MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

OBJECT ORIENTED PROGRAMMING (CSE –2143) MINI PROJECT REPORT ON

BANKING MANAGEMENT SYSTEM

SUBMITTED TO

Department of Computer Science & Engineering

by

Aditya Sinha, Roll No : , Redg no. :

Ameen Shaik, Roll No : , Redg no. : 220962320

Moulik Singh, Roll No : , Redg no. : 210962130

Sathvik, Roll No : , Redg no. : 220962404

(AI&ML) BatchA2

Name & Signature of Evaluator 1

Name & Signature of Evaluator 2

(July 2023 – Nov 2023)

ABSTRACT:

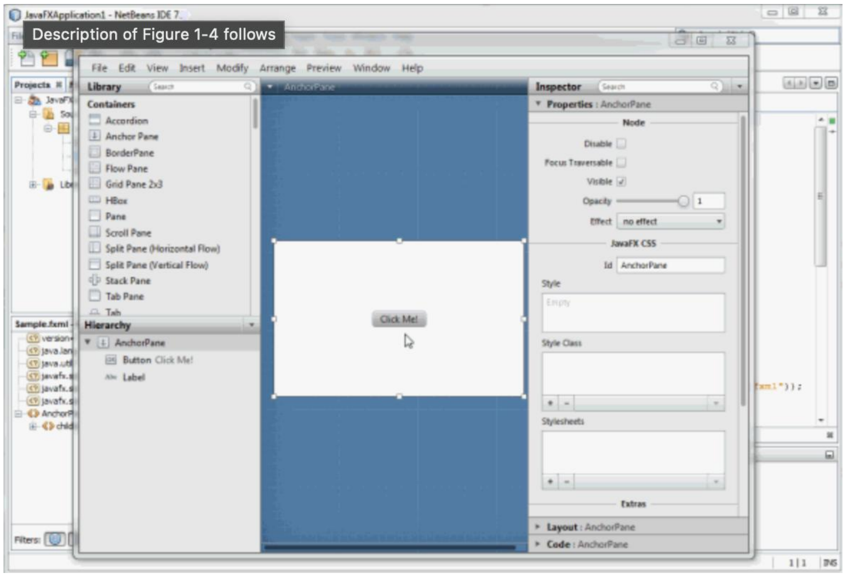
The JavaFX Banking Server Application is a secure and user-friendly banking software, employing JavaFX for its modern and intuitive user interface. It caters to both bank employees and customers, offering features like user authentication, account management, transaction history, customer support, and administrative controls. Real-time notifications, data security, and data analysis tools enhance the user experience and help banks provide personalized financial solutions. This project blends robust security with a responsive design, accessible through web and desktop interfaces, to improve banking services and customer satisfaction.

STATEMENT ABOUT THE PROBLEM

Traditional banking systems often lack the modern features and user-friendly interfaces required to meet the evolving needs of customers. This disconnect results in inefficiencies, security concerns, and reduced customer satisfaction. The need for a comprehensive, secure, and intuitive banking system is apparent, prompting the development of the JavaFX Banking Server Application.

WHY IS THE PARTICULAR TOPIC CHOSEN?

The choice of the "JavaFX Banking Server Application" project is motivated by several key factors:



1. **Technology Advancement**: The banking industry is evolving rapidly with advancements in technology. The adoption of modern technologies, such as JavaFX, is essential to stay competitive and meet the growing demands of tech-savvy customers.

2. **User Experience**: Customer satisfaction is paramount in the banking sector. By developing a user-friendly application with a modern interface, banks can enhance the overall user experience, making it easier for customers to perform transactions and access their financial information.

3. **Security Concerns**: Security is a top priority in the financial industry. This project addresses security concerns by implementing robust authentication methods, encryption protocols, and secure communication channels to protect sensitive customer data.

4. **Efficiency and Convenience**: The project aims to streamline banking processes, making it easier for customers to manage their accounts, access transaction history, and receive real-time notifications. This not only improves the customer experience but also enhances operational efficiency for bank employees.

5. **Data-Driven Decision Making**: Data analysis tools within the application enable banks to gain insights from customer data, allowing them to make informed decisions, offer personalized financial solutions, and enhance their services.

6. **Competitive Edge**: The adoption of innovative technologies like JavaFX gives banks a competitive edge in a crowded marketplace. Staying ahead in the digital age is crucial for attracting and retaining customers.

