SMART LOGICS

Software Architecture Document

Version <1.0>

Revision History

Date	Version	Description	Author
11/April/16	1.0	Software Architecture Document	Saranga Ekanayaka

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the ReserveMe. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

The Software Architecture Document describes the application architecture associated with the ReserveMe. This document contain the reasons for using architecture and the architectural constraints that applies to the system. To describe the overall architecture, following aspects of architecture are used.

1.3 Overview

Rest of this documentation contain what architectural designs are used to describe each of the Use-Case, Logical, Process, Deployment, and Implementation Views. And what kind of models contain those.

- Use-case view – This is a diagram which provide knowledge about functionalities of each users of the system.
- Logical View –. This describes the overall logical architecture of the application. Decomposition of the system into layers, and the communication between them are described here.
- Process view – The sequence of activities that is carried out by the system in order to achieve the required functions are described in this section.
- Deployment view –Physical configuration details are described here.
- Implementation view – The overall structure of sub packages and their dependencies are described using package diagrams.

This also contain size and performance and quality of the system.

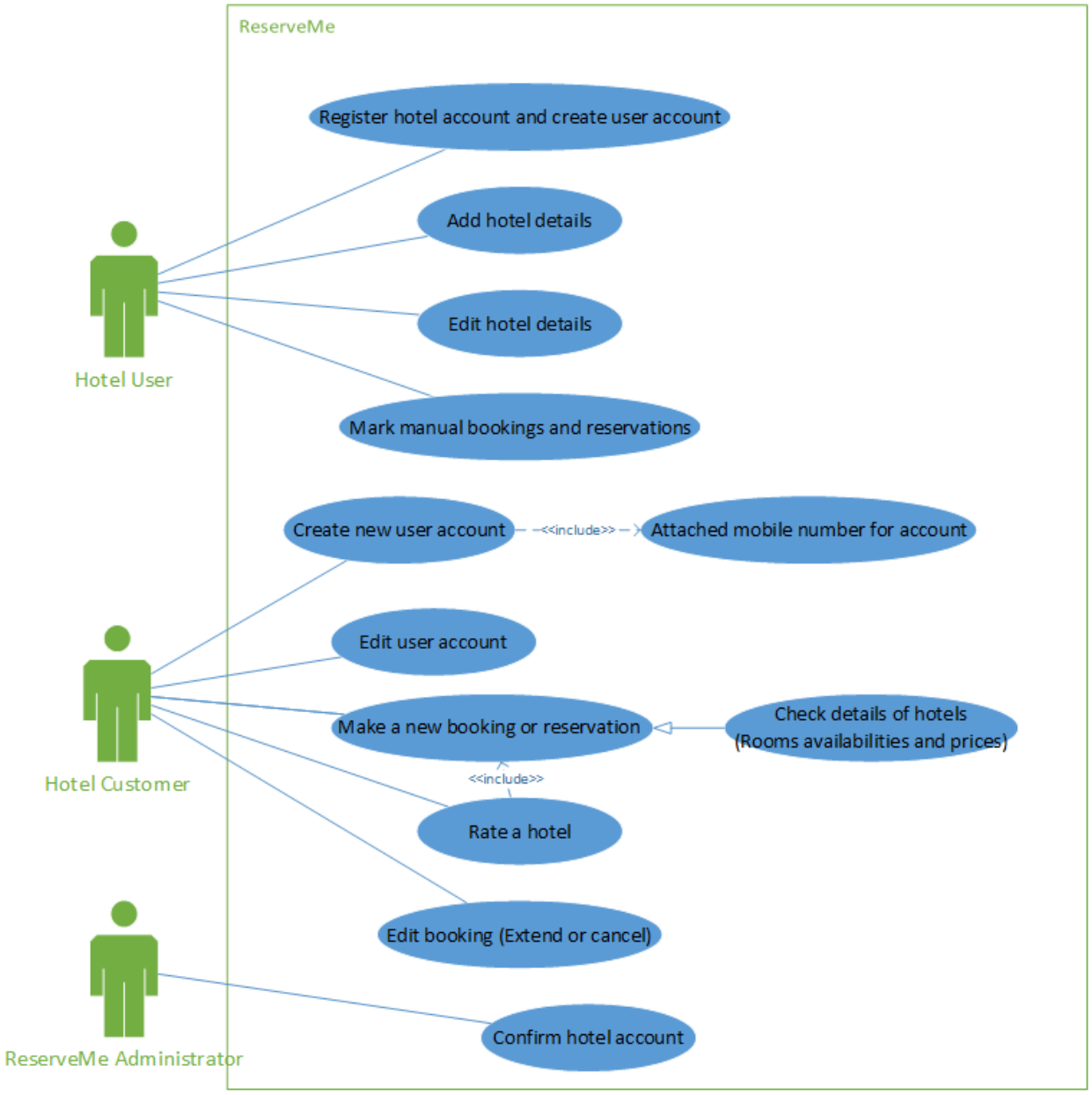
- Size and performance - A description of the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints.
- Quality - A description of nonfunctional requirements which are increases the quality of the system.

