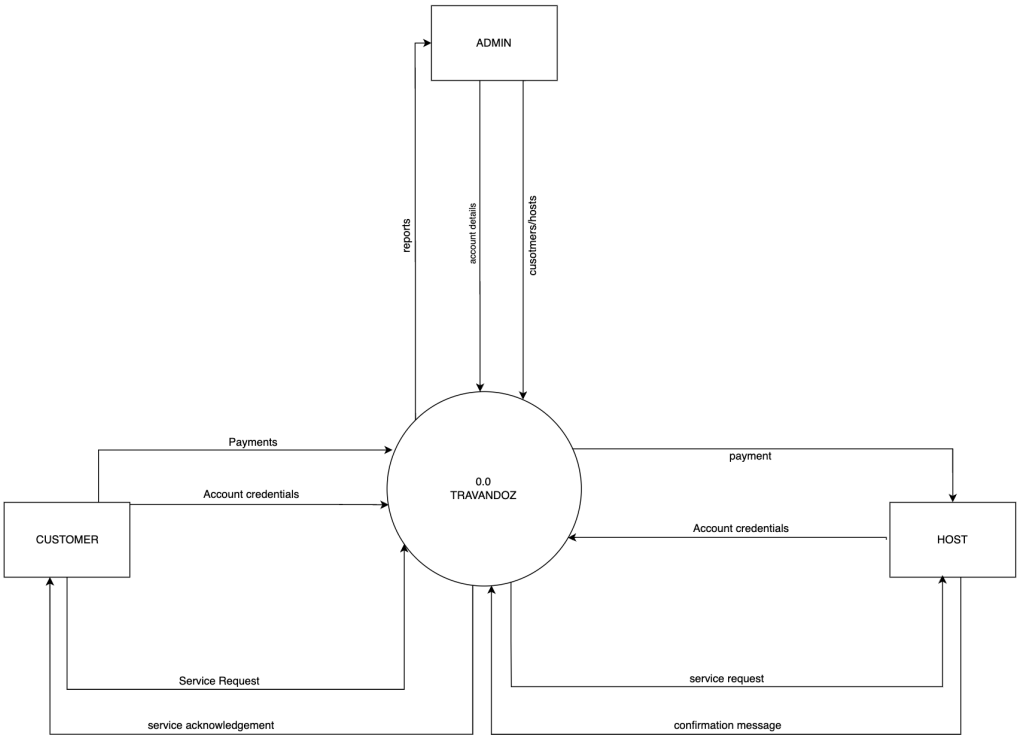
- The platform needs monthly updation of the information table in the database by the admin.
- New Features can be added as per future requirements.

3.2.6 Portability

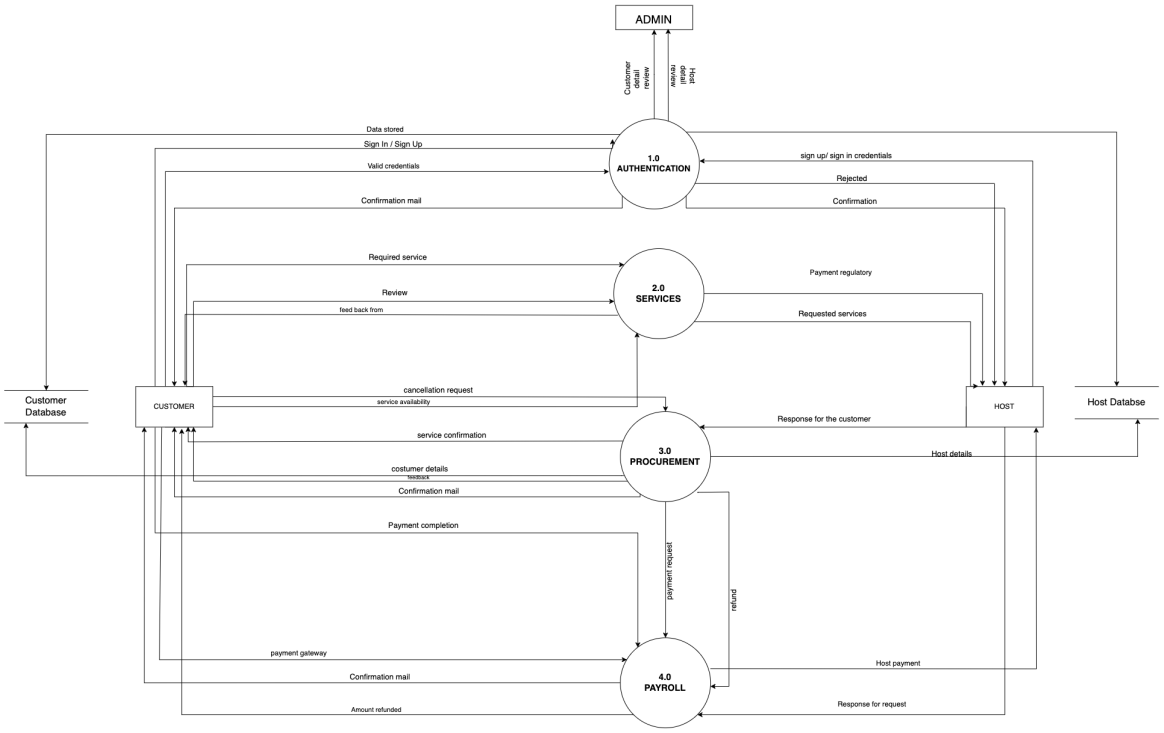
- The platform should be able to run perfectly on Internet Explorer 9 (and later), Google Chrome 7.0.517(and later), Mozilla Firefox 77.0.1 (and later), Opera Browser 51.0.2830.26 and Microsoft Edge.

