

**ÇANKAYA UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**

**Project Report**

**Version 1**

**CENG 408**

Innovative System Design and Development II

**<*PROJECT ID*>**

**<*PROJECT NAME*>**

*Name Surname*

*Student Number*

*Name Surname*

*Student Number*

Advisor: *The name of your advisor*

Table of Contents

[Table of Contents ii](#_Toc446345463)

[Abstract v](#_Toc446345464)

[Özet: v](#_Toc446345465)

[1. Introduction 1](#_Toc446345466)

[1.1 Company Background 1](#_Toc446345467)

[1.2 Problem Statement 1](#_Toc446345468)

[1.3 Background or Related Work 1](#_Toc446345469)

[1.4 Solution Statement 1](#_Toc446345470)

[1.5 Contribution 1](#_Toc446345471)

[2. Literature Search 2](#_Toc446345472)

[2.1 Library Research 2](#_Toc446345473)

[2.2 Internet Research 2](#_Toc446345474)

[3. Summary 2](#_Toc446345475)

[3.1 Summary of Conceptual Solution 2](#_Toc446345476)

[3.2 Technology Used 2](#_Toc446345477)

[4. Software Requirements Specification 3](#_Toc446345478)

[4.1 Introduction 3](#_Toc446345479)

[4.1.1 Purpose 3](#_Toc446345480)

[4.1.2 Scope of Project 3](#_Toc446345481)

[4.1.3 Glossary 3](#_Toc446345482)

[4.1.4 References 4](#_Toc446345483)

[4.1.5 Overview of Document 5](#_Toc446345484)

[4.2 Overall Description 5](#_Toc446345485)

[4.2.1 Product Perspective 5](#_Toc446345486)

[4.2.2 Development Methodology 5](#_Toc446345487)

[4.2.3 Product Functions 5](#_Toc446345488)

[4.2.4 User Characteristics 7](#_Toc446345489)

[4.2.5 Constraints 7](#_Toc446345490)

[4.2.6 Assumptions and Dependencies 8](#_Toc446345491)

[4.3 Requirements Specification 8](#_Toc446345492)

[4.3.1 External Interface Requirements 8](#_Toc446345493)

[4.3.2 Functional Requirements 9](#_Toc446345494)

[4.3.3 Performance Requirements 10](#_Toc446345495)

[4.3.4 Design constraints 11](#_Toc446345496)

[4.3.5 Software system attributes 11](#_Toc446345497)

[4.3.6 Other Requirements 12](#_Toc446345498)

[5. Software Design Description 13](#_Toc446345499)

[5.1 Introduction 13](#_Toc446345500)

[5.1.1 Purpose 13](#_Toc446345501)

[5.1.2 Scope 13](#_Toc446345502)

[5.1.3 Glossary 13](#_Toc446345503)

[5.1.4 References 14](#_Toc446345504)

[5.1.5 Overview of document 14](#_Toc446345505)

[5.2 Deployment diagram 14](#_Toc446345506)

[5.3 Architecture design (You can use your sections and headers) 14](#_Toc446345507)

[5.3.1 SDD subsections here… 14](#_Toc446345508)

[5.3.2 SDD subsections here… 15](#_Toc446345509)

[5.4 Data structure design (You can use your sections and headers) 15](#_Toc446345510)

[5.5 Use case realizations 16](#_Toc446345511)

[5.5.1 Use Case: Use Case 1 (Add use cases here) 16](#_Toc446345512)

[5.5.2 Use Case: Use Case 2 (Add use cases here) and so on… 17](#_Toc446345513)

[5.6 Interface design 17](#_Toc446345514)

[5.7 Help system design 18](#_Toc446345515)

[5.8 Index 18](#_Toc446345516)

[6. Test Plan 59](#_Toc446345517)

[6.1 Introduction 59](#_Toc446345518)

[6.1.1 Version Control 59](#_Toc446345519)

[6.1.2 Overview 59](#_Toc446345520)

[6.1.3 Scope 59](#_Toc446345521)

