Feldman M. Bagalso

BSIT - 2D

**Day 1**

First you open netbeans, then you click open project, open the designated project which is the bread app. Once you open the project, the project will have an unloadable project. To fix this error, you right click the project and click ‘clean and build’ This will adapt the project to your computer as it will download some plugins and extra tools so that the project can have a successful build.

Then once you have a successful build after waiting, another error will appear which is running the project. To resolve this, you click run, go to set configuration, click customize and you must browse the main class for the project to be able to run.

To implement Springboot fx click tools on the windows panel, then click options and then you click JavaFX. This will install the scene builder towards Java. Close netbeans and open again for the settings to take effect. If we want the fmxl file to open the scene builder, we first select build, then click tools, and then click java and finally javafx before clicking ok.

Then we go with the enumeration class. When an application starts, it will load the fmxl file to the scene builder, this is a class but it is called enum. It contains the values and it is almost similar to a list.

Once we are done with the process, we go on to creating databases. Xampp control panel is needed because it will make the specific tools and browsers to work. Before we start creating our database, we first must click ‘start’ Apache and MySQL but in order to test apache, it must be closed.

Next, open the browser and type localhost, this will show that the browser cannot be opened nor breached, this is because that the Apache is disabled. Activate Apache and this will grant you access to the website, MySQL ensures that you also have access to the website’s server. In order to create a database, click on the database, click the create button and name the database, then you select the number of columns that you intend to create.

From there on, you will be given a table with several columns to input the name, the type, the length/values, the default, the collation, attributes, null index, A.I comments, and virtuality. But we don’t have to input on them all, only input the name, the type and follow the specific characteristic of each column. Once you are done, you will see the structure of the table you created.

**Day 2**

First load all the necessary tools needed to continue the program such as Netbeans, XAMPP, and the localhost browser. Open the demo files of the breadapp and you will see a FXML file called ‘User’ Delete the existing fxml file in netbeans then drag and drop ‘User’ from the demo files folder to fxml folder in netbeans. Run netbeans and then you will encounter an error. Do not look at the last part of the output to see the error, instead look at the first parts of the springboot output. From there on, you will see the existing errors in a more specific way.

Once you find the problem, which the problem is about the usercontroller, go to the demo files and drag and drop the usercontroller to the main package with the main class. Run the program again and another error will appear, but the usercontroller error is not found anymore, but we’re still unable to load the fxml view.

Open UserController and you will see that the variables are commented out. It shows that UserController is implementing an interface and it is required to override the method. Right click fxml and place it on the right to be able to see the UserController window and fxml. Then at the UserController uncomment the variables. Go back to the demo files folder and get the models, generic, repository and services, drag and drop them to the main class. You will get several errors for this.

Go to user repository and try to import the Jparepository class. To get the dependency, you must go back to the initializer website. But we don’t need to follow that, go to the demo files and you will see a notepad called JPA and MySQLconnector. Copy those and go back to netbeans to paste it on pom.xml. In pom, you will see all the versions and you will see the version of your springboot that depends on the version that you must download. Then, at the last dependency, this is where you paste it. Format to code so that the code will look more organized. Right click the project and build with dependencies.

You will get another error again in UserController which tells that user does not recognize the user model because it is commented. Uncomment the import statement for user model and build and run again. You will finally get a build success. Open userservice and you will see the methods included in the bread app. You will see all the methods in this program such as the save, delete, update and such.

Go to other sources folder, select src/main/resources, select default package and go to bundle properties. This is where you place your window title only. In application properties, this is where you place go to the demo files and open the notepad application properties. Copy paste it on the application properties program folder. Change the database name and build and run. This is now you will get the database table, it will show a blank first but it will be added once you input a firstname and lastname. You are now done with users table.

Now the next objective is to create a new database table focused on products. You will need productrepository, productservice, product model, and productcontroller. You also need productfxml. Go to the localhost phpMyAdmin and create a new database table. Name it to products. Insert the necessary inputs and data types.

Go back to netbeans and go to other sources package, duplicate user.fxml and name it to product.fxml. Open the product.fxml in scene builder and remove most of the content except the ID, Table, column for the ID and others. Then you start renaming the contents of the table. Go to the right side of the code section and you will see the editing panel, edit the fx:id of each content from “user” to “product” save the table and check for any mismatches.

Now go back to netbeans and go to bundle properties. Copy and paste, then change the title user to products. Go to Fxmlviews data. Highlight a certain code, copy it right before semicolon, add a comma and then paste it. Don’t remove the semi colon. Next change all the ‘user’ and ‘users’ into ‘product’ and ‘products’ Get to initial scene method and change it to PRODUCTS

Copy and Paste UserRepository and name it to ProductRepository by refactoring it. Change the import of the program so that it will get from the product model, it will have an error because a product model does not exist yet. Do the same to IUserService, name it to IProductService, refactor it and change its import to product, another same error will come. Now do this towards UserService and name it to ProductService, then change all the ‘user’ and ‘users’ to ‘product’ and ‘products’ using a ‘find with what’ tool to easily replace all the same words to a desired word. Make sure the match case is enabled.

Under the models package there is the user model. Copy and paste, refactor it and rename the table annotation to products, then delete code that relates to all lastname, dob and gender. Also change the getters and setters from ‘user’ to ‘product’ as well as other variables. Find all the getFirstname and replace them to ‘getFroductname’ in the program. Do the same to setFirstname as well. The firstName variable also needs to be replaced to productName using the replace all tool.

Next step is to delete some of the unnecessary variables that don’t relate with the database table. So deleting lastname, dob, gender, columns, go to the saveProduct methods and remove the unnecessary variables such as setLastName, setdob and setgender. Go to clearFields and remove the unnecessary variables.

Once you are done, clean and build the product model, and if there are any errors, find the errors by looking at the first parts of the output. Once you saw the hint of the error, you can start debugging whatever problem is given until a solution is found. Also keep in mind that the name of the database tables have to match with the names from the program or else, the table that you made will never function when you input something on the ID and Productname.