Butterfly

By

JACOB WHEELER: Major in SE
NATHAN CHRISTIANSEN: Major in SE
NICHOLAS KAPTY: Major in SE

Instructor: Dr. Xiaocong Fan

Course Team Project
Computer Science and Software Engineering Department
Penn State Erie, The Behrend College

December 2015

CSSE Technical Report Series: SWENG411-2015Fall-003

Jake Wheeler, Nate Christiansen, Nick Kapty

Contents

Project Drivers

1. The Purpose of the Project
2. The Client, the Customer, and Other Stakeholders
3. Users of the Product

Project Constraints

4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions

Functional Requirements

7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements

Nonfunctional Requirements

10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational and Environmental Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements

Project Issues

18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Migration to the New Product
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

Jake Wheeler, Nate Christiansen, Nick Kapty

1. The Purpose of the Project

1a. The User Business or Background of the Project Effort

The purpose of the project is to create a music player application to play local music. Users will be able to sort by album, artist, song title, genre, as well as various other tags. Users will be able to log into social media to share what they are currently listening to.

1b. Goals of the Project

To create a quality music player that will allow users to listen to music, with customizable features and social media integration.

2. The Client, the Customer, and Other Stakeholders

2a. The Client

We are our own client, so we are using our own specifications.

2b. The Customer

The customer of our product would be anyone who wishes to use our music player.

2c. Other Stakeholders

Our only stakeholder is Professor Fan, who is teaching us as we develop the project.

3. Users of the Product

3a. The Hands-On Users of the Product

Any user who has the knowledge to use a computer and any passing interest in music can use our product.

3b. Priorities Assigned to Users

- **Key users: The users with Twitter and the ability to dig into the customization of our player. They will use all of the additional features of the product, such as playlists, social media features, etc.**

- **Secondary users: Users who simply use the player as a music player and nothing more. They will use the product for its local music playing ability only.**

3c. User Participation

We will act as users for this project, testing all of the features added. We will also consider outside input from unbiased third parties.

3d. Maintenance Users and Service Technicians

We will maintain and service our product after its initial release and all additional releases.

4. Mandated Constraints

4a. Solution Constraints

We will be using Java, because Java has a rich framework, and well developed and documented APIs. Java also runs on any operating system, as it is not platform dependent. The program will also meet all of our functional and non-functional requirements.

4c. Partner or Collaborative Applications

Our application will feature Twitter integration, and as such will use Twitter features and services.

4d. Off-the-Shelf Software

We will be using Twitter4J, an unofficial Java library for the Twitter API.

4e. Anticipated Workplace Environment

The environment targeted will be Windows machines with Java Runtime Environment installed, as well as internet connectivity.

4f. Schedule Constraints

The project must be completed by December 16, 2015, in order to receive full credit for the class.

5. Naming Conventions and Definitions

5a. Definitions of All Terms, Including Acronyms, Used in the Project

Butterfly – The name of our music player application

Song – Individual music file that can be played

Playlist – User selected list of songs

Artist – Creator of a song

Album – Collection of specific songs chosen by the artist

Tweet – A short message that is sent to Twitter that is written by the user.

6. Relevant Facts and Assumptions

6a. Facts

Our project will be developed in Java, and will be completed in one semester. The user will need a Twitter account to take advantage of all of the features that our application will offer.

6b. Assumptions

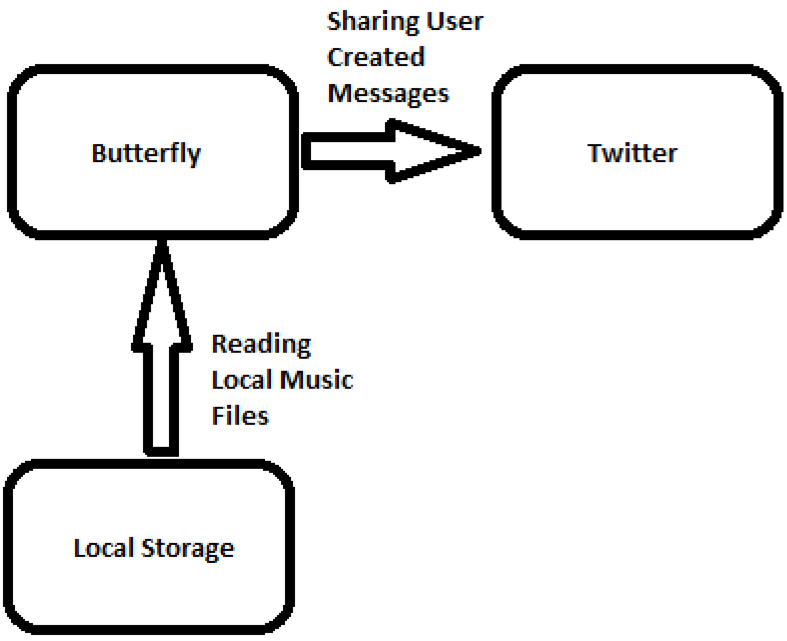
We assume that we will be able to develop the program using NetBeans and appropriate frameworks. We also assume Twitter integration will be available.

7. The Scope of the Work

7a. The Current Situation

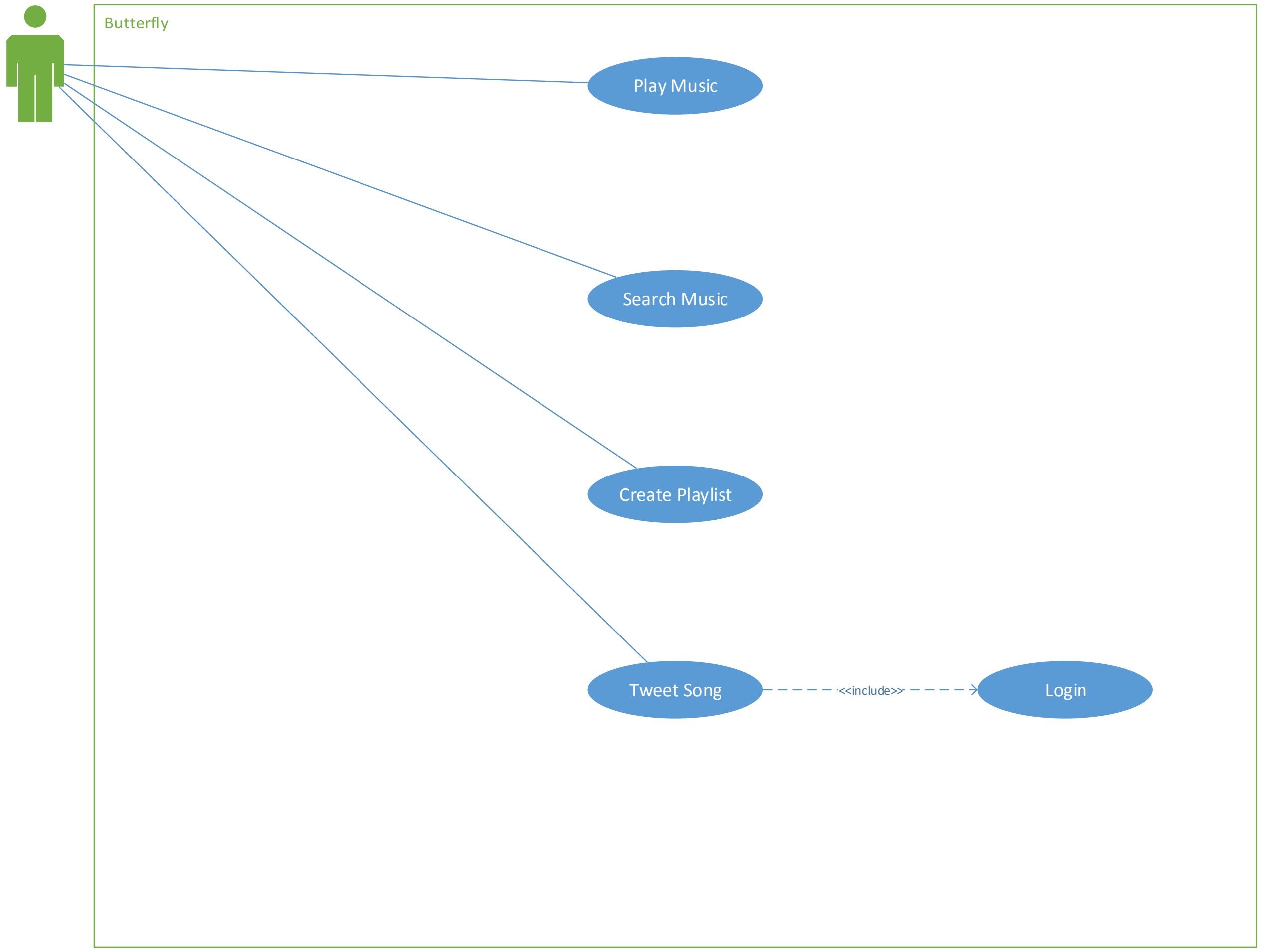
Currently there are no popular music players that allow users to engage with social media.

