SWT-AirlineReservationSystem

Jana Grunewald, Jeffry Munoz,

Breanna Rhodes, Tamara Vergara

10 February 2021

CEN 4072 – Software Testing (CRN 10421)

Group Project

2021

Instructor: Dr. Buckley

Software Engineering Department

Florida Gulf Coast University

Ft. Myers, FL 33965

Table Of Contents

Group Project Progress Report

Date: 11-Feb-2021
Course Code and Course Title (CRN Number): CEN4072 Software Testing (CRN 10421)

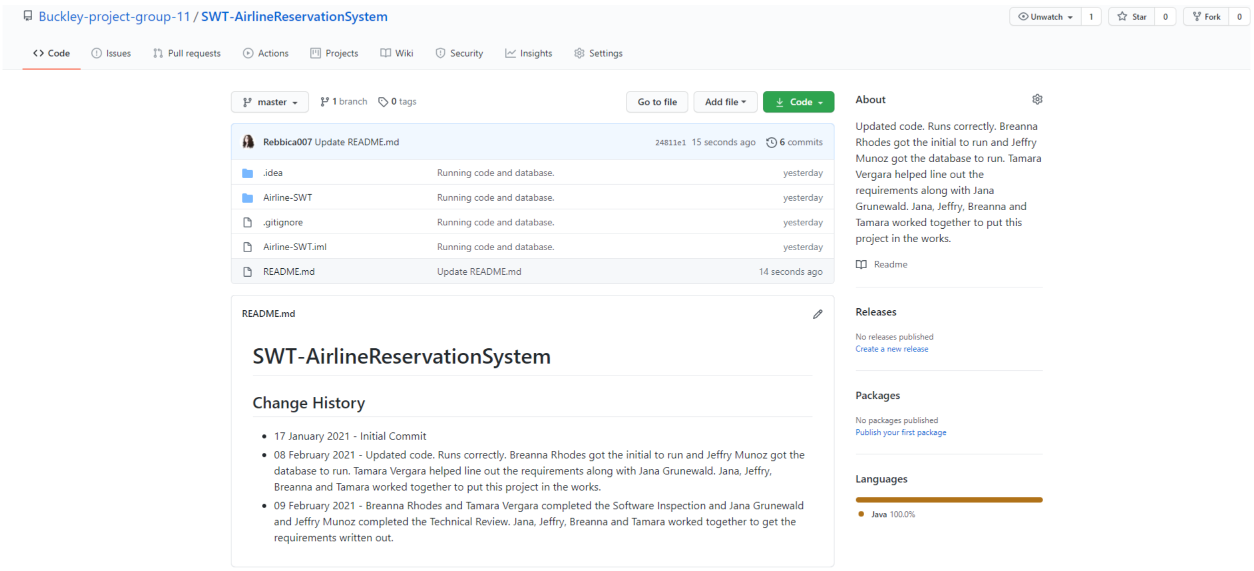
Software Application Title/Name: SWT - Airline Reservation System
Project Manager: Jana Grunewald
Group Members Name:
Jana Grunewald
Jeffry Munoz
Breanna Rhodes
Tamara Vergara
Project Phase (Milestone) Probing
From Report Date: 01-Feb-2021
Report Date: 11-Feb-2021
Complete by (Milestone): 11-Feb-2021

Description of Group Project Milestone

Project Milestone Name : Probing								
Requirement Description	Task ID	Task Description	Pass/Fail/Not Executed	Test Date	Responsible Primary Team Member	Responsible Secondary Team Member	Comment	Additional Comment
The user shall be able to add a unique customer record to the database	01	Add customer to database	Pass	8-Feb-2021	Jeffry Munoz	Jana Grunewald	Successful addition of new customer	
The user shall be able to search for a customer stored in the database using their customer ID.	02	Search for customer with valid ID	Pass	8-Feb-2021	Breanna Rhodes	Tamara Vergara	Correct customer was displayed	
The user shall be able to search for a customer stored in the database using their customer ID	03	Search for customer with invalid	Pass	8-Feb-2021	Tamara Vergara	Jana Grunewald	Error Message displays stating “Record not found”	
The user shall be able to update their profile	04	Update Customer information	Fail	9-Feb-2021	Jana Grunewald	Jeffry Munoz	Information was not updated	No error message was displayed
The user shall be able to add a unique flight	05	Add a new flight to database	Pass	9-Feb-2021	Jeffry Munoz	Jana Grunewald	Flight Successfully added	

record to the database								
The user shall be able to log into the application using their username and password	06	Use stored credentials to log into the application	Fail	9-Feb-2021	Jana Grunewald	Breanna Rhodes	No login page found	

