HDC – HGP - Assignment 01

UI Design Document

## **Student Name:** Chukwuemeka Wisdom Arinze

## **Student Number:** 2970177

1. Using an application such as draw.io or similar, create a wireframe of the layout of your UI, including explanation of what containers you used and why, as well as the naming convention used for each of your components.

|  |
| --- |
| **Wireframe and explanation:** |
| The main layout of the application is BorderPane, this is because with this layout I can position any sub layout left, right, centre and bottom.  The first sub layout is a grid. Grid helps to arrange UI elements in rows and columns. Looking at the brief of what we are meant to create, grid layout would work fine.  The second sub layout I choose was a VBox, each button in the brief is layered on top of each other, so that is why I went for a VBox. VBox aligns all elements vertically.  The third sub layout I choose was a grid pane. In the brief, the labels are aligned in rows and columns, so I decided a Grid would be better for this layout.I could have aligned them with a VBox but I am more comfortable with a grid that another reason I went for a grid  The naming convention for the UI elements I used is carmel case.  A screenshot of a computer  Description automatically generated  A screenshot of a computer  Description automatically generated |

1. Include screenshots of your Main User Interface. Why does your UI look the way it does? What design decisions did you make for this interface? Justify your decisions with a foundation of UI Design Principles.

|  |
| --- |
| **Screenshots and explanation:** |
| The way I designed my application was by following the brief, each element represented in the application was named saw that the user will be able to have an idea of the purpose of each UI element. The label at the very top has a margin style attached to them because they were touching the border of the application which does not look right.  The list view is placed on the left because it works better this way. As the user clicks on any name on the left, the information about the name clicked will be appear on the right.  There are also labels underneath the image view because they will give information about the image above. Each information has a label on the left which tells the user what the labels at the right contain.  The two buttons are in the main view of the application are in the same location because it would make no sense for buttons to be in different locations.  A screenshot of a computer  Description automatically generated  The Add student button and the remove student button are positioned in the same place. Because those two buttons interact with the list view and controls that interact with the csv file should be placed together.  Inside the Add student section, we have labels on the left that tell the user what information that is required in each text field. The combo box also was used to display a list of options because we want the user to select just one option from the list.  I also placed a button at the bottom of the image view so that whatever image the user selects (.jpg or .png) will appear right in the image view. The decision to place the button there was because we want those two UI elements to be in one place.  The font weight was also changed because the default size and weight were too small for users to see them |

1. Include screenshots of the ‘Add Student’ dialog. What components did you choose to include in the dialog? What design decisions did you make for this dialog? Justify your decisions with a foundation of UI Design Principles.

|  |
| --- |
| **Screenshots and explanation:** |
| From the brief, we need five information to be able to add a student. First name, surname, student number programme and a profile photo. The first three information will be provided through a text box, the last information will be provided through a combo box and the last one will require the user to select a profile picture from the system, which will then appear in the image view.  Text fields were appropriate to collect text from the user, for the programme we need just one option selected so I employed the use of a combo box. The last was a file dialog and image view. The file dialog helps the user to select an image and the image view displays selected image.  If the user makes a mistake, an alert box will tell them the problem when they try to add student.  Same as the main view buttons are placed together in the same location so are the text fields.  A screenshot of a computer  Description automatically generated |

1. Briefly explain the functionality of each of the methods in your application.

|  |
| --- |
| **Screenshots and explanation:** |
| constructor():  Initializes all the UI elements we will need in the application.  start():  All the resources that need to be executed so that our application can run are placed here.  init():  That’s where we place our which type of actions will invoke our methods.  main():  Launches the application.  custom methods:  Methods carry our functionality and are invoked through the actions in the init method. |

1. EXTRA FEATURE: Explain what your extra feature is, why you chose it, how it works etc. Include any relevant screenshots of your extra feature in action. Remember you must complete this section in detail to receive marks for your extra feature.

|  |
| --- |
| **Screenshots and explanation:** |
| This is executed any time the program starts. My custom functionality changes the background color to when the program is executed. The method generates a random number, and this random number will be the index for string of colors in an array. The random number is generated from 0 – 10, then the number will be used as an index for a string array that will return a color that will be placed in an inline style sheet as the background color.  A screen shot of a computer  Description automatically generated |

1. Describe the interactive elements in your UI. How did you make sure they are intuitive and easy to use?

|  |
| --- |
| **Explanation:** |
| The text fields just require the user to click on them and begin typing.  The combo box requires the user to select on option out of three and the selected option is what will be passed as the value for programme.  To get a photo from the system, the user clicks a button and the button opens the window dialog and the user can navigate to anywhere in the system to find a photo |

1. What do you envision to be the user base for this application? Justify your response. Describe the typical user of this application.

|  |
| --- |
| **Explanation:** |
| This application can be used to know which student in a school take a particular module or are which student are in a particular year. Class representatives could use it to share information to a group of students in this same class/year.  A teacher could use it to take attendance. |

1. How would you redesign this application? Draw a new wireframe detailing what you believe the ideal design to be for the StudentListFX application.

|  |
| --- |
| **Wireframe and explanation:** |
|  |

1. If you were to continue working on improving this application, what enhancements or updates would you add and why? Briefly explain what you would incorporate in future work, justify your reasoning.

|  |
| --- |
| **Explanation:** |
| I would incorporate sending the csv file to the cloud for storage. It is possible that keeping just one file with all that information in one place would not be a good idea because files can be corrupted, destroyed or wrong information could be placed in there. If it was on the cloud, it would be secure, we would have copies whenever we need it and we would know who was the last person to update, download and use it. |

1. If any part of your submission isn’t working, please detail this below including relevant screenshots.

|  |
| --- |
| **Screenshots and explanation:** |
| Unfortunately, I was not able to get the remove student button working. Colud figure out how to get it done |