<Hotel Reservation System>

System Design

<1.0>

<11.12.2016>

< Çağla Çınar

Berk Ergun

Serdar Ünlüsoy

Öner Ercan >

Prepared for

SE301 Software Engineering



<Hotel Reservation System>

Table of Contents

[1. Introduction](#_Toc433996772) **[1.](#_Toc433996772)**

[1.1. Purpose of the System](#_Toc433996773) **[1.](#_Toc433996773)**

[1.2. Design Goals](#_Toc433996774) **[1.](#_Toc433996774)**

[1.3. Definitions, Acronyms, and Abbreviations](#_Toc433996775) **[1.](#_Toc433996775)**

[1.4. References](#_Toc433996776) **[1.](#_Toc433996776)**

[2. Current Software Architecture](#_Toc433996777) **[1.](#_Toc433996777)**

[3. Proposed Software Architecture](#_Toc433996778) **[2.](#_Toc433996778)**

[3.1. Overview](#_Toc433996779) **[2.](#_Toc433996779)**

[3.2. System Decomposition 3](#_Toc433996780)**[.](#_Toc433996780)**

[3.3. Hardware Software Mapping](#_Toc433996781) **[4.](#_Toc433996781)**

[3.4. Persistent Data Management](#_Toc433996782) **[4.](#_Toc433996782)**

[3.5. Access Control and Security](#_Toc433996783) **[4.](#_Toc433996783)**

[3.6. Global Software Control](#_Toc433996784) **[5.](#_Toc433996784)**

[3.7. Boundary Conditions](#_Toc433996785) **[5.](#_Toc433996785)**

[4. Subsystem Services](#_Toc433996786) **[5.](#_Toc433996786)**

[5. References](#_Toc433996787) **[5.](#_Toc433996787)**

<Hotel Reservation System>

**SYSTEM DESIGN DOCUMENT [1]**

**1. Introduction**

**1.1. Propose of the System**

Online Hotel Reservation System has been named as "outing" by us in our project. Outing is capable of providing search in desired criteria to user and capable of letting making reservation in desired hotel to user.

**1.2. Design Goals**

Outing is planned to provide an advanced search and list them according to desired cities, dates, number of people who will travel and number of room and provide making fast and easy reservation by their desires.

The design goals of this system are; correctness, robustness and efficiency. We will provide the correctness of our system by satisfying the requirements we have mentioned before.

People will be able to do their advance searches and their reservation operations fast and easily via Outing. For instance, people will be able to do an advanced search by desired cities, dates, number of people and number of room, as a result of this advanced search, the best-fit hotels will be able to be listed and reservation will be able to be made according to preferred hotel.

**1.3. Definitions, Acronyms and Abreviations**

HRS : Hotel Reservation System

**1.4. References**

**-** Requirement Analysis Document (20.11.2016)

- booking.com

**2. Current Software Architecture**

The project is being developed for the newly created company, X as it's hotel reservation channel with users and has no predecessor. As there is no 'current system' working, it'd be wise to instead look at our available architecture options and explain how we have decided on our architecture.

The project has a couple of options available as it's architecture. A Model-view-controller approach could have been taken but as the project will be a website, it has been discarded as it's a more suited architecture for GUI applications and web applications. A client/server would be a fitting choice; where users use client websites through their choice of web browser to connect to the server. But future prospects of updates and function additions onto the system's capabilities make a different but similar architecture more attractive.

<Hotel Reservation System>

An n-tiered architecture would make the best choice for our hotel reservation system project; with an interface layer where users can interact the system through their web browsers, a logic layer where all the processing and communication is done and a storage layer where all the

records for all the users,hotels and reservations are kept. The seperation of logic and storage layers allow for more flexibility in making improvements to each and also provides reliablity for the system.

**3. Proposed Software Architecture**

**3.1. Overview**

We are building our system "outing" by using Three Tier architectural style. Three tier architectural system consists of 3 layers which are Presentation,Function and Storage respectively from higher to lower.

