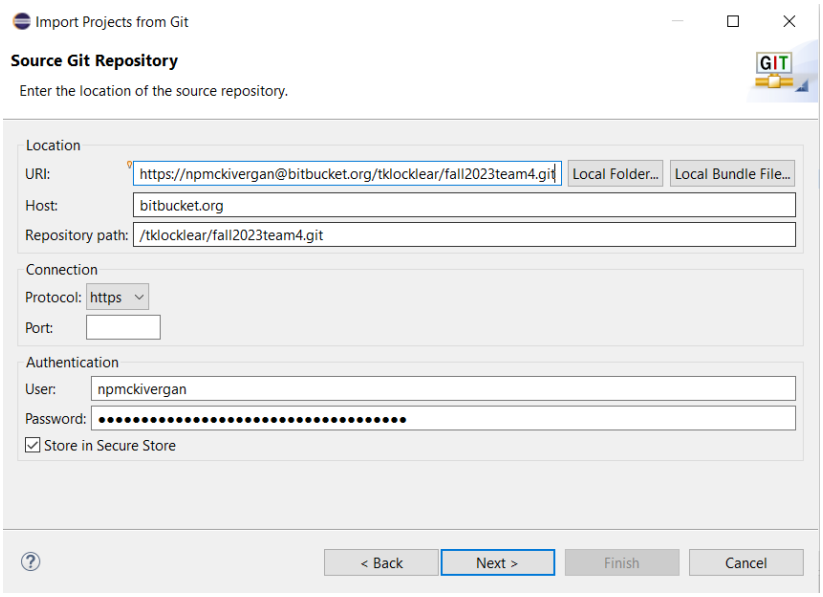
ChocAn User Manual

Table of Contents

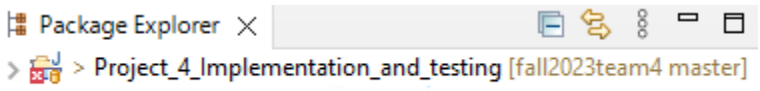
Setup

Cloning the Repo

The first step in setting up this project is cloning the remote GIT repository. To do this in Eclipse, enter the Java perspective and go to “File -> Import -> Git -> Projects from git”, and click next. Then click “Clone URL” and enter this URL: https://yourusername@bitbucket.org/tklocklear/fall2023team4.git, replacing the “your-username” field with your unique username. It should go to the following screen where you can enter your password:

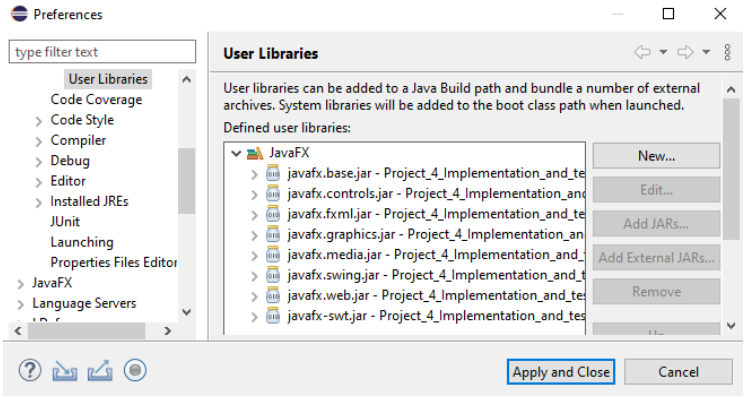


From here, proceed keeping the default settings until the pop-up window closes. Then, the repository should be accessible within the Eclipse package manager as shown below:



Installing Dependencies to Run from Source Code

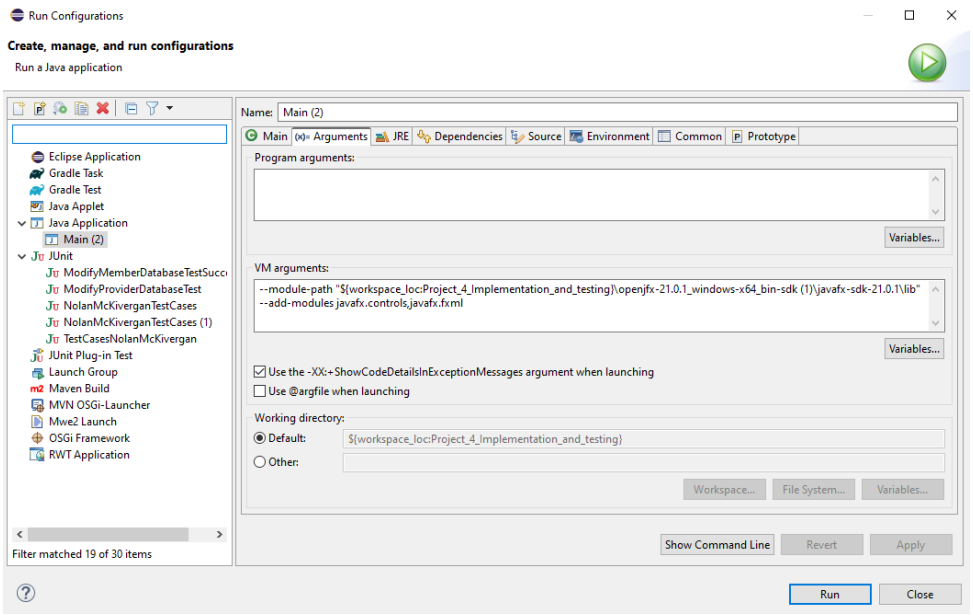
If you wish to run the project from the source code, you must first add JavaFX to the build path. To do this, install the latest e(fx)clipse from the Eclipse Marketplace. Then, select “Window -> Preferences -> Java -> Build Path -> User Libraries”. Add a new library titled “JavaFX”, select it, and click “Add External JARs”. Navigate to “your_repo_directory\ Project_4_Implementation_and_testing\openjfx-21.0.1_windows-x64_bin-sdk (1)\javafx-sdk-21.0.1\lib” with the your_repo_directory field filled with the path to your local repository. Add all the files in this directory to the library, and then click “Apply and Close”. It should look something like this by the end:



Now, right-click on the project folder and select “Run As -> Run Configurations”, and within the VM arguments field paste the following like so:

--module-path "${workspace_loc:Project_4_Implementation_and_testing}\openjfx-21.0.1_windows-x64_bin-sdk (1)\javafx-sdk-21.0.1\lib"

--add-modules javafx.controls,javafx.fxml



Now, you should be able to run the project from source code.

Creating the JAR

To create the JAR, right click on the project folder and select Export. Click “General -> Ant Buildfiles”, and proceed to the finish. This will generate a build.xml file. Open that file, and paste the following code as the last target tag in the file:

```
<target name="create-jar" depends="build">
  <mkdir dir="release"/>

  <jar destfile="release/project_4.jar" basedir="bin">
        <fileset dir ="bin" includes="**/*.class"></fileset>
        <manifest>
              <attribute name="Main-Class" value ="Software.Main"/>
        </manifest>
  </jar>
</target>
```

The end of the build.xml file should look something like this:

```
164    <target name="junitreport">
165        <junitreport todir="${junit.output.dir}">
166            <fileset dir="${junit.output.dir}">
167                <include name="TEST-*.xml"/>
168            </fileset>
169            <report format="frames" todir="${junit.output.dir}"/>
170        </junitreport>
171    </target>
172    <target name="create-jar" depends="build">
173        <mkdir dir="release"/>
174
175        <jar destfile="release/project_4.jar" basedir="bin">
176            <fileset dir ="bin" includes="**/*.class"></fileset>
177            <manifest>
178                <attribute name="Main-Class" value ="Software.Main"/>
179            </manifest>
180        </jar>
181    </target>
182 </project>
183
```

Finally, save the file and navigate to its parent directory in the command line. From the command line, run the following command: “ant create-jar”

NOTE: If you do not have ANT installed, you may do so here: https://ant.apache.org/manual/install.html