7b. The Context of the Work



8. The Scope of the Product

8a. Product Boundary

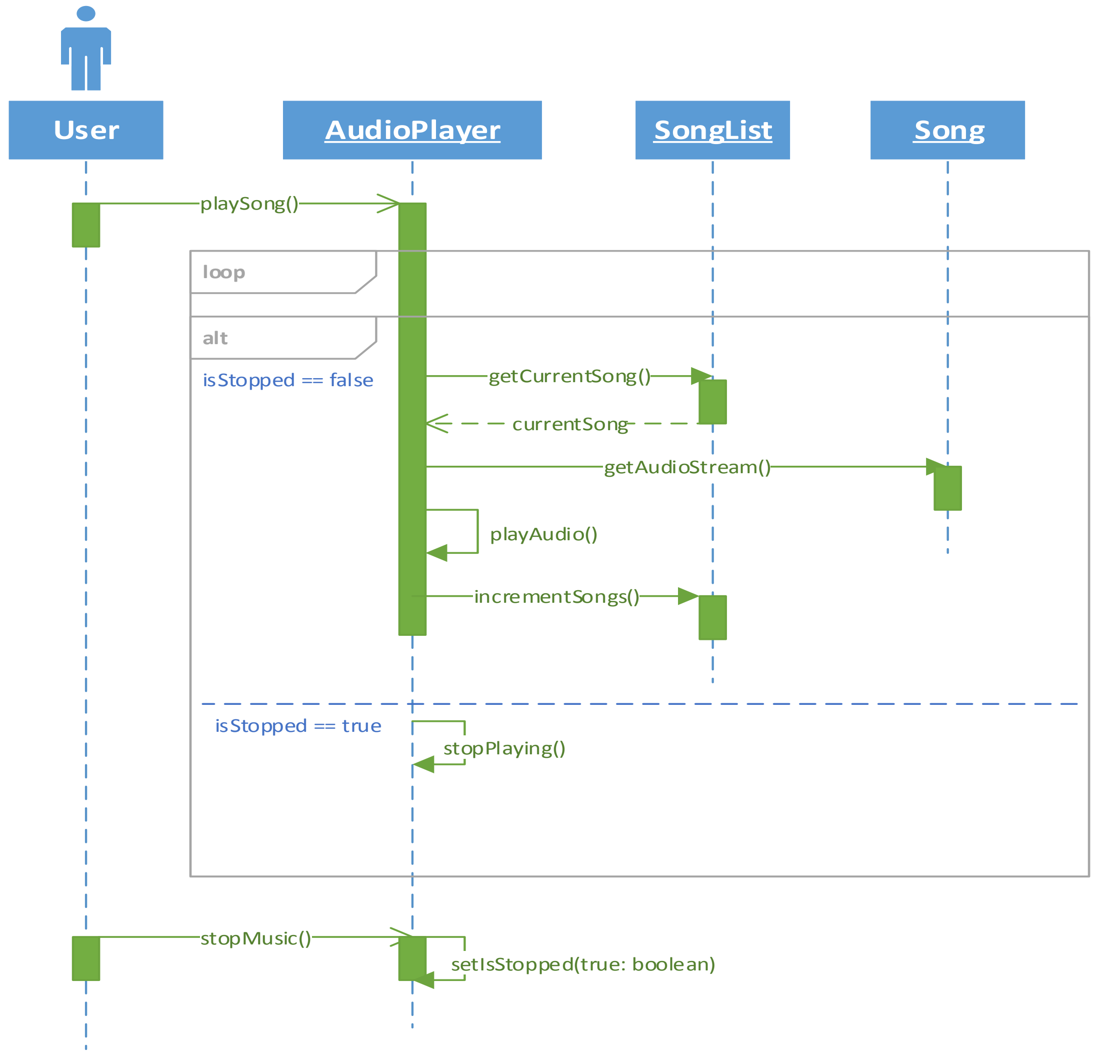


9. Functional and Data Requirements

9a. Functional Requirements

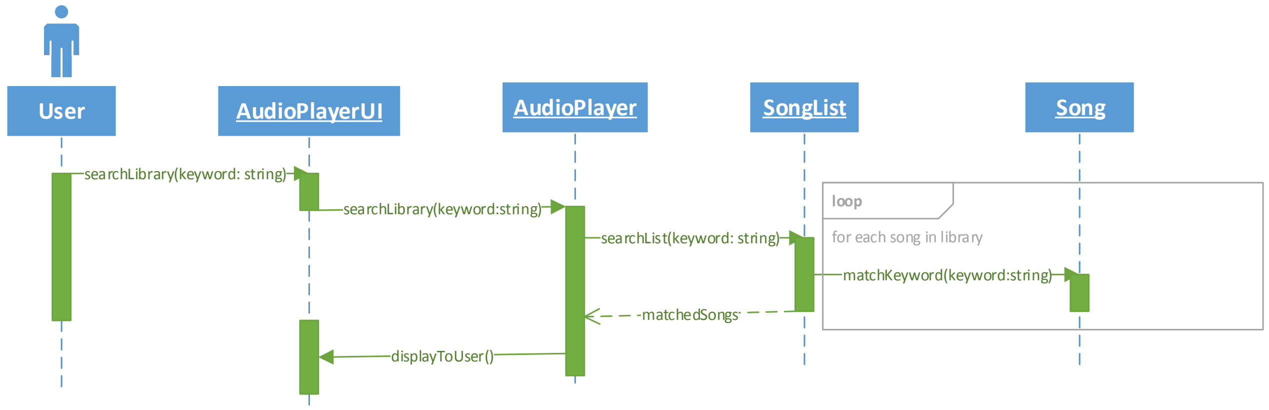
Use Case #1	Play Music
Goal in Context	To allow the user to play music
Scope	The Butterfly system
Level	Summary
Primary Actor	User
Preconditions	Song is highlighted/selected
Minimal Guarantee	Nothing happens
Success Guarantee	The selected song begins playing
Trigger	Play button or double click on the song
Description Step	
1	The user selects a song to play
2	The system finds the song file
3	The system begins audio output

Jake Wheeler, Nate Christiansen, Nick Kapty



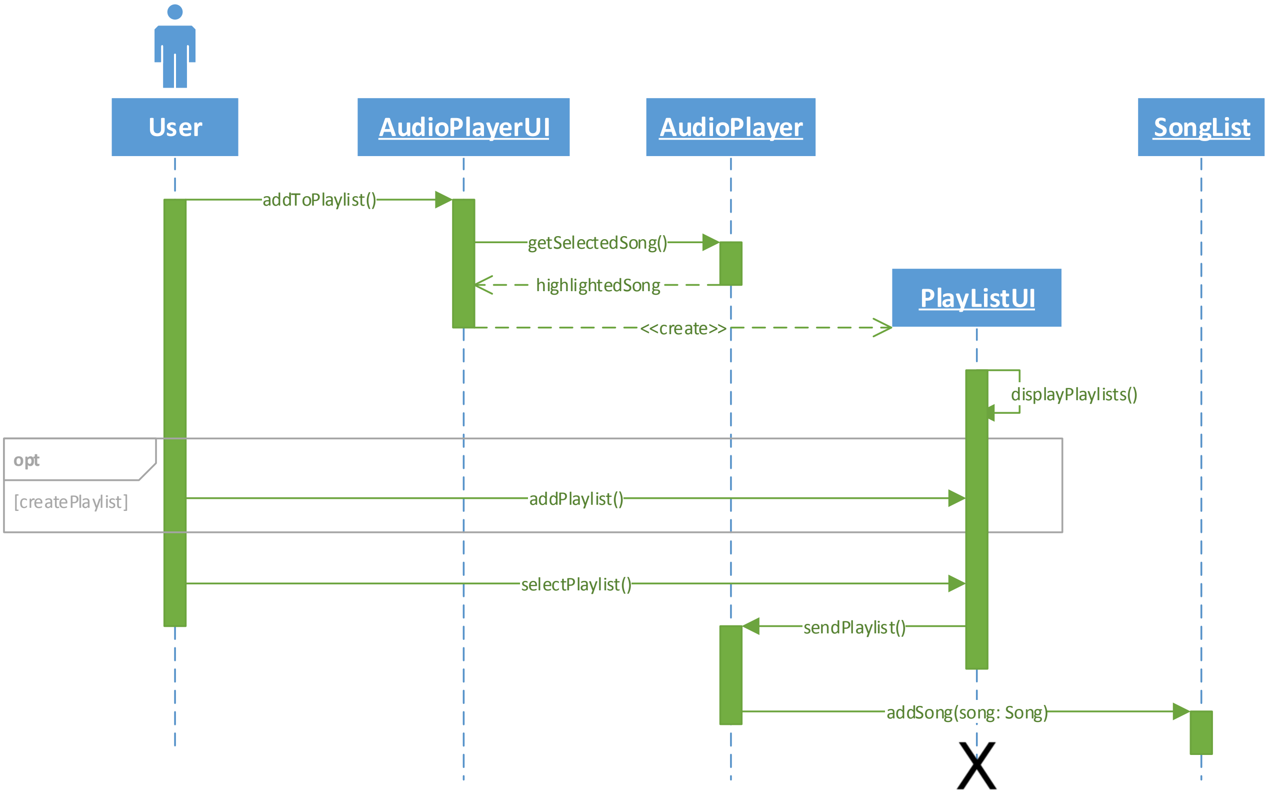
Use Case #2	Search Music
Goal in Context	To allow the user to search for a song
Scope	The Butterfly system
Level	Summary
Primary Actor	User
Preconditions	System is open and running
Minimal Guarantee	Tells user that song cannot be found
Success Guarantee	Songs matching the keywords appear to the user
Trigger	Typing in the search box
Description Step	
1	The user types keywords into the search field
2	The system searches the keywords in the list of songs
3	The system filters matching songs and displays them to the user

Jake Wheeler, Nate Christiansen, Nick Kapty


User
AudioPlayerUI
AudioPlayer
SongList
Song
searchLibrary(keyword: string)
searchLibrary(keyword:string)
searchList(keyword: string)
loop
for each song in library
matchKeyword(keyword:string)
matchedSongs
displayToUser()

