<Software Requirements> Specification

for

<Project Name>

Prepared by

<Group Member1>

<Group Member2>

<Group Member3>

<Group Name>

<date created>

<Advisor>

The project titled "This Part Will Be Filled in Bold and Word Heads with Capital Letters" prepared by Name Surname. has been accepted as a GRADUATION PROJECT in Ankara Science University, <DEPARTMENT Name> Engineering Dept. by the following jury with VOTE UNANIMOUS / VOTE MAJORITY.

Advisor: ………………………………….

<DEPARTMENT Name> Engineering Department

I confirm that this report is a Graduation Project in scope and quality.

Signature: …………………….

Member: ………………………………….

<DEPARTMENT Name> Engineering Department

I confirm that this report is a Graduation Project in scope and quality.

Signature: …………………….

Member: ………………………………….

<DEPARTMENT Name> Engineering Department

I confirm that this report is a Graduation Project in scope and quality.

Signature: …………………….

I confirm that this project, accepted by the jury, fulfills the requirements for being a Graduation Project, part 1. ......../….…/……

…………………

Prof./Assoc.Dr….

Head of <DEPARTMENT Name>

**CONFLICT OF INTEREST**

In this project study that we prepared in accordance with Ankara Science University Engineering Faculty Project Writing Rules.

We declare that;

* We have obtained the data, information, and documents are presented in the project report within the framework of academic and ethical rules,
* We have presented all information, documents, evaluations, and results in accordance with the rules of scientific ethics and morality,
* We have cited all the works I have benefited from in the project study with appropriate attribution,
* We have not made any changes to the data used,
* This report is original and has not been presented elsewhere before,
* We have prepared within the framework of workplace training within the scope of the studies and observations,

We, all group members, hereby declare that we accept all loss of rights that may arise against us in any other case declared above. ......../….…/……

Name Surname (1st Student) Signature………..…

Name Surname (2nd Student) Signature……….…

Name Surname (3rd Student) Signature……….…

Table of Contents

Table of Contents ii

1. Introduction 1

1.1 Purpose 1

1.2 Project Scope 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 Object Design ………………………………………………………………………………………………………3

2.4 User Classes and Characteristics 5

2.5 Operating Environment 6

2.6 Design and Implementation Constraints 6

2.7 Assumptions and Dependencies 6

3. System Features 7

3.1 System Feature 1 7

3.2 System Feature 2 (and so on) 7

4. External Interface Requirements 8

4.1 User Interfaces 8

4.2 Hardware Interfaces 9

4.3 Software Interfaces 9

4.4 Communications Interfaces 9

Appendix A: Glossary .12

Appendix B: Analysis Models 13

Appendix C: Issues List 14

# Introduction

## Purpose

<Identify the product whose software requirements are specified in this document. Describe the scope of the product that is covered by this SRS.>

## Project Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. An SRS that specifies the next release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision.>

## References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product.>

## Product Features

<Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram, is often effective.>

## Object Design

<Establish the relationships and identify the hierarchies between classes. In addition, create the internal details of the classes and their relationships, such as the data structure for each attribute and the algorithms for the operations. >

## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.>

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

3.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

3.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Object & System Design

* An Overview of Analysis 174
* Analysis Concepts
* Analysis Object Models and Dynamic Models
* Entity, Boundary, and Control Objects
* Generalization and Specialization

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

**5.4 Analysis Activities: From Use Cases to Objects**

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

* Mapping Use Cases to Objects with Sequence Diagrams
* Modeling Interactions among Objects with CRC Cards
* Identifying Associations
* Identifying Aggregates
* Identifying Attributes

## Manage Analysis

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

* Modeling State-Dependent Behavior of Individual Objects
* Modeling Inheritance Relationships between Objects
* Reviewing the Analysis Model
* Analysis Summary

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

* Documenting Analysis
* Assigning Responsibilities
* Communicating about Analysis
* Iterating over the Analysis Model

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>