PEREZ, PAULYNN YSABELLE A.

BSIT-2D

1. Download the file that is in the download-links in Discord that will lead us to the Google Drive of our Instructor
2. Click the “2nd Week of February Activity” and download the BreadApp Demo Files and BreadApp Demo
3. Open the Netbeans
4. Click File and then click Open Project and look for “BreadApp”. Once we open it, we will see the word “ unloadable”
5. Right click the BreadApp then select Clean and Build. Netbeans will update the files that are in the BreadApp and adapt it to the device that we are using
6. If it says “Build Success” then we have a blank project to start with
7. But there is still some error which is running the project. To settle with the problem first right click the breapp then set configuration click the customize and then run and click the browse for the project to run.
8. On the window panel click tools for us to implement the Springboot fx and then click option then click JavaFX. To install the scene builder to Java.
9. To open the scene builder we need to select build then click tools and then java then javafx and click ok
10. We go to the enumeration class. When the application starts it will load to fxml to scene builder it is called enum that contains the value that are in the user.fxml
11. We need to make a database first. We open the XAMPP Control panel, Click ‘start’ for Apache and MySQL, but in order to test apache, it must be closed.
12. Open the browser and type localhost, this will show that the browser cannot be opened, Because the Apache is closed. We must Activate Apache and this will grant you access to the website, same with MySQL. 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 want to create.
13. 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. Only Input the name and the Type. Once we are done there will be a structure of table that we make
14. Before we start, load the necessary tools that we need. Then, Check the diagram again to check what are the

next step that we're gonna do. Open the breadApp

Demo Files and open "User". Go back to Netbeans

and click styles.css then right click the

user.fxml and delete it. After deleting the User

in the Netbean. Drag the User file on Netbeans again

and run it. We will get an error. To look for an

error never look on the bottom part because that

is the previous error that we make. Check the

error.

1. In order for us to fix the error we drag and

draft the controller folder to the Main class.

Check the output again and run again the program

but were still getting an error which is we can't

load the fxml view. Go to scene builder and go

back again to Netbeans to see click the

UserController. So when we click on

UserController were gonna see a word initialize

so what it does is to get all the users into

table and load the users' details.

1. Right click the user.fxml and click 'edit' the code that is inserted in the user.fxml are set in private in the UserController.Go back to the diagram and see what is the next step that we're gonna do. After that, drag and draft the 'models', 'generic', ' repository' and 'service' in the Main Class.
2. There's an error after we add those in our Main Class, check again what the problem is. Open the UserService and we get an error in the 'import javax' code that we have and some parts of the code. So what we do is open our browser and search 'SpringInitializr'. We click 'Add

Dependencies' and put mySql. Then open the BreadApp Demo Files again and click the ' JPA and MySQL Connector Dependencies' we select all and copy/paste it to the Netbeans in 'pom.xml'. We can see in the 'pom.xml' what version of the device we're using if it is the latest or the previous one. So below the '<depencies>' code we paste the code from 'JPA and MySQL Connector Dependencies' . \*If it doesn't work restart the Netbeans and Format the code. Formatting the code makes it easier to read\*

1. After opening the Netbeans right click the breadapp and click run and dependencies. Check the code and we no longer see an error.

Go to UserService again, as we all can see there are two 'delete' methods. \*We can use override if we have the same method but have different parameters\*. After checking the code from UserService we open the UserController and there we can see the code from UserService but they are in private(-). After checking if there's no error, run the program and then we're gonna get the output which is in the scene builder but if we haven't started doing the program there's no output that will be shown but the Scene Builder will show up.

1. If we want to add another data to our Scene Builder, we need to edit our database first. In the table name put 'products' then input a name we use 'id and the length which is '11'then put it as primary. Input another name and call it 'product\_name' then the type is 'VARCHAR' and the length is the maximum which is '255'. We run it, we should get the data that we input on the table.
2. Go back to Netbaeans and add another folder on the fxml and we name it 'Product'. Then go to Scene Builder and change the User to Product. Delete everything that we're not gonna use and only the 'First Name' and the Back and Start Button. We need to change its properties too. Go back again to Netbeans and click the 'Bundle.Properties" and we add 'Property.title=Property Dashboard' Then to fxml view. We add the Copy code for the User and Paste it after the closing bracket but we need to put a comma first and paste it. Change everything from User to Property.
3. We copy the UserRepository and paste it, change the name to PropertyRepository and refactor. Same process from fxml.view we change everything from User to Property.
4. We copy the IUserService and paste it, change the name to IPropertyService and refactor. Same process until we are finished editing and deleting the unnecessary code or the code we no longer use. \*If the set of code are long, use "Replace" then on the bottom part where gonna see 'find What' and 'Replace With' like "Find What: user" and then "Replace With: Product" it will replace all the code that have the word user and replace it to Product.