**HOTEL RESERVATION**

**SYSTEM**

**System Design**

**Document**

15.07.2016

P.MELİS DELİPINAR

YAĞIZ ŞENER

ONUR CİMŞİT

ORHUN BÜYÜKAYDIN

**Prepared for**

**SE301 Software Engineering**

Table of Contents

[1. Introduction](#_Toc433996772) 3

[1.1. Purpose of the System](#_Toc433996773) 3

[1.2. Design Goals](#_Toc433996774) 3

[1.3. Definitions, Acronyms, and Abbreviations](#_Toc433996775) 3

[1.4. References](#_Toc433996776) 4

[2. Current Software Architecture](#_Toc433996777) 4

[3. Proposed Software Architecture](#_Toc433996778) 5

[3.1. Overview](#_Toc433996779) 5

[3.2. System Decomposition](#_Toc433996780) 6

[3.3. Hardware Software Mapping](#_Toc433996781) 9

[3.4. Persistent Data Management](#_Toc433996782) 9

[3.5. Access Control and Security](#_Toc433996783) 10

[3.6. Global Software Control](#_Toc433996784) 12

[3.7. Boundary Conditions](#_Toc433996785) 13

[4. Subsystem Services](#_Toc433996786) 13

**1. Introduction**

The following section provides an overview of the derived System Design Document for the project “Hotel Reservation System”. Purpose of the system, design goals, definitions, acronyms and abbreviations are specified below. This document also includes current and proposed software architectures, subsystem services and references.

**1.1 Purpose of the system**

The main purpose of this system is to provide a quality service to users. The users who are looking for a hotel should make reservations with just a few clicks. The system has some useful features that can help users to make reservations quite easily. Users can search and reserve a hotel room or cancel his/her reservation over the internet at any time. The system is completely free and the registration is only needed for making a reservation. Another feature included is to give users a choice to compare hotels with their ratings, which are given by the previous guests. Not just for users, hotels can also register their hotels to our system for advertising purposes.

**1.2 Design Goals**

The system is a web-based application that stimulates the users to make reservations to hotels. In this system there are three sides. First one is user side, second one is hotel side and finally the admin side. Users can use the system for free. S/he can search for hotels, look for their reviews and make a price research. Hotel users can register their hotels with their information and control the bookings made by users. When we look at the admin side, s/he can make some changes in the system according to the system rules.

All these parts are designed to be user-friendly. Our system has a simple and intuitive interface. A particular user of the system has no difficulty in reading the text on the display. Menu choices are presented in form of buttons, which contains texts, as well as little images illustrating the choices for better understanding.

**1.3 Definitions, Acronyms and Abbreviations**

RAD : Requirement Analysis Design

SDD : System Design Document

HTTP : The HyperText Transfer Protocol

UI : User Interface

JSP : Java Server Pages

**1.4 References**

[**http://www.javatpoint.com/uploading-file-to-the-server-in-jsp**](http://www.javatpoint.com/uploading-file-to-the-server-in-jsp)

[**http://stackoverflow.com/questions/3804591/efficient-method-to-generate-uuid-string-in-java-uuid-randomuuid-tostring-w**](http://stackoverflow.com/questions/3804591/efficient-method-to-generate-uuid-string-in-java-uuid-randomuuid-tostring-w)

[**http://stackoverflow.com/questions/15105322/sending-additional-data-with-multipart**](http://stackoverflow.com/questions/15105322/sending-additional-data-with-multipart)

**2. Current Software Architecture**

Web and mobile based reservation systems are very popular nowadays. In world and also in Turkey, “Booking.com” is one of the most popular and largest reservation system. With this kind of reservation systems, the user selects the city that where s/he wants to make his/her vacation and selects the best hotel for their purpose or their budget.

In our system, a user can search for hotels for free without signing in, but when a user selects a hotel to make a reservation, the system makes sure that the user is registered to the system. If not, the user fills the needed forms to register the system. Signing up is needed only for making a reservation. After the registration, user can make his/her reservation easily and s/he can make the payment with the system. Also, users can check their own past/future reservations with their information and any reservation made by user for the future can be cancelled by the user. Finally, a user can comment on his/her past reservations, but these comments needs to be approved by an admin. A hotel owner is a user too, who can register their hotels with their information and these hotel users can see their hotel information page and reservations page. A hotel user can also change the information and the pricing of the rooms. In addition, admin can approve or reject the hotel registrations and also an admin can approve or reject the comments of the users about hotels.

