# **ANALYSIS**

## **Breakdown of Program - IPO**

**Main Menu Class**

| **input** | **processing** | **output** |
| --- | --- | --- |
| [create avatar pressed] | new **JInternalFrame** |  |
| changeMenu method called | changeMenu method called  re-construct JPanel  - setText of sidebar JLabel to add age, gender, hair color  - construct start learning button | refresh with avatar personalization  - add age, gender, hair color, to sidebar  -add user icon with their features  display start learning button |
| [start learning pressed] | new JFrame (Learning Module)  destroy JFrame (Main Menu) | JFrame is removed  new JFrame pops up |

**JInternalFrame - Avatar Builder**

| **input** | **processing** | **output** |
| --- | --- | --- |
| [done avatar]  - JTextBox - Name  - JComboBox - Gender, Hair Color  - JSlider - Age | create instance of Avatar object (global)  - name = JTextBox value  - age = JSlider value  - gender = Gender JComboBox value  - hair color = Hair Color JComboBox value  call changeMenu method | JInternalFrame is removed  change contents on Main Menu Class |

**JFrame - Learning Module**

| **input** | **processing** | **output** |
| --- | --- | --- |
| (frame opens up) | (all the slides of the learning module will be inside an JPanel array. The JPanels of the JPanel array will be added to another JPanel (MainPanel))  first index of JPanel array is displayed |  |
| menu bar is pressed  - Home  - Learn  - Activity | find source | - if home: new main menu  - if activity:  if moduleDone is true: new activity  else: show JDialogBox |
| [next button is pressed]  if next is a checkpoint…  if next is the last slide… | - JPanel is re-constructed to next Panel of JPanel array  - MainPanel is revalidated and repainted  new JInternalFrame  - when JInternalFrame is destroyed  replace with end module button | next panel shows up  replace with end module button |
| [back button is pressed] | - JPanel is re-constructed to previous Panel of JPanel array  - MainPanel is revalidated and repainted | previous panel shows up |
| [end module button] | set moduleDone to true  new JFrame (Activity)  destroy JFrame |  |

**JInternalFrame - Checkpoint**

| **input** | **processing** | **output** |
| --- | --- | --- |
| [check work button]  answer will be inputted in JTextBox, JComboBox, or JRadioButton | (there will be a checkPointAnswersArray)  check to see if user response is equal to the designated answer | if not equal, display JDialogBox  if equal, destroy panel |

**JFrame - Activity**

| **input** | **processing** | **output** |
| --- | --- | --- |
| menu bar is pressed  - Home  - Learn  - Activity  - Template Class | find source | - if == current screen: do nothing  -if home: new main menu  - if learn: new learning module  - if template class: show code in new JInternalFrame (with ScrollBar) |
| [start button is pressed] | JPanel is reconstructed to show Cake template class (pre coded for the user)  JPanel is revalidated and repainted | Cake template class shows up |
| [next button is pressed]  if customer is ordering a cake  if user finished their cake  if customer is picking up the cake | JPanel is reconstructed to show customer order  JPanel is revalidated and repainted  check if instance and calling of get method is correct | show customer order + JTextBox + next  Show JDialogBox (“Added to Shelf”)  if is correct:  Show JDialogBox (“Happy Customer!”) |
|  | if 3 perfect cakes:  new JFrame (Quiz) |  |

**JFrame - Quiz**

| **input** | **processing** | **output** |
| --- | --- | --- |
| [submit button is pressed] | for all questions:  if answer is correct, points++  grade = points/totalPoints  if grade > .8 | Show JDialogBox of grade  Show JDialogBox of pass/fail  - if pass: new Certificate |
| [retry button is pressed] | destroy JFrame  new JFrame |  |

**JFrame - Certificate**

| **input** | **processing** | **output** |
| --- | --- | --- |
|  | Add JLabel name to Panel | Display certificate |

