PartyUp! SRS

Joshua Renn, Elijah Joaquim, Jon Fisher, Keyton Lanier, Jaylen Davis
CS 411W
Professor Thomas Valva

Table of Contents

1. Introduction **3**
1.1 Purpose 3
1.2 Scope 3
1.3 Definitions, Acronyms, and Abbreviations 3
1.4 References 4
1.5 Overview 4
2. Overall Description **4**
2.1 Product Perspective 4
2.2 Product Functions 4
Figure 1: Major Functional Components Diagram 5
2.3 User Characteristics 6
2.4 Design and Implementation Constraints 6
2.5 Assumptions and Dependencies 6

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document describes the requirements and overall design considerations for the PartyUp! web application. The document defines the system's functionality, design constraints, and interfaces to guide its development and testing.

PartyUp! is intended to provide a structured and efficient way for video game players to find compatible teammates, addressing the limitations of existing matchmaking systems in online multiplayer games. This SRS serves as a reference for developers, testers, and stakeholders throughout the software development lifecycle.

1.2 Scope

PartyUp! is a browser-based web platform designed to connect players seeking teammates for online multiplayer games. It enables users to search for or host parties using specific filters such as game title, platform, region, language, rank, and role.

The primary objectives of PartyUp! are:

- To enhance the quality of online matchmaking through structured search and filtering.
- To support real-time team formation and communication between players.
- To create an inclusive and safe community space for both casual and competitive gamers.

The system will consist of three major components:

- **Frontend**: A React-based user interface for browsing, filtering, and managing parties.
- **Backend**: Node.js and PHP servers managing matchmaking, authentication, and data communication.
- **Database**: A MySQL database storing user accounts, party listings, and community content.

1.3 Definitions, Acronyms, and Abbreviations

- **LAMP**: Linux, Apache, MySQL, PHP – a web development stack.
- **Frontend**: The part of the system that users directly interact with through a web browser.
- **Backend**: The server-side logic responsible for processing requests and managing data.
- **Party**: A group of players formed for cooperative or competitive gameplay.
- **Matchmaking**: The process of connecting players based on compatibility factors.

- **Community Hub**: The interactive area where users share experiences and communicate with others.

1.4 References

No external documents are referenced at this stage. Future versions may include design specifications, database schemas, and user interface prototypes.

1.5 Overview

The remainder of this document outlines the system's overall description and core features. Section 2 provides a general overview of the PartyUp! platform, including product functions, operating environment, and major components.

---

2. Overall Description

2.1 Product Perspective

PartyUp! is an independent, browser-based web application that serves as a centralized matchmaking and community platform for gamers. It addresses a common problem in modern multiplayer games—finding compatible teammates for cooperative or competitive play.

While many games feature built-in matchmaking, these systems often fail to match players based on criteria beyond rank or skill level, such as language, playstyle, or attitude. Social media and gaming forums, though helpful, rely on asynchronous communication and are not optimized for real-time matchmaking.

PartyUp! integrates and improves upon these concepts by providing a unified, real-time matchmaking experience accessible from any modern browser.

2.2 Product Functions

PartyUp! provides several core functionalities designed to enhance player connectivity and matchmaking efficiency:

- **Smart Filtering and Search System**:
  Players can create or search for parties using detailed filters such as game, platform, region, rank, role, and language.

- **Cross-Platform Accessibility**:
  The system operates on any device with a modern web browser, supporting PC, console, and mobile users alike.
- **Party Creation and Hosting**:
  Users can create customizable party listings with specific requirements, such as scheduled playtimes or role preferences.
- **Player Profiles and Preferences**:
  Players can maintain a profile with preferred games, playstyles, and communication preferences, allowing for faster and more accurate matchmaking.
- **Community Interaction**:
  A built-in community hub enables players to share content, guides, and discussions.
- **Integrated Moderation and Safety Tools**:
  Users can report toxic behavior or block disruptive players to maintain a safe environment.

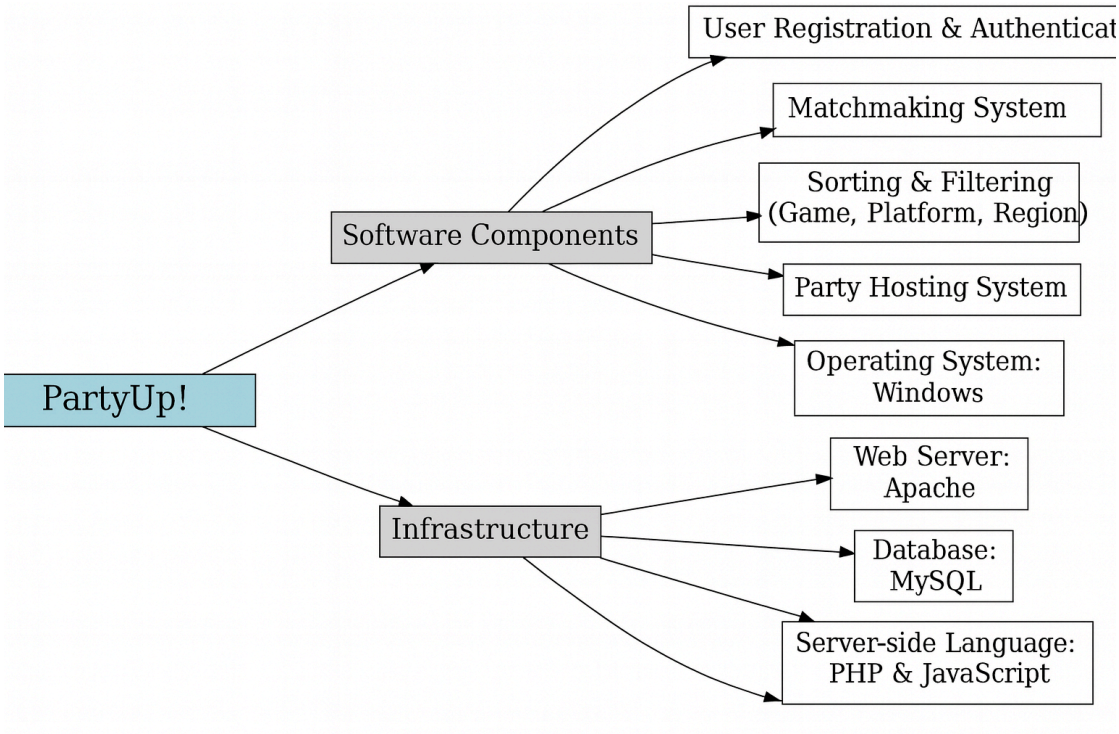

Figure 1: Major Functional Components Diagram

2.3 User Characteristics

The intended users include:

- **Casual Players** seeking friendly, cooperative gaming experiences.
- **Competitive Players** looking for skilled teammates for ranked or organized play.
- **Cross-Platform Players** who wish to connect across PC, console, and mobile.
- **Community Members** interested in sharing content, guides, or discussions.

Most users will possess basic computer literacy and familiarity with online gaming platforms.

2.4 Design and Implementation Constraints

- The system must function within standard browser capabilities (HTML5, CSS3, JavaScript ES6+).
- All user data must comply with standard web security practices, including encrypted passwords and HTTPS communication.
- The system must remain scalable to support future feature additions without architectural overhaul.

2.5 Assumptions and Dependencies

- Users will have access to a stable internet connection.
- The database server will remain operational for continuous matchmaking service.
- Third-party APIs or libraries (e.g., React, Node.js modules) will remain maintained and up to date.
- Browser compatibility will be maintained through ongoing updates and testing.