Overall, the project is chosen to address the changing landscape of banking, where technology plays a pivotal role in meeting customer expectations, ensuring security, and staying competitive in the financial services industry.

SCOPE:

The project scope includes developing a JavaFX-based banking application with features such as secure user authentication, account management, transaction history, customer support, administrative controls, real-time notifications, and data analysis tools. The application will offer a modern, user-friendly interface accessible through web and desktop platforms, with a focus on security, scalability, and compliance with financial regulations. Training, documentation, and ongoing support for bank employees will also be provided.

OBJECTIVE OF THE PROJECT:

The objectives of the "JavaFX Banking Server Application" project are as follows:

1. **Enhance User Experience**: Create a user-friendly and modern interface using JavaFX to improve the overall user experience for both customers and bank employees.

2. **Ensure Security**: Implement robust security measures, including multi-factor authentication and data encryption, to safeguard customer data and transactions.

3. **Streamline Banking Operations**: Provide efficient account management and transaction capabilities, allowing customers to open accounts, view balances, perform transactions, and access their transaction history.

4. **Real-Time Communication**: Enable real-time notifications to keep customers informed about their account activities and updates.

5. **Data Analysis**: Develop tools for data analysis, empowering the bank to make data-driven decisions, offer personalized financial solutions, and enhance services.

6. **Administrative Controls**: Offer administrative tools for bank employees to manage customer accounts, verify transactions, and monitor system performance.

7. **Compliance and Documentation**: Ensure compliance with relevant financial regulations and provide comprehensive documentation for users and administrators.

8. **Scalability and Extensibility**: Design the application to be scalable and extensible to accommodate future updates and additional features.

9. **Training and Support**: Provide training for bank employees and ongoing technical support to ensure the smooth operation of the system.

10. **Improved Customer Service**: Ultimately, the project aims to improve customer service, efficiency, and data security in the banking sector, giving the bank a competitive edge in a digitally evolving landscape.

EXISTING METHOD

Existing methods in banking include core banking software, online and mobile banking, ATMs, customer support, transaction history, security measures, administrative tools, data analysis, and regulatory compliance. These methods aim to provide secure and efficient financial services while adhering to industry regulations.

DISADVANTAGES

Disadvantages of existing banking methods include:

1. **Limited Accessibility**: Traditional banking methods may require physical presence at a branch, limiting accessibility for remote or busy customers.

2. **Security Concerns**: Security breaches, fraud, and identity theft are ongoing concerns, particularly with online and mobile banking.

3. **Technology Gaps**: Some customers may struggle to adapt to digital banking methods, creating a digital divide.

4. **Operational Costs**: Maintaining physical branches, ATMs, and call centers can be costly for banks, potentially leading to higher fees for customers.

5. **Transaction Delays**: Traditional transaction processing can be slower compared to digital methods, impacting the speed of financial transactions.

6. **Limited Personalization**: Traditional banking services may offer limited personalization compared to digital platforms.

7. **Maintenance and Upkeep**: Physical banking infrastructure requires ongoing maintenance and technology updates.

8. **Regulatory Compliance Challenges**: Adhering to complex financial regulations can be challenging for banks.

9. **Human Error**: In traditional banking, human error can occur in data entry, document processing, and customer interactions.

10. **Limited Self-Service**: Traditional methods may provide fewer self-service options, requiring customers to visit branches or call for assistance.

These disadvantages highlight the need for innovative solutions like the "JavaFX Banking Server Application" project to address accessibility, security, operational efficiency, and customer satisfaction in the banking industry.

PROPOSED SYSTEM:

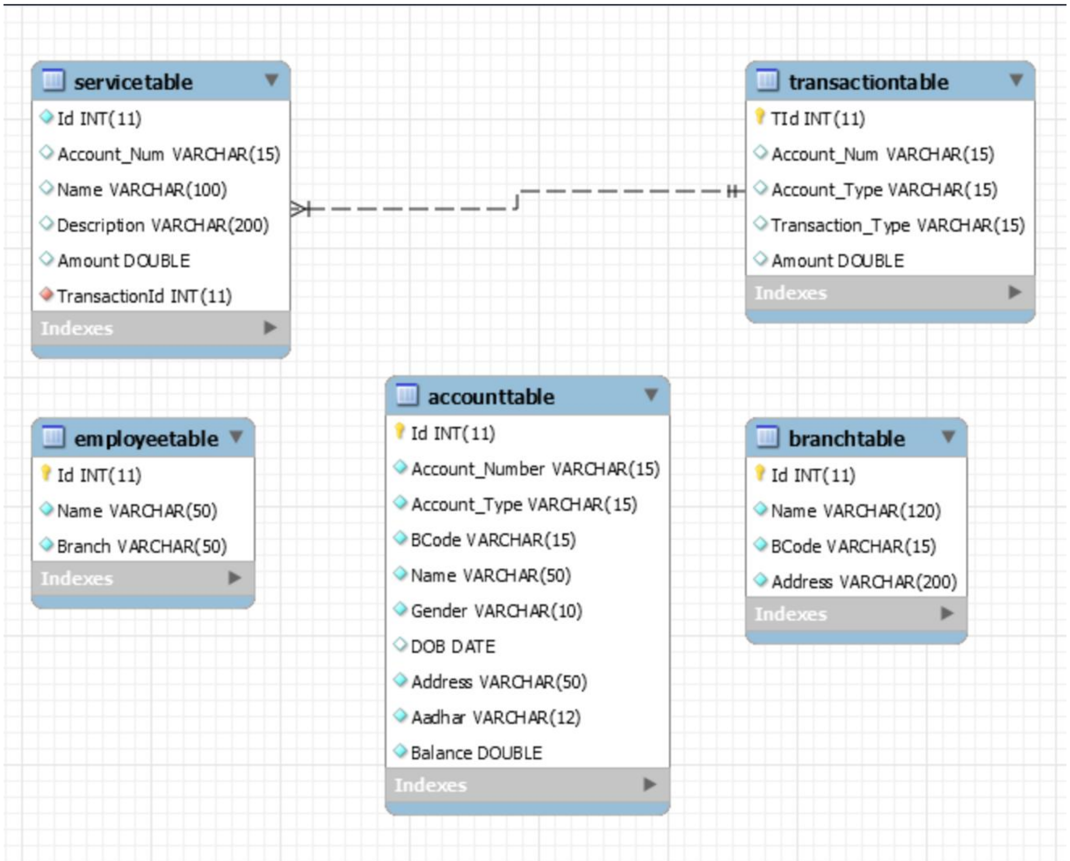
The proposed JavaFX-based banking system seeks to enhance user experience, security, and operational efficiency. It will offer a modern and visually appealing user interface, robust security features, streamlined operations, data analysis tools, administrative controls, and scalability. This system aims to improve customer satisfaction and compliance with industry

regulations.

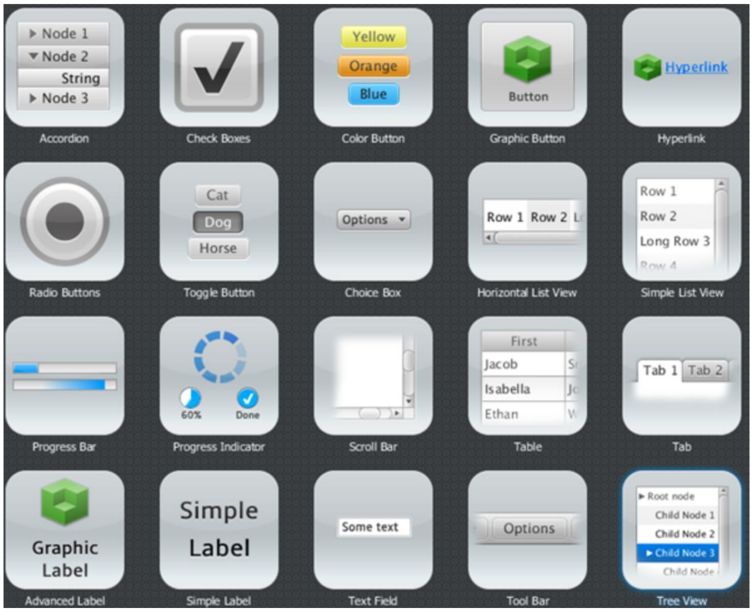
ADVANTAGES:

1. User-Friendly Interface.
2. Robust Security.
3. Efficient Operations.
4. Data-Driven Decisions.
5. Administrative Tools.
6. Accessibility.
7. Documentation and Training.
8. Scalability.
9. Compliance.
10. Competitive Edge.

