Software Documentation

for

<Restaurant Reservation System>

Version 1.0 approved

(15th July 2020)

Prepared by

<Nur Aida Farahanie binti Ahmad 192494>

<Anis Nur Hajar binti Ahmad Rizal 192560>

<Amir Khaidir bin Adzeman 192212>

<Muhammad Afif Mirza bin Mohd Farid 192128>

Table of Contents

1. Introduction

This section gives a scope description and overview of everything included in this SDD document. Also, the purpose for this document is described and a list of abbreviations and definitions are provided.

1.1. Purpose

P-1: This SDD describes the design documentation of RRS, a restaurant reservation system between customers and restaurateurs. It is a written description of a software product that will be used by the developer.

P-2: To give a software development team overall guidance to the architecture of the RRS system with interface details and design constraints.

1.2. Scope

S-1: The scope of this project includes the management of restaurant reservation between customer and restaurateur

S-2: This document gives a detailed description of the software architecture of the restaurant reservation system.

S-3: This SDD specifies the structure and design of some of the modules discussed in the SRS.

S-4: This SDD also provides some of the use cases that had transformed into sequential and activity diagrams.

S-5: The class diagrams show how the programming team would implement the specific module for this RRS.

1.3. Document's intended audience

The Software Description document is intended for:

· System Vendor (Team RRS): By using the SDD, the team member can easily understand the software and it helps the team member to improve the current project features or add new features to existing system.

· Users: By using the SDD, users can easily understand how to use the software. In this project our users can be anyone who wants to use Restaurant Reservation System to book their table in advance at a restaurant.

1.4. References

R-1: IEEE Software Engineering Standards Committee

R-2: RRS Software Requirement Specification (SRS) V1.0

1.5. Acronyms or Abbreviations

A-1: RRS – Restaurant Reservation System.

A-2: SDD – Software Design Description

A-3: SRS – Software Requirement Specification

A-4: IEEE - Institute of Electrical and Electronics Engineers

1.6. Overview

The remaining chapters and their contents are listed below.

O-1: This document is written according to the standards for Software Design Documentation explained in "IEEE Recommended Practice for Software Design Documentation".

O-2: Sections 3 – 4 contain simple explanation about RRS and the system analysis

O-3: Section 5-6 shows the detailed design of RRS.

O-4: Section 7 shows the system architectures in different views of web application solution.

O-5: Section 8-10 contains the design for the system databases and the describes about the API for RRS and its third-party component.

2. Project Background

Restaurant Reservation System (RRS) is a web-based application that consists of customer and restaurateur as the users for this system. From RRS, the customers are allowed to reserve a table at their selected restaurant and can update their reservation. On the other hand, restaurateur can manage the reservation by viewing the reservation made by the customer and set up the customer's table based on the information given by the system. The restaurateur can also cancel the reservation made by the customer if requested by the customer or if the customer late.

The overall purpose of this RRS is basically to create and develop a restaurant reservation system for both customers and restaurateur which can ease their reservation method and save time and effort for both customer and restaurateur. This is basically because usually when customer wanted to eat at their preferred restaurant, the restaurant is full-house and they have to wait in order to get a table, by having a reservation system, they can book earlier and get a table without having to wait.

3. System Analysis

3.1. External Interface Requirements

3.1.1. User Interface

Since the system is used by different users which are customer and restaurateur, the category of user interfaces (UI) is depending on the functions the user is authorized to perform. Customer can register at the register page of the system and fill out all the related information. Both customer and restaurateur need to sign in to use the functionality of the system. After sign in, customer will be redirected to homepage that has variety registered restaurant. Through this homepage interface, all basic user's functions are shown but can be accessed after login and based on user access authorization. Customer can only make reservations once they have already login. Customer also can view reservation that has been made. There are other functionalities that customer can perform whish are update and cancel. Customer can update the several information such as date and time. Customer also can cancel the reservation. Customer can view menu by clicking the menu tab provided. Restaurateur can see all the reservation that has been made by the customer at the view reservation page in the restaurateur side. Restaurateur also can cancel the reservation that has been made by the customer. The Restaurant Reservation System (RRS) is an independent system that doesn't utilize an outer system. Be that as it may, the system will require database the executives programming like MySQL to deal with the way toward putting away the data information, and web server programming that arranges the dispersion of clients in the nearby system.