[6.1.4 Terminology 59](#_Toc446345522)

[6.2 Features to be tested 59](#_Toc446345523)

[6.2.1 Login (LG) << Specify the name and acronym of the feature here 59](#_Toc446345524)

[6.2.2 Add User (AU) << Specify the name and acronym of the feature here 60](#_Toc446345525)

[6.3 Features not to be testes 60](#_Toc446345526)

[6.4 Item pass / fail criteria 60](#_Toc446345527)

[6.4.1 Exit Criteria 60](#_Toc446345528)

[6.5 References 60](#_Toc446345529)

[6.6 Test design specifications 60](#_Toc446345530)

[6.6.1 Login (LG) (Features to be tested) 60](#_Toc446345531)

[6.6.2 Add User (AU) (Features to be tested) 61](#_Toc446345532)

[6.7 Detailed Test Cases 59](#_Toc446345533)

[6.7.1 LG.AD.01 59](#_Toc446345534)

[6.7.2 LG.AD.02 59](#_Toc446345535)

[7. Test Results 59](#_Toc446345536)

[7.1 Individual Test Results 59](#_Toc446345537)

[7.2 Summary of Test Results 60](#_Toc446345538)

[7.3 Exit Criteria 60](#_Toc446345539)

[7.4 Known Problems 60](#_Toc446345540)

[7.5 Conclusion 60](#_Toc446345541)

[8. Conclusions 61](#_Toc446345542)

[Acknowledgement 61](#_Toc446345543)

[References 61](#_Toc446345544)

[Appendices 61](#_Toc446345545)

# Abstract

The abstract should contain a very short description (100 - 250 words) of the report. When you write the abstract, imagine that the reader will not read anything else, but you must get your major point across immediately. This, in fact, is what abstracts are all about. Keep in mind that an abstract represents a very short summary of the entire report and should not simply be a subset of the introduction or conclusion section.

First state the a) problem to be solved, and then b) your solution. Then specify c) your key results from the work and what you learned from the research.

**Key words:**

This section describes the least 3 key words of the project. (Please refer key word description list provided by YÖK)

# Özet:

Özet bölümü raporun çok kısa (100 – 250 kelime) olarak tanımını içermelidir. Özet yazarken, okuyucunun başka bir şey okumayacağı düşünülmelidir ama sadece önemli noktaların üzerinde durulmalıdır. Aslında bu bölüm tümü hakkındadır. Bu bölüm tüm raporun kısa bir özeti olduğu unutulmamalıdır ve giriş ve sonuç bölümlerinin bir alt elemanı değildir.

İlk bölüm a) çözülecek problem ve sonra b) senin çözümün, c) çalışma sonucunda öğrendiklerinden çıkardığın önemli sonuçlar.

**Anahtar Kelimeler:**

Bu bölümde projeyi anlatan temel kelimeler kullanılacak.

#  Introduction

The introduction should be approximately 0.5 to one page in length, and should contain the following information:

## Company Background

Give a brief information about company, such as history, what they do, organization …

## Problem Statement

State the problem to be solved. Why are you doing this work and what significance does it have in the relevant literature? Even if your project is applied (as opposed to research-oriented), you are building a system because a problem, requiring a solution in the form of a computer program, exists.

## Background or Related Work

 State who else has worked on this problem or similar problems (you should do most of your citations here). For applied projects, provide information on other existing programs which will use your program.

## Solution Statement

State your solution to the problem.

## Contribution

State how your solution builds upon and extends current technology.

Sometimes, the introduction can be split into subsections or more than one section including the following parts: Background, Related Work and Motivation.

# Literature Search

While working on your project, you have compiled a database of literature that supports your work. Literature sources can and should include the following:

## Library Research

Start with your library. Do a keyword search of all relevant online or card catalogues. Try different keywords because massive amounts of information are indexed and you must provide the right word that happened to be used to create the original index table.

## Internet Research

Exploit the massive Internet resource by using the information tools. Pay attention to only credible web sites.