Jake Wheeler, Nate Christiansen, Nick Kapty

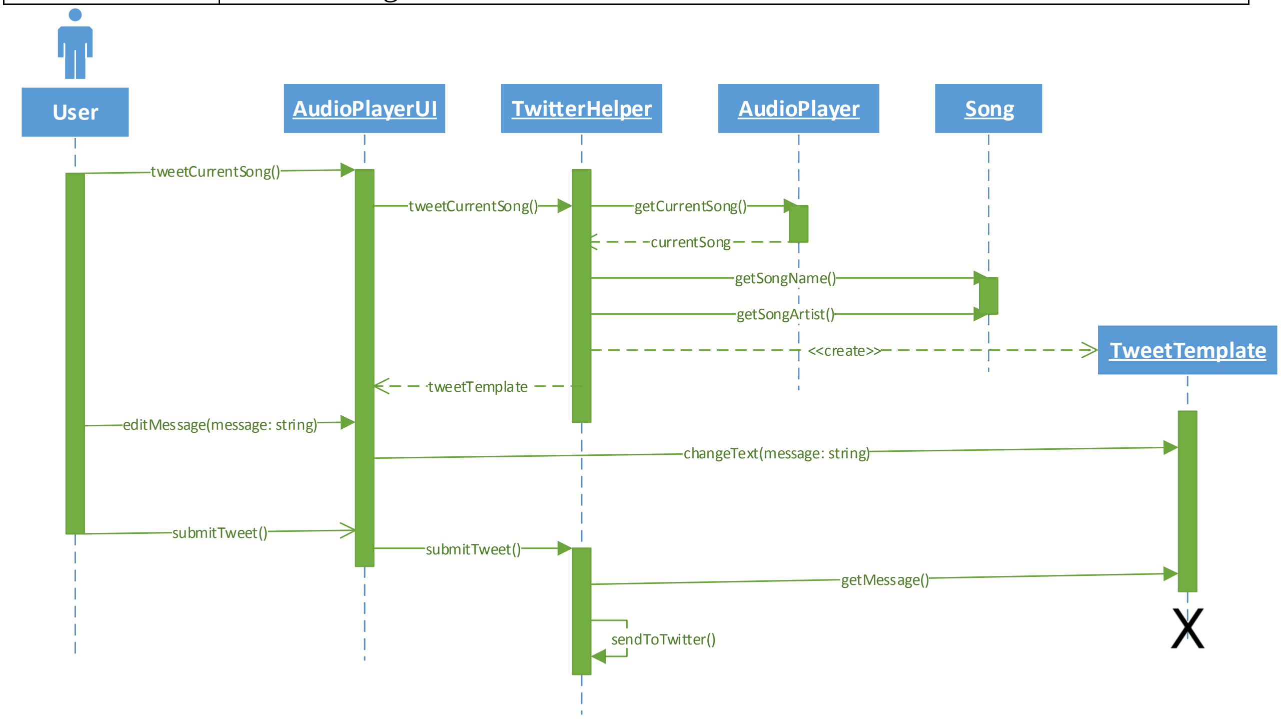
Use Case #3	Create Playlist
Goal in Context	To allow the user to create custom playlists
Scope	The Butterfly system
Level	Summary
Primary Actor	User
Preconditions	The user has at least one song
Minimal Guarantee	Nothing happens
Success Guarantee	The song is added to the playlist
Trigger	Right click, add to playlist
Description Step	
1	The user right-clicks the song
2	The user selects “add to playlist” option
3	The system prompts the user to create playlist or choose playlist
4	The user selects a playlist
5	The system adds the song to the playlist
Extension Step	
4a	The use selects create playlist
	A1: The system adds a new playlist to the list of playlists
	A2: The user then selects a playlist to add the song to



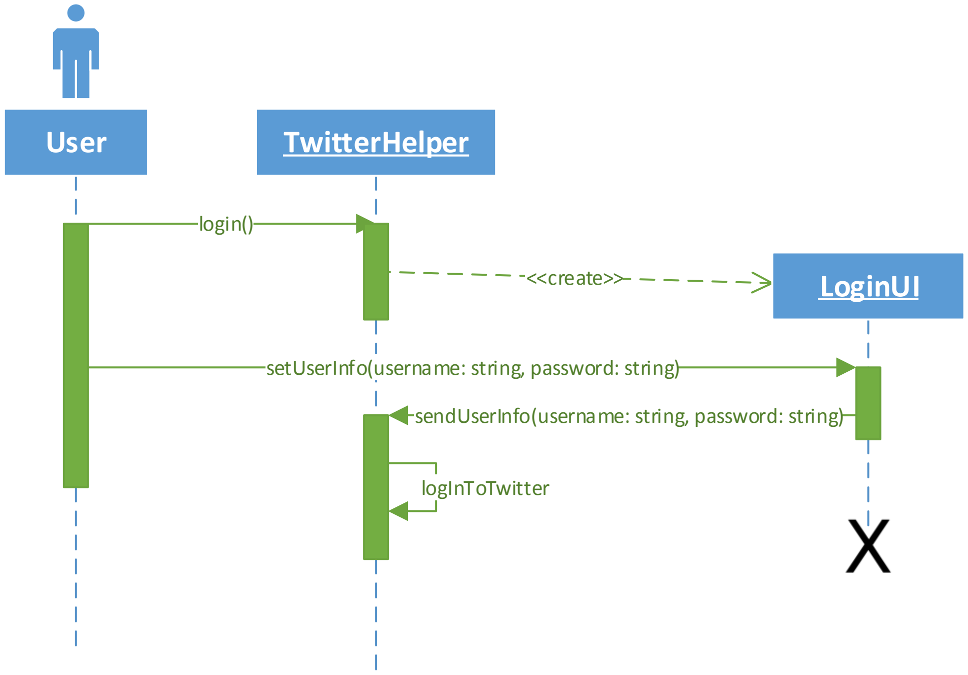
Jake Wheeler, Nate Christiansen, Nick Kapty

Jake Wheeler, Nate Christiansen, Nick Kapty

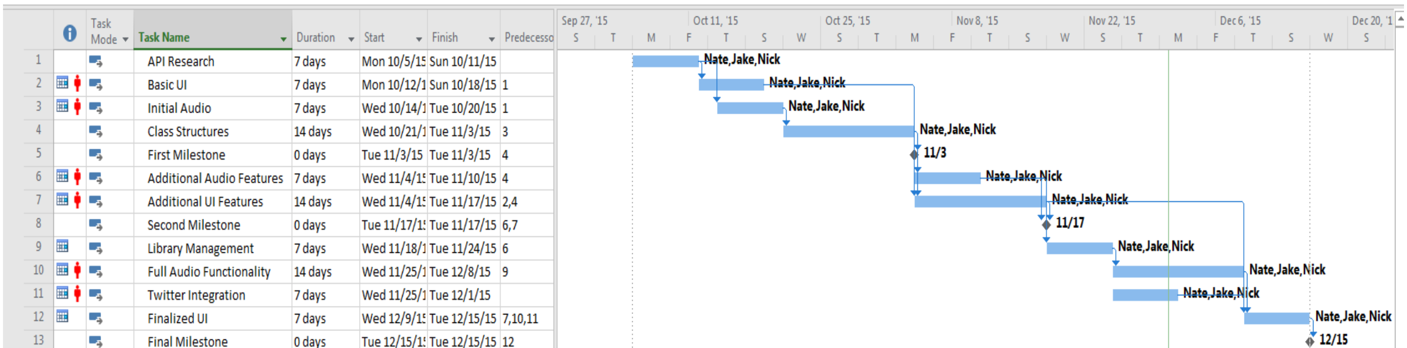
Use Case #4	Tweet Song
Goal in Context	To allow the user to tweet about currently chosen song
Scope	The Butterfly system
Level	Summary
Primary Actor	User
Preconditions	Song is playing, user has twitter account
Minimal Guarantee	Nothing happens
Success Guarantee	The tweet is posted by the user
Trigger	User selects to tweet about current song
Description Step	
1	System displays window that allows user to edit tweet template
2	The user submits the tweet
3	The system posts the tweet to the user's timeline
Extension Step	
1a	The user is not logged into Twitter
	A1: The Login subfunction is called



Use Case #8	Login
Goal in Context	To allow the user to login to Twitter
Scope	The Butterfly system
Level	Sub-function
Primary Actor	User
Preconditions	System is open and running, internet connection available
Minimal Guarantee	Nothing happens
Success Guarantee	User is logged into Twitter
Trigger	User attempts to login or access Twitter functionality
Description Step	
1	The user enters their account username and password
2	The system logs the user into Twitter

