**1.Analysis**

**Requirement**

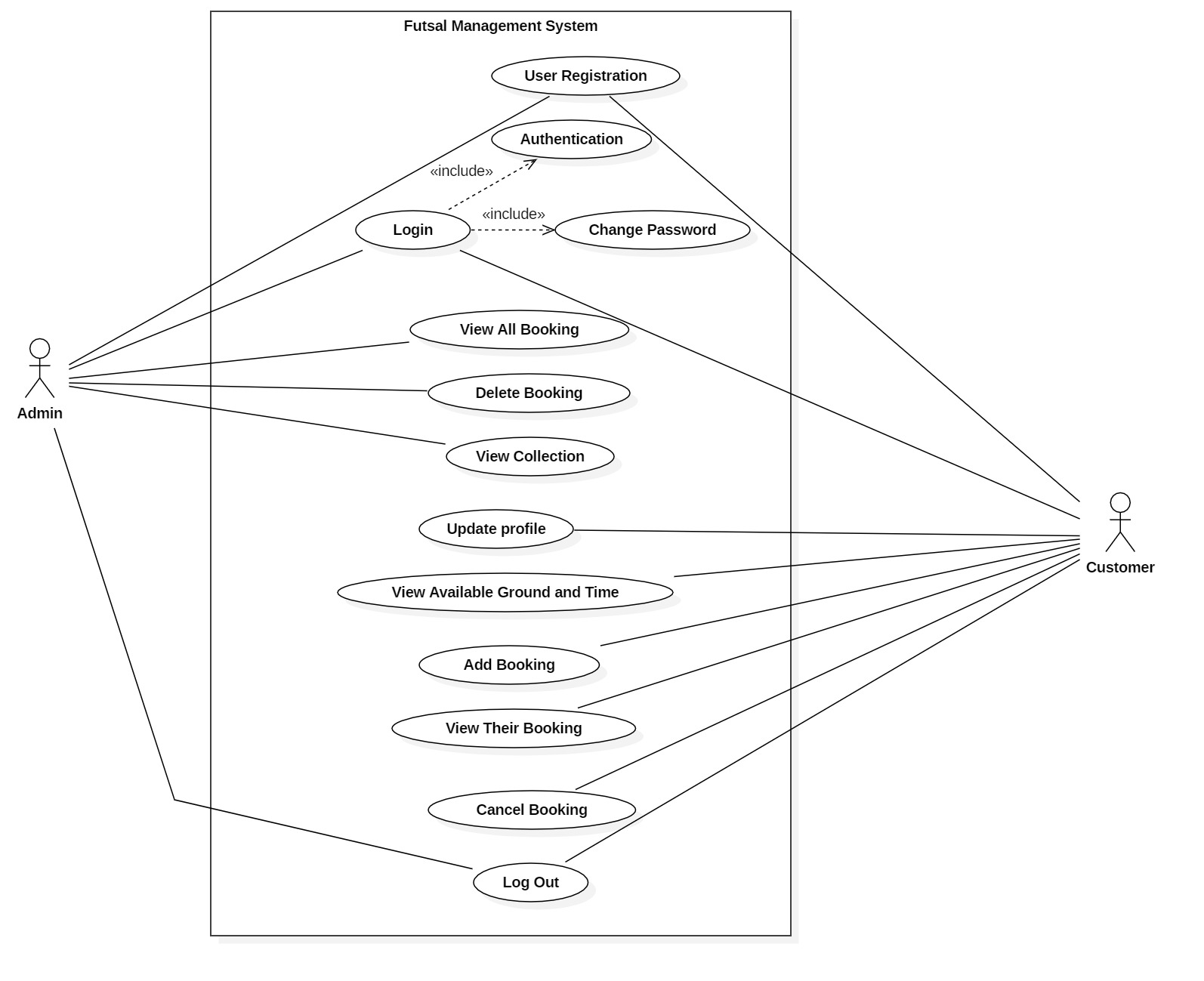
|  |  |  |
| --- | --- | --- |
| **Functional/Non-Functional(F/NF)** | **Requirements** | **MoSCoW** |
| F(R1) | Online Registration System | M |
| F(R2) | Login System | M |
| NF(R3) | Efficient Application | M |
| F(R4) | User friendly UI design | S |
| NF(R5) | Security | M |
| F(R6) | Provide information about timeslots (available/unavailable) for Reservation | M |
| F(R7) | Database System | M |
| F(R8) | Update Facilities for profile information’s and reservations | M |
| F(R9) | Allows user to remove reservations as per their needs | M |
| F(R10) | Prevent from reservation duplication | M |
| F(R11) | Billing System | S |
| F(R12) | Reward system for Customer that reach reservation milestones | C |
| NF(R13) | System Database should handle any amount of transaction data. i.e Scalability | C |
| NF(R14) | System should ensure Availability | S |
| NF(R15) | Reliable and Accurate processing | S |
|  |  |  |
|  |  |  |

**Use Case Diagram**

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users as known as **Actor** in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users.

