

Software Requirements Specification

for

PDF Share & Schedule

Version 0.1

Prepared by

Group Name: R&B

|  |  |  |
| --- | --- | --- |
| Ben Mikalenko | 11589806 | Ben.mikailenko@wsu.edu |
| Roman Stolyarov | 11689068 | Roman.stolyarov@wsu.edu |

|  |  |
| --- | --- |
|  |  |
| Date: | Friday, October 25th 2019 |
|  |  |
|  |  |
|  |  |

Contents

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 Behaviour 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 |
| --- | --- | --- | --- |
| Rough Draft  v0.1 | Ben Mikailenko  Roman Stolyarov | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded. | 10/03/2019 |

# Introduction

## Document Purpose

This document (v 0.1, Oct. 3 2019), describes the design, data, and structure of a web based PDF sharing and schedule system for music bands. This document describes a single subsystem of the web portion designed to only show scheduling and PDFs. This document does not include the instruments schedule or all the bands schedules in the venue. This will not be the complete known documentation, in later versions more specific documentation will be recorded.

## Product Scope

PDF share and schedule provides a hub for band members to easily share and access their chord charts.

Sharing:

Chord charts are uploaded in PDF format by the band leader. The file is then available for download by the rest of the band. Making distribution of chord charts quick and easy.

Scheduling:

Uploaded files are organised by date when the band is scheduled to play. Letting each band member know what songs they are playing and when.

## Intended Audience and Document Overview

This document is intended for the venue board organizer and band leader for consultation and future implementation of the product at hand. This document is intended for those who are interested in the set-up, and planning phase of the PDF share & Schedule website. Primarily any clients who are interested in using the format and layout of the project. This document will also be used between all the project managers that will be forseeing future updated versions of the website.

In the rest of the document, the layout will go as follows:

(It is recommended to continue reading the document in the order provided)

1. Product Description
   1. Purpose
   2. Functionality
   3. User Interface & Characteristics
2. Product Environment
   1. Operation & Hardware Platforms
   2. Possible Constraints
3. Product Utility
   1. WORK ON AS WE MOVE FORWARD

## Definitions, Acronyms and Abbreviations

Chord Chart – A form of musical document that lists a songs lyrics along with the chords played at each section

Date (in regards to the webpage) – An entry to the main webpage that has a real calendar date as a name and has zero or more PDFs attached and avalible for download.

GUI – General User Interface

PDF – Portable Document Format

## Document Conventions

The document follows IEEE formatting requirements. An arial 11 point font, single space and 1” margins are used.

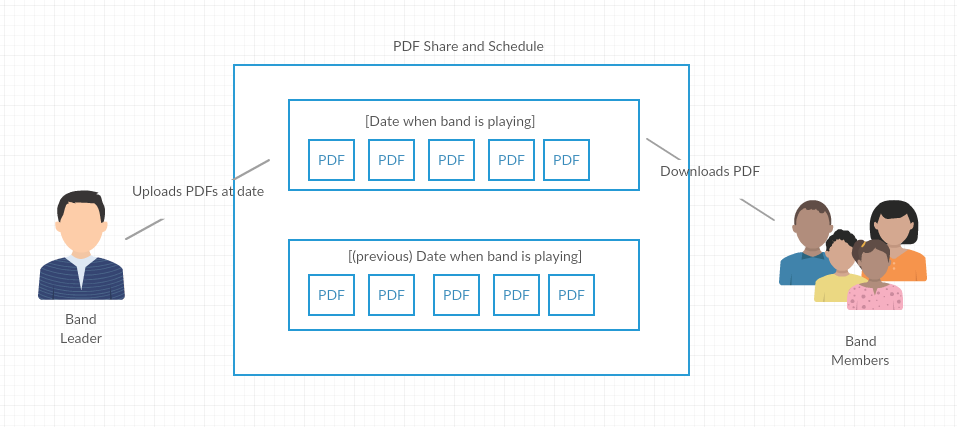
## References and Acknowledgments

NO REFERENCES YET (USE IEEE CITATION IF WRITING A REFERENCE)

# Overall Description

## Product Perspective

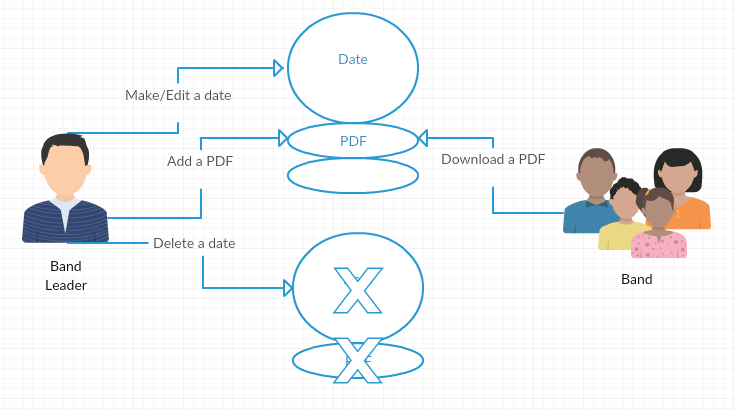
PDF Share and Schedule is a new, self-contained system helping file sharing between band members.



## Product Functionality

The major functions of the program are:

* Make, edit, and delete a date
* Add, remove, or download a PDF



## Users and Characteristics

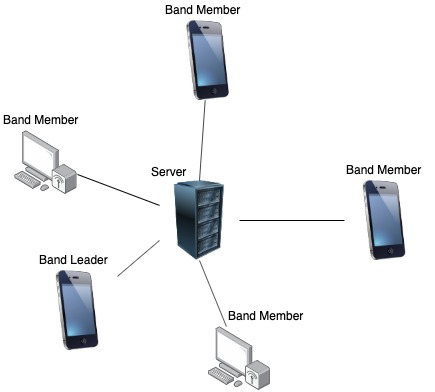
Band Leader:

The band leader is responsible for creating the dates and uploading the files. This is the most important user. They will use the “Make / Edit / Delete Date” and “Add / Remove PDF” functions.

Band Member:

The band members will only use one function and that is “Download”. They do not need to know much about the inner functions of the program.

## Operating Environment



The minimum platform requirements are any device that runs a web browser. The device must be able to download and store files to use the full cababilities of the program.

The hardware running the program is an external web server that can be accessed from any device meeting the minumum platform requirements.

## Design and Implementation Constraints

One constraint that the system might have that the developers want to consider is the devices that will be used in order to download the PDF’s. We are not able to test web-browsers on all given devices and therefore resulting in some devices not functioning properly. This goes hand-in-hand with testing all browsing applications as well.

Another constraint is the internet. When developing the code and pushing multiple versions to github, a solid internet connection needs to be available. Also, since this is an online application, when testing online functionality there must also be a good connnection. The internet may have a wide range of connection qualities and the developers are not able to consider all the possibilities when testing the connection of the system.

## User Documentation

In the system, there will be a constant toolbar that will list the available options such as add, delete, edit, as well as a help option. This help option contains the online manual. The manual gives a brief overview of what the available options are capable of doing and how to use them the way designed.

## Assumptions and Dependencies

We plan to use a third-party hosting application that allows us to have the code running on a external server.

We can assume that the product will be developed on multiple operating systems such as macOS and Windows OS. The source code will be distributed through a remote Github repository.

Therefore, we must assume that Github is working properly as well. Keeping a backup of the source code on individual personal computers will be used to insure no loss of information if Github stops working unexpectedly.

# Specific Requirements

## External Interface Requirements

### User Interfaces

A toolbar will be permenently locked at the top of the screen (even while you are scrolling through the webpage) that will help either add, edit, or delete existing box that contains the schedule and songs planned for that week.

Add button:

If the add button is hit under the toolbar, a GUI will pop up asking to input the date that is scheduled, and another add option that allows you to enter in multiple PDF’s for that current box.

Edit button:

If the edit button is hit under the toolbar, a GUI will pop up asking which box do you want to select to edit. If there is several boxes that will not fit in the whole GUI, a scroll option will appear to fit in previous added boxes.

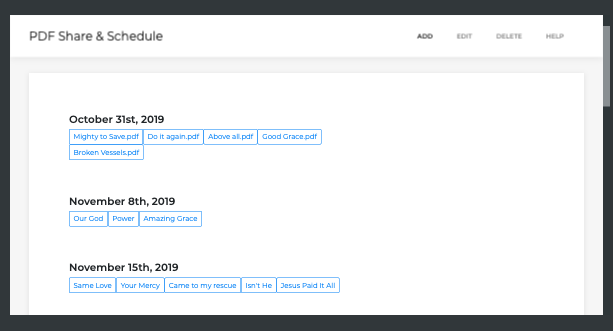
In this button option, once a certain box is selected, you are then able to edit the date, or edit the files that are already inputted in the box. You are able to either remove, or add more files to the scheduling box selected.

Delete button:

The delete button will allow you to remove an existing box by having a GUI pop up to select which scheduling box you want to remove.

Help button:

A help button can be found in the toolbar as well that can assist if the given functionality is not understandable for the customers (band members). The help button contains a help manual GUI box that contains what each funcationality does (add, edit, or delete).



(Example: Prototype Home Page)

### Hardware Interfaces

PDF Share and Scheule runs on the web and is accessed by any web browser. As long as there is a solid internet connection, any hardware device that runs web browsers can use PDF Share and Schedule. Some browsers are unable to run javascript, so it is required that the hardware have a newly updated browser.

### Software Interfaces

PDF Share and Schedule requires the software, online hosting application, operating system, and other software components to run a web based javascript application. The software components must be able to run the vital interpreters and compilers for the primary javascript framework (NodeJs), along with an ExpressJs library.

### Communications Interfaces

On the server side, PDFs will be uploaded and downloaded to the server via FTP. HTTP will be used to find the website via web browser. The the user side, communication will be help with the server via GUI (possibly bootstrap framework). A menu will be loacted on the top of the webpage allowing the user to bring the GUIs. A short user help manual will also be located in the menu.

## Functional Requirements

1. Make a date entry to main webpage

The web application must allow the user to make a date on the webpage (with attached PDFs) via a GUI.

* 1. Make a date GUI

The make date GUI must have access to add entries to the main webpage. The GUI must also have an area where the user can input a date (ex: December 21st 2019), and upload multiple PDFs from the local drive. User date input information should be required to submit. The GUI will have a button that sumbits the user inputted information to the main webpage. The GUI should also have a button that cancels the user input and exits the GUI.

1. Delete a date entry from the main webpage

The web application must allow the user to delete a date from the main page along the it’s attached PDFs

* 1. Delete a date GUI

The delete date GUI must access to delete any entry in the main webpage. The GUI must have a drop down menu where the user can scroll to find a date (ex: December 21st 2019), and a button to delete the selected date (along with its PDFs) from the server. The GUI should also have a button that cancels the user input and exits the GUI.

1. Edit a date entry on the main webpage

The web application must allow the user to edit a date (and attached PDFs) on the webpage via a GUI.

* 1. Edit a date GUI

The edit date GUI must access to edit any entry in the main webpage. The GUI must also have an initial drop down menu where the user can scroll to find a date (ex: December 21st 2019). The GUI must also have an empty “date name” and “PDFs” areas respectivley, underneath the drop down menu.

When the user has selected a date on the GUI, the date and PDFs should be palced into the apropriate areas below the drop down menu. The “date name” area in the GUI should be a text area where the user can edit. The “PDFs” area in the GUI should contain every PDF from the inputed date and allow the user to delete and add a PDF. There should be a button in the GUI to submit the revised date (along with revised PDFs) to the server. The GUI should also have a button that cancels the user input and exits the GUI.

1. Main webpage

The main webpage can be a long list containing every date entry. The main webpage must list dates (along with their PDFs) in order of upcomming calendar date. On any date in the main webpage, there must be links to download the uploaded PDFs. The main web page should communidate with the add, edit, and delete functions and update respectivley.

1. Main wepage menu

The web application must have menu with links to open the “make a date”, “delete a date”, “edit a date” GUIs, and manual.

## Behaviour Requirements

### Use Case View

Add a date – Adds a date entry (along with PDFs) to the main webpage

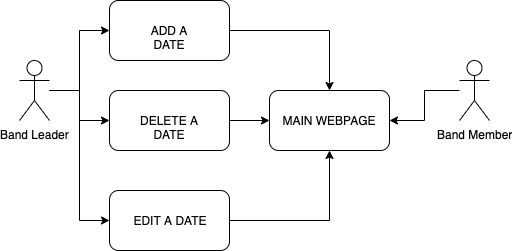
Delete a date – Deletes a date entry (along with PDFs) from the main webpage

Edit a date – Fetches a date entry, allows to edit date/PDFs, adds back to main webpage

Main webpage – Lists all date entries and allows PDFs to be downloaded

Band Leader – Runs what informaiton is on the webpage

Band Member – Downloads PDFs that the band leader uploads



# Other Non-functional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide relevant performance requirements based on the information you collected from the client. For example you can say “1. Any transaction will not take more than 10 seconds, etc…>

## Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

* Provide relevant safety requirements based on your interview with the client or, on your expectation for the product.
* Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, etc.>

# Other Requirements

<This section is **Optional.** 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.>

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.>*

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>