<Project Name>

Software Requirements Specification

<Date>

<Group Members>

Prepared for

CSE3044 Software Engineering Term Project

**Table of Contents**

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. General Description 4

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Characteristics 4

2.4 General Constraints 4

2.5 Assumptions and Dependencies 4

3. Specific Requirements 4

3.1 External Interface Requirements 5

3.1.1 User Interfaces 5

3.1.2 Hardware Interfaces 5

3.1.3 Software Interfaces 5

3.1.4 Communications Interfaces 5

3.2 Functional Requirements 5

3.2.1 <Functional Requirement or Feature #1> 5

3.2.2 <Functional Requirement or Feature #2> 5

3.3 Non-Functional Requirements 5

3.3.1 Performance 6

3.3.2 Reliability 6

3.3.3 Availability 6

3.3.4 Security 6

3.3.5 Maintainability 6

3.3.6 Portability 6

3.4 Inverse Requirements 6

3.5 Design Constraints 6

3.6 Logical Database Requirements 6

3.7 Other Requirements 6

4. UML Diagrams 6

4.1 Use Cases 6

4.1.1 Use Case #1 6

4.1.2 Use Case #2 6

4.2 Classes / Objects 6

4.2.1 <Class / Object #1> 6

4.2.2 <Class / Object #2> 6

4.3 Sequence Diagrams 7

4.4 Data Flow Diagrams (DFD) 7

4.5 State-Transition Diagrams (STD) 7

A. Appendices 7

A.1 Appendix 1 7

A.2 Appendix 2 7

# 1. Introduction

The introduction to the Software Requirement Specification (SRS) document should provide an overview of the complete SRS document. While writing this document please remember that this document should contain all of the information needed by a software engineer to adequately design and implement the software product described by the requirements listed in this document.

(Note: the following subsection annotates are largely taken from the IEEE Guide to SRS).

## 1.1 Purpose

Defining and describing the functions and specifications of the Petiverse mobile application is the primary goal of this Software Requirements Specification.This Software Requirements Specification illustrates, in clear terms, the system’s primary uses and required functionality as specified by our customer. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

## 1.2 Scope

*This subsection should:*

The software system being produced is called Petiverse. It is being produced for term project of Marmara University CSE3044 lecture. This system is designed to provide a social network among animal lovers and pet owners to support animals in need and create a network to share and learn new informations.

The Petiverse mobile application will allow any user to create an account to become a member. This creation process optionally includes sub creation of pet profile which users have. The system will allow members to browse, search, filter and share new posts to the network. The system will have different type of post to be posted which are pet friend, missing ads, animal in need (with pictures and location). That way users will be able to communicate and aid each other easily. The forum (blog) side of the system will have different categories such as type of animal, topics etc. that way users will be capable of find what they need more quickly.

Constraints ??

*(1) Identify the software product(s) to be produced by name; for example, 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. As a portion of this, it should:*

*(a) Describe all relevant benefits, objectives, and goals as precisely as possible. For example, to say that one goal is to provide effective reporting capabilities is not as good as saying parameter-driven, user-definable reports with a 2 h turnaround and on-line entry of user parameters.*

*(b) Be consistent with similar statements in higher-level specifications (for example, the System Requirement Specification), if they exist. What is the scope of this software product?*

## 1.3 Definitions, Acronyms, and Abbreviations

*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.*

**1.3.1 User Account**

A user account is allocation on a network server used to store a computer username, password, and other information. A user account allows or does not allow a user to connect to a network, another computer, or other share. Any network that has multiple users requires user accounts.

**1.3.2 Mobile Operating System**

A mobile operating system, also called a mobile OS, is an operating system that is specifically designed to run on mobile devices such as mobile phones, smartphones, tablet computers and other handheld devices.

**1.3.3 Database**

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, database can be classified according to types of content**1.3.4 Flutter**

**1.3.4 Browse**

**1.3.4 Search**

## 1.4 References

*This subsection should:*

*(1) Provide a complete list of all documents referenced elsewhere in the SRS, or in a separate, specified document.*

*(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.*

## 1.5 Overview

# This Software Requirements Specification document is divided in to multiple subsections. The first section includes explanations of the Purpose, Scope and Organization of the document. The first section also handles the description of project specific words, acronyms and abbreviations that will be used in the document. The second section of the document is separated into the following five different sections, each detailing specific details of system uses and their corresponding actions: Product Perspective, Product Functions, User Characteristics, Constraints, Assumptions and Dependencies, Apportioning of Requirements. The third section is an enumerated listing of all of the requirements described for this system. The fourth section encompasses all of the Use-case, Sequence, State and Class diagrams that model the system. In the fifth section there exists a Prototype of the system along with a sample scenario that graphically describes the use of the system. The sixth section contains a listing of all related reference materials used in this document. The seventh and final subsection is dedicated to providing a point of contact for any viewer of this document.

## 2. General Description

This section includes details about what is and is not expected of the Petiverse Mobile Application in addition to which cases are intentionally unsupported and assumptions that will be used in the creation of the application.

*This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.*

## 2.1 Product Perspective

Petiverse is a mobile application which supports a number of functions for users of the app.

The mobile application must be available to anyone using a mobile phone and interested in animals. As stated by the customer, there are no hardware or software limitations that will be used.

*This subsection of the SRS puts the product into perspective with other related products or*

*projects. (See the IEEE Guide to SRS for more details).*

## 2.2 Product Functions

This subsection of the SRS should provide a summary of the functions that the software will perform.

## 2.3 User Characteristics

The typical Petiverse user is simply by anyone that has access to a smart phone (For now its just for Android users but it will be discussed later for other mobile operating systems). It is assumed that the user is familiar enough with operating a smart phone.

