Problem Statement

In CyberCop 2.0, you will develop a GUI -based application with some changes in functionalities. The key changes are:

1. View all cases
2. Search, Add, Modify, or Delete a case
3. Use data in TSV and CSV formats with additional columns

Fig.1. and 2 show the opening screen and data-screen with names of important GUI controls.

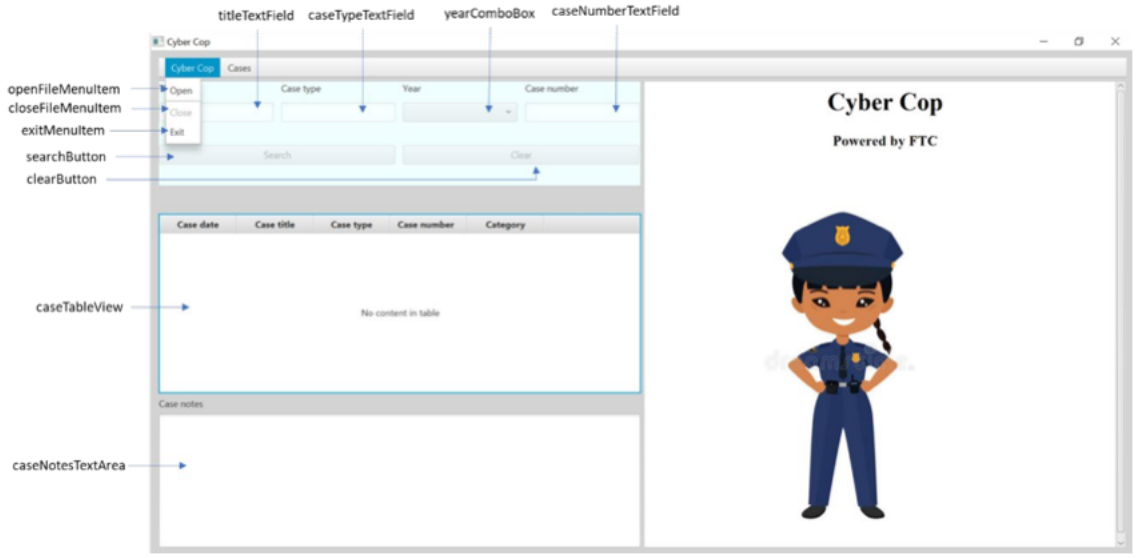

Figure 1: Cyber Cop Opening Screen