# Summary

## Summary of Conceptual Solution

This should be a conceptual description defining the solution to the problem. Avoid using code and implementation details here; instead, define the solution in terms of algorithms, pseudo code and clear mathematical reasoning. Also, use figures, tables, and statistics to get your point across. ***This description will be used in poster preparation.***

## Technology Used

Mention also the technology used to build the solution, such as java, .net, oracle, MySQL … Also include a block diagram of your solution.

# Software Requirements Specification

## Introduction

### Purpose

This subsection should

1. Delineate the purpose of the SRS;
2. Specify the intended audience for the SRS.

Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### Scope of Project

This subsection should

1. Identify the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.);
2. Explain what the software product(s) will, and, if necessary, will not do;
3. Describe the application of the software being specified, including relevant benefits, objectives, and goals;
4. Be consistent with similar statements in higher-level specifications (e.g., the system requirements specifications), if they exist.
5. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### Glossary

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents. Example:

Table 1 Glossary of SRS

|  |  |
| --- | --- |
| **Term** | **Definition** |
| KMS - Knowledge Management System | An online system that satisfy the knowledge creation, capturing, sharing and application requirements of the SMEs, the Clusters, the Beneficiary, and the TAT, as defined in the Terms of Reference (ToR) document of the Project. KMS System is cited simply “The System”. |
| SME | Small and Medium size Enterprise (the company) |
| Ministry of Economy (MoE) | A Public Authority of the KMS System. The beneficiary of the Project. MoE will host and own the KMS System as delivered. MoE will use the system to capture, share, distribute and apply the information, and will benchmark the clusters with the others by getting proper information from the system. |
| Publish | Adding, creating, viewing, reading, changing, updating, removing and deleting the content |
| Editor | A member that examines a content submitted to the KMS system by a member, and has the ability to recommend approval of the content for publication or to request that changes be made in the content, and has the ability and the authority to publish the content. Normally, the KMS Coordinator is the Editor. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
|  |  |
|  |  |

### References

This subsection should

1. Provide a complete list of all documents referenced elsewhere in the SRS;
2. Identify each document by title, report number (if applicable), date, and publishing organization;
3. Specify the sources from which the references can be obtained.

This information may be provided by reference to an appendix or to another document.

1. IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
2. SME-Empowering Project’s Inception Report, ECORYS.

### Overview of Document

This subsection should

1. Describe what the rest of the SRS contains;
2. Explain how the SRS is organized.

## Overall Description

This section of the SRS should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3 of the SRS, and makes them easier to understand.

### Product Perspective

This subsection of the SRS should put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the SRS defines a product that is a component of a larger system, as frequently occurs, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software. A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

### Development Methodology

Describe your development methodology here. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### Product Functions

This subsection of the SRS should provide a summary of the major functions that the software will perform. For example, an SRS for an accounting program may use this part to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires. Sometimes the function summary that is necessary for this part can be taken directly from the section of the higher-level specification (if one exists) that allocates particular functions to the software product.

Note that for the sake of clarity

1. The functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time.
2. Textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.

Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

#### Function 1 (Add a function of your product)

Describe the function in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

#### Function 2 (Add a function of your product)

Describe the function in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

#### Function 3 (Add a function of your product)

Describe the function in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

#### Function 3 (Add a function of your product) and so on…

Describe the function in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### User Characteristics

This subsection of the SRS should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. It should not be used to state specific requirements, but rather should provide the reasons why certain specific requirements are later specified in Section 3 of the SRS.

