

ONLINE BANKING APPLICATION

CONTENTS

1. Problem Statement.....	2
2. Proposed Wireframe	5
3 Application Architecture	6
4. Tool Chain	6
5. Development flow	7
6. Process-Requirement:	8
7. Rubrics/Expected Deliverables.....	9
Rest API (Products & Frameworks -> Compute & Integration):	9
Database (Products & Frameworks -> Database & Storage):.....	10
Testing.....	10
8. Frontend	10
9. Methodology	10
Agile	10

1. PROBLEM STATEMENT

Online Banking App is microservices based application which allows the user to register and login and do the banking transactions.

The objective of this document is to provide a comprehensive overview of the Online Banking System. It aims to describe the system's purpose, features, interfaces, functionality, operational constraints, and responses to external stimuli.

The online banking system is designed to be universally applicable, available wherever banking services are offered. It offers increased efficiency and ease of record-keeping compared to traditional banking methods. With the online system, individuals can conveniently access their accounts according to their authorized permissions. Due to its numerous advantageous features and expedited transaction processes, banks are increasingly favouring the online banking system over traditional methods.

Functionalities:

This software will have following functionalities.

- Online balance check and transaction information:
- Customer will be able to check his balance online while sitting at home by accessing the database of the bank using his/her password and account no. allotted him by the bank.
- Save or view up to 1 year history of transaction:
- It will be easy for the customer to view or save his history transactions up to past 1year transactions. It will provide him the opportunity to maintain his bank balance and needs.

Balance transfer:

This system will provide a path to the customer of the bank to transfer his balance to other account in easy steps. A small transfer fee will be applicable for this transaction.

Online record Entry:

Bank staff will input and maintain their record online. It will be easy and efficient for them to serve more and more people in less time.

Online record search:

Bank staff will easily search a record and update it if needed. Transactions will be faster even physically from the branch because it will be very easy for the bank staff to check the balance of a specific person and update its record if necessary.

Online Billing Option:

Customers will be able to shop online and pay the bills from their account. A secure way will be provided for the billing. Online shopping will provide them the easiest way to buy and sell their items.

Check book Allotment:

If the customer's checks have been completed, a new check book will be allotted to him.

User Characteristics:

There are various kinds of users for the product. Usually, web products are visited by various users for different reasons.

The users include:

- Chancellor who will be acting as the controller and he will have all the privileges of administrator.
- All the persons who need to perform banking.

Generals Constraints:

Some general constraints should be defined which will have a great part in the overall succession of the online banking project.

Hardware Requirements:

As this system is an online Web-based application so a client server will be the most suitable Organizational style for this system. Computer systems will be needed by each of the actor as well as that user must be connected to the internet. So, concisely following hardware will be needed.

- Computer systems
- Internet availability

Safety and Security:

This Project must be safe and secure because customers will directly contact their account through the internet. Software will have to identify the valid customer according to his/her bank details and password. So, it is a difficult task to prevent the system by major disasters by preventing the unauthorized access to the system.

Assumptions and Dependencies:

Following are the assumptions and dependencies which are related to this online banking project. This project is a stand-alone project so it will not affect the system where it will be embedded. This project is a web-based project while the staff was addict of using traditional methods of data storage and retrieval so they will be trained a bit to jump to it.

This system will not depend on any other module. It will be a web-based so everyone will independently contact it.

It is will not affect the environment at all.

Banks will feel free to adopt it because it will not be so much expensive.

As this project contains valuable and new features so it will probably remove the previous online banking systems embedded in some banks.

Specific Requirements:

How the online banking will interact with the environment, what will be the functional and non-functional requirement. These all the steps should be defined here for providing a powerful base to the design phase. The design of the project will completely depend on the functional and non-functional requirements. So these should be defined clearly and accurately for the effectiveness.

