

Software Requirements Specification

for

Animal Database

Version <1.0>

Prepared by

Group Name: Group 20

|  |  |  |
| --- | --- | --- |
| Vinh Duong | 11613642 | vinh.duong@wsu.edu |
| John Stevenson | 011639605 | john.stevenson@wsu.edu |
| Yevgeniy Diriyenko |  | Yevgeniy.diriyenko@wsu.edu |
| Seth Lanante | <student #> | seth.lanante@wsu.edu |
| <name> | <student #> | <e-mail> |

|  |  |
| --- | --- |
|  |  |
| Date: | November 6, 2020 |
|  |  |
|  |  |
|  |  |

Content

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Behavior Requirements 6

4 Other Non-functional Requirements 7

4.1 Performance Requirements 7

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

5 Other Requirements 8

Appendix A – Data Dictionary 9

Appendix B - Group Log 10

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Vinh Duong  John Stevenson  Yevgeniy Diriyenko  Seth Lanante | This is the draft of the SRS. Some Revisions will be made if needed when creating our project. | 11/06/20 |

# Introduction

## Document Purpose

The product of specified in this document is an animal database. This SRS will cover all parts of the product including the data to be maintained/manipulated and the different pages the user will be able to navigate.

## Product Scope

The software being specified is an animal database that users will be able to upload to and interact with the content they made along with other user’s content. The objectives and goals of this product is to allow the user to upload content to the database and interact with the content that they and other users have made. The benefit of this product is purely for entertainment reason by allowing the user to have some fun.

## Intended Audience and Document Overview

The intended audience of this SRS is the professor and kids who would be into these kinds of things. This document is also intended for the developers so that everyone can use this information to build the database. The rest of this SRS contains technical information of the product, what the product needs to do and other considerations like what we will use to make the software. For developers, the suggested reading order of this document will be the product perspective and then the functional requirements. The suggested reading order for the player user who would use this software would be the Use Case View section since it shows visually the functionality and would be the one that pertains most to them. For the professor, the recommended reading is the over sections and then reading the document in order since it explains from the beginning what this software should do.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Browser | Interface for connecting to the product made |
| Database | Where the information for animals will be stored |
| HTTPS | Secure method of communication for web browsers. |
| Software | Set of instructions to implement a functionality. |
| SRS | Software Requirements Specification |

## Document Conventions

This document will follow the IEEE conventions. The font size will be Ariel size 11. Italics will be used for comments. The Document will be single space with a 1” margin. Section will header will start with the section number and the section title. Subsection will be the section number followed by a period and subsection number and then a title.

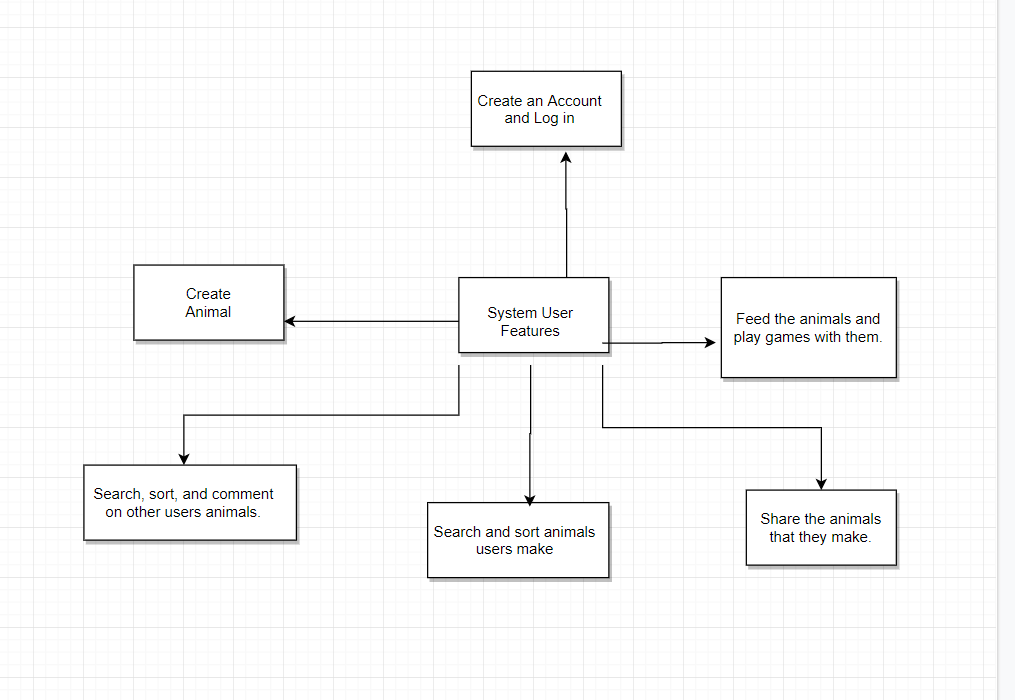
## References and Acknowledgments

We did not cite or paraphrase any other source to the best of our knowledge.