## **Data Sources, Types, Structures**

| **SOURCES**:   * Textbook Chapter 3 [Chapter 3 Introduction to Classes and Objects](https://drive.google.com/file/d/1nDpLJzpp6qQyfkI2kc5zFz7vPA5IpVST/view) * My code from Unit 3 HW | **TYPES:**   * OBJECTS   + Avatar: used to build user’s avatar at start, introduce concept   + Cake: used for Activity * Strings: JTextBox inputs, Paragraphs to display * Integers: Indexes in arrays, JSlider input * Doubles: Final Grade * Booleans: isCompleted (for each class, used to determine was portions of the CAI are locked/unlocked) | **STRUCTURES:**   * Regular Arrays   + LearningPanelArray (JPanel[]): Makes it easy to move back and forth between panels (next index, previous index) * 2D Arrays   + Question/Answer Array ([# of questions][2], String[][]): Will help to keep questions and answers together and organized, compared to 2 separate arrays |
| --- | --- | --- |

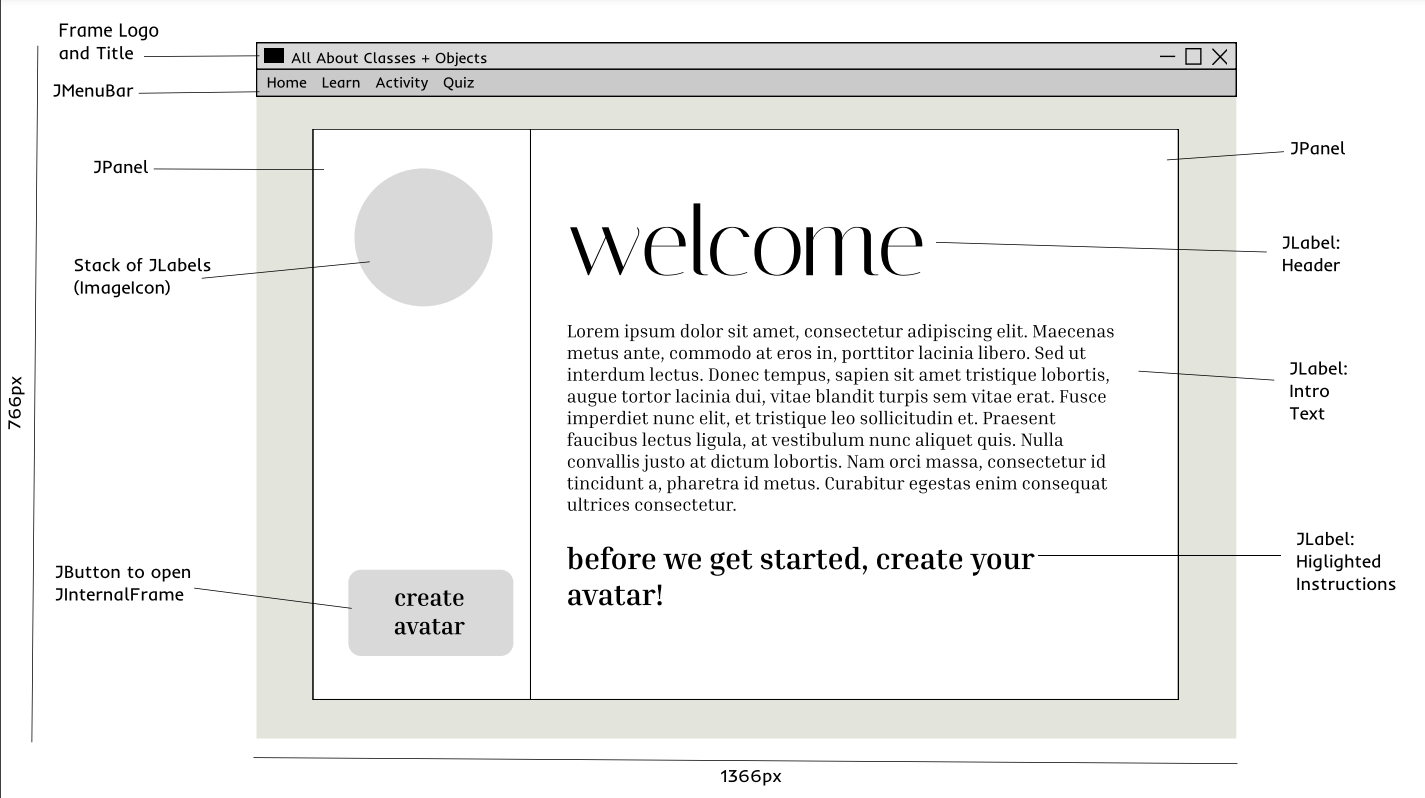
## **Issues and Concerns:**

* How to create a JInternalFrame
* Can you stack PNG files on top of each other (to create user avatar icon in main menu)
* How to use JProgressBar; how to update it in real time
* How to add a read-aloud option for text (is this too complicated?)

