Worship Together

Design Specification

Table of Contents

[Requirements 3](#_Toc510267695)

[Design 4](#_Toc510267696)

[Data 4](#_Toc510267697)

[Interface 5](#_Toc510267698)

[Behavior 5](#_Toc510267699)

[Verification 6](#_Toc510267700)

[Manual 6](#_Toc510267701)

[Automated 6](#_Toc510267702)

# Requirements

The application shall…

1. Be usable by non-technical and non-musical users.
2. Enable users to view this week’s songs for the upcoming worship service.
3. Enable users to search for a specific song among all available songs.
4. Enable users to play the four choral parts and/or piano accompaniment simultaneously.
5. Enable users to view the lyrics & sheet music while songs are playing
6. Enable users to independently control volume of each choral part (soprano, bass, etc).
7. Enable users to navigate forward and backward in the currently playing song.
8. Enable music leaders to update the weekly song list.
9. Enable music leaders to register new churches.
10. Enable users to join church (no password required).
11. Enable music leaders to set and change the password for updating the weekly song list.
12. (Advanced) Enable users to match pitch with the first note of any choral part.

Would love to pawn off security on Google docs and let users edit data directly…

Forget about publishing weekly songs for now – only have song list and player

# Design

## Data

All application information will be stored in the following directory hierarchy:

* doc
  + wireframe – Wireframe description of the user interface
  + design – Design specification for the application (this document)
* code
  + client
    - HTML files to be downloaded by the browser
    - CSS files for styling the HTML display
    - JavaScript files for browser behavior
  + server
    - JavaScript web server code for serving up the client files and music
* music
  + One sub-directory per song.
  + Directory names shall have the following format: <num>.<psalm>.<title>
    - <num> - A series of digits specifying the song number (optional)
    - <psalm> - The Biblical Psalm prefixed with “ps\_” (optional)
    - <title> - The title of the song, spaces replaced with underscores
  + Examples of directory names:
    - 311.Jesus\_Shall\_Reign
    - ps\_114.Psalm\_114\_(NKJV)
    - 182.ps\_138.With\_All\_My\_Heart,\_My\_Thanks\_I’ll\_Bring
    - On\_Christ\_The\_Solid\_Rock\_(Fugue)
  + Each song directory shall contain one or more of the following files:
    - music.\* - Sheet music for song (visual)
    - soprano.\*, alto.\*, tenor.\*, bass.\* - Choral parts (audio)
    - piano.\* - Piano accompaniment (audio)
* church
  + One sub-directory per church
  + Directory names shall have this format: <name>.<city>.<state>.<country>
  + Spaces in names are replaced with underscores
  + Examples of directory names:
    - Christ\_Church.Moscow.ID.USA
    - Christ\_Reformed\_Church.Oakland.ME.USA
  + Each church directory shall contain the following files/sub-directories:
    - password – shall contain the password for updating the weekly song list
    - this\_week – sub-directory
      * Contains relative soft-links to song directories for this week’s songs. For example, “../../../ 311.Jesus\_Shall\_Reign”

## Interface

The browser shall utilize the following messages to make requests of the web server:

1. / - Empty message redirects to code/client/home.htm
2. code/client/<filename> - Download a file from the “code” directory
3. music/list – Download a list of directory names under the “music” directory
4. music/<directory>/parts – Download a list of available parts
5. music/<directory>/<filename> - Download a particular music file
6. church/create/<church\_name> - Create a church with the specified name
7. <selected church stored in browser cookie – ask permission?>
8. church/<church\_name>/this\_week/list – List of song directories for this week
9. church/<church\_name>/this\_week/clear?password=<password> - clear weekly list
10. church/<church\_name>/this\_week/add/<directory>?password=<password> - add
11. church/<church\_name>/this\_week/remove/<directory>?password=<password> - remove
12. church/<church\_name>/update\_password/<directory>?password=<password>,new\_password=<new\_password>

## Behavior

The web server shall behave as follows:

1. Upon receiving an empty request, the web server shall redirect the browser to “code/client/home.htm”.

# Verification

## 

## Manual (Browser)

1. Navigate to the website with no request. Verify that the browser is redirected to code/home.htm.

## 

## Automated (Web Server)

1. Submit request for the list of music. Verify that the response includes all directories under music.
2. Create new music sub-directory with properly formatted song name. Verify that a request for the list of music includes the new directory.
3. Delete the previously created music sub-directory, verify that a request for the list of music no longer includes that directory.
4. Submit church creation request with specified password. Verify new church directory exists with password file containing specified password.