**UML Diagram On ONLINE AIRLINE BOOKING SYSTEM**

**PROJECT REPORT**

**18CSC202J- OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY**

**(2018 Regulation)**

**II Year/ III Semester**

**Academic Year: 2022 -2023**

By

T.ASHISH (RA2111032010004)

B.ABHINAY(RA2111032010007)

Under the guidance of

**Dr. GOUTHAMAN. P**

**Assistant Professor**

**Department of Networking and Communications**



**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SCHOOL OF COMPUTING**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Kancheepuram**

**NOVEMBER 2022**

**BONAFIDE CERITIFICATE**

This is to certify that **18CSC202J - OBJECT ORIENTED DESIGN AND**

**PROGRAMMING LABORATORY project report** titled “**UML**

**DIAGRAM ON ONLINE AIRLINE BOOKING SYSTEM”**

is the bonafide work of **T.ASHISH (RA2111032010004),**

**B.ABHINAY(RA2111032010007)**  who took part in this project and completed with in the due time.

**Signature of the Guide** **Signature of the II Year Academic Advisor**

Dr. Gouthaman. P -------------------------

**Assistant Professor** **Professor and Head**

Department of NWC Department of NWC

SRM Institute of Science and Technology SRM Institute of Science and Technology

**About the course**:-

18CSC202J/ 8AIC203J - Object Oriented Design and Programming are 4 credit courses with L T P C as 3-0-2-4 (Tutorial modified as Practical from 2018 Curriculum onwards)

**Objectives:**

The student should be made to:

● Learn the basics of OOP concepts in C++

● Learn the basics of OOP analysis and design skills.

● Be exposed to the UML design diagrams.

● Be familiar with the various testing techniques

**Course Learning Rationale (CLR):**

The purpose of learning this course is to:

1.Utilize class and build domain model for real-time programs

2.Utilize method overloading and operator overloading for real-time application development programs

3.Utilize inline, friend and virtual functions and create application development programs

4.Utilize exceptional handling and collections for real-time object-oriented programming applications

5.Construct UML component diagram and deployment diagram for design of applications

6.Create programs using object-oriented approach and design methodologies for real-time application development

**Course Learning Outcomes (CLO):**

At the end of this course, learners will be able to:

1.Identify the class and build domain model

2.Construct programs using method overloading and operator overloading

3.Create programs using inline, friend and virtual functions, construct programs using standard templates

4.Construct programs using exceptional handling and collections

5.Create UML component diagram and deployment diagram

6.Create programs using object oriented approach and design methodologies

**Table 1: Rubrics for Laboratory Exercises**

(Internal Mark Splitup:- As per Curriculum)

|  |  |  |
| --- | --- | --- |
| **CLAP-1** | 5=(2(E-lab Completion) + 2(Simple Exercises)( from CodeZinger, and any other coding platform) + 1(HackerRank/Code chef/LeetCode Weekend Challenge) | Elab test |
| **CLAP-2** | 7.5=(2.0(E-lab Completion)+  2.0 (Simple Exercises)( from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge) | Elab test |
| **CLAP-3** | 7.5=(2.0(E-lab Completion(80 Pgms)+  2.0 (Simple Exercises)( from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge) | **2 Mark -** E-lab Completion **80 Program** Completion from 10 Session (Each session min 8 program)  **2 Mark -** Code to UML conversion GCR Exercises  **3.5 Mark - Hacker Rank** Coding challenge completion |
| **CLAP-4** | 5= 3 (Model Practical) + 2 (Oral Viva) | * **3 Mark** – Model Test * **2 Mark** – Oral Viva |
| **Total** | 25 |  |

