

**OBJECT-ORIENTED PROGRAMMING**

**CEP (PROJECT REPORT)**

**“VIRTUAL BANKING SYSTEM”**

* **INSTRUCTOR NAME**:

1. DR. BURHAN KHAN
2. Engr. MUHAMMAD MISBAH

* **GROUP MEMBERS:**

1. SHAYAN HASHMI (22K - 4865)
2. SYED AMMAR ZULFIQAR (22K - 4845)

* **SECTION:** BEE-2A



* **TABLE OF CONTENTS:**

|  |  |
| --- | --- |
| **CONTENTS**: | **Page Number:** |
| 1. Introduction | 3 |
| 1. Literature Review | 4 |
| 1. Methodology | 5 |
| 1. Flowchart Diagram | 6 |
| 1. Code Explained Properly | 7 |
| 1. Results | 8 |
| 1. SDGs | 9 |
| 1. Conclusion | 10 |
| 1. References | 11 |

1. **INTRODUCTION:**

* A virtual banking system is referred to as a digital platform that enables users to conduct financial transactions and gain access to financial services over the Internet. It is a creative approach that seeks to increase consumer comfort, speed, and security while simultaneously enhancing the effectiveness and lowering the cost of banking operations.
* The virtual banking system project in Java is an innovative solution that provides a seamless banking experience for customers while improving the efficiency and cost-effectiveness of banking operations.
* Some of the primary features of a virtual banking system include online account management, money transfers, chequebook, and loan applications. The system should also provide robust security measures to protect customer information and financial transactions from unauthorized access.

1. **LITERATURE REVIEW:**

The literature review has shown that there is a significant amount of research on virtual banking systems in Java. The studies have presented different approaches to the design and development of virtual banking systems using Java and other related technologies such as JSP, Servlets, and MySQL. The studies have also highlighted the importance of security features in virtual banking systems. The literature review provides a valuable resource for anyone interested in designing and developing a virtual banking system in Java.

Java-based Virtual Banking System by Mayank and Arpit Agarwal. The authors of this research article go over the planning and execution of a Java-based virtual banking system. They give a broad overview of the user interface, database structure, and system architecture.

Java and MySQL-Based Virtual Banking System by Riddhi Vora, Priyanka R. Kudale, and Pooja S. Wagh. The design and construction of a virtual banking system using Java and MySQL are discussed in this paper. The system's requirements, design, and implementation are covered by the authors. The system design, graphical user interface, and security aspects are also covered in detail.

Virtual Banking System with Secure Authentication using Java by J. C. Rayudu and V. R. Venkatesh. This research paper presents the design and development of a virtual banking system with secure authentication using Java. The authors discuss the system's security majors such as biometric authentication and one-time passwords.

Virtual Banking System with Fraud Detection in Java by S. K. Sharma and R. K. Sharma. This study presents the design and development of a virtual banking system with fraud detection in Java. The authors discuss the system requirements, design, and implementation. They also provide details on the system architecture, user interface, and security features, such as transaction monitoring and anomaly detection.

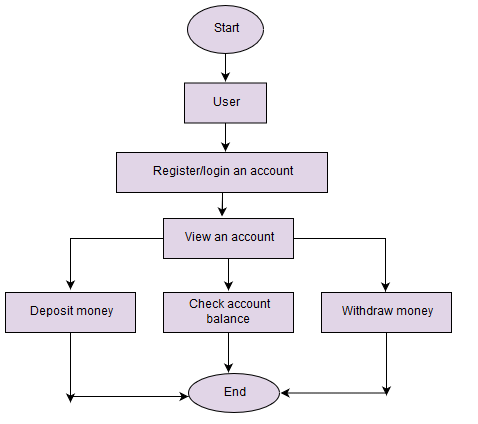
1. **METHODOLOGY**:

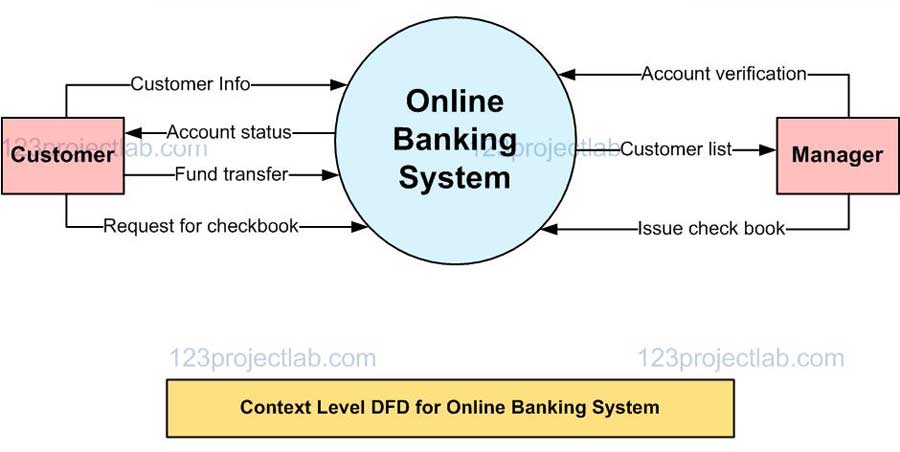
The methodology of developing a virtual banking system in Object-Oriented Programming (OOP) using Java involves the following steps:

* **Requirement Analysis:** The first step is to gather and analyze the requirements for the virtual banking system. This includes identifying the features and functionalities required by the system, such as account management, transaction processing, and security measures.
* **Object-Oriented Design:** The next step is to design the virtual banking system using OOP principles. This involves identifying the objects and their relationships that will make up the system. The system can be divided into various modules such as the user interface, account management, transaction processing, and security.
* **Implementation:** Once the design is finalized, the system can be implemented using Java programming language. This involves writing the code for the different classes and methods that make up the system. For example, classes for user accounts, transactions, and authentication can be created.
* **GUI:** Graphical User Interfaces (GUIs) are commonly used in bank management systems to provide an intuitive and user-friendly interface for bank employees and customers. It might include account information, transaction history, login pages, and other features. GUIs can help to simplify complex banking processes and improve the overall efficiency of bank operations.
* **Testing:** After implementation, the virtual banking system should be tested to ensure that it meets the requirements and functions correctly. This includes unit testing to ensure that each module and component works as expected and interacts correctly with other modules.

