

Final Project Proposal

Name: Zhixin Liu

Project: Music Player

Basic Introduction:

The project is going to build a music player based on music volume recommending. Detailed, the application is focus on providing user with a bunch of online music volume where each volume would contain 10 songs in a same music style. User could choose different volume in various flavors to listen to music, collect his own favorite music or volumes, and edit personal information including user name, user image, etc. Basically, the program will contain 5 view controller, including VolumeChooseController, PlaySongController, VolumeDetailController, SongDetailController, and UserAccountController. Each view controller is a subclass of ViewController, implementing specific functions and build relation with other view controller. This program uses Core Data to build database which contains user information, volume information, and song information. For more specified, user information contains user name, profile photo, user favorite volume list, user favorite song list; volume information contains volume image, volume name, volume style, song list; song information contains song image, song name, artist information, including artist name and brief introduction.

Application Flow:

1. Once the users open the application, they will first see the VolumeChooseController. The VolumeChooseController shows all the volume collections, each collection provide the functions for users to play songs of the volume and see detail of the volume, namely this controller provides methods to access PlaySongController and VolumeDetailController.
2. After users choose a volume to play, they will go to PlaySongController. The PlaySongController shows the current playing information, and also provide methods for users to access SongDetailController of current playing song, and VolumeDetailController of current volume.
3. If users choose to see the detail of the current playing song in step 2, they will go to SongDetailController. The SongDetailController shows the song information including image, lyric and artist information. Also, this controller provide methods to access to PlaySongController.
4. If users choose to see the detail of the current volume in step 2 or step 1, they will then go to VolumeDetailController. The VolumeDetailController shows all volume information including song list, volume image. It also provides function to access to SongDetailController, PlaySongController.
5. UserAccountController shows the user's own information including name, profile photo, favorite volume collection, favorite song list, as well as providing user with methods to change personal profile photo, personal name, and access to SongDetailController and VolumeDetailController.

- Each controller excepts UserAccountController would implement the function in button to lead users go to UserAccountController.
- Each controller provides back key for users to return to the forward controller.

Music Source/Information:

I am going to using a free music source online, which provide various style.

This source provide a volume/channel source URL:

<http://www.douban.com/j/app/radio/channels>

On the meantime providing a series of songs in URL like:

<http://douban.fm/j/mine/playlist?channel=14>

More specific, I am going to write a network protocol to connect my application with these music source which are actually the JSON data, then extract the volume and song information including song name, song image, song style, song length and artist.

Last I will use NSArray to store all these data and import some of them into my Core database, here some means these song information that user add them into his favorite collection.

Core data structure:

Entity_1: user information, contains user favorite song list(Array), collection(Array), user name(String), user image(Binary)

Entity_2: volume information, contains volume name(String), volume style(String)

Entity_3: song information, contains song name(String), song length(Double), song image(Binary), lyric(String), artist(String)

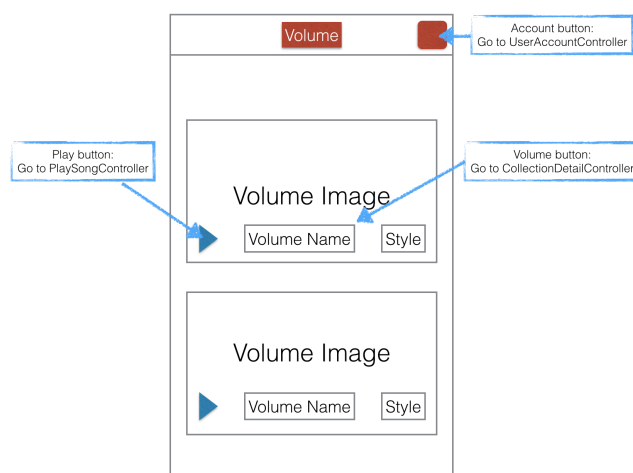
ViewController Description:

VolumeChooseController:

Components: UIImageView, Button, Label, View, CollectionView

Access to: VolumeDetailController, PlaySongController, UserAccountController

Sketch view:

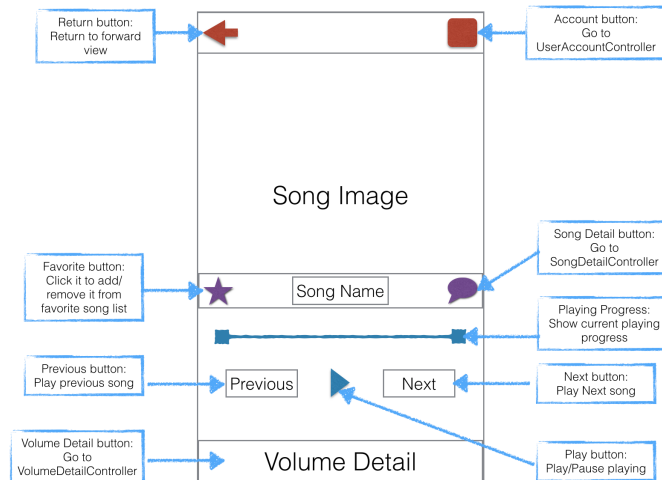


PlaySongController

Components: UIImageView, Button, Label, Progress View

Access to: VolumeDetailController, SongDetailController, UserAccountController

Sketch view:

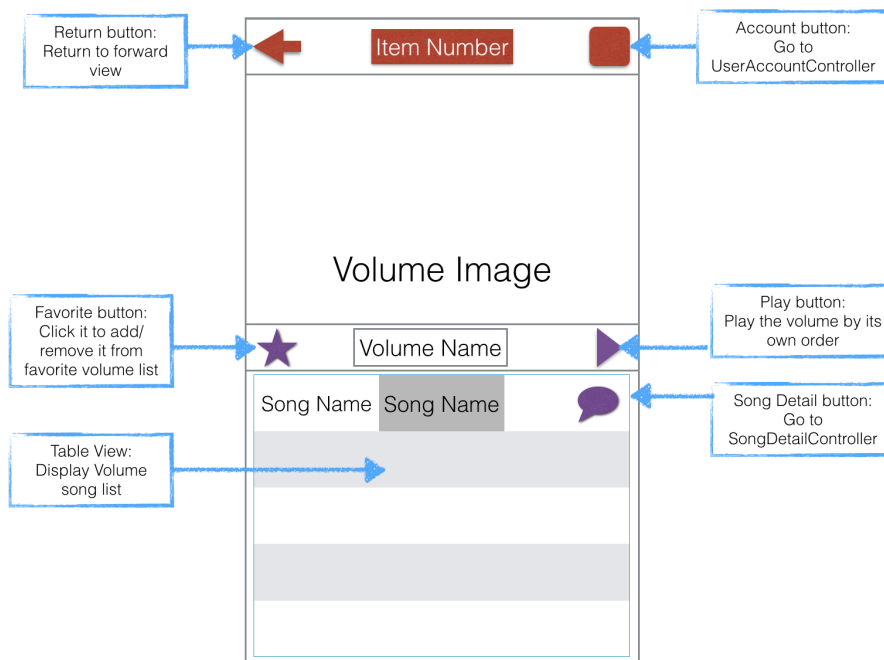


VolumeDetailController

Components: UIImageView, Button, TableView, Label

Access to: SongDetailController, UserAccountController, PlaySongController

Sketch view:

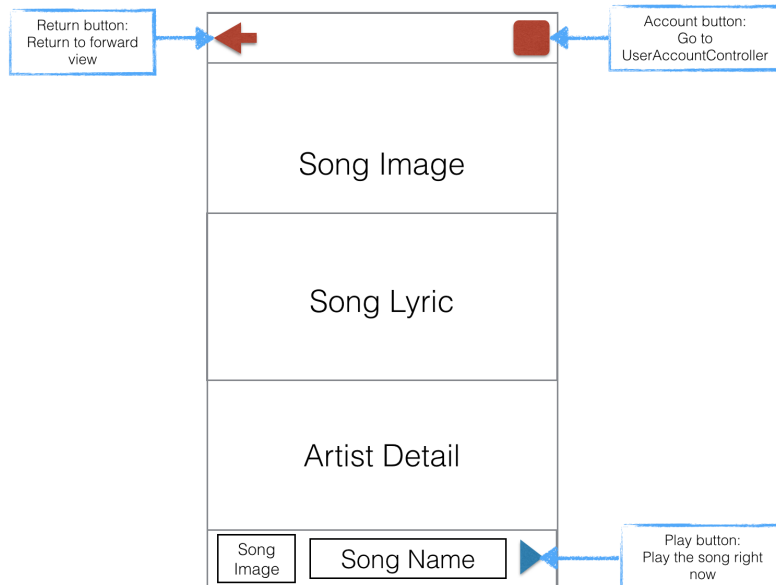


SongDetailController

Components: UIImageView, Button, Label, View

Access to: VolumeDetailController, PlaySongController, UserAccountController

Sketch view:

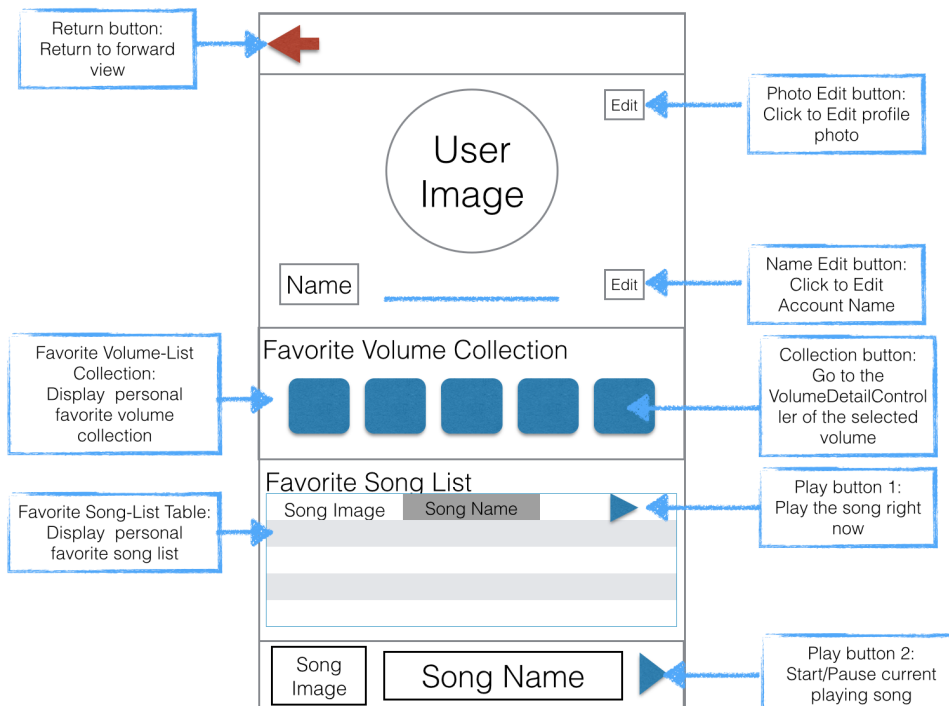


UserAccountController

Components: UIImageView, Button, CollectionView, TableView, Label, View

Access to: VolumeDetailController, PlaySongController, SongDetailController

Sketch view:



Optional Project Criteria:(Satisfied)

- Implement interesting gesture recognition to perform appropriate tasks and implement a custom view that performs non-trivial custom drawing
- Allow the user to interact with the device camera and/or photos
- Play sounds and/or movies that are relevant to the application (i.e., the application should require media playback for its functionality - random button noises won't count)