**COURSE ASSESSMENT PLAN FOR OODP LAB**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **List of Experiments** | **Course Learning Outcomes (CLO)** | **Blooms Level** | **PI** | **No of Programs in each session** |
| 1. | Implementation of I/O Operations in C++ | CLO-1 | Understand | 2.8.1 | 10 |
| 2. | Implementation of Classes and Objects in C++ | CLO-1 | Apply | 2.6.1 | 10 |
| 3, | To develop a problem statement. 1. From the problem statement, Identify Use Cases and develop the Use Case model. 2. From the problem statement, Identify the conceptual classes and develop a domain model with a UML Class diagram. | CLO-1 | Analysis | 4.6.1 | Mini Project Given |
| 4. | Implementation of Constructor Overloading and Method Overloading in C++ | CLO-2 | Apply | 2.6.1 | 10 |
| 5. | Implementation of Operator Overloading in C++ | CLO-2 | Apply | 2.6.1 | 10 |
| 6. | Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams and Collaboration diagrams | CLO-2 | Analysis | 4.6.1 | Mini Project Given |
| 7. | Implementation of Inheritance concepts in C++ | CLO-3 | Apply | 2.6.1 | 10 |
| 8. | Implementation of Virtual function & interface concepts in C++ | CLO-3 | Apply | 2.6.1 | 10 |
| 9. | Using the identified scenarios in your project, draw relevant state charts and activity diagrams. | CLO-3 | Analysis | 4.6.1 | Mini Project Given |
| 10. | Implementation of Templates in C++ | CLO-3 | Apply | 2.6.1 | 10 |
| 11. | Implementation of Exception of Handling in C++ | CLO-4 | Apply | 2.6.1 | 10 |
| 12. | Identify the User Interface, Domain objects, and Technical Services. Draw the partial layered, logical architecture diagram with UML package diagram notation such as Component Diagram, Deployment Diagram. | CLO-5 | Analysis | 4.6.1 | Mini Project Given |
| 13. | Implementation of STL Containers in C++ | CLO-6 | Apply | 2.6.1 | 10 |
| 14. | Implementation of STL associate containers and algorithms in C++ | CLO-6 | Apply | 2.6.1 | 10 |
| 15. | Implementation of Streams and File Handling in C++ | CLO-6 | Apply | 2.6.1 | 10 |

**LIST OF EXPERIMNENTS FOR UML DESIGN AND MODELLING:**

**To develop a mini-project by following the exercises listed below.**

1. To develop a problem statement.

2. Identify Use Cases and develop the Use Case model.

3. Identify the conceptual classes and develop a domain model with UML Class diagram.

4. Using the identified scenarios, find the interaction between objects and represent them

using UML Sequence diagrams.

5. Draw relevant state charts and activity diagrams.

6. Identify the User Interface, Domain objects, and Technical services. Draw the partial

layered, logical architecture diagram with UML package diagram notation.

**Suggested Software Tools for UML:**

StarUML, Rational Suite, Argo UML (or) equivalent, Eclipse IDE and Junit

ABSTRACT

UML Diagram (Unified Modeling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML describes the real-time systems, it is very important to make a conceptual model and then proceed gradually. The conceptual model of UML can be mastered by learning the following three major elements

1) UML Building block

2) Rules to connect the building blocks

3) Common mechanism of UML

**UML** is powerful enough to represent all the concepts that exist in object-oriented analysis and design. UML diagrams are representation of object-oriented concepts only. Thus, before learning UML, it becomes important to understand OO concept in detail.

1) Objects

2) Class

3) Abstraction

4) Polymorphism

5) Inheritance

6) Encapsulation

**Types Of UML Diagrams:**

1) Class Diagram

2) Use Case Diagram

3) Sequence Diagram

4) Collaboration Diagram

5) State Diagram

6) Activity Diagram

7) Component Diagram

8) Deployment Diagram

**MODULE DESCRIPTION**

**Class** − Class represents a set of objects having similar responsibilities.

**Interface** − Interface defines a set of operations, which specify the responsibility of a class.

**Collaboration** −Collaboration defines an interaction between elements.

**Use case** −Use case represents a set of actions performed by a system for a specific goal.

**Component** −Component describes the physical part of a system.

**Node** − A node can be defined as a physical element that exists at run time.

**Interaction** − Interaction is defined as a behavior that consists of a group of messages exchanged among elements to accomplish a specific task.

**Package** − Package is the only one grouping thing available for gathering structural and behavioral things.

Association is basically a set of links that connects the elements of a UML model. It also describes how many objects are taking part in that relationship.

Generalization can be defined as a relationship which connects a specialized element with a generalized element. It basically describes the inheritance relationship in the world of objects.

