|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | ucy_logo.gif | UNIVERSITY OF CYPRUS  **DEPARTMENT OF COMPUTER SCIENCE** | cs_logo.png | |
| EPL 343 SOFTWARE ENGINEERING **Specifications Document** System NameMember 1Member 2Member 3Member 4Member 5Member 6 <DD.MM.YYYY> |

Table of Contents

[1. Introduction 4](#_Toc525123694)

[1.1 Purpose 4](#_Toc525123695)

[1.2 Scope 4](#_Toc525123696)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc525123697)

[1.4 References 4](#_Toc525123698)

[2. Data store 5](#_Toc525123699)

[*2.1* *Data* definition 5](#_Toc525123700)

[2.2 Size calculation 5](#_Toc525123701)

[3. Use Cases 6](#_Toc525123702)

[3.1 Actors 6](#_Toc525123703)

[3.1.1 Actors diagram 6](#_Toc525123704)

[3.1.2 Actor descriptions 6](#_Toc525123705)

[3.2 Use Case Descriptions 6](#_Toc525123706)

[3.2.1 <*Name of Use Case 1*> 6](#_Toc525123707)

[4. Domain Model Class Diagram 8](#_Toc525123708)

[5. Activities Plan 9](#_Toc525123709)

[5.1 Development Phases 9](#_Toc525123710)

[5.2 Activity Plan 9](#_Toc525123711)

[5.2.1 Milestones 9](#_Toc525123712)

[5.2.2 Deliverables 9](#_Toc525123713)

[5.2.3 Time plan 9](#_Toc525123714)

[6. Appendices 10](#_Toc525123715)

Revision Chart

This chart contains a history of this document’s revisions. The entries below are provided solely for purposes of illustration. Entries should be deleted until the revision they refer to has actually been created.

The document itself should be stored in revision control, and a brief description of each version should be entered in the revision control system. That brief description can be repeated in this section. Revisions do not need to be described elsewhere in the document except inasmuch as they explain the development plan itself.

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| Draft | TBD | Initial draft created for distribution and review comments | TBD |
| Preliminary | TBD | Second draft incorporating initial review comments, distributed for final review | TBD |
| Final | TBD | First complete draft, which is placed under change control | TBD |
| Revision 1 | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| Revision 2 | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| etc. | TBD | TBD | TBD |

# Introduction

*This section should provide an overview of the entire document. No text is necessary between the heading above and the heading below unless otherwise desired.*

## Purpose

*Describe the purpose of this specification and its intended audience.*

## Scope

*Identify the software product(s) to be produced by name. Explain what the products will and will not do. Describe how the software will be used, and identify relevant benefits, objectives, and goals.*

## Definitions, Acronyms, and Abbreviations

*Define all terms, acronyms, and abbreviations used in this document.*

## References

*List all the documents and other materials referenced in this document. This section is like the bibliography in a published book.*

# Data store

*In this section you should describe the elements that need to be stored in the system databases and the requirements in space allocation.*

## *Data* definition

*Describe each data base element and its type including the exact information that needs to be stored for each element.*

## Size calculation

*Define the total size of data required to be stored for the system. You can use a tabular format containing the element studied and the size needed (e.g. in Kbytes). For each description please take into account the data that needs to be inserted to the system initially and the data that will be added during the system operation. Make a calculation for the total size of the data store.*

# Use Cases

## Actors

*An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actors that will be participating in the use cases.*

### Actors diagram

*Describe in one or two paragraphs the environment in which the actors exist. Introduce the diagram by explaining it in words. If necessary, insert an actor inheritance diagram that illustrates the relationships that exist between the various actor types.*

