**44643 Mobile Computing – iOS**

**Assignment06**

**30 Points**

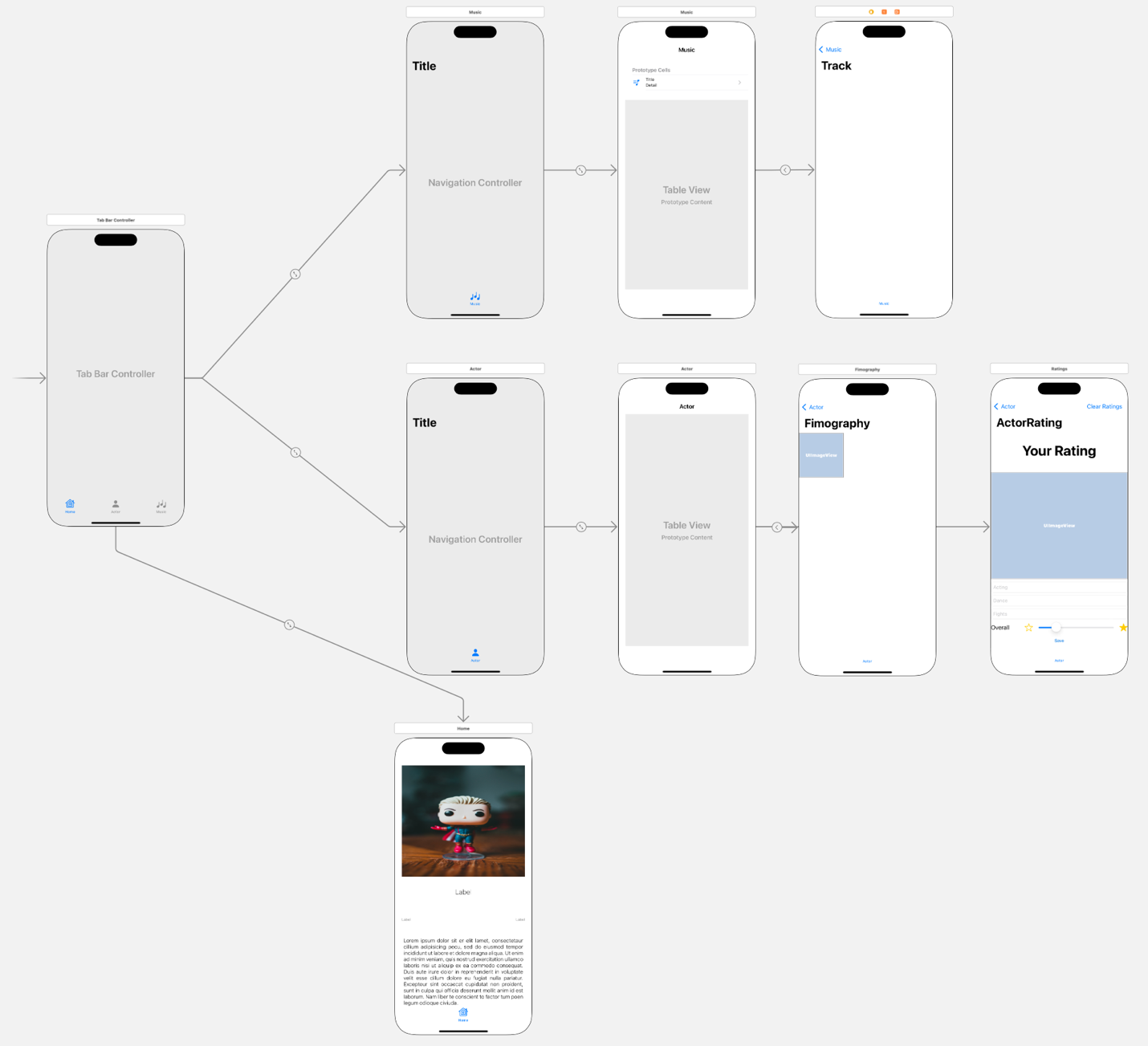
**Please follow the following instructions to complete this assignment.**

