
Software Requirements Specification

for

Darkspear

Prepared by:

**Rachel Tidwell
Jeffrey Heredia
Andy Thorsen
Michael Thorsen**

Date Created:

October 25, 2018

Table of Contents

Table of Contents.....	1
Revision History.....	2
1. Introduction.....	2
1.1 Purpose.....	2
1.2 Document Conventions.....	2
1.3 Intended Audience and Reading Suggestions.....	2
1.4 Product Scope.....	3
1.5 References.....	3
2. Overall Description.....	3
2.1 Product Perspective.....	3
2.2 Product Functions.....	4
2.3 User Classes and Characteristics.....	5
2.4 Operating Environment.....	5
2.5 Design and Implementation Constraints.....	5
2.6 User Documentation.....	5
2.7 Assumptions and Dependencies.....	5
3. External Interface Requirements.....	5
3.1 User Interfaces.....	5
3.2 Hardware Interfaces.....	6
3.3 Software Interfaces.....	6
3.4 Communications Interfaces.....	6
4. System Features.....	6
4.1 Clan Search.....	6
4.2 Searching for Other players.....	6
4.3 Chatting with Other Players.....	7
4.4 Friends list.....	7
5. Other Nonfunctional Requirements.....	7
5.1 Performance Requirements.....	7
5.2 Safety Requirements.....	8
5.3 Security Requirements.....	8
5.4 Software Quality Attributes.....	8
5.5 Business Rules.....	8
Appendix.....	9

Revision History

Name	Date	Reason For Changes	Version
Creation	10/25	Creation of SRS	1.0
Major revision	10/31	Adding and update most of the content throughout SRS	1.1

1. Introduction

1.1 Purpose

The following described product is Darkspear Version 1. This SRS will describe the entire system for the product.

1.2 Document Conventions

Each requirement statement in this SRS will have its' own priority. When an item has high significance in the SRS, it will be highlighted.

1.3 Intended Audience and Reading Suggestions

This SRS will contain a description of the product, the external interface requirements[user, hardware, software and communications interfaces], followed by system features, other non-functional requirements(safety, security, software quality, business rules and performance.) Finally the SRS will list any other requirements.

The intended audience for this SRS is current and future developers and marketing persons.

For developers, it is pertinent to read through the entire document from start to finish. Marketing persons should read the introduction, overall description of the product, and system features.

1.4 Product Scope

The software specified is operated inside a GUI that allows a user to join clans based on video game preferences. The application also allows users to chat in a chat box. The application is useful for gamers (entertainment) as it will help to unify gamers based on interest and skill level.

Our objectives with this application are:

1. To gather a user base of 250+, in order for users to get the best of the experience (more users means more already existing clans)
2. To implement a chat feature to allow the users to communicate from within the application
3. To implement a RPG based text game within the application

Our goals:

1. To get a user base of 1000+
2. To make the application usable across all platforms
3. To expand the application globally

1.5 References

We do not yet have any references included in this SRS.

2. Overall Description

2.1 Product Perspective

This product aims to unite gamers across the world for a common cause. Often times gamers are fortunate enough to find a game they truly love, but are unfortunate enough to not know others who are also playing this game. This app aims to help them connect with other gamers who play the same games at a similar difficulty.

2.2 Product Functions

1. Creating profiles
 - a. Casual player
 - i. Listing games they play
 - ii. Contact information creation (optional)
 - iii. Creating Game priority
 - iv. Listing desired clan allocations s
 - b. Competitive player
 - i. Sub user of casual
 - ii. Additional listing of competitive game
 - c. Clan Leader
 - i. Sub user of casual
 - ii. Game they own a clan of
 - iii. Clan information documentation
2. Players can join and create clans
 - a. Create clan
 - i. User should be able to select which game the clan is for
 - ii. Choose the rank range the clan operates in.
 - iii. Ability to restrict age range
 - b. join clan
 - i. User should search for a game and chooses one from the list of relevant clans
3. Commenting
 - a. Members
 - i. Post comments
 - ii. Comment on a comment
 - iii. Edit their old comments
 - b. Clan Leader
 - i. Post comments
 - ii. Comment on a comment
 - iii. Edit any comments
4. Data retrieval
 - a. Searching for other users to play games with.
 - b. Searching for clans to join.
5. Direct messages
 - a. Users can personally message other users
6. Make and manage a friends list
 - a. Add friends
 - b. Remove friends
 - c. Favorite friend.