#### User Type 1 (Add a user type here)

Describe the user type in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

#### User Type 2 (Add a user type here) and so on…

Describe the user type in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### Constraints

This subsection of the SRS should provide a general description of any other items that will limit the developer’s options. These include

1. Regulatory policies;
2. Hardware limitations (e.g., signal timing requirements);
3. Interfaces to other applications;
4. Parallel operation;
5. Audit functions;
6. Control functions;
7. Higher-order language requirements;
8. Signal handshake protocols (e.g., XON-XOFF, ACK-NACK);
9. Reliability requirements;
10. Criticality of the application;
11. Safety and security considerations

### Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

## Requirements Specification

### External Interface Requirements

This should be a detailed description of all inputs into and outputs from the software system. It should complement the interface descriptions in section 2 and should not repeat information there. It should include both content and format as follows:

1. Name of item;
2. Description of purpose;
3. Source of input or destination of output;
4. Valid range, accuracy, and/or tolerance;
5. Units of measure;
6. Timing;
7. Relationships to other inputs/outputs;
8. Screen formats/organization;
9. Window formats/organization;
10. Data formats;
11. Command formats;
12. End messages.

#### User interfaces

Describe the user interfaces in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

#### Hardware interfaces

Describe the hardware interfaces in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

#### Software interfaces

Describe the software interfaces in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

#### Communications interfaces

Describe the communications interfaces in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### Functional Requirements

Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These are generally listed as “shall” statements starting with “The system shall …”

These include

1. Validity checks on the inputs
2. Exact sequence of operations
3. Responses to abnormal situations, including
	1. Overflow
	2. Communication facilities
	3. Error handling and recovery
4. Effect of parameters
5. Relationship of outputs to inputs, including
	1. Input/output sequences
	2. Formulas for input to output conversion

It may be appropriate to partition the functional requirements into subfunctions or subprocesses. This does not imply that the software design will also be partitioned that way.

#### Functional Requirement 1 (Add a functional requirement)

Describe the function requirement in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

#### Functional Requirement 2 (Add a functional requirement)

Describe the function requirement in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

#### Functional Requirement 3 (Add a functional requirement) and so on…

Describe the function requirement in detail. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### Performance Requirements

This subsection should specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole. Static numerical requirements may include the following:

1. The number of terminals to be supported;
2. The number of simultaneous users to be supported;
3. Amount and type of information to be handled.

Static numerical requirements are sometimes identified under a separate section entitled Capacity. Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.

All of these requirements should be stated in measurable terms.

For example:

“95% of the transactions shall be processed in less than 1 s.” rather than, “An operator shall not have to wait for the transaction to complete.”

NOTE: Numerical limits applied to one specific function are normally specified as part of the processing subparagraph description of that function.

### Design constraints

This should specify design constraints that can be imposed by other standards, hardware limitations, etc.

#### Standards compliance

This subsection should specify the requirements derived from existing standards or regulations. They may include the following:

1. Report format;
2. Data naming;
3. Accounting procedures;

Audit tracing.

For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values.

### Software system attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes be specified so that their achievement can be objectively verified. Subsections provide a partial list of examples.

#### Reliability

This should specify the factors required to establish the required reliability of the software system at time of delivery.

#### Availability

This should specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart.

#### Security

This should specify the factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to

1. Utilize certain cryptographical techniques;
2. Keep specific log or history data sets;
3. Assign certain functions to different modules;
4. Restrict communications between some areas of the program;
5. Check data integrity for critical variables.

#### Maintainability

This should specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc. Requirements should not be placed here just because they are thought to be good design practices.

#### Portability

This should specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include the following:

1. Percentage of components with host-dependent code;
2. Percentage of code that is host dependent;
3. Use of a proven portable language;
4. Use of a particular compiler or language subset;
5. Use of a particular operating system.

### Other Requirements

Describe any other requirements. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

# Software Design Description

## Introduction

###  Purpose

Describe purpose of this section. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### Scope

Describe scope of this section. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

###  Glossary

Example glossary for SDD.

Table 2 Glossary of SDD

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Alum | Graduate of Jacksonville State UniversityComputing and Information Sciences Department |
| BDE | Borland Database Engine |
| CI | Configuration Item |
| CIS | Computing and Information Sciences Department |
| Html | Hyper text markup language |
| IEEE | Institute of Electrical and Electronic Engineers |
| QA | Quality assurance |
| SCMP | Software Configuration Management Plan |
| SDD | Software Design Document |
| SEI | Software Engineering Institute, Pittsburgh, Pa |
| SQAP | Software Quality Assurance Plan |
| SRS | Software Requirement Specification |
| Tbd | To be decided |
| Tbn | To be named |
| Web Site | A place on the world wide web |

### References

References for your SDD document.