# Overall Description

## Product Perspective

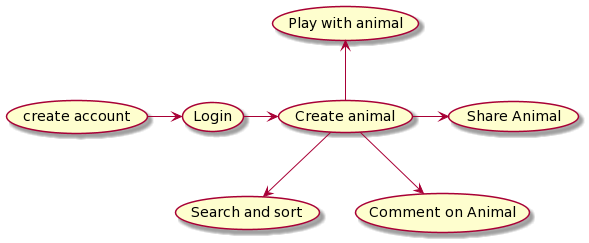
The animal database will be accessible through a web browser. The user will be able to create an account with the animal database website and log in with that account. Once logged in the user will then be able to create an animal through a form they fill out and they are able to upload a picture if they want to. Once they create their animals, the user will be able to sort the animals they have based off of search terms they put in and it can be filtered by different fields like name and animal type. There will be some activity the user can do for fun like feed the pets and play games with them. Additionally, there will be an option to have a shared database where users can share the animals with everyone else who use the database and sort the animals. They will be able to comment on the animals if they choose to.



(Context Diagram)

## Product Functionality

* Allow the user to create an account that they can use to log in.
* The user will be able to create an animal.
* User will be able to share the animals with other users.
* The user can search for and sort the animals they and other users create.
* The user will be able to comment on other user’s animals.
* The user will be able to feed their animals and play games with them.



(Data Flow Diagram)

## Users and Characteristics

**Player user:** Will be most of the user who will use this game to have fun.

## Operating Environment

The software will be hosted on a windows machine and the minimum requirements are the ability to run all the necessary computation and have enough space on the system to hold the software. The software will be tested using Google Chrome.

## Design and Implementation Constraints

A constraint is the security aspect for logging in since we need to make sure that the credentials are secured. Under normal circumstances we would also need to make sure the website is in HTTPS to keep the user credentials secured. We are also going to be constrained by whatever limitation there is for the database. The language constraint we have is that we are going to be implementing the animal database in JavaScript.

## User Documentation

We would need to include an online tutorial with the user manual for the people who are going to be using the database. This is necessary to walk through to the user what they can do with the database and how to use the functionalities of it.

## Assumptions and Dependencies

Some of the assumptions we are making for the user is that they have a web browser that can run JavaScript. We are assuming that the user has the necessary hardware requirements to run any computation we need to do and be able to render any image we give them. We are also assuming that the user has enough space to download our software.

# Specific Requirements

## External Interface Requirements

### User Interfaces

When the user first loads up the software they will be presented with the option to login or make a new account along with the basic forgot password link. After the user log in they will be presented with the home page that describes what the product is and what they can do with it. Along the top of the screen will be the buttons for the different pages that the user can go to. The page to upload a animal will have the different fields that the user can fill like the name, type, and description along with an option to upload a photo. The page to view their animals will have the user’s animals listed out on the screen with a search bar at the top and a drop down menu next to it that allows the user to search the animals by the field and string they type in. The user will then be able to click on the animal to view the animal details. This layout will be similar for the page where the user can see all the animals that everyone has shared. The page that will allow the user to share their animals is similar to the page to view their animals except now there will be checkboxes next to each animals and a share button in the bottom right that will be unclickable until the user checks at least one animal. The feed animal page will is similar to the view animal page except when they click on the animal, they will be taken to a page that mainly features the picture of the animal and a button that will allow the user to feed the animal. The play with animal page will allow the user to pick a animal and then play a game with the picture of the animal off to the side.

### Hardware Interfaces

* **Display Monitor:** A basic monitor that can display images that is shown through the web browser is sufficient for this software.
* **Input Devices:** The user will just need a mouse and keyboard to be able to use our software.