2. Architectural Goals and Constraints

This application uses three-tier architecture for its design. The architecture has three layers, a user interface layer (view layer) on top, a business logic layer in the middle, and a data storage layer at the bottom. Use case design, Logical view, and process views can be used to represent the application communicates in the three layers.

3. Use-Case View

3.1 Use-Case Realizations



Add hotel details, Edit hotel details, Make a booking and rate hotel service are the most important use cases of this system.

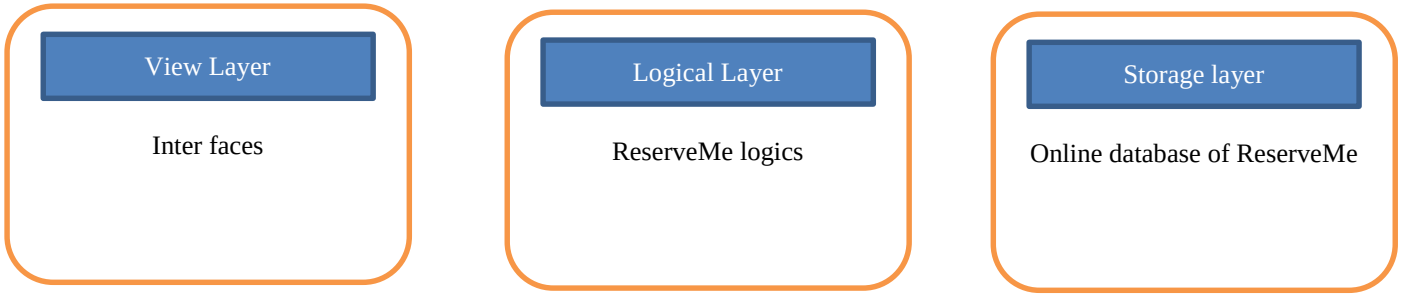
Use case name	Add hotel details/ Edit hotel details
Actor	Hotel user
Description	Add details of hotel for check customers when before make a booking. In hear they mark position of the hotel, Number of rooms they have. Capacity, prices, availability of each room
Preconditions	User must log their hotel account first
Main flow	After loge hotel account they can open edit detail tab and they can edit or add details via it
Successful end/post condition	Applies changes
Fail end/post condition	
Extensions	

Use case name	Make a new booking or reservation
Actor	Hotel customer
Description	Main function of this system is allows customers to find accommodation. So hotel customers can make booking via this system.
Preconditions	User must log their hotel account first. Rooms must be available at that time.
Main flow	After loge hotel account they can booking available rooms. If customer want they can check details of each hotels and ratings of them.
Successful end/post condition	Mark booked room as reserved
Fail end/post condition	No any booking
Extensions	