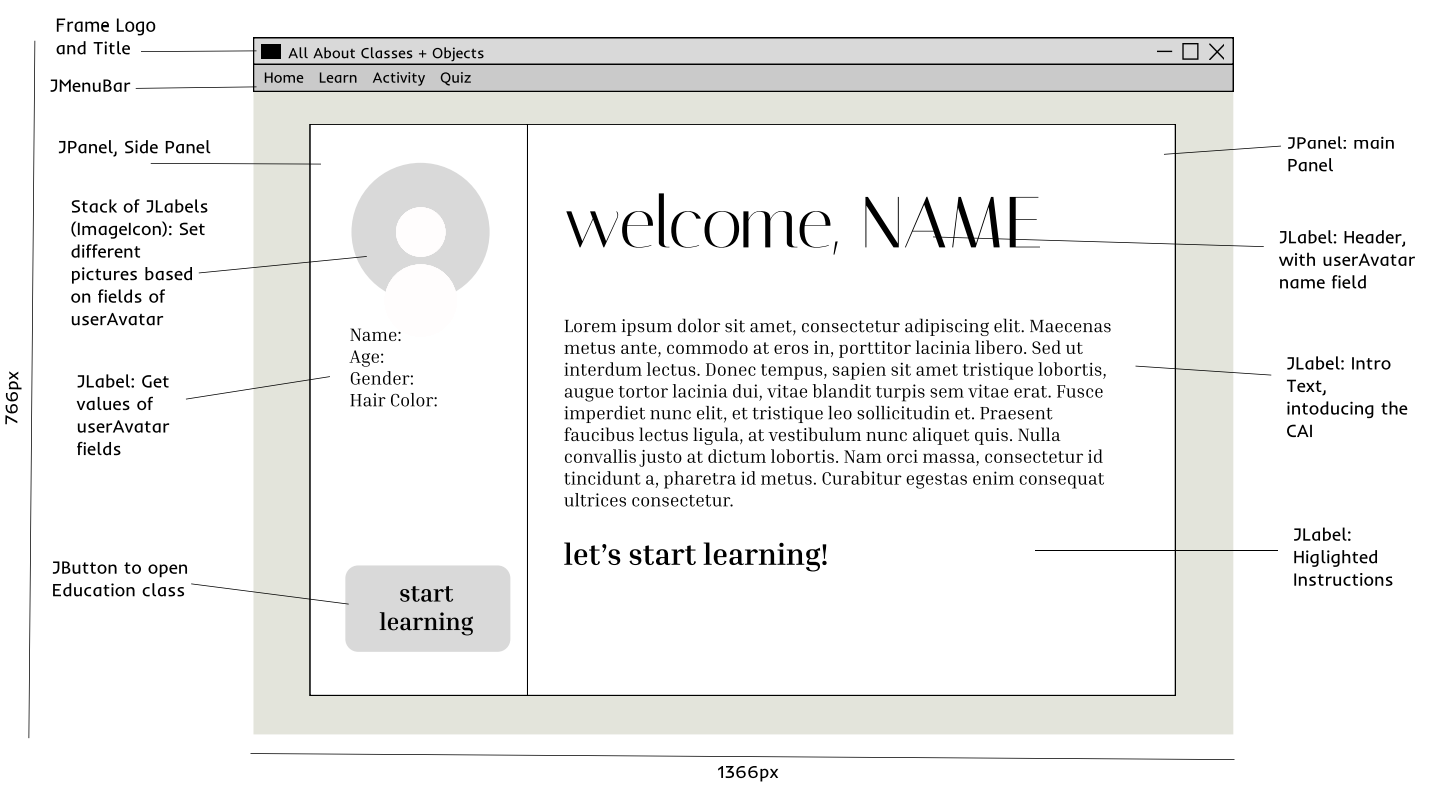
# **DESIGN**

## **Prototypes**

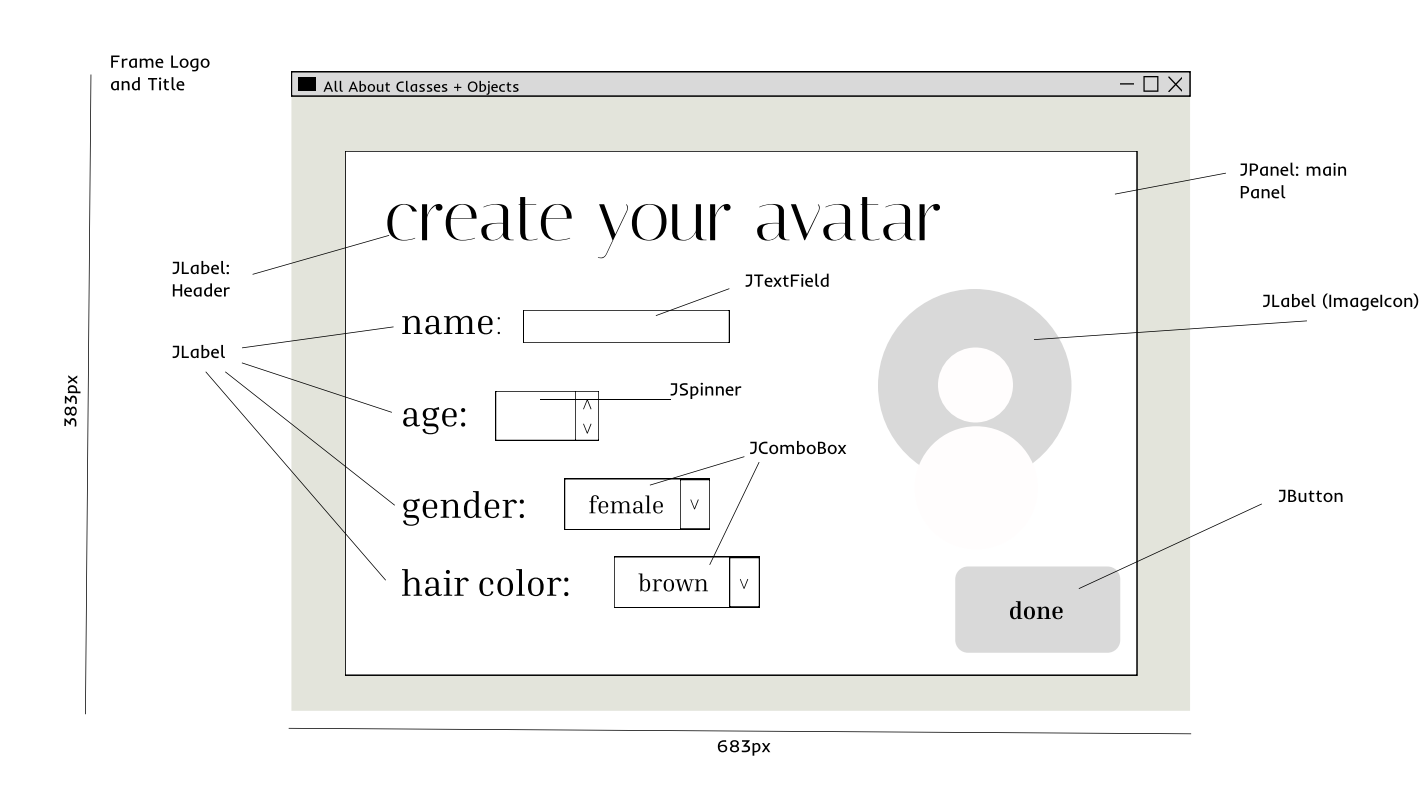
**Main Menu Class**

****

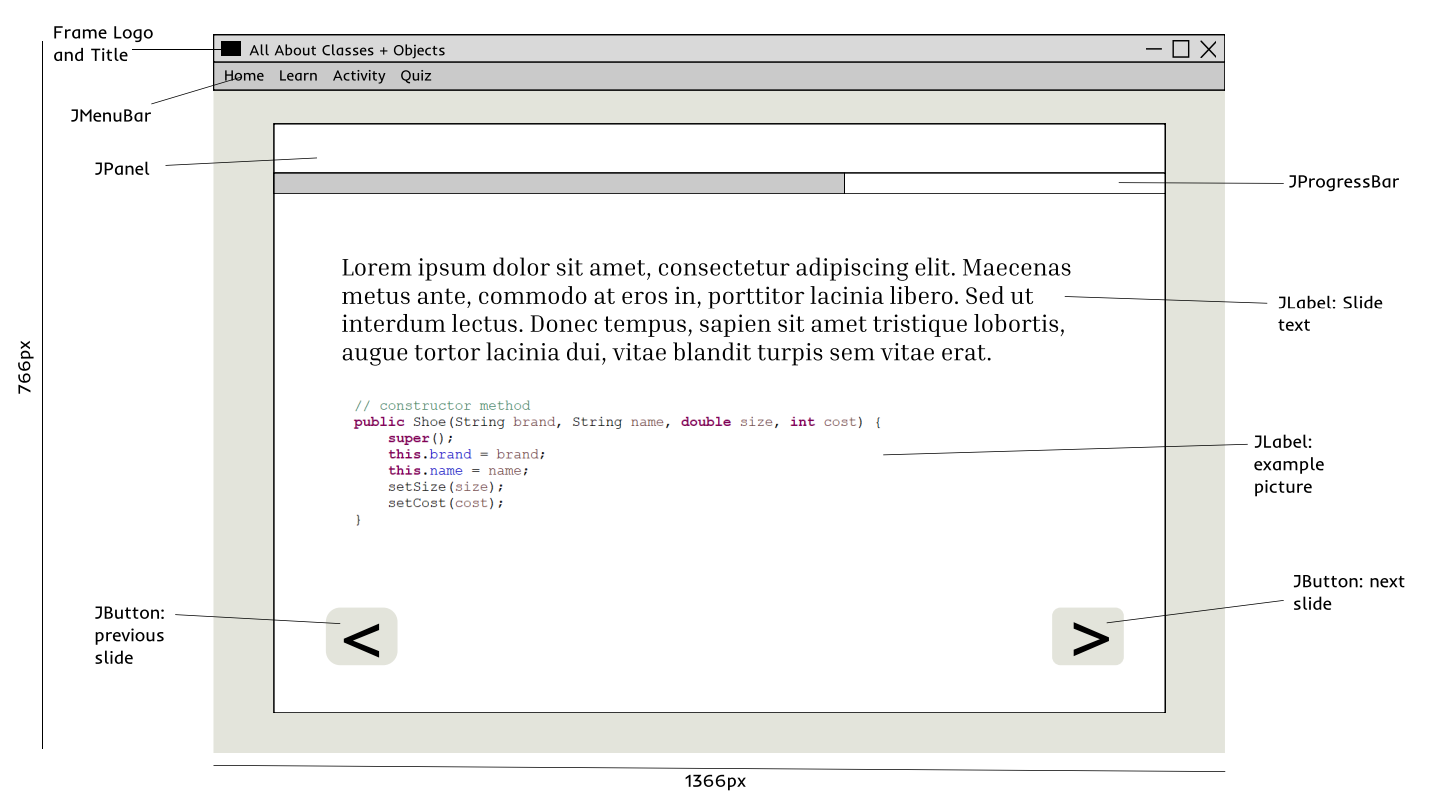
**(After Avatar isMade)**

****

**JInternalFrame: AvatarBuilder**

****

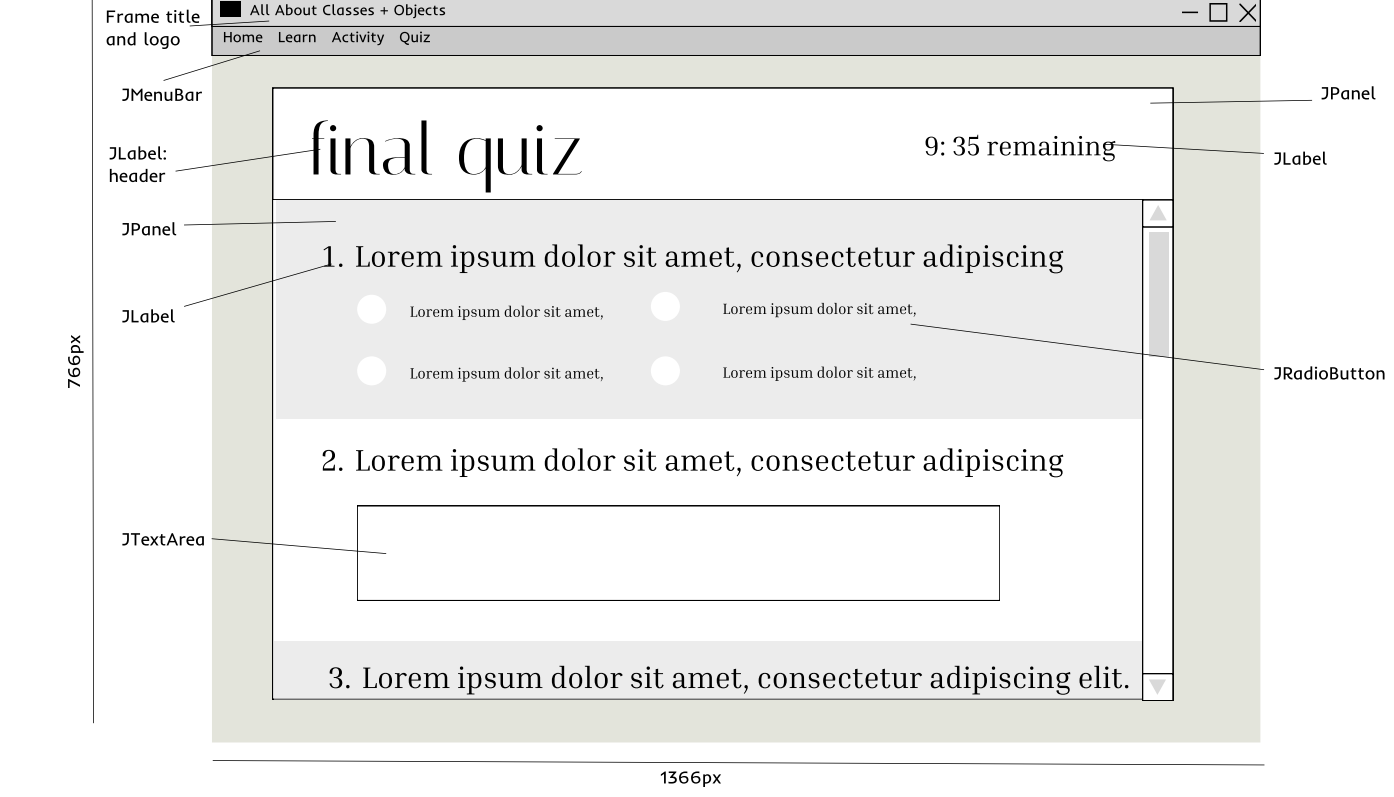
**Education Class**

****

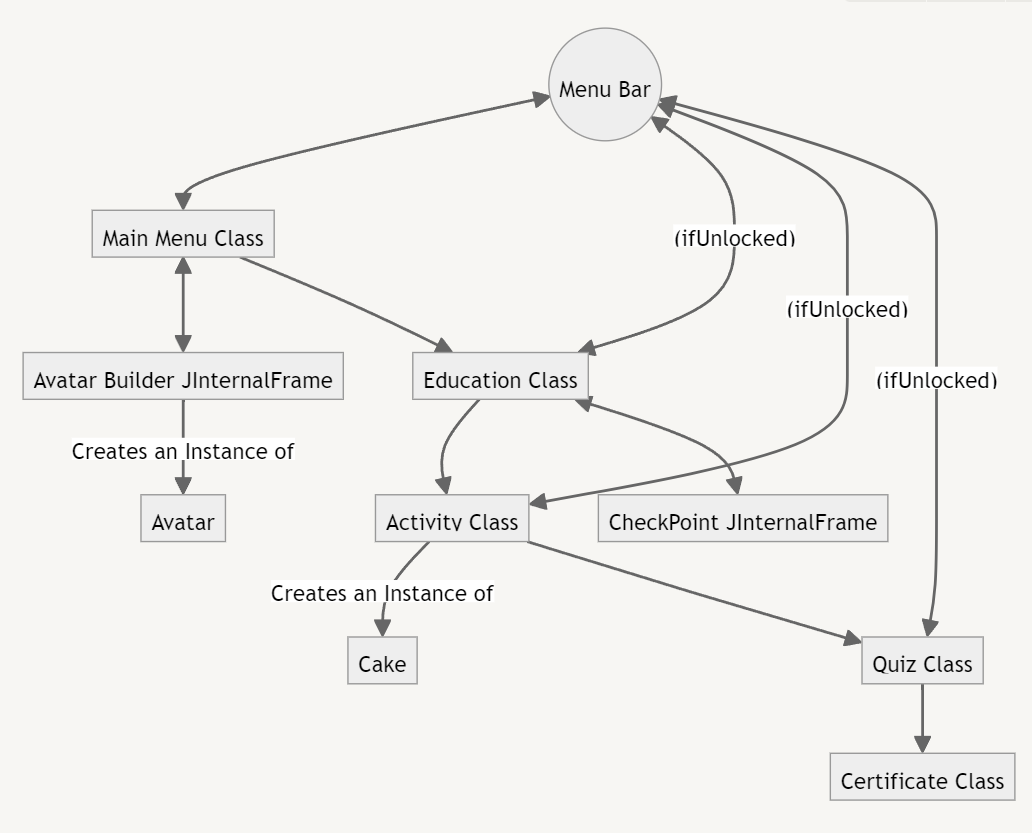
**Activity Class**

****

**Quiz Class**

****

## **Structure Chart (UML Class Diagram)**



## **Algorithms**

| **Common Methods (perhaps make them into a separate class?)** | **MenuBarPressedMethod**   1. Action Listener for MenuBar: call **MenuBarPressedMethod** 2. If button source is *Main Menu* button && current frame is not Main Menu    1. new Main Menu Class (2ND VERSION: DO NOT PROMPT AVATAR BUILDER)    2. destroy current frame 3. If button source is *Learn* button && current frame is not Education    1. if userAvatar is not equal to