DATE:02/05/2024

Bank management system

ATM simulator system

ANUM RAMZAN(SP21318)
M DANIYAL MUGHAL(SP21300)
SYED SAJID ALI SHAH(SP21284)
KHATIJA HUSNAIN(SP21311)

SUBMITTED TO:Ms. FATIMA GILLANI

DEPARTMENT OF SOFTWARE ENGINNERING
NATIONAL UNIVERSITY OF MODERN LANGUAGES

ABSTRACT

The online banking system being developed here allows users to conveniently and securely access their accounts. A distributed architecture was used in the system's design, and Java was used to create the user interfaces for simplicity of maintenance and cross-platform compatibility.

User registration, money deposits and withdrawals, account balance checks, PIN changes, transaction histories, money transfers, bill payment, mobile and online banking, SMS and email alerts are just a few of the features that the system provides. In order to provide dependability, scalability, performance, usability, accessibility, compatibility, maintainability, availability, portability, interoperability, flexibility, modifiability, reusability, and testability, the system has been developed to meet a variety of non-functional needs.

Modern authentication and authorization technologies were used in the system's development to provide appropriate user-level accessibility. In order to remove any irregularities that can result from database transactions carried out by organizational administration and general users, the system has been normalized.

Because the user interfaces are browser-specific, the entire system can be made distributedly accessible. To guarantee that it satisfies all functional and non-functional requirements, the system has undergone extensive testing. In conclusion, this online banking system offers a variety of functionalities and guarantees dependability, scalability, performance, accessibility, compatibility, maintainability, availability, portability, interoperability, flexibility, modifiability, reusability, and testability. It also gives users safe and easy access to their accounts.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1 Overview:

The online banking system being developed here allows users to conveniently and securely access their accounts. A distributed architecture was used in the system's design, and Java was used to create the user interfaces for simplicity of maintenance and cross-platform compatibility.

1.2 About the project:

A number of features are available through the online banking system, such as user registration, deposit and withdrawal of funds, checking account balances, changing PINs, transaction histories, money transfers, bill payment, mobile and internet banking, SMS and email alerts. In order to provide dependability, scalability, performance, usability, accessibility, compatibility, maintainability, availability, portability, interoperability, flexibility, modifiability, reusability, and testability, the system has been developed to meet a variety of non-functional needs.

1.3 Purpose:

Offering clients a comfortable and safe online financial management platform is the aim of this project. Customers can do banking operations, including deposits, withdrawals, transfers, and bill payments, from the convenience of their homes or while they're on the road thanks to the system's many features.

1.4 Scope:

This project's scope involves designing and developing an online banking system that gives users convenient, safe access to their accounts. A distributed architecture was used in the system's design, and Java was used to create the user interfaces for simplicity of maintenance and cross-platform compatibility.

1.5 Functional components of the project:

User registration, login and authentication, deposits and withdrawals, account balance checks, PIN changes, transaction histories, money transfers, bill payment, mobile and online banking, SMS and email notifications are among the functional components of this project.

1.6 Functionalities to enhance the project:

Some functionalities that can be added to enhance this project include:

- Integration with third-party payment gateways for online shopping and bill payments.
- Implementation of a chatbot for customer support and inquiries.
- Development of a mobile app for iOS and Android devices.
- Implementation of voice-activated commands for hands-free banking.

CHAPTER-2

Requirement Specification

2.1 description of each functionality:

The project consists of several classes, each with its own set of functionalities. Here's a breakdown of each class and its purpose:

2.1.1 Login:

This class is responsible for handling the login functionality of the online banking system. It has a GUI that prompts the user to enter their card number and PIN. When the user clicks the "SIGN IN" button, the program checks if the entered card number and PIN match any records in the database. If there's a match, the program opens the Transactions class for that user. If not, it displays an error message.

2.1.2 Signup: This class is responsible for handling the signup functionality of the online banking system. It has a GUI that prompts the user to enter their personal and account details. When the user clicks the "Next" button, the program checks if all the required fields are filled out. If they are, the program creates a new record in the database with the user's information. If not, it displays an error message.

