LAB REPORT

Submitted by

Prudhivi Sai Ganesh [RA2011003011057]

Under the Guidance of

Dr. V. Bibin Christopher

ASSISTANT PROFESSOR, Computing Technologist

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

with specialization in CORE

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

JUNE 2022

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this lab report titled **"Software Engineering and Project Management"** is the bonafide work done by Prudhivi Sai Ganesh (RA2011003011057) who carried out the lab exercises under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE

Dr. Bibin Christopher V

SEPM – Course Faculty

ASSISTANT PROFESSOR
Department of Computing Technologist

ABSTRACT

Although networked hospitality businesses as GreetHosts are a recent phenomenon, a rapid growth has made them a serious competitor for the hospitality industry with important consequences for tourism and for tourist destinations. The purpose of this paper is to analyze the nature of the phenomenon, its potential further development in the next five years and the impact this developments will have on tourism, on hotels and on city destinations.

GreetHosts has been described in conceptual studies about the so-called “Sharing economy”, or more recently in empirical studies about isolated effects of holiday rentals. This paper contextualizes the evolution of networked hospitality and seeks to synthesize the sum of its impacts, thus enabling businesses and local governments to define positions and strategies.

TABLE OF CONTENTS

Chapter No. : 1

Title of the experiment : Frame a project team, analyze and identify a Software project. To create a business case and Arrive at a Problem Statement for the Greet Hosts

Project Title: Greet Hosts

Aim :

To Frame a project team, analyze and identify a Software project. To create a business case and Arrive at a Problem Statement for the <title of the project>

Project Description

Greet host application hosts list many different kinds of properties, from single rooms, apartments, and housest to houseboats, caravans, and even castles

Business Case

ONE PAGE BUSINESS CASE TEMPLATE

THE PROJECT In bullet points, describe the problem this project aims to solve or the opportu

- Greet host application hosts list many different kinds of properties, from single rooms, apartments, and housest to houseboats, caravans, and even castles

THE HISTORY

In bullet points, describe the current situation.

- A vacation rental—whether a home, apartment, condo, boat, yurt, shared space, or other property—represents tremendous value, compared to hotels, for flexible, adventurous travel consumers

LIMITATIONS

List what could prevent the success of the project, such as the need for expensive equipment, bad weather, lack of special training, etc.

- What You See May Not Be What You Get
- Potential Damage
- Added Fees,Taxes

APPROACH

List what is needed to complete the project.

- Sanity for displaying the content
- Javascript and CSS to provide functionality and style to our site
- Google maps API and API keys

BENEFITS

In bullet points, list the benefits that this project will bring to the organization.

- Wide selection, Free listings
- Hosts Can Set Their Own Price
- Customizable SearchesAdditional Services
- Protections for Guests and Hosts

Result :

Thus, the project team formed, the project was described, the business case was prepared and the problem statement was arrived.

Chapter No. : 2

Title of the experiment: Identify the appropriate Process Model for the project and prepare Stakeholder and User Description.

Aim:

To identify the appropriate Process Model for the project and prepare Stakeholder and User Description.

Project Title: Greet Hosts

Selection of Methodology

- **< Summarize their understanding of "Waterfall" or "Agile" Methodology>**
- We have planned to follow the Agile Methodology , it is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating. Continuous collaboration is vital, both with team members and project stakeholders. A project management methodology characterized by building products using short cycles of work that allow for rapid production and constant revision. o Scrum - A PM methodology in which a small team is led by a Scrum master, whose main job is to clear away all obstacles to completing work. Work is done in short cycles called sprints, but the team meets daily to discuss current tasks and roadblocks of Extreme Project Management (XPM)- A PM methodology where the project plan, budget, and final deliverable can be changed to fit evolving needs, no matter how far along the project is.

Incorporate information to below table regarding stakeholders of the project [Make use of below examples]

Stakeholder Register:

Project Stakeholder Name	**Specific Information Needs**	**Project Interests**	**Impact on Project**	**Role**
	Types & Frequency of Communication	*Specific Areas of Interest and Participation*	*Positive, Negative, Influencer, Supporter, Roadblock*	*Decision Maker, Collaborator, Participant, Consultant, Information Recipient*
Sai Ganesh	Low	Project Management	Supporter	Decision maker
Pranav	Initial Discussion	Sponsoring	Influencer, Positive	Collaborator
Vijay kumar	Daily Reports	Developing the back end	Positive	Specialist
Sai Ganesh	Daily reports	Developing the front end	Positive	Specialist
Pranav	Frequent based on progress	Operation and Testing	Positive	Tester
Sai Ganesh	Frequent Meeting	Lead the technicalities	Positive	Consultant
Vijay kumar	After the software	End user, Trying the product	Positive, Negative	Information Receipt

For Example

Stakeholder Name	**Activity / Area / Phase**	**Interest**	**Influence**	**Priority (High / Medium/Low)**
Regional Head of Sales & Marketing	Subscription using mobile App	High	High	1
Finance Account Receivable consultant	Multiple Currency Payment	High	Low	3

Interest and Influence matrix

Interest	Influence
High	High
Low	Low
Low	High
High	Low

Stakeholder	Interests	Estimated Project Impact	Estimated Priority
Owner	Achieve targets, Increase sales margin	High	1
Sponsor	Provides new market to expand ventures Negotiate funding for project Reviews changes to project environments.	Med	3
Team members	Demand incentives Retain and upgrade skills New product excitement	High	2
Project Manager	Lead the team in every aspect. Accountable for entire project scope, team, success & failure	High	2
Investors	Promoter of the investment, Provides necessary financial resources	Low	5
Resource Manager	Resource planning and allocation. Ensuring adequate resource according to project needs and budget.	Med	4
Suppliers	Ensuring feasible and realistic in every aspect Managing divergence from budgeted cost.	Med	6
End Users	Provides feedback	Low	7

Result:

Thus the Project Methodology was identified and the stakeholders were described.

Chapter.No:3

Title of the experiment : identify the system, functional and non-functional requirements for the project

.

Aim:

To identify the system, functional and non-functional requirements for the project.

Project Title: Greet Hosts

System Requirements: Html, JavaScript, CSS, figma, React JS, windows 8 or later memory 4GB minimum /8GB recommended, screen resolution-1280*1024 or later, application window size - 1024*680 or later, internet connection required.

Functional Requirements:

- The system must allow the customer to register for booking.
- The system must notify on selection of unavailable rooms/apartments/lodges/own houses while making a booking.
- The system must view list of available rooms/apartments/lodges/own houses during reservation
- The system should allow host and user to login to the system using their username and password.
- The system shall search through all the possible rooms available and shall provide the best possible room host needs.
- The system must be able to display reservation summary for successfully committed reservation.
- The system shall provide the nearest rooms available to the host's vacation spot.

Non-Functional Requirements:

- Performance
- Availability
- Usability
- Security
- Error handling
- Ease of use

Result :

Thus the requirements were identified and accordingly described.

Chapter No. : 4

Title of the experiment : Prepare Project Plan based on scope, Calculate Project effort based on resources, Find Job roles and responsibilities

Aim:

To Prepare Project Plan based on scope, Calculate Project effort based on resources, Find Job roles and responsibilities

Team Members:

Sl No	Register No	Name	Role
1	RA2011003011057	Sai Ganesh	Lead
2	RA2011003011050	M. Vijay kumar Reddy	Member
3	RA2011003011046	Pranav Reddy	Member

Requirements

<Incorporate the Project plan template>

Result:

Thus, the Project Plan was documented successfully.

1. Project Management Plan

Describe the key issues driving the project. **[Min 3 Focus Areas]**

Focus Area	Details
Integration Management	Governance Framework Project Team Structure Roles & Responsibilities of Team Change Management (Change Control, Issue Management) Project Closure
Scope Management	Scope Statement Requirement Management (Gathering, Control, Assumption, Constraint Stakeholder) Define Deliverable Requirement Change Control Activities and Sub-Tasks
Schedule Management	Define Milestones Schedule Control
Cost Management	Estimate Effort Assign Team Budget Control
Quality Management	Quality Assurance: Quality assurance will be managed including governance, roles and responsibilities, tools and techniques and reporting Quality Control: Specify the mechanisms to be used to measure and control the quality of the work products
Resource Management	Estimate and Manage the need People: People & Skills Required Finance: Budget Required Physical: Facilities, IT Infrastructure
Stakeholder	Identifying, Analyzing, Engaging Stakeholders
Communication Management	Determine communication requirements, roles and responsibilities, tools and techniques. [Type of Communication, Schedule, Mechanism Recipient]
Risk Management	Identifying, analyzing, and prioritizing project risks

Procurement Management	Adhering to organization procurement process

2. Estimation

2.1. Effort and Cost Estimation

Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
Implantation of Loading Bar,	Date picker	User selects the date on the day they need to stay	1	500
	Room selection	User selects the room according to their needs	1.5	750
	Map rendering	Allows the user to know all available routes to their destination	2.5	1250
Server side rendering	Sending the data	Sending data to server	2	1000
	Receiving the data	Receiving the data from server for quick rendering	2	1000

Effort (hr)	Cost (INR)
1	500

2.2. Infrastructure/Resource Cost [CapEx]

< OneTime Infra requirements >

Infrastructure Requirement	Qty	Cost per qty	Cost per item
Computer lab	1	0	0

2.3 Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	Network, System, Middleware and DB admin Developer , Support Consultant	1	2,000,000	2,000,000
License	Operating System Database	1	62000	DataBase-47000

	Middleware Visual Studio IDE			Visual studio-15000
Infrastructures	Server, Storage and Network	1	20000	20000

3. Project Team Formation

3.1. Identification Team members

Name	Role	Responsibilities
Prudhvi Ganesh	Key Business User (Product Owner)	Provide clear business and user requirements
Vijay kumar	Project Manager	Manage the project
Pranav Reddy	Business Analyst	Discuss and Document Requirements
Prudhvi Ganesh	Technical Lead	Design the end-to-end architecture
Vijay kumar	UX Designer	Design the user experience
Prudhvi Ganesh	Frontend Developer	Develop user interface
Vijay kumar	Backend Developer	Design, Develop and Unit Test Services/API/DB
Pranav Reddy	Tester	Define Test Cases and Perform Testing

3.2. Responsibility Assignment Matrix

RACI Matrix	Team Members			
Activity	**Name (BA)**	**Name (Developer)**	**Name (Project Manager)**	**Key Business User**
User Requirement Documentation	Prudhvi Ganesh(A)	Pranav Reddy(c)	Vijay kumar()I	Prudhvi Ganesh/Vijay kumar(R)

A	Accountable
R	Responsible
C	Consult
I	Inform

Reference

1. https://www.pmi.org/
2. https://www.projectmanagement.com/
3. https://www.tpsgc-pwgsc.gc.ca/biens-property/sngp-npms/ti-it/ervcpgpm-dsfvpmpt-eng.html

