

Criterion C - Development

Introduction

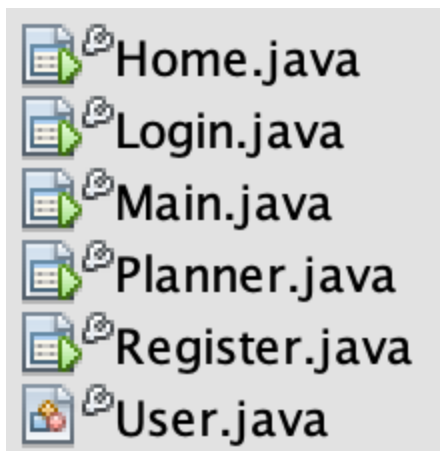
This product is made with Java in the IDE Netbeans. Netbeans is used in this program as it allows users to easily access and utilize Java's Swing Tool in order to make a GUI for their applications. This program has functions such as making and managing multiple user accounts, a working database, and accessing the internet.

List of Techniques

- Parameter passing
- Importing and utilizing classes
- Number generation and assignment
- For loop
- While loop
- Try and catch
- Threads
- Sleep method
- Set and get methods
- Method returning a value
- Arrays
- User defined objects made from an OOP "template" class
- Encapsulation of private methods
- Making an array of objects
- Buffered Readers
- Simple and compound selection (if/else)
- Searching
- Saving to a file
- Writing to a file
- Reading a file
- Utilizing paths to open applications
- Error handling
- GUI tabs
- GUI popup menus
- GUI combo boxes
- Final and new
- Parsing a file using StringTokenizer
- Inheritance between a superclass and a subclass
- Use of some other specialized library

Structure of the Program

What: There are a total of six classes in the application: “Home.java”, “Login.java”, “Main.java”, “Planner.java”, “Register.java”, and “User.java”. All of the classes, except User.java, are GUI classes, where each have a function on their own. The User class is used by the “Main” and “Planner” classes in order to display information specific for different users. Another example is how the “Home” class works with the “Planner” class in order to check if the planner for the current date is empty or not; if the calendar in “Planner” is empty for the current date, then “Home” will be able to display the button to launch the video game for users to play.



Screenshot 1: All of the classes in the program

Why: This program is a product of OOP, which means the program includes characteristics of OOP such as abstraction and encapsulation. The main reason for having different classes would be for abstraction; different classes will have different jobs which makes it easier to debug and maintain the program. Furthermore, with public and private attributes, the program’s template classes can only be manipulated through other public methods, such as “get” and “set”. This means that the program utilizes encapsulation in order to protect the program from unwanted changes.

Data Structures Used

What:

- Arrays
- Array of objects
- Files

Why: Arrays were utilized because of the ease of debugging, since during debugging the program can easily show which values are being used and sent and which were not. Another reason why arrays were used is because multiple data items can be used in under one name, meaning multiple data items under “Planner”, where data items such as the id, name, and urgency of the task is stored. Furthermore, arrays are directly accessible, meaning it is faster for the program to search for the wanted array with its wanted data items.