Signup2:

This class is the second page of the signup process. It prompts the user to enter their additional account details, such as their religion, category, income, education, occupation, PAN number, Aadhaar number, and senior citizen status. When the user clicks the "Next" button, the program checks if all the required fields are filled out. If they are, the program creates a new record in the database with the user's information. If not, it displays an error message.

Signup3:

This class is the third page of the signup process. It prompts the user to select their account type and card number. It also displays the user's form number and PIN. When the user clicks the "Submit" button, the program creates a new record in the database with the user's account details. If the user clicks the "Cancel" button, the program exits.

Conn: This class is responsible for handling the database connection for the online banking system. It has a method called getConnection() that returns a Connection object, which is used by the other classes to interact with the database.

2.1.3 Transactions:

This class is the main interface for the online banking system. It has a GUI that displays the user's account balance and offers various options for transactions, such as deposit, withdrawal, fast cash, mini statement, pin change, and balance enquiry. When the user

clicks on any of these options, the program opens the corresponding class for that transaction.

2.1.4 Deposit:

This class is responsible for handling the deposit functionality of the online banking system. It has a GUI that prompts the user to enter the amount they want to deposit. When the user clicks the "DEPOSIT" button, the program checks if the entered amount is valid. If it is, the program updates the user's account balance in the database. If not, it displays an error message.

2.1.5 Withdrawl:

This class is responsible for handling the withdrawal functionality of the online banking system. It has a GUI that prompts the user to enter the amount they want to withdraw. When the user clicks the "WITHDRAW" button, the program checks if the entered amount is valid and if the user has enough balance in their account. If both conditions are met, the program updates the user's account balance in the database. If not, it displays an error message.

2.1.6 FastCash:

This class is responsible for handling the fast cash functionality of the online banking system. It has a GUI that offers the user predefined amounts to withdraw. When the user clicks on any of these amounts, the program checks if the user has enough balance in their account. If they do, the program updates the user's account balance in the database and dispenses the cash. If not, it displays an error message.

2.1.7 Pin:

This class is responsible for handling the pin change functionality of the online banking system. It has a GUI that prompts the user to enter their current PIN and the new PIN twice for verification. When the user clicks the "CHANGE" button, the program checks if the entered PINs match. If they do, the program updates the user's PIN in the database. If not, it displays an error message.

2.1.8 BalanceEnquiry: This class is responsible for handling the balance enquiry functionality of the online banking system. It has a GUI that displays the user's account balance.

2.2 Functional Requirements:

User Authentication: The system should allow users to log in using their card number and PIN.
User Registration: The system should allow new users to register by providing their personal and account details.
Deposit Money: The system should allow users to deposit money into their account.
Withdraw Money: The system should allow users to withdraw money from their account.
Check Account Balance: The system should allow users to check their account balance.
Change PIN: The system should allow users to change their PIN.
Fast Cash: The system should allow users to withdraw a predefined amount of money without entering their account details.
Transaction History: The system should allow users to view their transaction history.
Transfer Money: The system should allow users to transfer money to other accounts.
Pay Bills: The system should allow users to pay their bills online.
Mobile Banking: The system should allow users to access their account and perform transactions using their mobile phone.
Internet Banking: The system should allow users to access their account and perform transactions using the internet.
SMS Banking: The system should allow users to perform basic transactions using SMS.
Email Alerts: The system should allow users to receive alerts about their account activity via email.

2.3 Non-Functional Requirements:

Security: The system should be secure and protect user data from unauthorized access.
Reliability: The system should be reliable and perform consistently without crashing or freezing.
Scalability: The system should be scalable and able to handle a large number of users and transactions.
Performance: The system should have fast response times and minimal lag.
Usability: The system should be easy to use and understand.
Accessibility: The system should be accessible to users with disabilities.
Compatibility: The system should be compatible with different operating systems and devices.
Maintainability: The system should be easy to maintain and update.
Availability: The system should be available 24/7 with minimal downtime.
Portability: The system should be portable and accessible from different locations.
Interoperability: The system should be able to work with other systems and applications.
Flexibility: The system should be flexible and able to adapt to changing user needs.
Modifiability: The system should be easy to modify and customize.
Reusability: The system should be reusable and able to be used for different purposes.
Testability: The system should be easy to test and debug.