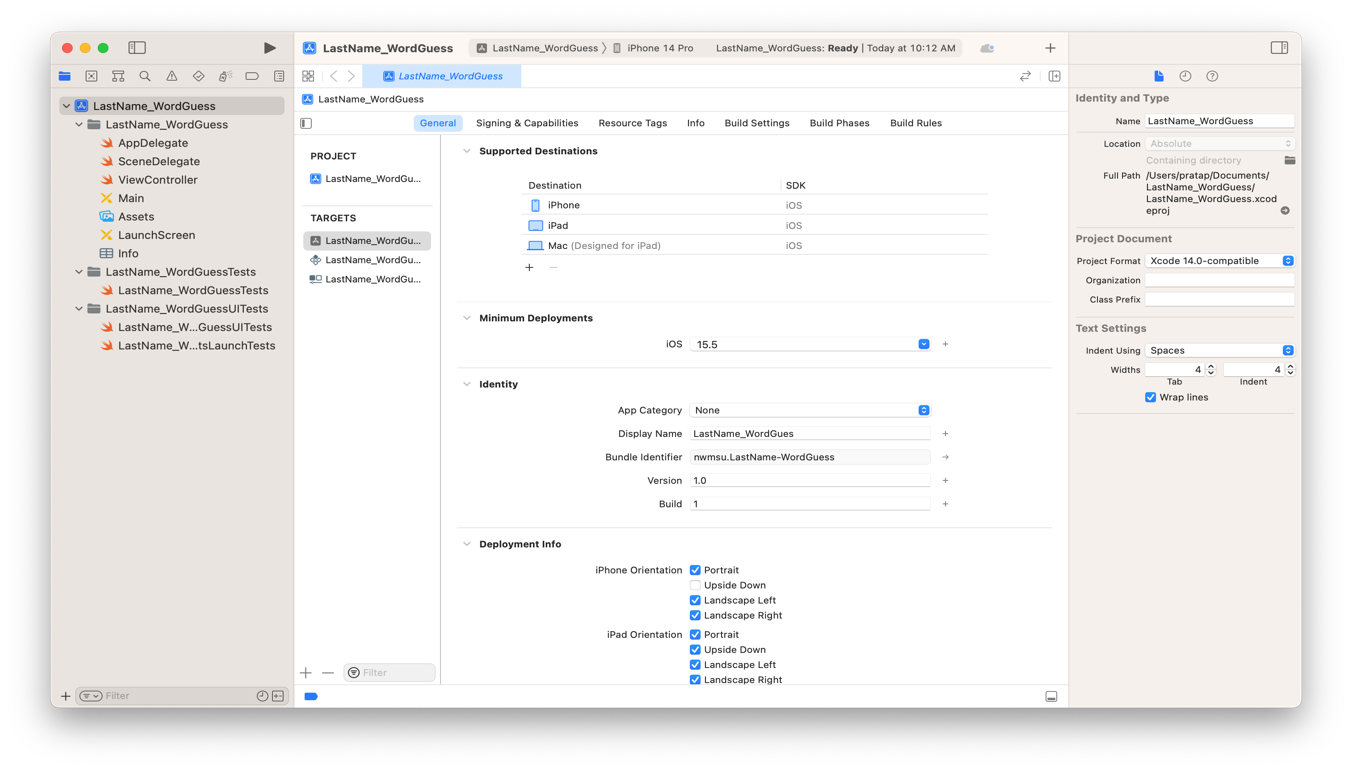
**Assignment 4**

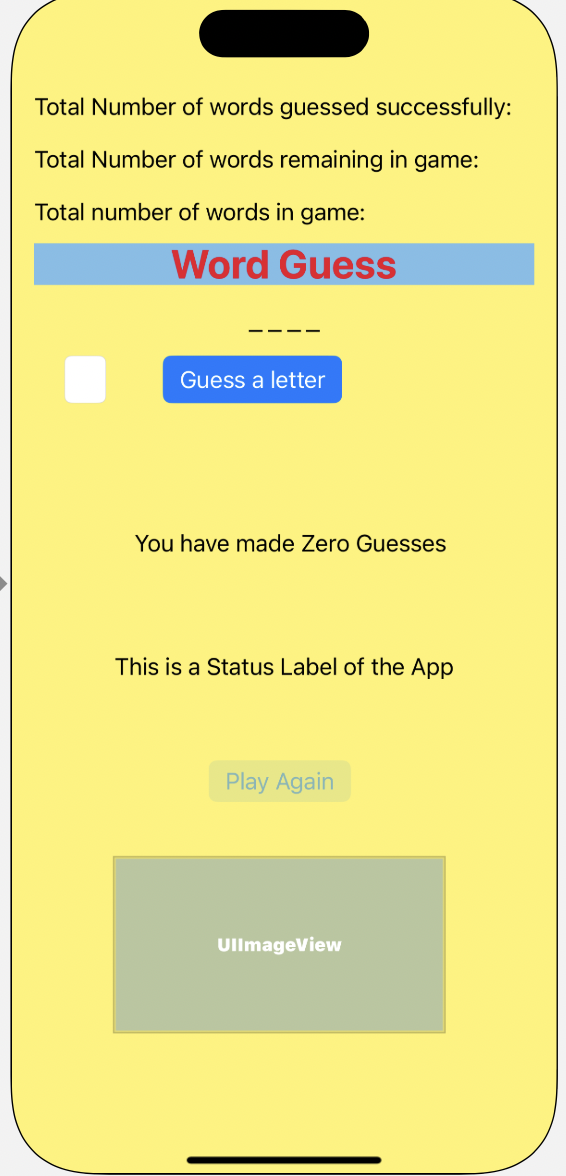
**Points 20**

**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 the product name as **Lastname\_WordGuess**, “**nwmsu**” for the organization identifier, “**storyboard**” as the interface and swift as the language.
5. Click on next and select an appropriate location to save your app and click on create. A project directory will be loaded.
6. When you are done with selecting the location of the project on your local storage, moving further you can see project settings screen. Set your **Project Format** to “Xcode 14.0-compatible” in your Project Document settings.
7. Now look into **Deployment Info** settings on the same project settings and set it to “iOS 15.5”. Check below image and highlighted portion with red and follow accordingly.



1. **Note:** Follow the exact instructions given and don’t modify any other settings as it causes application not to run in different machines. Use constraints wherever necessary.
2. From the project navigator, click on “Main.storyboard” file, and a blank mobile screen will be loaded where the required fields for an app need to be added.
3. Add three labels: “**Total number of words guessed successfully”**, “**Total number of words remaining in game”, “Total number of words in game”** and add constraints accordingly.
4. Create IBOutlet connections for all the labels and name them as “**wordsGuessedLabel”,** **“wordsRemainingLabel”,** and **“totalWordsLabel”.**
5. Add another label for the title of the app and name it as “**Word Guess**”. Use your own styling, and add constraints to this label (Top, Left, Right: 8,16,16) and add a height constraint of your own.
6. Add in another label and fill it with “\_ \_ \_ \_”, text in the label should be centered. Apply constraints as follows: Top, Left, Right: 8,16,16 and give it a height of: 20. Make an IBOutlet Connection and give it a name as “**userGuessLabel”.**
