Photonest

Requirements Specification and Analysis

1.0

08.03.2020

Dilara Ünbay

Nursena Karakulah

Feyzullah Berkay Danış

Özay Ezerceli

Prepared for

SE301 Software Engineering



Table of Contents

1. Introduction 1

1.1. Purpose of the System 1

1.2. Scope of the System 1

1.3. Objectives and Success Criteria of the Project 1

1.4. Definitions, Acronyms, and Abbreviations 1

1.5. Overview 1

2. Current System 1

3. Proposed System 1

3.1. Overview 1

3.2. Functional Requirements 2

3.3. Nonfunctional Requirements 2

Usability 2

Reliability 2

Performance 2

Supportability 2

Implementation 2

Interface 2

Packaging 2

Legal 2

3.4. System Models 2

Scenarios 2

Use case model 2

Object model 2

3.5. Project Schedule 2

4. Glossary 3

5. References 3

REQUIREMENTS ANALYSIS DOCUMENT [1]

The Requirement Analysis Document (RAD) should be written after the use case model is stable, that is, when the number of modifications to the requirements is minimal. The requirements, however, are updated throughout the development process when specification problems are discovered or when the scope of the system is changed.

Please set your word processor’s language to English, enable spell checker to detect the misspellings, and follow the formatting in this document.

# Introduction

The purpose of this section is to provide a brief overview of the function of the system and the reasons for its development, its scope, and references to the development context (e.g., reference to the problem statement written by the client, references to existing systems, feasibility studies). The introduction also includes the objectives and success criteria of the project [1].

## Purpose of the System

## Scope of the System

## Objectives and Success Criteria of the Project

## Definitions, Acronyms, and Abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the RAD.

## Overview

This subsection should:

* Describe what the rest of the RAD contains
* Explain how the RAD is organized.

# Current System

If the new system will replace an existing system, this section describes the functionality and the problems of the current system. Otherwise, this section describes how the tasks supported by the new system are accomplished now.

# Proposed System

Documents the requirements elicitation and the analysis model of the new system

## Overview

Presents a functional overview of the system.

## Functional Requirements

Describes the high-level functionality of the system.

## Nonfunctional Requirements

Describes user-level requirements that are not directly related to functionality. This includes usability, reliability, performance, supportability, implementation, and interface, operational, packaging, and legal requirements.

### Usability

### Reliability

### Performance

### Supportability

### Implementation

### Interface

### Packaging

### Legal

## System Models

Describes the scenarios, use cases, object model, and dynamic models for the system. This section contains the complete functional specification, including mock-ups illustrating the user interface of the system and navigational paths representing the sequence of screens.

### Scenarios

A scenario is an instance of a use case.

### Use case descriptions

**1)**  *Use case name* **Logout***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Logout” function from the profile page of the application by clicking the “Logout” option.  
   * 1. **SYSTEM** presents a warning text to the **User**.
2. The **User** reads the message and clicks the “OK” button.
   * 1. **SYSTEM** responds by re-directing the **User** to the login page.

*Entry condition* The **User** has clicked the Logout option.*Exit conditions* The **User** has been re-directed to the login page.

The **User** has clicked on the CANCEL button.   
*Quality* The **User** is immediately re-directed to the login page.  *Requirements*

**2)**  *Use case name* **RegisterUser***Participating* Initiated by **Visitor**  *actorsFlow of events*

1. The **Visitor** activates the “Register User” function from the login page of the application by clicking the “Sign Up” text.  
   * 1. **SYSTEM** presents a sign up form to the **Visitor**.
2. The **Visitor** fills the form by entering an e-mail, a full name, a username and a password. The **Visitor** submits the form by clicking the “Sign Up”.
   * 1. **SYSTEM** receives the sign up form, checks the given e-mail and sends a verification mail to the given e-mail.

5.  The **Visitor** receives the verification mail and clicks the link to activate his/her account.

6. **SYSTEM** presents an informing message to the **Visitor**. 

7.  The **Visitor** reads the message and ends the sign up process by clicking the “Activate” button. 

8. **SYSTEM** presents a success message.

*Entry condition* The **Visitor** has clicked the Sing Up button.*Exit conditions* The **Visitor** has received a success message.

The **Visitor** has left the required fields empty.

The **Visitor** has entered an invalid e-mail.   
*Quality* The success message is shown immediately after the **Visitor** clicks the *requirements* Activate button.

**3)**  *Use case name* **Login***Participating* Initiated by **Visitor**  *actorsFlow of events*

1. The **Visitor** opens the application.  
   * 1. **SYSTEM** presents a login form to the **Visitor**.
2. The **Visitor** fills the form by entering an e-mail and a password. The **Visitor** submits the form by clicking the “Login” button and activates “Login” function.
   * 1. **SYSTEM** receives the login form, checks the given information and re-directs the **User** to the home page.

*Entry condition* The **Visitor** has opened the application.*Exit conditions* The **Visitor** has been re-directed to the home page.

The **Visitor** has left the required fields empty.

The **Visitor** has entered an invalid information.   
*Quality* The **Visitor** is immediately re-directed to the login page.  *Requirements*

### Use case model

A use case is a generalization of a number of scenarios. Therefore, the number of scenarios must be equal to or greater than the number of use cases.

### Object model

The analysis object model, depicted with UML class diagrams, includes classes, attributes, and operations. The analysis object model is a visual dictionary of the main concepts visible to the user.

## Project Schedule

Prepare Gannt Chart, and add it to this section.

# Glossary

To establish a clear terminology, developers **identify the participating objects** for each use case. Developers should **identify, name, and describe them** unambiguously and collate them into a glossary.

# References

This subsection should:

* Provide a complete list of all documents referenced elsewhere in the RAD, or in a separate, specified document.
* Identify each document by title, report number - if applicable - date, and publishing organization.
* Specify the sources from which the references can be obtained.

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.