tatiloradaburada.com

System Design

1.0

15.12.2019

Ali Haydar Konuk

Oğuzhan Aydın

Fatih Mehmet İdgü

Yaren Çoşkun

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc433996772)

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

[1.2. Design Goals 1](#_Toc433996774)

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

[1.4. References 1](#_Toc433996776)

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

[3. Proposed Software Architecture 1](#_Toc433996778)

[3.1. Overview 1](#_Toc433996779)

[3.2. System Decomposition 1](#_Toc433996780)

[3.3. Hardware Software Mapping 2](#_Toc433996781)

[3.4. Persistent Data Management 2](#_Toc433996782)

[3.5. Access Control and Security 2](#_Toc433996783)

[3.6. Global Software Control 2](#_Toc433996784)

[3.7. Boundary Conditions 2](#_Toc433996785)

[4. Subsystem Services 2](#_Toc433996786)

[5. References 2](#_Toc433996787)

SYSTEM DESIGN DOCUMENT[1]

# Introduction

tatiloradaburada.com is a website that has 3-tier architecture and response time, robustness, reliability, security, portability, extensibility and usability design goals. This document will be going to take requirement analysis document and take it to developers point of view. It will bring time saving on the implementation phase.

## Purpose of the System

The main purpose of the tatiloradaburada.com is, providing an easy, understandable and quality reservation service for the users who are looking for a hotel to do vacation. The system is useful for users, search and reserve a hotel room or cancel his/her reservation over the internet at any time. Users can make comment to indicate their satisfaction. By this way, users can find the best hotel for their vacation to make a reservation

## Design Goals

* **Response Time:**

User's requests or actions must be acknowledged fast and put in to action as quick as possible, because one of the tatiloradaburada.com's main goal is to save time while searching a hotel.

* **Robustness:**

Users can give invalid input to tatiloradaburada.com and tatiloradaburada.com must avoid this by giving enough information to user and also having such input fields so that the input errors that can be caused by the users are minimized.

* **Reliability:**

System has to continue operations that user requests without errors. Reliability of the system should be high and the data taken from user should be retained securely. A hotel is added to website if and only if admin is approval.

* **Security:**

tatiloradaburada.com must retain the important data that belongs to user successfully. This must be avoided so authentication system should work perfectly and the password or personal information of users must be encoded.

* **Portability:**

tatiloradaburada.com can be used from any device that opens a web browser.

* **Extensibility:**

In future stages of tatiloradaburada.com new functions can be added with requests and feedbacks from users. The system can support any function that involve hotelowners and registeredusers.

## Definitions, Acronyms, and Abbreviations

tatiloradaburada.com = our website

## References

Tatilbudur.com is our referenced website. Tatilbudur.com functions are a bit bigger in range. Tatilbudur.com has different types of tours and vehicles you can choose. We tried to keep our system easier and more understandable.