1. Open Xcode from the launchpad of your Mac.
2. Click on create a new Xcode project. Select the iOS template and click on the App application.
3. Click on next which will prompt you to choose options for the project.
4. Provide product name as **LastnameEntertainmentApp**, “**edu.nwmissouri**” for organization identifier, “**Storyboard**” as interface and “**Swift**” as the language.
5. Click on next and select an appropriate location to save your app and click on create. A blank project directory will be loaded.
6. From the project navigator click on “Main.storyboard” file, a blank mobile screen will be loaded in the Editor, where the required fields for an app need to be added.

**The View**

6

5



4

3

2

1

Figure 1 The View

1. Design the view for the app as shown in Figure 1.
   1. You will design a tab bar application with 3 tabs, namely, Home, Actor, and Music.
   2. Actor tab and Music tab are navigation applied tabs. Each of them consists of a single table view consisting of cells with style Subtitle. For Actor tab, use Detail Accessory indicator, and for Music tab, use Disclosure indicator.
   3. For the Actor tab’s table view’s cells, tapping on a cell will bring up the selected actor’s gallery. However, tapping on the accessory indicator will show actor’s rating screen, where a user can set his/her personal rating to the actor.
   4. In case of the Music tab, tap on the disclosure indicator/tap on a cell will play the track selected that is available on the YouTube.
      1. You need to have the YouTube Data API access to achieve this functionality.
   5. Use the exact same images and titles as shown in the Figure 1, for the tab bar items. Moreover, prefer large titles for the navigation bars.

Table 1 The View Controllers

|  |  |
| --- | --- |
| **Screen** | **View Controller** |
| 1 | HomeVC |
| 2 | ActorVC |
| 3 | FilmographyVC |
| 4 | ActorRatingVC |
| 5 | MusicVC |
| 6 | PlayMusicVC |

1. Refer to the Table 1 to create required classes and assign them to proper view controllers in the storyboard.

Table 2 The Auto Layout

|  |  |  |
| --- | --- | --- |
|  | |  |
| A picture containing text  Description automatically generated | Graphical user interface, text, application  Description automatically generated | |

Table 3 HomeVC's properties for stack views

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stack View** | **Axis** | **Alignment** | **Distribution** | **Spacing** |
| ProfileSV, BigContainerSV, OverallSV | Vertical | Fill | Fill | Standard |
| ContactSV | Horizontal | Fill | Fill Equally | Standard |
| ImageSV, InputSV | Verticcal | Fill | Fill Equally | Standard |

1. Apply auto layout to the app as per the details provided in Tables 2 and 3.

**The Model**

1. Create the following structures in your model file and create required number of instances for actors and music tracks using their respective initializers.
   1. Provide at least 5 actors and 5 music tracks.

|  |  |
| --- | --- |
|  |  |

**The Controller**

1. HomeVC

Table 4: UI elements configuration for HomeVC

|  |  |  |
| --- | --- | --- |
| **UI element** | **Purpose** | **Outlet/action name** |
| 3 labels | To display user’s full name | fullNameLBL |
| To display user’s phone number | emailLBL |
| To display user’s email | phoneLBL |
| 1 text view | To display user’s bio | bioTextView |
| 1 image view | To display user’s image | profileIV |

* 1. Create required outlets as per Table 4.
  2. When HomeVC loads, display all the outlets information.
  3. Set the following properties to the profileIV.
     1. Corner radius to 20.0
     2. Masks to bounds to true.
     3. Border with to 2.0
     4. Border color to (Red: 0, Green: 103, Blue: 71)

1. ActorVC

Table 5: UI elements configuration for ActorVC

|  |  |  |
| --- | --- | --- |
| **UI element** | **Purpose** | **Outlet/action name** |
| 1 table view | To display list of actors | actorTableView |

* 1. Create required outlets as per Table 5.
  2. Design a custom table view cell via a .xib file to show actors’ information. Each cell should have an image view to display the actor’s icon, and a couple of labels to show actor’s name and years active details.
     1. Corner radius to 5.0
     2. Clips to bounds to true.
     3. Border with to 1.0
     4. Border color to black.
  3. Load up list of actors with required animations and perform appropriate segues.
     1. On tapping the cell, show actor’s filmography.
     2. On tapping the cell’s accessory indicator, show actor’s rating screen.

