|  |  |
| --- | --- |
| **Project – External Documentation** |  |
| ISYS6197 | ISYS6197003  Business Application Development |
| ***Valid on*** *Odd Semester Year 2023/2024* |

* Project Title

Business Application Development - cangkIR

* Introduction

cangkIR is a store on the internet where you can buy different kinds of cups. The cups come in different shapes and sizes, and they have a wide range of prices. cangkIR doesn't have a physical store that you can visit. To buy cups from cangkIR, you need to use their application. In the app, people can either be an Admin, who manages the available cups, or a User, who can buy cups by putting them in a cart and checking out. Admins can do things like adding, updating, and deleting cups from the database. Your friend IRvin wants to start an online shop too, and you've agreed to help them make the app. The app should be created using the JavaFX programming language and MySQL for the database. Also, make sure not to use regex, Java FXML, JavaFX Scene Builder, or any library outside of what's covered in our class materials when building the application.

* Report / Documentation

1. In this project we will choose to use MVC architectural pattern. Where our project was divided into 3 main packages. The first is Model, this package will store the class that have to used to modelling the data inside database. This class will represent the table inside the database. The second is View. View package containing class that used to arrange UI elements. Inside the class we define how the ui and javaFX component will be arranged. Last is Controller, this package contains app logic like event handling in this class we define how all components will be interacting with database and application logic. We integrate controller and view class by using inheritance, view class inherited by controller class. To integrate all of this class, we define it as the main class. In the main class we create each controller class object and run initialization function of each page in start method of main class so the page will be built.
2. For the page that need table view, we create model using class SimpleStringProperty and SimpleIntegerProperty, we use this class so if we want to bind model class with tableview, it’s easier to do because we just need to get the property when we want to set the cellfactory in tableview.
3. Login Page A screenshot of a login box

   Description automatically generated

This is the login page, it requires to input user’s name and password to login to the home page, or redirect to the register page if account is not created yet by clicking the hyperlink. This page uses vbox to contain login label, username grid, password grid, login button, and register link. This page also uses grid pane to set the username grid and password grid. This page is created by using these components:

* **Label** for the login title, username title, and password title,
* **Text field** for the username input
* **Password** field for the password input
* **Button** for login button
* **Hyperlink** to redirect to register page
* **Alert** to display pop-up errors

Algorithm:

* So the first step to login, I will check the username inside the field text, is it exist or not (this function will select the username based on the database, if the user data not found -> it will be get an error, The function will return false if get an error. Otherwise, if it found, the function will store the selected row into user object and also return true.
* If username found, it check again is the password is correct according to the username in database or not. If true the it will check the role, is it admin or user. If user redirect to homepage, if admin redirect to cup management.

A screenshot of a computer

Description automatically generated

This pop-up error message validation occurs when either username field or password field is empty.

1. Register PageA screenshot of a computer

   Description automatically generated

This is the register page, it requires username, email, password, and gender to create an account or redirect the page to login page if the account is already created by clicking the login page hyperlink. This page is created using the vbox to contain the register label, username grid, email grid, password grid, gender grid and button, and login page hyperlink. This page also uses grid pane to set the username grid, password grid, email grid, and gender grid. To get the user id is to count total row in the table msuser. This page is created using these components:

* **Label** for register title, username title, email title, password title, and gender title
* **Text field** for username and email input
* **Password** field for password input
* **Radio** button to select gender
* **Button** for register button
* **Hyperlink** to redirect to login page
* **Alert** to display pop-up errors

Algorithm:

* First step retrieve the data from every field text that exist, after that the data will be checked based on the all validation like email must ends with @gmail.com, password length, and for alphanumeric validation we use function isLetterOrDigit when looping into string that have been converted to array of char. This function check is the character a letter or digit if not both the function return false.
* If all validation have been passed, the next thing it needs to check is the username and email are unique or not. This done by using try and catch. Because I modified the column for username and email by adding UNIQUE keyword. The sql with give me an error if the username or password is not unique. If get the error, the alert pop up will be shown.
* If email and username is unique. The data will be inserted by using INSERT query to database. For the userId we use these following step
  + First count all row in the database using aggregate function COUNT
  + After get the count of all rows we use string format to padd 0 to the id string + count result + 1

A screenshot of a computer

Description automatically generated

This is one of the pop-up error message that will occur when the username field is empty. There are many other errors that will happen such as: same username in the database, email format, password length and alphanumeric validation. The difference just the content text inside it.

