Hospital Appointment System

Requirements Specification and Analysis

Version 0.1

21.06.2016

Muhammed Bilal Kutlu

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc453932853)

[1.1. Purpose of the System 1](#_Toc453932854)

[1.2. Scope of the System 1](#_Toc453932855)

[1.3. Objectives and Success Criteria of the Project 1](#_Toc453932856)

[1.4. Definitions, Acronyms, and Abbreviations 1](#_Toc453932857)

[1.5. Overview 1](#_Toc453932858)

[2. Current System 1](#_Toc453932859)

[3. Proposed System 1](#_Toc453932860)

[3.1. Overview 1](#_Toc453932861)

[3.2. Functional Requirements 2](#_Toc453932862)

[3.3. Nonfunctional Requirements 2](#_Toc453932863)

[Usability 2](#_Toc453932864)

[Reliability 2](#_Toc453932865)

[Performance 2](#_Toc453932866)

[Supportability 2](#_Toc453932867)

[Implementation 2](#_Toc453932868)

[Interface 2](#_Toc453932869)

[Packaging 2](#_Toc453932870)

[Legal 2](#_Toc453932871)

[3.4. System Models 2](#_Toc453932872)

[Scenarios 2](#_Toc453932873)

[Use case model 2](#_Toc453932874)

[Object model 2](#_Toc453932875)

[Dynamic model 2](#_Toc453932876)

[User interface—navigational paths and screen mock-ups 3](#_Toc453932877)

[4. Glossary 3](#_Toc453932878)

[5. References 3](#_Toc453932879)

[6. Project Plan 3](#_Toc453932880)

REQUIREMENTS ANALYSIS DOCUMENT[

# Introduction

## Purpose of the System

* Hospital Appointment System (HAS) is a web-based application that provides the users get appointment from the hospital. It is like hospital automation.
* There are two level of users;

1. User
2. Administration
3. Staff

* We will call Hospital Appointment System as HAS.
* HAS includes a few information about the user (patient) like Name, disease, appointment date etc. to future usage.
* HAS provides user to list the available date and the personal (doctors) to get appointment in that time interval.
* HAS generates reports for the appointment information and identification number for that appointment.

## Scope of the System

* The system will be used to get social and personal information from the patient and creating an online appointment by using that information.
* The system will be stored data for every staff and patient.
* The system will be used to list of the patients, their quick information and their appointment information.
* The system will be provided to the staff if the patient attended to his own appointment or not.
* The system will be blocked the user who did not come his appointment for 3-times.

## Objectives and Success Criteria of the Project

* Proper and conducive project plan
* Assigning tasks to the team members by using GitHub.
* Reviewing and doing a rework when needed.
* Managing project risks efficiently.
* Allocating time for process improvement

## Definitions, Acronyms, and Abbreviations

HAS: Hospital Appointment System.

## 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

* Usability

Program should be use easily by People who have the avarage information about using computer .It must be usable at any time.

* Reliability

People can’t access personal information which recorded on database by developer.

* Performance

This system must be quick.

It supports more than one user at same time. But it is not developed to people who are take appointment at the same time and same appointment.

Information which are uploads on database must be updatable according to the requirements.

* Supportability

Developer be responsible to provide continuance, compatibility and testability on created program.

* Implementation

There is not create unproblematic system so there will be errors. However those error should be decreases.

* Interface

Application can use on devices which have Internet and its browser should be usable.

* Packaging

System’s all steps as a package are within GitHub.

* Legal

Project’s all contents are protected by the law of copyright.

## 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 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.

### Dynamic model

The dynamic model is depicted with sequence diagrams and with state machines. Sequence diagrams represent the interactions among a set of objects during a single use case. State machines represent the behavior of a single object (or a group of very tightly coupled objects). The dynamic model serves to assign responsibilities to individual classes and, in the process, to identify new classes, associations, and attributes to be added to the analysis object model.

When working with either the analysis object model or the dynamic model, it is essential to remember that these models **represent user-level concepts, not actual software classes or components.**

### User interface—navigational paths and screen mock-ups

# 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]).

# Project Plan

This subsection provides the rough cost, effort estimations. Further information will be delivered in lecture.

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