

Use Case Diagram

```
@startuml
left to right direction
actor Customer
actor BankEmployee
rectangle "Online Banking System" {
    usecase "Register Account" as UC1
    usecase "Login" as UC2
    usecase "Check Account Balance" as UC3
    usecase "Transfer Funds" as UC4
    usecase "Pay Bills" as UC5
    usecase "View Transaction History" as UC6
    usecase "Apply for Loan" as UC7
    usecase "Manage Beneficiaries" as UC8
    usecase "Logout" as UC9
    usecase "Approve Loan Application" as UC10
    usecase "Reset Password" as UC11
}
Customer -- UC1
Customer -- UC2
Customer -- UC3
Customer -- UC4
Customer -- UC5
Customer -- UC6
Customer -- UC7
Customer -- UC8
Customer -- UC9
Customer -- UC11
BankEmployee -- UC10
UC7 ..> UC10 : "Requires Approval"
UC2 ..> UC11 : "Forgot Password?"
@enduml
```

Class Diagram

```
@startuml
class Customer {
    +registerAccount()
    +login()
    +checkBalance()
    +transferFunds()
    +payBills()
    +viewTransactionHistory()
    +applyForLoan()
    +manageBeneficiaries()
    +logout()
    +resetPassword()
}
```

```
}
```

```
class BankEmployee {  
    +approveLoanApplication()  
}
```

```
class Account {  
    -accountNumber  
    -balance  
    +getBalance()  
    +updateBalance()  
}
```

```
class Transaction {  
    -transactionId  
    -amount  
    -date  
    +getTransactionDetails()  
}
```

```
class Loan {  
    -loanId  
    -amount  
    -interestRate  
    -status  
    +applyLoan()  
    +getLoanDetails()  
    +calculateEMI()  
}
```

```
class Notification {  
    -message  
    +sendEmail()  
    +sendSMS()  
}
```

Customer --> Account

Customer --> Transaction

Customer --> Loan

BankEmployee --> Loan : "approves"

Loan --> Notification : "sends status update"

Transaction --> Notification : "sends receipt"

@enduml

Sequence Diagram

```
@startuml
actor Customer
participant "Online Banking System" as OBS
participant Account
participant Transaction
participant Loan
participant BankEmployee
participant Notification

Customer -> OBS: login()
OBS -> Account: verifyCredentials()
Account -> OBS: authenticationSuccess()
OBS -> Customer: loginSuccess()

Customer -> OBS: checkBalance()
OBS -> Account: getBalance()
Account -> OBS: returnBalance()
OBS -> Customer: displayBalance()

Customer -> OBS: transferFunds()
OBS -> Transaction: initiateTransfer()
Transaction -> Account: updateBalance()
Account -> Transaction: confirmTransaction()
Transaction -> OBS: transactionSuccess()
OBS -> Notification: sendSMS()
Notification -> Customer: transferReceipt()

Customer -> OBS: applyForLoan()
OBS -> Loan: processLoanApplication()
Loan -> BankEmployee: requestApproval()
BankEmployee -> Loan: approveLoan()
Loan -> OBS: loanApproved()
OBS -> Notification: sendEmail()
Notification -> Customer: loanStatus()

Customer -> OBS: logout()
OBS -> Customer: logoutSuccess()
@enduml
```

Activity Diagram

```
|
@startuml
start
:User Opens Online Banking Portal;
```

```

:Enter Credentials;
if (Are Credentials Valid?) then (Yes)
  :Display Dashboard;
  repeat
    :Choose an Action;
    switch (Selected Action)
      case (Check Balance)
        :Retrieve Account Balance;
        :Display Balance;
      case (Transfer Funds)
        :Enter Transfer Details;
        if (Is Transfer Valid?) then (Yes)
          :Process Transfer;
          :Send SMS Notification;
          :Transfer Successful;
        else (No)
          :Show Error Message;
        endif
      case (Apply for Loan)
        :Submit Loan Application;
        if (Loan Approved?) then (Yes)
          :Approve Loan;
          :Send Email Notification;
          :Notify User;
        else (No)
          :Reject Loan;
          :Notify User;
        endif
      case (Reset Password)
        :User Requests Password Reset;
        :Send Reset Email;
        :User Enters New Password;
        :Password Updated;
    endswitch
  repeat while (User Wants Another Action?)

  :Logout;
  :End Session;
else (No)
  :Show Login Error;
endif
stop
@enduml

```

Component Diagram

```

@startuml
component "User Interface" as UI
component "Authentication Service" as Auth

```

component "Banking Services" as Banking
component "Transaction Service" as Transaction
component "Loan Processing" as Loan
component "Notification Service" as Notify
component "Database" as DB

