Music Player and daemon

Vikash Agrawal
Prashant Jain

October 1, 2011

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe all the requirements of the Music Player and daemon (MPD). The intended audience include all developers, hobbyists and students willing to learn some new. It will be licenced under GPLv3 and the project will be soon hosted at www.github.com.

Developers should consult this and its revisions as the only source or requirements for the project. They should not consider any requirements statements, written or verbal as valid until they appear in this document or its revision.

1.2 Scope

The proposed software product is Music Player and daemon. This software would be a full fledged music player encorporating all basic such as features to Play, Pause and Stop, or be it volume control etc. Moreover the selling feature of the musicplayer is to retrieve lyrics of current playing song also broadcasting it over http and binding it to a particular port number.

The intentions of this software is to make a music player, and on the way learn, Qt using C++ , phonon backend and how to take a dip into Open Source development and contributions.

1.3 Definitions, Acronyms, and Abbreviation

MPD	Music Player And daemon
Qt	Qt Framewors
CP	Currently Playing Song
LY	Lyrics Of the Song
Ph	Phonon backend
GUI	Graphical User Interface
SRS	Software Requirement System
Linux	Linux Operating System

1.4 Refrences

Qt documentation, man pages and phonon library articles have been used as a refrence for this document

1.5 Overview

This Software Requirement Specification (SRS) is the requirements work product that formally describes the Music Player and daemon (MPD). It includes the result of analysis done for the project Various techniques were used to elicit the requirements and we have identified your needs, analyzed and refined them. The objective of this document therefore is to formally describe the systems high level requirements including functional requirements, non-functional requirements and business rules and constraints. The detail structure of the documents is organized as follows:

Section 2 of the document includes description of the product, user characteristics, general constraints and aassumptions for the Music Player. This model demonstrates the development's team understanding for the product and aims to mazimize the teams ability to build a system that does support the business

Section 3 presents the detail requirements which comprise the domain model.

2 General Description

2.1 Product Perspective

The Music Player and daemon is a player that is fully functional which fetches lyrics for the currently playing song and simultaneously broadcasts music over http to a prescribed port number.

2.2 Product Functions

The system functions can be described as follows:

- **Add Music** User can add music to the current playing list and by this the music library will display all the music files selected
- **Playing a song** User can Play, Pause and Stop the music from the queue. By this user can simulataneously view the view the length of music, and where is the current scroll using a scrollbar
- **Fetching Lyrics** The lyrics of playing song is fetched from various third party sites and this will displayed in the same window but in a separate text area
- **Broadcasting** This might require root permissions of from the user and then all the current playing songs will be sent the a port number from where other users can listen it with just having the ip and the specified port number

2.3 User Characteristics

The music player will be used by end users in both Linux and Windows (using cross-compilation) The system will be using a Graphical User Interface (GUI) and be as user friendly as possible. End User They are the main users of the software, and may comprimise of personel from any domain

2.4 General Constraints

- The system has to be delivered within 3 months
- The system should with compatible with Linux and Windows

- The system should fetch lyrics as fast as possbile, considering network constraints
- The system should be as light weight as possible
- Exisisting players dont support broadcasting of music so it should be of primary concern
- The system must be user friendly

2.5 Assumptions and Dependencies

- It is assumed that there is no compatiblity issue in the software and the Operating system
- It is assumed that there is no problem in the network connectivity for the fetching of lyrics
- It is assumed that the backend of the music player is well integrated and supported by the system
- It is also assumed that, the output by crosses compilation runs natively in Windows

3 Specific Requirements

This section describes the specfic requiements of the software

3.1 Functional Requirements

SRS001	Add Song	To Add music to the list
SRS002	Play	To Play the currently selected song
SRS003	Pause	To pause the currently playing song
SRS004	Stop	It will stop the currently playing song
SRS005	Lyrics	Fetch lyrics on the current playing songs
SRS006	Shuffle	Playing song in shuffle mode
SRS007	Broadcasting Music	Broadcast the current playing song over http
SRS008	ID3 tags	The music library will display all the ID3 tags and for all songs

3.2 Design Constraints

SRS009	Desktop Applications	The system shall be a desktop application
SRS010	Operating System	The development will be done on Linux
SRS011	GUI Toolkits	Qt using C++ will be used for the Graphical User Interface of the system
SRS012	Backend	The music player will use phonon as the backed
SRS013	Network Connectivity	The system will use network connectivity for fetching lyrics

3.3 Non-Functional Requirements

3.3.1 Security

SRS014	Root - Login.	The system will require root permissions for starting the port and cofiguring it to broadcast music over that

3.3.2 Performance

SRS015	Response Time	The response time to play a song should be less that 0.5 sec
SRS016	Capacity	The capacity of songs that can be added to the music library shouldnot restricted to any number
SRS017	Licence	The system will be licenced under GPLv3
SRS018	User-Interface	The interface should respond within 5 seconds

3.3.3 Mainatainablity

SRS019	Errors	The system should keep logs of all errors and crashes
SRS020	Verification	The system should include test cases to test the installed system and cofigurations or ports

3.3.4 Reliablity

SRS021	Availablity	The system should be available at all times