1. User Navigation BarA white screen with a black border

   Description automatically generated with medium confidence

This is the user navigation bar. This navigation bar is created using border pane, menu bar, menu and menu item. The components used are:

* Menu bar
* Menu
* Menu item

By pressing the menu bar, it will display a row of menu item of home, cart, and log out menu. By pressing the home menu, it will redirect user to home page. By pressing the cart menu, it will redirect user to cart page. By pressing the log out menu, it will redirect user to log in page. The function is simple I just set stage to the scene that I would like to access and done. To access the scene in other class we are using static keyword so it can be accessed without make an object

1. Admin Navigation BarA white background with a white border

   Description automatically generated with medium confidence

This is the admin navigation bar. This navigation bar is created using border pane, menu bar, menu and menu item. The components used are:

* Menu bar
* Menu
* Menu item

By pressing the menu bar, it will display a row of menu item of cup management and log out menu. By pressing the cup management menu, it will redirect admin to the cup management page, and by pressing the log out page will redirect admin to the log in page. The function is simple I just set stage to the scene that I would like to access and done. To access the scene in other class we’re using static keyword so it can be accessed without make an object

1. Home Page A screenshot of a computer

   Description automatically generated

This is the home page which is used to select which cup the user want to add to their cart by selecting the cup from the cup list and choosing the amount of the cup using the spinner and adding it to the cart by pressing the add to cart button. The total price of one cup type will be multiplied by the number user chose from the spinner. This page uses vbox to contain the cup list label title, cup name label, price label , cup list table, spinner, and add to cart button. This page uses grid pane for cup name label, price label, storing cup list and price labels, cup list table, cup list label and the vbox. This page is created by using these components:

* Label for label title, cup name label, and price label
* Table view to view the cup list and price table
* Spinner to select cup’s quantity
* Button to insert the cup name and quantity to the cart
* Alert to display pop-up errors

Algorithm

* First step is retrieve all the data inside the database using SELECT statement in sql. After executing query, the data that have been filled in result set will be transferred to observable list. This observable list consists of cup model.
* After the data have been retrieved, we bind event listener to table view. Every selected row is clicked we will get the object of the row. After it get the cup price and cup name. Set the label text to the price and name based on the object that have been selected.

A screenshot of a computer

Description automatically generated

This pop-up error message occurs when the add to cart button is pressed without selecting any cup from the cup list table. There is also another pop-up information message if the add to cart button is pressed with selected cup that is item successfully added to the cart.

1. Cart Page A screenshot of a computer

   Description automatically generated

This is the cart page. This page displays user’s cup list cart. This page is used to proceed user’s cart page to the checkout confirmation pop-up. In this page, user can also delete their item by selecting the cup in the cup cart list table. In this page user can also choose the courier service and also choose to include delivery insurance. The total price will be calculated by adding all the cup’s price in the cup cart list table. This page is created using vbox to contain user’s name cart label title, delete item label, courier label, courier price label, total price label, user’s cup cart list table, delete item button, checkout button, courier combo box, and the use delivery insurance checkbox. This page uses grid pane to set delete item label, courier label, courier price label, total price label, storing cup name, cup price, quantity, total labels, user’s name cart table, user’s name cart label and the vbox. This page is created by using these components:

* Label for user’s cart label, delete item label, courier label, courier price label and total price label
* Table view to view the user’s cart cup(s) inside the database
* Button to delete items from the table and checkout
* Alert to display pop-up errors
* Combo box to choose courier choice
* Check box to choose to use delivery insurance option

Algorithm:

* First retrieve the cart data based on the user that have been login. It was done by accessing the user login object inside the login class. We use static keyword so it can be accessed directly through the class. For showing the data in table view we use similar step from previous page, using SELECT statement, transfer from result set to observable list.
* Inside the retrieval process we also sum the price of all that cup so it can updated the total price label.
* For the courier combobox we retrieve it same as retrieve the cart data, Use select query to get the data, store it on observable list. We use setConverter to convert the display in comboBox to string which is the name of the courier.
* For the total price logic we use this following step:
  + First check, is the courier combo box is already selected or not? If have been selected, get the data and we store the selected courier object into variable. Sometime user will change the courier so to handle the price we store it on courier price variable. If user choose another courier the price will be updated, decrease the total based on current courier price and add again based on new courier price.
  + For use insurance the logic is quite same, if check the checkbox, total price will be added 2000 if not, will be decrease 2000
* Every time the cart is update, the total price will be restart, and component like courier combo box and use insurance check box will be set into default state.
* To handle the refresh data, we handle it by binding an event listener to the scne. If mouse is hover to the scene, refresh function will be executed. But we already make sure to be only once using the refresh function every time the cart menu is opened.
* Checkout process we following these steps:
  + First check the table is empty or not, if not we can proceed to the next step
  + There is pop up confirmation will be shown, and if user click yes the process will continue.
  + All data that need to be in transaction table is retrieved based on user that been login, courier that been choosed , use insurance or not and insert it using INSERT statement in sql.
  + Next step to add transaction detail, we loop through in cart items, and execute the INSERT function
  + After the data have been inserted, table will be cleared and all the component reset into default state.

A screenshot of a computer

Description automatically generated

This pop-up error message occurs when the delete item button is pressed without selecting any cup from the user’s cart cup list table. There is also another pop-up information message if the delete item button is pressed with selected cup that is cart deleted successfully.

1. Checkout Confirmation Pop-up A screenshot of a computer

   Description automatically generated

This pop-up window shows when checkout button from the cart page and the user’s cart is not empty. This window will be closed if the user chooses no and will return to the cart page. Otherwise when the user chooses yes, user’s cart will be emptied and a new transaction header and details will be created, the window will be closed and will show an pop-up information message that the checkout is successful. This window using jfxtas library, we store the window inside the group component or wrap it inside group, we did this due to the error that we faced if we not wrap it with another pane, we get null pointer exception every time mouse is hovering top on the window.

1. Cup Management Page A screenshot of a computer

   Description automatically generated

This is the cup management page which admin can see all the cups inside the database. They can add cups to the database by pressing the add cup button with the cup name and cup price field filled. They also can update the cup’s price by selecting the cup in the table and change the cup’s price in the cup price field and pressing the update price button. Lastly they also can remove the cup in the database by selecting the cup in the table and pressing the remove cup button. This page is created using vbox to contain cup management label title, cup name label, cup price label, cup name text field, cup price text field, add cup button, update price button, remove cup button, and the cup list table. This page uses grid pane to set cup name label, cup price label, storing cup name, storing cup price, cup management label and the vbox. This page is created by using these components:

* Label for cup management label title, cup name label, cup price label
* Table view to view cup list in the database
* Text field for cup name field and cup price field
* Button to add cup, update price, and remove cup.

Algorithm

* For showing current cup in table view, we use similar step in home page, use SELECT statement, transfer the data inside result set into observable list, and set items for table view to observable list.
* For the add cup we use these following step:
  + First check if all the name and price field are been filled or not. If yes we can continue to insert the data
  + Second, we will check is the cup name is exist or not, we handle this by using try and catch, sql will return the error because we have modified cup name column to be UNIQUE.
  + If cup name don’t exist yet, insert the cup into table
* For the Update price we using these following step:
  + Check whether any cup that have been selected or not
  + Use UPDATE statement in database to set new price based on price field text
* For the remove process we using these following step:
  + Check whether any cup that have been selected or not
  + Use DELETE statement to delete cup based on selected cup

A screenshot of a computer

Description automatically generated

This is one of the pop-up error message that shows when admin presses the update price button without selecting any cup from the table. There are many more error pop-up messages that will show such as: not filling the cup name field, cup name already exists, cup price range, and information message when all validation is accepted.

* Reference
* <https://docs.oracle.com/javase/8/javafx/api/toc.htm>
* <http://jfxtras.org/doc/8.0labs/jfxtras/labs/scene/control/window/Window.html>
* <https://shorturl.at/iquLX>
* Group Member
* 2602116264 – Jovito Rizki Gunawan
* 2602115886 – Billie Winardi
* 2602114896 – Benedictus Julius Vanesa
* 2602110992 – Jolin Christie Lee