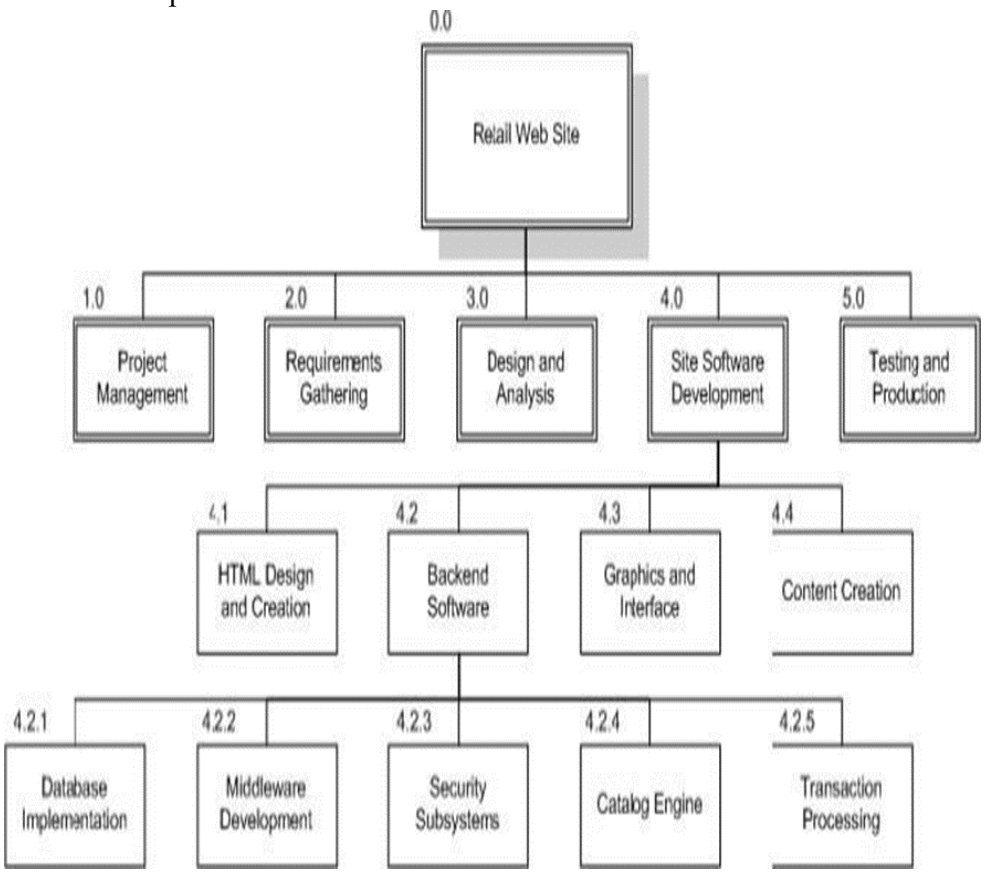
Chapter No:5

Aim:

To Prepare Work breakdown structure, Timeline chart and Risk identification table

<Incorporate WBS, Timeline chart and Risk table>

WBS – Examples



0.0 Retail Web Site

1.0 Project Management

2.0 Requirements Gathering

3.0 Analysis & Design

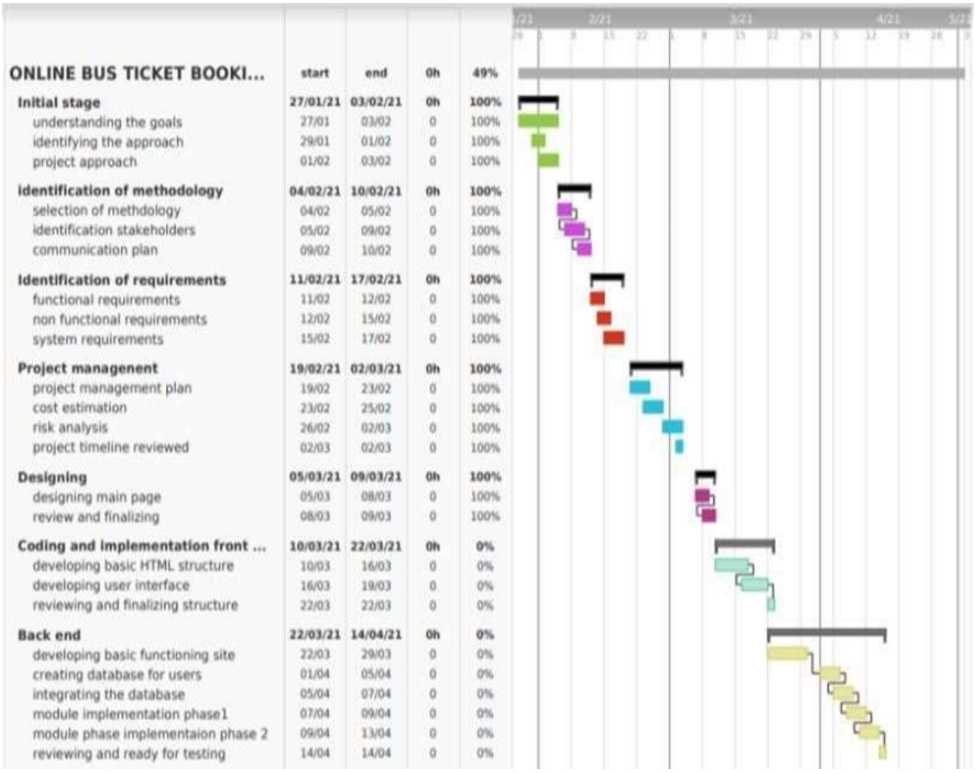
4.0 Site Software Development

- 4.1 HTML DesignandCreation
- 4.2 Backend Software

- 4.2.1 Database Implementation
- 4.2.2 Middleware Development
- 4.2.3 Security Subsystems
- 4.2.4 Catalog Engine

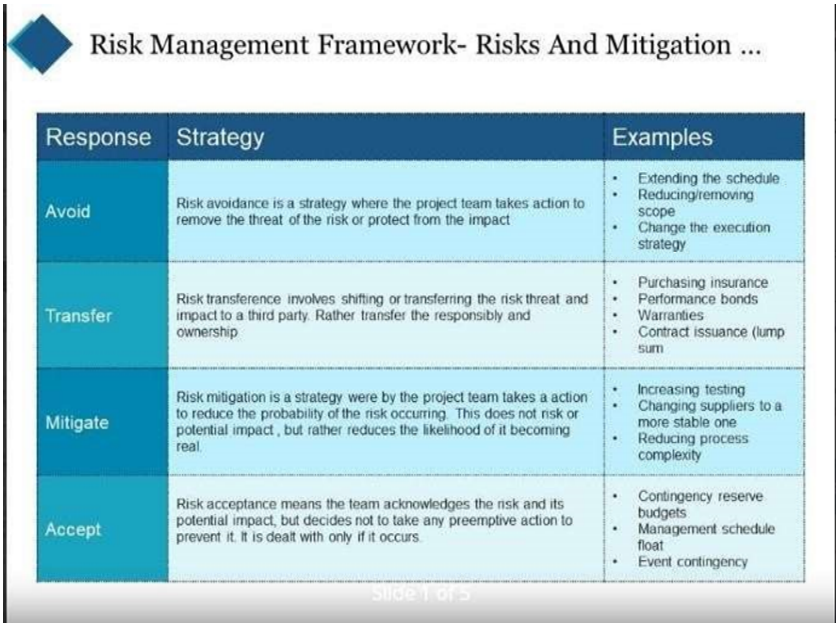
- 4.2.5 Transaction Processing
  - 4.3 Graphics and Interface
  - 4.4 Content Creation
  - 5.0 Testing and Production

TIMELINE – GANTT CHART



ONLINE BUS TICKET BOOKI...	start	end	0h	49%
Initial stage	**27/01/21**	**03/02/21**	**0h**	**100%**
understanding the goals	27/01	03/02	0	100%
identifying the approach	29/01	01/02	0	100%
project approach	01/02	03/02	0	100%
identification of methodology	**04/02/21**	**10/02/21**	**0h**	**100%**
selection of methdology	04/02	05/02	0	100%
identification stakeholders	05/02	09/02	0	100%
communication plan	09/02	10/02	0	100%
Identification of requirements	**11/02/21**	**17/02/21**	**0h**	**100%**
functional requirements	11/02	12/02	0	100%
non functional requirements	12/02	15/02	0	100%
system requirements	15/02	17/02	0	100%
Project managenent	**19/02/21**	**02/03/21**	**0h**	**100%**
project management plan	19/02	23/02	0	100%
cost estimation	23/02	25/02	0	100%
risk analysis	26/02	02/03	0	100%
project timeline reviewed	02/03	02/03	0	100%
Designing	**05/03/21**	**09/03/21**	**0h**	**100%**
designing main page	05/03	08/03	0	100%
review and finalizing	08/03	09/03	0	100%
Coding and implementation front ...	**10/03/21**	**22/03/21**	**0h**	**0%**
developing basic HTML structure	10/03	16/03	0	0%
developing user interface	16/03	19/03	0	0%
reviewing and finalizing structure	22/03	22/03	0	0%
Back end	**22/03/21**	**14/04/21**	**0h**	**0%**
developing basic functioning site	22/03	29/03	0	0%
creating database for users	01/04	05/04	0	0%
integrating the database	05/04	07/04	0	0%
module implementation phase1	07/04	09/04	0	0%
module phase implementaion phase 2	09/04	13/04	0	0%
reviewing and ready for testing	14/04	14/04	0	0%

RISK ANALYSIS – SWOT & RMMM



Risk Management Framework- Risks And Mitigation ...

Response	Strategy	Examples
Avoid	Risk avoidance is a strategy where the project team takes action to remove the threat of the risk or protect from the impact	• Extending the schedule • Reducing/removing scope • Change the execution strategy
Transfer	Risk transference involves shifting or transferring the risk threat and impact to a third party. Rather transfer the responsibly and ownership	• Purchasing insurance • Performance bonds • Warranties • Contract issuance (lump sum
Mitigate	Risk mitigation is a strategy were by the project team takes a action to reduce the probability of the risk occurring. This does not risk or potential impact , but rather reduces the likelihood of it becoming real	• Increasing testing • Changing suppliers to a more stable one • Reducing process complexity
Accept	Risk acceptance means the team acknowledges the risk and its potential impact, but decides not to take any preemptive action to prevent it. It is dealt with only if it occurs.	• Contingency reserve budgets • Management schedule float • Event contingency

Slide 1 of 5

WBS and Risk Management Plan

Lab Session #5

Table of Contents

1. Executive Summary

In our project, a milestone is a specific point in time within our project lifecycle which is used to measure the progress of Park My Pro toward its ultimate goal. In project management, milestones are used as signal posts for significant events, decision points, or deliverables such as: The project's start date (March 2022) and Project end date(July 2022) Risk management is the process of analysing exposure to risk and determining how to best handle such exposure. Our project undertakes a best practices approach and focuses on understanding the key risks and managing them within acceptable levels.

2. WBS With Project Schedule

< Assign team members for sub-tasks based on RACI and skill requirement>

Module (#)	Activity (#)	Sub-Task(#)	Assignee(s)	Planned Start Date	Planned End Date	Actual Start Date	Actual End Date	Status
UX Design	Design the user interface	Display the user privacy policy	Prudhivi sai ganesh	08 Mar 2022	15 Mar 2022	-	-	Open
		Create a login page for the user to enter their credentials		11 Mar 2022	20 March 2022	10 Apr 2022	10 Apr 2022	Closed
		Present information provided by the backend server		10 May 2022	20 May 2022	-	-	Open
Database management	Designing backend	Refer to what data must be exchanged between user and admin	Vijay kumar	08 Apr 2022	1 May 2022	10 Apr 2022	12 Apr 2022	Closed

		Acquire the empty parking slots from park my pro using the credentials provided		05 May 2022	09 May 2022	11 Apr 2022	15 Apr 2022	Closed
		Integrate back end to the front end of the website		24 May 2022	25 June 2022	-	-	Open
Admin	Control and repository maintenance	Create a privacy policy for the user to ensure the safety of their credentials	Sai ganesh, Vijay Kumar, Pranav Reddy	03 June 2022	30 June 2022	-	-	Open
		Maintain the server / repository		10 June 2022	14 June 2022	-	-	Open
Test releases	Testing	Test the software with multiple test cases	Pranav Reddy	14 June 2022	25 June 2022	-	-	Open
		Report bugs		20 June 2022	26 June 2022	-	-	Open

3. Risk Identification

SWOT ANALYSIS

STRENGTHS	WEAKNESSES
• Strong need to students and staff in the institution • Knowledged team We are able to respond very quick and give good customer care	• • Low reputation Small team Unforeseen problems

OPPORTUNITIES	THREATS
Broadening the team • Broadening the project • Unique product •	• Failing to use https or any other security protocols • Sudden rework of Park My Pro framework
	• Technical bugs

3.1.List (Describe) Register

<Issue can potentially occur in future and list all risks identified >

Risk ID (#)	Risk Description	Impact Description
R01	Technical Bugs	Sudden rework of the Park My Pro framework
R02	Disbanding of a team member	Not being able to complete the project before the deadline
R03	Developing Glitch	Failing to use https or any other security protocols
R04	Inability to gain user commitment	Lack of effort / diligence

3.2.Managing Risk

<Risk should be categorised So action can be derived to address these risks could become an issue in future>

Risk ID (#)	Status [Open Closed]	Risk Appetite [Accept/ Mitigate/ Transfer/Avoid]	Action	Action Owner	Target Date	Remarks
R01	Closed	Accept	Circumscribe the damage, Look into the potential problems and take precautions	Team Member	05 July 2022	

R02	Closed	Avoid/Mitigate	Training the developers skillfully without any further obstacles/ shotcomings	Team Leader	10 July 2022	
R02	Closed	Accept	Communicate the fix and resolve the issue	Team Member	10 July 2022	
R04	Closed	Accept	Frequent and clear communication with the user	Team Member	08 July 2022	

Reference

1. https://www.pmi.org/
2. https://www.projectmanagement.com/
3. https://www.tpsgc-pwgsc.gc.ca/biens-property/sngp-npms/ti-it/ervcpgpm-dsfvpmpteng.html

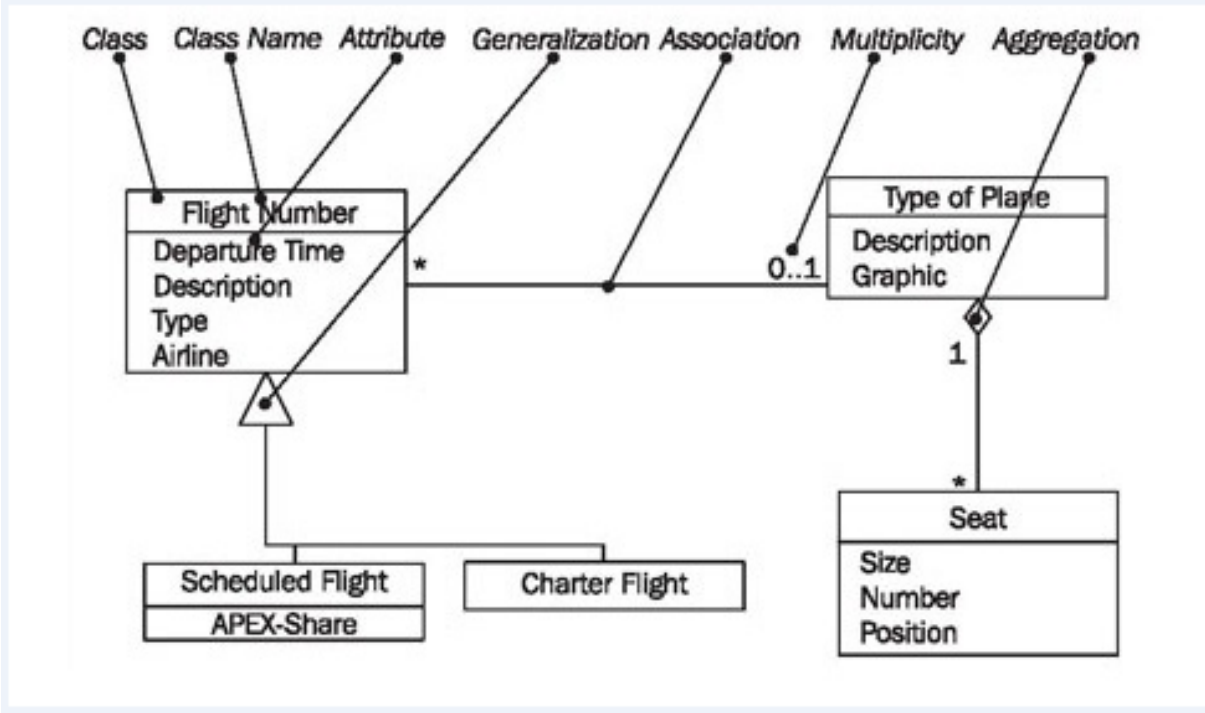
Result: Thus, the work breakdown structure with timeline chart and risk table were formulated successfully.

Chapter No. : 6

Title of the experiment : Design a System Architecture, Use case and Class Diagram

Aim:

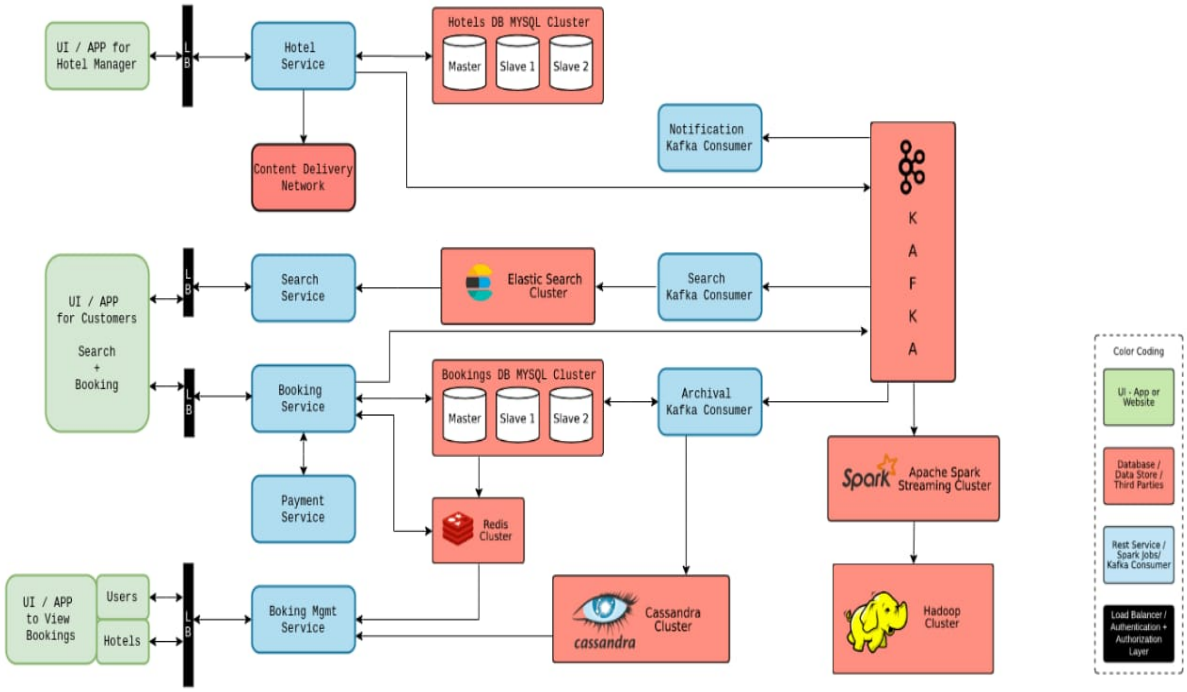
To Design a System Architecture, Use case and Class Diagram



