Software Requirements Specification

for

Dawg Barks

Version 1.0 approved

Prepared by Group 9

MSU – Intro to SE

Sept. 2022

# Introduction

## Purpose

The product is Dawg Barks v1.0. This document will cover the general specifications of the product.

## Document Conventions

No typographical standards have been decided upon aside from those already implemented by default.

## Intended Audience and Reading

This document is intended for the developers, project manager, and documentation writers of Group 9. It contains documentation on the software requirements of Dawg Barks (possibly both for developers and users). [insert reading notes here].

## Product Scope

Dawg Barks will be a “Facebook-like” social media web application intended for students, faculty, and other such members of Mississippi State University. One of the main objectives is to replicate all of the standard features expected of such an application.

## References

There are currently no other documents associated with this document.

# Overall Description

Thus far, this is a new, self-contained product as it is the first version

## Product Functions

This product will perform functions normally found in certain popular social media sites such as:

* Allowing a user to like another user’s post
* Allowing a user to create posts (mainly text, but potentially also images/videos).
* Allow user to interact with other users
* Allow user to see feed from friends

## User Classes and Characteristics

There are two planned primary user types of Dawg Barks in the form of the standard users and the development users AKA dev users. The standard user base of Dawg Barks will have a large variety of different backgrounds and varying levels of education, but will all have some affiliation with MSU whether that be directly through being a student or faculty member, or more indirectly through being a worker at MSU or an alumni. Standard users will have access to all of the features mentioned in section 2.2 of this document. Members of the development team will have access to the features mentioned in section 2.2, but will also have access to a number of administrative security features.

## Operating Environment

Being a web application, this software will be operating in a server-client environment on a web browser.

## Design and Implementation Constraints

It will be limited to just what can be done on a browser. Any interaction with the users’ system (aside from basic user input) will be kept to a minimum.

# System Features

## URL Routing

4.1.1 Description and Priority

The basic setup for the webpage’s links. Priority: high

4.1.2 Stimulus/Response Sequences

As the user navigates the site, the site will display content according to Python modules with the URL as an input.

4.1.3 Functional Requirements

Req 1: Python modules that take in a URL as input to determine displayed content.

Req 2: A handler for broken/nonexistent URLs.

## Template setup

4.1.1 Description and Priority

The basic setup for the webpage. Priority: medium

4.1.2 Stimulus/Response Sequences

The user sees and interacts with the content of the site and the site responds to those interactions according to which interactions the user performs.

4.1.3 Functional Requirements

Req 1: Basic skeleton of the website elements (possibly in HTML).

Req 2: Integration of functional code with the website elements.

**3.3 Static Files**

4.1.1 Description and Priority

The front-end elements of the webpage. Priority: medium

4.1.2 Stimulus/Response Sequences

The user sees the displayed webpage and all of it’s associated aesthetics according to the design implemented in the front-end markup code.

4.1.3 Functional Requirements

Req 1: Basic skeleton of webpage elements (see Req 1 of feature 3.2).

Req 2: A definition for how those elements and the page itself look (possibly in CSS).

Req 3: Discussed information on how the webpage should look.

# Other Nonfunctional Requirements

## Performance Requirements

It will have to handle multiple users using the site at the same time and will have to hold the data (user info, posts, pictures) of multiple users.

## Safety Requirements

The main safety concerns are in the interactions between different users.

## Security Requirements

All user login information (username/email and password) should be inserted into a database in an encrypted format to reduce the chance of identity theft.

## Software Quality Attributes

Availability and portability are important because it needs to be accessible to the internet. Reliability and robustness are also important because this needs to function relatively autonomously over long periods of time.

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.> None so far.

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

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

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>