

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

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.

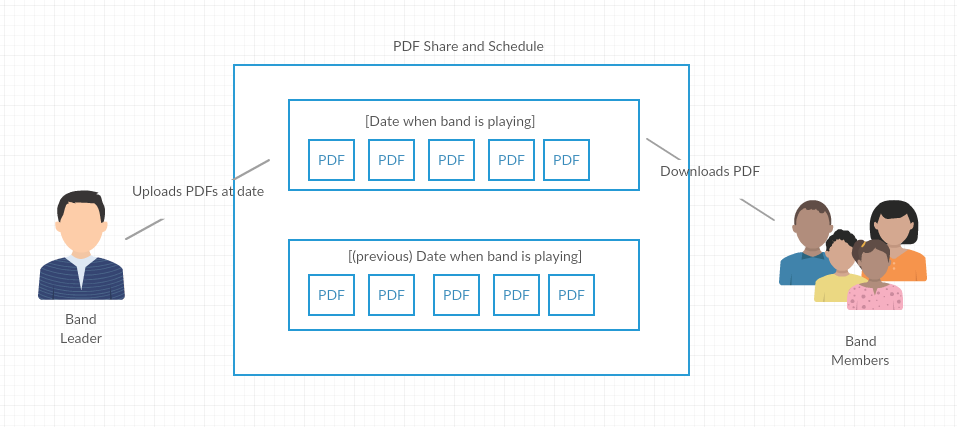
## References and Acknowledgments

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

# Overall Description

## Product Perspective

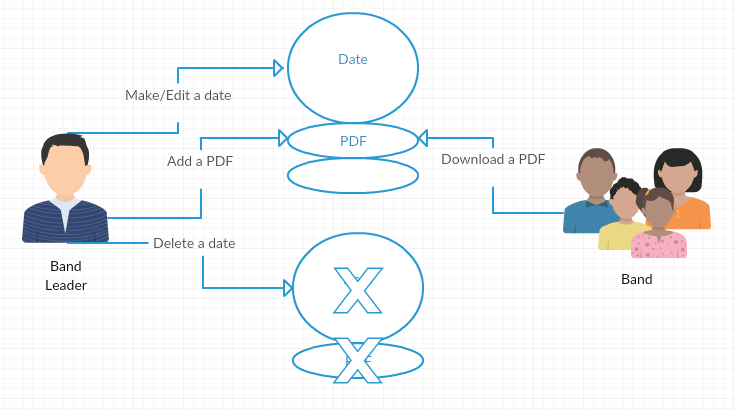
PDF Share and Schedule is a new, self-contained system helping file sharing between band members.



## Product Functionality

The major functions of the program are:

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



## Users and Characteristics

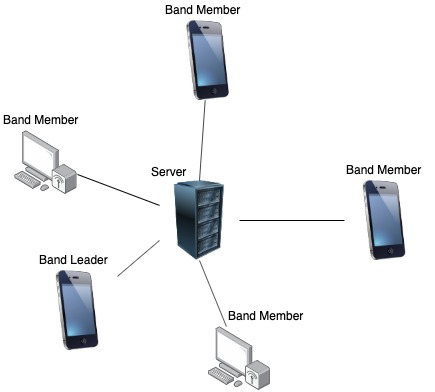
Band Leader:

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

<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).

TO DO: In this section you need to consider all of the information you gathered so far, analyze it and correctly identify relevant 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

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.

TO DO: You will not actually develop any user-manuals, but you need to describe what kind of manuals and what kind of help is needed for the software you will be developing. One paragraph should be sufficient for this section.>

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

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

TO DO: Provide a short list of some major assumptions that might significantly affect your design. For example, you can assume that your client will have 1, 2 or at most 50 Automated Banking Machines. Every number has a significant effect on the design of your system. >

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

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., Cancel) that will appear on every screen, error message display standards, and so on. Define the software components for which a user interface is needed.

TO DO: The least you can do for this section is to describe in words the different User Interfaces and the different screens that will be available to the user. Optional: You may also provide an initial Graphical User Interface design (does not have to be final).>

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).

### Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>

### Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

### Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

## Functional Requirements

*< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions with specific explanations regarding every function.*

*TO DO: Break the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.*

## Behaviour Requirements

### Use Case View

<A use case defines a goal-oriented set of interactions between external actors and the system under consideration.

TO DO: Provide a use case diagram which shows the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram.>

# 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>