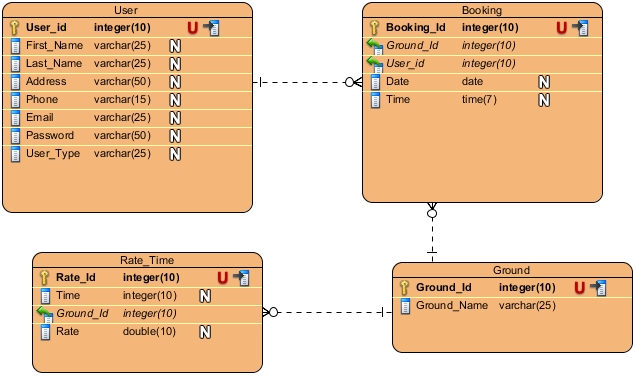
Characteristics of use case diagram are:

* Use case diagrams do not show interactions between actors.
* Actors are represented by stick figures.
* Actions are represented by ovals in which a broad description of the process is placed.
* A specific interaction is defined as a line which connects the actor and the action they can perform.



**ER-Diagram**

The Entity Relationship Diagram is simply a graphical way of representing data and their relationship with other data. Normally defines how entities like objects, peoples, concepts are related to each other. Three types of methods are present in order represent data using ER diagram, they are UML, Crows Feet and CHEN. I produced EG Diagram using Crows Feet method using software known as Oracle Visual Paradigm.



**Class Diagram (Initial)**

Class

A description of a group of objects all with similar roles in the system, which consists of:

* Structural features (attributes) define what objects of the class "know"
  + Represent the state of an object of the class
  + Are descriptions of the structural or static features of a class
* Behavioral features (operations) define what objects of the class "can do"
  + Define the way in which objects may interact
  + Operations are descriptions of behavioral or dynamic features of a class

Class Notation

A class notation consists of three parts:

1. Class Name
   * The name of the class appears in the first partition.
2. Class Attributes
   * Attributes are shown in the second partition.
   * The attribute type is shown after the colon.
   * Attributes map onto member variables (data members) in code.
3. Class Operations (Methods)
   * Operations are shown in the third partition. They are services the class provides.
   * The return type of a method is shown after the colon at the end of the method signature.
   * The return type of method parameters are shown after the colon following the parameter name.
   * Operations map onto class methods in code

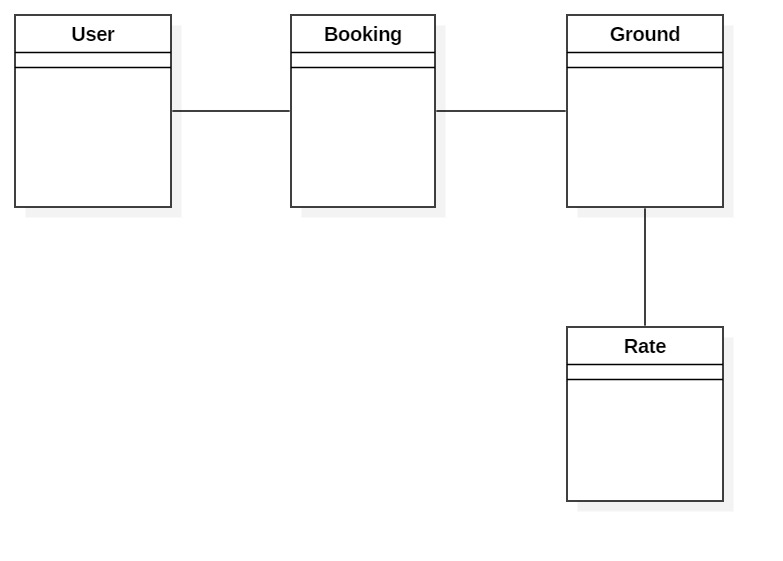
**Class Diagram**

In software engineering, a class diagram in the Unified Modelling Language (UML) is **a type of static structure diagram** that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Purpose of Class Diagrams