7. Now add a Text Field with the constraints (Height: 34, Width: 30). In the attribute Inspector, select the Spell Checking to NO, Correction: NO, Capitalization to: All Characters. Make a IBOutlet connection for this TextField and give it a name “**guessLetterField”.**
8. Now add a button next to the above Text Field and give the text as “**Guess a Letter**”**.** Apply Constraints (Width: 150, Height: 30). Add the above button and Text Field to a stack and add a Top: 8 constraint for this stack. When making an IBAction connection, give it a name “**guessLetterButtonPressed”.**
9. Now add a Label below the stack which is a Hint label, and give the constraints as (Top, Left, Right, Height: 8,16,16,40) and name it as “**hintLabel”** while making a IBOutlet connection.
10. Now add in another label below the above label for the game status. Give the constraints as (Top, Left, Right, Height : 8,16,16,80). Make sure the text can display on multiple lines without being cut off with “....” and set the default text as “**You have Made Zero Guesses**” and make sure the text is centered. Name it as “**guessCountLabel”** while making an IBOutlet connection.
11. Now add in another label below the above label for the game status. Give the constraints as (Top, Left, Right, Height: 8,16,16,80). Make sure text will show on multiple lines without being cut off with “…”. Set the default text as “**This is a status label of App**” and make sure the text is centered. Name it as “**statusLabel”** while making an IBOutlet connection.
12. Now add a play again button and apply constraints as (Top, Width, Height: 8,90,30). Make the button hidden by default. Name it as “**playAgainButtonPressed”** while making and IBAction connection.
13. Add an image view at last and name it as **displayImage** while making connection. When a word is guessed correctly respective image should be displayed.
14. Now we are done with the UI part. Make sure all the connections for the controls are made correctly.