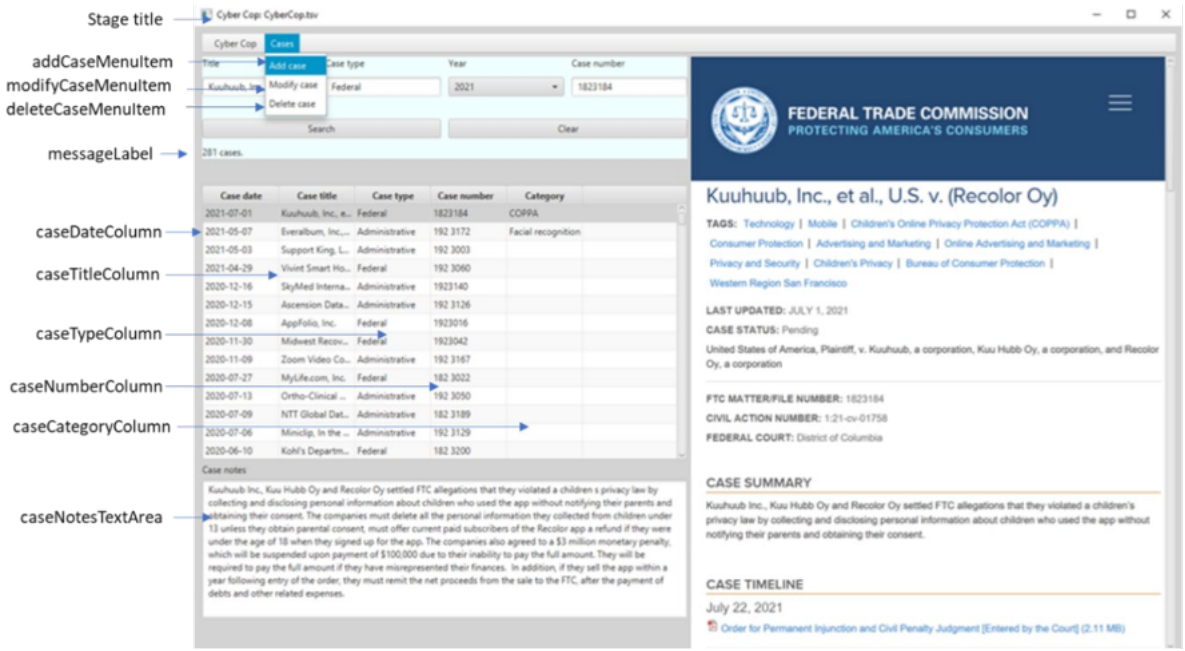

Figure 2: Cyber Cop with CyberCop.TSV opened (Note: The data and web-view will look a little different for the data-set given to you. The reason is explained in the Data section of this document.)

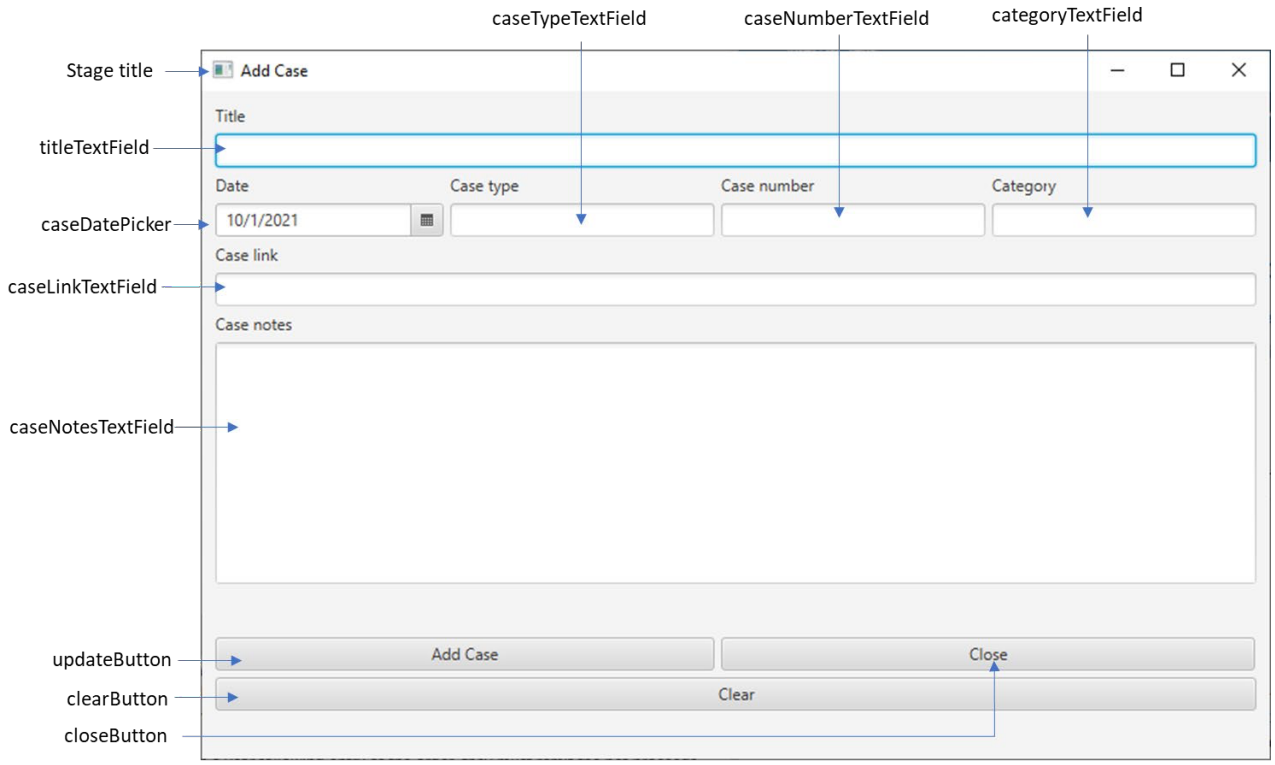

Figure 3: Case View GUI controls to Add/Modify/Delete cases

For more details on the functionality, refer to the demo video-clips provided on Canvas.