### Overview of document

Give an overview of SDD document. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

## Deployment diagram

Give a deployment diagram (figures, tables etc.) for your project process. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

## Architecture design (You can use your sections and headers)

Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### SDD subsections here…

Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### SDD subsections here…

Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

## Data structure design (You can use your sections and headers)

Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Example DB table:

Table 3 Data Field Type

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Attribute Type** | **Attribute Size** |
| LastName | String | 30 |
| FirstName | String | 30 |
| MaidenName | String | 30 |
| Address1 | String | 50 |
| Address2 | String | 50 |
| City | String | 30 |
| State | String | 2 |
| Zip | Int | 6 |
| Year | Int | 4 |
| AdditionalDegrees | String | 50 |
| Spouse | String | 30 |
| Children | String | 50 |
| CurrentEmployment | String | 50 |
| ReceiveEmails | Boolean | 1 |
|  |  |  |

## Use case realizations

Describe use case realizations and provide figures. Use this style for the paragraph. Sample figure:



Figure 1 System Sequence Diagram

Always cite your figures, table inside the text.

### Use Case: Use Case 1 (Add use cases here)

Refer to use cases provided in SRS. Provide and cite figures…



Figure 2 Survey Sequence Diagram

### Use Case: Use Case 2 (Add use cases here) and so on…

Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

## Interface design

Illustrate the interface design, provide figures, screenshots. Cite and explain each figure. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.



Figure 3 Home page



Figure 4 Home page

Provide other screenshots.

##  Help system design

Describe help system / user manuals for your product.

## Index

Borland Database Engine, 13

Configuration Item, 13

Database, 13

Institute of Electrical & Electronic Engineers, 13

Quality Assurance, 13, 14

Software Configuration Management Plan, 13

Software Design Document, 13

Software Engineering Institute, 13

Software Quality Assurance Plan, 14

Software Requirement Document, 14

System, 18

Use Case, 16

# Test Plan

## Introduction

### Version Control

Table 4 Version Control

|  |  |  |
| --- | --- | --- |
| **Version No** | **Description of Changes** | **Date** |
| 1.0 | First Version | May 22, 2010 |
|  |  |  |
|  |  |  |

### Overview

What is to be tested? Describe in 2-3 sentences

### Scope

What is the scope? Describe what this document will include.

### Terminology

Table 5 Terminology

| **Acronym** | **Definition** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

## Features to be tested

This section lists and gives a brief description of all the major features to be tested. For each major feature there will be a Test Design Specification added at the end of this document.

### Login (LG) << Specify the name and acronym of the feature here

Describe the feature in 2-3 sentences

### Add User (AU) << Specify the name and acronym of the feature here

Describe the feature in 2-3 sentences

## Features not to be testes

List all the features not to be tested and explain why?

## Item pass / fail criteria

Describe the general rule to use to decide when a test case passes and when it fails.

### Exit Criteria

Describe under what conditions the testing of the product is considered successful. Some examples are:

* 100% of the test cases are executed
* 95% of the test cases passed
* All High and Medium Priority test cases passed

## References

1. IS502\_Group1\_SRS\_V2.0, December 12, 2009 *<< Give reference to your SRS / SDD documents*

## Test design specifications

### Login (LG) (Features to be tested)

#### Subfeatures to be tested

1. Admin (LG.AD)

Describe in 1-2 sentence this subfeature.

1. Gest (LG.GT)

Describe in 1-2 sentence this subfeature.