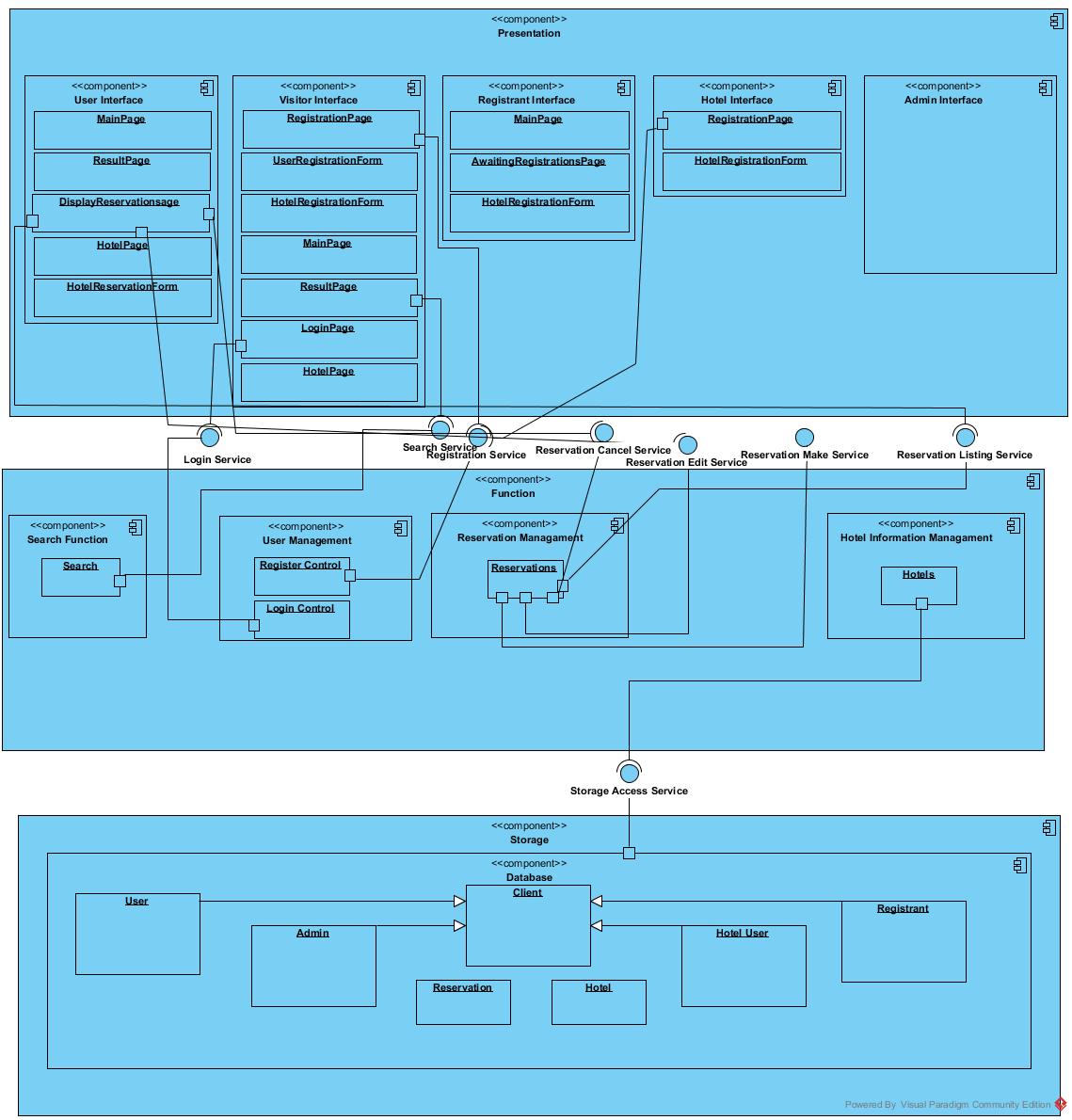
- Presentation layer is the interface for client.It contains forms and pages which are at the top level.

- Function layer is at middle.It manipulates the data which comes from database.

- Storage layer is the lowest layer and contains the database.

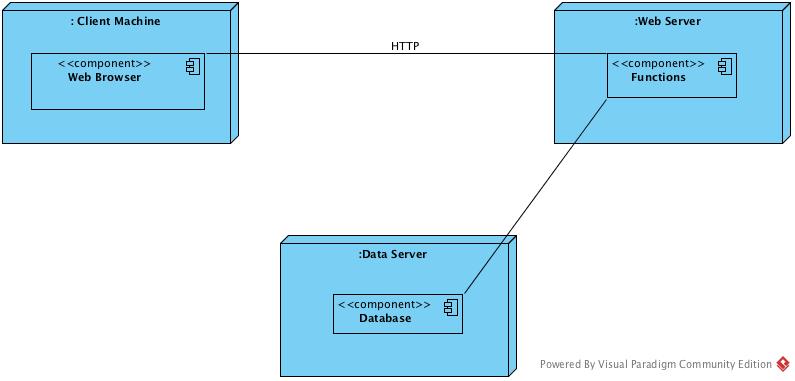
<Hotel Reservation System>

**3.2. System Decomposition**



<Hotel Reservation System>

**3.3. Hardware Software Mapping**



Client uses a machine to connect our server with HTTP protocol via a web browser on the machine. The data server gathers the data from database to webserver. Web browser performs functions and manipulates the data. All of the interactions with client performed on webrowser.

**3.4. Persistent Data Managment**

Our system uses relational database.

**3.5. Access Control and Security**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Users** | **Hotels** | **Reservations** |
| **User** |  | pickHotel()  search() | <<create>>  makeReservation()  editReservations() |
| **Hotel User** |  | initiateHotelRegistration() | viewReservations() |
| **Registrant** |  | <<create>>  displayForm()  approveForm()  rejectForm() |  |
| **Admin** | <<create>>  removeUser() | removeHotel() |  |

**3.6. Global Software Control**

This issue is handled by MySQL protocols.

<Hotel Reservation System>

**3.7. Boundary Conditions**

When system first initiated a registration page comes up and the first user signs up as Admin. Thus system should not be initiated publicly at first. Then Admin adds registrant by hand. After these steps system starts functioning normally which means main page starts

<Hotel Reservation System>

being displayed at first. Since the system is a webpage, it's available 24/7. System is only shutted down when a maintanence is needed.While maintanance is on going system displays a related message and cant be used. Since all the data is saved on a web server in case of a failure the protocol's of the web server are applied.

**4. Subsystem Services**

**5. References**

1 - Bruegge B. & Dutoit A.H.. (2010). Object-Oriented Software Engineering Using UML, Patterns, and Java, Prentice Hall, 3rd ed.

2 - Instructor's Course Slides