Solution Design

A class diagram for Cyber Cop 2.0 is provided in Fig.4. A brief description of each class is given below.

Table 1: Class descriptions

Class name	Description
CyberCop	Extends Application. This class is partially coded. Please refer to code comments in CyberCop.java for more details.
CCView	This class contains all GUI controls for the opening scene as shown in Fig. 1 and 2. This class is fully coded.
CCModel	This class is responsible for all data-related operations. • readCases(): reads TSV or CSV file and initializes caseList and caseMap • buildYearMapAndList(): loads yearMap and yearList. • searchCases(title, caseType, year, caseNumber): returns a list of Cases that meet the search criteria
Case	• A Java bean with private properties and public getters, setters. • Overrides toString() to return caseNumber. • Implements Comparable to sort on caseDate in the order starting with most recent
CaseReaderFactory	A class whose createCaseReader() method returns an appropriate caseReader, i.e., TSVReader or CSVReader depending on the extension of the filename passed to it.
CaseReader	An abstract class with a non-default constructor to initialize its filename and an abstract readCases() method

CSVCaseReader	Implements readCases() and returns caseList, i.e. an ArrayList of Cases. See Data in next section for more details.
TSVCaseReader	Implements readCases method and returns caseList, i.e. an ArrayList of Cases. See Data in next section for more details.
CaseView	An abstract class that builds the GUI for Add/Delete/Modify views. It has an abstract method buildView. This class is fully coded.
AddCaseView	Extends CaseView. Its buildView() method returns a Stage with empty GUI controls for a case. The caseDatePicker control is initialized to show current date by using the following code snippet in buildView() method: `caseDatePicker.setValue(LocalDate.now());`
ModifyCaseView	Extends CaseView. Its buildView() method returns a Stage with 'currentCase' displayed
DeleteCaseView	Extends CaseView. Its buildView() method returns a Stage with 'currentCase' displayed

Event Handlers

OpenFileMenuItemHandler	Opens dialog box for user to select a case file. Invokes ccModel's readCases() and buildMapAndList() methods to populate caseList and yearList. Selects first record in caseTableView and makes currentCase point to it. Updates stage title, and messageLabel. Sets isFileOpen to true.
CloseFileMenuItemHandler	Clears all GUI elements and sets isFileOpen to false.
ExitMenuItemHandler	Exits the application
SearchButtonHandler	Displays cases that contain data entered in titleTextField, caseTypeTextField, yearComboBox, and caseNumberTextField. Updates messageLabel accordingly.
ClearButtonHandler	Clears the data entered in titleTextField, caseTypeTextField, yearComboBox, and caseNumberTextField
CaseMenuItemHandler	A common handler for three menu items - Add case, Modify case, Delete case. Depending on which menu item was chosen, it creates an instance of AddView, ModifyView, or DeleteView respectively. Also creates an instance of AddButtonHandler, ModifyButtonHandler, or DeleteButtonHandler, and binds it to caseView's updateButton. The caseView's clearButton clears all data from the view and closeButton closes its stage.

AddButtonHandler	At this point, we will not check for duplicate case entry.
	To extract data from caseDatePicker control, use the following code snippet:
	
	`caseView.caseDatePicker.getValue().format(`
	`DateTimeFormatter.ofPattern("yyyy-MM-dd"))`
	
	The new case is added to caseList and messageLabel is updated

ModifyButtonHandler	Takes the data from all GUI controls and updates currentCase's properties so that they are updated in the main screen's view.
DeleteButtonHandler	Removes the currentCase from caseMap and caseList. Updates messageLabel.

