Installation and Maintenance Guide

Author Names

Brandon Bales
Emma Belanger
Matt DePero
Ryan Potts

1 Introduction

The Conflict Catcher was designed for a user to insert a .csv file containing course information for Miami University. Once submitted, the program parses through the file and identifies classes that have potentially conflicting exam times. The exam time for a class is in relation to the days it meets and the time it meets on that day. For scheduling purposes there are some classes that meet on a day and time that is not traditional. The Conflict Catcher has the ability to identify the exam times for normal and irregular classes. Once the exam times are identified, the program then shows any classes that have the potential to conflict and reports the results.

2 Installation Guide

The latest version of Eclipse must be installed in order to run this program. For instructions on how to install Eclipse see this link:
http://www.eclipse.org/downloads/index.php?show_instructions=TRUE

After Eclipse is installed, create a new project folder to hold all of the necessary files and source code. Next you want to unzip the file and move all of the files included in it into the project folder. Then, in Eclipse, open the source code file underneath the project file and open the package called “Main.” There will be two classes in this package. Open the one called “Main.java” and run it as a Java application. The interface will then run and follow the User’s Guide below to learn how to use it.

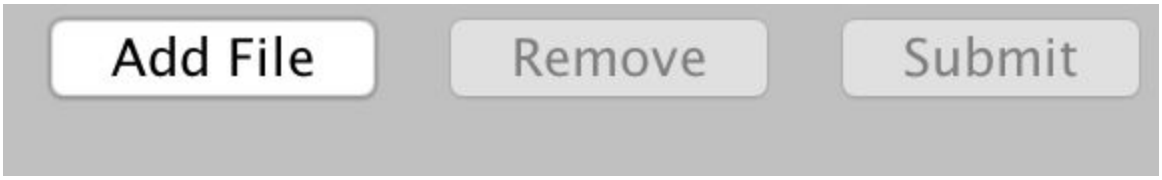
When the “Add File” button is pressed, it will display a window prompting the user for a file on the user’s computer. The user can navigate to any file including Desktop and

Documents and click on the chosen file and then click select. This is the selected file. Once the “Submit File” button is pressed, the program will read that selected file.

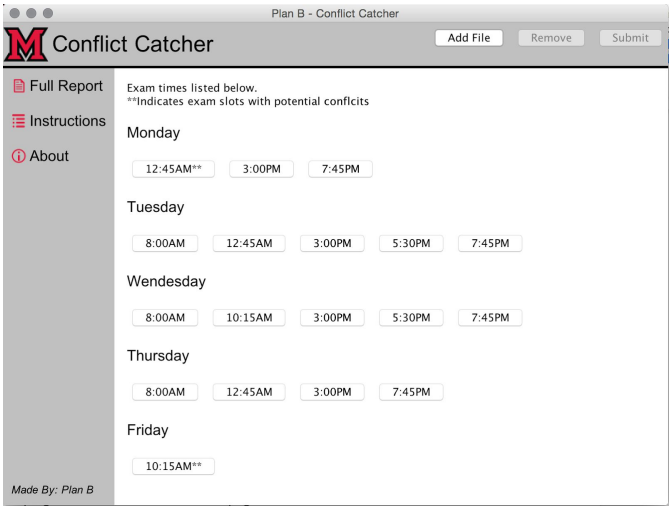
3 User’s Guide

To run the GUI, go to the Main package, and run the Main class. You will then see this interface.

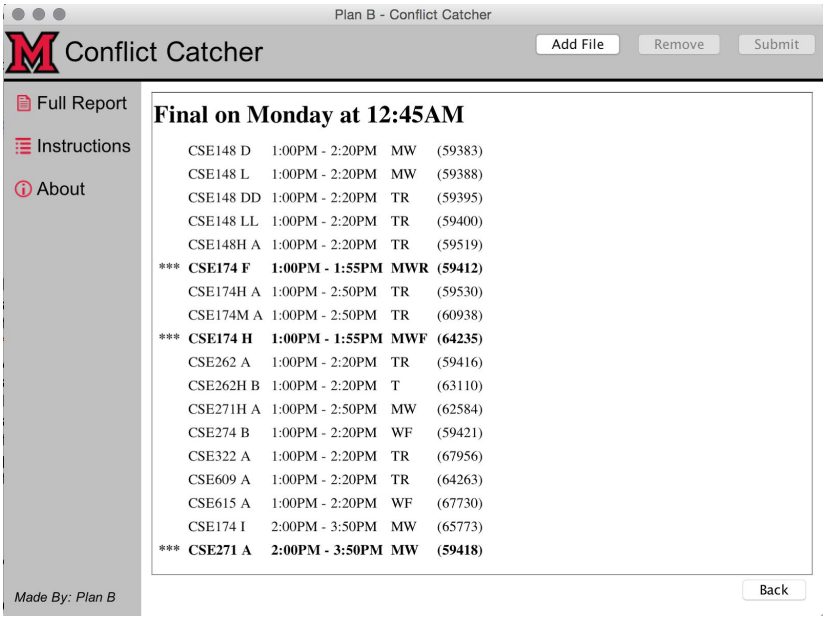
To add a file to be checked by the program click the ‘Add File’ button in the top right. You will be prompted to find the file with the file explorer which looks like this.