### Actor descriptions

*Give a description of each actor in a tabular format. Each table should include the: actor name, actor aliases, actor description and any relations with other actors. Repeat the section below for each actor.*

|  |  |
| --- | --- |
| **Description** | *Brief description of the actor and its role in the system. This description should be no more than a small paragraph and should give the reader an understanding of the role of the actor in the organisation. Mandatory.* |
| **Aliases** | *Any other names by which this actor may be known. A simple list is sufficient. Say ‘None’ if there are none.* |
| **Inherits** | *The ancestors for the actor. Some actors may be specialised types of other actors. A simple list of the names of the ancestors will suffice. Say ‘None’ if there are none.* |
| **Actor Type** | *Active/Passive - Person/External System. Mandatory.* |

## Use Case Descriptions

### <*Name of Use Case 1*>

*Use a representative name. Give a description and a diagram of each use case. You should provide the following information for each use case:*

* *Diagram*
* *Description*
* *Actors: List the actor names that perform the scenario.*
* *Precondition: Specify any conditions that must be met before this scenario can be performed.*

*Relate each use case with a user story if possible. A one-to-one mapping will not be possible in some cases. Do not confuse the two (see this article: http://www.batimes.com/articles/user-stories-and-use-cases-dont-use-both.html).*

# Domain Model Class Diagram

*A Domain Model Class Diagram is a high level object model of the classes that exist in the business domain supported with a general statement about each Class covered within the model.*

*Insert a copy of the class diagram here. Complex systems may need to be partitioned into Subject Areas with each subject area containing a subset of the classes (e.g. in a telecommunication system one subject area may refer to the billing system and another one to the customer care system).*

# Activities Plan

*In this Section provide a Gantt diagram of the tasks, tests, and activities you need to perform to complete the project as a whole (including full implementation and testing).*

## Development Phases

*Describe the lifecycle used and the phases of the project (e.g. design, implementation etc.). Given an overview of the activities contained in each phase*

In our project we are using the waterfall model. By definition the waterfall model will have the following main phases:

* Requirement Engineering
* Design
* Implementation
* Testing
* Maintenance

The phases will be executed in the order they are listed above after Validation and Verification of each phase.

## Activity Plan

### Milestones

* Start of Software Engineering Requirements phase
  + First meeting with Signal Generix as our client
  + First interview with stakeholders
  + Second meeting with client to take feedback for software requirements specification document
  + Acquire the sensors from Signal Generix
* Start of Software Specifications phase
  + Third meeting with client to take feedback for specifications document
* Start of Design phase
  + Fourth meeting with client to take feedback for design document
* Start of Implementation phase
  + Database installation
  + Graphical user interface finalization
  + Implementation of features
  + First working prototype
  + Fifth meeting with client for feedback
* Start of Testing phase
  + Meeting with stakeholder for feedback
  + First flight test
* Start of Maintenance phase
  + Complete refinement
* Final Product Release

### Deliverables

* First deliverable
  + Software Engineering Requirements document
  + Video with client/stakeholders describing the problem that our software will resolve(2-4 minutes)
  + Instance of assignment activities
* Second deliverable
  + Specifications document
  + Project’s timetable
  + Use case diagrams
  + Instance of assignment activities
* Third deliverable
  + Design document
  + UML diagrams
  + Instance of assignment activities
* Forth deliverable (prototype)
  + Source code
  + SQL script
  + Client’s feedback and comments
  + Instance of assignment activities
* Fifth deliverable (final)
  + Source code
  + SQL script
  + Instance of assignment activities

### Time plan

*Give a time overview of the activities that will be performed including their duration and the resources allocated in each activity (e.g. team members).This should have the following format:*

|  |  |  |
| --- | --- | --- |
| **Activity** | **Duration** | **Resources** |
| <Activity Name and short description> | <Activity duration> | <Team members involved> |
| … | … | … |

*Provide also the Gantt diagram corresponding to the activities here. Indicate the critical path of your project.*

# Appendices

*Include supporting detail that would be too distracting to include in the main body of the document.*

*Screens of the system prototype should be added here, although the prototype will also be delivered separately from this document. The system prototype shows how the interface will look for the end-user. In this section you should provide screenshots of the final system. You should also include specifications for the input and output needed in each form (description of data flows).*