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details).

## 2.4 General Constraints

Security concerns ? passwords encryption etc.

*This subsection of the SRS should provide a general description of any other items that will*

*limit the developer’s options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints).*

## 2.5 Assumptions and Dependencies

Assumptions- it is things we assume are in place which contribute to the success of the project.

Dependencies- other projects or triggers that your projects depend on or are a beneficiary of out projects.

Project planning and risk management, teamwork, focusing management attention, and generating successful project outcome is the key to the assumptions and dependencies. We define software assumption as software development knowledge taken for granted or accepted as true without evidence.

Client:

We have assumed that all of the user’s mobile phones are in proper working condition and that the user is capable of operating these phones’s basic functions including but not limited to being able to login using application.

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 might 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.

# 3. Specific Requirements

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.

Each requirement in this section should be:

* Correct
* Traceable (both forward and backward to prior/future artifacts)
* Unambiguous
* Verifiable (i.e., testable)
* Prioritized (with respect to importance and/or stability)
* Complete
* Consistent
* Uniquely identifiable (usually via numbering like 3.4.5.6)

Attention should be paid to the carefully organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

The user interface for the software shall be compatible to any android smart phone that has Internet connection which user can access to the system. Because this is a mobile application, the user interface design ought to be highly user-friendly. This mobile based application would have a user-friendly interface providing the users with an easy-to-navigate and convenient social experience.

### 3.1.2 Hardware Interfaces

Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g Modem, WAN-LAN.

(Android system specifications, requirements)

### 3.1.3 Software Interfaces

Divide into to parts

Front-End Technologies

Back-End Technologies

Operating systems

Database

Coding Language

Framework

Editor IDE

### 3.1.4 Communications Interfaces

## 3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

First, our projects will include the sign-up option for the users. Every user can sign up using their phone number as well as their email address. But if they are guest visitors, they can also visit (Ana sayfa postların olduğu yer ??). And they can also search, filter and navigate among posts. But in this case, they will not get fully experience the futures that our application offers such as creating pet profile, matching similar pets to be friend etc. So it will be better to sign up before dive into out social network, that way users can get specific notifications which matches with their interests. To create and share a new post users must be signed up. Sign-up process and verification ???

### 3.2.5 Browse Feature

3.2.5.1 Introduction / Description: This feature enables user to browse in corresponding page among posts

3.2.5.2 Inputs / Display: Vertical swipe movements

3.2.5.3 Processing: When user swipes up, this gesture is validated and finally brings past posts to the screen. When user swipes down, this gesture is validated and finally brings new posts to screen.

3.2.5.4 Outputs: View new or past posts.

3.2.5.5 Constraints: If user at the top of the page swipe down gesture will finalize. Also if user is at the bottom of the page swipe up gesture will not finalize.

3.2.5.6 Error/Data Handling: If corresponding constraint is not satisfied an error message will be shown such as “There does not exist new post”

### 3.2.6 Animal Help

3.2.6.1 Introduction / Description: This feature enables user to navigate animal help page. Users can see nearby animals in need in this page.

3.2.6.2 Inputs / Display: User clicks animal help page button on navigation bar.

3.2.6.3 Processing: When user clicks the button it will navigate to page and load the existed animal help posts.

3.2.6.4 Outputs: View animal help page.

3.2.6.5 Constraints:

3.2.6.6 Error/Data Handling:

### 3.2.2 <Functional Requirement or Feature #2>

…

## 3.3 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

### 3.3.1 Performance

Application must be lightweight and must send load posts instantly.

### 3.3.2 Reliability

It is very important that the app is reliable for users use Petiverse simultaneously. All data collected by Petiverse shall be preserved safely and should follow data hiding.

### 3.3.3 Availability

### 3.3.4 Security

Keep your password safe and don’t sare it with any other people, applications, or websites under any circumstances. We also suggest using a different password for every service you use. Besides from as Petiverse, we will take all the important measure to make out application safe for our users.

### 3.3.5 Maintainability

Petiverse is a new developing app and hence has maintenance on loop. Petiverse will release all new updates first and looks for acceptance from its customers. Petiverse aims to surprise its users by releasing fresh updates with new features.

### 3.3.6 Portability

Petiverse can be used on any mobile device operating on android.

Permissions can be added ??

## 3.4 Inverse Requirements

They explain what the **system shall not do**. State any \*useful\* inverse requirements.

## 3.5 Design Constraints

Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

Login id & password of user must be valid.

Signup details – mandatory valid details must be provided by user.

New accounts with existing account details cannot be created.

Sufficient memory must be available in order to browse and load posts.

Internet availability.

Location of a user must be provided if users wants to share post with location.

OS required: ?

User must agree to all terms and conditions lay by Petiverse if not user cannot use the application.

## 3.6 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

## 3.7 Other Requirements

Catchall section for any additional requirements.

# 4. UML Diagrams

## 4.1 Use Cases

### 4.1.1 Use Case #1

### 4.1.2 Use Case #2

…

## 4.2 Classes / Objects

### 4.2.1 <Class / Object #1>

4.2.1.1 Attributes

4.2.1.2 Functions

<Reference to functional requirements and/or use cases>

### 4.2.2 <Class / Object #2>

…

## 4.3 Sequence Diagrams

## 4.4 Data Flow Diagrams (DFD)

## 4.5 State-Transition Diagrams (STD)

# A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## A.1 Appendix 1

## A.2 Appendix 2

Acknowledgment: Sections of this document are based upon the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984). The SRS templates of Dr. Orest Pilskalns (WSU, Vancover) and Jack Hagemeister (WSU, Pullman) have also be used as guides in developing this template.