null       1. new Education Class       2. destroy current frame    2. else       1. Show JDialogBox that Class is locked, create Avatar to unlock 4. If button source is *Activity* button && current frame is not Activity    1. if learnIsCompleted is equal to true       1. new Activity Class       2. destroy current frame    2. else       1. Show JDialogBox that Class is locked, complete Education to unlock 5. If button source is *Quiz* button && current frame is not Quiz    1. if activityIsCompleted is equal to true       1. new Quiz Class       2. destroy current frame    2. else       1. Show JDialogBox that Class is locked, complete Activity to unlock 6. Else: do nothing |
| --- | --- |
| **Main Menu Class** | **CreateAvatarMethod**   1. Action Listener for *Create Avatar* Button: call **CreateAvatarMethod** 2. new JInternalFrame    1. Set up contents of Frame (JPanels, JLabels, JComboBoxes, JTextFields)    2. Add *Done* button       1. Add ActionListener:          1. Set value of Avatar based on user selections          2. Change JFrame Screen             1. Set Name, Age, Hair Color, Gender to Avatar field, add PNG JLabels based on Avatar fields             2. Remove Create Avatar Button, Create Avatar Instruction             3. Add Start Learning Button, Start Learning Instruction             4. Destroy JInternalFrame |
| **Education Class** | **NextSlideMethod**   1. Action Listener for *Next Slide* Button: call **NextSlideMethod** 2. Remove old Slide    1. Remove LearningPanelArray[Current Slide] 3. Add new Slide    1. Current Slide += 1    2. Add LearningPanelArray[Current Slide]    3. Revalidate MainPanel    4. Repaint MainPanel   **PreviousSlideMethod**   1. Action Listener for *Previous Slide* Button: call **PreviousSlideMethod** 2. Remove old Slide    1. Remove LearningPanelArray[Current