**@RAYS**

Corporate Meeting Management System

BY:

Raushan Kumar Mahaseth

Date:03/25/2017

**@RAYS**

Table of Contents Page No

[1. Introduction 3](#_Toc291443302)

[1.1 Purpose 3](#_Toc291443303)

[2. Requirements - Use-Case – Usage Scenarios 4](#_Toc291443304)

[3. Design of System 5](#_Toc291443305)

[4. Detailed Design 6-8](#_Toc291443306)

[4.1 <Object.Name> Usage 6](#_Toc291443307)

[4.2 Interaction Diagram 7-8](#_Toc291443308)

[5. Issues, Risk and Dependencies 9](#_Toc291443309)

[6. Design Alternatives 10](#_Toc291443310)

7.FutureConsiderations….…………………………………………………....…...….10

[8. References 10](#_Toc291443311)

# Introduction

|  |
| --- |
| This is an Online Corporate Meeting management system project that serves the functionality of reserving a hall for organizing meetings. The system allows only registered users to login and new users canbe added by the admin on the application. This is a web application developed in Java Technology implementing Spring MVC Framework. The project provides most of the basic functionality required for a meeting. Once the user is logged into the system it allows the user to select available datesto reserve the hall. Then the system will allow the user to choose the hall from the list of available halls.User is then directed to the detail page. This page provides detail information on the hall name, hall capacity, base cost of hall, venue where hall is located, and venue address. User is now directed to the reservation page. This page provides information on the total reservation cost (including facility and tax amount). After user confirms reservation, user is directed to the payment page where payment is made. Once reservation is done, user can track history of their reservations and details related to those reservations. After reservation is done, all this data is logged in the database and the user is given a receipt number for his /her booking. This data is then sent to the administrator (website owner) and they may interact with the customer as per customer’srequirement and contact data stored in the database. |
|  |

# Purpose

It is important to identify the [target audience](https://en.wikipedia.org/wiki/Target_audience), devise the meeting concept, and coordinate the technical aspects before launching the meeting.**Corporate Meeting management** is the application of [project management](https://en.wikipedia.org/wiki/Project_management) to the creation and development of large scale events such as conference, seminar, meeting, convention and many more. It involves studying the brand, identifying the [target audience](https://en.wikipedia.org/wiki/Target_audience), devising the event concept, and coordinating the technical aspects before actually launching the event.

**Event planning**  include budgeting, scheduling, site selection, acquiring necessary [permits](https://en.wikipedia.org/wiki/License), coordinating transportation and parking, arranging for speakers or entertainers, arranging decor, event security, [catering](https://en.wikipedia.org/wiki/Catering), coordinating with third party vendors, and emergency plans.

This application helps people to save their valuable time, money and make their life easy. It helps to identify desired venue and make online reservationspromptly.

Corporate people will use this application.

# Requirements - Use-Case – Usage Scenarios

1. The application helps client to save money. It calculates exact cost for the event. Customer can therefore make reservation based on their budget.
2. The application helps to save valuable time of the customer. Customer will have access to all the resource at a single place.
3. The application has two use case: 1) Admin, 2) User

Admin can add customer and view added customer. User can make new reservation and view their reservation history.

1. **Design of System**

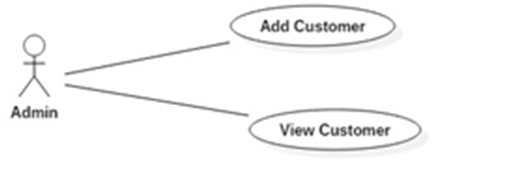


Figure1.a: Admin use case

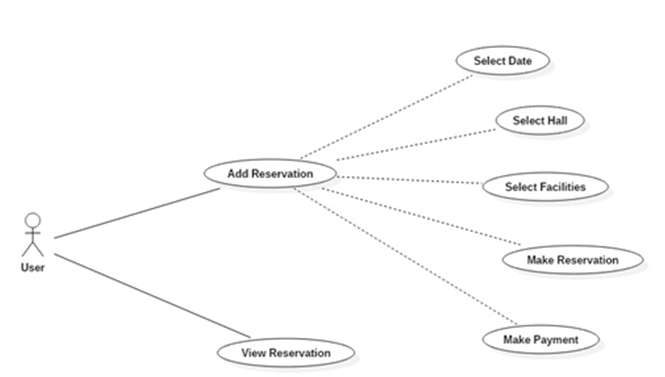


Figure1.b: User use case

**Explanation:**

There are two types of user: Admin and Customer.

Admin has authority to addcustomer and view customer Information. Customercan make new reservation and view details on the reservation.