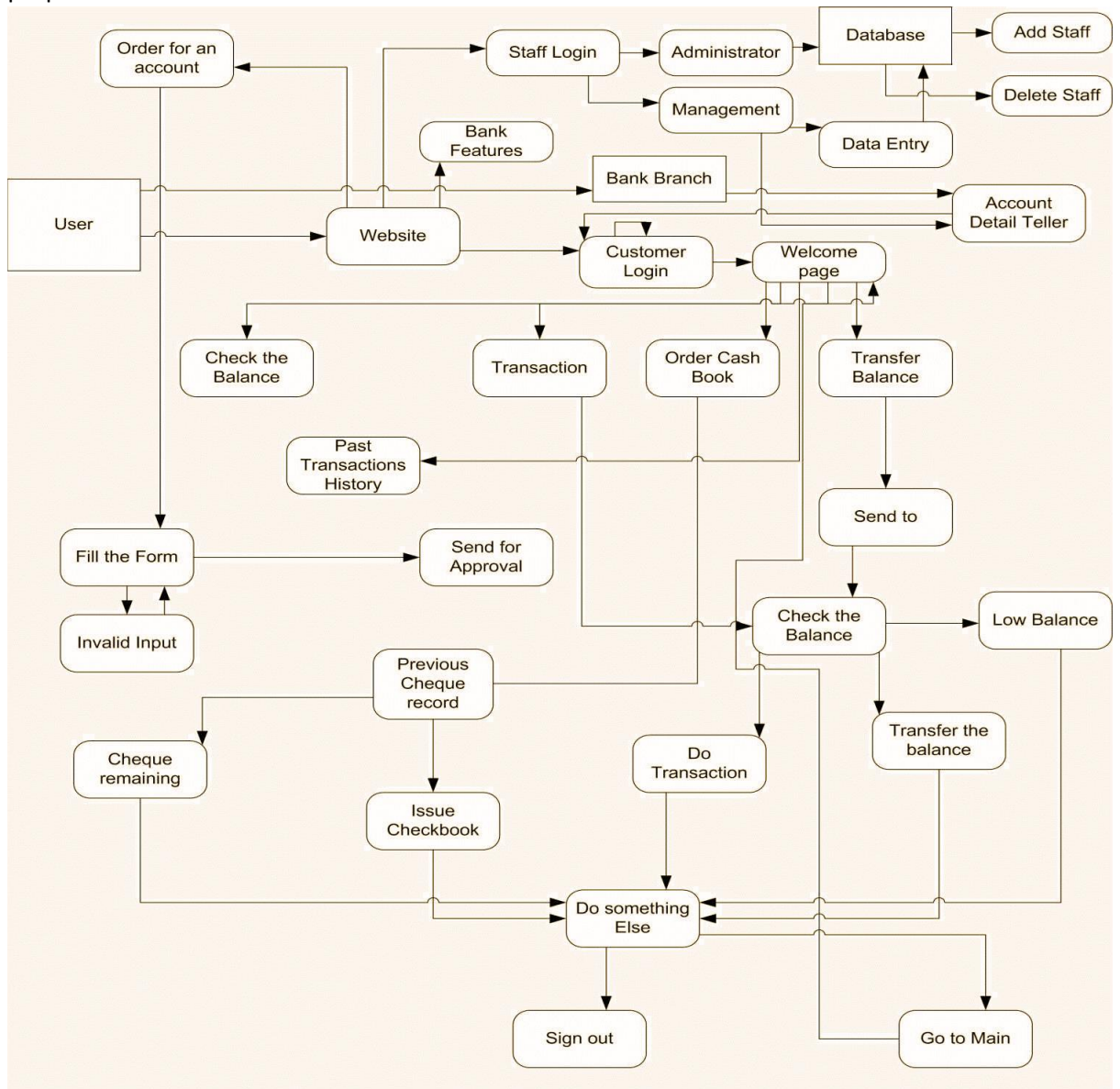
Functional Requirements:

Following are the services which this system will provide. These are the facilities and functions required by the customer.

