SongHaven

Team :D (Colon Capital D)

Introductions

Who we are and what we do

Agenda

- Team Structure
- Repository / Communication
- Project Overview
- Basic Front End
- GUI Design
- Basic Back End
- Gantt Chart
- Technical Specifications
- Risk Management

Team Structure

- We have decided to separate our team based on our skills and experience.
  - Eric >> Project Coordinator|Programmer
  - Jordan >> Senior Programmer|Architect
  - Corey >> Junior Programmer|Tester
  - Christy >> Librarian|Organizer|Analyst

Repository/Communication

- Git
  - https://github.com/jghibiki/SongHaven.git
  - Commit messages
  - Group workshop to learn Git
- Git training from http://try.github.io

Project Overview

Host machine (Server):

- Accepts song requests from users
- Is populated with files from users
- Randomly queues songs when no requests

Client Website

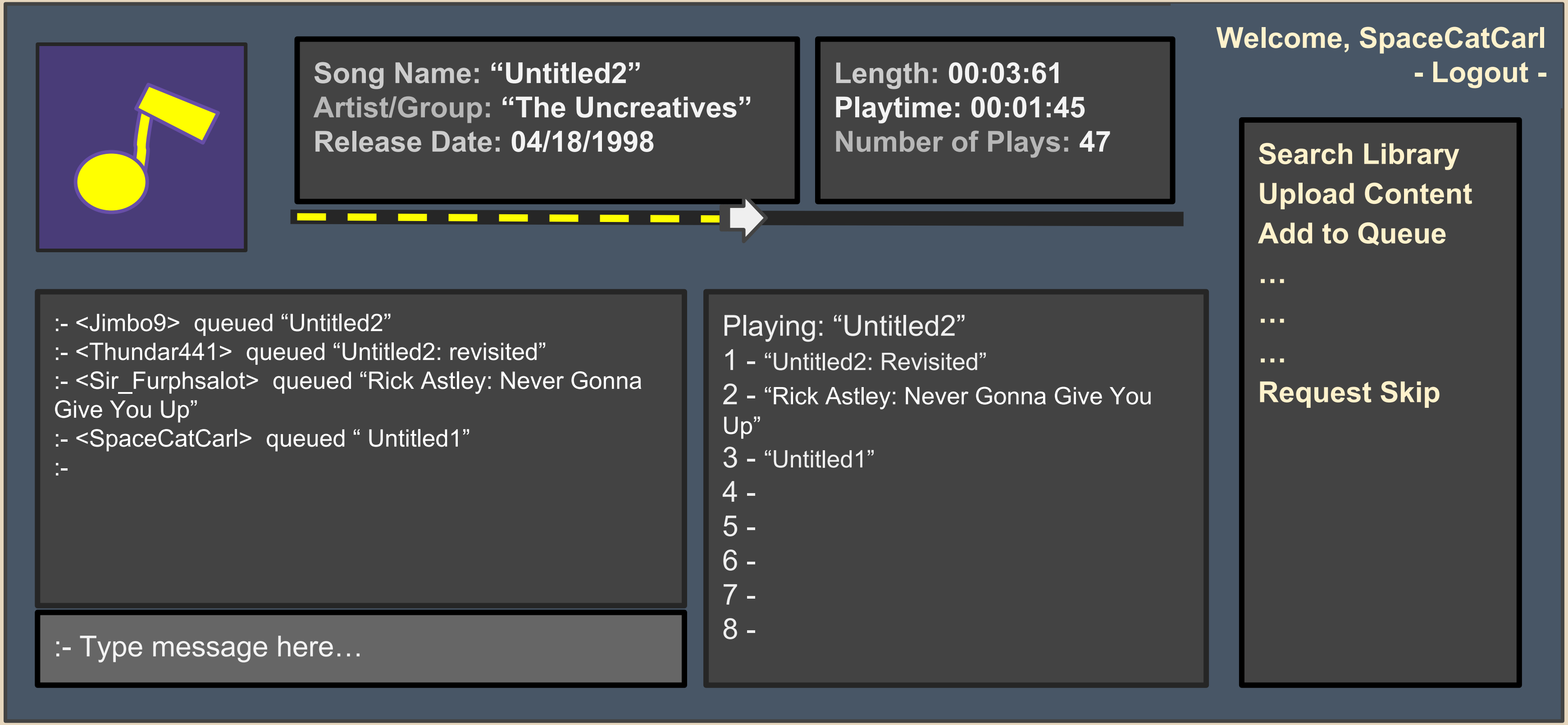
- Allows users to interface with the host to:
  - Search and request available music
  - Upload music
  - View the current playback queue