Overall, the methodology of developing a virtual banking system in OOP Java involves a structured approach that emphasizes requirements gathering, design, implementation, graphical user interfaces (GUIs), and testing. This helps to ensure that the final product is of high quality and meets customer needs.

1. **FLOW-CHART DIAGRAM:**





1. **CODE EXPLAINED PROPERLY:**

* The virtual banking system is built using Java with a graphical user interface (GUI). It allows customers to perform basic banking operations such as deposits, withdrawals, chequebook requests, and loan applications.
* First, we would need to create a set of graphical components such as buttons, text fields, and dialog boxes that allow the user to interact with the system. To do this we use Swing GUI form in Apache NetBeans. When the program is launched, the main window of the application appears with buttons to log in as an admin or user or you have to create an account for the user. To create a button that allows the user to deposit money, withdrawal, or any other operation we use buttons and labels from the right side of the GUI tool palette in NetBeans.
* After the program is launched the main window of the application appears with buttons for admin and user, then if a customer taps on the user button then various banking operations will appear on the screen such as deposit, withdrawal, request for chequebook, user’s current account balance and request for loan application.
* When the user clicks on a button, a new window or dialog box appears, prompting the user to enter the necessary information for the operation. For example, to deposit money, the user clicks on the "Deposit" button. The program launches a new dialog box that prompts the user to enter the amount to deposit and the account number. The user enters the information and clicks the "Deposit" button. The program then processes the deposit by adding the deposited amount to the account balance. The program stores customer data such as account numbers, names, and account balances in a text file. When the user performs a banking operation, the program reads and writes to this file to update the account balances and transaction history.
* To implement the loan application feature, we create a separate button or menu option that allows the user to access the loan application form. When the user clicks on this button, the program would launch a new window or dialog box, that prompts the user to enter their personal and financial information, as well as the details of the loan they are applying for. Once the user has filled out the form, they would submit it to the bank, and the system would process the application and provide a response and a confirmatory slip as well.
* For security majors, the system requires the user to enter their login credentials to access their account. This ensures that only authorized users can access their accounts. The system uses access control techniques to prevent unauthorized access to sensitive areas of the system. The system restricts access to certain features and data to only authorized users, based on their roles and permissions. However, the admin of the bank can see all money transaction histories of users, loan applications, and chequebook requests of users.

1. **RESULTS:**

* Increased accessibility: A virtual banking system can increase accessibility for customers by allowing them to perform banking transactions from anywhere at any time.
* Enhanced security: Virtual banking systems can implement advanced security features to protect customer data and prevent fraudulent activities.
* Improved efficiency: Virtual banking systems can streamline banking operations and reduce the time required to complete transactions, resulting in improved efficiency.
* Increased customer satisfaction: By providing customers with a more convenient and secure banking experience, virtual banking systems can enhance customer satisfaction.
* Cost savings: Virtual banking systems can reduce operational costs by automating processes, reducing paperwork, and minimizing the need for physical branches.

1. **SUSTAINABLE DEVELOPMENT GOALS (SDGs):**

* **Quality Education:** Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.
* **Decent Work and Economic Growth:** Promote sustained, inclusive, and sustainable economic growth, full and productive employment, and decent work for all.

1. **CONCLUSION:**

In conclusion, a virtual banking system project in Java can provide numerous benefits to both customers and banks. With the increasing popularity of online and mobile banking, virtual banking systems can offer customers a more convenient and secure banking experience while reducing operational costs for banks. The project can help to increase accessibility by allowing customers to perform banking transactions from anywhere at any time. The virtual banking system can be designed with advanced security features such as encryption, biometric authentication, and one-time passwords to protect customer data and prevent fraudulent activities. By automating processes, reducing paperwork, and minimizing the need for physical branches, virtual banking systems can improve efficiency and reduce operational costs for banks. This can result in cost savings for both customers and banks. Overall, a virtual banking system project in Java can be a valuable investment for banks and can provide customers with a more accessible, secure, efficient, and satisfying banking experience.

1. **REFERENCES:**

* https://www.youtube.com/watch?v=dUNNXVhuGJQ (Dispose and set visible)
* https://www.youtube.com/watch?v=jC7QctPRmls (trim)
* https://www.youtube.com/watch?v=wZQc1Nct0-s (empty)
* https://www.youtube.com/watch?v=lke3lsLHr5k (Labels, radio buttons, text field, panels, text area)
* https://www.youtube.com/watch?v=3RSXHVyzdmg(de limiter)
* https://www.youtube.com/watch?v=WTOc8VKjBvQ (Has. Next)
* https://www.youtube.com/watch?v=vuz48JCQK5c (Split string)
* https://www.youtube.com/watch?v=Vy2l3lGAb2I (File handling in Java)
* https://www.youtube.com/watch?v=1XAfapkBQjk (Exception in Java)
* https://www.youtube.com/watch?v=mwgSpPyUCTc (File reader and writer in Java)
* https://www.youtube.com/watch?v=N3H60u4f9VI (buffer reader)