**User Account Use-Case**

Table of Contents

[Overview 2](#_Toc478567909)

[Technologies used 2](#_Toc478567910)

[Tools Used 2](#_Toc478567911)

[Assumptions 2](#_Toc478567912)

[Resource URI’s 2](#_Toc478567913)

[Logging in the Application 2](#_Toc478567914)

[Checking account details 2](#_Toc478567915)

[Deposit money: 2](#_Toc478567916)

[Adding test data (Not the part of use case, based on assumption) 2](#_Toc478567917)

[Database Tables 3](#_Toc478567918)

[Validation/Error Message 3](#_Toc478567919)

[Repository URL 3](#_Toc478567920)

[Application Flow with screenshot 3](#_Toc478567921)

# Overview

This application has been created w.r.t. use case provided for user account management. Using this App user can do the following functionalities.

* User will be able to login the application with valid credentials.
* User will be able to see the personal and account details with provided values in database.
* User can submit the amount in his/her own account as well as some other’s account as well.

# Technologies used

* AngularJs 1.x
* Restful Web Service (Jax-RS)
* Hibernate

# Tools Used

* Eclipse Mars 2.0
* Brackets
* Git Repository
* H2 Database
* Maven

# Assumptions

* There are 2 main roles : Admin and User
* Admin Role: Access provided for personal details and Deposit Money Tab.
* User Role: Access provided for all three tabs i.e. personal details, account details and Deposit Money Tab
* Assuming that test data is already provided and it’s not the part of test cases.
* Data needed to be provided in the table structure decided for this use case during development.
* For development H2 database is being used and interacted with hibernate. If different database to be used then driver name and dialect needed to be updated in hibernate configuration.

# Resource URI’s

## Logging in the Application

users/{userName}/{password}

## Checking account details

users/{customerId}/accounts/{accountId}

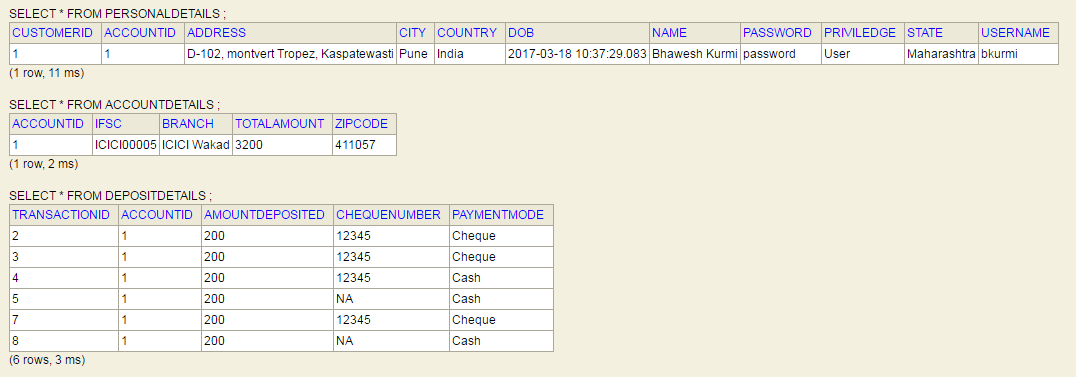
## Deposit money:

users/accounts/{accountId}/deposits/{amountDeposited}/{paymentMode}/{chequeNumber}