Slide] 3. Add new Slide    1. Current Slide -= 1    2. Add LearningPanelArray[Current Slide]    3. Revalidate MainPanel    4. Repaint MainPanel |
| **Activity Class** | **CheckCakeMethod**   1. Action Listener for *Next* Button: call **CreateCakeMethod** 2. If TextField.getValue() == QuestionAnswerArray[question#] [2]    1. Perfect Cake++    2. If Perfect Cake == 3       1. New Quiz Class    3. Else       1. Call NextCustomer Method |
| **Quiz Class** | **GradeAnswersMethod**   1. Action Listener for **Submit** Button: call **GradeAnswersMethod** 2. For # of questions    1. If AnswerArray[question#].getValue() == QuestionAnswerArray[question#] [2]       1. Marks++ 3. Grade = Marks/# of question 4. If Grade > 0.8    1. Pass = true 5. Show JDialogBox to announce Grade, Pass/Fail result    1. If Pass == true       1. new Certificate Class |

# **SCHEDULE**

| **DAY** | **TASK** |
| --- | --- |
| Winter Break (Non-IT tasks) | - Write the text for the Education Class/Slideshow (find out how many slides are needed) + figure out sample code  - Write the checkpoint questions (2-3 MAX)  - Write the quiz questions and answers  - Write the script for the Cake Artist Activity  - Find all the pictures needed for the Avatar Icon  - Find background music  - Make blank certificate for end  - Learn how to create a JInternalFrame, use a JProgressBar |
| January 8th | - Create the Objects (Avatar, Cake)  - Create a JPanel[] LearningPanelArray (slides)  - Create the skeleton for the Education Class (GOAL: get the panels to revalidate and refresh like a slideshow, when the button is clicked, go back and forth) |
| January 9th | - Add text to the JPanels of the slideshow, add pictures  - Create checkpoint JInternalFrame pop-ups + exit button  - Create quiz class, with questions unformatted  -Create bare main menu Class, w/o avatar builder |
| January 10th | - Add grader code, show JDialogBoxes  - Create Certificate class, with certificate displayed  - Create Avatar Builder JInternalFrame  - Create an instance of the Avatar class based on the user’s input  - Format user icon on main menu |
| January 13th | - Connect the Menu, Avatar, Education, Quiz, and Certificate class together  - add finalizing formatting details (no need to add menu bar)  - Start Cake Artist class (make skeleton) |
| January 14th | - Finish Cake Artist skeleton  - Connect it to the rest of the App (have working app) |
| January 15th | - Add sound effects  - Add MenuBar connections  - Code pop-ups if Class is locked |
| January 16th | - Start Formatting details |
| January 17th | - Continue formatting details |
| January 18th | - Finish formatting  - Project Testing  - ½ of Test matrix |
| January 19th | - finish Test Matrix  - Project Header |

# 