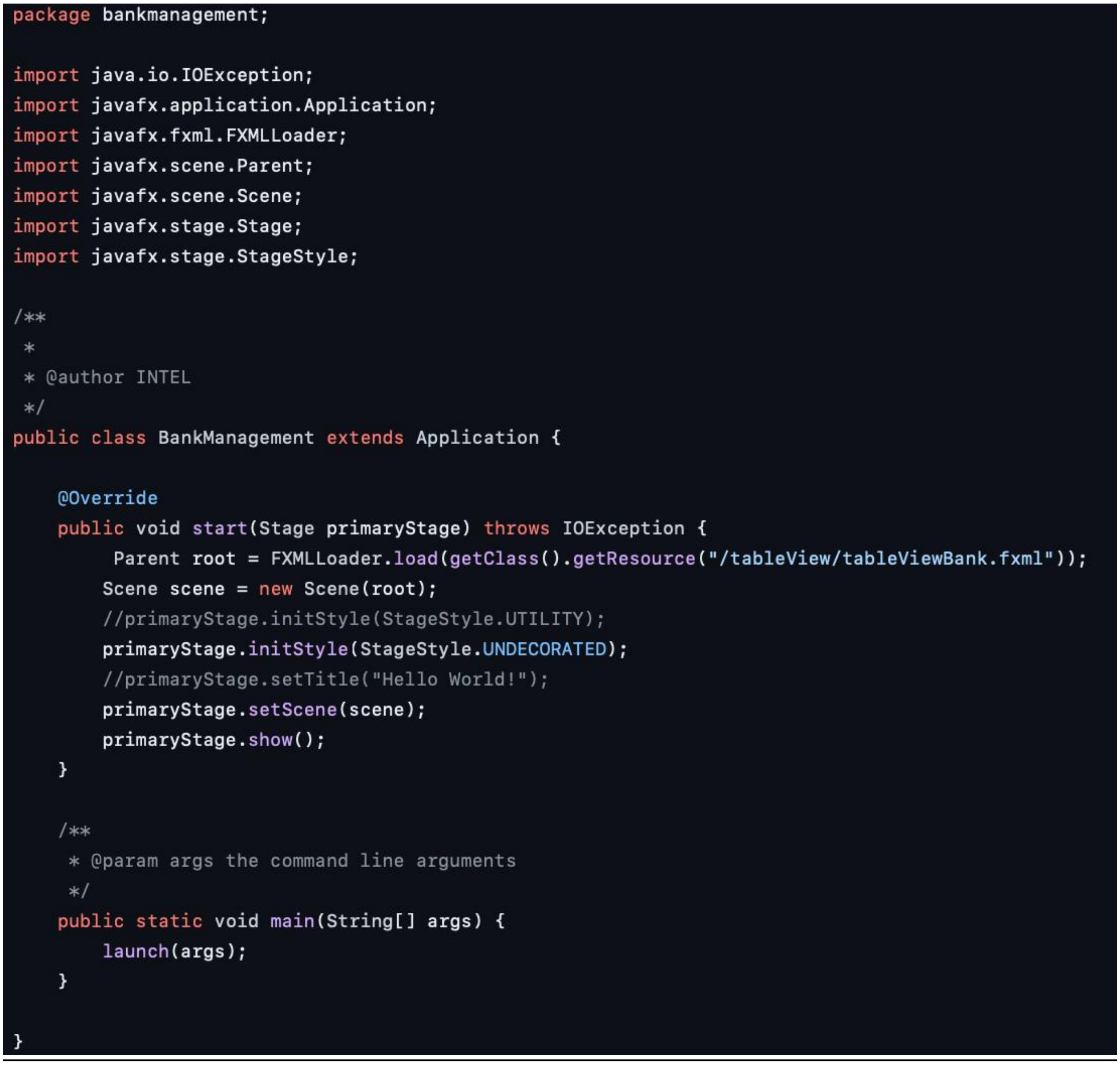
PROJECT FLOW:



JavaFX control buttons:

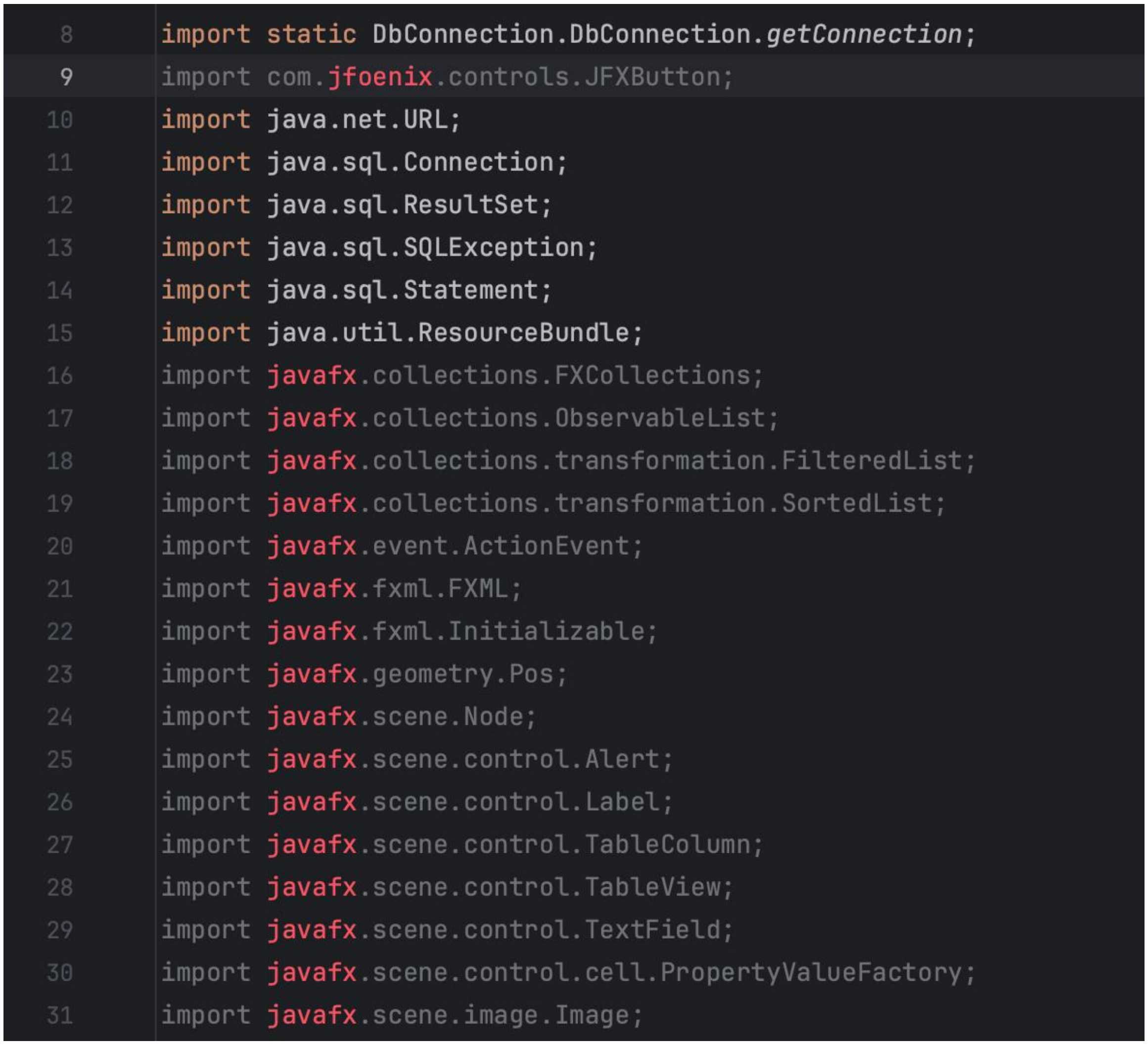


JavaFX:



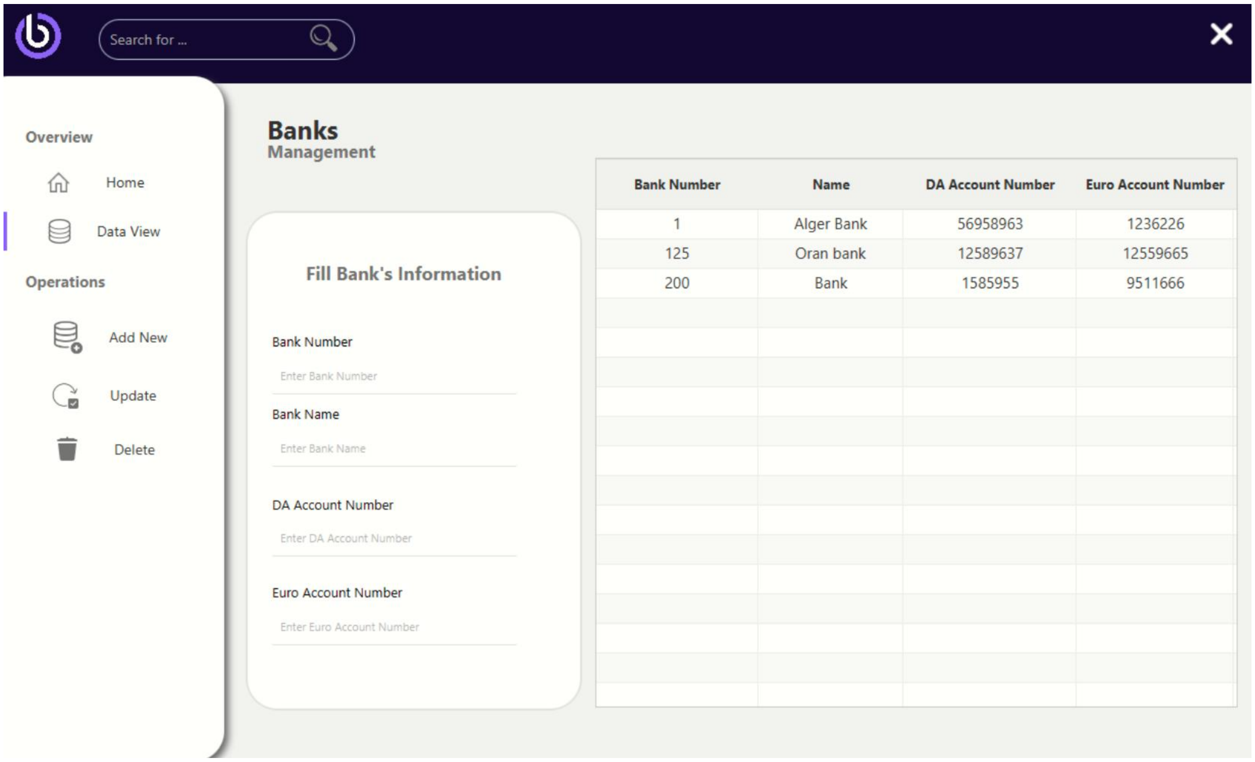
```java
package bankmanagement;

import java.io.IOException;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.stage.StageStyle;

/**
 *
 * @author INTEL
 */
public class BankManagement extends Application {

    @Override
    public void start(Stage primaryStage) throws IOException {
         Parent root = FXMLLoader.load(getClass().getResource("/tableView/tableViewBank.fxml"));
        Scene scene = new Scene(root);
        //primaryStage.initStyle(StageStyle.UTILITY);
        primaryStage.initStyle(StageStyle.UNDECORATED);
        //primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }

}
```

JavaFX provides powerful layout management capabilities, such as GridPane, HBox, VBox, and StackPane, which enable precise control over the arrangement of UI components. In the online shopping simulator project, these layouts are used to design product listings, shopping carts, checkout forms, and other essential elements of the application.



```java
import static DbConnection.DbConnection.getConnection;
import com.jfoenix.controls.JFXButton;
import java.net.URL;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.collections.transformation.FilteredList;
import javafx.collections.transformation.SortedList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.geometry.Pos;
import javafx.scene.Node;
import javafx.scene.control.Alert;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.image.Image;
```

USER INTERFACE:



Banks
Management

Fill Bank's Information

Bank Number

Enter Bank Number

Bank Name

Enter Bank Name

DA Account Number

Enter DA Account Number

Euro Account Number

Enter Euro Account Number

Bank Number	Name	DA Account Number	Euro Account Number
1	Alger Bank	56958963	1236226
125	Oran bank	12589637	12559665
200	Bank	1585955	9511666

SOFTWARE FRONT END REQUIREMENTS

H/W CONFIGURATION:

Processor	- I3/Intel Processor
Hard Disk	- 160GB
Key Board	- Standard Windows Keyboard
Mouse	- Two or Three Button Mouse
Monitor	- SVGA
RAM	- 8GB

S/W CONFIGURATION:

•	Operating System	: Windows 7/8/10
•	Server Side Script	: REACT JS
•	Programming Language	: JAVA
•	IDE/Workbench	: Intellij Idea
•	Technology	: Java 1.7
	Server Deployment	: Tomcat Server

REFERENCES:

[1] Graphical User Interfaces JavaFX GUI Basics, Event Programming and GUI UI Controls, CSE260, Computer Science B: Honors Stony Brook University.

[2] support.smartbear.com; Testing JavaFX Applications – Overview; support.smartbear.com/testcomplete/docs/app- testing/desktop /java-fx/about.html

[3] Java All-in-One For Dummies, Doug Lowe.

-