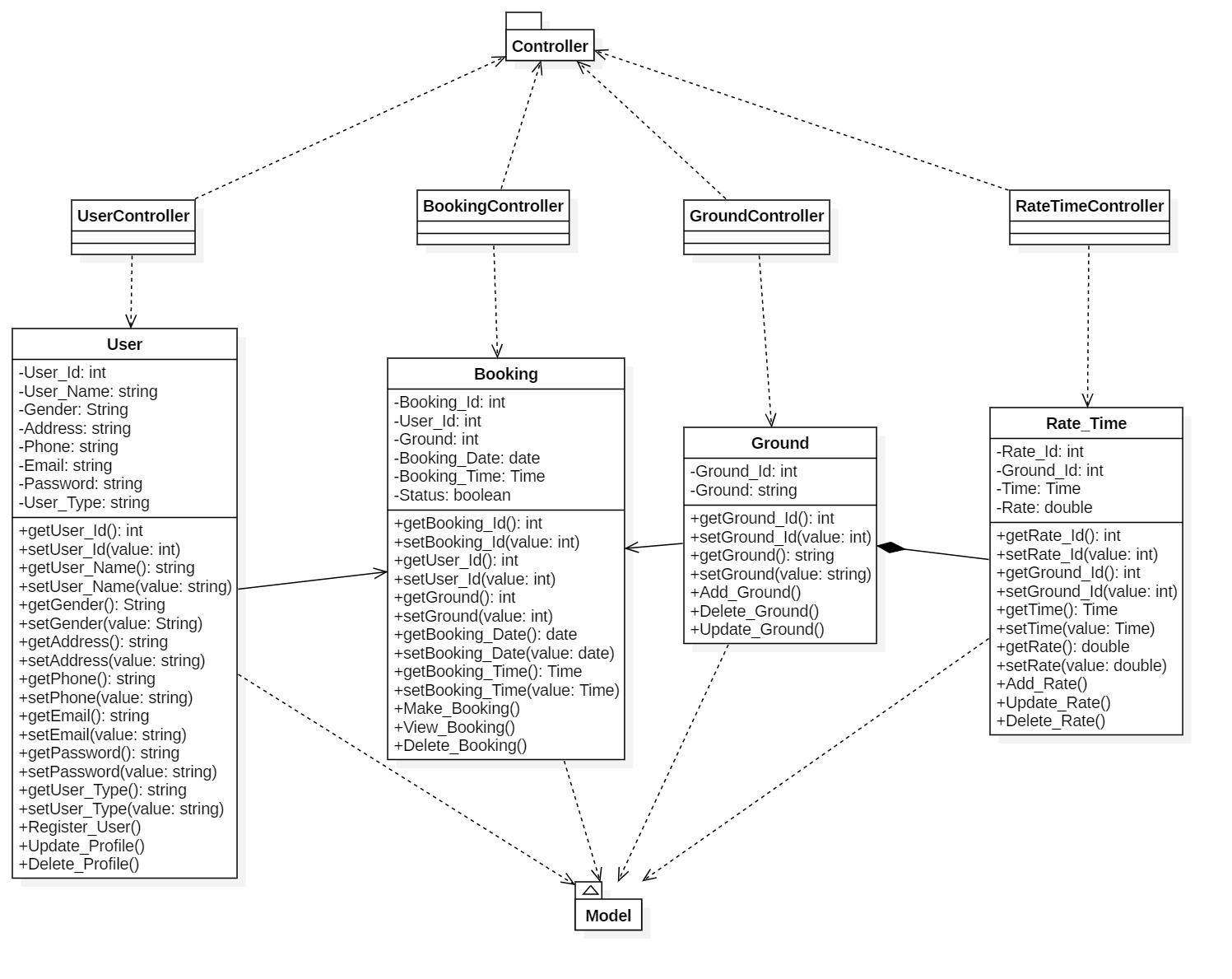
* Shows static structure of classifiers in a system
* Diagram provides basic notation for other structure diagrams prescribed by UML
* Helpful for developers and other team members too
* Business Analysts can use class diagrams to model systems from business perspective

**Initial Class Diagram**



**2. Design**

**1.1 Structural Model**



**1.2 Behavioral Model**

**Activity Diagram**

Activity diagrams are known as **workflow diagrams**. They are much like flow-charts, except more structured.

