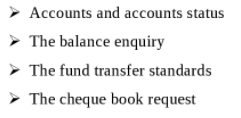
Digital Bancs product is defined as the automated delivery of new and traditional banking products and services directly to customers through electronic, interactive communication channels. It includes the system that enables financial institution customers, individuals of businesses, to access accounts, transact business or obtain information on financial products and services through internet.

It includes different modules such as providing information of



After logging in to the application with a valid user, the user can check account types with account status.

A user can create account by providing required information such as name, address details, contact details, date of birth, account type etc.

A user can transfer funds from one currency type to another currency type between different accounts.

A user can request for checkbook for the selected account for which he wants to issue a check book.

I have worked on developing backend services of Fund transfer module for currency conversion and Forex service for exchanging currency from one type to another type between different accounts.

There are two restful micro services such as currency-conversion-service Forex service

FOREX-SERVICE is an internal service that determines Foreign Exchange market rate of specific currency from database with the use of JPA, and returns the results for presentation to consumers. Consumers never talk directly to this service, though other applications we write may re-use this service.

CURRENCY-CONVERSION-SERVICE can convert a bucket of currencies into another currency which is a consumer-facing microservice that interprets and routes requests to internal services to perform appropriate currency conversions with the use of FOREX-SERVICE.

**Spring cloud Config Server** is used which provides a centralized **server** for delivering external configuration properties to an **application** and a central source for managing this configuration across deployment environments.

Spring-cloud-OpenFeign rest client is used to communicate both the microservices which provides pluggable annotation support to call the REST services instead of writing representative boilerplate code.,

spring-cloud-starter-netflix-eureka-client is used for service registration and service discovery. It holds information about all client service applications. Each microservice registers itself with the Eureka naming server.

both our CURRENCY-CONVERSION-SERVICE and FOREX-SERVICE will register their URLs with Eureka at start-up, and CURRENCY-CONVERSION-SERVICE will use Eureka to discover required details from FOREX-SERVICE .

Spring cloud load balancer balances load between multiple instances of microservices by fetching service information from eureka client.

Spring cloud api gateway is implemented for all the microservices which provides features like

* Service Aggregation
* Authentication, authorization and Security
* Rate Limits
* Fault Tolerance

A LoggingFilter class is implemented which extends Zuul filter class to log the request details which pass through API gateway .

Spring Cloud Sleuth assigns a **unique Id** to each request a user made.

Zipkin distributed tracing server is used to manage the traces details of the subsequent services.

/to centralize all the logs from all the microservices, Rabbit MQ server is used. All the microservices puts the log messages on the queue called **RabbitMQ,** and the Zipkin picks these log messages from the RabbitMQ./

Zipkin UI Dashboard is used to trace the request details fired by the user.

Resilence 4j is used for features like Circuit Breaker, Rate Limiter, Retry or Bulkhead.

Circuit breaker is used maintain the flow of microservices uninterrupted in case of any microservices failure.



Why Currency exchange service is taken as separate service?

Because the currency exchange is used with many other microservices as per example for ATM related transactions.

forex card as a prepaid travel card that you can load with a foreign currency of your choice. You can use a forex card just like a credit or debit card to pay for your expenses in a local currency abroad. You can withdraw local cash from an ATM.

A forex card comes with two main variants–multicurrency forex cards and single currency cards. A single currency card has limited use, and you will incur higher cross-currency charges if you use it in another currency. A multicurrency card for example, can travel with you wherever you go. You can load it with up to 23 currencies and use it across the world. You can also shuffle funds from one currency to another whenever you need via prepaid NetBanking -- for example, if you are visiting two countries which have different currencies.

**Project 2:**

KYC – Know Your Customer process that is followed in large Financial Institutions (FI) involves understanding the type of accounts & products a customer owns in an institution, verifying the documentary evidences and ID’s produced by the customer to prove the reliability of the information furnished

Know Your Customer enables banks to understand its customers and their financial dealings to serve them better and manage its risks prudently.