3.1.2. Hardware Interface

Just like any other system, The Restaurant Reservation System (RRS) requires a basic computer that consists of CPU, monitor, keyboard, and mouse or laptop, and smartphones for input and output. Besides, for the direction, the system is connected with GPS or Maps device in every organization vehicle to get the transportation details. The map allows customers to locate the restaurant and navigate there easily. The least amount of space needed by the hard disk is 80 GB and a minimum of 2 GB RAM space to run the booking process.

3.1.3. Software Interface

The system will require software Prime Faces for its more interactive support to program and create The Restaurant Reservation System. We have chosen the Windows operating system for its best support and user-friendliness. The system also requires software like MySQL to manage the database of the system. Reservation process of the system that organizes the distribution of users in the local network and able to do this online booking. Users can also use internet browsers such as Google Chrome and Mozilla Firefox to book the restaurant.

3.1.4. Communication Interface

The system communication interface completely depends on server software to ensure correct send and retrieves data from the database. Other than that, it is an online protocol to connect between GPS and the system. Besides, emails also will be used as the communication platform between the customer and restaurateur. RRS also supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking, for instance.

3.2. Product Functions

3.2.1 Reserve Table

RT-1: User need to insert and select the date required for their table reservation

RT-2: User need to insert and select the time required for their table reservation

3.2.2 Register

R-1: New customer must register in the system in order to reserve table and use the system

R-2: Customer will first have to click the 'Sign Up' button

R-3: Customer need to fill out the information in the registration form such as name, email, phone number and password

R-4: Customer click 'sign up now' button to register their account

R-5: Data will be saved into the database once they registered their account

3.2.3 Sign In

SI-1: Authorized customer and restaurateur sign in into the system using their email address and password

SI-2: Customer and restaurateur enter their registered email address and password

SI-3: Customer and restaurateur choose to sign in as customer or restaurateur at the drop-down list provided

SI-4: Customer and restaurateur click 'Sign In'

3.2.4 Update Reservation

UR-1: Authorized customer can update their reservation if there are any changes that they wanted to do

UR-2: Customer makes the changes to his or her reservation information

UR-3: Customer clicks 'Update' button

UR-4: Data will be saved and updated in database
UR-5: Customer successfully update the reservation information

3. 2. 5 Cancel Reservation

CR-1: Authorized customer and restaurateur can cancel the reservation that has been made

CR-2: Customer or restaurateur choose to cancel the reservation

CR-3: Customer and restaurateur is prompt to choose whether to confirm cancelation or not

CR-4: Customer successfully cancel the reservation

3. 2. 6 View Menu

VM-1: Authorized customer can view menus offer at the selected restaurant

VM-2: Customer chooses the 'Menu' tab and food category will be shown

VM-3: Customer clicks the 'Beverages' tab to view the list of beverages offered

VM-4: Customer clicks the 'Menu List and Price' tab to view the price list of the menu

3. 2. 7 View Reservation

VR-1: Authorized customer and restaurateur can view the reservation that has been made

VR-2: Customer and restaurateur click the Calendar icon in the homepage

VR-3: Customer and restaurateur will be redirected into 'view reservation' page

VR-4: View reservation page will display all the reservations that have been made

4. Design Considerations

4.1. Assumptions and Dependencies

AD1-: Since the system is a web-based system there is a need for the internet browser. It will be assumed that the users possess internet connectivity.

AD-2: It is assumed that the user aware of basic operations of a computer and web pages.

AD-3: RRS running depends on the availability of internet connection at all the time of usage.

AD-4: The whole flow of the system depends on the successful login of the users.

4.2. General Constraints

GC-1: The languages that will be provided in the RRS is only English language.

GC-2: The restaurant reservation system is implemented using Glassfish server for application server and Java DB for the database.

