

Design Document

Version X.Y – yyyy.mm.dd

Created yyyy.mm.dd

Project Name

Name 1 (role-if any)

Name 2(role-if any)

...

Name N(role-if any)

Table of Contents

1	INTRODUCTION.....	3
1.1	DOCUMENT SCOPE	3
1.2	INTENDED AUDIENCE	3
1.3	PROJECT OVERVIEW.....	3
2	SYSTEM REQUIREMENTS	3
3	DESIGN SPECIFICATION.....	4
3.1	DESIGN PRINCIPLES/GUIDELINES	4
3.2	COMPONENTS.....	4
3.3	STATIC STRUCTURE	4
3.4	OBJECT INTERACTION	4
4	APPENDIX.....	4
5	GLOSSARY	5

1 Introduction

1.1 Document Scope

Here you should describe what the purpose of this document is, and what it is not (e.g. it does not describe how a user uses the system.) No-one likes to have to read a few pages to figure out if they're looking at the right document.

1.2 Intended audience

This section should describe the stakeholders you are targeting this document at. This is another way to let people quickly discover whether they're reading the wrong document. Hint: in your current assignment, this is not the end-user.

1.3 Project Overview

What is it that you're building? Cross references or links to other documents may be relevant. This is a non-technical overview to provide the right context for understanding the following technical sections.

2 System Requirements

You've already written a requirements document which details what features that your system must have. This is most likely written in terms of what must be available to the end-user via the user interface, or possibly, written to describe the environment in which the completed application must run. You've also written a User's Manual.

In this section, I'd like you to try to re-state the requirements you've identified, and the behavior you've specified in the User's Manual, as requirements on the software that you're designing. For example, if your GUI has several *modes* (e.g. selecting a route, walking a route, publishing photos) you may wish to describe what actions your software must perform in response to, say, a particular button click when in a particular mode. Where possible you should cross reference the system requirement to the item that caused it to be generated (i.e. a behavior specified in the UM, or an item in the Requirements Document), and numbers should be given to each system requirement.

E.g.

SR1. When in route-overview mode, pressing the right-arrow button will scroll the map panel (see Figure X for the window layout) so that the next waypoint on the route is centered in the window. The info panel will also be updated to display the current waypoint's name, and distance. X-Ref: UM-UseCase14

SR2. When in walking mode, pressing the up-arrow button will load the next available waypoint image into the photo panel. Once all available images are exhausted, it will cycle back to the first image. X-Ref: ReqDoc-1.2, UM-2.3

Note that these example requirements are expressed in terms of how the software elements (in these examples, GUI panels) must be manipulated to implement the user

requirements. Feel free to deviate from the format I've given. E.g. add subsections if they help the organization, etc.

3 Design Specification

3.1 Design Principles/Guidelines

Describe how you have approached any complicated parts of the design. For example, have you abstracted out a generic DatabaseInterface class so that changing the database from MySQL to Oracle is simply a matter of rewriting the DatabaseInterface class? Have you left interfaces for future expansion, 3rd party additions, etc.? Have you required that all data items being transferred to your application are tagged with a format version number in case future releases of the application change the transmission format?

These are not necessarily things that you should do, but it gives an example of the type of “design decisions” that belong in this section. Each decision should be accompanied by the reason it was taken.

3.2 Components

The idea here is describe any logical divisions that you have made in your project, whether for deployment (e.g. server, client) or for logical convenience (GUI, backend logic, data access). You should describe the purposes that the components fulfill in addition to any graphical representation. I would suggest UML Package diagrams. I would also suggest numbered subsections for the components. (i.e. “3.3.3 The ABC Component”)

3.3 Static Structure

Here you will talk about the Classes which make up your design. You should cover the reason that each class exists (i.e. its role in its component.) I expect that there will most likely be a UML static structure diagram per component in 3.2, which would probably be nicely arranged into numbered subsections below. (i.e. “3.3.3 The ABC Component”)

3.4 Object Interaction

Here is where you tie it all together. For each System Requirement in Section 2, give a description of how the objects interact to achieve the required result. If several System Requirements are similar, discuss them together (as a group) to avoid redundancy. For each group of System Requirements with different behavior, you should illustrate the interaction using a UML Interaction Diagram (either Sequence Diagrams or Communication Diagrams.) Any small differences from the illustration by a similar System Requirement in the group can be addressed in the text. Again, I would suggest putting one group per numbered subsection below.

4 Appendix

Add any additional info you would like to refer to here. For example, in Section 2, I referred to a diagram of the window layout. A subsection of the appendix would be an ideal place for such a figure to appear.

5 Glossary

If you've defined your own terms to make it easier to reference parts of your design, User's Manual, or Requirements Document, build a sorted list of definitions here. This includes acronyms such as UM for Users' Manual, etc.