3.3 Data flow diagram

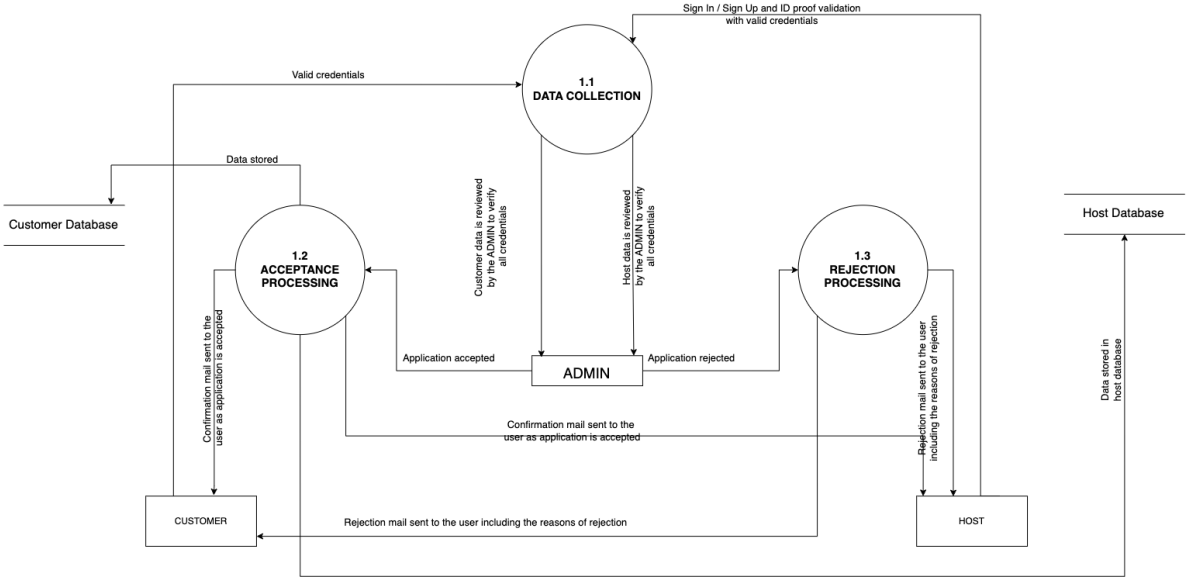
Context Level Diagram

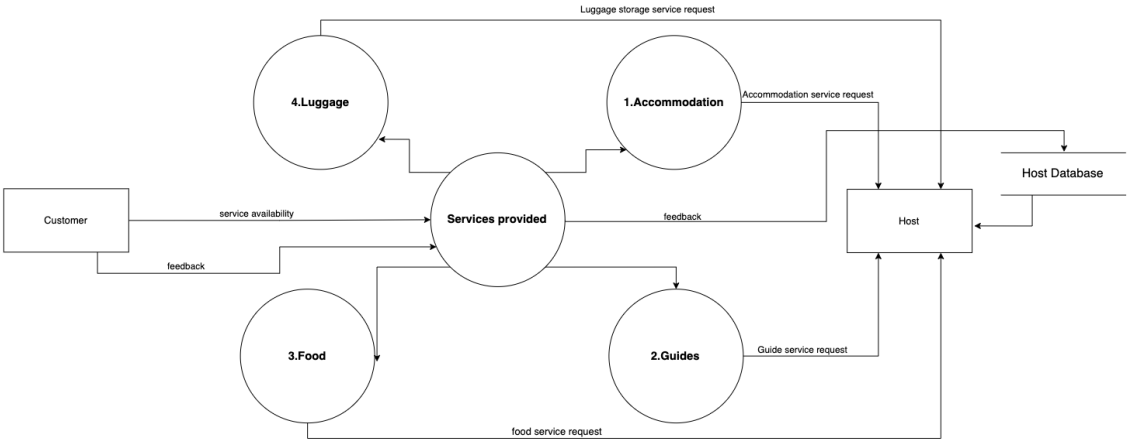


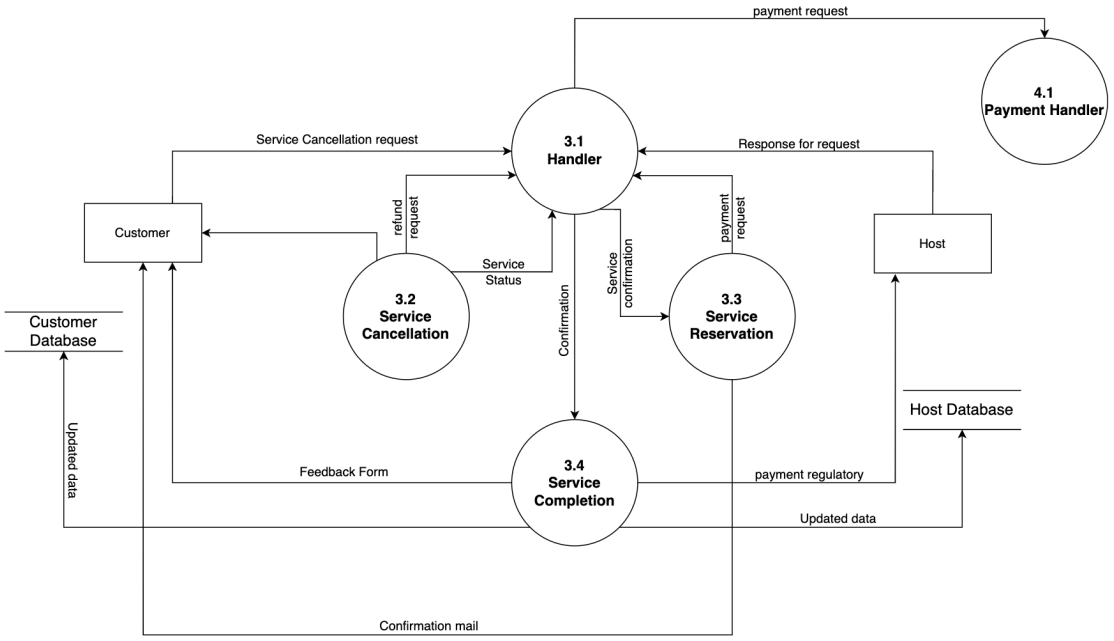
Level - 1

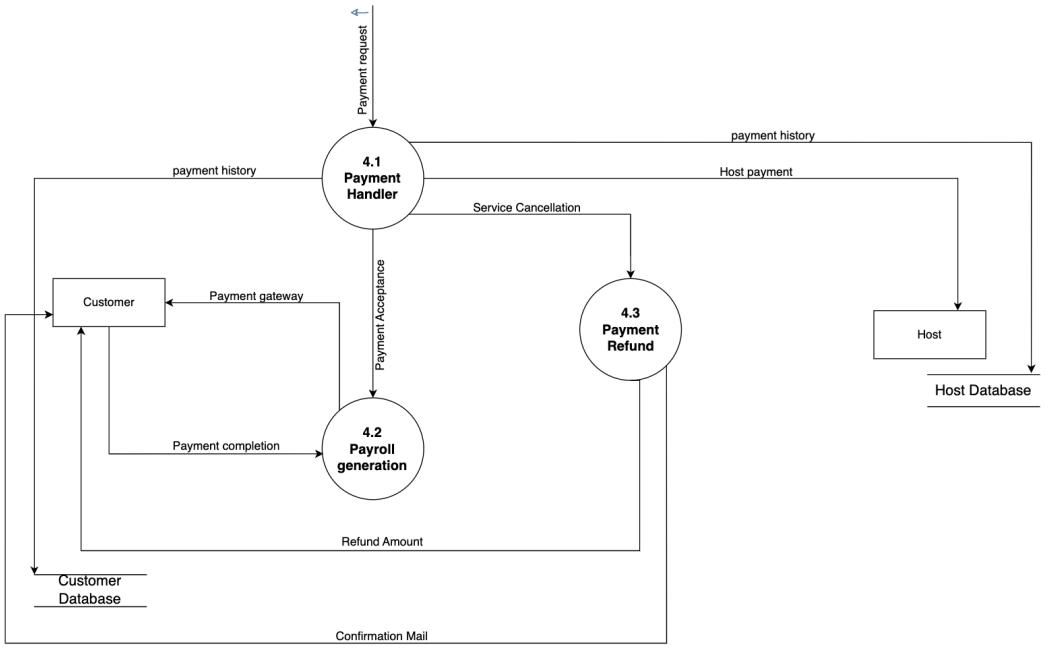


Level - 2


Sign In / Sign Up and ID proof validation
with valid credentials
1.1
DATA COLLECTION
Valid credentials
Data stored
Customer Database
Host Database
1.2
ACCEPTANCE
PROCESSING
1.3
REJECTION
PROCESSING
Customer data is reviewed
by the ADMIN to verify
all credentials
Host data is reviewed
by the ADMIN to verify
all credentials
Confirmation mail sent to the
user as application is accepted
Application accepted
ADMIN
Application rejected
Rejection mail sent to the user
including the reasons of rejection
Data stored in
host database
Confirmation mail sent to the
user as application is accepted
CUSTOMER
Rejection mail sent to the user including the reasons of rejection
HOST


Luggage storage service request
4.Luggage
1.Accommodation
Accommodation service request
Host Database
Customer
service availability
Services provided
feedback
Host
feedback
3.Food
2.Guides
Guide service request
food service request





3.4 Data Dictionary

- Reports - total amount+ timeline + average expenditures + past services
- Login- Email_id+Password
- Phone No- digit+ digit+digit+ digit+digit+digit +digit+digit+digit +digit
- hosts - host name + service provided + amount

- Payments - all the payment related details either from the user side or the host side
- Confirmation- Yes/No
- Offers- Type(cashback, discount) + Outlet_name + Description
- Coupon Details- Coupon_code + Description + T&C
- Refund- Order_details + Refund_amount
- customers - customer name + service requested + amount
- Order History- Order_details + Amount + Date_of_order+time_of_order + Locations_visited+Feedback
- Address Details- House_no+Block+Sector+City+ State+ Pin_code

4. External interface requirements

1. User Interface: The system should have an intuitive and easy-to-use user interface that allows users to easily navigate the platform, search for accommodations, make bookings, and communicate with hosts.

2. Payment Gateway: The software should integrate with a secure payment gateway to enable users to make payments for their bookings using various payment methods.

3. Social Media Integration: The platform should allow users to sign up and log in using their social media accounts, and also share their experiences and reviews on social media.

   Messaging and Communication: The software should have a messaging system that allows users to communicate with each other securely and efficiently.

4. Geolocation: The system should be able to detect the location of the user and provide them with relevant recommendations based on their location.

5. Email Notifications: The platform should send out automated email notifications to users regarding booking confirmations, cancellations, payment receipts, and other relevant information.

6. API Integration: The software should allow for integration with third-party APIs such as Google Maps and local weather services to provide users with relevant information.

7. Mobile App Integration: The system should be accessible through a mobile app that allows users to access the platform on their mobile devices.