## Adding test data (Not the part of use case, based on assumption)

users/addtestdata

# Database Tables



# Validation/Error Message

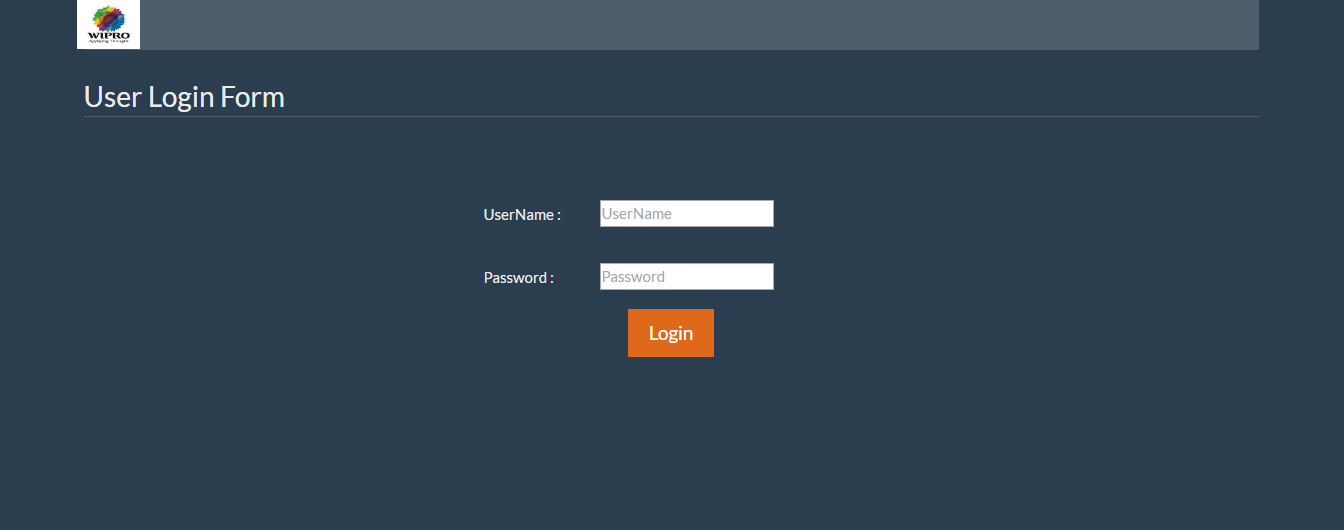
* If User/Admin trying to deposit money and server or Database is down.
  + Exception Message: “Database or Server is down.”
* If User/Admin trying to deposit money via Cash/Cheque and Account does not exist.
  + Exception Message: “Account does not exist.”
* User/Admin trying to login and username or password is invalid.
  + Exception Message: “Invalid Username and Password.”

# Repository URL

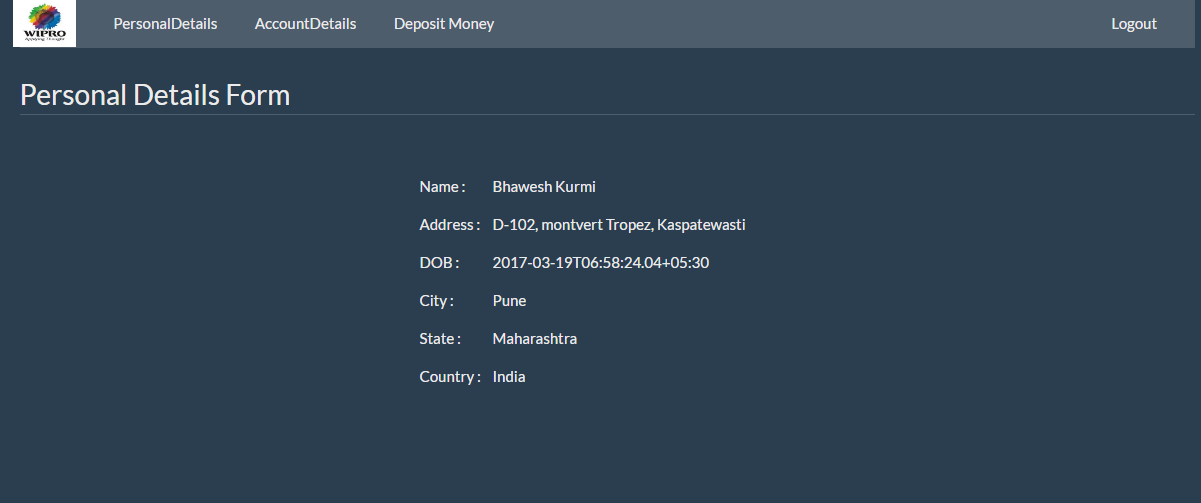
* UI Code (Angular JS)
  + <http://wosggitlab.wipro.com/PA316586/WproProjectUISubmission.git>
* RESTFul API (JAX-RS)
  + <http://wosggitlab.wipro.com/PA316586/UserAccountAPI.git>
* User Account API Document
  + http://wosggitlab.wipro.com/PA316586/WiproUseCaseDocuments.git