GC-3: The restaurant reservation system is implemented by using JavaServer Faces (JSF) and prime faces framework for presentation layer.

4.3. Development Method

RRS is designed with flexibility for further development or modification. The system is divided into manageable processes that are grouped to sub-modules and modules that are built with abstraction.

5. Detailed Design

This detailed design will explain all the important things about this system by all classes in
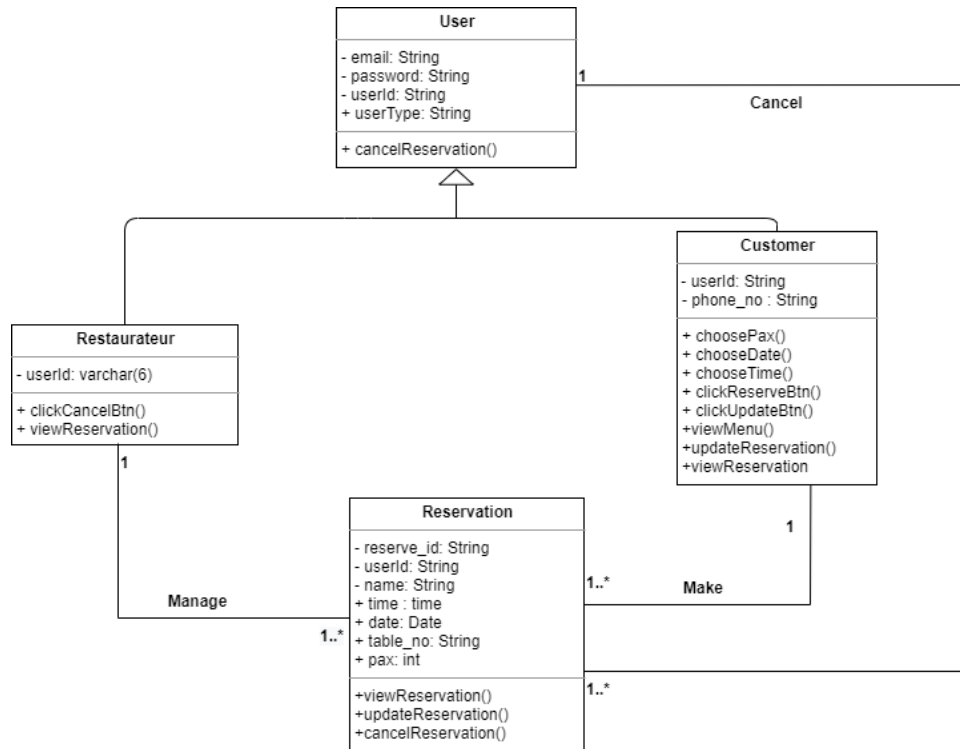
the whole system.

5.1. Class Diagram

Figure 1: Class diagram for RRS system

6. System Architecture

6.1. Architecture Overview

Design pattern used in Restaurant Reservation System (RRS) is Model-View-Controller architecture which divides this RRS into three interconnected parts that consists of:

i)   Model: It is responsible for managing the data and objects for the application
ii) View: It is responsible for displaying the user interfaces and the output to the user
iii) Controller: It is responsible for acting as the brain or logic of the system. It connects the model and view layer.

Considering with the Restaurant Reservation System representation:

i)   Model: Java Class
ii) View: Java Server Faces (JSF) xhtml
iii) Controller: Managed Bean