Use case name	Rate hotels
Actor	Hotel customer
Description	Rating will help to get a knowledge about service of hotels for customers before make a booking. Customers can rate the service of hotel they booked.
Preconditions	User must log their hotel account first. Foe rate a hotel they must have to make a booking of that hotel.
Main flow	If customer has made a booking they can rate the service of that hotel.
Successful end/post condition	Update the rating state of particular hotel
Fail end/post condition	
Extensions	

4. Logical View

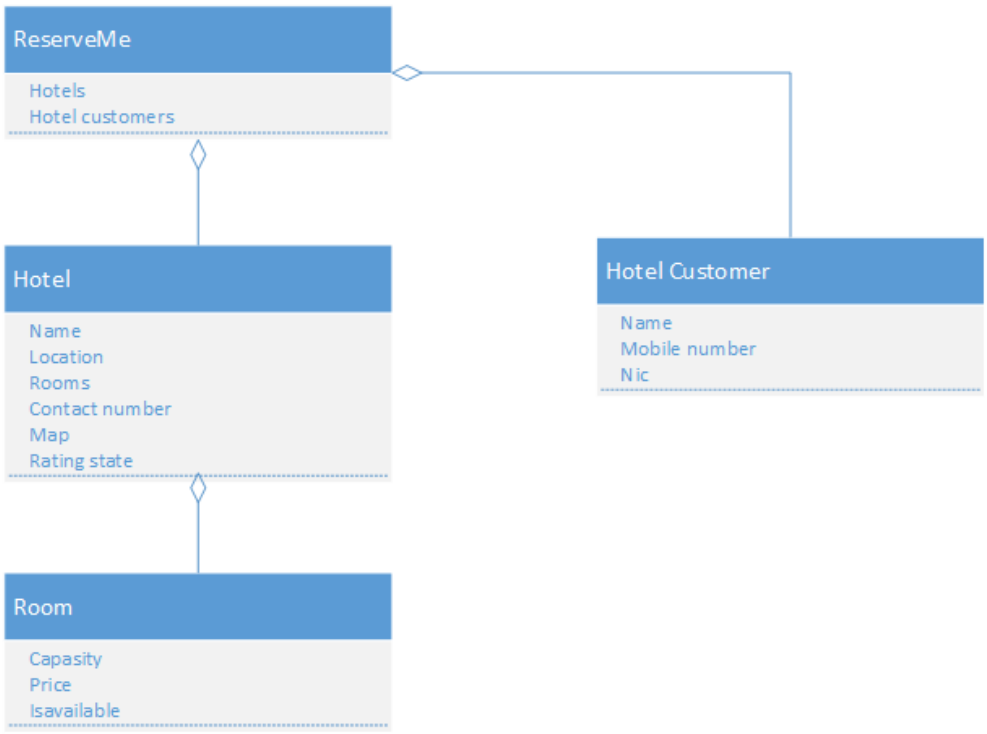
4.1 Overview



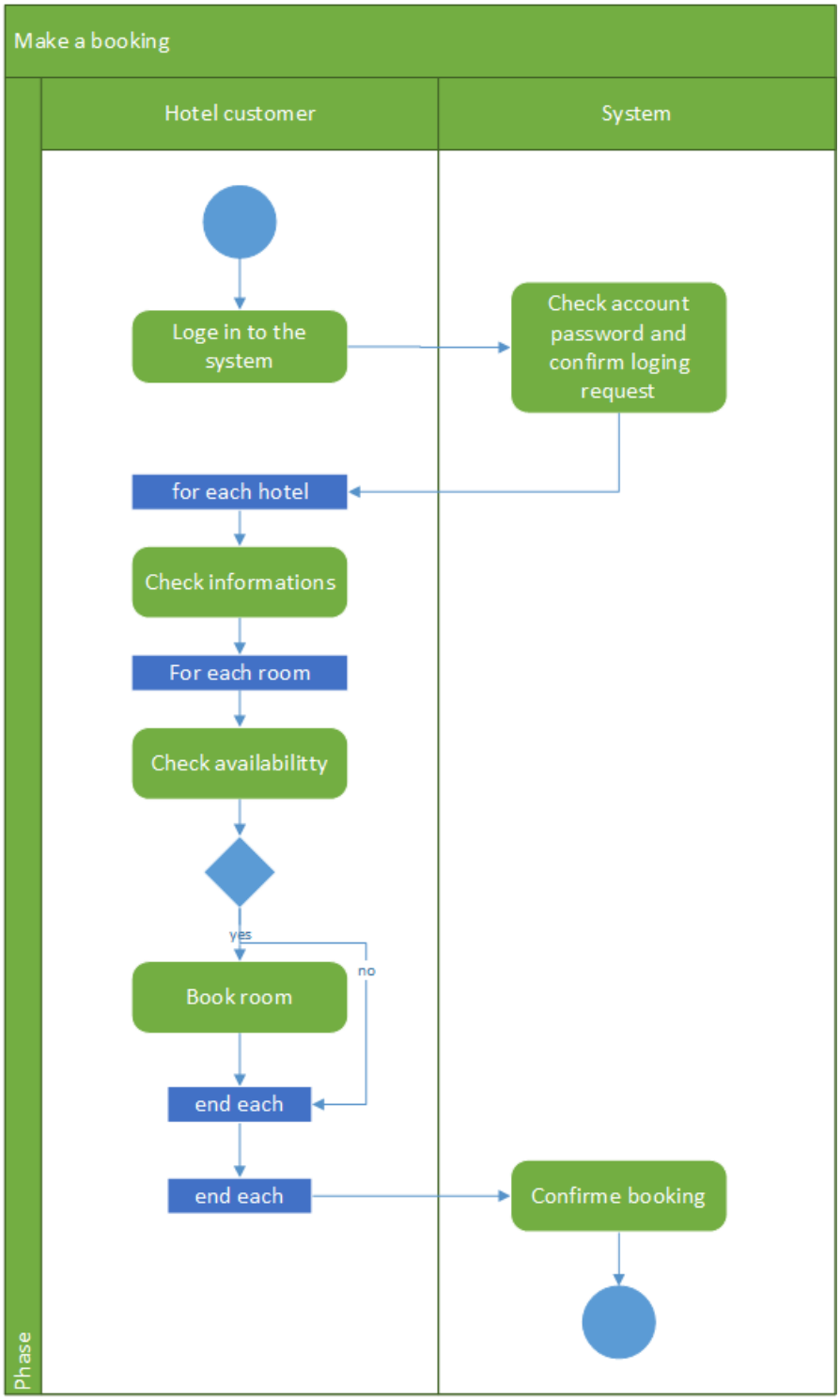
In an overview the architecture can be presented in three layers as View layer, logical layer, Storage layer. The View layer provides the functionality required by using the application by user. All the graphical interfaces are relate to that layer. The logical layer is used to decouple the user from the data storage and thus achieving security and privacy of data. The storage layer provides the data related functionality and methods of retrieving the data from data storage.