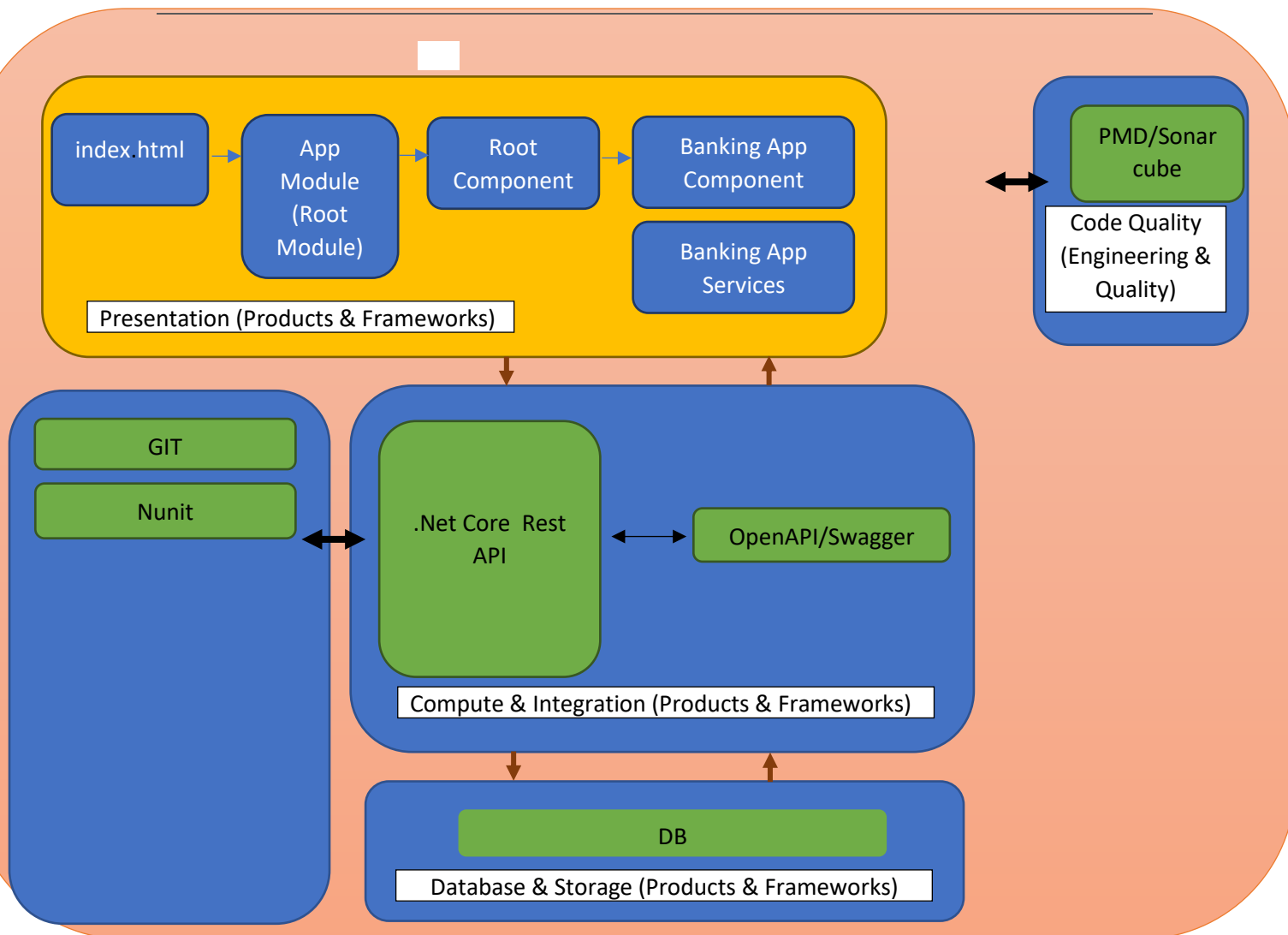
- Online balance check.
 - Online shopping opportunity.
 - Online data entry by the staff.
 - Updating the data.
 - Balance transfer.
 - Check book Allotment.
-

2.PROPOSED WIREFRAME

1. UI needs improvisation and modification as per given use case. Below diagram is for reference purpose.



3 APPLICATION ARCHITECTURE



4. TOOL CHAIN

Competency	Skill	Skill Detail
Engineering Mindset	Networking and Content Delivery	
	Ways of Working	
	Consulting Mindset	
	DevOps	
Programming Languages	Application Language	C#
Products & Frameworks	Presentation	Angular/ React & Redux
		Karma & Jasmine
	Compute & Integration	.net core web API

	Database & Storage	
		MS SQL /My SQL
	Governance & Tooling	Git
		Nunit
Engineering Quality	Code Quality	Sonar Cube
Platform		

6.DEVELOPMENT FLOW

<u>Business Requirement</u>					
1	Backend	Rest API, Database, Messaging, Log/Monitoring, Testing		Code Submission and Evaluation, Panel Presentation	It is mandatory to complete this module with 60% to proceed with the next Frontend MC
2	Front End	Angular/React		Code Submission and Evaluation, Panel Presentation	It is mandatory to complete each module with 60% to proceed with the next Cloud MC
3	Cloud	Compute, Identity, Compliance, Security and Content Delivery		Code Submission and Evaluation, Panel Presentation	It is mandatory to complete this Cloud module with 60%

7.PROCESS-REQUIREMENT:

Customer Login:

Each Customer will have its account Id and password. This page will require both of these attributes for them to access their account.

Bank Features:

It isn't sure that each visitor of the Bank's website will be a customer. He/she would be a normal visitor interested in reading the features bank provides. The website's main page should provide him the basic features and benefits of the bank to these types of users.