2.3 User Classes and Characteristics

Users of our system will be casual gamers, competitive gamers, and clan leaders. Each person can be one or more of each of these types of people. Users can search for clans to join or just look for other people to play games with. Users who are members of clans participate in clan activities while clan leaders organize players, create events, and can recruit new members.

2.4 Operating Environment

Our product is designed with Windows 10 and MacOS in mind. It is a local executable that launches a GUI that the user interacts with. The application itself connects to a remote database to store information that isn't hosted locally.

2.5 Design and Implementation Constraints

Objects of concern include the feasibility of hosting the database on a raspberry pie. In addition, a method of keeping the chat updated on the user's end.

2.6 User Documentation

NYI (Not yet implemented)

2.7 Assumptions and Dependencies

There will be constraints on time (as this is a university project) as well as constraints on marketability (not having financial backing to monetize.) This will affect the product as it needs many users in order for each user to achieve the full experience.(Many number of clans to choose from.)

3. External Interface Requirements

3.1 User Interfaces

After logging in users will find a list of all of their clans with a sidebar to the left listing all of their friends with the option to click on them and pull up a chat window at the bottom left. Once an clan icon is selected the user will be brought to the clans main page where the user will see a clan photo, clan

bio and other relevant information as well as a clan bulletin board and a clan chat side bar replaces. If instead on choosing one of the clans you select a

3.2 Hardware Interfaces

The app will support Windows and OSX which will communicate with a server.

3.3 Software Interfaces

The software is designed in python using tkinter which accesses a MySQL server. Data held on the server consists of user information such as games played and their ranking in them, user photo, clan info such as bios and all messages to and from all users.

3.4 Communications Interfaces

This application will communicate with a database. Our application will query the database to populate each clans page as well as the chat functions.

4. System Features

4.1 Clan Search

4.1.1 The user can search for various clans for different games all ranging from different difficulties. The app then provides the user with a list of suitable clans.

4.1.2 Stimulus/Response Sequences

User navigates to a clan search page, enters search parameters and then is given a list of clans that meet the search criteria.

4.1.3 Functional Requirements

REQ-1: A database holding information about clans.

4.2 Searching for other players

4.2.1 This feature is for users that want to find other players to play games with that match games the user plays.

4.2.2 Stimulus/Response Sequences

User navigates to a player search page, enters some search parameters and then is given a list of players that meet the search criteria.

4.2.3 Functional Requirements

REQ-1: A database holding information about players.

4.3 Chatting with other players

4.3.1 This feature will allow users to chat with other players, there are chat areas for each clan and for each game.

4.3.2 Stimulus/Response Sequences

Users will be able to enter messages into a chat box and see messages from other people.

4.3.3 Functional Requirements

REQ-1: access to a datastream containing the messages for the chat.

4.4 Friends list

4.4.1 This is a list of other players created to keep track of people the user has enjoyed playing or wants to play more with.

4.4.2 Stimulus/Response Sequences

Users will be able to add and remove other users to a list

4.4.3 Functional Requirements

REQ-1: Each user will need a table in a database to store users they have on their friends list.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Chat feature must be operable and fast, lag time will discourage users from communicating. Furthermore, game-clan matchability must be consistent with user's level of proficiency to prevent users from being ill-matched and therefore not contribute well with clan.

5.2 Safety Requirements

Users will be asked for their birthdate, as well as a disclaimer about age. The clan leader will have an ability to restrict the age range for the clan. Users not following common respect and decency rules will be banned from usage of the application.

5.3 Security Requirements

As far as security goes, there will be safeguards against people being able to obtain other user's passwords or personal information. The application will not contain extremely sensitive data, so the most important aspect is protecting personal information. The greatest concern is SQL injection.

5.4 Software Quality Attributes

Availability and reliability will be extremely important to the user. Users will need to be able to communicate effectively with their clan.

Usability is important to the app: we aim for it to be very simple for the average person to operate. It is meant to be a fun experience used when playing games, not a difficult process that makes communication with other gamers more difficult.

5.5 Business Rules

All users can search for clans, and chat in the chat box
Clan leaders are able to add members, drop members, and restrict age range.
Clans themselves will have a max number of members, depending on the game

Appendix

Team Roles

Team Leader
Lead Programmer
Quality Assurance
Designer

Andrew Thorsen
Jeffrey Heredia
Michael Thorsen
Rachel Tidwell