8. Language Support: The platform should support multiple languages to cater to users from different regions and countries.

9. Accessibility: The system should be accessible to users with disabilities, including those using screen readers and other assistive technologies.

10. Hardware interfaces: These would include the specific hardware components required to run the software. For example, the minimum requirements for the user's computer or mobile device, including processor speed, memory, and storage capacity.

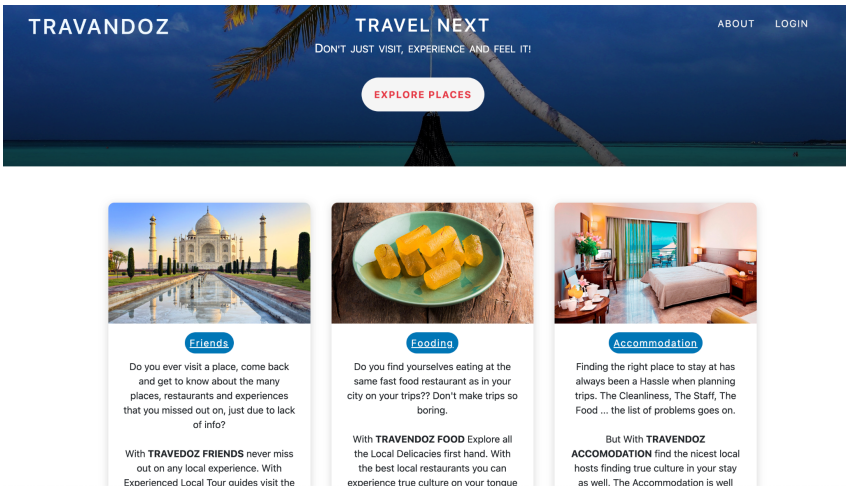
11. Software interfaces: These would include the programming languages, libraries, and other software components required to develop and run the software. This would also include the application programming interface (API) specifications for any third-party software components used by the system.

12. Network interfaces: These would include the protocols and standards for communication between the software system and external networks, such as the internet.

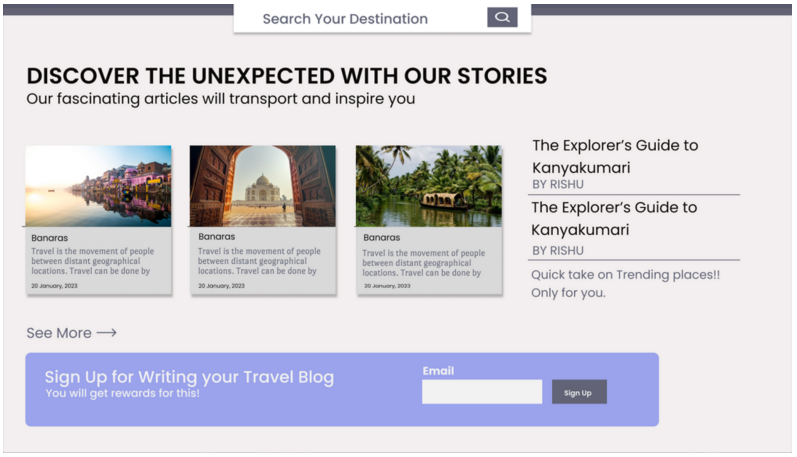
13. User interfaces: These would include the specific design and layout of the graphical user interface (GUI) or command-line interface (CLI) used by the end-user to interact with the system.

4.1 USER INTERFACE

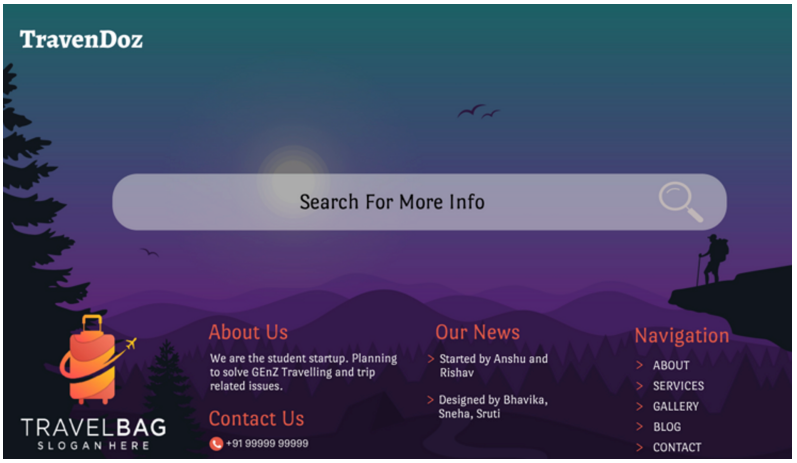
LANDING PAGE



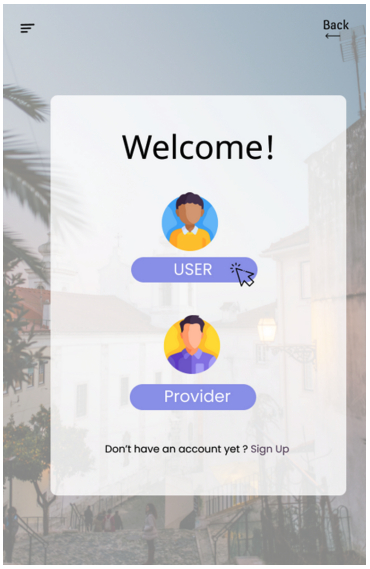
FEATURED PAGE



FOOTER DESIGN



Login Page



Sign in

Username*

Password*

SIGN IN

Sign up

Name*

Phone*

Email*

Password*

Confirm password*

SUBMIT

5. ESTIMATION

5.1 Function Point

S.No.	Question	Grade Value
1	Does the system require reliable backup and recovery?	5
2	Are specialized data communication required to transfer information to or from the application?	3
3	Are there distributed processing functions?	2
4	Is performance Critical?	3
5	Will the system run in an existing, heavily utilized operational environment?	5
6	Does the system require online data entry?	2