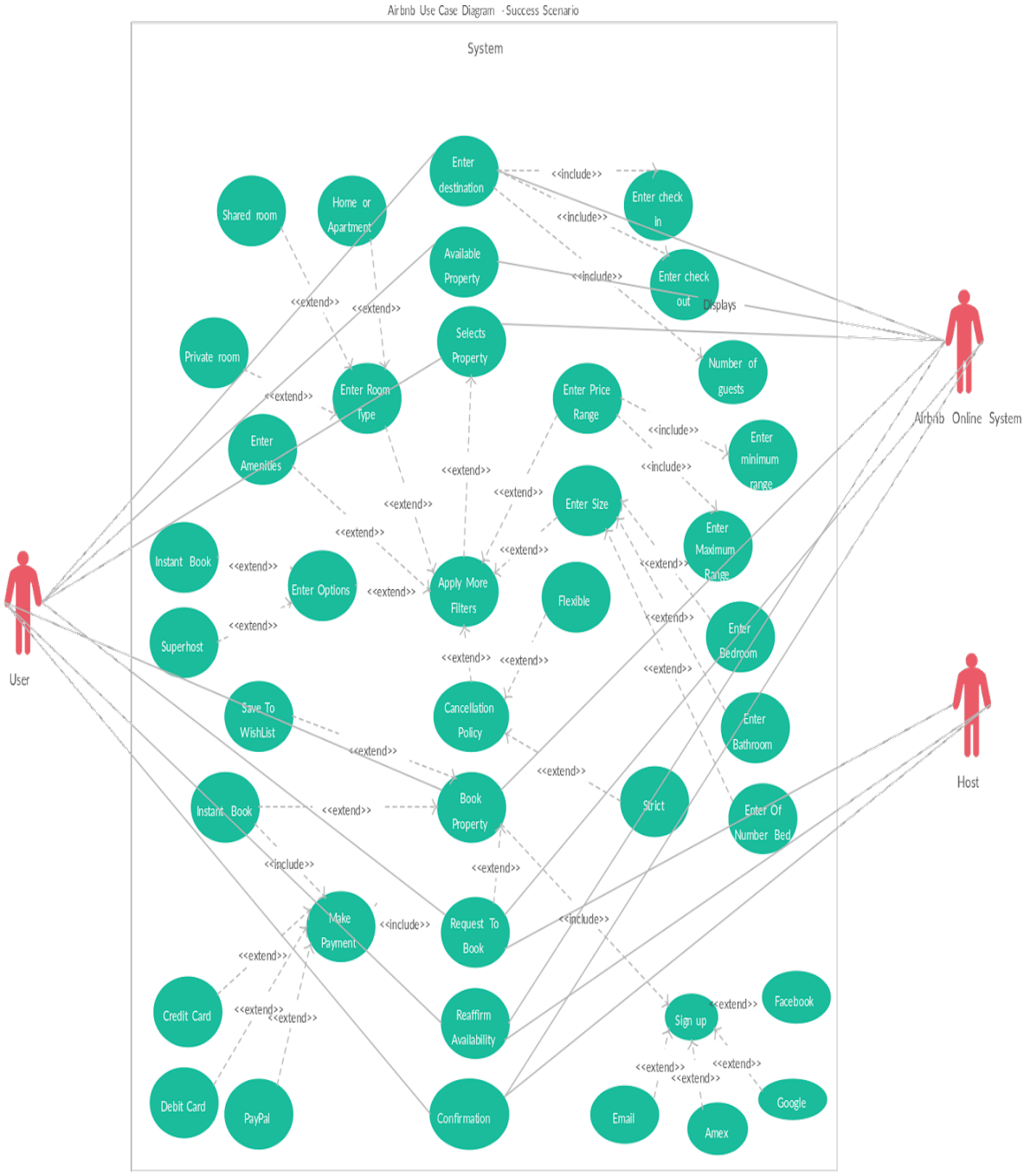
Requirements

<System Architecture, Use Case and Class Diagram>

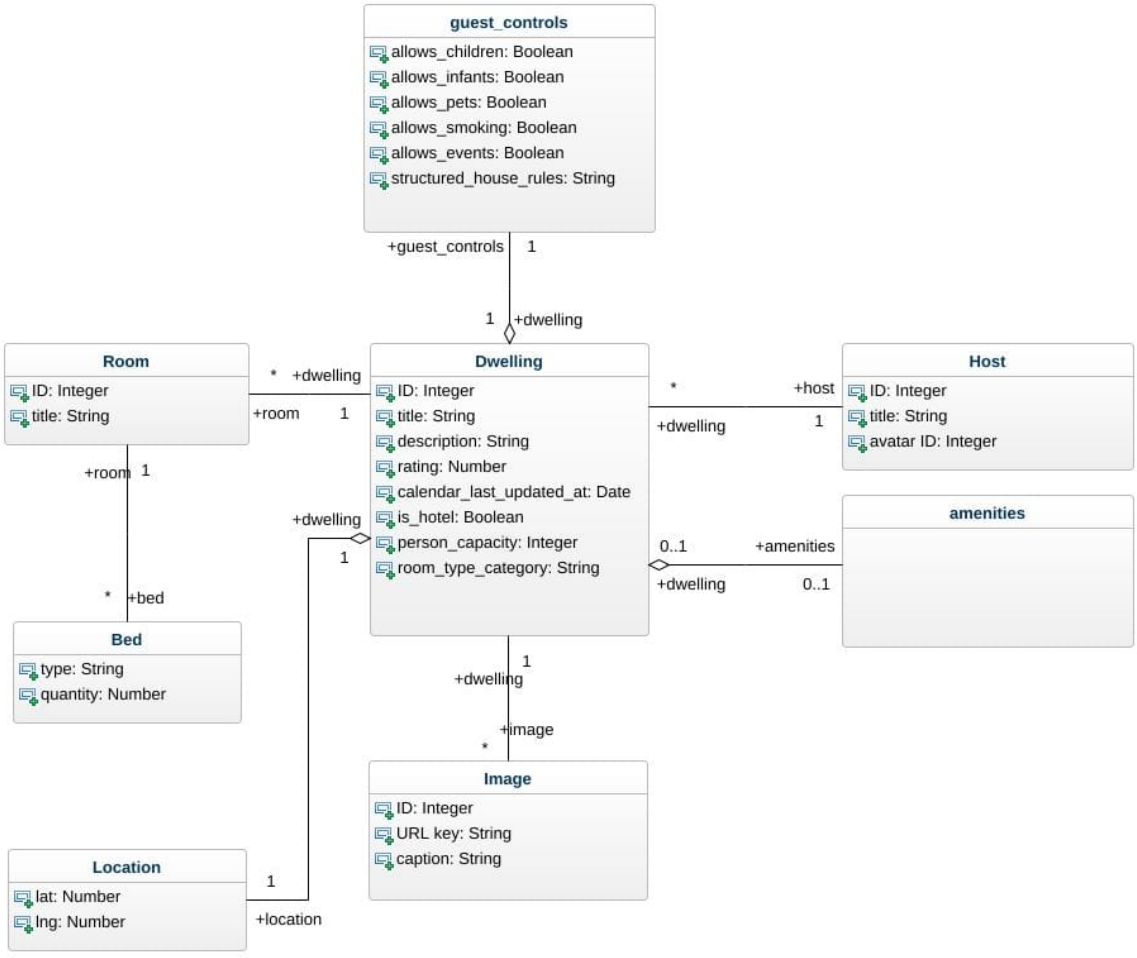
system architecture :



use case diagram :



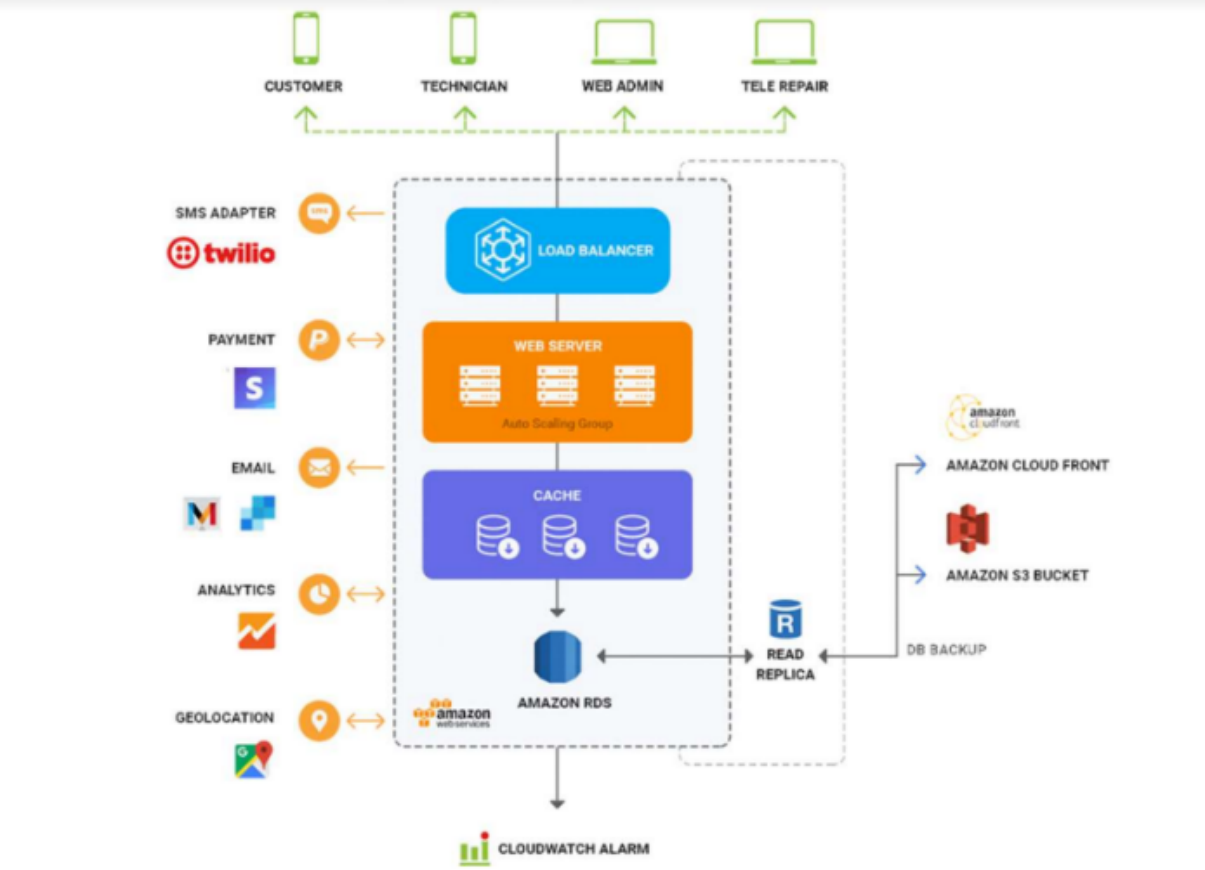
class diagram :



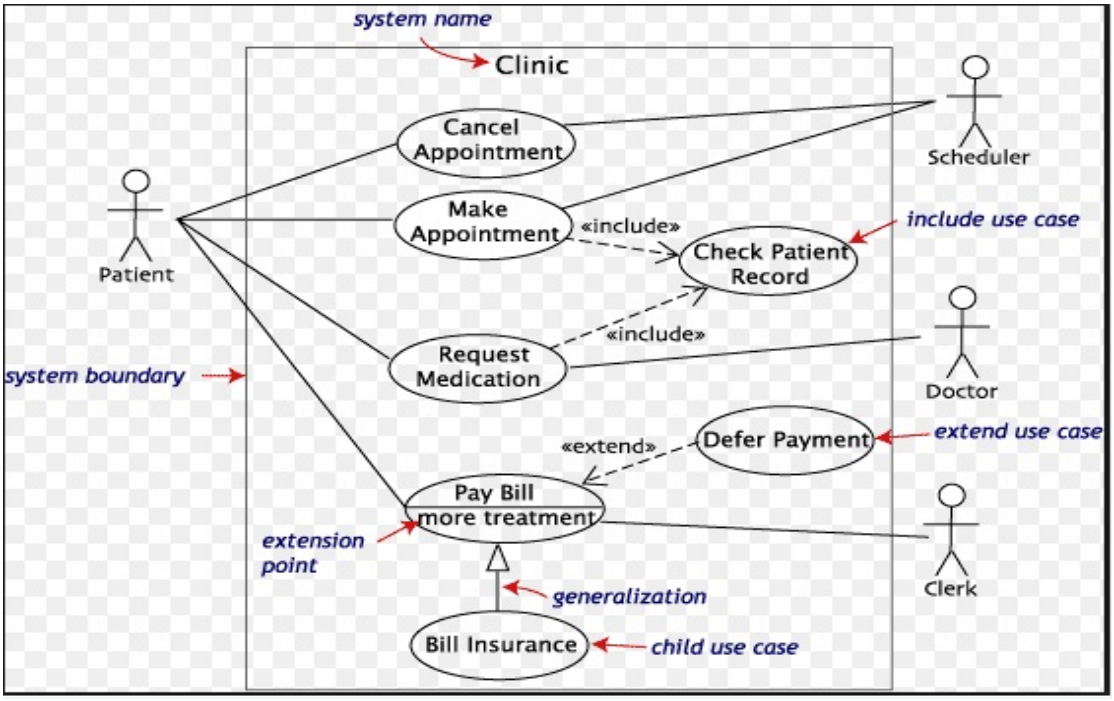
Result:

Thus, the system architecture, use case and class diagram created successfully.