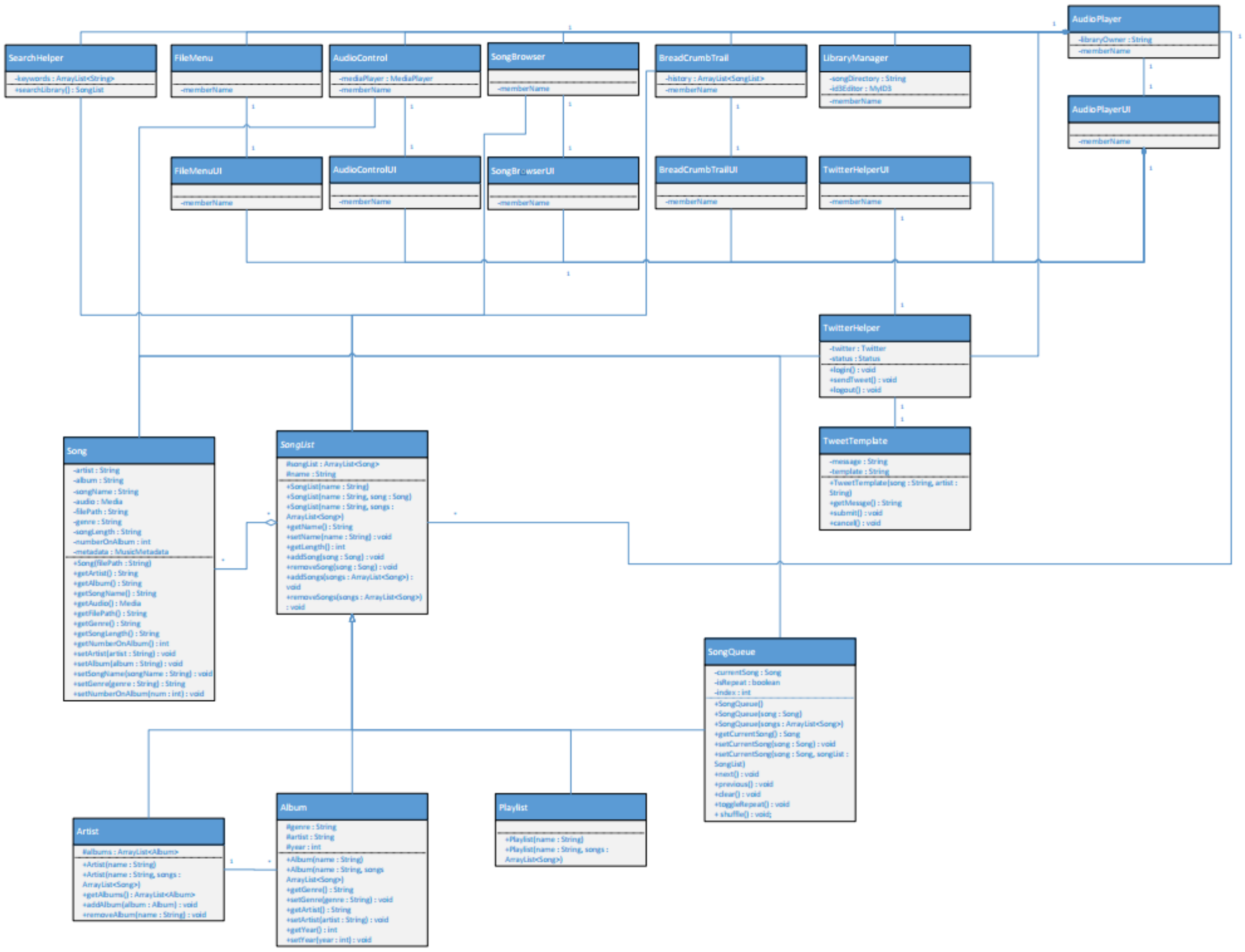

User
TwitterHelper
login()
<<create>>
LoginUI
setUserInfo(username: string, password: string)
sendUserInfo(username: string, password: string)
logInToTwitter
X

10. Project Plan

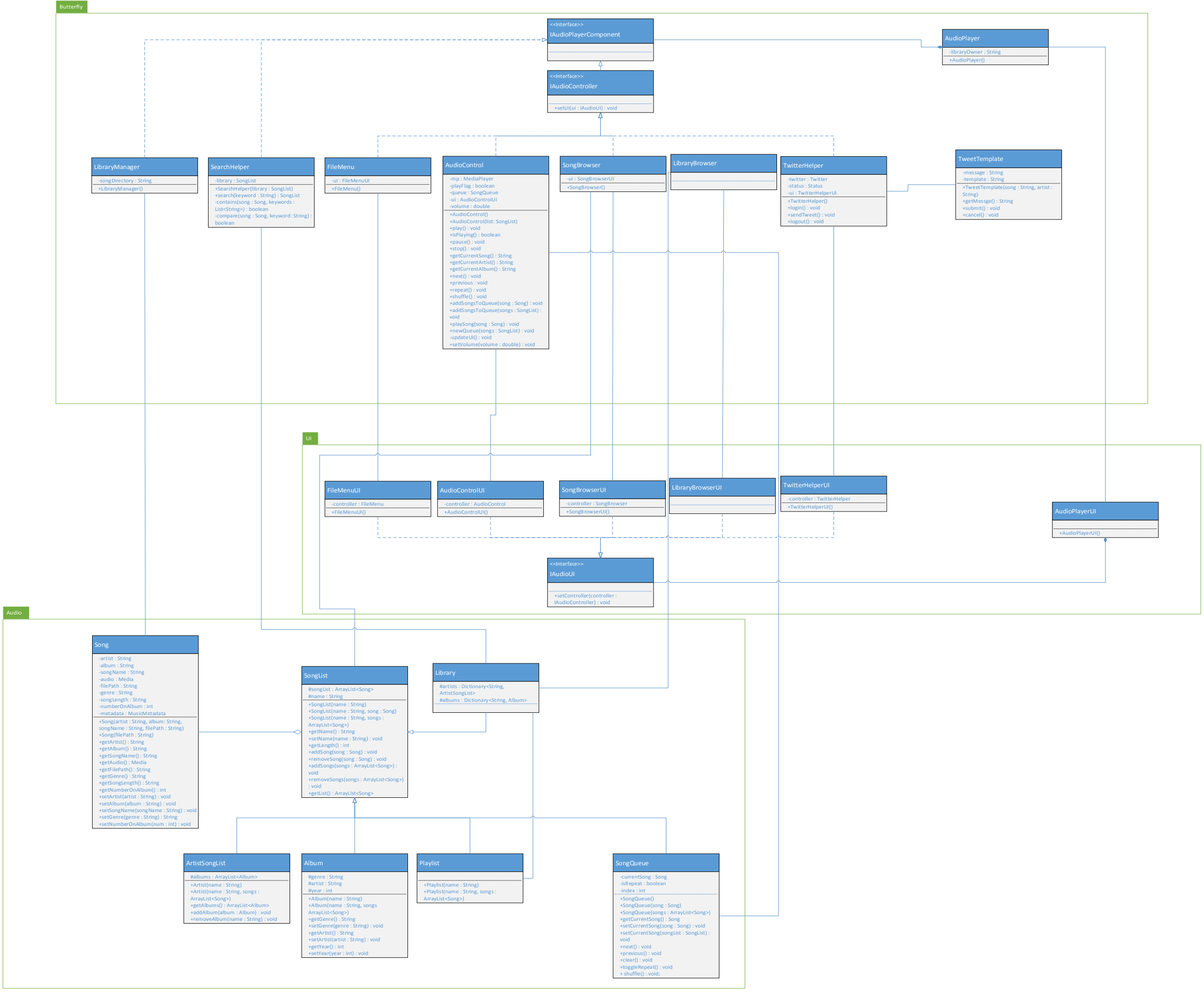


		Task Mode	Task Name	Duration	Start	Finish	Predecesso
1			API Research	7 days	Mon 10/5/15	Sun 10/11/15	
2			Basic UI	7 days	Mon 10/12/1	Sun 10/18/15	1
3			Initial Audio	7 days	Wed 10/14/1	Tue 10/20/15	1
4			Class Structures	14 days	Wed 10/21/1	Tue 11/3/15	3
5			First Milestone	0 days	Tue 11/3/15	Tue 11/3/15	4
6			Additional Audio Features	7 days	Wed 11/4/15	Tue 11/10/15	4
7			Additional UI Features	14 days	Wed 11/4/15	Tue 11/17/15	2,4
8			Second Milestone	0 days	Tue 11/17/15	Tue 11/17/15	6,7
9			Library Management	7 days	Wed 11/18/1	Tue 11/24/15	6
10			Full Audio Functionality	14 days	Wed 11/25/1	Tue 12/8/15	9
11			Twitter Integration	7 days	Wed 11/25/1	Tue 12/1/15	
12			Finalized UI	7 days	Wed 12/9/15	Tue 12/15/15	7,10,11
13			Final Milestone	0 days	Tue 12/15/15	Tue 12/15/15	12

11. Class Diagram 1 (Domain Concepts)



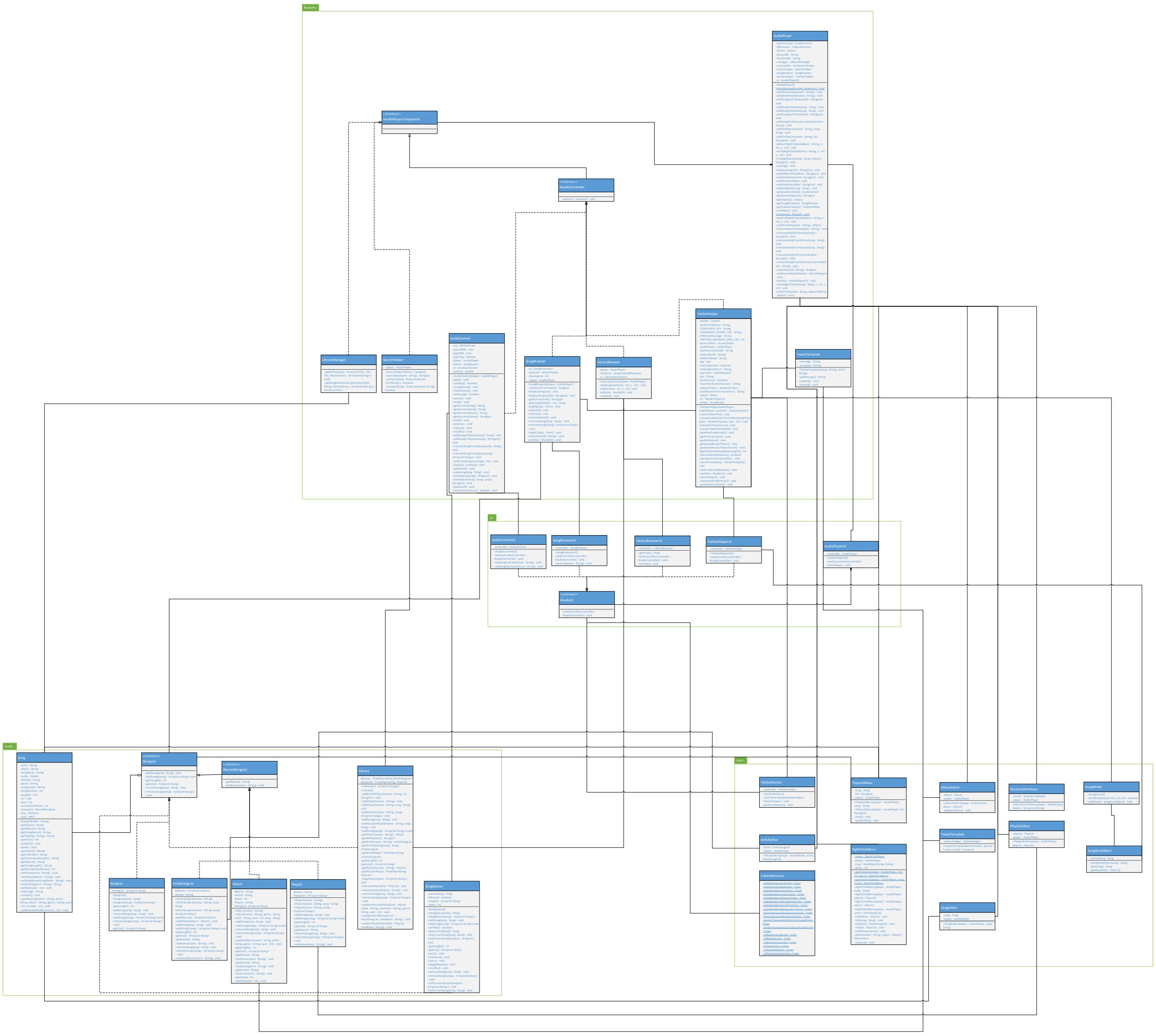
12. Class Diagram 2 (System Domain Model)