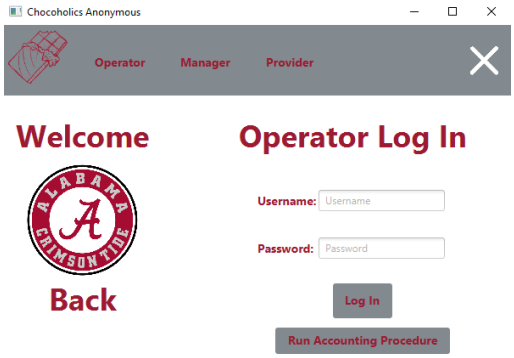
Now, there should be a JAR titled project_4.jar within the release folder.

Use

To run the program from the source code, hit the large green play button. To run the JAR, double-click the JAR file.

General Operations

Upon starting the application, you will be greeted by this screen



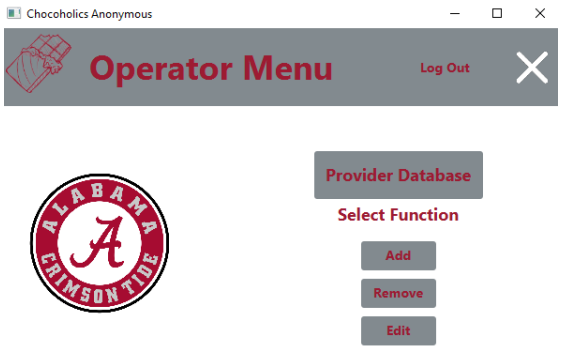
There are options to log in as a user, but a few operations are accessible from this page.

To run the main accounting procedure, simply click the “Run Accounting Procedure” button.

To close the application, either close the window or click the large white X.

Logging in and Out

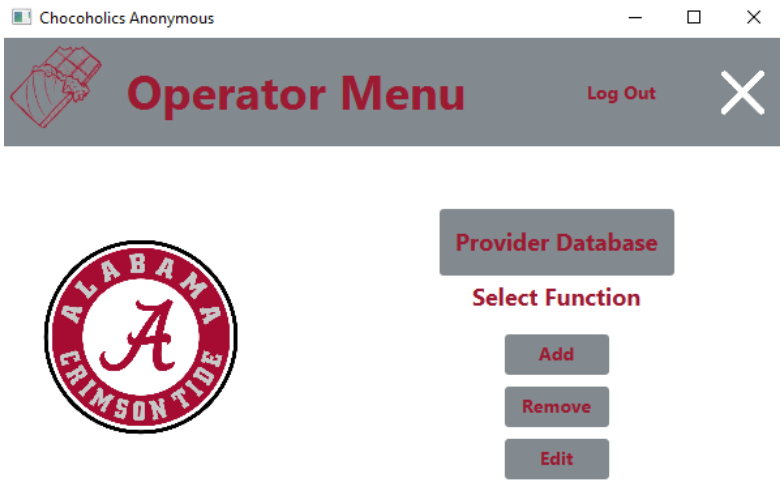
To log in, select the type of user you are logging in as and enter your login credentials. If successful, you will be presented with a menu of options for that type of user like in the example below.



To log out at any point click the “Log Out” button or the chocolate bar icon.

Operator Operations

After logging in as an operator, you will be greeted with this menu:



As an operator, you can add to, remove from, and edit the provider and member databases. Click on the “Provider Database” or “Member Database” to toggle back and forth between the two.