The overall of this architecture is mainly managed by the controller. The end user which is the client send a request and it will reach at the controller component, then the controller component is responsible to choose a particular technology as a model component. The model will retrieve requested data from the database and responses back to the controller. From the received response, the controller modifies the data in view and send the output to the client.
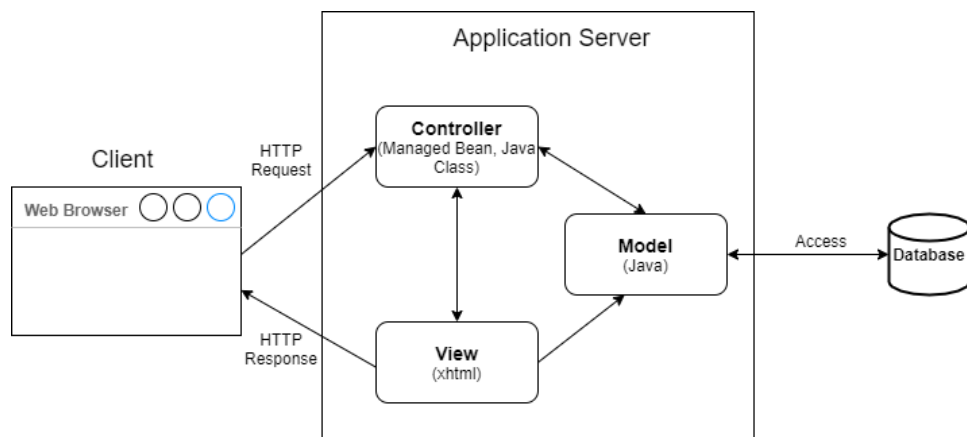

Figure 2: The illustration of the MVC architecture in client server

6.2. Physical View

This is the deployment diagram for Restaurant Reservation System. This diagram visualized the hardware processors or devices of this system, the links of communication between them and the placement of software files on that hardware.
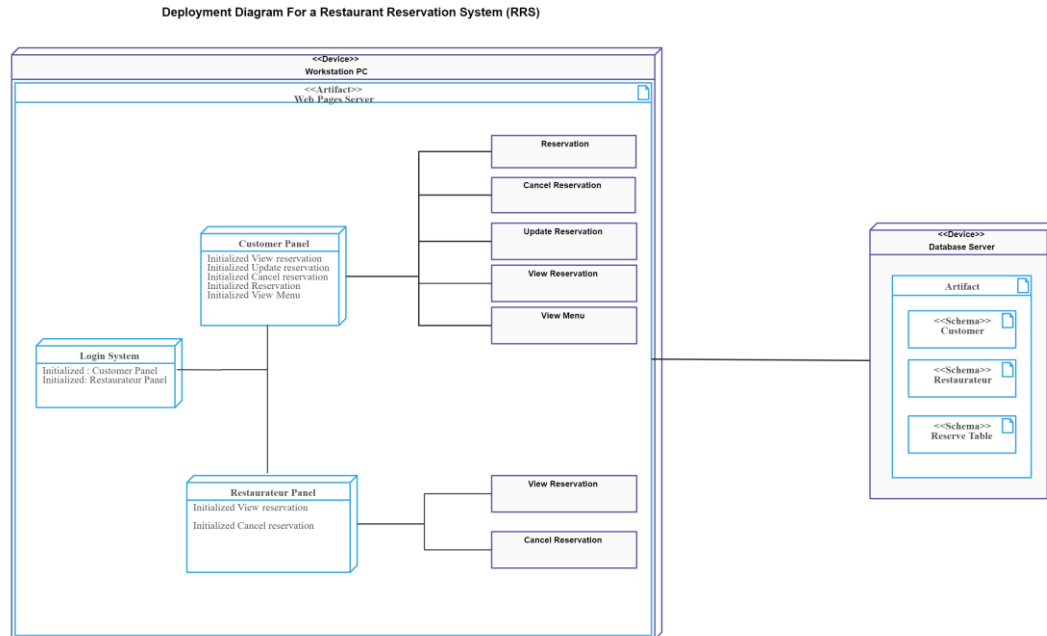