Screenshot 2: Utilization of arrays to help users login to the application

Adding on from arrays, Files were used in the program since they are easy to manipulate and store on a computer, rather than using technologies such as a cloud or a remote server for storing data. Files are also quick and easy to access, making the application more efficient when the files are being used.

```
public void login(){
    String usernameLogin = usernameTF.getText();
    String passwordLogin = passwordTF.getText();

    boolean isLogin = false;

    String path = "User.txt";
    File file = new File(path);

    String line = null, email = null, user = null, pass=null, id=null;

    try{
        BufferedReader br = new BufferedReader(new FileReader(file));
        while((line = br.readLine()) != null){

            String [] array = line.split("\\|");

            id = array[0];
            user = array[1];
            email = array[2];
            pass = array[3];

            if(usernameLogin.equals(email) && passwordLogin.equals(pass)){
                isLogin = true;
                if(isLogin){

                    Home homeForm = new Home(id,user,email,pass);
                    //Home homeForm = new Home();
                    homeForm.setVisible(true);
                    this.setVisible(false);

                } else {

                    int m = JOptionPane.showConfirmDialog(
                        null,
                        "Login Unsuccessful",
                        "Try again?",
                        JOptionPane.YES_NO_OPTION);

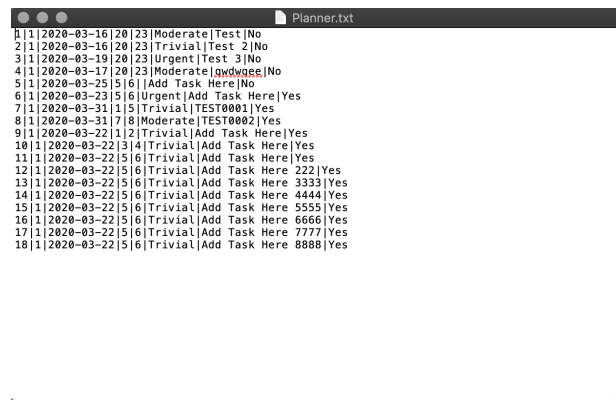
                }

            }

        }
    }
}
```



```
1|Seksan_Manee|email@gmail.com|password
2|2222|2222|2222
3|3333|3333|3333
```



```
1|1|2020-03-16|20|23|Moderate|Test|No
2|1|2020-03-16|20|23|Trivial|Test 2|No
3|1|2020-03-19|20|23|Urgent|Test 3|No
4|1|2020-03-17|20|23|Moderate|gadgets|No
5|1|2020-03-25|5|6|Add Task Here|No
6|1|2020-03-23|5|6|Urgent|Add Task Here|Yes
7|1|2020-03-31|1|5|Trivial|TEST0001|Yes
8|1|2020-03-31|7|8|Moderate|TEST0002|Yes
9|1|2020-03-22|1|2|Trivial|Add Task Here|Yes
10|1|2020-03-22|3|4|Trivial|Add Task Here|Yes
11|1|2020-03-22|5|6|Trivial|Add Task Here|Yes
12|1|2020-03-22|5|6|Trivial|Add Task Here 222|Yes
13|1|2020-03-22|5|6|Trivial|Add Task Here 3333|Yes
14|1|2020-03-22|5|6|Trivial|Add Task Here 4444|Yes
15|1|2020-03-22|5|6|Trivial|Add Task Here 5555|Yes
16|1|2020-03-22|5|6|Trivial|Add Task Here 6666|Yes
17|1|2020-03-22|5|6|Trivial|Add Task Here 7777|Yes
18|1|2020-03-22|5|6|Trivial|Add Task Here 8888|Yes
```

Screenshot 3 & 4: Inside User.txt and Planner.txt text files used to record user/planner information

Main Unique Algorithms

What: A unique algorithm used in the program would be creating and manipulating text files inside the computer in order to store and retrieve user information and planner information for the user to change and employ. Another unique algorithm would be displaying Twitter inside the application itself (for Windows OS users only). Finally, Threads are also used inside the application.

First, the method `storeData`, an example of an algorithm for manipulating text files, works overall by the pseudo code and screenshot below.

```
// Creates a new String variable called path with assigned value of "User.txt" in order to access
the text file later
// Creates String variables assigned to different text fields for the user's name, email, password,
and confirmation of password
// Creates new BufferedReader with parameters of FileReader(path) to read files of the path
variable
// If the password and confirm password matches in their text fields, prompt the user to register
// Checks if the text file "User.txt" exists on the computer yet or not; If does not exists, creates
new file with the name "User.txt" from path and starts writing information down on the text file
created
// Information is written down into text file
// Each set of information is separated with "|" in the order of how the user registered
```

```
public void storeData() throws FileNotFoundException, IOException
{
    String path = "User.txt";

    String nameRegister = nameTF.getText();
    String emailRegister = emailTF.getText();
    String passwordRegister = passwordField.getText();
    String confirmPasswordRegister = confirmPasswordField.getText();

    BufferedReader reader = new BufferedReader(new FileReader(path));
    int id = 0;
    while (reader.readLine() != null) id++;
    id += 1;
    reader.close();

    if(passwordRegister.equals(confirmPasswordRegister)){
        int n = JOptionPane.showConfirmDialog(
            null,
            "Would you like to register?",
            "",
            JOptionPane.YES_NO_OPTION);
        //System.out.println(n); // Use n for response
        if(n==0)
        {
            File file = new File(path);
            FileWriter writer;
            try {
                writer = new FileWriter(file, true); //True = Append to file, false = Overwrite
                writer.write(String.valueOf(id));
                writer.write("|");
                writer.write(nameRegister);
                writer.write("|");
                writer.write(emailRegister);
                writer.write("|");
                writer.write(passwordRegister);
                writer.write("\n");
                writer.close();
                System.out.println("Write success!");
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}
```