1. Now **declare an array with at least 5 words and the related hints of your own**. Choose 5 images for above words to display in the image view when word is guessed correctly. Whenever we build the app, userGuessLabel in the UI should update with the same number of underscores (“\_”) as the letters in the word of the array.
2. Initially, **totalWordsLabel** should contain the number of words in the array: Words Guessed: 0, Words Remaining : 5, Words in Game : 5
3. Whenever the guessed word is correct these labels should be updated accordingly.
4. The **guessLetterField** Label should take only the last character even though if you enter multiple characters. Type in a letter in this Text Field using the keyboard of the simulator. If the keyboard is not showing up, then select I/O option for the simulator, then you can find Keyboard and then toggle the keyboard.
5. Guess a letter button should enabled only if there is any text in the Text Field or else disable it (**use editing changed event**)
6. Declare a constant, **maxNumOfWrongGuesses** as 10. You are allowed to make only 10 wrong guesses for each word.
7. When the application is launched, it should display as shown in **Figure 01.**
8. The **guessCountLabel** should display the count of the guesses you have used. You can refer to **Figure 02.**
9. Whenever you have guessed the word correct, related image for the word should be displayed and play again button should be enabled. On clicking, it should go to the next word. **Refer to Figure 03.**
10. If you used all the available guesses for a word, **guessCountLabel** should display “You have used all the available guesses, Please play again”. The “Play Again” button should be enabled, once you click on play again button you will get another chance to guess the same word. (Note: It should not start from the beginning of the words). **Refer to Figure 04 and Figure 05.**
11. **Refer to Figure 06,** which is the last word in the array. You can observe the **wordsRemainingLabel** at the top.
12. Whenever all the words in the array are completed, **statusLabel** should display “Congratulations, You are done, Please start over again” and the app should start over again resetting all the labels and play again button should be enabled. On clicking play again button, the app should start again from the beginning (**Refer to Figure 07).**
13. Refer to the below images for output and observe how the “Words Guessed”, “Words Remaining” labels updating accordingly for every correct guess.

Graphical user interface, text, application

Description automatically generated

**Figure 01**

Graphical user interface, text, application

Description automatically generated Graphical user interface, application, website

Description automatically generated

**Figure 02 Figure 03**

Graphical user interface, text, application, website

Description automatically generated Graphical user interface, text, application, website

Description automatically generated

**Figure 04 Figure 05**

Graphical user interface, text, application, website

Description automatically generated Graphical user interface, text, application

Description automatically generated

**Figure 06 Figure 07**

**Please submit your app as compressed file, your compressed files should contain Lastname\_WordGuess folder and Lastname\_WordGuess.xcodeproj file. Please check your submission by downloading the submitted file and recheck it in Xcode.**