Order for an Account:

A new visitor the Bank's website would be interested in opening a new account in the Bank. So he must be provided an easy path to create a new account in the bank.

Fill the Form:

New comer should have to fill the form to register him/her self with the bank. After filling the form, If the values inputted by the user were logical correct, his contact details will be sent to the administration block else he will be asked to input the values again.

Welcome Page:

After a user will be login, he will provided an interface offering different tasks (Here this interface will provide many of the functionalities, which the customer needs in the software). He has to choose a task to carry on his work.

Staff Login:

On the Website main page, A staff login link will also be provided. Bank staff will use to input their ID's and passwords to access their account. Here the type of staff will also be recognized, if he will be of administration block, he will be sent to the administration module else he will be sent to the record management module.

Check the balance:

After logging in, if the user wants to check his balance he will have to click the balance check link. It will tell him his current balance of the account through which he is logged in.

Transfer Balance:

If user wants to transfer his money to some other account, then this module will provide him this opportunity. He will input the account details of the receiver. After this process, server will check the balance of the user and if the transfer balance will be less than the account balance then transfer will take place else, he will be alarmed that he has lo balance.

Account detail teller:

If the user physically contacts the Bank branch, then he will provide his account detail to the management staff who will inform him about his account. Users will be able to do every task at the branch that they can do online from their home.

Order Cash Book:

If user's Cheque book has been finished, he will be able to order a new cheque book from this module.

External Interface Requirements:

These requirements are discussed under the following categorisation.

User interface:

Application will be accessed through a Browser Interface. The interface would be viewed best using 1024 x 768 and 800 x 600 pixels resolution setting. The software would be fully compatible with Microsoft Internet Explorer for version 6 and above.

No user would be able to access any part of the application without logging on to the system.

8. RUBRICS/EXPECTED DELIVERABLES

REST API (PRODUCTS & FRAMEWORKS -> COMPUTE & INTEGRATION):

- a. Use Spring Boot to version and implement the REST endpoints.
- b. Implement HTTP methods like GET, POST, PUT, DELETE, PATCH to implement RESTful resources:

POST	/api/v1.0/blog/register	Register as new blogger
GET	/api/v1.0/ blog /login	Login
GET	/api/v1.0/ blog /<username>/forgot	Forgot password
GET	/api/v1.0/ blog /domain/all	Get all blog on a particular domain
POST	/api/v1.0/ blog /<username>/add	Post new blog
PUT	/api/v1.0/ blog /<username>/update/<id>	Update blog
DELETE	/api/v1.0/ blog /<username>/delete/<id>	Delete blog
PUT	/api/v1.0/ blog /<username>/like/<id>	Like blog
POST	/api/v1.0/ blog /<username>/reply/<id>	Comment to other blog

- c. *username may be partial or complete username
 - d. Use necessary configuration in place for REST API in application.properties or bootstrap.properties or application.yml; whichever is applicable.
 - e. Package Structure for Spring Boot Project will be like com.blogapp.* with proper naming conventions for package and beans.
 - f. Use configuration class annotated with @Configuration and @Service for business layer.
 - g. Use constructor-based dependency injection in few classes and setter-based dependency injection in few classes.
 - h. Follow Spring Bean Naming Conventions
-

DATABASE (PRODUCTS & FRAMEWORKS -> DATABASE & STORAGE):

1. As an application developer:
 - a. Implement ORM with EF CORE Data with My SQL or MS SQL Server
 - b. Have necessary configuration in place for REST API in application.properties or bootstrap.properties or application.yml OR API based configuration; whichever is applicable.

TESTING

1. Perform proper testing using Spring Test/Selenium and Cucumber and do proper CI/CD
2. The test coverage should be of 100%
3. Test Suites should cover both positive and exception handlings

9.FRONTEND

1. Develop the front end for all user stories.
2. Implement using either Angular or React
3. Implement all the Front-End validation rules
4. Proper naming conventions and folder structures
5. Implement using proper SOLID design principles
6. Perform unit and integration testing for the front end application.

11.METHODOLOGY

AGILE

1. As an application developer, use project management tool along to update progress as you start implementing solution.
 2. As an application developer, the scope of discussion with mentor is limited to:
 - a. Q/A
 - b. New Ideas, New feature implementations and estimation.
 - c. Any development related challenges
 - d. Skill Gaps
 - e. Any other pointers key to UI/UX and Middleware Development
-