\*UML DIAGRAM\*

Online Airline Ticket Booking Management System

**AIM:**To study the UML diagrams of use case and class diagrams of Online Airline Ticket Booking Management System.

**Problem Description:**To study the UML diagrams of use case and class diagrams of Online Airline Ticket Booking Management System. .With the emegence of the digital era in these modern times online Airline ticket booking system are used more often than ever.with the frequent use of these tools comes in the issue of frequently visiting airport or completing booking which take often more time so this is theone which makes easier than visiting Airport.

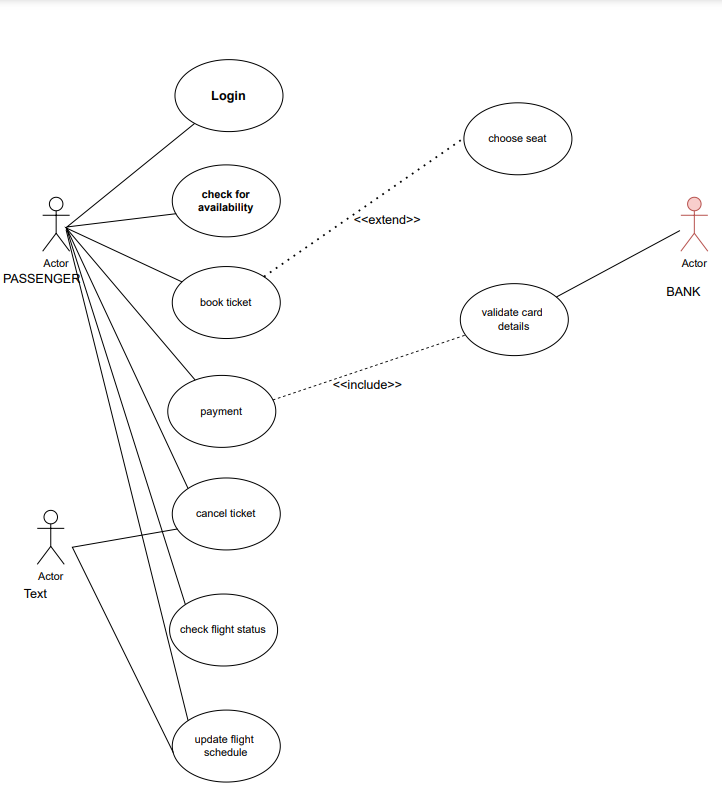
Done by:

T. Ashish(RA2111032010004)

B. Abhinay Reddy(RA211032010007)

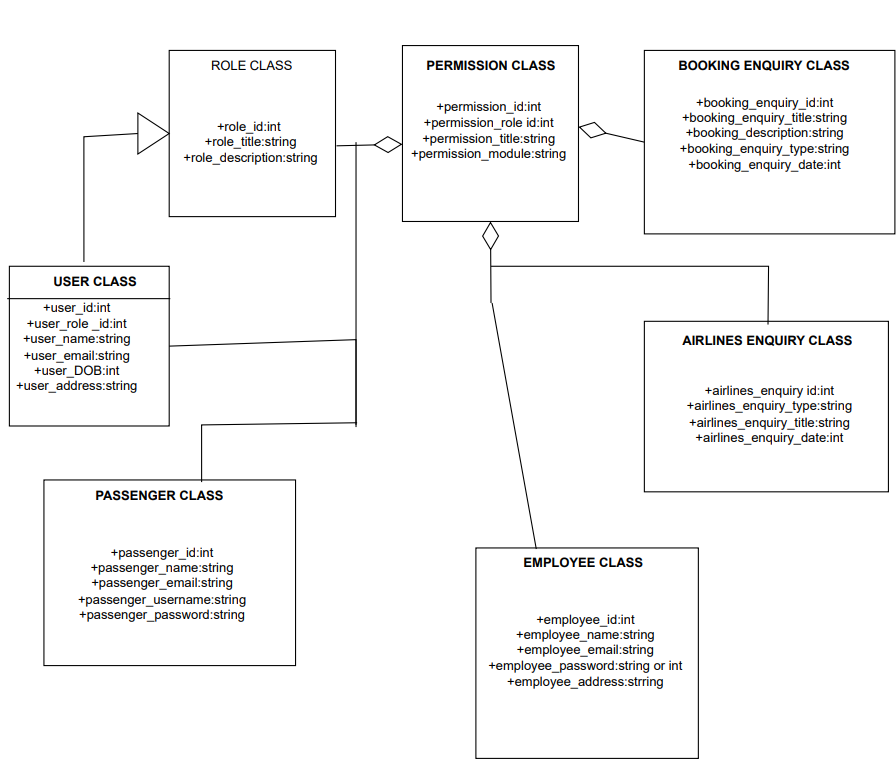
**USE CASE DIAGRAM**

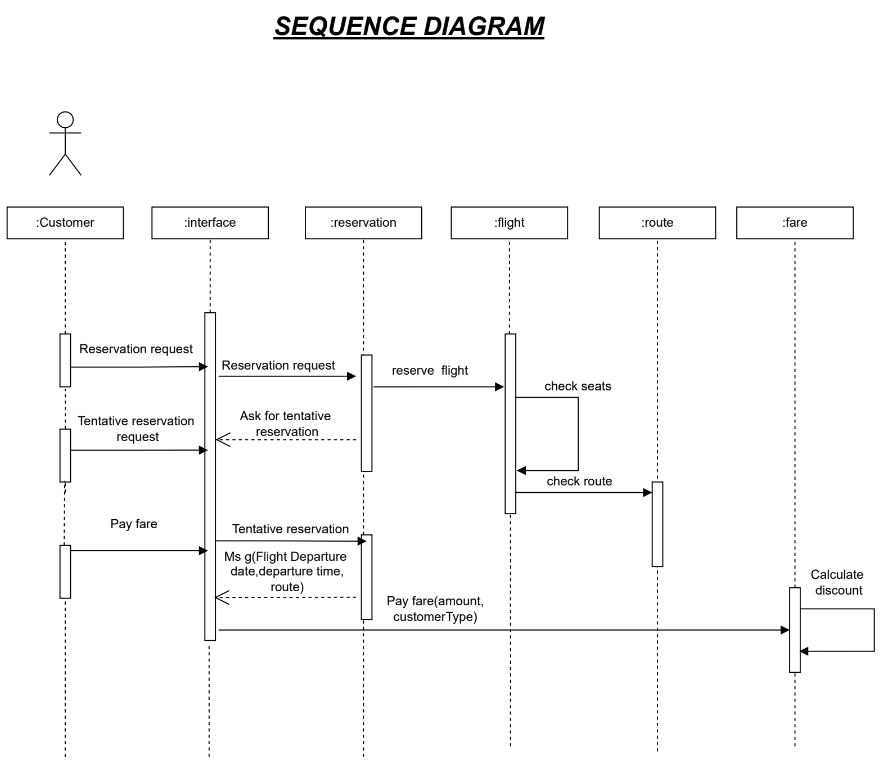
A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships.



**CLASS DIAGRAM**

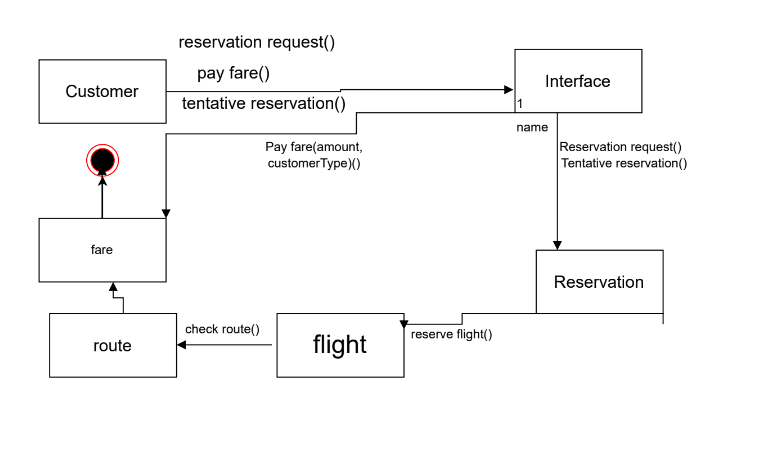
The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.;