Current system is being made by using Jsp and Java. Also, for database creation, MySQL Workbench is used. In all current reservation systems, flow of events works in this way. However, details about the system can be vary from one to another such as which architecture the system use or is the system supported by any hardware equipment etc.

**3. Proposed Software Architecture**

Our system is designed as a “as-is” system. The main purpose of the system is to ensure simplicity of operations. Users should make a reservation easily without any problems. A user can surf on the site without making a registration, but for a reservation attempt, registration is mandatory. Registration and and making a reservation is completely free.

The system inferface is designed as quite simple, target-driven and easy to use to fulfill our main purpose. Because of our limited resources, we keep the hotel side system much more basic than the user side, but the user side is done by taking examples from “Booking.com”.

**3.1 Overview**

The system intends to display the hotels to user, which are in accordance with their requests. This operation is designed to be user friendly and it can be done with just a few clicks. When we look at the hotel side, this part of the system is also user friendly and easy to use. Hotels are able to control their own reservations, rooms and their pricing. Finally, an admin is always checking the main parts of the system and if necessary, s/he can intervene quite easily.

**3.2 System Decomposition**

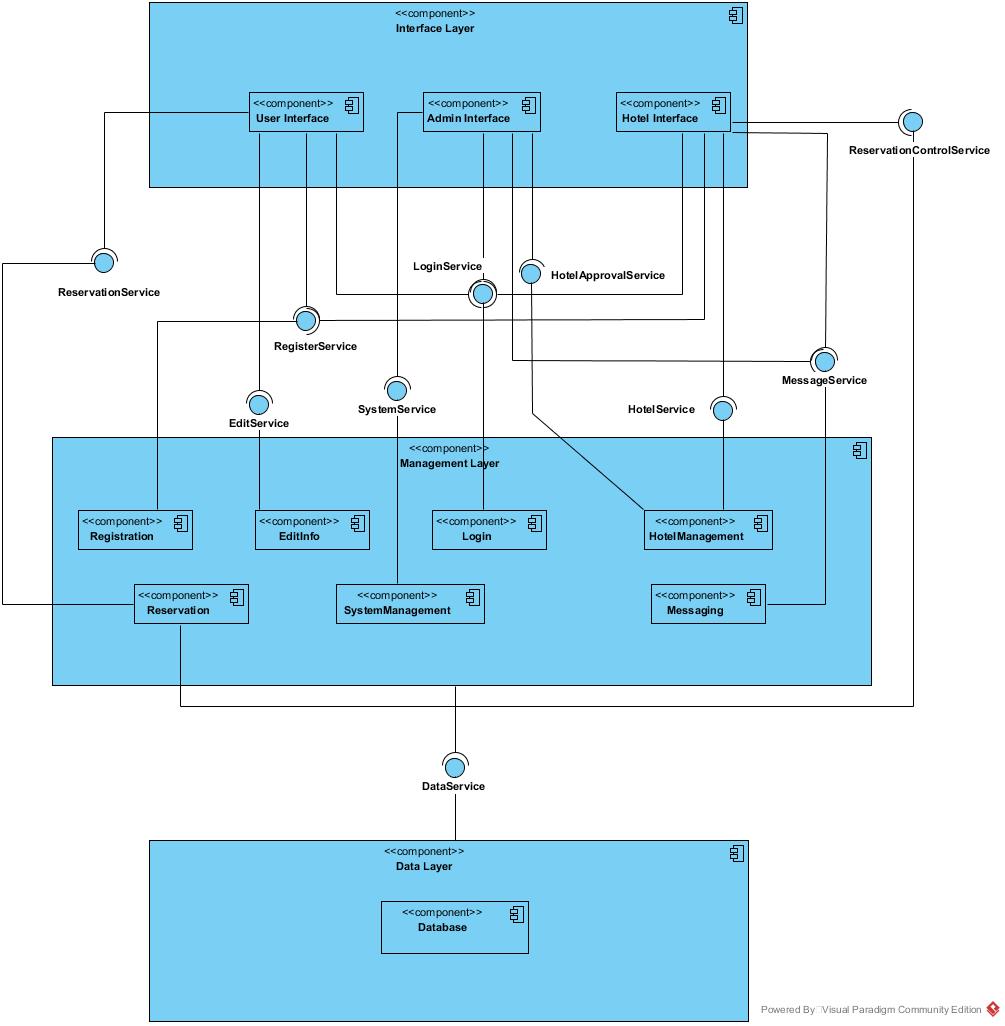
Hotel Reservation Application is a system which is containing 3 layers. These are User interface, Management Layer and Data Layer.