Butterfly
<<Interface>>
IAudioPlayerComponent
AudioPlayer
-libraryOwner : String
+AudioPlayer()
<<Interface>>
IAudioController
+setUI(ui : IAudioUI) : void
LibraryManager
-songDirectory : String
+LibraryManager()
SearchHelper
-library : SongList
+SearchHelper(library : SongList)
+search(keyword : String) : SongList
-contains(song : Song, keywords : List<String>) : boolean
-compare(song : Song, keyword: String) : boolean
FileMenu
-ui : FileMenuUI
+FileMenu()
AudioControl
-mp : MediaPlayer
-playFlag : boolean
-queue : SongQueue
-ui : AudioControlUI
-volume : double
+AudioControl()
+AudioControl(list: SongList)
+play() : void
+isPlaying() : boolean
+pause() : void
+stop() : void
+getCurrentSong() : String
+getCurrentArtist() : String
+getCurrentAlbum() : String
+next() : void
+previous : void
+repeat() : void
+shuffle() : void
+addSongsToQueue(song : Song) : void
+addSongsToQueue(songs : SongList) : void
+playSong(song : Song) : void
+newQueue(songs : SongList) : void
-updateUI() : void
+setVolume(volume : double) : void
SongBrowser
-ui : SongBrowserUI
+SongBrowser()
LibraryBrowser
TwitterHelper
-twitter : Twitter
-status : Status
-ui : TwitterHelperUI
+TwitterHelper()
+login() : void
+sendTweet() : void
+logout() : void
TweetTemplate
-message : String
-template : String
+TweetTemplate(song : String, artist : String)
+getMessge() : String
+submit() : void
+cancel() : void
UI
FileMenuUI
-controller : FileMenu
+FileMenuUI()
AudioControlUI
-controller : AudioControl
+AudioControlUI()
SongBrowserUI
-controller : SongBrowser
+SongBrowserUI()
LibraryBrowserUI
TwitterHelperUI
-controller : TwitterHelper
+TwitterHelperUI()
AudioPlayerUI
+AudioPlayerUI()
<<Interface>>
IAudioUI
+setController(controller : IAudioController) : void
Audio
Song
-artist : String
-album : String
-songName : String
-audio : Media
-filePath : String
-genre : String
-songLength : String
-numberOnAlbum : int
-metadata : MusicMetadata
+Song(artist : String, album: String, songName : String, filePath : String)
+Song(filePath : String)
+getArtist() : String
+getAlbum() : String
+getSongName() : String
+getAudio() : Media
+getFilePath() : String
+getGenre() : String
+getSongLength() : String
+getNumberOnAlbum() : int
+setArtist(artist : String) : void
+setAlbum(album : String) : void
+setSongName(songName : String) : void
+setGenre(genre : String) : String
+setNumberOnAlbum(num : int) : void
SongList
#songList : ArrayList<Song>
#name : String
+SongList(name : String)
+SongList(name : String, song : Song)
+SongList(name : String, songs : ArrayList<Song>)
+getName() : String
+setName(name : String) : void
+getLength() : int
+addSong(song : Song) : void
+removeSong(song : Song) : void
+addSongs(songs : ArrayList<Song>) : void
+removeSongs(songs : ArrayList<Song>) : void
+getList() : ArrayList<Song>
Library
#artists : Dictionary<String, ArtistSongList>
#albums : Dictionary<String, Album>
ArtistSongList
#albums : ArrayList<Album>
+Artist(name : String)
+Artist(name : String, songs : ArrayList<Song>)
+getAlbums() : ArrayList<Album>
+addAlbum(album : Album) : void
+removeAlbum(name : String) : void
Album
#genre : String
#artist : String
#year : int
+Album(name : String)
+Album(name : String, songs ArrayList<Song>)
+getGenre() : String
+setGenre(genre : String) : void
+getArtist() : String
+setArtist(artist : String) : void
+getYear() : int
+setYear(year : int) : void
Playlist
+Playlist(name : String)
+Playlist(name : String, songs : ArrayList<Song>)
SongQueue
-currentSong : Song
-isRepeat : boolean
-index : int
+SongQueue()
+SongQueue(song : Song)
+SongQueue(songs : ArrayList<Song>)
+getCurrentSong() : Song
+setCurrentSong(song : Song) : void
+setCurrentSong(songList : SongList) : void
+next() : void
+previous() : void
+clear() : void
+toggleRepeat() : void
+ shuffle() : void;

13. State Diagram

We did not create a state diagram for this project.

14. Class Diagram 3 (Final System Model)



15. Project Deployment

<u>How to setup the music player portion</u>:

1.) Butterfly can be downloaded from here: https://github.com/jakewheeler/SWENG411-Butterfly

2.) Butterfly is distributed in a zipped folder. Unzip the folder.

3.) Put all of the contents from that into a single folder anywhere you like.

4.) Run the Butterfly.jar file.

5.) Butterfly should be open. The first step while it is running would be to click on the File tab in the upper-left corner. Click "Manage Music Folders" tab.

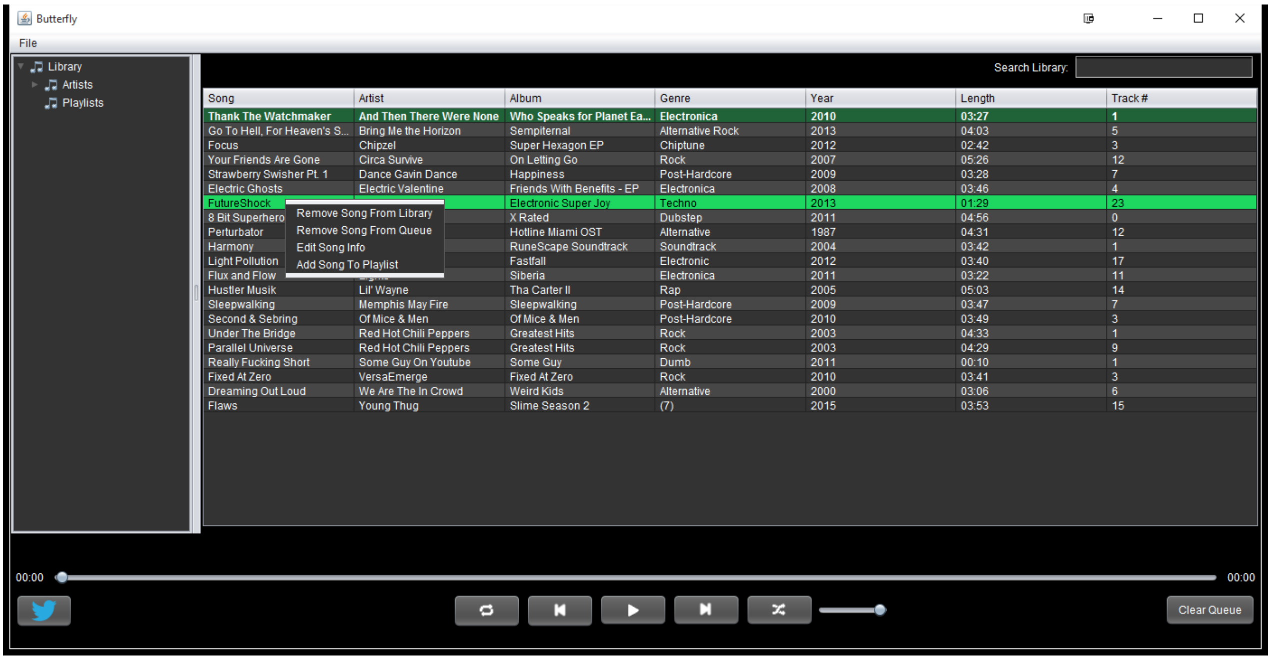
6.) Click "Add folder" to select a directory containing MP3 files and press Open.

