**YeOrada**

System Design Document

1.0

02.04.2020

Ahmet Can Terlikçi

Yaren Çoşkun

Ali Haydar Konuk

Fatih Mehmet İdgü

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

# Introduction

YeOrada 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 YeOrada is, providing an easy, understandable and user-friendliness search service for the clients who are looking for restaurant, café or bar. Clients can give rate, make comment and share photo about the place. By this way clients find best fitting place for themselves.

## Design Goals

* **Ease of Use:**

A user who has never used search system before, must be able to learn how to find a place in a short time which is about 15 minutes by searching.

* **Response Time:**

The search function in the system must be display a result at most 30 seconds to the user. Other actions must be acknowledged at most 45 seconds.

* **Robustness:**

Clients can write invalid or wrong input to the empty fields, and YeOrada must avoid this these invalid inputs by giving enough information to client also YeOrada has 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:**

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

* **Extensibility:**

In future stages of YeORada new functions can be added with requests and feedbacks from users. The system can support any function that involve customers and clients.

.

## Definitions, Acronyms, and Abbreviations

## References

zomato.com is our reference website.

# Current Software Architecture\*\*\*\*\*\*

Describe the architecture of the system being replaced**. If there is no previous system**, this section can be replaced by **a survey of current architectures for similar systems**. The purpose of this section is to make explicit the background information that system architects used, their assumptions, and common issues the new system will address.

# Proposed Software Architecture

## Overview

Present a bird’s-eye view of the software architecture and briefly describes the assignment of functionality to each subsystem.

## System Decomposition

Describe the decomposition into **subsystems and the responsibilities** of each. **This is the main product of system design.**

## Hardware Software Mapping

Describe how subsystems are assigned to hardware and off-the-shelf components. It also lists the issues introduced by multiple nodes and software reuse.

## Persistent Data Management

Describe the persistent data stored by the system and the data management infrastructure required for it. This section typically includes the description of **data schemes, the selection of a database, and the description of the encapsulation of the database**.

## Access Control and Security

Describe the user model of the system in terms of an access matrix. This section also describes security issues, such as the selection of an authentication mechanism, the use of encryption, and the management of keys.

## Global Software Control

Describe how the global software control is implemented. In particular, this section should describe how requests are initiated and how subsystems synchronize. This section should list and address synchronization and concurrency issues.

## Boundary Conditions

Describe the start-up, shutdown, and error behavior of the system. (If new use cases are discovered for system administration, these should be included in the requirements analysis document, not in this section.)

# Subsystem Services

Describe the **services provided by each subsystem**. Although this section is usually empty or incomplete in the first versions of the SDD, this section serves as a reference for teams for the boundaries between their subsystems. The interface of each subsystem is derived from this section and detailed in the Object Design Document.

# References

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on [1]).

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