Group Project Milestone Screenshot



Rebbica007 Update README.md 24811e1 15 seconds ago 6 commits

.idea	Running code and database.	yesterday
Airline-SWT	Running code and database.	yesterday
.gitignore	Running code and database.	yesterday
Airline-SWT.iml	Running code and database.	yesterday
README.md	Update README.md	14 seconds ago

README.md

SWT-AirlineReservationSystem

Change History

- 17 January 2021 - Initial Commit
- 08 February 2021 - Updated code. Runs correctly. Breanna Rhodes got the initial to run and Jeffry Munoz got the database to run. Tamara Vergara helped line out the requirements along with Jana Grunewald. Jana, Jeffry, Breanna and Tamara worked together to put this project in the works.
- 09 February 2021 - Breanna Rhodes and Tamara Vergara completed the Software Inspection and Jana Grunewald and Jeffry Munoz completed the Technical Review. Jana, Jeffry, Breanna and Tamara worked together to get the requirements written out.

About

Updated code. Runs correctly. Breanna Rhodes got the initial to run and Jeffry Munoz got the database to run. Tamara Vergara helped line out the requirements along with Jana Grunewald. Jana, Jeffry, Breanna and Tamara worked together to put this project in the works.

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Java 100.0%

Group Project Meeting Minutes

Meeting Date: 2/01/21
Team Members present:
Jana Grunewald
Jeffry Munoz
Breanna Rhodes
Tamara Vergara
Summary: During this meeting tasks were assigned for the completion of the first deliverable.

Task	Due Date	Primary Team Member	Secondary Team Member
Set up tools and environment to run program	5 pm - 2/05/21	Breanna Rhodes	Jeffry Munoz
Technical Review of current product	5 pm - 2/06/21	Jeffry Munoz	Jana Grunewald

Software Inspection of current code	5 pm - 2/07/21	Breanna Rhodes	Tamara Vergara
Defining Requirements	5 pm - 2/07/21	Tamara Vergara	Jana Grunewald
Categorizing Requirements	5 pm - 2/07/21	Tamara Vergara	Jana Grunewald

Meeting Date: 2/08/21
Team Members present:
Jana Grunewald
Jeffry Munoz
Breanna Rhodes
Tamara Vergara
Summary: During this meeting we discussed the requirements and the project progress report. We also discussed what test should be run on the existing software and who should do so. It was decided that the functional requirements would be tested first.

Task	Due Date	Primary Team Member	Secondary Team Member
Test requirement FUN-1	5 pm - 2/08/21	Jeffry Munoz	Jana Grunewald
Test requirement FUN-2	5 pm - 2/09/21	Breanna Rhodes	Tamara Vergara
Test requirement FUN-3	5 pm - 2/08/21	Breanna Rhodes	Tamara Vergara
Test requirement FUN-4	5 pm - 2/09/21	Tamara Vergara	Jana Grunewald
Test requirement FUN-5	5 pm - 2/10/21	Jeffry Munoz	Jana Grunewald
Test requirement FUN-6	5 pm - 2/10/21	Breanna Rhodes	Jana Grunewald
Test requirement FUN-7	5 pm - 2/10/21	Jana Grunewald	Jeffry Munoz

Technical Review

Main.java

- Use of older standard (Java Swing) may limit the applications longevity when compared to the use of the newer standard (JavaFX)
    - Also uses old JDBC driver class (com.mysql.jdbc.Driver) when there is an updated version available (com.mysql.cj.jdbc.Driver)
- `private void initComponents() {}` Offers a way to access the other modules but no way to close them
    - Multiple instances of the same module such as addCustomer can be opened at once without the ability to use the provided dropdown menu to close them
- Class variables such as `private javax.swing.JMenu jMenu1` can be local variables inside of the `initComponents()` method

Login.java

- Offers another way to log in but no way to log out
- Once logged in there is no indication of what user is logged in
- Repeated `public static void main(String args[])` class could be condensed
- Class variables such as `private javax.swing.JButton jButton1;` can be local variables inside of the `initComponents()` method
- Could create another class or method for database connectivity instead of repeating the same lines of code in every class EX:
    - `Class.forName("com.mysql.jdbc.Driver");`
    - `con = DriverManager.getConnection("jdbc:mysql://localhost/airline","root","");`

ticketreport.java

- Repeated database connectivity lines
- Repeated catch blocks
    - `catch (ClassNotFoundException ex) {`
    - `Logger.getLogger(ticket.class.getName()).log(Level.SEVERE, null, ex);}`
    - `catch (SQLException ex) {`
    - `Logger.getLogger(ticket.class.getName()).log(Level.SEVERE, null, ex);}`
- Class variables such as `private javax.swing.JButton jButton1;` can be local variables inside of the `initComponents()` method
    - Repeated variable names from Login.java