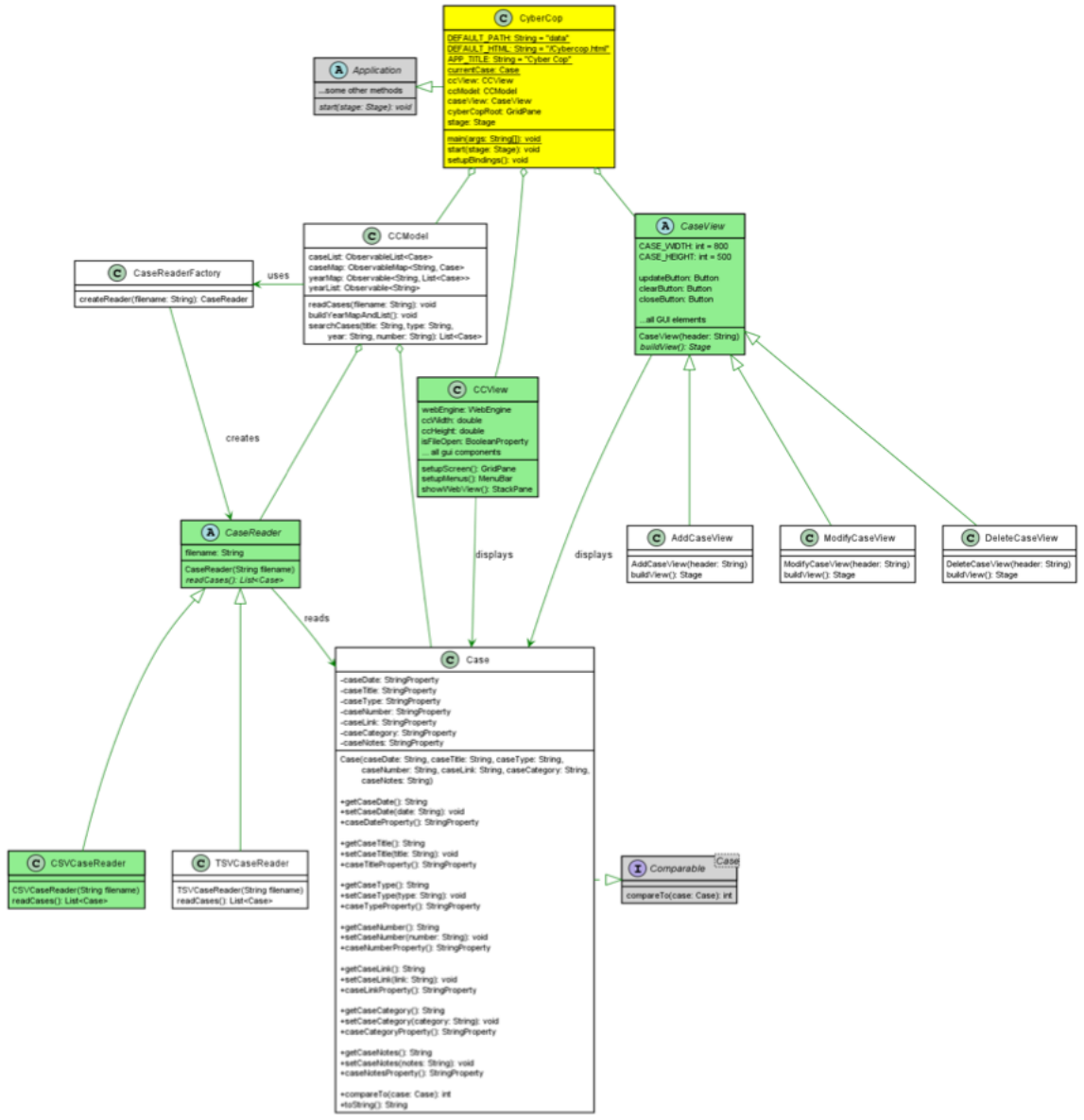

Figure 4: Cyber Cop 2.0 Class Diagram

Legends:

Grey: Library class
Green: Fully coded
Yellow: Partially coded
No color: To be coded

The following diagram shows event handlers required in Cyber Cop. You may code them as member classes, inner classes, or lambda functions, as per your preference.