*Screenshot 5: Main portion of
the algorithm used to
create/edit User.txt*

Second, the method createContent's pseudocode and screenshot of algorithm:

```
// Creates a new Java Panel (JPanel) called contentPane which is assigned to a certain border layout
// Set the border layout to the dimensions (0, 0, 760, 660) to fit within window of its application
// Create a new JPanel called webBrowserPanel dedicated to display "Twitter.com"
// Use JWebBrowser to create a browser that navigates to "https://twitter.com"
// Center the content inside the pane in the application
```

```
public static JComponent createContent() {
    JPanel contentPane = new JPanel(new BorderLayout());
    contentPane.setBounds(0, 0, 760, 660);
    try{
        JPanel webBrowserPanel = new JPanel(new BorderLayout());
        webBrowserPanel.setBorder(BorderFactory.createTitledBorder("Twitter"));
        //webBrowserPanel.setBounds(38, 34, 133, 500);

        final JWebBrowser webBrowser = new JWebBrowser();
        webBrowser.navigate("https://twitter.com");
        webBrowserPanel.add(webBrowser);
        contentPane.add(webBrowserPanel, BorderLayout.CENTER);
    }catch(Exception e)
    {
        //Nothing
    }
    return contentPane;
}
```

Screenshot 6: Algorithm for displaying Twitter inside the application

Finally, the use of threads inside the application with pseudocode and screenshot of its use:

```
// Create a new Thread
// In a indefinite while loop, check lblDate (label showing today's date) on the "Home" page
// Check every 500 milliseconds with sleep(500)
```

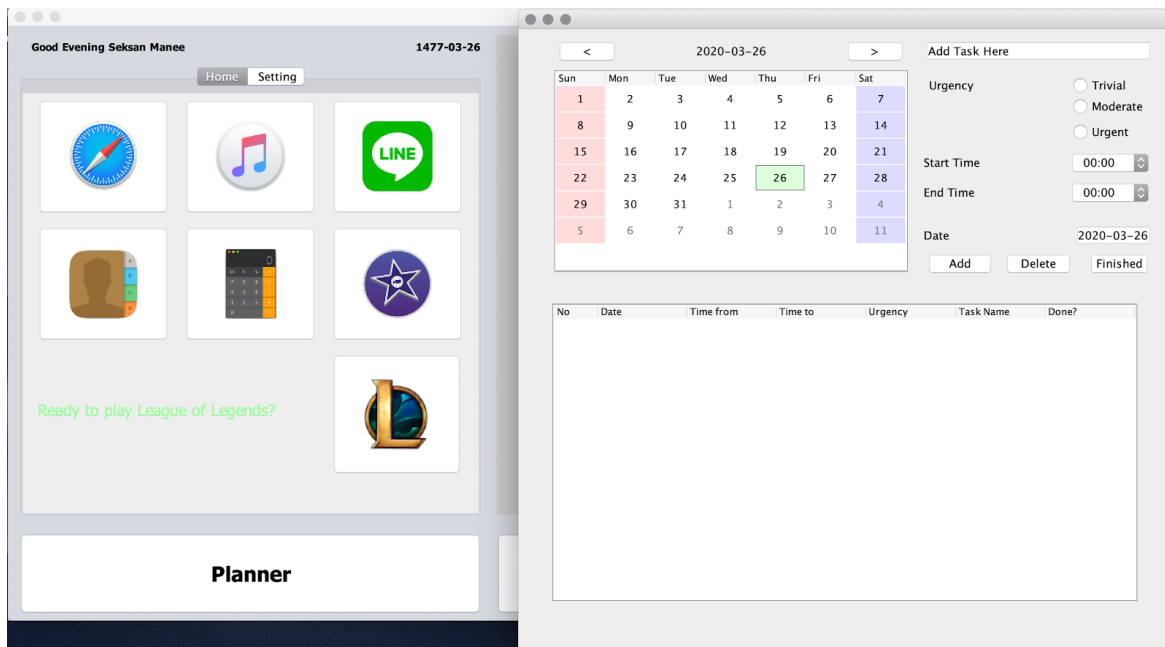
```
new Thread(() ->
{
    while(true)
    {
        checkReady(lblDate.getText());
        try {
            Thread.sleep(500);
        } catch (InterruptedException ex) {
            Logger.getLogger(Home.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}).start();
```

Screenshot 6: Algorithm for displaying Twitter inside the application

Why: The unique algorithm of writing to files is created because it has the benefit of not relying on the internet to function properly. Files can be read efficiently to retrieve both "User" and "Planner" information in order to the user to manipulate. The algorithm for displaying Twitter enables Mr.Manee to easily access Twitter (and other websites) without launching any other applications. For threads, creating a new thread will allow the application to run the process concurrently with other processes without hindering the overall performance.