7.) You should see the directory's file path that you have selected. Hit OK and your library should populate with the MP3s from the chosen directory

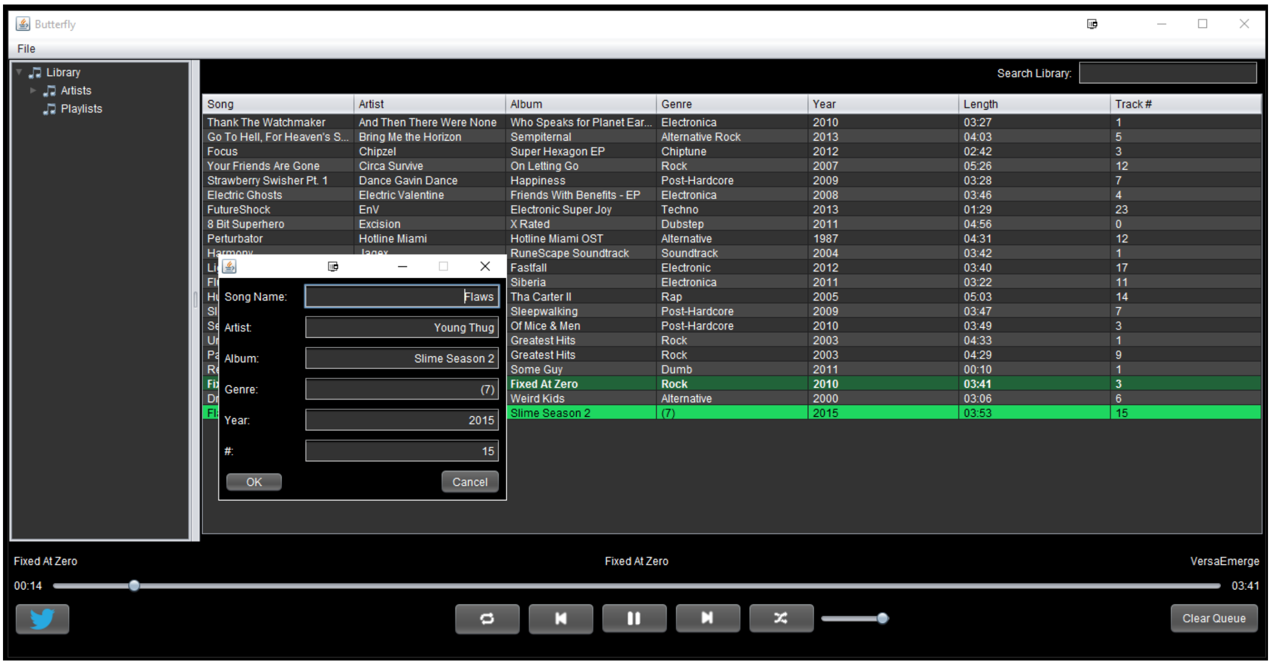
How to setup the Twitter integration:

1.) Press the Twitter button in the lower left-hand corner.
2.) Your default browser will open with a link to Twitter. If you are not already logged in to Twitter, you will have to do so.
3.) Once logged in, Twitter will supply a PIN. Copy and paste this PIN into the window that pops up in Butterfly. Once entered, hit OK.
4.) You will now see a window showing details about your song. You may enter any text that you like within the limited amount of characters allowed. Hit "Post Tweet" to send your tweet to your timeline or press "Cancel" to go back to Butterfly.

16. System Screenshots



Song	Artist	Album	Genre	Year	Length	Track #
Thank The Watchmaker	And Then There Were None	Who Speaks for Planet Ea...	Electronica	2010	03:27	1
Go To Hell, For Heaven's S...	Bring Me the Horizon	Sempiternal	Alternative Rock	2013	04:03	5
Focus	Chipzel	Super Hexagon EP	Chiptune	2012	02:42	3
Your Friends Are Gone	Circa Survive	On Letting Go	Rock	2007	05:26	12
Strawberry Swisher Pt. 1	Dance Gavin Dance	Happiness	Post-Hardcore	2009	03:28	7
Electric Ghosts	Electric Valentine	Friends With Benefits - EP	Electronica	2008	03:46	4
FutureShock		Electronic Super Joy	Techno	2013	01:29	23
8 Bit Superhero		X Rated	Dubstep	2011	04:56	0
Perturbator		Hotline Miami OST	Alternative	1987	04:31	12
Harmony		RuneScape Soundtrack	Soundtrack	2004	03:42	1
Light Pollution		Fastfall	Electronic	2012	03:40	17
Flux and Flow		Siberia	Electronica	2011	03:22	11
Hustler Musik	Lil' Wayne	Tha Carter II	Rap	2005	05:03	14
Sleepwalking	Memphis May Fire	Sleepwalking	Post-Hardcore	2009	03:47	7
Second & Sebring	Of Mice & Men	Of Mice & Men	Post-Hardcore	2010	03:49	3
Under The Bridge	Red Hot Chili Peppers	Greatest Hits	Rock	2003	04:33	1
Parallel Universe	Red Hot Chili Peppers	Greatest Hits	Rock	2003	04:29	9
Really Fucking Short	Some Guy On Youtube	Some Guy	Dumb	2011	00:10	1
Fixed At Zero	VersaEmerge	Fixed At Zero	Rock	2010	03:41	3
Dreaming Out Loud	We Are The In Crowd	Weird Kids	Alternative	2000	03:06	6
Flaws	Young Thug	Slime Season 2	(7)	2015	03:53	15



Song	Artist	Album	Genre	Year	Length	Track #
Thank The Watchmaker	And Then There Were None	Who Speaks for Planet Ear...	Electronica	2010	03:27	1
Go To Hell, For Heaven's S...	Bring Me the Horizon	Sempiternal	Alternative Rock	2013	04:03	5
Focus	Chipzel	Super Hexagon EP	Chiptune	2012	02:42	3
Your Friends Are Gone	Circa Survive	On Letting Go	Rock	2007	05:26	12
Strawberry Swisher Pt. 1	Dance Gavin Dance	Happiness	Post-Hardcore	2009	03:28	7
Electric Ghosts	Electric Valentine	Friends With Benefits - EP	Electronica	2008	03:46	4
FutureShock	EnV	Electronic Super Joy	Techno	2013	01:29	23
8 Bit Superhero	Excision	X Rated	Dubstep	2011	04:56	0
Perturbator	Hotline Miami	Hotline Miami OST	Alternative	1987	04:31	12
Harmony	Jagex	RuneScape Soundtrack	Soundtrack	2004	03:42	1
Li		Fastfall	Electronic	2012	03:40	17
Fl		Siberia	Electronica	2011	03:22	11
Hu		Tha Carter II	Rap	2005	05:03	14
Sl		Sleepwalking	Post-Hardcore	2009	03:47	7
Se		Of Mice & Men	Post-Hardcore	2010	03:49	3
Ur		Greatest Hits	Rock	2003	04:33	1
Pa		Greatest Hits	Rock	2003	04:29	9
Re		Some Guy	Dumb	2011	00:10	1
Fi		Fixed At Zero	Rock	2010	03:41	3
Dr		Weird Kids	Alternative	2000	03:06	6
Fl		Slime Season 2	(7)	2015	03:53	15

Song Name: Flaws
Artist: Young Thug
Album: Slime Season 2
Genre: (7)
Year: 2015
#: 15