# Detailed Design

## 4.1****<Object.Name> Usage****

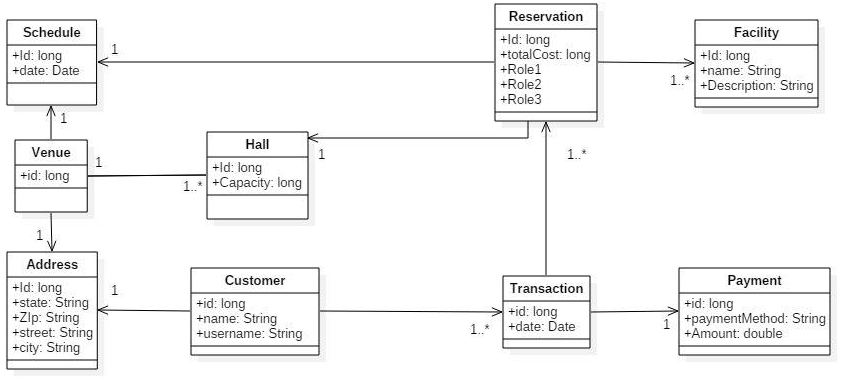


Figure 2: Class Diagram

**Explanation:**

The above class diagram has nine entity. These entities interact with each other via one to one unidirectional relationship, one to many unidirectional/bidirectional relationship. One customer has multiple transactions and one address. One transaction at a time has multiple reservation and one payment. One reservation has multiple facilities, one hall and one schedule. Venue has multiple halls, one schedule and one address. One hall has one venue.

* 1. **Interaction Diagram**

## ****C:\Users\chetan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\sequence5.jpg****

Figure 1.a: New reservation by customer

**Explanation:**

When the user logins to the system, the login controller handles the user information(username and password)to authenticate and provide access to the requested form on the basis of the user role. After successful login, the user is directed to web flow and start’sto make newreservation. Reservation helper component validates customer, sets schedule, sets hall,sets facilities, makes reservation and makes payment. Reservation service saves new reservation from reservation repository.

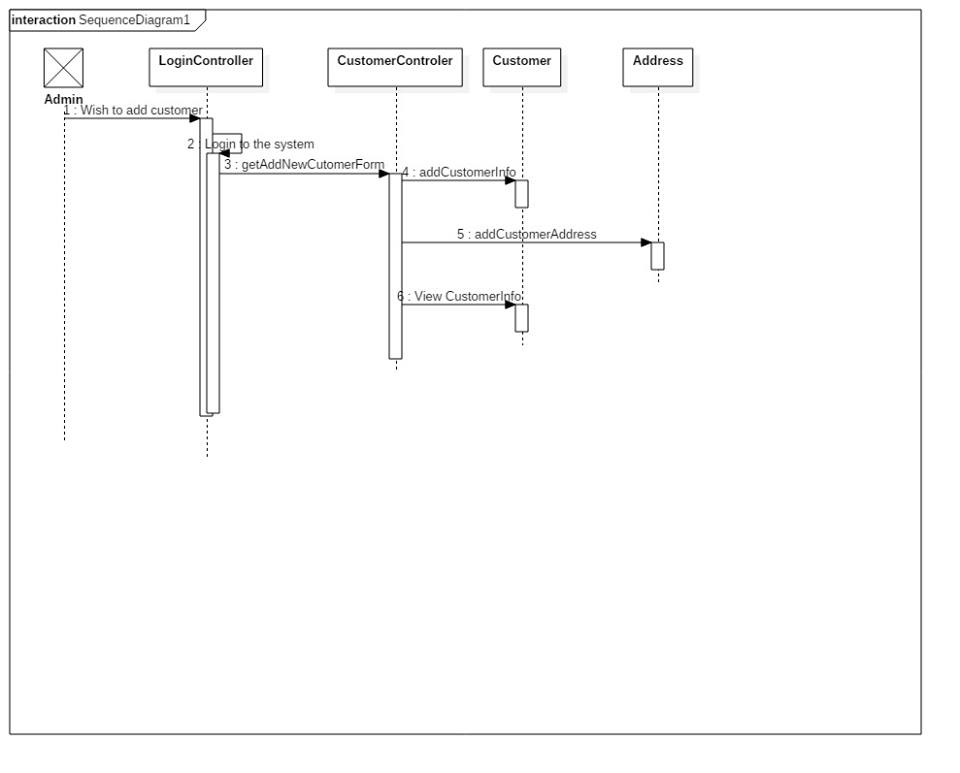


Figure 1.b: Add customer by admin

**Explanation:**

When the user logins to the system, the login controller handles the user information (username and password) to authenticateand provide access to the requested form on the basis of the user role. After a successful login, admin gets addNewCustomer form from customer controller. Customer controller then adds customer information and customer address. Finally, admin can view added customer information from customer controller.

# Issues, Risk and Dependencies

* 1. **Issues:**

We had following issues while developing our application:

* + 1. Spring Web flow Issue:
* Web flow needed extra configuration in order to work with tiles.
* Model binding with validation needs separate variables for segregated validation of complex properties. But this was not implemented as required.
* All sub properties of model attributes declared in flow definition file had to be initialized. This was done using user validation helper component.
  + 1. **Security Issue:**
* Exception thrown by security was difficult to handle especially while using volunteer interceptor.
  1. **Risk:**
     1. No unit testing is done.Only partial integration testing is done.

# Design Alternatives

We had problem in spring web flow design at first (in version1 of our application). Spring web flow design did not work. As an alternative, we designed custom web flow by the implementation of controller (version2). At the end, we fixed issues in spring web flow and switched our application to version 1.

# Future Considerations

Future considerations that could be added to this application

1. Customer sign up
2. Facilities addition.
3. Payment gateway management.

# References

1. Spring MVC beginner’s guide.
2. <http://projects.spring.io/spring-webflow/>