**Money and Monetary**

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **Name of the Project** | **Money and Monetary** | |
| ***2*** | **Problem Statement** | Design a system Using Mobile, which credits extra spent by customers back into their accounts, as rounded off figure  NWG Bank's aim is to provide all personal customers a way to save easily to aid advancement in the Bank's goal of helping more customers save more.  As a “round up “service, customers who make a debit card transaction have that transaction rounded up to the nearest pound and that ‘extra’ credited to an instant access savings account. | |
| **3** | **Objective/ Vision** | Design a platform for customers which automatically rounds up purchases done by debit card to the nearest pound and then transfers the difference from your spending account into your wallet balance which can be transferred to your savings account. The platform should provide the options to customers for using their wallet balance in making the next transactions if the wallet balance reaches to a certain amount. | |
| **4** | **Users of the System** | All customers. | |
| **5.** | **Assumptions** | 1. Customers is registered with the bank already using mobile | |
| ***6*.** | **Functional Requirements** | 1. Customers should register and then login with the platform. While registering, customers need to provide the following information, the mobile number and email for registering.  2. Customers can check their wallet balance and the savings account balance.  3. Customers can make purchases from the platform.  4. Sample Application for Product purchase  5. Transaction History  6. Savings –Transfer Process | |
| **7.** | **Non-functional requirements** | 1. Application should be responsive to display consistently across multiple device screens. 2. The password should not be saved directly into the database (encryption method should be used) 3. UI should be user friendly 4. Backend / Front end should have been tested 5. URLs should be protected 6. Documented using Swagger 7. AOP – cross cutting concerns | |
| **8.** | **Tools and Technologies to**  **be used** | | 1. Frontend – Angular 2. Backend - Spring Boot 3. Database – MySql |
|  |  |  |  |

Components of the Project:

|  |  |
| --- | --- |
| Database Models | Description |
| **Customer** | CustomerId  CustomerEmail  CustomerMobile  CustomerName  CustomerPassword |
| **Account** | AccountId  CustomerId  WalletBalance  SavingsAccountBalance |
| **Transaction** | AccountId  CustomerId  TransactionId  TransactionDate  TransactionAmount |
|  |  |

|  |  |
| --- | --- |
| Services | Description |
| **Customer** | RegisterCustomer  UpdateCustomer  ValidateCustomer  GetDetails |
| **Transaction** | GetAllTransactions  AddTransaction |
| **Account** | GetWalletBalance  GetAccountBalance  StoreBankAccountDetails |
|  |  |

|  |  |
| --- | --- |
| UI Components | Description |
| **Login** | Customer perform login using this component |
| **Signup** | Customer can sign up using this component |
| **Edit Profile** | Customers can edit personal info such as name, mobile and password. |
| **CustomerDashboard** | Shows Customer Details, Wallet Balance and Account Balance |
| **Cart Page** | For making a purchase by entering the amount. |
| **Transaction History** | Shows previous Transactions. |

|  |  |
| --- | --- |
| **Future Scope** | We can provide some rewards on each transaction which can be redeemed. |