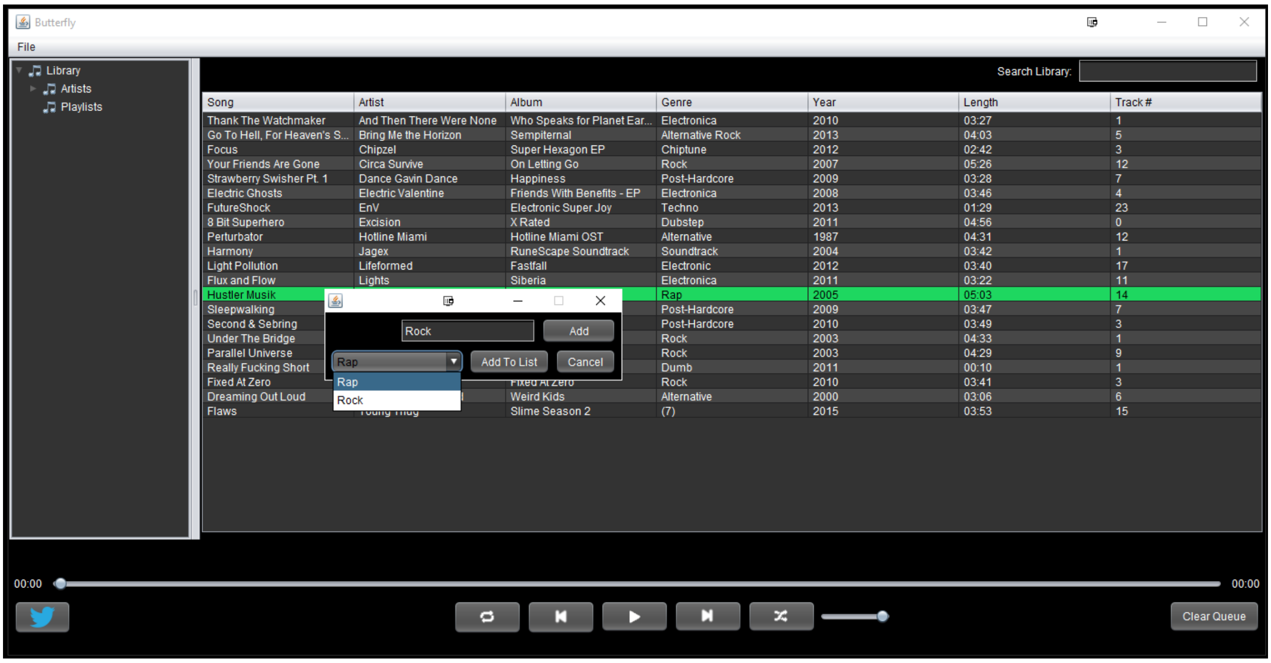
Fixed At Zero

Fixed At Zero

VersaEmerge

00:14

03:41



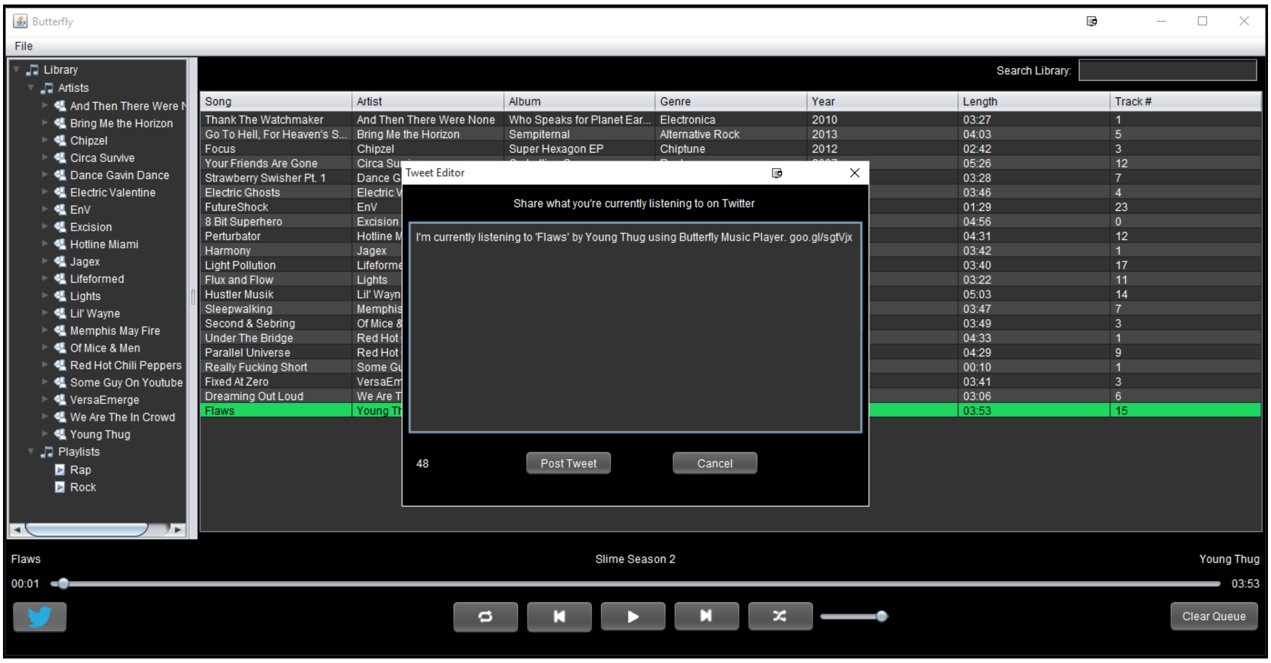
Song	Artist	Album	Genre	Year	Length	Track #
Thank The Watchmaker	And Then There Were None	Who Speaks for Planet Ear...	Electronica	2010	03:27	1
Go To Hell, For Heaven's S...	Bring Me the Horizon	Sempiternal	Alternative Rock	2013	04:03	5
Focus	Chipzel	Super Hexagon EP	Chiptune	2012	02:42	3
Your Friends Are Gone	Circa Survive	On Letting Go	Rock	2007	05:26	12
Strawberry Swisher Pt. 1	Dance Gavin Dance	Happiness	Post-Hardcore	2009	03:28	7
Electric Ghosts	Electric Valentine	Friends With Benefits - EP	Electronica	2008	03:46	4
FutureShock	EnV	Electronic Super Joy	Techno	2013	01:29	23
8 Bit Superhero	Excision	X Rated	Dubstep	2011	04:56	0
Perturbator	Hotline Miami	Hotline Miami OST	Alternative	1987	04:31	12
Harmony	Jagex	RuneScape Soundtrack	Soundtrack	2004	03:42	1
Light Pollution	Lifeformed	Fastfall	Electronic	2012	03:40	17
Flux and Flow	Lights	Siberia	Electronica	2011	03:22	11
Hustler Musik			Rap	2005	05:03	14
Sleepwalking			Post-Hardcore	2009	03:47	7
Second & Sebring			Post-Hardcore	2010	03:49	3
Under The Bridge			Rock	2003	04:33	1
Parallel Universe			Rock	2003	04:29	9
Really Fucking Short			Dumb	2011	00:10	1
Fixed At Zero		Fixed At Zero	Rock	2010	03:41	3
Dreaming Out Loud		Weird Kids	Alternative	2000	03:06	6
Flaws	Young Thug	Slime Season 2	(7)	2015	03:53	15

Rock

Rap
- Rap
- Rock

00:00

00:00

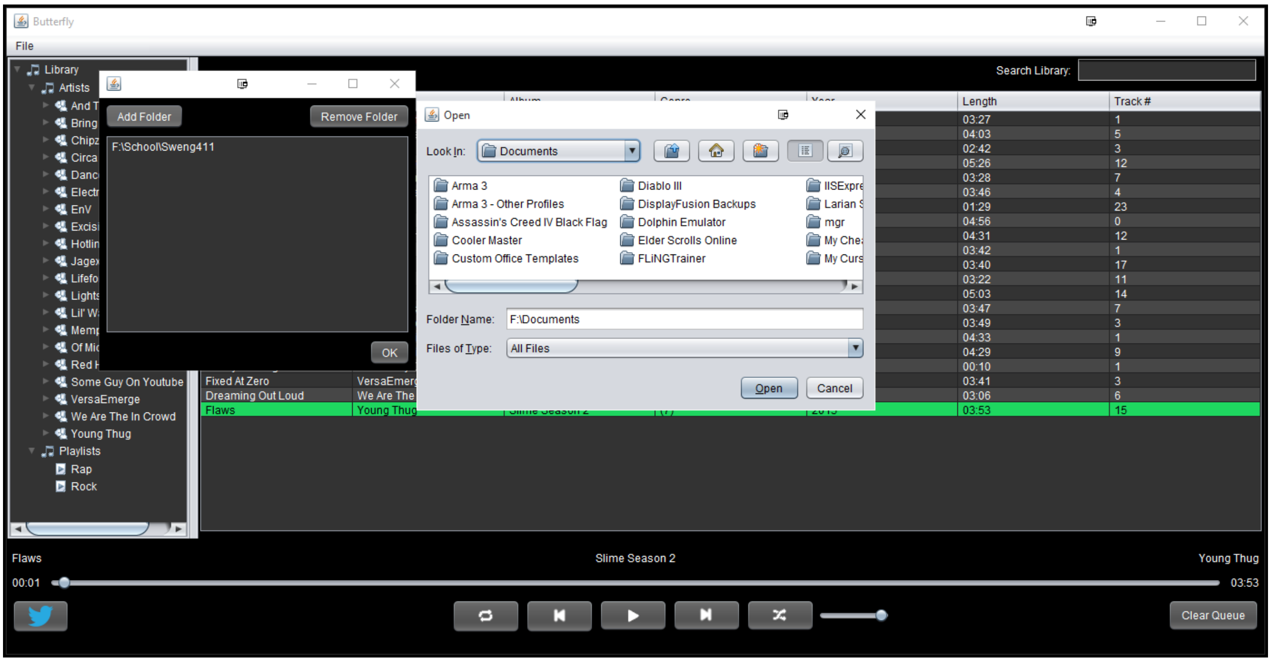


Song	Artist	Album	Genre	Year	Length	Track #
Thank The Watchmaker	And Then There Were None	Who Speaks for Planet Ear...	Electronica	2010	03:27	1
Go To Hell, For Heaven's S...	Bring Me the Horizon	Sempiternal	Alternative Rock	2013	04:03	5
Focus	Chipzel	Super Hexagon EP	Chiptune	2012	02:42	3
Your Friends Are Gone	Circa Su				05:26	12
Strawberry Swisher Pt. 1	Dance G				03:28	7
Electric Ghosts	Electric V				03:46	4
FutureShock	EnV				01:29	23
8 Bit Superhero	Excision				04:56	0
Perturbator	Hotline M				04:31	12
Harmony	Jagex				03:42	1
Light Pollution	Lifeforme				03:40	17
Flux and Flow	Lights				03:22	11
Hustler Musik	Lil' Wayn				05:03	14
Sleepwalking	Memphis				03:47	7
Second & Sebring	Of Mice &				03:49	3
Under The Bridge	Red Hot				04:33	1
Parallel Universe	Red Hot				04:29	9
Really Fucking Short	Some Gu				00:10	1
Fixed At Zero	VersaEm				03:41	3
Dreaming Out Loud	We Are T				03:06	6
Flaws	Young Th				03:53	15

Share what you're currently listening to on Twitter

I'm currently listening to 'Flaws' by Young Thug using Butterfly Music Player. goo.gl/sgtVjx