The software only accepts csv files. Find the file and open it. If it is the wrong file just click the ‘Remove’ button in the top right. After adding the file, click the ‘Submit’ button. To see if there are any exams that could overlap click on the text that says “Full Report” on the right. Your screen will look like this now.



If no files were submitted nothing will be shown. The times that have conflicts will have asterisks on them. Click on any of the buttons to see all classes that have exams at that time. The software will now show this.



All classes that have conflicting exams during that time will be bolded and have asterisks on the left.
It is possible for the program to read in multiple files, after submitting the first file, just add and submit a second, and the program will check all the exam times in both of the files with each other as well. The about page just gives a brief description of the

program and who all made it. If you need instructions there are also instructions listed in the program as well in the instructions page.

4 Gallery Maintenance

Our GUI was coded in a very object oriented way. There is a hierarchy to how things are placed (ex: a button is in a layout manager within another layout manager on a panel in a frame…). To be able to change how the GUI looks, or the placement of buttons and panels, you need to know how the layout managers work in relation to each other. The layout aspect of the GUI, the buttons, and the function of the buttons can all be manipulated in the main package in the Main.java file. Also inside the main package you will find a Constants.java class. In this file you can manipulate any constants such as: the width or height, or the text that is posted in the About and Instructions panels and it will change that variable throughout the entire program. So there is no need to go through each class and change it individually.

The Constants.java class there is a section of code for each panel that allows you to print text to the panel. This is done as a string, but by using HTML to format the words into paragraphs, headings, lists and whatever is needed. This is used for the About and Instructions pages.

In addition to the Main package there is also a GUI package. This package initiates the the different panels for the About, Instructions, and Report page. Lastly, we also created a second source folder named “img.” This folder contains all of the images that we used in the GUI. The img folder includes the icons that are used before About, Instructions, Report page and any other subsection labels, even the Miami “M.”

You may receive a message that says “Successfully processed file, with warnings.” printed beneath the “Add File” button in the upper left hand corner after submitting a file. This means there were classes in the file that were considered labs to a lecture class and were combined into one course, or there were duplicates that were also combined into one course.

5 Software Maintenance

The software can support other departments as long as they follow the same .csv file input format. Each course is separated into multiple parts which we defined as different classes for the software. Each course has a class, an instructor, and a subject which are all separate classes (Java classes, not school classes), and all other information that is specific to the class is located in the the Class class. To modify the final exam

assignment rules you have to go into the Category class and change all the separate category times and meeting days. Currently there is not a way to retain the results that are shown in the full report of the interface. To see how to best add these features read section 7 Known Issues and Vulnerabilities below.

6 Debugging Tips

Areas of Complexity:

- Errors and Categories classes:
    - The Errors and Categories class hold the main logic behind how the program sorts out each class into a designated exam time and determines whether they overlap. Most of our project time was spent on figuring out this logic and testing it to make sure it functions properly.

Known Previous Bugs (these are all fixed now):

- The program would not take in multiple files.
- The program would list multiple sections of the same class as conflicting with each other, even though that would be impossible for a student to be in two sections of the same class.
- The software would assume labs to be separate classes from the classes they were assigned to.

7 Known Issues and Vulnerabilities

The program can:

- Read multiple files without producing duplicate classes.
- Combine labs and lectures into one course.
- Display exam times for every class.
- Take in large files.
- Respond properly to files inputed that are not the right type or have missing information (ex. will give an error if not a .csv file).

Vulnerabilities:

- Cannot change final exam schedule through the user interface:
    - The given exam schedule for Fall 2015 is hardcoded in the Errors class, located in the CourseData package. The user has no way to change this code through the user interface. If the user wanted to change the exam schedule, the best way would be to create a new tab on the graphical user interface and allow the Errors class to take in variables that the user can

alter through the user interface. There is code in the Main.java file commented out that creates a new JPanel called “Edit.” All that’s needed is the implementation of edit and the interface to do so.

- Does not account for group exams:
    - The program does not account for group exams due to the inability to tell whether a professor chooses to have a group exam from the given information in the .csv file. In order to take group exams into account there needs to be added information for what professors are being offered group exam times, what classes, and whether the professors chooses to have them. Code in the Errors class in the CourseData package would need to be added to account for the new exam categories a course could fall under.

- Does not account for regional classes:
    - The program does not account for courses not at Miami University’s main campus in Oxford, Ohio. This means the program does not take into account courses on regional campuses. If one wanted to take regional courses new categories would need to be added and accounted for in the Errors class that follow the final exam schedule at each regional campus.

- Cannot save report to a file:
    - The program does not save the report or export it as any type of file. If one wanted to do this code would need to be added to create a method that takes the output data and instead of printing to the console or GUI it prints to a new text file.

- Search bar:
    - There currently is no search ability for the program, but there is code commented out in the Main.java class that allows for a new JPanel titled “Search.” All that would be needed is the code to implement the search function and the bar for the user to type in.