7	Does the online data entry require the input transaction to be built over multiple screens or operations?	4
8	Are the ILFs updated online?	5
9	Are the inputs, outputs, files, or inquiries complex?	3
10	Is the internal processing complex	2
11	Is the code designed to be reusable	5
12	Are conversions and installations included in different organizations?	5
13	Is the system designed for multiple installations in different organizations?	2

Rate on each factor on scale of 0 to 5:
0 : No Influence
1 : Incidental
2: Moderate
3: Average
4: Significant
5: Essential

		Estimated Count			Weighing Factor		Total
Information Domain Value	Simple	Average	Complex	Simple	Average	Complex	
External Input	3	3	3	3	4	6	39
External Output	8	6	4	4	5	7	90
External Inquiries	2	1	1	3	4	6	16
Number of Logical Files	5	3	0	7	10	15	65
External Interface Files	0	2	0	5	7	10	14

Value adjustment factor **= 48**

Unadjusted functional points(UFP):39+90+16+65+14= **224**

Complexity adjustment factor(CAF): 0.65 + (0.01 * 48)

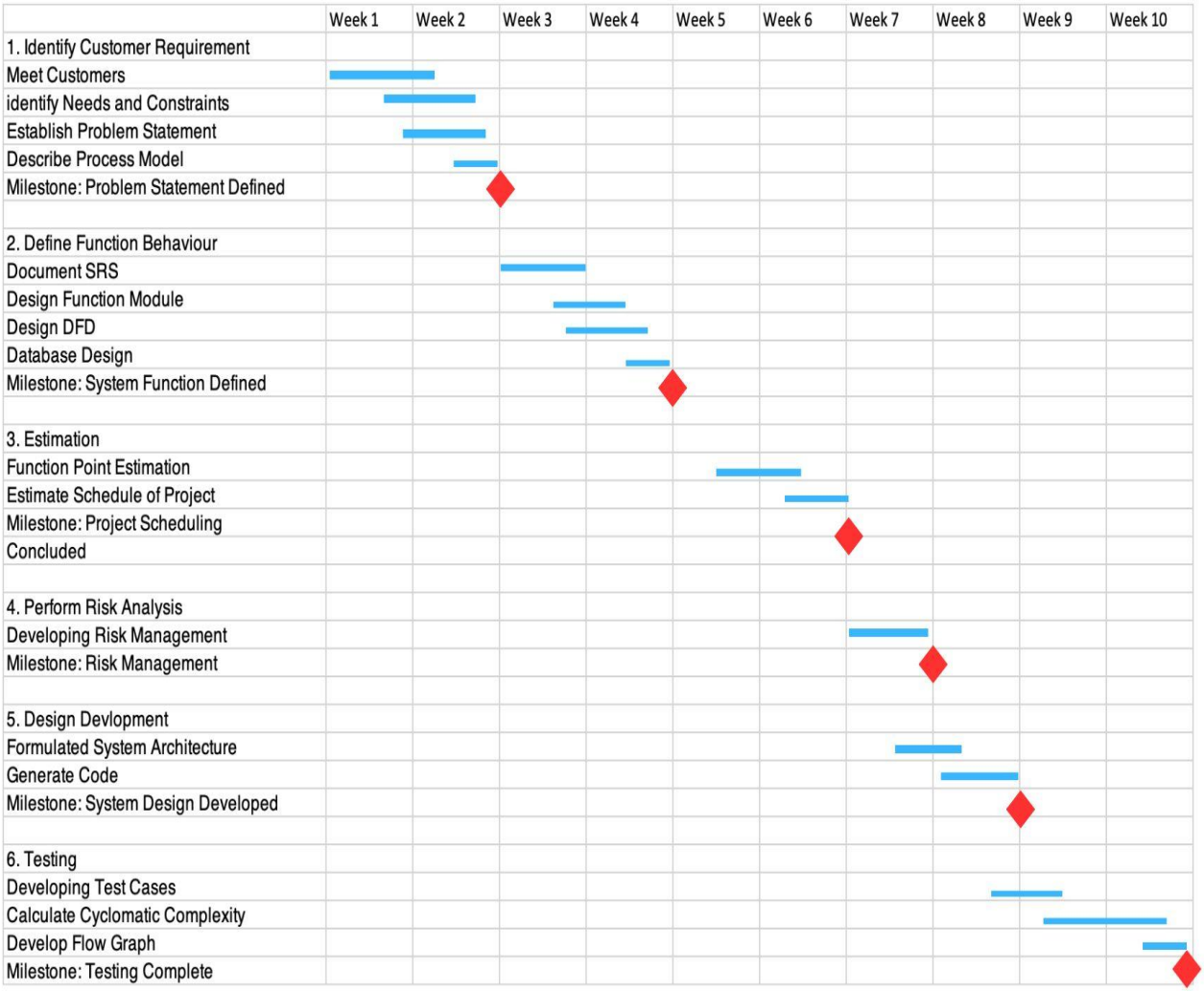
= **1.13**

Function Point Metric (FP): **UFP*CAF**

= 224*1.13

= **253.12**

6. SCHEDULING



7. RISK MANAGEMENT

7.1 Risk Table

Risk	Category	Probability	Impact
Inadequate product size	Product size	40%	Marginal
Unclear product definition	Product definition	60%	Marginal
Unexpected market changes	Business impact risk	80%	Critical
Inadequate staffing	Staff risk	50%	Marginal
Technical failure during development	Technical risk	30%	Marginal
Delay in project timeline	Project Risk	40%	Marginal
Loss of key customer	Customer Risk	80%	Critical
Inadequate development environment	Development Environment	40%	Negligible

7.2 RISK MITIGATION, MONITORING & MANAGEMENT PLAN (RMMM PLAN)

RMMM PLAN(For risks above cutt-off line)

Loss of key customer

Risk Mitigation:

- Develop a customer retention plan.
- Diversify the customer base to reduce dependency on any single customer.
- Regularly gather customer feedback and address concerns.