UI --> Auth : User Login
Auth --> DB : Verify Credentials
Auth --> UI : Authentication Success/Failure
UI --> Banking : Request Account Info
Banking --> DB : Fetch Account Details
Banking --> UI : Display Account Details
UI --> Transaction : Transfer Funds
Transaction --> Banking : Validate Funds
Transaction --> DB : Update Account Balance
Transaction --> Notify : Send Transaction Alert
Transaction --> UI : Show Transaction Status
UI --> Loan : Apply for Loan
Loan --> Banking : Check Eligibility
Loan --> DB : Store Loan Request
Loan --> Notify : Send Loan Status Update
Loan --> UI : Loan Approval/Rejection
@enduml

Deployment Diagram

@startuml
node "User Device" {
 component "Web Browser" as Browser
}

node "Bank Server" {
 component "Web Server" as WebServer
 component "Application Server" as AppServer
 database "Banking Database" as DB
 component "Notification Service" as Notify
}
@enduml

Browser --> WebServer : HTTP Request (Login, Transactions)
WebServer --> AppServer : Process Request
AppServer --> DB : Fetch/Update Data
DB --> AppServer : Send Response
AppServer --> Notify : Send Email/SMS Alerts
AppServer --> WebServer : Send Processed Data
WebServer --> Browser : Display Result
@enduml

State Chart Diagram

@startuml

[*] --> Idle

Idle --> LoggingIn : Enter Credentials

LoggingIn --> LoggedIn : Authentication Successful

LoggingIn --> Blocked : Too Many Failed Attempts

Blocked --> Idle : Reset Password

LoggedIn --> CheckingBalance : Check Balance

CheckingBalance --> LoggedIn : Balance Displayed

LoggedIn --> TransferringFunds : Transfer Funds

TransferringFunds --> TransactionCompleted : Transfer Successful

TransferringFunds --> LoggedIn : Transfer Failed

LoggedIn --> ApplyingForLoan : Apply for Loan

ApplyingForLoan --> LoanProcessing : Processing Loan

LoanProcessing --> LoanApproved : Loan Approved

LoanProcessing --> LoanRejected : Loan Rejected

LoanApproved --> LoggedIn

LoanRejected --> LoggedIn

TransactionCompleted --> LoggedIn

LoggedIn --> UpdatingProfile : Update Profile

UpdatingProfile --> LoggedIn : Profile Updated

LoggedIn --> LoggingOut : Logout Request

LoggingOut --> Idle : Session Ended

@enduml

Collaboration Diagram

@startuml

title Collaboration Diagram for Online Banking System

object Customer

object OnlineBankingSystem

object Account

object Transaction

object Loan

object BankEmployee

object Notification

' User login process

Customer -down-> OnlineBankingSystem : (1) login()
OnlineBankingSystem -down-> Account : (2) verifyCredentials()
Account -down-> OnlineBankingSystem : (3) authenticationSuccess()
OnlineBankingSystem -down-> Customer : (4) loginSuccess()
OnlineBankingSystem -down-> Customer : (5) loginFailed()
Customer -down-> OnlineBankingSystem : (6) resetPassword()

' Checking balance

Customer -down-> OnlineBankingSystem : (7) checkBalance()
OnlineBankingSystem -down-> Account : (8) getBalance()
Account -down-> OnlineBankingSystem : (9) returnBalance()
OnlineBankingSystem -down-> Customer : (10) displayBalance()

' Fund transfer

Customer -down-> OnlineBankingSystem : (11) transferFunds()
OnlineBankingSystem -down-> Transaction : (12) initiateTransfer()
Transaction -down-> Account : (13) debitAmount()
Transaction -down-> Account : (14) creditAmount()
Transaction -down-> OnlineBankingSystem : (15) transactionSuccess()
OnlineBankingSystem -down-> Notification : (16) sendSMS()
Notification -down-> Customer : (17) transferReceipt()

' Loan Application

Customer -down-> OnlineBankingSystem : (18) applyForLoan()
OnlineBankingSystem -down-> Loan : (19) processLoanApplication()
Loan -down-> BankEmployee : (20) requestApproval()
BankEmployee -down-> Loan : (21) approveLoan()
Loan -down-> OnlineBankingSystem : (22) loanApproved()
OnlineBankingSystem -down-> Notification : (23) sendEmail()
Notification -down-> Customer : (24) loanStatus()

' Logout process

Customer -down-> OnlineBankingSystem : (25) logout()
OnlineBankingSystem -down-> Customer : (26) logoutSuccess()

@enduml