In Unified Modeling Language (UML), an activity diagram is a graphical representation of an executed set of procedural system activities and considered a state chart diagram variation. Activity diagrams describe parallel and conditional activities, use cases and system functions at a detailed level.

An activity diagram is used to model a large activity's sequential work flow by focusing on action sequences and respective action initiating conditions. The state of an activity relates to the performance of each workflow step.

Following are the Notational Elements used in Activity Diagram:

* **Swim Lane**

Used to indicate which actors or objects are responsible for the action. They are indicated by a series of lines partitioning the diagram.

* **Initial Node**

The starting point for the diagram. This is represented by a single filled circle.

* **Activity Final Node**

The termination point for the activities. There may be several of these in a diagram. This is a filled circle surrounded by a border.

* **Flow**

The flow represents the order in which activities are performed. Indicated by arrows.

* **Fork**

A fork indicates parallel processing – activities that can be undertaken at the same time. A fork is indicated by a thick bar where one flow enters and multiple flows leave.

* **Join**

A join indicates the end of parallel processing, and is indicated by a thick bar where multiple flows enter and only one leaves.

* **Decision**

A decision represent a choice that must be taken, and is represented as a diamond with a single flow entering and one or more flows leaving.

* **Activity**

An activity is the baseline step in an activity diagram, and is represented by a rounded oval. An activity is any logically discreet action that must be taken throughout the course of the activity.