#### Test Cases

Here list all the related test cases for this feature

Table 6 Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **TC ID** | **Requirements** | **Priority** | **Scenario Description** |
| *LG.AD.xx* | *Give corresponding requirement no* | *High or Medium or Low* | *A brief description* |
| LG.AD.01 | 3.1 | H | Enter a valid admin user id and password |
| LG.AD.02 | 3.1 | H | Enter a valid admin user id and blank password |
| … |  |  |  |
|  |  |  |  |

### Add User (AU) (Features to be tested)

#### Subfeatures to be tested

1. Admin (AU.AD)

Describe in 1-2 sentence this subfeature.

1. Guest (AU.GT)

Describe in 1-2 sentence this subfeature.

#### Test Cases

Here list all the related test cases for this feature

Table 7 Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **TC ID** | **Requirements** | **Priority** | **Scenario Description** |
| AU.AD.01 | 3.2 | H | Add an admin user who is not already in db |
| AU.AD.02 | 3.3 | H | Add an admin user who already exist |
| … |  |  |  |
|  |  |  |  |

##

## Detailed Test Cases

### LG.AD.01

Table 8 LG.AD.01

|  |  |
| --- | --- |
| **TC\_ID** | LG.AD.01 |
| **Purpose** | Enter a valid admin user id and password |
| **Requirements** | 3.1 |
| **Priority** | High. |
| **Estimated Time Needed** | 5 Minutes |
| **Dependency** | Add User test cases should pass |
| **Setup** | An admin user should be created. |
| **Procedure** |  [A01] Go to login page. |
| [A02] Enter a valid admin user id. |
|  [A03] Enter the valid password for this user |
|  [A04] Click on the “Login” button. |
|  [V01] Observe that the login is successful and the admin page appears |
|  - |
| **Cleanup** | Logout |

### LG.AD.02

…

# Test Results

## Individual Test Results

Table 9 Individual Test Results

| **TC ID** | **Priority** | **Date Run** | **Run By** | **Result** | **Explanation** |
| --- | --- | --- | --- | --- | --- |
| LG.AD.01 | H | 25.04.2016 | Ahmet Nuryüzlü | Pass |  |
| LG.AD.02 | H | 25.04.2016 | Zeki Bahtiyar | Fail | This problem will be fixed in the next release |
| … |  |  |  |  |  |
|  |  |  |  |  |  |

## Summary of Test Results

Table 10 Test Results

| **Priority** | **Number of TCs** | **Executed** | **Passed** |
| --- | --- | --- | --- |
| H | 25 | 25 | 24 |
| M | 60 | 60 | 58 |
| L | 70 | 70 | 69 |
| **Total** | **155** | **155** | **151** |

Describe here if the exit criteria met or not.

## Exit Criteria

Describe here if the exit criteria met or not.

Table 11 Exit Criteria

|  |  |
| --- | --- |
| **Criteria** | **Met or Not** |
| 100% of the test cases are executed | Y |
| 95% of the test cases passed | Y |
| All High and Medium Priority test cases passed | N |

## Known Problems

Describe here any known problems.

## Conclusion

In this section describe the decision whether the product is ready for use or not.

# Conclusions

State what you learned from your work. Sometimes this section is labeled Conclusion, Summary or Concluding Remarks. In this section:

* Summarize what you did. This can be viewed as the evidence.
* State what you learned (the actual conclusions that you a drawing).
* State future work and directions, and then list any open problems.

List the advantages and disadvantages of your work. In what ways is your solution deficient or lacking? You are not divulging a weakness in your work when you state problems that still remain. You may also have a separate section for future work.

# Acknowledgement

Acknowledge any individuals who have helped you during the course of the project, including your project advisor. If you have been supported by a company or a scholarship, then this should also be gratefully acknowledged.

# References

All senior projects, especially research-oriented projects, should include at least a few references to the literature. You can choose any style, but be consistent and complete in your reference list. You should pick a particular style such as one promoted by a technical society (ACM, IEEE, SPIE, API, or AIP).

# Appendices

Include brief code fragments, and illustrations of results that are too verbose for the body of your report. Reference these in the body of your report. If you have a lot of raw data that you think might be interesting, you can place examples of that here as well.