# Current Software Architecture

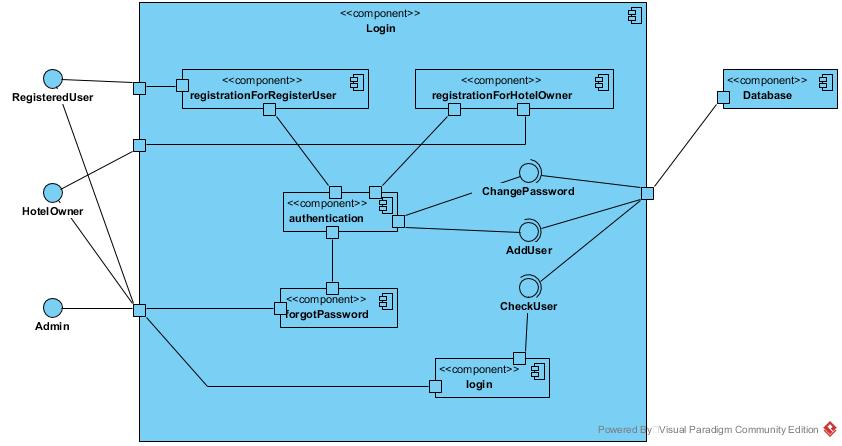
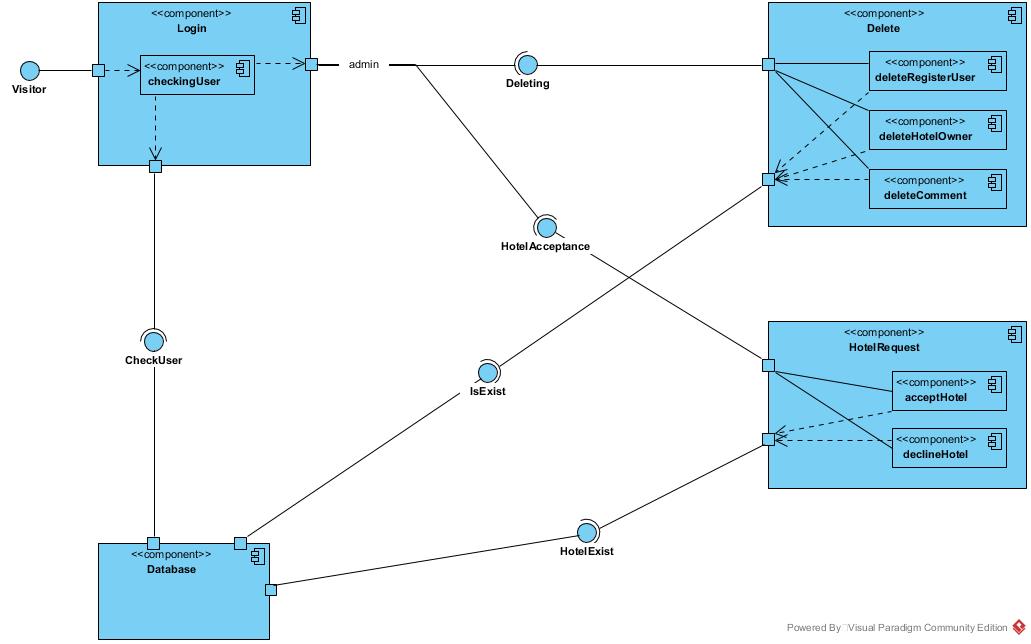
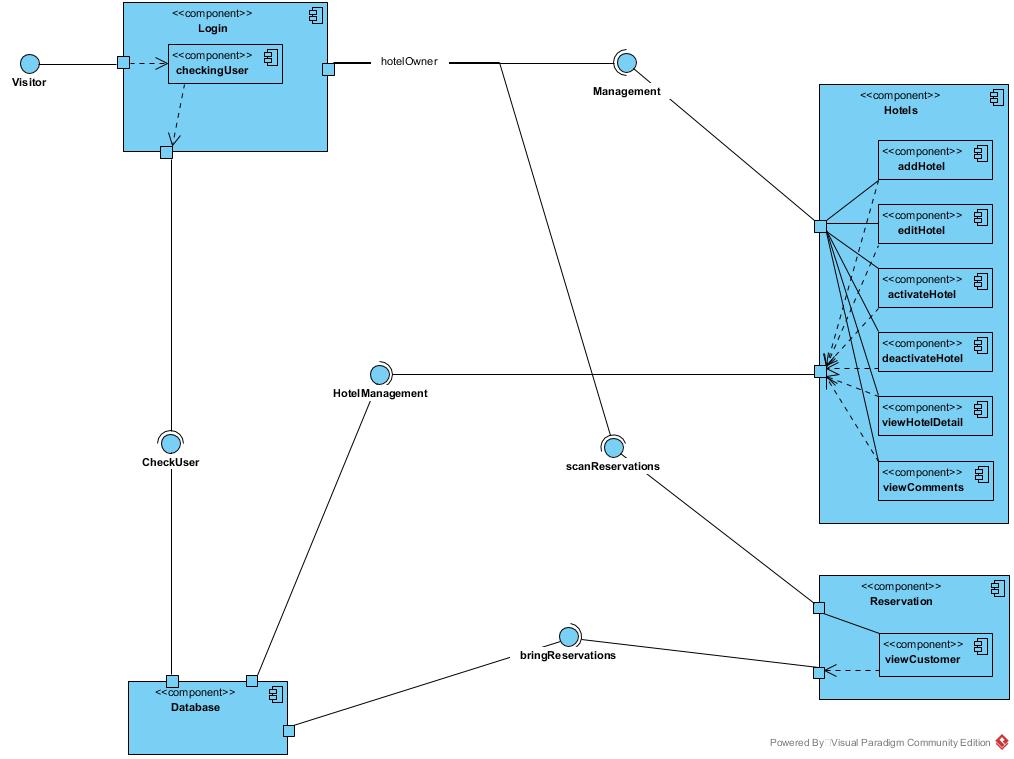
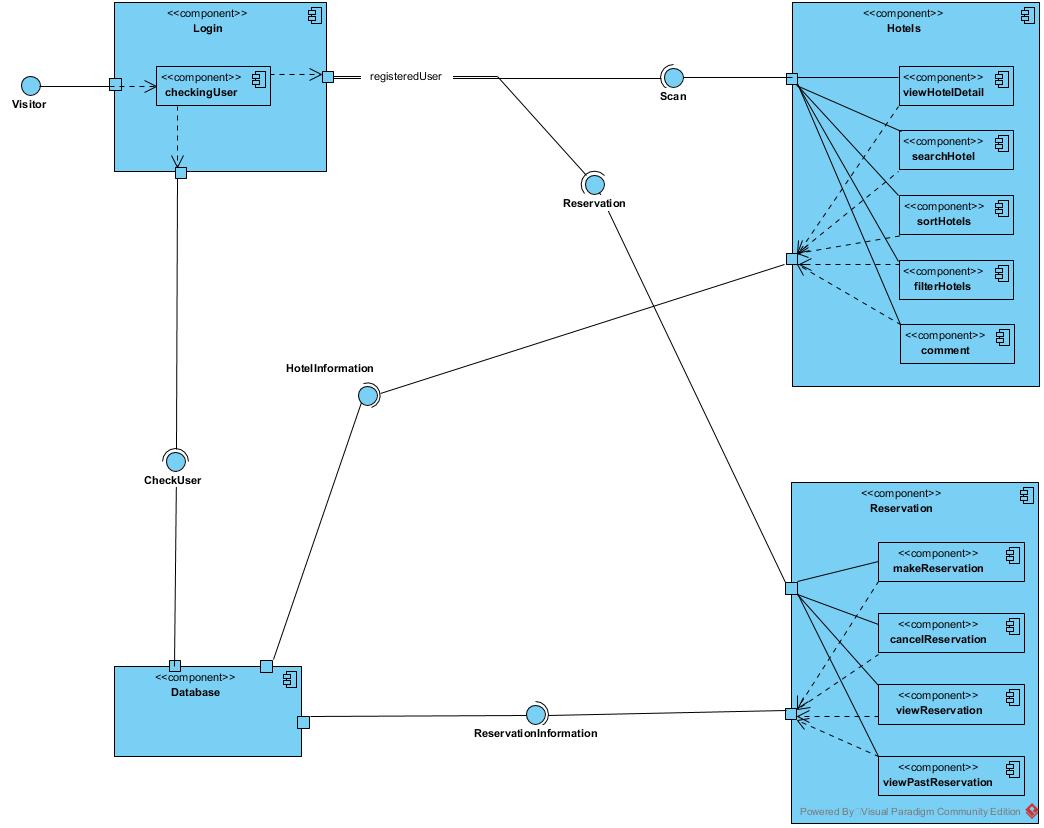
We use 3-tier architecture. A 3-tier architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing. They are often used in applications as a specific type of client-server system. 3-tier architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers. Doing so gives greater flexibility to development teams by allowing them to update a specific part of an application independently of the other parts. This added flexibility can improve overall time-to-market and decrease development cycle times by giving development teams the ability to replace or upgrade independent tiers without affecting the other parts of the system..