userCreation.java

- Repeated database connectivity lines
- Repeated catch blocks just like in ticketreport.java
- Class variables such as `private javax.swing.JButton jButton1;` can be local variables inside of the `initComponents()` method
    - Repeated variable names from Login.java and ticketreport.java

airline.sql

- SYSTEM_USER = ADMIN; caused an error. This was already accomplished when setting up the MySQL server, so it was commented out.
- CS003, John Peter's photo was too large for the database. Had to extend the packet to 32M.

Software Inspection

The software was inspected using the Walk-Through method.

The program will first be accessed by running the Main class in order to create an account.There are 4 tabs at the top, I clicked the User tab because I need to create a new User account.

- Selected 'UserCreation'
- A window popped up to create a new account that is to be stored into the database in the User table.
- 'FirstName', 'LastName' and 'User Name' all have inappropriate spacing.
    - 'FirstName' should be corrected to 'First Name'
    - 'LastName' should be corrected to 'Last Name'
    - 'User Name' should be corrected to 'Username'
- I entered my information into the fields and clicked the 'Add' button.
    - A new window popped up that said 'User Createdd…..'
        - 'Created' is spelled incorrectly.
    - After I close out of the 'User Createdd….' window I am still in the User Creation window with the fields still containing my information.
        - The fields should be cleared after generating a new User to refrain from the user clicking the 'Add' button again and creating another duplicated account or causing an error.
        - The User Creation window should also close after a User is created, and the program should be running under the current user's account that was just created.
        - In order to leave the User Creation window, the user would have to hit a 'Cancel' button which can be confusing for the user because they do not want to cancel the account they just made.
    - After I created my User account, I checked the database and my new account was there under the User table.

I now want to assure that my Username and Password are accessed properly from the database when logging in. In order for me to check this, I ran the Login class.

- The login information was retrieved correctly and I was successful in logging into the program using my new User account that I had just created.
- The first tab that I opened was 'Tickets' then I selected 'Ticket Report'.
    - This window is supposed to display currently purchased tickets in the database.
    - The 'Date' column is cut off because the size of the window is too small.
    - The window is not adjustable in size and therefore cannot be expanded to view the entire date of the flight.
    - There is an arrow box in the top left corner of the box that looks like it should display options for the window, however no options are available, making this arrow box useless.
- I opened another tab 'Flights' and selected 'Add Flight'.
    - The window opened directly over top the "Ticket Report' window.
    - You can have multiple windows open within this program at the same time.
    - The flight ID is populated for me as 'FO004' which would be the next incremented ID number after the most recent entry ('FO003').
    - I entered:
        - 'My Flight' for *Flight Name*

  - ‘Uk’ for *Source*
  - ‘India’ for *Departure*
  - ‘Feb 23, 2021 for *Date*
  - ‘11:00’ for *Dep Time*
  - ‘8’ for *Arr Time*
    - Air is spelled incorrectly.
  - ‘760’ for *Flight Charge*
- Which led to a window that popped up with the message ‘Flight Createdd…’
  - Created is spelled incorrectly again.
  - I closed out of this window.
- I am still in the ‘Add Flight’ window and all the fields still contain the information that I added.
  - These fields should all be cleared after the ‘Add’ button or the ‘Cancel’ button are clicked.
  - I checked the database under the ‘flight’ table and my newly added flight has been added correctly.
- I logged back into the program now to book a plane ticket.
  - First I open the ‘Ticket Report’ window so that I can see if the program dynamically populates new tickets as they are created or if the system needs to be closed out and reopened.
  - Next, I open the ‘Book Ticket’ window and begin booking my ticket.
  - The ticket number has been auto generated with the correct number of ‘TO004’.
  - I need to search for the customer account that I previously created to populate the ‘FirstName’, ‘LastName’, and ‘Passportno’ fields.
    - Inappropriate spacing for the ‘FirstName’ and ‘LastName fields.
    - ‘Passportno’ should contain a less condensed field name. It could say ‘Passport #’ or ‘Passport Num’.
  - The fields were populated with the correct information.
  - I searched for the flight that I previously created by entering the *Source* and the *Departure*.
    - The flight was successfully found from the flights database table and populated in the table view.
      - The ‘Date’ column is cut off.
  - I selected the flight from the table view and the ‘Flight no’, ‘Flight Name’, ‘Depart Time’ fields all populated correctly.
    - ‘Flight no’ should be specified better. Maybe change to ‘Flight Num’ or ‘Flight #’ for better specification.
    - The ‘price’ field can be changed manually even though there is already a price assigned to that flight from the database.
    - The ‘seats’ field is incremented by pressing up and down arrows, depending on the desired number of seats.
      - The upper arrow is grayed out as if it isn’t an option to be pressed; however, the button can be pressed and increments the number of seats correctly.