Figure – 1: Coupling view of Subsystem Decomposition

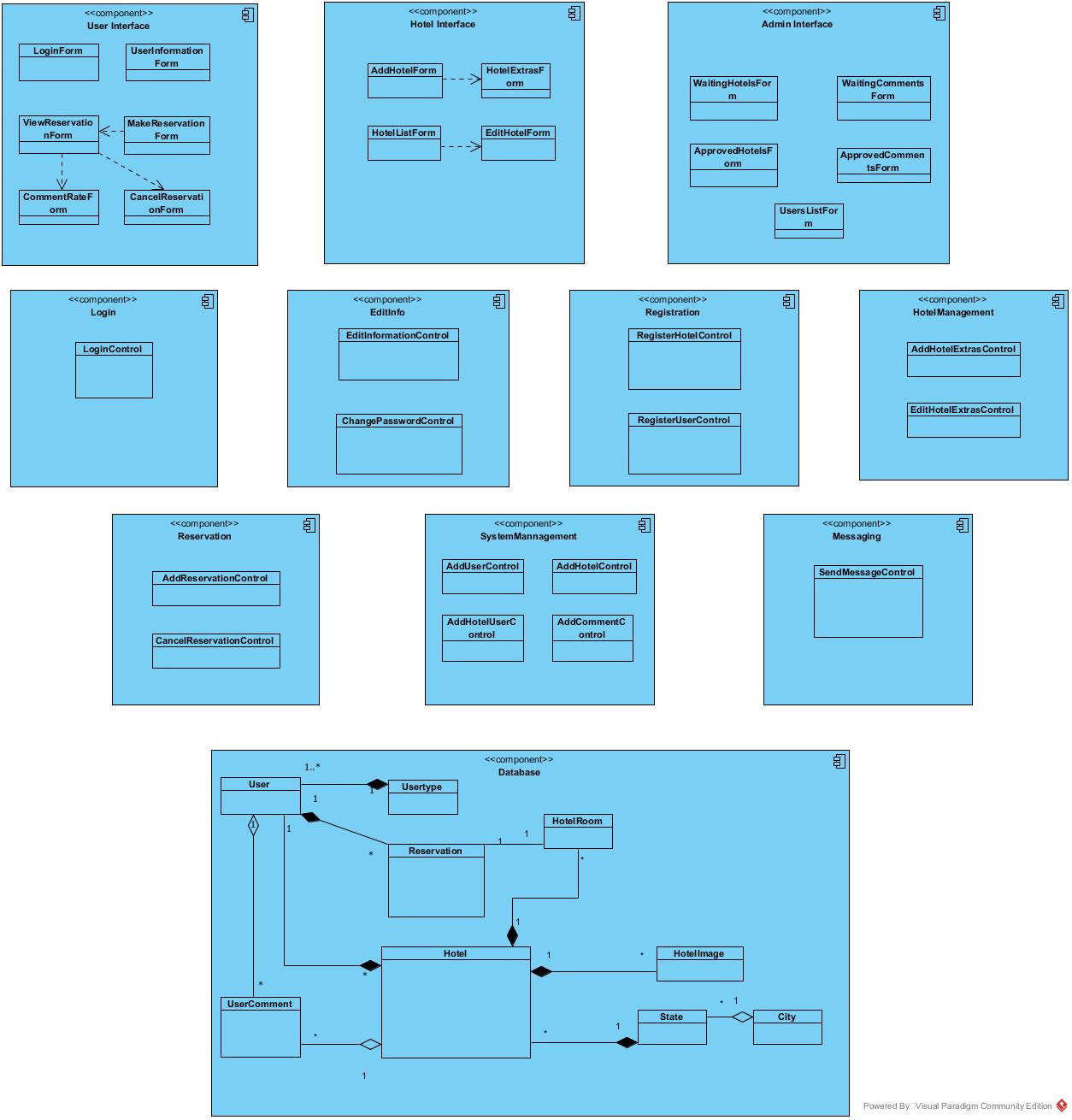
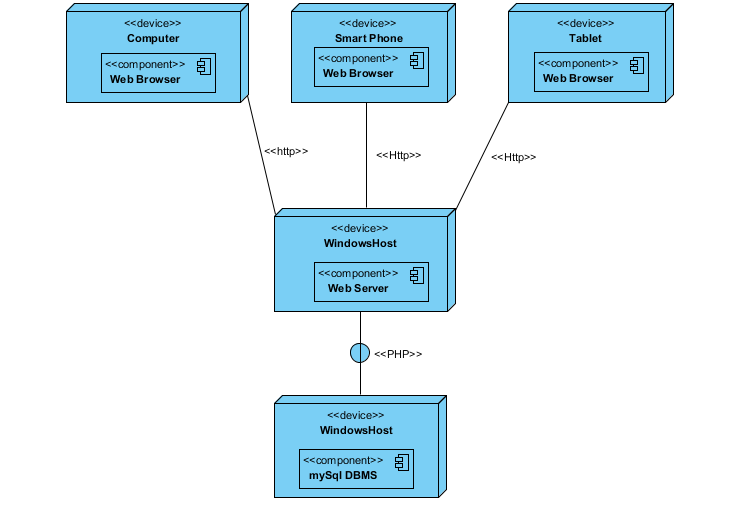