**Some Activity Diagram associated with my project are:**

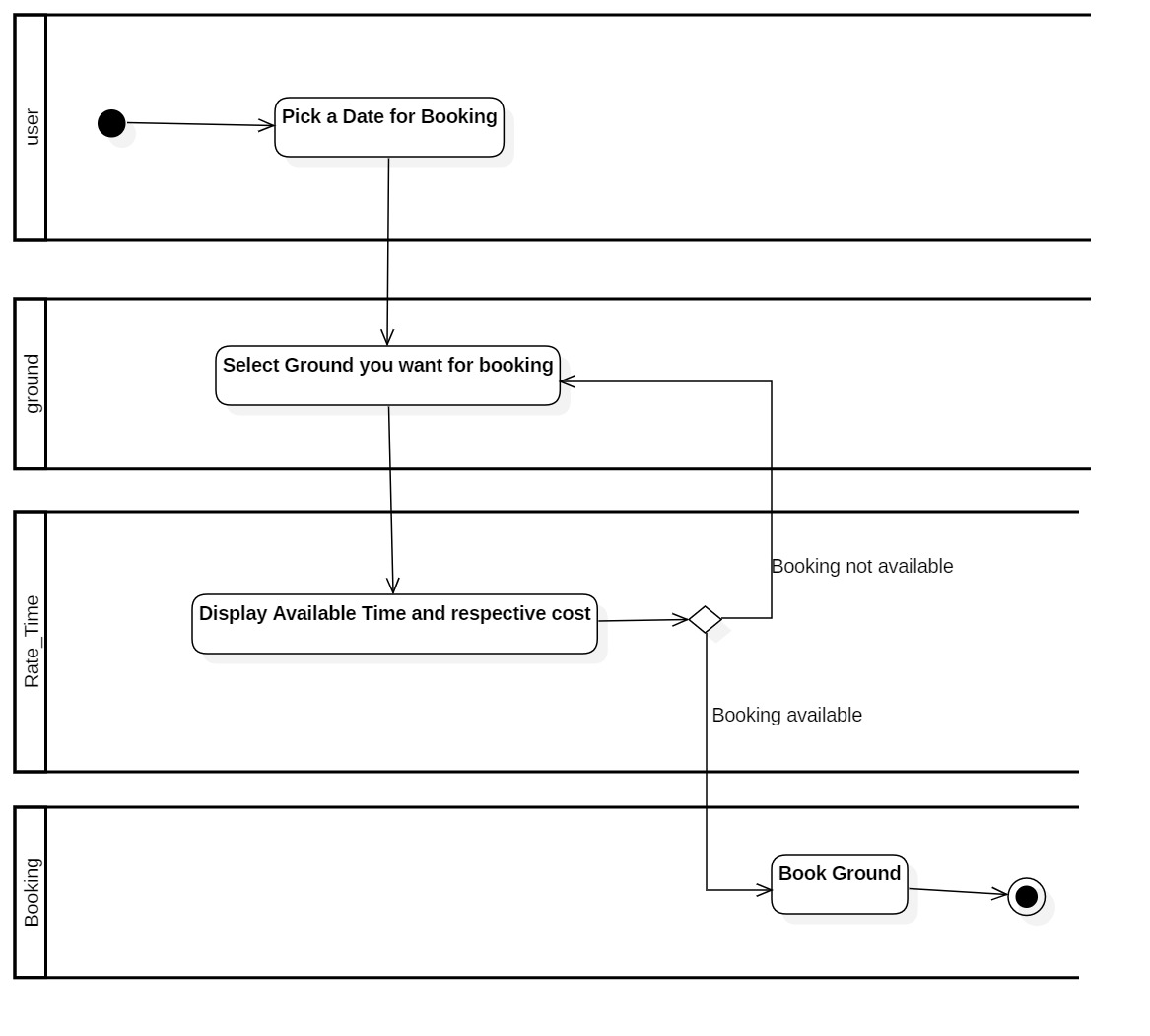
****

Figure : Viewing Available Timeslots

Activity

This activity explains about process that user perform in order to make their futsal booking. Firstly user gets log in and after that he/she pick a date for their booking and select ground that they want. After that system will display booking time slots for that ground. Now they will have a choice if they can make their booking or not. Finally, they can book a ground as their choice and end this activity.

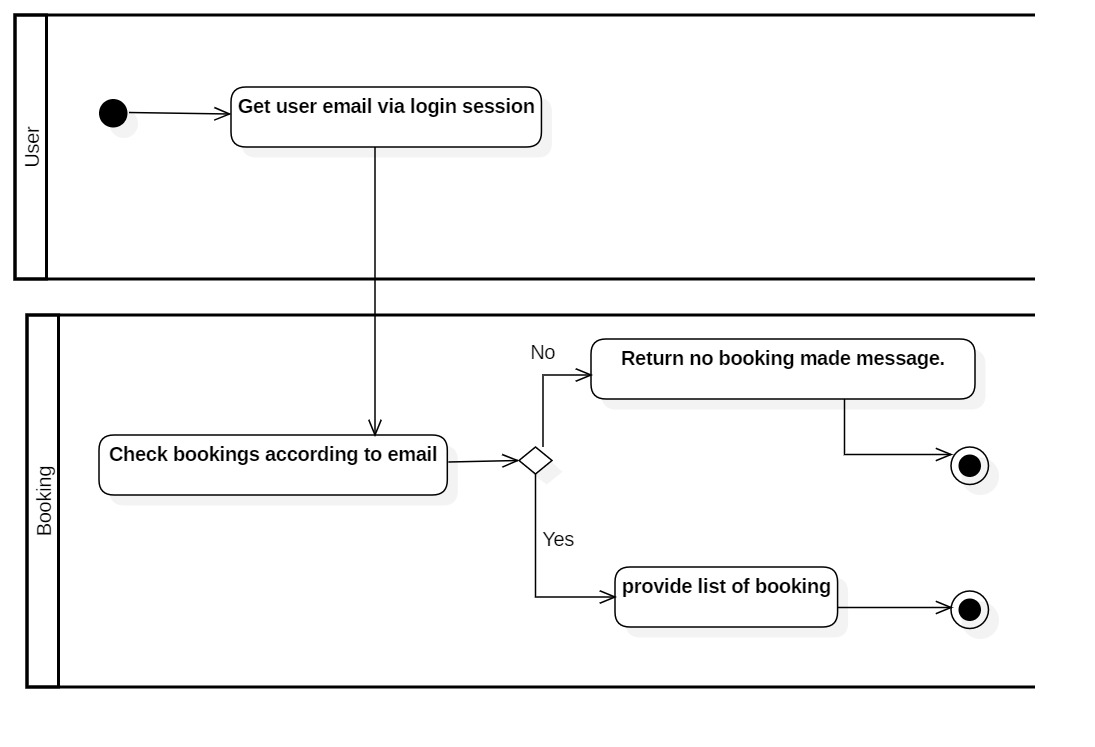
****

Figure : Viewing Booking Made

Activity

This activity explains about process that takes in place for displaying booking that client had been made.