User Interface/GUI Work

What: The first page that the user will see will be the “Login” page, where users will have a chance to login if they have an account or register if they do not. By clicking “Register”, the user will be navigated to the “Register” page in order to create an account. When users login, they will encounter the “Main” page, with different buttons for frequently used applications and a space displaying Twitter for ease of access to information. When users click “Planner”, an additional window to the “Planner” page will appear where they can add and remove tasks of the day in a calendar that is easy to see and manipulate.

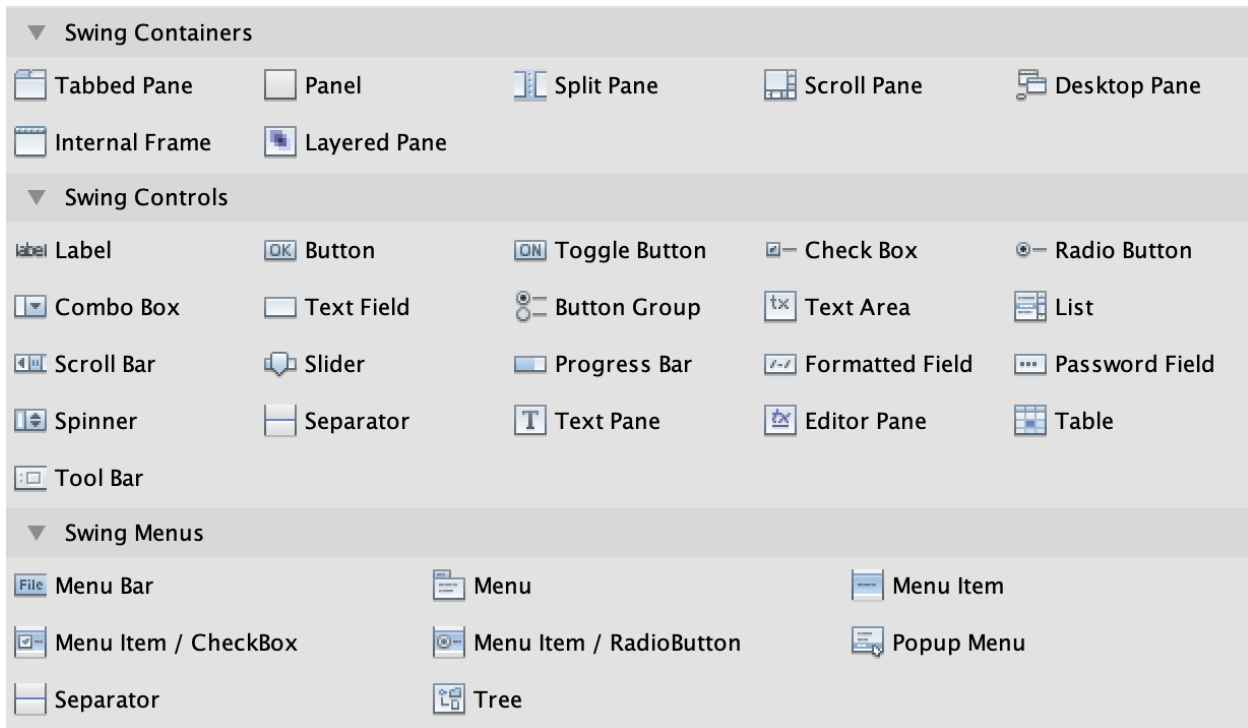


Screenshot 7: An example of the new prototype home page and planner page opening concurrently

Why: A GUI interface not only helps with organizing the program but most importantly allows the user to easily use the application. A GUI helps the user navigate and use the application efficiently since the user will know what the application will do when the user clicks on something inside. With the many pages that are easy to navigate to and get rid of, the screen is also not jumbled up and the user can easily see different parts of the application only when they want to.

Software Tools Used

What: The main IDE used was Netbeans, since it offers very easy methods to utilize Java's Swing Tool. Other than Netbeans, the IDE IntelliJ was also employed since IntelliJ offers ease of creating constructors, overloaded constructors, "get" methods, and "set" methods when creating a class.



Screenshot 8: A portion of Netbean's offered Swing components for building applications

Why: Netbeans was chosen because it had the target of creating a GUI for my client, which will allow my client to easily use the program. Furthermore, Netbeans makes creating a GUI very simple and efficient when compared to other IDEs such as IntelliJ or a text editor.

Word Count: 881