SYSTEM ARCHITECTURE – Example



USE CASE DIAGRAM – Example



CLASS DIAGRAM – Example

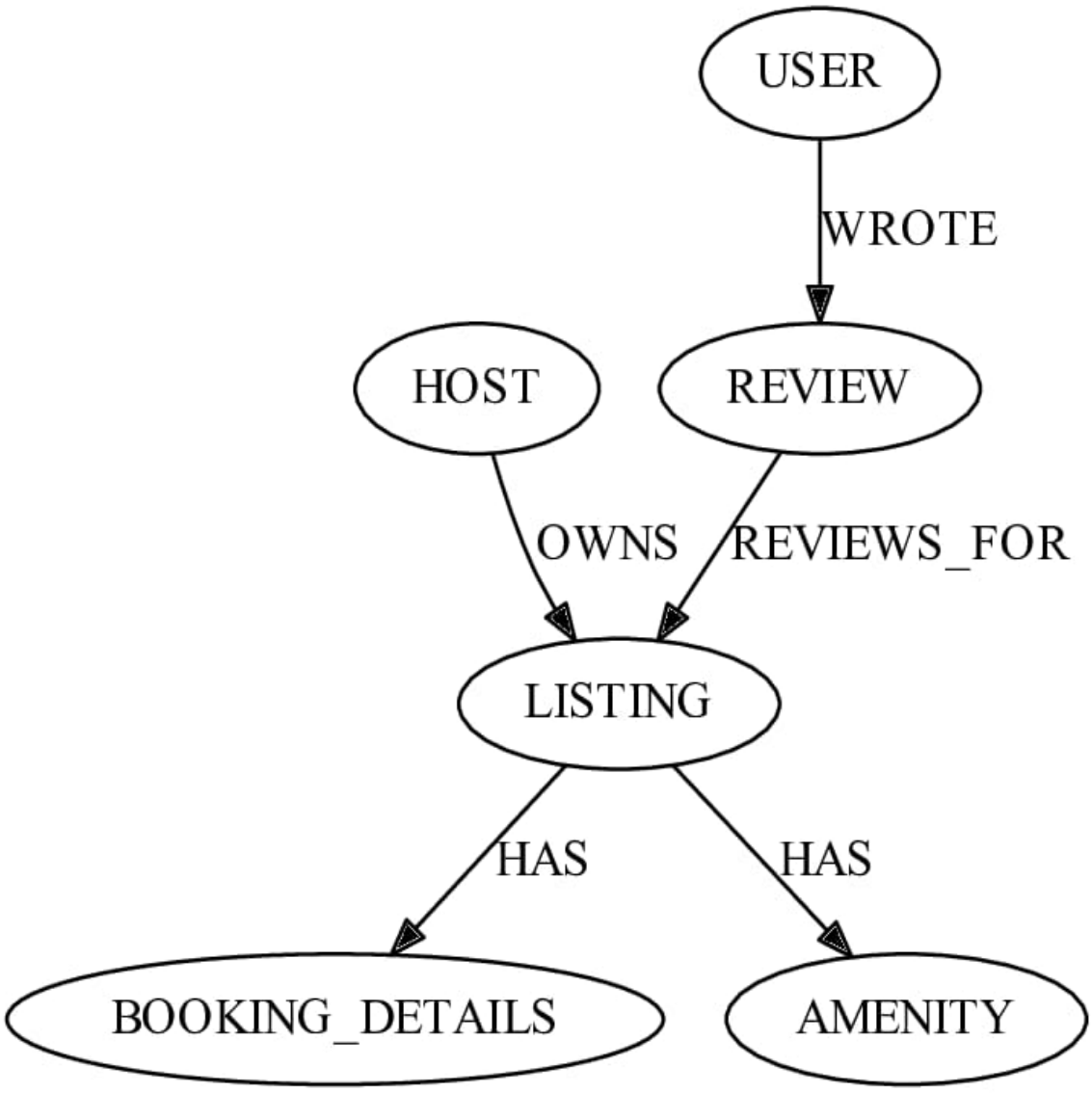
Chapter No: 7

Title of the experiment : create the Entity Relationship Diagram

Aim

To create the Entity Relationship Diagram

<ER Diagram >



<u>*/ ER Diagram, Notation and Example</u>

What is ER Diagram?

- ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.
- ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.
- At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

What is ER Model?

- ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyze data requirements to produce a well-designed database.
- ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database.
- ER Modeling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

Why use ER Diagrams?

Here, are prime reasons for using the ER Diagram

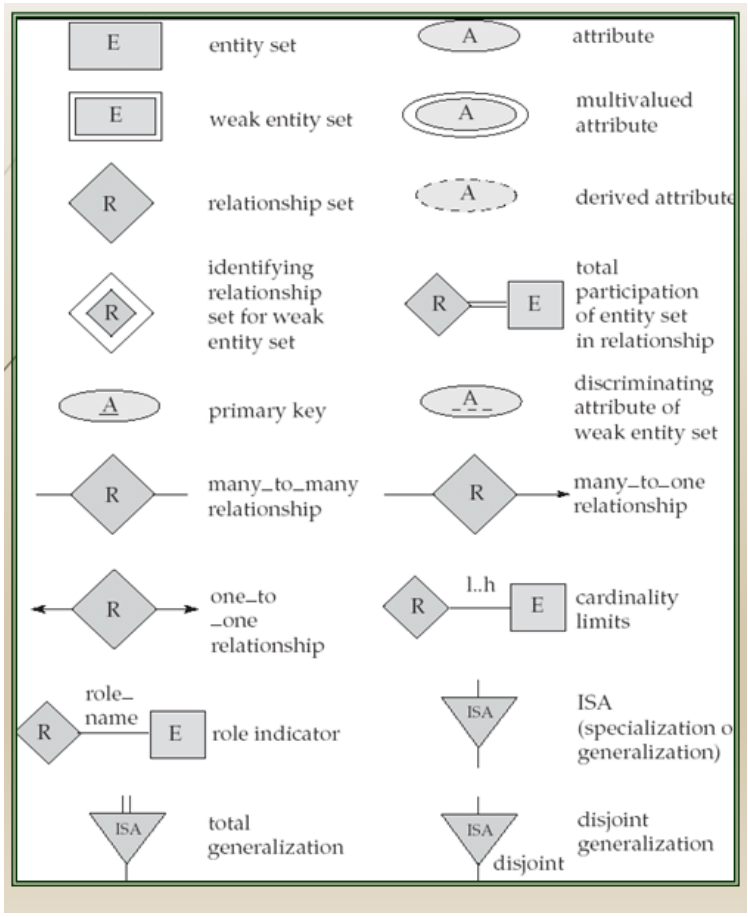
- Helps you to define terms related to entity relationship modeling
- Provide a preview of how all your tables should connect, what fields are going to be on each table
- Helps to describe entities, attributes, relationships
- ER diagrams are translatable into relational tables which allows you to build databases quickly
- ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
- The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
- ERD Diagram allows you to communicate with the logical structure of the database to users

Components of the ER Diagram

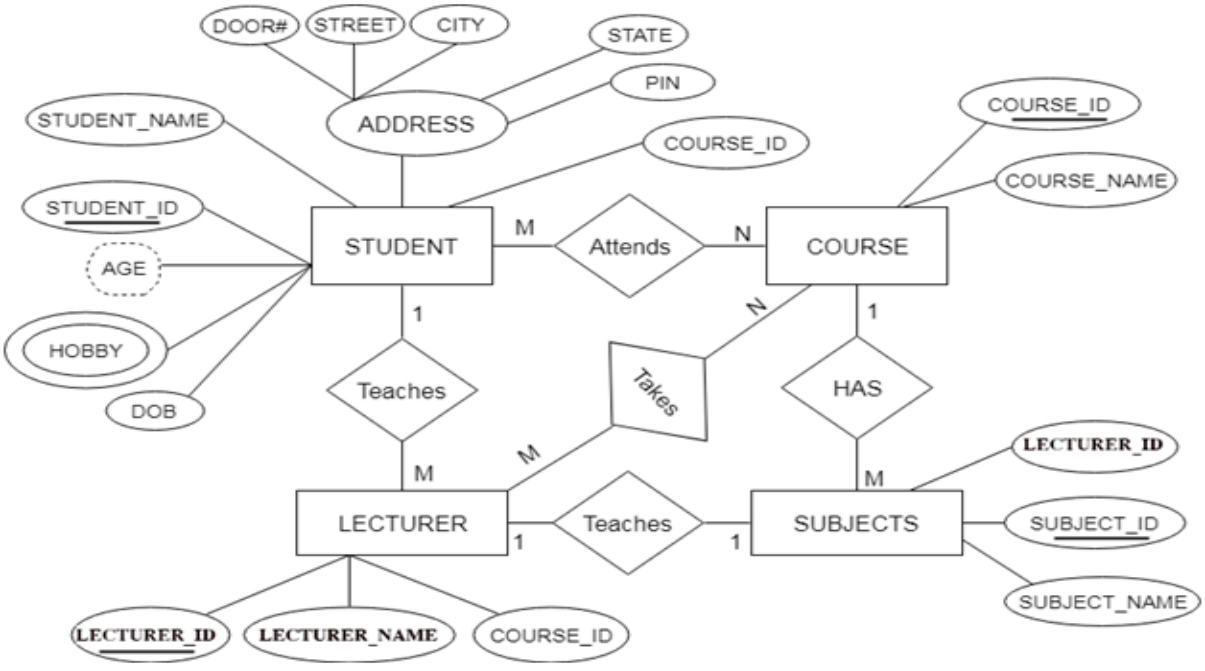
This model is based on three basic concepts: Entities, Attributes, Relationships

ER Diagram – Notations

- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
- Double ellipses represent multivalued attributes.
- Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes



ER Diagram of University Database



ADDITIONAL NOTES

- A database can be modeled as a collection of entities, relationship among entities.
- An entity is an object that exists and is distinguishable from other objects.
Example: specific person, company, event, plant
- Entities have attributes.
Example: people have names and addresses
- An entity set is a set of entities of the same type that share the same properties.
Example: set of all persons, companies, trees, holidays

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- We express cardinality constraints by drawing either a directed line (->), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.
Example: customer = (customer-id, customer-name, customer-street, customer-city)
loan = (loan-number, amount)
- Domain – the set of permitted values for each attribute
- Attribute types:
1. Simple and composite attributes.
2. Single-valued and multi-valued attributes
E.g. multivalued attribute: phone-numbers
3. Derived attributes-Can be computed from other attributes
E.g. age, given date of birth

Cardinality
- For a binary relationship set the mapping cardinality must be one of the following types:
1. One to one
A customer is associated with at most one loan via the relationship borrower. A loan is associated with at most one customer via borrower
2. One to many
A loan is associated with at most one customer via borrower, a customer is associated with several (including 0) loans via borrower
3. Many to one
A loan is associated with several (including 0) customers via borrower, a customer is associated with at most one loan via borrower
4. Many to many
A loan is associated with several (including 0) customers via borrower, a customer is associated with several loans (including 0) via borrower

Weak Entity Set
- An entity set that does not have a primary key is referred to as a weak entity set and represented by double outlined box in E-R diagram.