# Proposed Software Architecture

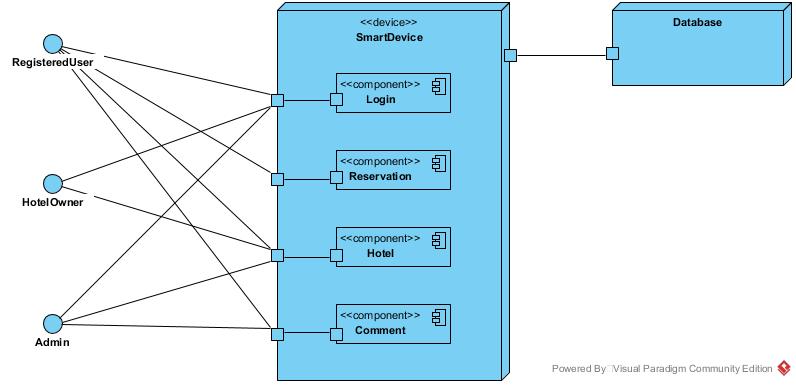
## Overview

* **Registered User Interface:** This interface provides that services which contains viewing all hotels, searching, sorting and filtering hotels with their properties and keywords. Also, this interface’s services contain making reservation and commenting about hotels.
* **Hotel Owner Interface:** This Interface provides that services which contains viewing hotels details. This interface’s services contain adding new hotel, editing existing hotel, activating or deactivating existing hotel. Also, this interface’s services contain selected any hotel’s reservation information.
* **Admin Interface:** This Interface provides that services which contains deleting registered user account, hotel owner account and comment. Also, this interface’s services contain accept or decline of sent add hotel request.
* **Login Subsystem:** This subsystem contains login, registration for both register user and hotel owner, changing password.