Firstly, after log in system will get user email via session and when user clicks on page that display their booking, system will check every list of booking made via that email id. If no result is found with that id then it will show message saying they have no booking made and if it found results in booking list then it will display every booking made by them in respective web page.

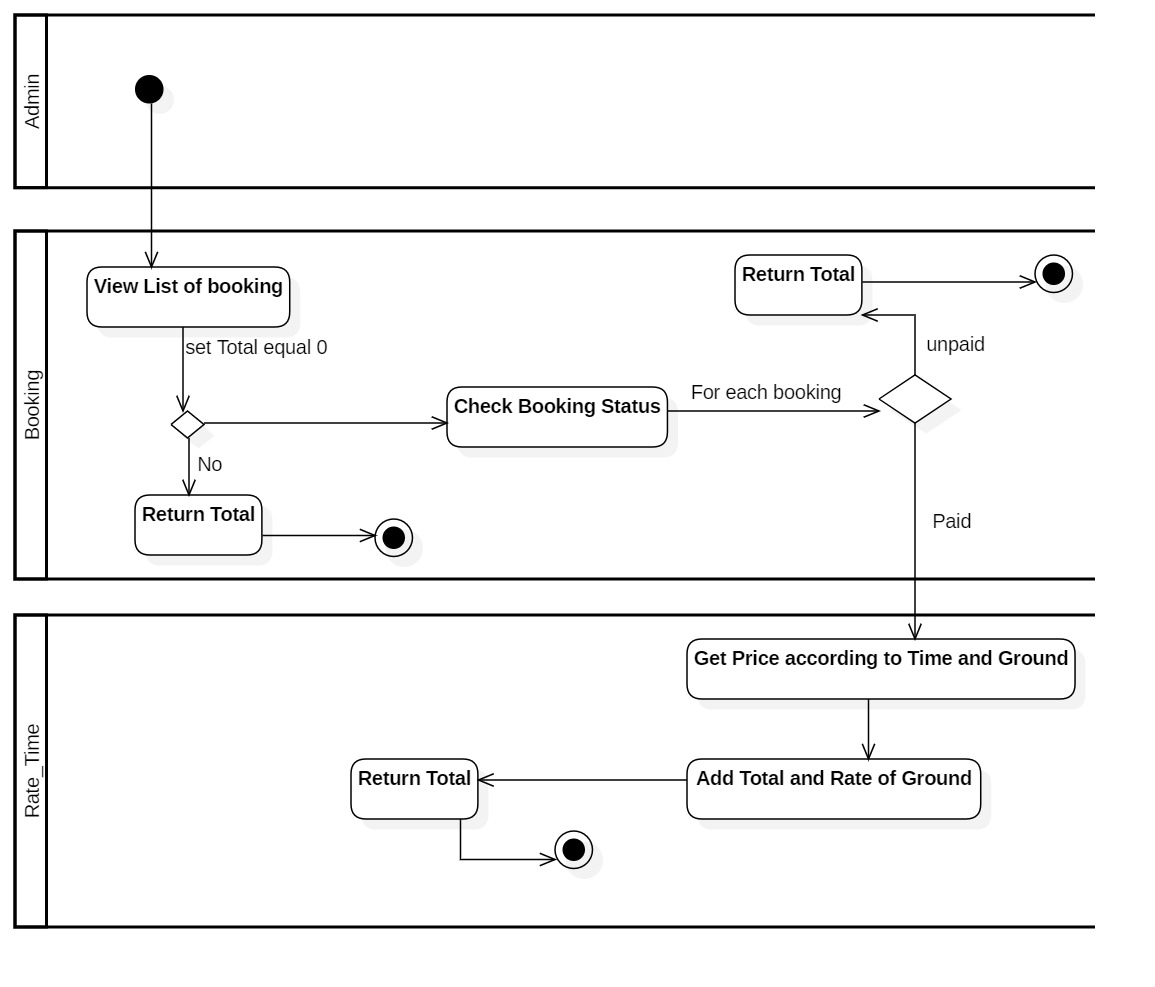
****

Figure : Calculating Paid Bookings

Activity

This activity shows that how admin can calculate total amount that they earned via booking. Here admin will list all bookings and after listing all booking. If booking were made then system will return 0 as total value and if booking exists then it will sum all prices of that booking whose status are paid if not paid then system will return 0 as total. Finally, total sum is returned as their earning.