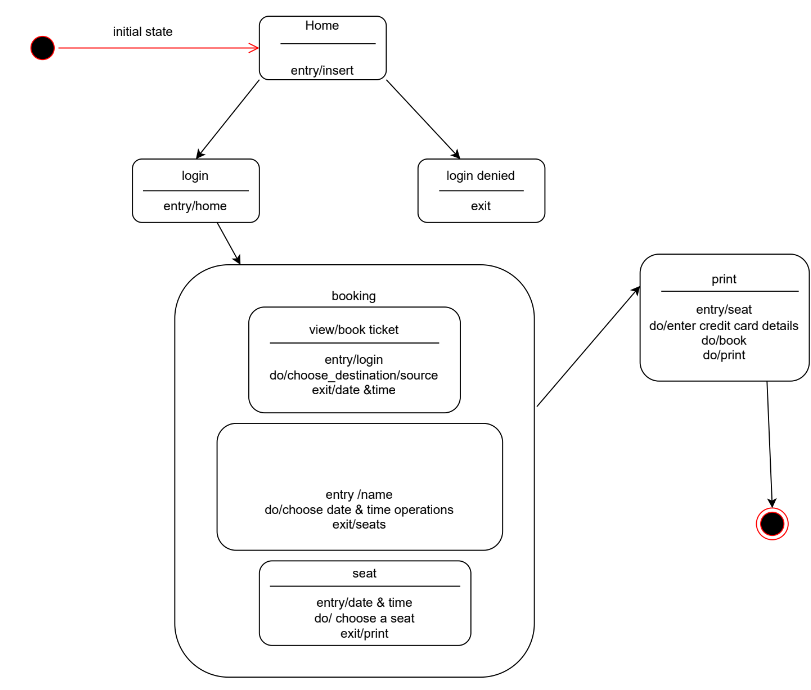
The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time.

**COLLABRATION DIAGRAM**



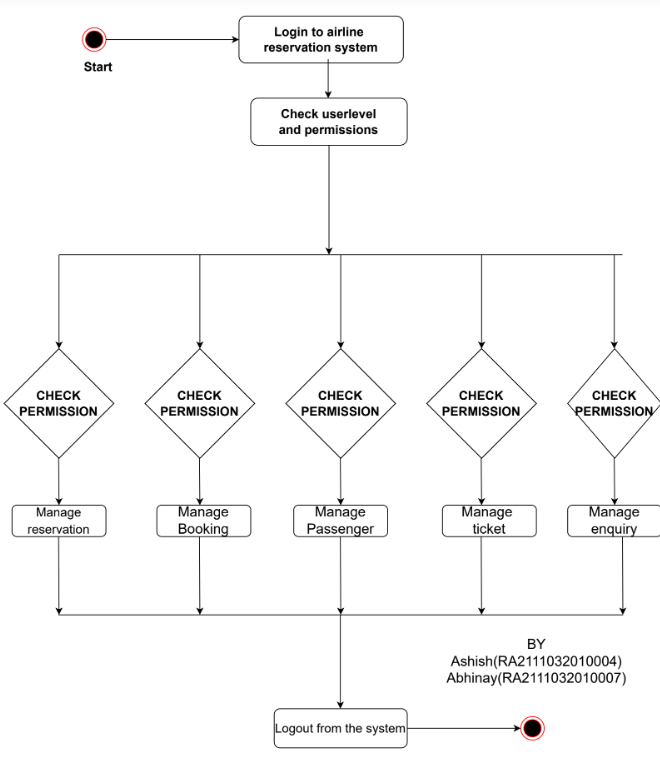
The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming.