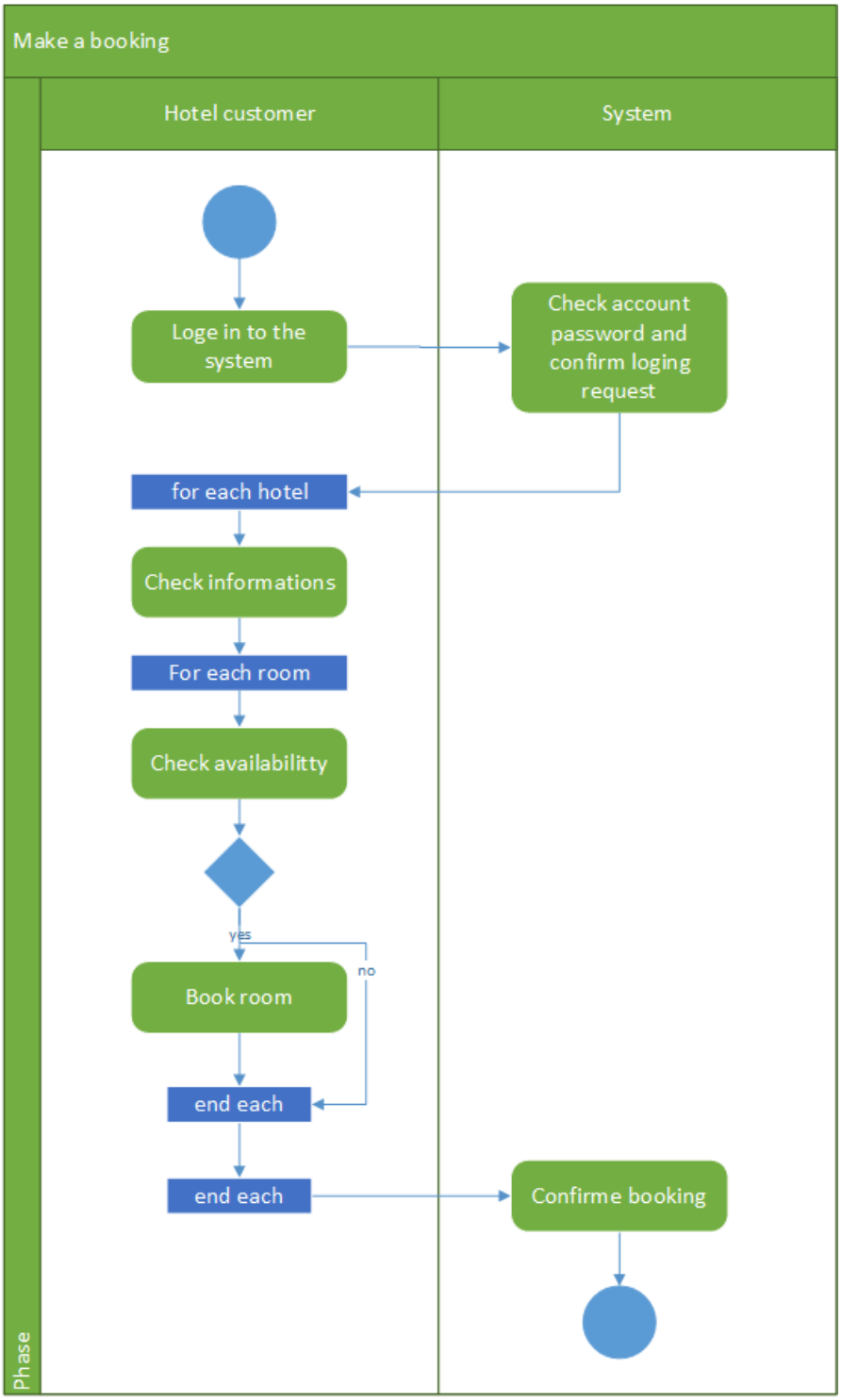
4.2 Architecturally Significant Design Packages

There are three classes in this system. ReserveMe has list of hotel and hotel customers and each room has rooms. Those are describe in class diagram.



5. Process View




Make a booking
Hotel customer
System
Loge in to the system
Check account password and confirm loging request
for each hotel
Check informations
For each room
Check availabilitty
yes
no
Book room
end each
end each
Confirme booking
Phase

6. Deployment View

7. Implementation View

7.1 Overview

.

7.2 Layers

Include the Package diagram and describe

8. Data View (optional)

The data are stored into online database.

9. Size and Performance

The major function of the product is allows travelers to find accommodations easily. Thus some processing of data is required. Therefore the product is expected to perform efficiently.

10. Quality

A user cannot directly communicate with the database. The layers are independent and uses well defined interface for inter communication. The layering makes it easy to achieve important features such as portability and reusability of the system.