Risk Management:

- Communicate with stakeholders and develop a plan to retain the customer if **possible.**
- Diversify the customer base if necessary.
- Modify the project plan if the loss of a key customer impacts timelines or budgets.

Risk Monitoring:

- Regularly monitor customer satisfaction and feedback.
- Conduct regular reviews of the customer retention plan.

Unexpected market changes:

Risk Mitigation:

- Conduct regular market research to identify potential changes.
- Develop contingency plans for potential changes.
- Regularly review and update the business strategy.

Risk Management:

- Execute the contingency plan if necessary.
- Communicate with stakeholders and adjust the business strategy if required.
- Modify the project plan if the changes impact project timelines.

Risk Monitoring:

- Regularly monitor the market for changes.
- Review the business strategy on a regular basis to ensure it is still appropriate.

8. DATA DESIGN

GUIDES

Attribute	Data Type	Constraint
Guide_id	Int	Primary Key
Email	Varchar	Not Null
Location	Varchar	Null
F_name	Varchar	Not Null
L_name	varchar	Not Null

HOST

Attribute	Data Type	Constraint
Host_id	Int	Primary Key
Email	Varchar	Not Null
F_name	Varchar	Null
L_name	Varchar	Not Null

HOTELS

Attribute	Data Type	Constraint
Property_id	Int	Primary Key
Address	Varchar	Not Null
Location	Varchar	Not Null
Capacity	Int	Not Null
User_id	Int	Not Null
Price	Int	Not Null
Rating	Int	Null

PRIVATE PROPERTY

Attribute	Data Type	Constraint
Property_id	Int	Primary Key
Address	Varchar	Not Null
User_id	Int	Not Null
Rating	Int	Null
Price	Int	Not Null

INTERNATIONAL USER

Attribute	Data Type	Constraint
User_id	Int	Primary Key
Booking Charge	Int	Not Null
Name	Varchar	Not Null
Country	Varchar	Not Null
Driving_Licence	varchar	Not Null

DOMESTIC USER

Attribute	Data Type	Constraint
User_id	Int	Primary Key
Booking Charge	Int	Not Null
Name	Varchar	Not Null
Aadhar_ID	Int	Not Null

PROPERTY BOOKING

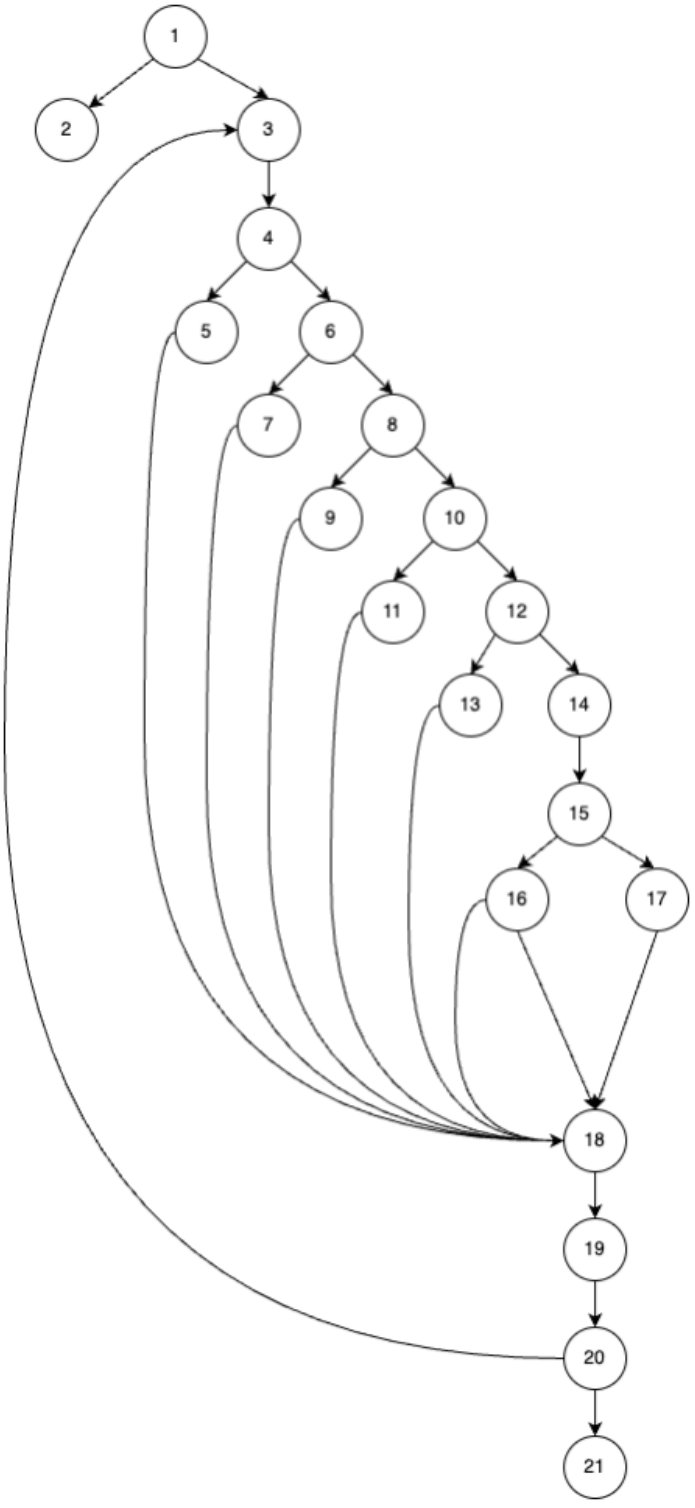
Attribute	Data Type	Constraint
User_id	Int	Primary Key
Booking_id	Int	Not Null
Total_prc	Varchar	Not Null
Status	Varchar	Not Null