# Application Flow with screenshot

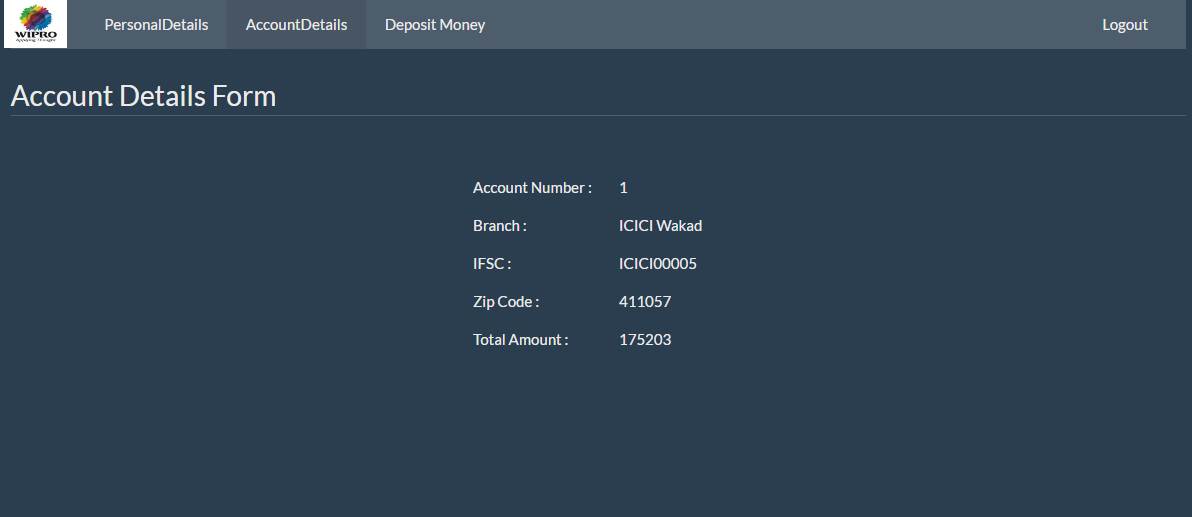
1. Application Login



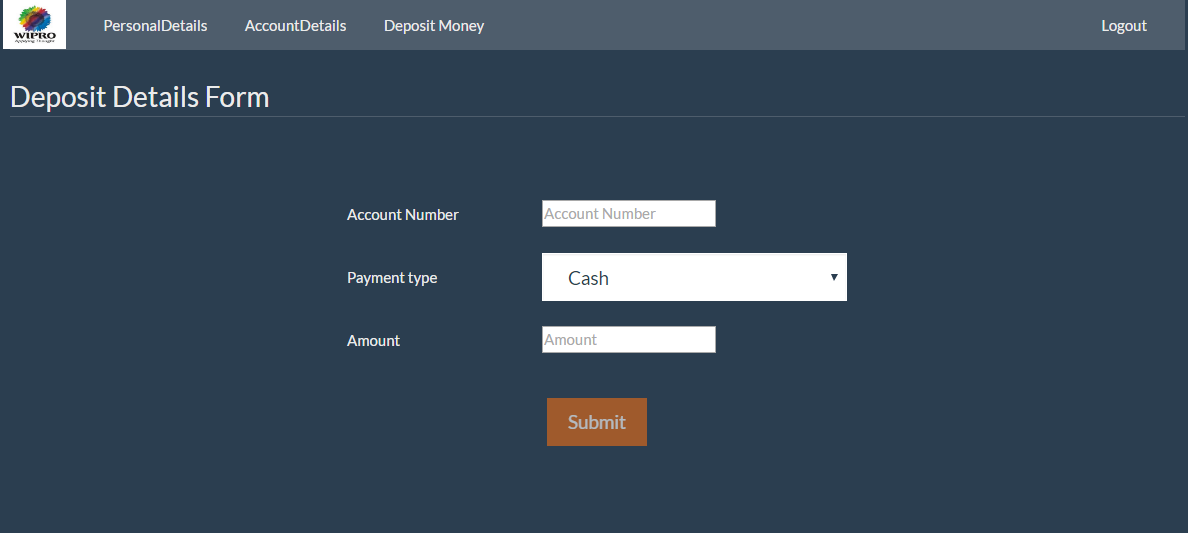
1. Personal Details



1. Account Details(Enabling tab as per type of user (Admin/User))



1. Deposit Money with Cash



1. Deposit Money with Cheque