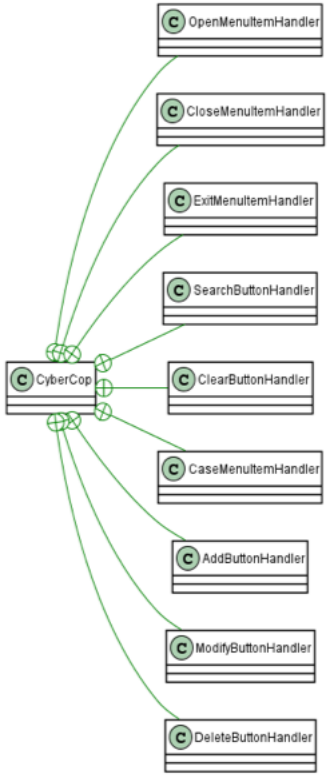

Figure 5: Handlersused in Cyber Cop 2.0

Data

TSV Data set: A sample of this data set is shown in Table 2. It has case type (i.e., Federal, Administrative) extracted out into a separate column. It has three additional columns - Case link, Case category, and Case notes. However, not all rows may have data in these columns.

Table 2: TSV data sample

Date	Title	Type	Case number	Case Link	Case Category	Case notes
2021-07-01	Kuuhuub, Inc., et al., U.S. v. (Recolor Oy)	Federal	1823184	https://www.ftc.gov/enforcement/cases-proceedings/1823184/kuuhuub-inc-et-al-us-v-recolor-oy	COPPA	Kuuhuub Inc., Kuu Hubb Oy and Recolor Oy settled FTC allegations that they violated a children s privacy law by ...
2021-05-07	Everalbum, Inc., In the Matter of	Administrative	192 3172	https://www.ftc.gov/enforcement/cases-proceedings/192-3003/support-king-llc-spyfonecom-matter	Facial recognition	Everalbum settled Federal Trade Commission allegations that it deceived consumers about its use of facial recognition technology…

CSV Data set: This data has same columns as above, but since some of the values, such as Title or Case notes can have commas within them, they are enclosed within double-quotes as shown in Table 3.

Table 3: CSV data sample

Date	Title	Type	Case number	Case Link	Case Category	Case notes
2021-07-01	"Kuuhuub, Inc., et al., U.S. v. (Recolor Oy) "	Federal	1823184	https://www.ftc.gov/enforcement/cases-proceedings/1823184/kuuhuub-inc-et-al-us-v-recolor-oy	COPPA	"Kuuhuub Inc., Kuu Hubb Oy and Recolor Oy settled FTC allegations that they violated a children s privacy law by ... "
2021-05-07	"Everalbum, Inc., In the Matter of"	Administrative	192 3172	https://www.ftc.gov/enforcement/cases-proceedings/192-3003/support-king-llc-spyfonecom-matter	Facial recognition	"Everalbum settled Federal Trade Commission allegations that it deceived consumers about its use of facial recognition technology…"

CyberCop 2.0 should be able to read data in both formats using Polymorphism. The TSV data will be read as usual by parsing and splitting data on tabs. But for CSV data, we will use a 3rd party utility called CSVParser[1] by including an external JAR file **commons-csv-1.5.jar**. It greatly simplifies and also optimizes reading large and complex csv files. The code to use this file has been provided to you in CSVCaseReader.java. To use the jar file, right click on project → Build Path → Add External JARs...Select the jar file. If this jar file is not correctly included in the build path, then CSVCaseReader.java will show compilation errors.