**STATE CHART DIAGRAM**

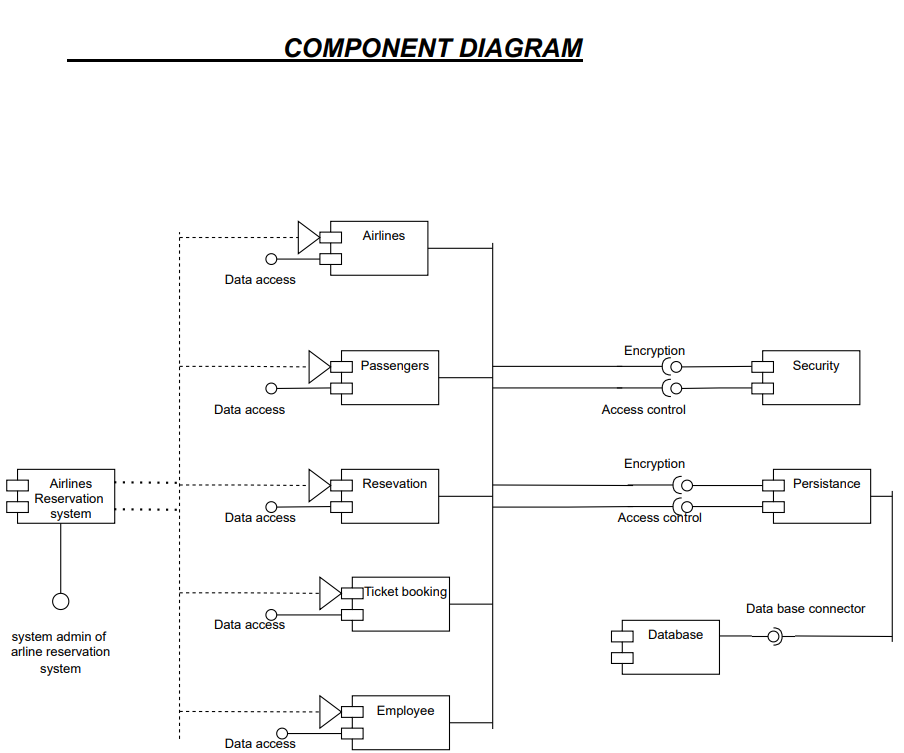


The state machine diagram is also called the Statechart or State Transition diagram, which shows the order of states underwent by an object within the system. It captures the software system's behavior. It models the behavior of a class, a subsystem, a package, and a complete system.

**ACTIVITY DIAGRAM**

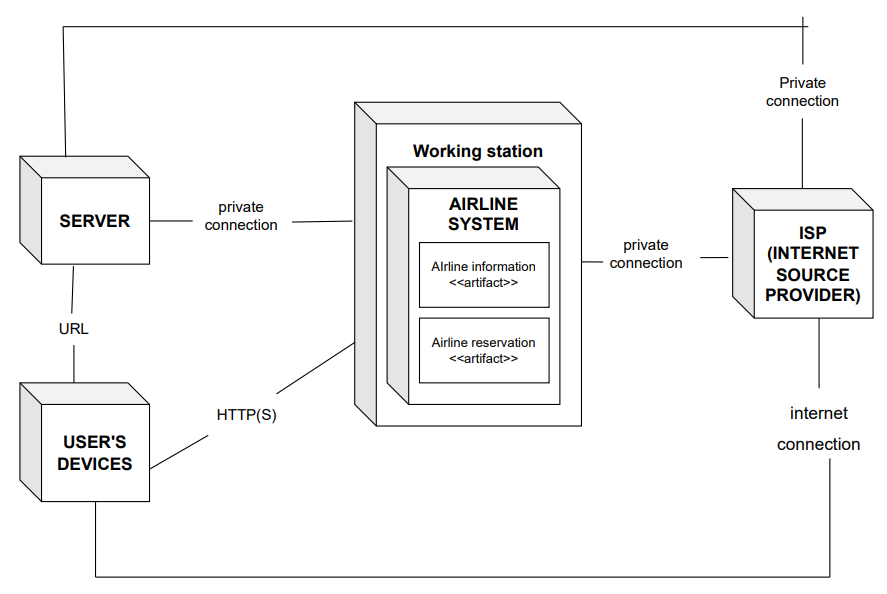


In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities.



A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

**DEPLOYMENT DIAGRAM**



The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships.

**Conclusion :**

***Therefore all the UML diagrams have been studied and implemented successfully.***

**References**

1) [https://www.startertutorials.com/uml/uml-diagrams-railway-reservation-s ystem.html](https://www.startertutorials.com/uml/uml-diagrams-railway-reservation-s%20ystem.html)

2) <https://app.diagrams.net/>

3) [https://itsourcecode.com/uml/deployment-diagram-for-railway-reservatio n-system/](https://itsourcecode.com/uml/deployment-diagram-for-railway-reservatio%20n-system/)

4) [https://www.freeprojectz.com/uml-diagram/railway-reservation-system-u ml-diagram/](https://www.freeprojectz.com/uml-diagram/railway-reservation-system-u%20ml-diagram/)