9. CODING

```
#include <iostream>
using namespace std;
int main(){
	int id,pwd;
	cout<<"Welcome to Travandoz: \nEnter the Choice\n";
	cout<<"Please login/signup\n";
	cin>>id;
	cin>>pwd;
	if(validate(id,pwd)){   // (1)
	int choice;
	cout<<"Welcome user, Enter the choice: \n";
	cout<<"1. Visit profile \n2.Book an accomodation \n3.Dine in \n4.Hire a guide
\n5.Find a lugguage centre \n6.View booking history \n7.Plan a tour";
	cin>>choice;
	do{  //(3)
	if(ch==1){ //(4)
	profile();  //(5)
	}
	else if(ch==2){ //(6)
	bookAccomodation(); //(7)
	}
	else if(ch==3){  //(8)
	bookFood(); //(9)
	}
	else if(ch==4){ //(10)
	hireGuide(); //(11)
	}
	else if(ch==5){ //(12)
	luggage();      //(13)
	}
	else if(ch==6){ //(14)
	bookingHistory(); //(15)
	}
	else if(ch==7){  //(16)
	cout<<"Enter the date for the tour (dd/mm/yyyy)\n";
	int day,month,year;
	cin>>day;
	cin>>month;
	cin>>year;
	if(validDate(day,month,year)){ //(17)
```

```
                planTour(); //(18)
        }
        else{  //(19)
                cout<<"Invalid Date";
        }
        }
        int inp;
        cout<<"Do you want to continue"<<endl;
        cin>>inp;
        }while(inp==1);  //(20)
}
    else{ //(2)
        cout<<"Wring credentials"<<endl;
        }
} //(21)
```

10. TESTING



Cyclomatic Complexity:

1. E = 27, N = 20

V(G) = E- N +2

= 27 – 20 +2

= 9

2. V(G) = 8+1 = 9

3. V(G) = TOTAL NO. OF REGIONS

= 9

- **Independent Paths**
  1. 1->3->4->5->18->19->20->21
  2. 1->2
  3. 1->3->4->6->7->18->19->20->21
  4. 1->3->4->6->8->9->18->19->20->21
  5. 1->3->4->6->8->10->11->18->19->20->21
  6. 1->3->4->6->8->10->12->13->18->19->20->21
  7. 1->3->4->6->8->10->12->14->15->16->18->19->20->21
  8. 1->3->4->6->8->10->12->14->15->17->18->19->20->21

- Boundary Value Analysis

The analysis is done for the input of date

Range of values:

For years:

2023 <= years <= 2024 (current year +1)

For Month:

Current month <= month <= 12

For Date:

For month values in (1,3,5,7,8,10,12)

Current date <= date <= 31

For month values in (4,6,9,11)

Current date <= date <= 30

For month value in (2)

For leap years:

Current date <= date <= 28

For Non Leap years

Current date <= date <=29

For this case, current date is considered as **26/04/2023**

Boundary Value Test Cases:

Test Case	Date	Month	Year	Expected Output
1	25	04	2023	Invalid Date
2	26	03	2023	Invalid Date
3	27	04	2023	Valid Date
4	26	05	2023	Valid Date
5	27	04	2023	Valid Date
6	26	04	2022	Invalid Date
7	26	04	2024	Valid Date
8	26	04	2025	Invalid Date
9	31	04	2023	Valid Date
10	30	03	2023	Invalid Date
11	28	06	2024	Invalid Date
12	29	13	2023	Invalid Date
13	29	08	2023	Valid Date

Robust Test Cases:

Test Case	Date	Month	Year	Expected Output
1	29	02	2023	Invalid Date
2	29	02	2021	Invalid Date
3	29	02	2024	Valid Date
4	00	01	2023	Invalid Date
5	01	13	2024	Invalid Date
6	01	01	0001	Invalid Date

7	01	01	10001	Invalid Date
8	00	04	2023	Invalid Date
9	10	00	2023	Invalid Date
10	11	06	2024	Invalid Date
11	12	05	2023	Valid Date
12	Abcd	Bc	Sdf	Invalid Date
13	30	02	2023	Invalid Date
14	22	03	2023	Invalid Date
15	31	02	2024	Invalid Date
16	31	04	2023	Invalid Date
17	30	05	2023	Valid Date
18	31	06	2024	Valid Date
19	30	01	2023	Invalid Date

Worst Case Test Case

Test Case	Date	Month	Year	Expected Output
1	25	04	2023	Invalid Date
2	26	03	2023	Invalid Date
3	27	04	2023	Valid Date
4	26	05	2023	Valid Date
5	27	04	2023	Valid Date
6	26	04	2022	Invalid Date
7	26	04	2024	Valid Date
8	26	04	2025	Invalid Date