Figure 3: The illustration of the RRS Deployment Diagram

6.3.Development View

6.3.1.   Component Diagram

This is the development view for Restaurant Reservation System and is shown in a component diagram. This diagram visualized the relationship between different smaller components that is wired to form a complex component in this system.
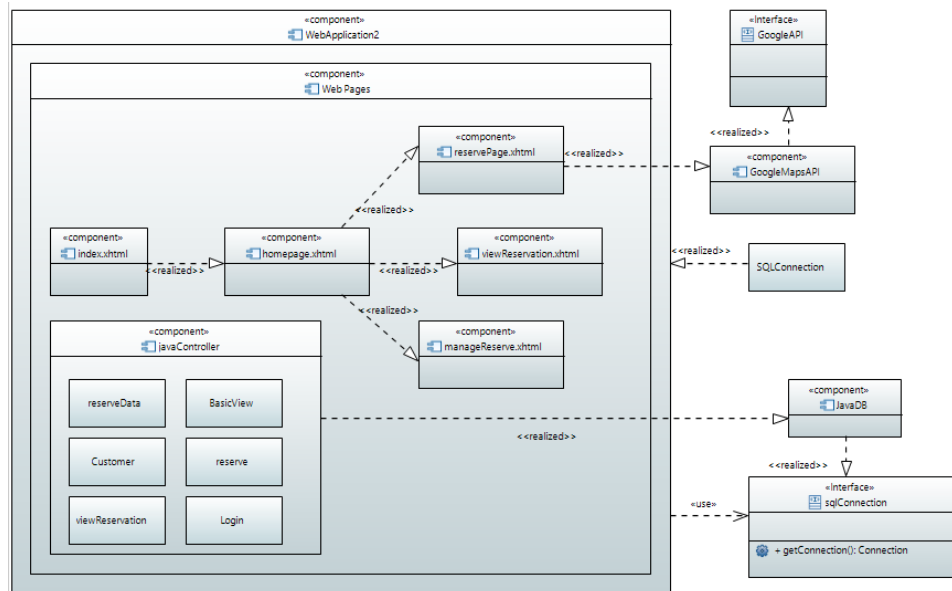

Figure 4: The Component Diagram of the RRS System

7. Database Design

The database design for this Restaurant Reservation System is illustrated in an Entity Relationship Diagram (ERD). This illustration represents the data that describes how data is connected to each other. This diagram consists of entities in this system which are composited with their attributes or properties related.
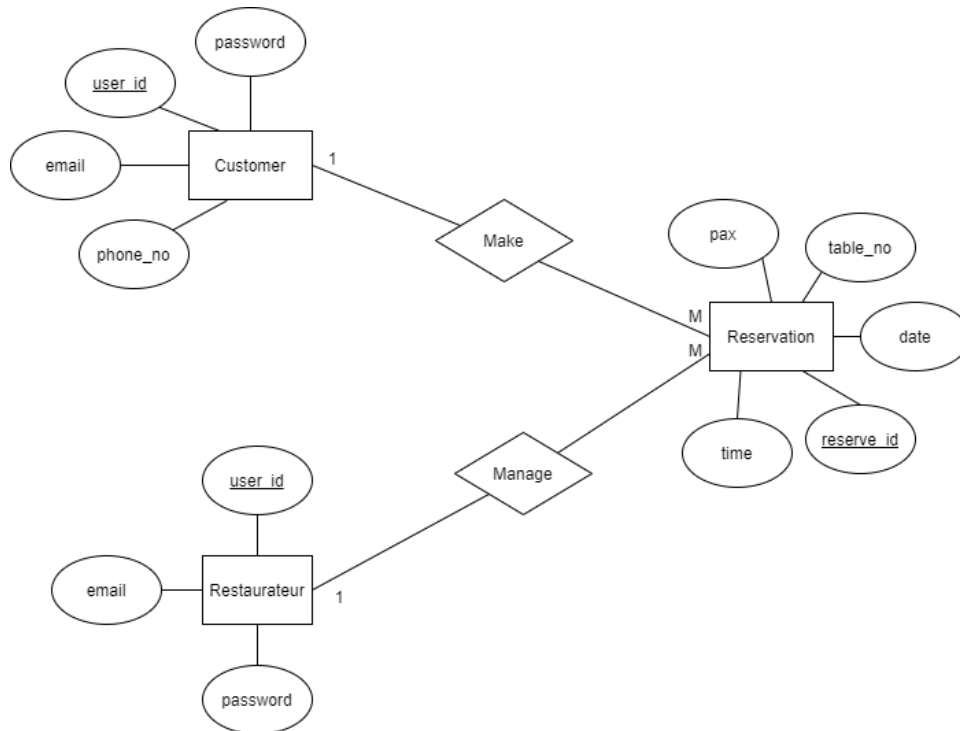