### Software Interfaces

* A database will be used to store all the details of the animals that the user creates. There will be multiple subsets of the database, one for the user’s animals that they made, one for the animals that the user can share, and one for the global shared animals viewable by everyone.
* The web browser will be primary interface for the software with the user sending data through it and receiving it in the form of web pages.
* Will use the underscore function to help streamline part of the code.

### Communications Interfaces

Under normal circumstances HTTPS will be implemented to make sure that the credentials for the user is secured. The browser will be the primary way that the user will communicate with our software. The user will text, user their mouse, and be able to upload files to the web browser. The web browser will then respond appropriately with correspond pages.

## Functional Requirements

**Creating an account:** The user will be able to create an account and log in with it. The user would ideally be able to do a password reset and change the password if they want to. Password will need to be secured in some way.

**Creating an animal:** The user can create an animal to put in their own database. This will take the form of fields they can fill out like the name, description, and type of animal. The user will also be able to upload a picture if they choose to.

**Sharing animals:** The user will be able select any animal that they want to share to a public list that will be viewed by everyone.

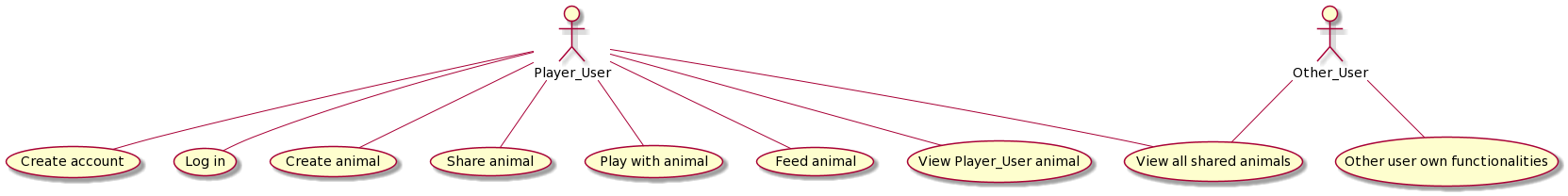
**Viewing animals:** The user will be able to see the list of all their animals and organize the list by the search string they put in corresponding to one of the animal fields like the type of animal or their name. The user can then click on one of the animals to view more information about it. This will work similarly for viewing all the shared animals

**Feeding animals:** The user would be able to select their animals to go feed them by clicking on button to feed them. If they feed their animals a certain amount of times, then the animal will get a star that can be seen when the animal is viewed. The animal will get more stars the more they are fed.

**Playing with animals:** There will be a page where the user will be able to play a game with the picture of the animal they have selected off to the side with text box coming out of it from a pool of pre made phrases.

## Behavior Requirements

### Use Case View



(Use Case Diagram)

# Other Non-functional Requirements

## Performance Requirements

It will not take more than five seconds for any of the main pages to load so that the user will not have to wait long to get what they want. It will not take longer than ten seconds to list all the animals so that the user can quickly look through for the animal they want.

## Safety and Security Requirements

Under ideal circumstances the log in credentials will be encrypted so that it cannot be accessed by unauthorized parties. Also, under ideal circumstances we will be using HTTPS so that the username and passwords won’t be sent over plain text. Some other considerations is if user uploads graphic images or graphic comments, then the developers or moderators will need to go in with a special account that can remove the content and possibly remove the user.

## Software Quality Attributes

4.3.1 Availability:

The software would ideally be available all the time except for some periods of maintenance that happens during times we know that user traffic is low.

4.3.2 Portability:

For portability the software would ideally work on any kind of devices with adjustment being made for the between the screen sizes.

4.3.3 Testability and Usability:

For testability and usability, we would go through all the links and perform every action that we can just to make sure that everything works as it should. We would also have other people go through the software so that they can report bugs and test performance.

# Other Requirements

Don’t have any other requirements.

Appendix A – Data Dictionary

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

|  |  |
| --- | --- |
| Browser | Interface for connecting to the product made |
| Database | Where the information for animals will be stored |
| HTTPS | Secure method of communication for web browsers. |
| Software | Set of instructions to implement a functionality. |
| SRS | Software Requirements Specification |

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>

|  |  |
| --- | --- |
| 10/10/20 | Went over Git hub and tried to get the GitHub desktop app working. Talked about the SRS document. Agreed on a new meeting time. |
| 10/16/20 | Catch up on trying to get the git hub accounts working and worked on the SRS document. |
| 10/24/20 | Had to change communication method for a group member and added them to the git hub repository. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |