



Assignment Cover Letter (Individual Work)

Student Information:	Surname	Given Names	Student ID Number
1. 2. 3. 4. 5.	Arvin	Lee	2301956134
Course Code	: COMP6510	Course Name	: Programming Languages
Class Major	: L2BC : Computer Science	Name of Lecturer(s) : 1. Jude Joseph Lamug Martinez	
Title of Assignment (if any)	Final Project		
Type of Assignment	Final Project		
Submission Pattern			
Due Date	: 20 th June 2020	Submission Date	: 20 th June 2020
<p>The assignment should meet the below requirements.</p> <ul style="list-style-type: none"> 1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions. 2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission. 3. The above information is complete and legible. 4. Compiled pages are firmly stapled. 5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission. 			

Plagiarism/Cheating

Binus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student:

- 1.
- 2.
- 3.
- 4.
- 5.
- etc.


 Arvin Lee
 20 June 2020

**) Delete the inappropriate option*

1. Introduction

Jialat Zipper is a zipper application similar but a simpler version of WinRaR and its type. The application is built based on Java Programming Language by using Oracle JavaFX 8 platform. The main focus of the application is its interactivity with user, the structure of the program, and the flexibility of its component.

2. Terminology

The name Jialat comes from a chinese word 吃力 (Pronounced in chinese as *Chi Li* and hokkien dialect as *Jia Lat*) means “taking a lot of energy”. It was chosen as the name of the application because of two reason.

1. The application is to zip and unzip, which takes a lot of work to do.
2. The application requires quite an extra effort to make especially when made in an unsupportive period of time.

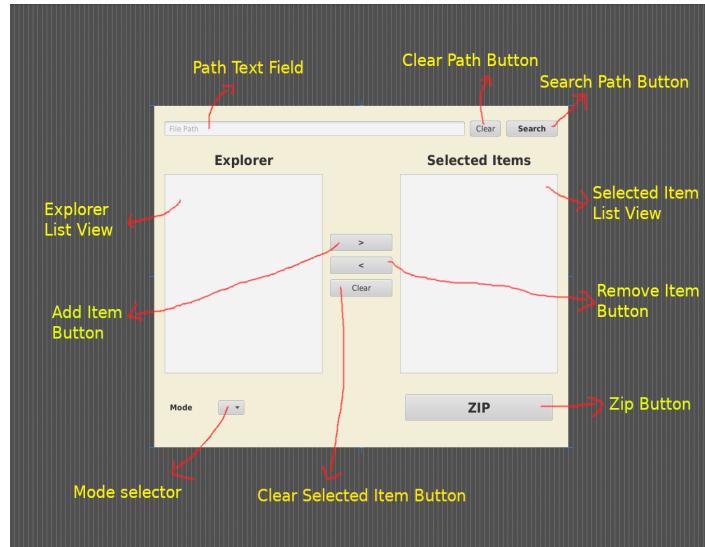
Specification

Specification		Description
General		
OS	:	Manjaro Lyssia 20.0.3
Language	:	Java Programming Language
Version	:	8
Platform	:	JavaFX Oracle
IDE	:	IntelliJ Community Edition 2:2020.1.1-1
UML	:	AVAILABLE
Documentation	:	AVAILABLE
Comments	:	AVAILABLE
Libraries	:	<ul style="list-style-type: none">• Zip files library zip4j-2.6.0
Application		
Aims	:	<ul style="list-style-type: none">• Simplicity• User friendly• Class structure• Flexibility
Features	:	<ul style="list-style-type: none">• File searcher area in application• Zip multiple files with / without password• Unzip multiple files with / without password• Select a destination path for zip/unzip result• Name the zip file / unzip folder• Show zip/unzip progress
Page/Menu	:	3 (Main Page, Zip Page, Unzip Page)
Style		
GUI	:	AVAILABLE
Font-Family	:	System
Scene Base Color	:	#ffffae3
Window min size	:	800*600 (w * h)

3. App Overview

a. Main Page

The main page is where user chooses/selects the item to zip/unzip. The mode (zip/unzip) can be changed through the mode selector. The explorer list view is for traveling through folders to find files, user can search the path directly from the path test field. Once a file is found, user can either double click it or press the add item button to add it to selected item list view. The user can not add the ".." (back option) or the files which already exist in the selected item list view. The user can proceed to the next page by pressing zip button at the bottom right of the window.