Figure 5: Entity relationship diagram for RRS

7.1.Data Dictionary

CUSTOMER TABLE

CUSTOMER			
email	Customer email	varchar2(40)	
user_id	Customer id	varchar2(20)	pk
phone_no	Customer contact	int (12)	
password	Customer password	varchar2(20)	

Table 1: Customer Table

RESTAURATEUR TABLE

RESTAURATEUR			
email	Restaurateur email	varchar2(40)	
password	Restaurateur password	varchar2(20)	
user_id	Restaurateur id	Varchar2(20)	pk

Table 2: Restaurateur Table

RESERVATION TABLE

RESERVATION			
reserve_id	Reservation Id	varchar2(10)	pk
time	Reservation time	varchar2(10)	
date	Reservation date	Date	
table_no	Number of reserve table	varchar2(10)	
phone_no	Customer contact	varchar (15)	
pax	Pax for reservation	int (5)	
email	Customer email	varchar2(40)	

Table 3: Reservation Table

8. Component Description

No.	Component	Description
1.	API	This component is the application programming interface that allows two applications to interact with each other. For RRS, the API used is the Google Maps API.
2.	Models	This component handles the requests including processing of data and business rules. For this system, a model might handle the table reservation requests for a customer to book a table.
3.	Views	Views is the UI components for different requests. For RRS, this component will display the information needed by the user for example the interface to update or cancel a reservation.
4.	Controllers	Controller is a component that determine which components to call for a particular request. In RRS, when the user click the "Book a Table" button, the controller will call the models to process the data and give back to the controller and the controller will passed the data, which is the UI for table reservation to be displayed to the customer by Views component.
5.	Prime Faces	Prime Faces is a UI component for JSF (Java Server Faces) based application that is being used in RRS for the front-end of this system.

Table 4: Component Description

9. Description on API and Third-Party Component

API that is being used in Restaurant Reservation System is Google Maps API. This section will explain about how this RRS interacts with Google Maps API. From this RRS, the system will allow the user to access the location of the restaurant that has been reserved. As this Google Maps is embedded into RRS web pages, RRS can retrieve data from Google Maps and the location, business hours, reviews, phone numbers and other information about the restaurant will be displayed to the customers.

10. Glossary

i. Class diagram

Type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

ii. API

API stands for Application Programming Interface which act as the intermediary of the software that enables two programs or applications to interact to each other. In other words, API is the messenger who delivers requests to the provider from the requester and then return answers back to the requester.

iii. User interface

The user interface (UI) is everything designed into an information device with which a person may interact. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website.

iv. Primary key (PK)

A primary key, also called a primary keyword, is a key in a relational database that is unique for each record.

v. Software Design Documentation (SDD)

A written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project.

vi. Entity Relationship Diagram (ERD)

An entity relationship diagram (ERD) explains the relationships of sets iof entity stored in a database. An entity is an object or a component of data. An entity set is a group of similar entities. These entities can have attributes that define its attributes.

vii.   Component Diagram

        A component diagram explains how components are connected together to form larger components or software systems and can ease in figuring out the model implementation details.

viii.   Deployment Diagram

        Deployment diagram is a UML diagram type that explains the execution infrastructure of a system, including modules such as hardware or software execution environments, and the intermediary linking them. Deployment diagrams are usually used to represent the physical hardware and software of a device.

11. Bibliography

1. Sevim.A.(2011). Software Design Description for AJCON, Applet to JSF Converter Version 2.0. Retrieved from https://senior.ceng.metu.edu.tr/2011/groupteaplet/docs/Software-Design-Descriptions-Detailed.pdf

2. Cao, W., Wan, Y., Tu, H., Shang, F., Liu, D., Tan, Z., ... & Xu, Y. (2011). A web-based appointment system to reduce waiting for outpatients: a retrospective study. *BMC health services research*, *11*(1), 318.