> ***NOTE**: To avoid hitting FTC website multiple times as you code and debug, **SmallCSV-CCData.tsv** and **SmallTSV-CCData.csv** with data about 10 cases is provided to you. These cases have local html files for case-links that will be stored in web folder as shown in Fig.6. The video clips on Canvas and the screenshot in Fig.2 show some direct links only as a sample. Please use these small files for scenario testing.*

[1] https://commons.apache.org/proper/commons-csv/apidocs/org/apache/commons/csv/CSVParser.html

Instructions:

- Download the files from Canvas and store them in a project and a package as shown in Fig. 6.
- Make sure the VM Arguments String has the following:

--module-path="YOURPATH TO JAVAFXLIB" --add-modules=javafx.controls,javafx.fxml,**javafx.web**

- Complete your code.
- Do not change the 'fully-coded' files given to you. You can modify the design of other classes as long as test-cases pass. However, making too many changes may need rework for you in HW3. .
- Write your name and Andrew id as comments at the top in Java files you need to submit. Do not submit the 'fully-coded' files given to you.
- You will test your program in two ways: GUI interaction (refer to scenarios in video-clips on Canvas) and TestCyberCop.java. These two tests will get you 80% of the points. Other criteria applied to evaluate your program are:
  1. Documentation (5%): Your code should be well-commented, i.e., neither too many comments, nor too few. Yes, this requires a little bit of your judgment! Name your variables in a self-explanatory way. Write your name and Andrew id at the top in the comments in each class. Indent your code properly. (In Eclipse, press Ctrl-A to select all your code and then Ctrl-I to indent)
  2. Code quality (5%): coding conventions, no unused variables/ libraries, etc. Use your judgment to assess these criteria.
  3. Code robustness (5%): Your program should not throw any errors while processing. You can safely assume that the user will not enter any garbage input.
  4. Submission instructions (5%): Zip your java files into AndrewId-hw2.zip. Fully-coded files or test-file should not be included in the zip. Do not submit any other folders, class files, test file, text files, and rest of your kitchen sink! Only last submission will be graded. Wrong files, incorrect package name, etc. may cost some points.

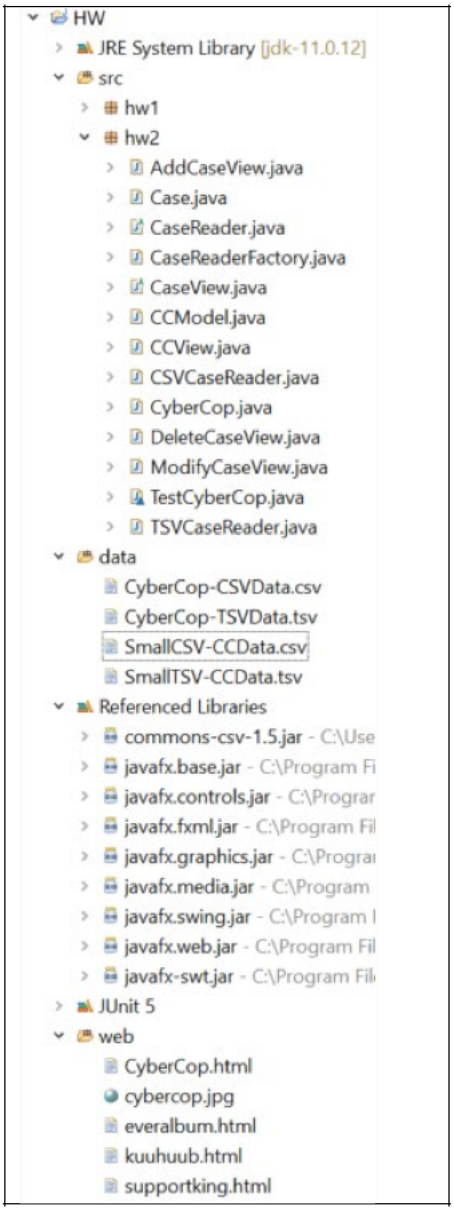

*Figure 6: Project setup. Note special icon for data and web folders. They are created as **'source folder'**. While not necessary for data folder, source folders are automatically included in build path.*

NO LATE SUBMISSIONS PLEASE! If you are unable to submit on time, you lose all the points. Please avoid last minute submission as Canvas may decide to quit on you! Learn to trust technology only to the extent you should! Do not take that risk! **Late submissions will not be accepted. Best.**