1. FilmographyVC

Table 6: UI elements configuration for FilmographyVC

|  |  |  |
| --- | --- | --- |
| **UI element** | **Purpose** | **Outlet/action name** |
| 1 collection view | To display an actor’s movies gallery | moviesCV |

* 1. Create required outlets as per Table 6.
  2. Create a Cocoa touch class PosterCVC that is a subclass of UICollectionViewCell. Set PosterCVC as the moviesCV’s cell type.
     1. PosterCVC consists of an image view with outlet name moviePosterIV.
  3. Use PosterCVC type while dequeuing reusable cell in the collection view.

1. ActorRatingVC

Table 6: UI elements configuration for ActorRatingVC

|  |  |  |
| --- | --- | --- |
| **UI element** | **Purpose** | **Outlet/action name** |
| 1 label | To display “Your Rating” | n/a |
| 1 image view | To show actor’s image | actorIMG |
| 3 text fields | To read and update actor’s acting ability | actingTF |
| To read and update actor’s dancing ability | danceTF |
| To read and update actor’s fight sequences’ ability | actionEpisodesTF |
| 1 slider | To set overall performance rating | overallPerformanceSlider |
| 2 buttons | UIButton Save | save(\_:) |
| UIBarButtonItem Clear Ratings | clearRatings(\_:) |

* 1. Create required outlets as per Table 6.
  2. Set the following properties to the profileIV.
     1. Corner radius to 20.0
     2. Clips to bounds to true.
     3. Border with to 2.0
     4. Border color to black.
  3. Interacting with any of the text fields will present a mini picker view that pops up from the bottom of the screen. Selecting a row in the picker view will update the appropriate text field’s content.
  4. Save button will save the personal choices given to an actor.
  5. Clear ratings button will clear everything and resets all the personal choices.

1. MusicVC

Table 7: UI elements configuration for FilmographyVC

|  |  |  |
| --- | --- | --- |
| **UI element** | **Purpose** | **Outlet/action name** |
| 1 table view | To display a list of music tracks | musicTV |

* 1. Create required outlets as per Table 7.
  2. Load up list of music tracks and perform appropriate segue on tapping the table view cell.
     1. Set cell’s image to system image “music.note.list”
  3. Add a leading swipe gesture to the table view cells such that the swipe action will enable the user to add a song to Favorites section.
     1. No duplicates in the Favorites section.

1. PlayMusicVC

Table 8: UI elements configuration for PlayMusicVC

|  |  |  |
| --- | --- | --- |
| **UI element** | **Purpose** | **Outlet/action name** |
| 1 YTPlayerView | To play a YouTube video | videoPlayerView |

* 1. Create required outlets as per Table 8.
  2. Play the selected YouTube video when the view is loaded.

1. Useful libraries
   1. Install ViewAnimator via Cocoa Pods.
   2. SDWebImage
      1. <https://github.com/SDWebImage/SDWebImage.git>
   3. YouTubeiOSPlayerHelper
      1. <https://github.com/youtube/youtube-ios-player-helper.git>

**Note:**

* Make sure to apply auto-layout to the app such that the responsive UI works across all iOS devices and in all orientation modes.
* Provide high resolution files, for instance @2x and @3x versions, of your images in case of not loading image content via URLs.
* When gathering your app assests (images) make sure they are copyright free.

**Sample Output (Video Link):**

<https://app.vidgrid.com/view/QdKvfjZm8SWP>