Front End

Web interface (Runs on client)

- Home page
  - Shows the current queue
  - Shows the currently playing song
- Search Page
  - Allows user to search for songs
  - Displays search results for songs
  - Allows user to request available songs
- Upload Page
  - Allows users to upload songs
- Library Page
  - Allows user to browse the music library

GUI Design (Rough)



Song Name: “Untitled2”
Artist/Group: “The Uncreatives”
Release Date: 04/18/1998

Length: 00:03:61
Playtime: 00:01:45
Number of Plays: 47

:- <Jimbo9> queued “Untitled2”
:- <Thundar441> queued “Untitled2: revisited”
:- <Sir_Furphsalot> queued “Rick Astley: Never Gonna Give You Up”
:- <SpaceCatCarl> queued “ Untitled1”
:-

Playing: “Untitled2”
1 - “Untitled2: Revisited”
2 - “Rick Astley: Never Gonna Give You Up”
3 - “Untitled1”
4 -
5 -
6 -
7 -
8 -

Backend

Single Server

- REST-ful services
- Handles api calls made by the web front end
- Manages file upload storage
- Performs library management & cataloging by song metadata
- Populate database with song metadata and user requests

Gantt Chart	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Objective															
Pick Groups	🟩														
Decide on a project	🟩														
Assign Responsibilities		🟩													
Determine Risk Management		🟩													
Set Specifications			🟩												
Design (Database and Entity and service models)			🟩	🟩											
Create Entities					🟩										
Music Player					🟩										
Web Interface					🟩	🟩	🟩								
Authentication					🟩	🟩	🟩								
Music upload						🟩	🟩	🟩							
User Functions						🟩									
Administration Functions						🟩									
User Management Functions							🟩								
Testing Functional/Integration								🟩	🟩	🟩	🟩	🟩	🟩		
Final Documents														🟩	
Presentation															🟩

Web Interface (Client Side)

- We will use Python Flask for our project
  - Python flask allows us to host both html pages and web services
  - Using this, we will use a web site to control our host machine.
  - We will cache frequently hit pages, to improve performance

Service Backend

- For our backend, we will also be using Python Flask to expose REST web services to control the host machine.
    - We will also use this to access and control our database, which will use an object relational mapper (ORM).
    - For this, we will be using Peewee.

Audio Player Backend

- Our backend will control an audio player attached to our host machine.
  - We will most likely be using MPlayer for this.
  - MPlayer includes a listener that can be easily manipulated by writing to a file.

Database

- As our database, we have chosen MySQL.
  - We chose this, because of general familiarity
  - Integration with the school (they already have it installed)
  - General ease of use.

Risk Management

- Lack of experience
  - Open communication, documentation, and tutorials
- Personnel issues
  - Open communication
- Hidden complexity
  - Stay on schedule, bottom up programming, become familiar with framework tools

Risk Management

- Scheduling issues
  - Communication outside of meetings
  - Keep Google + calendar updated
  - Split up tasks
- Unexpected project fault
  - Keep logs of processes and database transactions
  - Keep a version history

Conclusion

Questions?

Comments?

Concerns?

Compliments?

Cornbread?

Cream Soda?

Creampuffs?

Cthulhu?

Camels?

Cabbage?

Caribou?

Cactus?