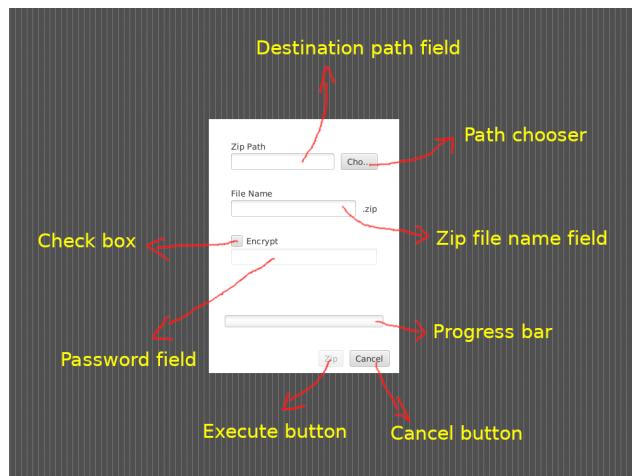


Picture 3.1 Main Page

b. Zip Page

A window will appear after user pressed the zip button in the main page, depending on the mode, if user selects zip mode, the content of the window will be asking for information about the zip such as the path where the result will be located, the name of zipped zip file, and whether to encrypt the zip with password. User needs to fill the path text field, and name text field in order to be able to execute zip, it is optional to choose the encryption, however once the check box is checked, user need to input a password of minimal 8 characters long.

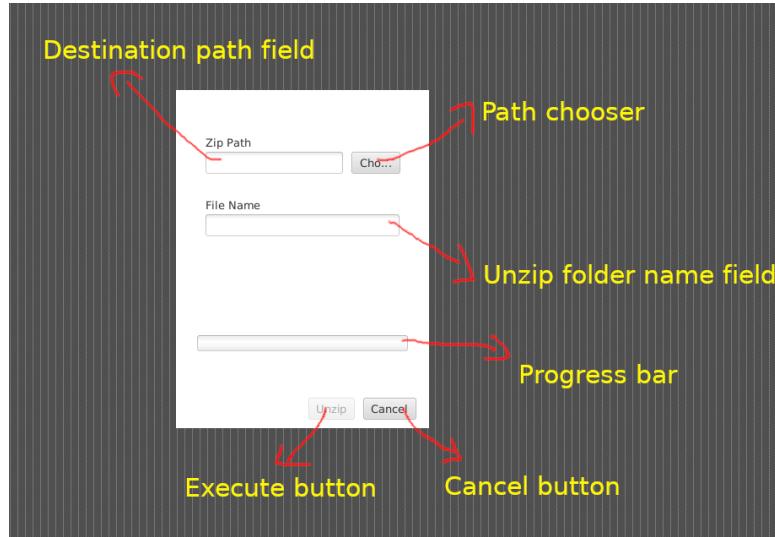
After the execute button is being pressed, the progress bar will start to fill up depends on the progress up and the window will be frozen until the task is finished. The cancel button will turn into done button and click-able once the task is completed.



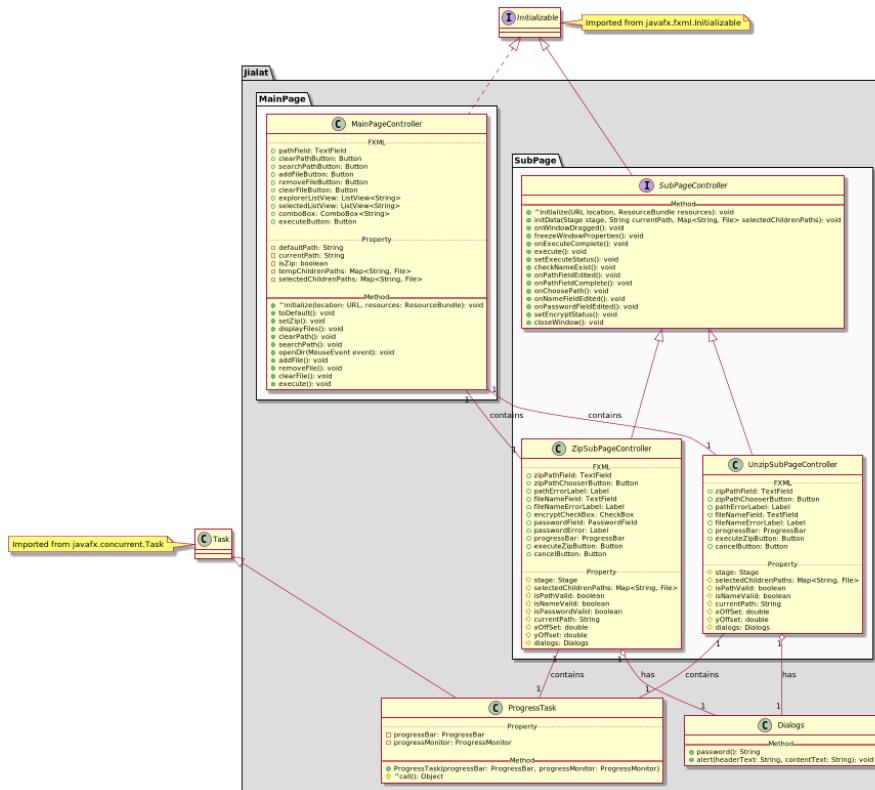
Picture 3.2 Zip Page

c. Unzip Page

The unzip page will appear if the user is in unzip mode. Similar to how the zip page looks like, however