Example : Consider the entity set payment which got three attributes : payment_number, payment_date and payment_amount. Payment numbers are sequential starting from 1 generally separately for each loan. Although each payment entity is distinct, payments for different loans may share the same payment number. Thus this entity set does not have a primary key.

Discriminator

- The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set

Example: discriminator of weak entity set payment is the attribute payment_number since for each loan a payment number uniquely identifies one single payment for that loan.

Specialization-Generalization-ISA

- E-R model provides means of representing these distinctive entity groupings
- Process of designating subgroupings within an entity set is called specialization depicted by triangle component labelled ISA ("is a")
- Bottom up design process in which multiple entity sets are synthesized into higher level entity set - Generalization
- ISA relationship may also be referred to as superclass-subclass relationship
- Higher and lower level entity sets are designated by the terms superclass and subclass.
- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.

Total & Partial Participation

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set

E.g. participation of loan in borrower is total, every loan must have a customer associated to it via borrower

- Partial participation: some entities may not participate in any relationship in the relationship set

Example: participation of customer in borrower is partial

Cardinality limits

- Cardinality limits can also express participation constraints
- Minimum and maximum cardinality is expressed as l..h where l is the minimum and h is the maximum cardinality
- Minimum value of 1 indicates total participation of entity set in relationship set
- Maximum value of 1 indicates entity participates in atmost one relationship set.
- Maximum value of * indicates no limit

Role indicator

- Entity sets of a relationship need not be distinct
- The labels "manager" and "worker" are called roles; they specify how employee entities interact via the works-for relationship set.
- Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.
- Role labels are optional, and are used to clarify semantics of the relationship

Disjoint Generalization

- Disjointness constraint requires that an entity belong to more than one lower level entity set.

Example: account entity can satisfy only one condition for account_type attribute ; entity can either be savings or chequing account but not both.

Result:

Thus, the entity relationship diagram was created successfully.

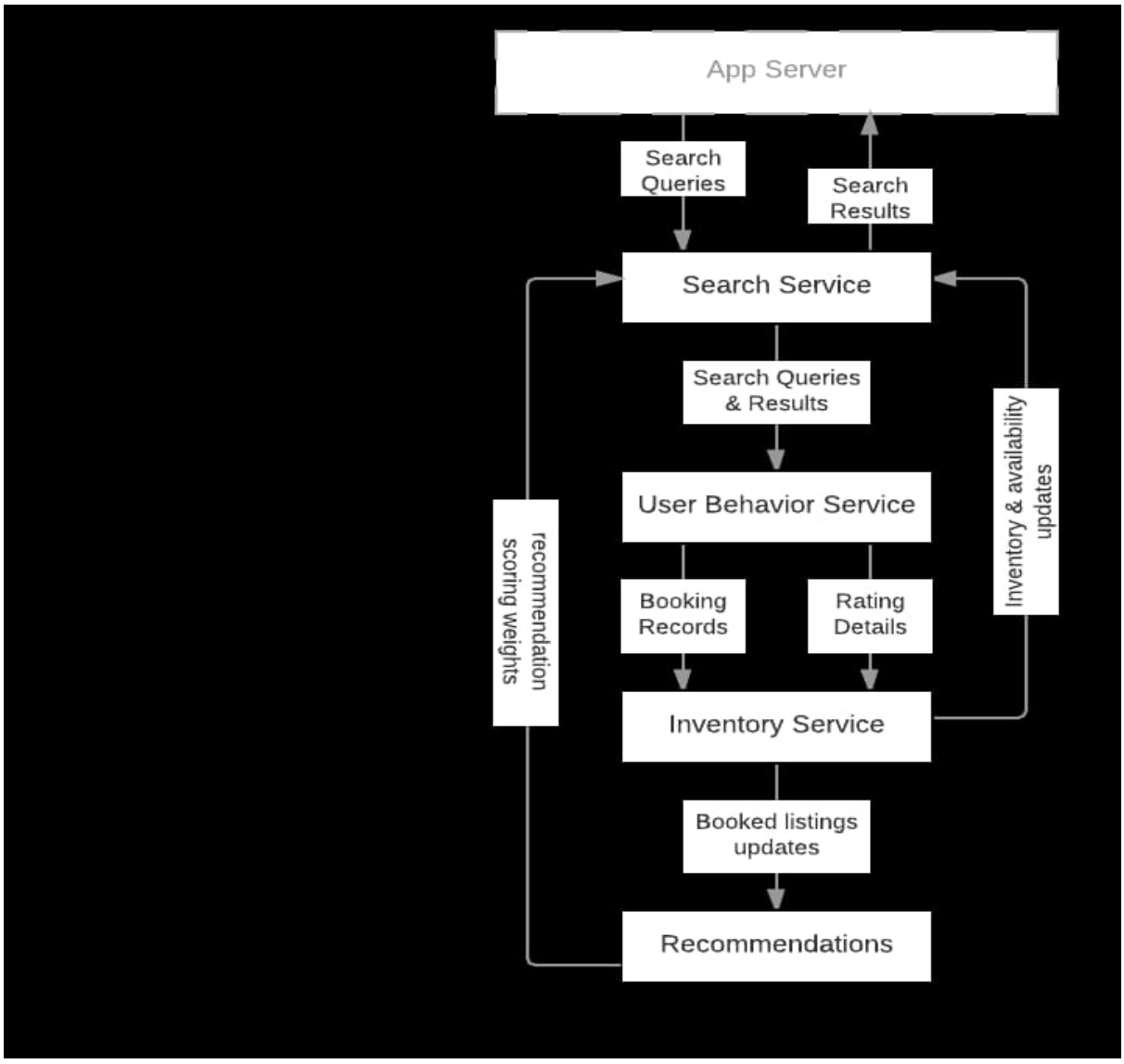
Chapter No: 8

Title of the experiment : Develop the data flow diagram up to level 1 for the Greet Hosts

Aim:

To develop the data flow diagram up to level 1 for the Greet Hosts

<DFD >



Data Flow Diagram

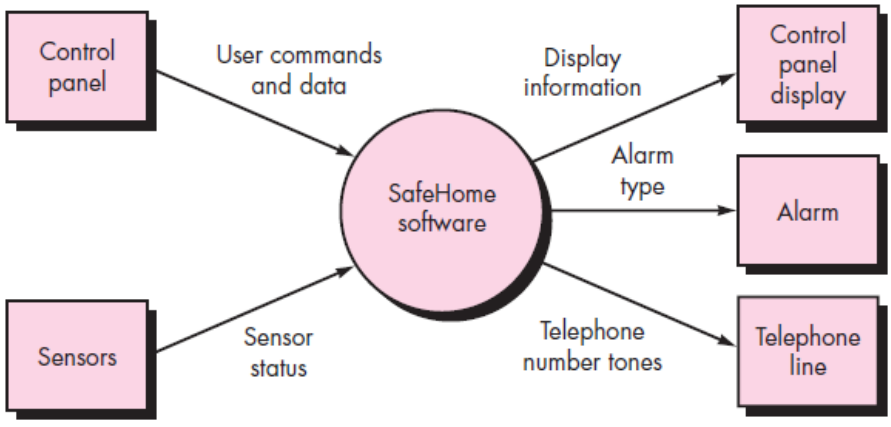
The DFD takes an input-process-output view of a system. That is, data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects are represented by labeled arrows, and transformations are represented by circles (also called bubbles). The DFD is presented in a hierarchical fashion. That is, the first data flow model (sometimes called a level 0 DFD or context diagram) represents the system as a whole. Subsequent data flow diagrams refine the context diagram, providing increasing detail with each subsequent level.

The data flow diagram enables you to develop models of the information domain and functional domain. As the DFD is refined into greater levels of detail, you perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of data as it moves through the processes that embody the application.
A few simple guidelines can aid immeasurably during the derivation of a data flow diagram:
(1) Level 0 data flow diagram should depict the software/system as a
single bubble;
(2) Primary input and output should be carefully noted;
(3) Refinement should begin by isolating candidate processes, data objects, and data stores to be represented at the next level;
(4) All arrows and bubbles should be labeled with meaningful names;
(5) Information flow continuity must be maintained from level to level and
(6) One bubble at a time should be refined. There is a natural tendency to overcomplicate the data flow diagram. This occurs when you attempt to show too much detail too early or represent procedural aspects of the software in lieu of information flow.

*/ For Example

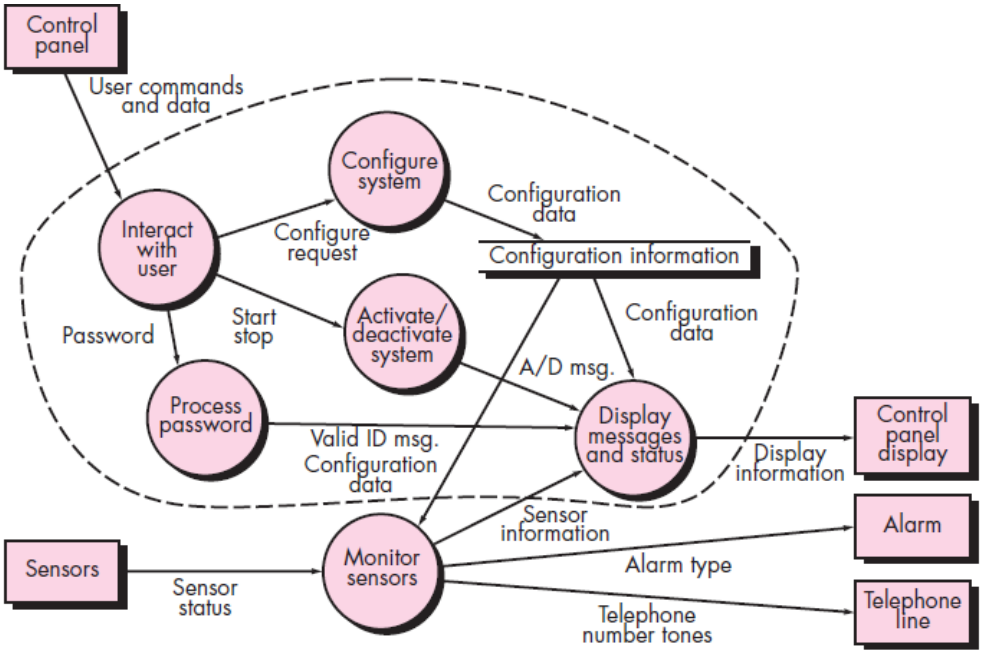
DFD Level 0

Context-level DFD for the *SafeHome* security function



DFD Level 1

Level 1 DFD for the *SafeHome* security function



Result:

Thus, the data flow diagrams have been created for the Airbnb

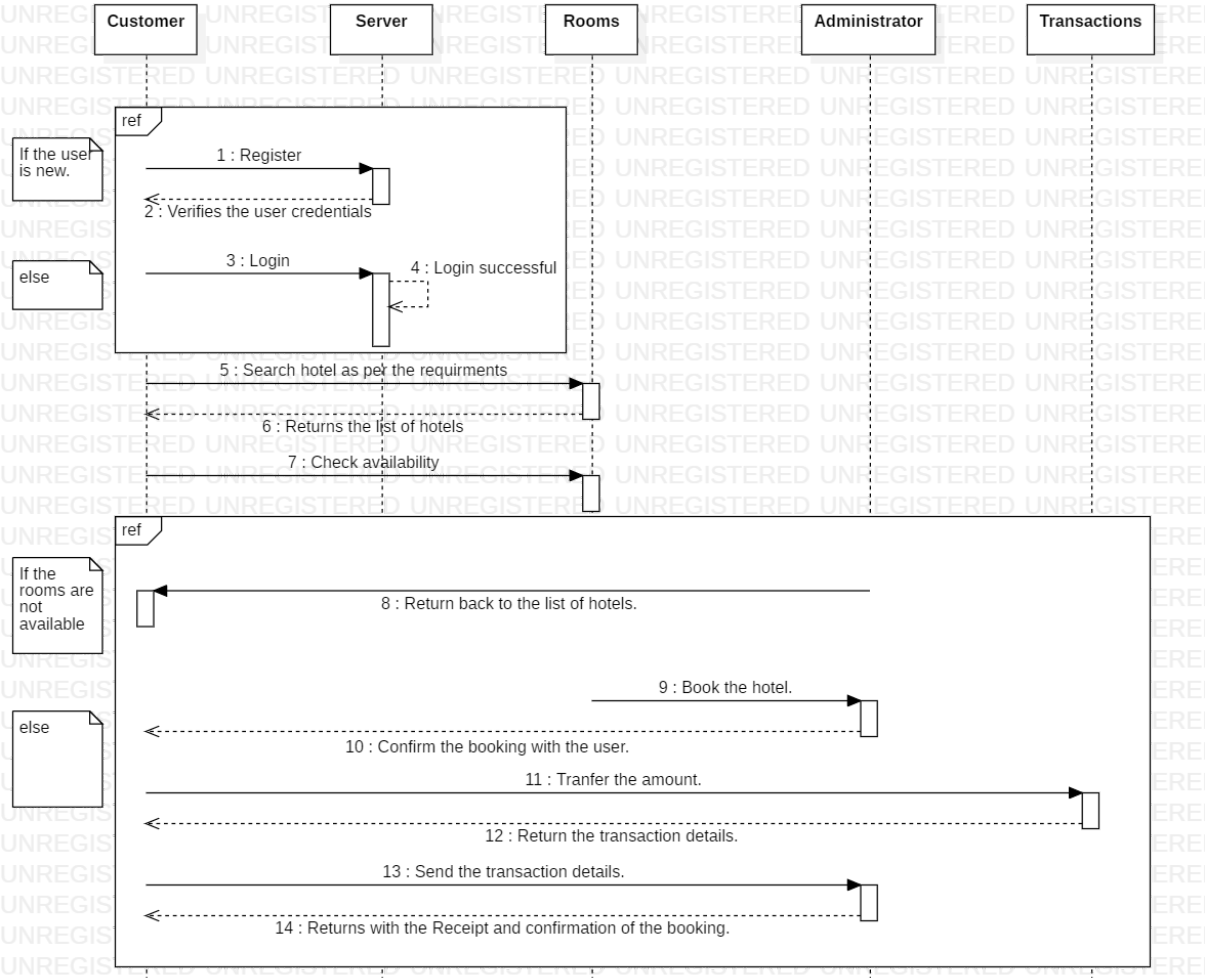
Chapter No. : 9

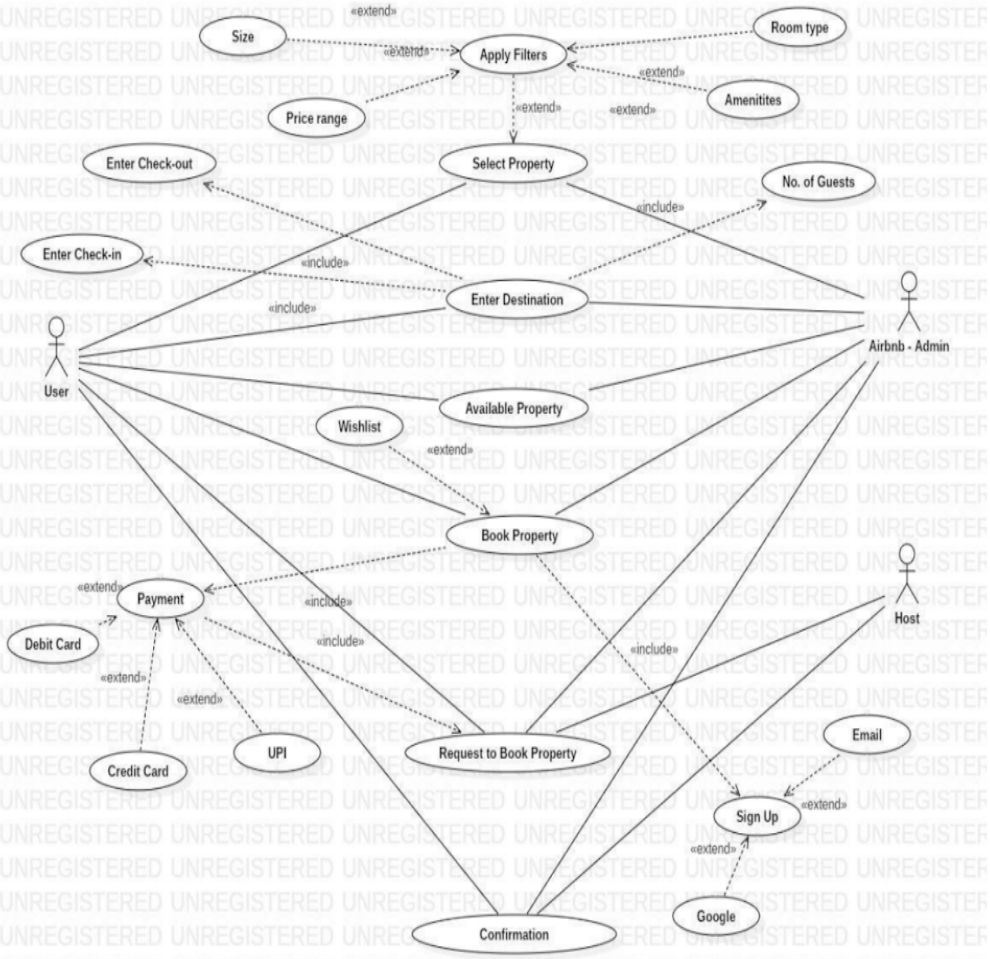
Title of the experiment : create the sequence and collaboration diagram for the GreetHosts

Aim:

To create the sequence and collaboration diagram for the GreetHosts

Sequence and Collaboration Diagram



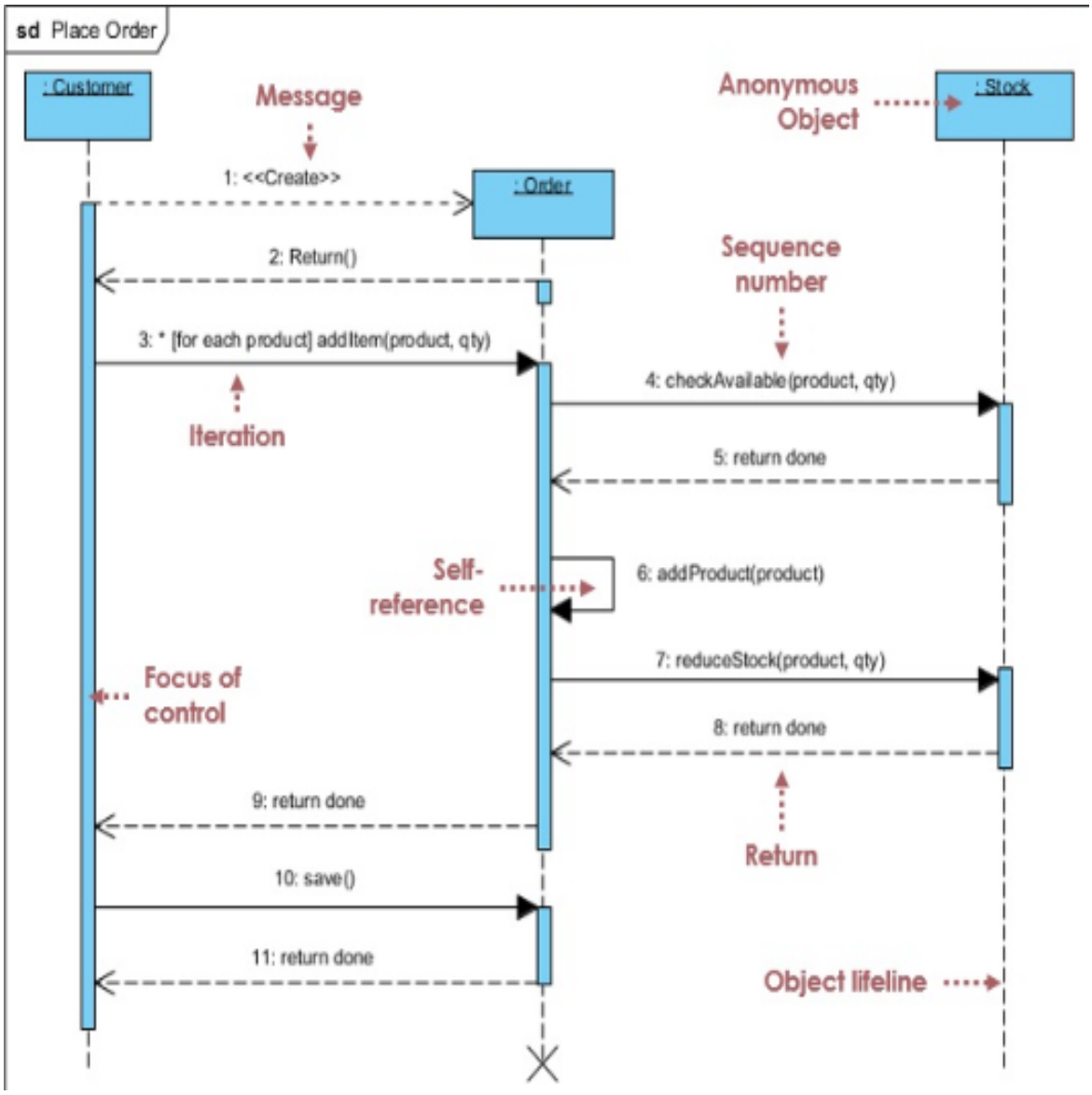


Result:

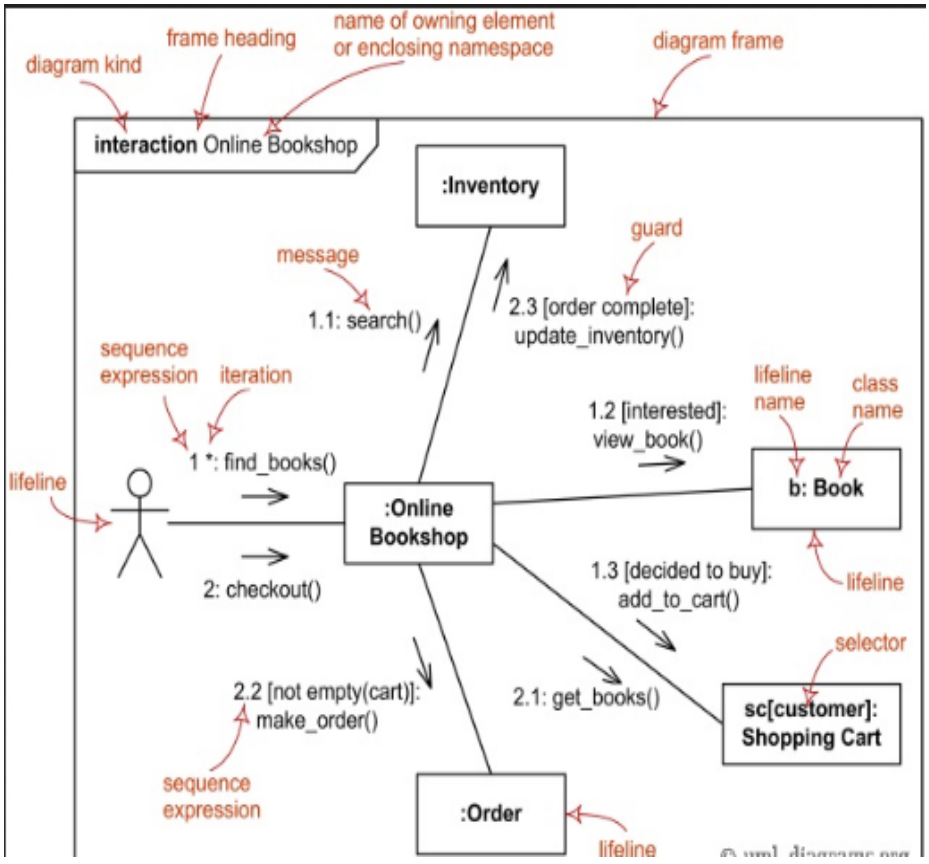
Thus, the sequence and collaboration diagrams were created for the GreetHosts.

***/ For Example**

Sequence Diagram



Collaboration Diagram



Chapter No. : 10

Title of the experiment : Develop the testing framework and/or user interface framework for the GreetHosts

Aim:

To develop the testing framework and/or user interface framework for the GreetHosts

<Incorporate the necessary information regarding testing/user interface of the project>

User interface of the project:

Our frontend of the project will consist of a website which includes application of JavaScript and CSS. The website will consist of search bar through which the user will interact with the website and obtain the recommendations based on it. The search bar will be based on python and Machine learning models which will interact with the database systems. The website will interact with the API'S from the dataset and the obtained data will be shown on the website. The admin will manage the details of the users which in turn will avoid the bots and intruders ruin our website and hence increasing the user-experience.

Testing:

The website will be tested thoroughly on various devices so that our website don't get crashed on a set of devices. The testing will be done by sharing a link to a piece of people on various devices so that we can check whether the website is working properly or not. There are various types of testing phases which will be discussed below:

Functional testing: It checks for different functions that are performed or offered by a website. It checks for different activities that a website can perform:

- It tests the JavaScript and CSS that is been used to build the website.
- It tests the forms that are present on the website.
- It checks for the links of all types present on the web page.

Compatibility Testing: Compatibility testing is done on the basis of different contexts. The compatibility testing contexts are listed below in the list:

- **Based on Browser:** It checks for the working of the website in different browsers such as Firefox, Chrome, Safari, etc.

- **Based on Operating systems:** It checks for the compatibility of the website on different operating systems like: Windows, Linux, and Mac.

Usability testing: Usability testing is the most important testing that has to be done before deploying a website for live use. This testing is performed by users or a team of clients.

- It checks the controls of the website.
- Checks the content of web pages.
- Checks whether the user requirements are satisfied or not.

Interface testing: Interface testing is done to check or verify the interface and the data flow from one system to another system. It checks for data flow from the **Web server-side** and **Database server** side.

- **Database Server:** Here it checks for the data that is transported from the database is as required or not. It measures the correctness of the data that is delivered.
- **Web server:** It will check for the webserver that it handles all the queries that are being asked by the users.

Performance testing: It checks for the performance of the website under different conditions.

- It checks for the response time for any query at different connections speed.
- It tests for the load that a website can handle under various conditions.
- It checks for the performance of the website when the load exceeds the upper limit of the maximum load.
- It checks for the recovery of the website after the breakdown of the website due to excessive load.

<u>*/ For example</u>

Executive Summary

<<defines the scope, objective, and approach to test the software application>>

Test Plan

Scope of Testing

<<summarize the scope of testing >

Functional: Are all modules covered? Any exception for any modules ? Does automation cover all functional test cases or Regression – Critical Path Test Cases ?

Non-Functional: Are all NFR (Non-Functional Requirements) covered?

Types of Testing, Methodology, Tools

Category	Methodology	Tools Required
Functional Requirements	Manual	Word Template

Result:

Thus, the testing framework/user interface framework has been created for GreetHosts.

Chapter No: 11

Title of the Experiment: Develop the test cases manual for the GreetHosts

Aim:

To develop the test cases manual for the GreetHosts

Test Case

Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1.	System Performances	Can a system provide recommendations under 50ms at the 99.99th percentile?	User will enter the car name in search box, then system will recommend car in least possible time	List of cars will be available.	List of cars	Pass	Success

2	Verify user registration	Accept valid mobile number.	1. User clicks on User Registration link 2. Enter the mobile Number on the text box 3. Click Register button	User should be taken to the next page for entering more user details	User should be taken to the next page for entering more user details	Pass	Success

Non-Functional Test Cases

Test ID (#)	**Test Scenario**	**Test Case**	**Execution Steps**	**Expected Outcome**	**Actual Outcome**	**Status**	**Remarks**

1	Security	How a system is safeguarded against deliberate and sudden attacks from internal and external sources?	A password should be in encryptd format • Application or System should not allow invalid users • Check cookies and session time for application	Secure	Secure	Pass	Success
2	Reliability	The extent to which any software system continuously performs the specified functions without failure.	• How system is giving output while searching same car every time?	Reliable output.	Reliable outcome	Pass	Success
3	Efficiency	The extent to which any software system can handles capacity, quantity and	• How the system behaves under heavy load?	Speed of computer is fast.	Speed of computer is moderate.	Pass	Success

		resp onse time .					
4	Usab ility	The ease with which the user can learn, operate, prepare inputs and outputs	• Is the system easy to use? • Help is provided for the users to understand the application/w eb site.	Emails and phone conversa tions are used to provide assistanc e.	Emails and phone conversa tions are used to provide assistanc e.	Pa ss	Succ ess

		through interactio n with a syste m.					

Result:

Thus, the test case manual has been created for the Car sales system.

Chapter No: 12

Title of the Experiment: Provide the details of architectural Design/Framework/Implementation

Aim: To provide details of architectural design/framework/implementation

<provide the details of architectural design/implementation with screenshots Minimum three modules to be completed use of software on their choice to implement>

Full documentation with the coding

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.
      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
```

The Front Page. A book lover's retreat.

★ 4.95 · 20 reviews · Bloomington, Illinois, United States

Share Save

What this place offers

- Kitchen
- Wifi
- Dedicated workspace
- Free driveway parking on premises – 1 space
- 55" HDTV with Amazon Prime Video, Disney+, HBO Max, Hulu, Netflix
- Window AC unit
- Bath
- Shared backyard – Not fully fenced
- Hair dryer
- Refrigerator

Show all 52 amenities

₹8,822 night ★ 4.95 · 20 reviews

CHECK-IN Add date | CHECKOUT Add date

GUESTS 1 guest

Check availability

Report this listing

```
<!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.
    You can add webfonts, meta tags, or analytics to this file.
    The build step will place the bundled scripts into the <body> tag.
    To begin the development, run `npm start` or `yarn start`.
    To create a production bundle, use `npm run build` or `yarn build`.
  -->
 </body>
</html>
```

Json code:

```
{
 "short_name": "React App",
 "name": "Create React App Sample",
 "icons": [
  {
   "src": "favicon.ico",
   "sizes": "64x64 32x32 24x24 16x16",
   "type": "image/x-icon"
  },
  {
   "src": "logo192.png",
   "type": "image/png",
   "sizes": "192x192"
  },
  {
   "src": "logo512.png",
   "type": "image/png",
   "sizes": "512x512"
  }
 ],
 "start_url": ".",
 "display": "standalone",
 "theme_color": "#000000",
 "background_color": "#ffffff"
}
```

Java script file:

```
import React from 'react';
import { Routes, Route } from "react-router-dom";
import Home from "./pages/Home";
import Rentals from './pages/Rentals';
import './App.css';

const App = () => {
 return(
```

```
    <Routes>
      <Route path="/" element={<Home />} />
      <Route path="/rentals" element={<Rentals />} />
    </Routes>
  )
};

export default App;
```

CSS file**:**

```
* DatePicker and Input <div> */
.sc-gGCDDS {
    outline: 0px !important;
    padding-left: 0px !important;
    padding-top: 3px !important;
    padding-bottom: 3px !important;
}

/* Select inner <div> */
.sc-fbyfCU {
    border: 0px !important;
    padding-left: 0px !important;
    padding-top: 3px !important;
    padding-bottom: 3px !important;
    height: 40px !important;
    font-size: 14px !important;
}

/* Select outer <div> */
.sc-jUosCB {
    width: 170px !important;
}

/* DatePicker and Input <input> */
.sc-faUpoM {
    height:26px !important;
    font-size:14px !important;
}

/* DatePicker <span> */
.sc-lcepkR {
    height: 70% !important;
}
```

Result:
Thus, the details of architectural design/framework/implementation along with the screenshots were provided.