Figure – 2: Cohesion view of Subsystem Decomposition

Our system decomposition, the subsystems are: User Interface, Hotel Interface, Admin Interface, Registration Subsystem, Login Subsystem, Edit Info Subsystem, Hotel Management Subsystem, Reservation Subsystem, System Management Subsystem, Messaging Subsystem and Database Subsystem. **User Interface** contains Login Form, User Information Form, Password Change Form, View Reservation Form, Make Reservation Form and etc.

As summary, User Interface provides services to display common forms such as log in, password change, making reservation, canceling reservation to all users. **Hotel Interface** has Add Hotel Form, Edit Hotel Form, Hotels List Form, and Add Hotel Extras Form. So, Hotel Interface provides services to display forms which are related hotel owners’ functions such adding a new hotel. Admin Interface contains Waiting Hotels Form, Waiting Comments Form, Approved Comments Form, and Approved Hotels Form. So, **Admin Interface** provides services to display forms which are related admins’ functions such as approving hotels form, approving comments from users on their reservations. In addition, our system decomposition has 8 subsystems. **Login Subsystem** provides services for users (user, hotel, and admin) to login, it contains Login Control object**. Edit Info Subsystem** provides services for users to update their information and to change password, it contains Edit Info Control object to update and Change Password Control object to change password. **Registration Subsystem** provides services for users and hotels to register to Hotel Reservation System. **Hotel Management Subsystem** provides services for hotel users to add extras to their own hotels or edit their own hotels’ extras. **Reservation Subsystem** provides services for users to make new reservations or to cancel any reservation. **System Management Subsystem** provides services for adding user, hotel user to system and also provides to add hotels and comments to system by an admin approval. **Messaging Subsystem** provides services for users (admin and hotel user) to send messages about hotels before approving or rejecting the hotel. Admin can request new things by sending a message to hotel user with sending a message. **Data Access Subsystem**; contains all our persistent objects, this part could be called Model of MVC(model view controller).

**3.3 Hardware Software Mapping**

****

**3.4 Persistent Data Management**

There are some persistent data stored by the system so that the data can outlive a single execution of the system. We decide to store persistent data in a “database”. These data are; Users of the system (Registered User, Admin, Hotel User), Hotel data is offered by different Hotel Users, that includes their own hotel information, cost, images. City and State data are also stored in the system. These two data are related to hotel data. Related to the Hotel data, we also store Hotel Room data for each Hotel which has “N” instances related to each hotel. Reservation data of both hotel room and user are stored in the system. In addition, User comments and rate data is stored in the system as well. User can rate and comment for a accomplished reservation. Hotel and Comment data are stored in the system and admin reach these data to approve requests. These are the persistent data that are need to be stored in database so that the data can outlive a single execution of the system. In Hotel Reservation, we decided to use relational database for the data management infrastructure, due to its advantages on the usage and implementation.