- I incremented the number of seats to 2, and it displayed the price of two tickets at the price pulled from the database.
- I changed the price from ‘760’ to ‘800’ and the total price for the two seats remained the same.
    - The total price should change when the ticket price is changed manually from the user.
- After all the fields were completed, I clicked the ‘Book’ button.
    - A window popped up with the message ‘Ticket Bookeed….’
        - ‘Booked’ is spelled incorrectly.
    - I selected ‘Ok’ in the window, and the window closed the message.
    - All the fields still contained the information that I entered for booking my new ticket.
        - These fields should be cleared to keep the user from double booking their ticket.
- I went back to the ‘Ticket Report’ window and the newly booked ticket did not populate while the window was open.
    - To make the software exist in live time, the ‘Ticket Report’ window should update after a ticket is booked without having to reopen the window again.
- The flight I booked changed the ticket price correctly to the one I entered manually.
    - In the database under the ‘ticket’ table, the price of the flight was changed to the one entered manually.
    - In the database under the ‘flight’ table the price of the flight that I created is still the initial price and did not change when I entered the price manually for the ticket.
- I closed out the ‘Ticket Report’ window and then reopened it again and the newly added ticket was there.
    - The table view in the ‘Ticket Report’ window is populating when the window opens rather than when a ticket is booked.
    - I selected cancel in the ‘Ticket Report’ window and it closed.

I want to see if the ‘Ticket Report’ information is password protected, so I run the Main class to check this.

- The ‘Ticket Report’ window contains all of the ticket information without having an account with the program.
- I can also search the customer database without having an account and get the customers information and photo.
- I can also create flights.

These features within this program should be password protected and require you to have the correct account. There should be two types of accounts, an admin account and a customer account. An admin account should be able to have the capability to create flights and customers, as well as have access to the ‘Ticket Report’ window. The customer account should only be able to book a ticket for the existing flights, and not be able to create their own flight. Finally the ‘User Creation’ tab should be moved to the login screen where the user can create a new account if they do not already have one. This way the Main driver class can just call the Login class before proceeding any further in the program.

Functional Requirements

ID	Requirement	Testable?
FUN-1	The application shall allow users to add a unique customer record to the database	Y
FUN-2	The application shall allow users to add a unique user record to the database	Y
FUN-3	The application shall allow users to search for a customer stored in the database using their customer ID	Y
FUN-4	The application shall allow users to view a table of all tickets	Y
FUN-5	The application shall allow users to add a unique flight record to the database	Y
FUN-6	The application shall allow users to log into the application using their unique username and password (login credentials?)	Y
FUN-7	The application shall allow users to update their profile	Y
FUN-8	The application shall allow users to search for a ticket using customer ID	Y
FUN-9	The application shall allow users to book a ticket based on departure time	Y
FUN-10	The application shall allow users to book a ticket based on departure location	Y
FUN-11	The application shall allow users to book a ticket based on class	Y
FUN-12	The application shall allow users to book a ticket based on price	Y
FUN-13	The application shall allow users to book a ticket based on amount of seats	Y
FUN-14	The application shall create unique IDs for new customers	Y
FUN-15	The application shall create unique IDs for new users	Y
FUN-16	The application shall create unique IDs for new flights	Y
FUN-17	The application shall create unique IDs for new tickets	Y
FUN-18	The application shall display the information of available flights meeting the user's criteria	Y
FUN-19	The application shall allow users to upload images to their profile	Y

Non-functional Requirements

ID	Requirement	Testable?
NONFUN-1	The application shall store information in a MySQL database	Y
NONFUN-2	The application shall run on Java 15	Y
NONFUN-3	The application's GUI shall be implemented using Java Swing	Y
NONFUN-4	The application shall prevent SQL injection using prepared statements	Y
NONFUN-5	The application's GUI shall be written in English	Y
NONFUN-6	The application shall run on Windows 10 machines	Y
NONFUN-7	The application shall run on macOS machines	Y
NONFUN-8	The application shall load screens in less than 6 seconds	Y
NONFUN-9	The application shall be less than 1 MB in size	Y
NONFUN-10	The application shall take images up to 32MB	Y
NONFUN-11	The application shall use no more than 4GB of memory	Y