there is no check box nor password field in the unzip page. The user needs to fill in the path where the unzipped zip file will be located and its folder name. Once the execute button is pressed, a dialog



window will show up and ask for password if the zip file is encrypted. If the user inputs the wrong password, the folder of the extraction will not contain anything. The rest works the way it worked in the zip page.

Picture 3.3 Unzip Page

4. Class Design

picture 4.1 UML Diagram

Jialat application consist of 3 Main Classes, the MainPageController class is the class for the MainPage view, the ZipSubPageController class and UnzipSubPageController which implement the SubPageController Interface and basically responsible for the zip page view and unzip page view. The Interface is used instead of Abstract Class

or Inheritance because of its flexibility. For example when using abstract class, the `setExecuteStatus` method is supposed to be implemented the children class, however since it is used in almost all of the methods including the one implemented in the abstract class itself. As a result, it needs to be implemented in the abstract class and not in the children class. However, such option is not using the abstract class to its fullest. Therefore, interface is used.

There 2 extra classes namely Dialog Class which responsible for dialog pop up such as password dialog (ask for password) and alert dialog (show alert), and ProgressTask Class which is responsible for updating the progress of zip/unzip task. These two classes are instantiated in both ZipSubPageController class and UnzipSubPageController Class to carry out some task which requires them.

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
JalaZipper SubPageController.java
JalaZipper src Jalaat SubPage SubPageController
Project src .idea Documentation lib out rsc src Jalaat
MainPage MainPage.fxml MainPageController
SubPage SubPageController unzipmainFXML.fxml UnzipSubPageController zipmainFXML.fxml ZipSubPageController
Dialogs ProgressTask
Main UML JalaZipper.Impl README.md External Libraries Scratches and Consoles
Main.java MainPage.java SubPageController.java UnzipSubPageController.java ZipSubPageController.java ProgressTask.java Dialogs.java
Main.java
1 package Jalaat.SubPage;
2
3 import javafx.fxml.Initializable;
4 import javafx.stage.Stage;
5
6 import java.io.File;
7 import java.util.Map;
8
9
10 /**
11  * An interface which extends Initializable and implemented by its kind such as the ZipSubPage and UnzipSubPage which are used
12  * after the main window to get more information about the execution, e.g. the path where the result of zip/unzip will be located,
13  * etc.
14 */
15 public interface SubPageController extends Initializable {
16     void initData(Stage stage, String currentPath, Map<String, File> selectedChildrenPaths); //The method to pass in data from
17     void outside of the class
18     void onWindowDragged(); //Method triggered whenever the window is dragged
19     void freezeWindowProperties(); //Method called to freeze the content of the page
20     void onExecuteCompleted(); //Method triggered upon the completion of a task
21     void setExecuteStatus(); //Method used to set the execute status
22     void checkNameExist(); //Method used to check if a file name is exist
23     void onPathFieldEdited(); //Method triggered whenever the path field is edited
24     void onPasswordFieldComplete(); //Method triggered upon finish editing the path field
25     void onChoosePath(); //Method used to choose path
26     void onNameFieldEdited(); //Method triggered whenever the name field is edited
27     void setEncryptStatus(); //Method used to set the encryption status
28     void setPasswordFieldEdited(); //Method triggered whenever the password field is edited
29     void closeWindow(); //Method called to close window
30 }
```

5. Codes and Demonstration

The demonstration video can be watch in the following link:

https://drive.google.com/file/d/19p5Ys2D1xJWS7Y09jKBxzDY0g_TH1F2s/view?usp=sharing

The codes will not be explained in this report but will be fully described in the documentation, it can be accessed in the github link, in the Documentation folder or by pressing this link:

<https://github.com/IlzJokerzII/JavaFinalProject/tree/Test2/Documentation>

Picture 5.1 SubPageController Class

6. Reference

- i. Zip4J :
[\(github link\)](https://github.com/srikanth-lingala/zip4j) <https://jar-download.com/artifacts/net.lingala.zip4j/zip4j>