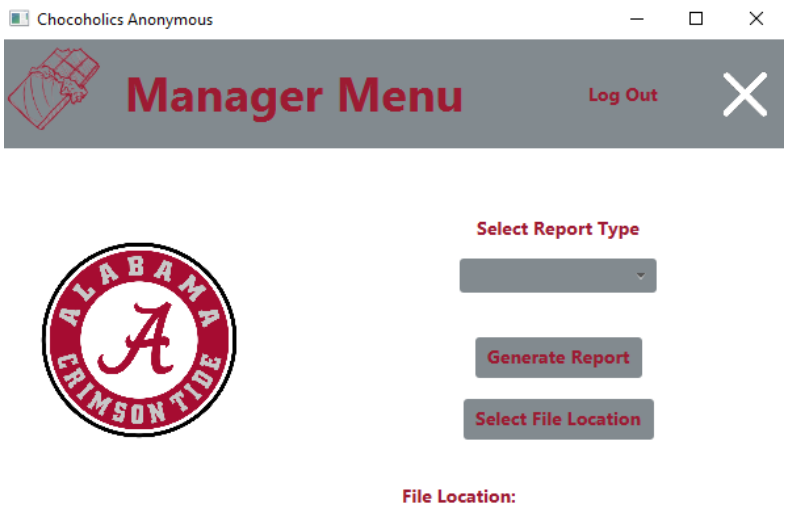
The Add operation prompts you for the new members information. It will assign the new member or provider an ID number and add them to their respective database.

The Remove function prompts you to enter a member or provider ID number, and if it is valid, will remove them from the database.

The Edit function prompts you to enter the member or provider ID number, and will display all the current information about that member or provider. You can change any fields that need to be changed, and then save the changes.

Manager Operations

Managers are shown this menu after logging in:



Managers can generate a report, but must first specify a report type and location respectively.

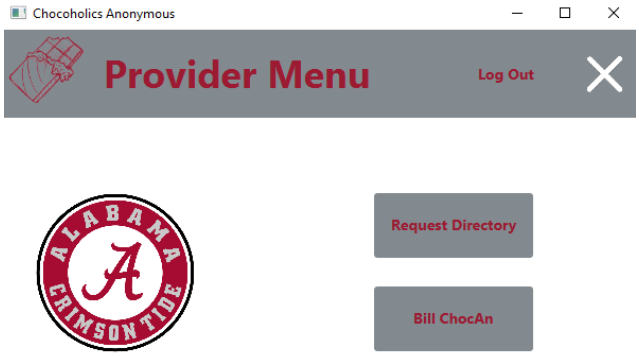
First, click “Select File Location”, and navigate to the location on your machine where you want the generated report saved to.

Next, select the type of report you’d like to generate from the drop-down menu.

Finally, click “Generate Report”, and the report will be generated and saved at the specified location.

Provider Operations

Providers have these options after logging into the system:

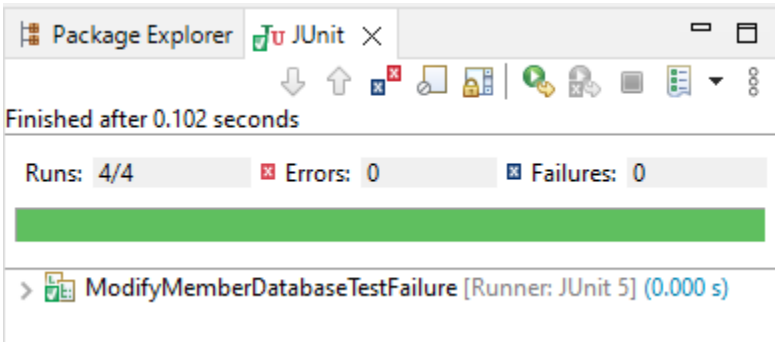


The “Request Directory” button creates the provider directory as a file called report.txt and places it within the Software package.

The “Bill ChocAn” button begins the billing procedure. The provider is prompted to enter the member number to bill for and then the code associated with the service provided. Then, the provider enters additional information about the visit and their own number, and clicks “Bill” to sent the bill to ChocAn. At any point in the process the provider can click “Load Information” to validate the member number and service codes are valid.

Validation

Each component of this software system can be individually validated by running the included JUnit test cases. To do this in Eclipse, within the Java perspective navigate to “src -> Tests”, which is the folder where all the test cases are held. To perform the test, right click on the test you wish to carry out and select “Run as -> 1 JUnit Test”. The following screen will pop up indicating which tests passed and failed:



Here, all the tests passed, indicating the component is working as desired.