9	31	04	2023	Valid Date
10	30	03	2023	Invalid Date
11	28	06	2024	Invalid Date
12	29	13	2023	Invalid Date
13	29	08	2023	Valid Date
14	29	02	2023	Invalid Date
15	29	02	2021	Invalid Date
16	29	02	2024	Valid Date
17	00	01	2023	Invalid Date
18	01	13	2024	Invalid Date
19	01	01	0001	Invalid Date
20	01	01	10001	Invalid Date
21	00	04	2023	Invalid Date
22	10	00	2023	Invalid Date
23	11	06	2024	Invalid Date
24	12	05	2023	Valid Date
25	Abcd	Bc	Sdf	Invalid Date
26	30	02	2023	Invalid Date
27	22	03	2023	Invalid Date
28	26	04	2022	Invalid Date
29	26	04	2024	Valid Date
30	26	04	2025	Invalid Date
31	31	04	2023	Valid Date
32	30	03	2023	Invalid Date

33	28	06	2024	Invalid Date
34	29	13	2023	Invalid Date
35	29	08	2023	Valid Date
36	29	02	2023	Invalid Date
37	29	02	2021	Invalid Date
38	31	04	2024	Invalid Date
39	29	02	2027	Invalid Date
40	31	01	2023	Valid Date
41	32	05	2023	Invalid Date
42	26	04	2022	Invalid Date
43	26	04	2024	Valid Date
44	26	04	2025	Invalid Date
45	31	04	2023	Valid Date
46	30	03	2023	Invalid Date
47	28	06	2024	Invalid Date
48	29	13	2023	Invalid Date
49	29	08	2023	Valid Date
50	29	02	2023	Invalid Date
51	29	02	2021	Invalid Date
52	29	02	2024	Valid Date
53	00	01	2023	Invalid Date
54	26	04	2022	Invalid Date
55	26	04	2024	Valid Date
56	26	04	2025	Invalid Date

57	31	04	2023	Valid Date
58	30	03	2023	Invalid Date
59	28	06	2024	Invalid Date
60	29	13	2023	Invalid Date
61	29	08	2023	Valid Date
62	29	02	2023	Invalid Date
63	29	02	2021	Invalid Date
64	29	02	2024	Valid Date
65	00	01	2023	Invalid Date
66	00	01	2023	Invalid Date
67	01	13	2024	Invalid Date
68	01	01	0001	Invalid Date
69	01	01	10001	Invalid Date
70	00	04	2023	Invalid Date
71	10	00	2023	Invalid Date
72	11	06	2024	Invalid Date
73	12	05	2023	Valid Date
74	29	13	2023	Invalid Date
75	29	08	2023	Valid Date
76	29	02	2023	Invalid Date
77	29	02	2021	Invalid Date
78	31	04	2024	Invalid Date
79	29	02	2027	Invalid Date
80	31	01	2023	Valid Date

81	32	05	2023	Invalid Date
82	26	04	2022	Invalid Date
83	26	04	2024	Valid Date
84	26	04	2025	Invalid Date
85	31	04	2023	Valid Date
86	30	03	2023	Invalid Date
87	28	06	2024	Invalid Date
88	29	13	2023	Invalid Date
89	00	01	2023	Invalid Date
90	00	01	2023	Invalid Date
91	01	13	2024	Invalid Date
92	01	01	0001	Invalid Date
93	01	01	10001	Invalid Date
94	00	04	2023	Invalid Date
95	10	00	2023	Invalid Date
96	11	06	2024	Invalid Date
97	12	05	2023	Valid Date
98	26	04	2025	Invalid Date
99	31	04	2023	Valid Date
100	30	03	2023	Invalid Date
101	28	06	2024	Invalid Date
102	29	13	2023	Invalid Date
103	29	08	2023	Valid Date
104	29	02	2023	Invalid Date

105	Sdas	Sdsff	Sdsf	Invalid Date
106	26	04	2022	Invalid Date
107	26	04	2024	Valid Date
108	26	04	2025	Invalid Date
109	31	04	2023	Valid Date
110	30	03	2023	Invalid Date
111	28	06	2024	Invalid Date
112	29	13	2023	Invalid Date
113	29	08	2023	Valid Date
114	29	02	2023	Invalid Date
115	29	02	2021	Invalid Date
116	31	04	2024	Invalid Date
117	29	02	2027	Invalid Date
118	31	01	2023	Valid Date
119	00	01	2023	Invalid Date
120	01	13	2024	Invalid Date
121	01	01	0001	Invalid Date
122	01	01	10001	Invalid Date
123	00	04	2023	Invalid Date
124	10	00	2023	Invalid Date
125	11	06	2024	Invalid Date

Equivalence Class Testing

O_1 = {<D,M,Y> : Accepted Dates if all are valid inputs}

O_2 = {<D,M,Y> : Invalid Date if any input makes the date invalid }

I_1 = { month : 1<m<12}

I_2 = {month : m<1 }

I_3 = {month: m>12}

I_4 = {day : 1<D<31}

I_5 = {day: D<1}

I_6 = {day: D>31}

I_7 = {year: 2023<year<2024}

I_8 = {year: year<2023}

I_9 = {year: year>2024}

Test Case	Date	Month	Year	Expected Output
1	29	06	2023	Valid Date
2	-1	04	2023	Invalid Date
3	14	14	2023	Invalid Date
4	08	08	2023	Valid Date
5	23	0	2023	Invalid Date
6	22	15	2023	Invalid Date
7	31	05	2023	Valid Date
8	14	07	2021	Invalid Date
9	18	09	2026	Invalid Date