## System Decomposition

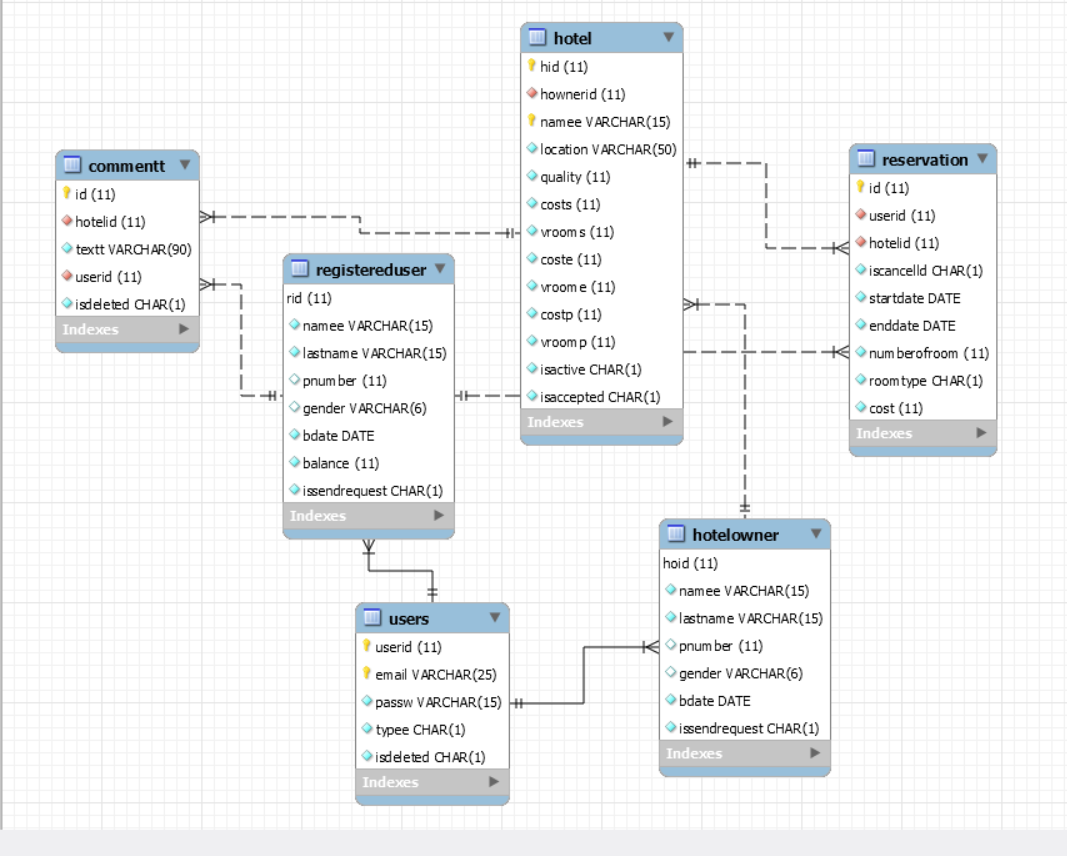


## Hardware Software Mapping



## Persistent Data Management

tatiloradaburada.com Database Model ER.



## Access Control and Security

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **.** | **System** | **Reservation** | **Hotel** | **Comment** |
| **Visitor** | Register() | Filter()  Search() |  |  |
| **RegisteredUser** | login() logout() forgetPassword() | Filter()  Search()  makeReservation()  cancelReservation()  showReservations()  showPastReservations() |  | makeComment()  showComments() |
| **HotelOwner** | register() login() logout() forgetPassword() | ShowReservations() | addHotel()  editHotel()  activateHotel()  deactivateHotel() |  |
| **Admin** | login() logout() forgetPassword()  deleteAccount() |  | acceptHotel()  declineHotel() | DeleteComment() |

The users of tatiloradaburada.com register into the system with register form. They can be hotel owner or registered user. Users can login into the system with their email and password. In database, passwords keep encoded. Any user can’t see real string of passwords. The user type checks from the database “User” table, user type attribute.

## Global Software Control

The tatiloradaburada.com will be implemented with an object-oriented program, multi-step sequences and users connecting to the service at the same time. Multiple users may access to database, to avoid corruption of the data.

## Boundary Conditions

Tatiloradaburada.com starts-up by admin clicking “Start” on server. With the initialization, users may register or log in to the system.

Within the implementation, all predicted errors and exceptions are ready to be handled, instead of terminating the website, Changing requests of the database will be saved instantly to keep the data loss or corruption minimum.

To shut down the system, the admin clicks “Stop” on server. All logged in users will be logged out and disconnected to the system. Data saved to the database until the last request before the termination.

# Subsystem Services

tatiloradaburada.com has five main subsystem that are responsible for actions of different users.three subsystems are as follows.

1. Login Subsystem
2. Hotel Owner Subsystem
3. Registered User Subsystem
4. Admin Subsystem
5. **Login Subsystem**

This subsystem is responsible for authentication and account actions of all user. Some futures are provided for all user. On the other hand some of them like register is not.

Operations are like listed below.

* Register()
* Login()
* Logout()
* ForgetPassword()

**B. Hotel Owner Subsystem**

This subsystem is responsible for actions that are done by HotelOwner, This subsystem provides to hotel owner adding new hotel and responsible for his/her hotel to edit existing one, adding new one also can see reservations which are reserved his/her hotel.

Operations are like listed below.

* ShowReservations()
* addHotel()
* editHotel()
* activateHotel()
* deactivateHotel()

**C.** **Registered User Subsystem**

This subsystem is responsible for actions that are done by RegisteredUser.

Full list of operations on this subsystem as follows.

* Filter()
* Search()
* makeReservation()
* cancelReservation()
* showReservations()
* showPastReservations()
* makeComment()
* showComments()

**D.Admin Subsystem**

This subsystem is responsible for actions that are done by Admin. This subsystem responsible for submitted requests. This subsystem provides accept or deny submitted requests.

Full list of operations as follows.

* acceptHotel()
* declineHotel()
* DeleteComment()

# References

RAD of tatiloradaburada.com

Eric J. Braude, Michael E. Bernstein, Software Engineering: Modern Approaches 2/e Wiley, 2011

www.jinfonet.com