The commonly known advantages of relational database are listed as below;

* Avoids data duplication
* Avoid inconsistent records
* Easier to change data
* Easier to change data format
* Data can be added and removed easily
* Easier to maintain security

**3.5 Access Control and Security**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ACTOR/CLASS** |  | **Functions** |  | **User.java** |  | **Hotel.java** |  | **HotelRoom.java** |  |
|  |  |  |  |  |  |  |  |  |  |
| **User** |  |  |  | takeUserInfo()  takeSelectedReservation()  takeFirstname()  takeUserId()  checkType()  checkTypePages()  checkRegistered()  userDetails()  takeUserInfo() |  | checkPassword() |  |  |  |
| **System Administrator** |  |  |  | addUser()  editUser()  takeUserInfo()  checkTypePages()  checkRegistered()  checkAdmin()  userDetails()  takeUserInfo() |  | checkPassword() |  | takeRoomCost()  takeRoomType()  takeHotelId() |  |
| **Hotel** |  |  |  | checkTypePages()  checkRegistered()  checkRegisteredHotel()  userDetails()  sendMessage()  takeMessageSelected() |  | checkType()  checkTypePages()  checkRegistered()  checkPassword() |  | addHotelRoom()  takeRoomCost()  takeRoomType()  takeHotelId() |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ACTOR/CLASS** |  | **Functions** |  | **User.java** |  | **Hotel.java** |  | **reservation.java** |
|  |  |  |  |  |  |  |  |  |
| **User** |  |  |  | takeSelectedReservation() |  | takeHotelList() |  | cancelReservation()  takeMyReservations()  addComment() |
| **System Administrator** |  |  |  |  |  |  |  |  |
| **Hotel** |  |  |  | deleteMessage()  messageNotify()  readFlag()  readMessage()  takeUserInfo() |  | addHotel()  takeHotelname()  takeHotelId() |  |  |

**3.6 Global Software Control**

The Hotel Reservation application architecture have an explicit software control. The system’s dynamic control distributed among different controllers such that each page delegates some responsibility to other pages. The request initiations are event-driven:

* The user opens a web browser and enter the application’s web site.
* Web browser checks whether Java plug-ins are installed or not. And controls whether the application can be usable or not due to plug-ins.
* The user initiates the User Interface Subsystem by opening the application.
* System Management Subsystem checks the login information that are taken from the user. Requests the user information from Database Management Subsystem.
* System Management Subsystem redirects the user to the index page and handles user’s inputs and actions.
* Database Management Subsystem receives the data from System and keeps them all. Only developer can reach to database directly and edit information on database.
* Shortly, System Management Subsystem observe the authorization of actors and take user’s request and organize them. Each action from user is a “request” for the system.

Hotel Reservation system is a multi-threaded system. Some functions are synchronized all the time. For instance, remaining room number for a hotel is refreshing when any reservation is completed or cancelled. So, the user see and reach to the recent data every time.

**3.7 Boundary Conditions**

**S**tartup : enter the website and select city

ShutDown . logo ut and close the website

Error Conditions :

-Signing in :

\*Username and password do not match

\*Username or password is wrong

\*Username does not exist.

-Siigning up :

\*E-mail address does not valid

\*Mandatory fields do not filled

-Searching By Distinct

\*When searching by distinct, ne recorded hotel may be found by a user

-Select Payment

\*During online operation, a credit card does not valid.

**4. Subsystem Services**

**4.1 The User Interface Subsystem**

-Gets user input

-Sends user information to the Connection Directory Subsystem

-Allows commenting and rating

**4.2 The Reservation Subsystem**

-Takes items to buy inside from UI Subsystem.

-Sends these items to the Hotel Subsystem.

**4.3 The EditInfo Subsystem**

- Sends user details to the User Interface Subsystem.

**4.4 The Administrator Management Subsystem**

-Handles the Connection Directory Management

-Takes the admin information

**4.5 The Connection Directory Subsystem**

-Receives the different requests from the different subsystems

**4.6 The Hotel Management Subsystem**

-Takes items to buy inside from UI Subsystem.

-Sends these items to the Hotel Subsystem.

**4.7 The Messaging Subsystem**

**-** Sends these items to the Hotel Subsystem.

-Takes items to buy inside from Admin Subsystem

**4.8 The Registration Subsystem**

-Provides usability to system to user interface subsystem and hotel interface subsystem.

**5.**

[**http://www.codejava.net/java-se/jdbc/insert-file-data-into-mysql-database-using-jdbc**](http://www.codejava.net/java-se/jdbc/insert-file-data-into-mysql-database-using-jdbc)

[**http://www.tutorialspoint.com/jsp/jsp\_file\_uploading.htm**](http://www.tutorialspoint.com/jsp/jsp_file_uploading.htm)