There are various modules included. Such as

Client registration

Client Data Capture

Client due diligence

Client acceptance

Client will be registered by creating his own username and password.

After the authenticated client login, client can add all required KYC details to his account.

Client due diligence is the process of assessing client’s background to determine their identity and the level of risk they possess.

The clients will be accepted whose identity is established by conducting due diligence appropriate to the risk profile of the client.

All the server side logic for client data capture have been developed using express module of Node.

Body-parser is used for parsing the incoming request bodies.

Mongoose is used for connecting to mongo DB by creating a schema then creating a mongoose model using the schema. All the client data will be stored in mongo db.

The mongoose model is used for different CRUD operations in the database.

Robo 3T is used as the graphical user interface for mongo DB.

EJS is used for generating web pages that can include dynamic data and can share templated pieces with other web pages (such as common headers/footers).

Express route parameters is used to capture values at specific positions in the URL and chainable route handlers are used for creating modular routes

Nodemon is used for automatically restarting the node application when file changes are detected.

Lodash is a JavaScript library which is used provides utility functions for common programming tasks

Dotenv is used to store environment-specific variables

Express-session is used to store session for each user of our application.

Passport is used as authentication middleware which comes with a comprehensive set of strategies.

Passport-local-mongoose module is used for auto-generating salt and hash fields in the Database.

# passport-google-oauth20 is used for authenticate using google. Mongoose findOrCreate Plugin is used during creation of new google strategy.

Agile workflow(sdlc method)

## SDLC Phases

**Given below are the various phases:**

* Requirement gathering and analysis
* Design
* Implementation or coding
* Testing
* Deployment
* Maintenance

### #1) Requirement Gathering and Analysis

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

**For Example,** A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like what kind of transactions will be done, how it will be done, in which currency it will be done, etc.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

### #2) Design

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

### #3) Implementation or Coding

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

### #4) Testing

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Retesting, regression testing is done until the point at which the software is as per the customer’s expectation. Testers refer SRS document to make sure that the software is as per the customer’s standard.

### #5) Deployment

Once the product is tested, it is deployed in the production environment or first [UAT (User Acceptance testing)](https://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/) is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

### #6) Maintenance

After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

I have completed my B.tech in information and technology from institute of technical education and research. Currently I am working as a system engineer in TCS. I have worked on Bancs Digital and KYC compliance product

TCS BαNCSRetail LOS is developed specifically to support the Retail Loan Application Processing needs of banks and other financial institutions. It supports processing of Loan Applications right from ingestion, through Adjudication and finally fulfillment. BαNCSRLOS has an in—built Rule Engine which handles automatic loan approvals and workflow system to define Queues and Alerts to the Loan officers within an Organization. The in-built document handling capabilities within the system helps an organization to achieve paperless office.

* **Application Creation:**This scenario is used to create a new Loan Application in LOS. User can apply for different kind of loan as per his requirement (Residential Finance, Personal Finance, and Auto Lease).Customer details are fetched from Core Banking based on ID Type and ID number captured on the screen. The loan product details will be entered by the user and on submitting the form a new application is ingested in LOS and the application will move to the personal queue of the logged in user.

There was a group of 41 Jewish soldiers surrounded by Roman army, and they didn't want to get caught. So, they sat down in a circle and came up with an algorithm. Everybody had a sword, and starting from person #1 in the circle, everybody will kill the next living person on the left. So, #1 will kill #2. #3 will kill #4, #5 will kill #6 and so on. The last living person will have to commit suicide to avoid getting caught by Romans.

The soldier called Josephus preferred to be caught over committing suicide. So, in the group of 41 soldiers, he chose the location where he will be the last person living.

Write a program to figure out, in a group of given N people, where should Josephus sit to live at the end of all internal killing.

There is a mathematical solution to this problem. But, your program should use the brut force method to find the position. The output of the program may look like this:

Solving Josephus problem for 5 soldiers:

1 kills 2

3 kills 4

5 kills 1

3 kills 5