48



F:\School\Sweng411

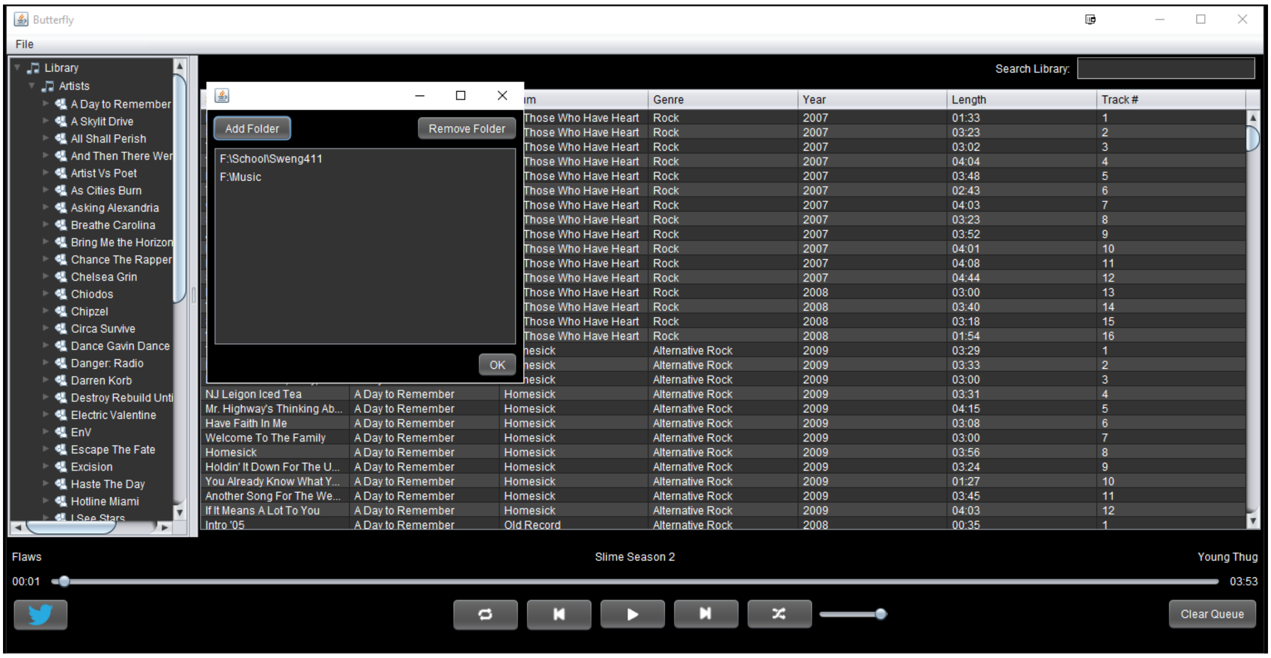
Look In: Documents

Arma 3
Arma 3 - Other Profiles
Assassin's Creed IV Black Flag
Cooler Master
Custom Office Templates

Folder Name: F:\Documents

Files of Type: All Files

Song	Artist	Album	Genre	Year	Length	Track #
					03:27	1
					04:03	5
					02:42	3
					05:26	12
					03:28	7
					03:46	4
					01:29	23
					04:56	0
					04:31	12
					03:42	1
					03:40	17
					03:22	11
					05:03	14
					03:47	7
					03:49	3
					04:33	1
					04:29	9
					00:10	1
Fixed At Zero	VersaEmerg				03:41	3
Dreaming Out Loud	We Are The				03:06	6
Flaws	Young Thug				03:53	15



Library
Artists
A Day to Remember
A Skylit Drive
All Shall Perish
And Then There Wer
Artist Vs Poet
As Cities Burn
Asking Alexandria
Breathe Carolina
Bring Me the Horizon
Chance The Rapper
Chelsea Grin
Chiodos
Chipzel
Circa Survive
Dance Gavin Dance
Danger: Radio
Darren Korb
Destroy Rebuild Unti
Electric Valentine
EnV
Escape The Fate
Excision
Haste The Day
Hotline Miami
I See Stars

F:\School\Sweng411
F:\Music

		m	Genre	Year	Length	Track #
		Those Who Have Heart	Rock	2007	01:33	1
		Those Who Have Heart	Rock	2007	03:23	2
		Those Who Have Heart	Rock	2007	03:02	3
		Those Who Have Heart	Rock	2007	04:04	4
		Those Who Have Heart	Rock	2007	03:48	5
		Those Who Have Heart	Rock	2007	02:43	6
		Those Who Have Heart	Rock	2007	04:03	7
		Those Who Have Heart	Rock	2007	03:23	8
		Those Who Have Heart	Rock	2007	03:52	9
		Those Who Have Heart	Rock	2007	04:01	10
		Those Who Have Heart	Rock	2007	04:08	11
		Those Who Have Heart	Rock	2007	04:44	12
		Those Who Have Heart	Rock	2008	03:00	13
		Those Who Have Heart	Rock	2008	03:40	14
		Those Who Have Heart	Rock	2008	03:18	15
		Those Who Have Heart	Rock	2008	01:54	16
		hesick	Alternative Rock	2009	03:29	1
		hesick	Alternative Rock	2009	03:33	2
		hesick	Alternative Rock	2009	03:00	3
NJ Leigon Iced Tea	A Day to Remember	Homesick	Alternative Rock	2009	03:31	4
Mr. Highway's Thinking Ab...	A Day to Remember	Homesick	Alternative Rock	2009	04:15	5
Have Faith In Me	A Day to Remember	Homesick	Alternative Rock	2009	03:08	6
Welcome To The Family	A Day to Remember	Homesick	Alternative Rock	2009	03:00	7
Homesick	A Day to Remember	Homesick	Alternative Rock	2009	03:56	8
Holdin' It Down For The U...	A Day to Remember	Homesick	Alternative Rock	2009	03:24	9
You Already Know What Y...	A Day to Remember	Homesick	Alternative Rock	2009	01:27	10
Another Song For The We...	A Day to Remember	Homesick	Alternative Rock	2009	03:45	11
If It Means A Lot To You	A Day to Remember	Homesick	Alternative Rock	2009	04:03	12
Intro '05	A Day to Remember	Old Record	Alternative Rock	2008	00:35	1

Flaws
Slime Season 2
Young Thug
00:01
03:53

17. Project Source Codes

Attached separately

18. Project Presentation Slides

Attached separately

19. Project Video Clips

Attached separately

20. Open Issues

Given enough time, we would have liked to implement a couple of additional features. We would have liked to give the user the ability to customize the appearance of the application. We also would have liked to save the user’s settings to a file and load them on startup. We would have liked to update the song time while dragging the bar to skip through the song.

21. Software Process Model

Our team used the agile method of development while working on Butterfly. We used this methodology so that we can continually increment on our design and add additional features as we develop and test.

22. Software Architecture Used

We used the MVC design architecture for Butterfly. Our model is the Song and SongList classes, the view is all of the UI elements, and the controllers are the systems that Butterfly uses.

23. Design Patterns

We used general hierarchy for the SongList portion of the project. We used the façade pattern for Butterfly's systems that interact. We used the singleton pattern for the RightClickMenu.

24. Key Design Decisions

We initially were going to have YouTube functionality with our music player, however we discovered that it is against YouTube's terms of service to pull audio from their videos.

25. Team Communications

Our team met once a week after labs on Fridays, and we also occasionally met and discussed the project throughout the week when necessary. In order to keep track of source code, we used Git version control.

26. Task Allocation and Responsibilities

We kept a to-do list in a folder on the repository, and communicated the tasks that each of us were working on so that no overlap occurred.

Jake Wheeler, Nate Christiansen, Nick Kapty

27. Lessons learned by Each Team Member

Nate Christiansen –

If I had another month to work on Butterfly, I would very much like to work on optimization of resources. I think that while everything works pretty well, there is probably a lot of improvements to be made on how everything is handled. There is likely many spots that are way more resource intensive than should be. Especially revolving around ISongLists and Songs. I would also have liked to add more features to the music player, such as: more settings, the ability to remove songs from the library and have them remain gone until added, the ability to delete songs from the computer, improved search functionality, and improved social functions. Finally I would like to have made a custom look and feel that could be applied to everything to achieve a more cohesive feel for the project.

I found the most difficult part of the project to be the system design. More specifically, design documentation. In terms of implementation, everything came pretty smooth I think. All of our ideas translated pretty well into code, and changing designs in the code was pretty easy. Butterfly was relatively easy overall. However, keeping the design documents up to date was a challenge, as well as constantly keeping track of system designs that are always changing as we discover better ways to implement things. Our class diagram is very large and has many interwoven classes, which leads to a painful experience when trying to add, remove, or update it.

Jake Wheeler –

If I had another month to work on Butterfly, some things I would like to work on more would be the user interface, saving user settings, supporting file types aside from MP3s, allowing the user to select custom colors to create their own themes, and expanding the Twitter feature.

I think our current project is very solid overall. I think that another month of work would do good things for the UI. I would like it to look slightly more consistent as well as throw around the idea of using a different, more modern look & feel. Another thing I would have liked to do is allow the user to see what other people are currently listening to within the Butterfly application.

Jake Wheeler, Nate Christiansen, Nick Kapty

I think that system design is the most difficult task. Our UML class diagram is very large and has been constantly updated during the development of Butterfly. I think that the design is the most difficult task because your team may decide on a design but later realize that the design will not work correctly or there is a better way to do something. I found it difficult to keep the design and code at the same state at all times.

Nick Kapty – Given another month to work on Butterfly, I would like to work on making the user interface as aesthetically pleasing as possible. Right now our interface is pretty dull. For example, the Twitter sharing feature could be blue instead of our default gray and black to match Twitter's look and feel. Right now, there are no music players that I know of that offer full customization of the colors of any and all windows in the application, so a menu dedicated to full customization is also an interesting possible feature.

The most difficult part of the project for me was split about evenly between system design and implementation. For design, it was difficult keeping our documentation and diagrams up to date based on our currently implemented code. We fell behind on updating our class diagram most specifically.

In terms of implementation, being a less experienced/skilled programmer than Jake and Nate left me confused and I fell behind on some elements of our implementation. As a result from my perspective the level of the system that we implemented was kind of difficult for me to understand in some places and I was unable to contribute to some parts as much as I would have liked to.