Software Design Document (SDD)

Version 1.0

“e-Banking Management System”

Project By:

Bhagvat Giri

## 1.1. Purpose:

## The purpose of this document is to provide a detailed design

## description of the software product to allow for more advanced

## software development and design creation of software with

## fundamental understanding and idea of the design that is to be

## developed or structured and how it should be produced.

## majority if the contents are derived from eBMS SSR (Software

## specific Requirements) and some are assumed. This document

## include general description of design elements and their

## interaction, how system will be structured, data and functional

## structures are aimed to be discussed more to help

## generate test cases and safety requirements, design details also

## derived from SRS document.

## 1.2. Scope:

## 

## 

## An e-banking management system will be applicable

## 

## everywhere, where banking exists. It will be more efficient and

## easier way to have a Record on systems through which everyone

## can easily access it according to his/her rights as compared to

## the traditional banking system. People should be motivated to

## use e-banking management system instead of the traditional

## banking system as it contains man features and fastest

## techniques for the information transactions and display. Objective

## of the system is to make electronic banking easily accessible to

## everyone and manging their transactions on it. In these document

## design views for the system are presented with corresponding

## design elements and structured with related UML diagrams.

## 1.3.Definitions and Abbreviations:

## eBMS – e-Bank Management System

## Validation- checking if something is accurate or officially

## Acceptable

## Database- collection of all the data and information managed

## by a system

## Stakeholder- A person with an interest in a project who is

## not a developer

## Visitor/ User: Anyone who is using this website or has

## active account on it

## 

## Software Requirement

## Specifications: A paper or a document that lists out all the

## features or functions of a system and the

## conditions under which it must operate.

## 

## Credentials: Some inputs needed to authorize and validate

## One’s identity

## 

## PIN: Personal Identification number

## 

## SDD: Software Design Description

## Use Case Diagram: A type of static structure diagram in UML that describes user's interaction with the system

## Class Diagram: A type of static structure diagram in UML that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes

## Block diagram: A diagram showing in schematic form the general arrangement of the parts or components of a complex system or process.

## UI: User Interface

## 1.4. References:

## 

## <http://tomcat.apache.org/>

## <https://www.researchgate.net/publication/289247259_E-Banking_Management_Issues_Solutions_and_Strategies>

## <https://journals.sagepub.com/doi/abs/10.1177/2393957517736457>

## <https://www.onespan.com/blog/top-banking-regulations-security-compliance-requirements>

## <https://byjus.com/govt-exams/functions-of-bank/>

## <https://www.geeksforgeeks.org/data-flow-diagram-for-online-banking-system/>

## <https://lucid.app/documents#/dashboard>

## <https://www.academia.edu/31558728/BANKING_SYSTEM_SYSTEM_DESIGN_DOCUMENTBANKING_SYSTEM_SYSTEM_DESIGN_DOCUMENT>

## 2. Design Overview:

## 2.1: Design Goals

## The Design goals represent the desired qualities of e-Banking

## Management system and provide a consistent set of criteria that

## must be considered when design decisions are made. They system

## will have following goals to be achieved to qualify as a successful

## system:

## 🡪Robustness: The system should be robust enough to manage and

## handle any invalid input from the users.

## 🡪Reliability: The System must perform banking operations with

## 

## no errors or no discrepancies.

## 

## 🡪Security: System Security is one of the most important

## 

## non-functional requirements.

## 🡪Administrator Cost: The system should have minimum

## Admin/user maintaining cost.

## 🡪Utility and Usability: The system should be user-friendly and

## 

## In easy to control environment for both admin and customer.

## 2.2. Deployment Diagram for eBMS:

## Deployment Diagram gives abstract and structural idea of

## eBMS system and workflow architecture. Diagrams for

## higher level and lower level are as shown as below:

## 🡪eBMS Deployment Diagram- higher level

## Diagram Description automatically generated

## 🡪eBMS Deployment Diagram: Lower Level:

## Diagram Description automatically generated

**2.3. Class ER diagram For eBMS:**

* Class ER diagram will show structural view of system entities and

relationship between the different entities. Here PK denotes

primary key and FK denotes foreign key for references between

tables. Different classes in the diagram are listed below:

* Customer\_types
* Transactions
* Transactions\_types
* Accounts
* Function\_Library
* Account\_types
* Customer
* There are also data types given in the diagram for easy

understanding of every entity mapped on different classes

primary key for every table should be unique and limited to

one primary key per table. Diagram is shown Below:

🡪**Class ER Diagram of eBMS:**

**Diagram, schematic

Description automatically generated**

**2.4. System Architecture Diagram:**

**(ref SRS)**

* System architecture diagram shows the whole system architecture

with showing every functional part of the system. System

architecture diagram is as follows:

**Diagram

Description automatically generated**

* As shown above eBMS system works the functions of every node

Is from user and system input.

## The User will have Web interface and login page to visit the

## 

## eBMS system which he/she can interact with. Initial page is a

## login page for an active user to login with the right credentials.

## If login creds of user are authentic then he/she will be redirected

## to eBMS system home page where he/she can perform actions

## according to his/her requests. All of these actions are limited for

## normal user activities.

## If there are some wrong creds input are given or suspicious

## activities are detected then, warning massage will pop to alert

## user to input correct creds and log will be created for that user,

## 

## and if such activity keeps occurring then system time-out for 40

## mins will be initiated, and security mechanisms will be activated

## while creating logs.

## Authorized user will have access to his/her records and account

## details and statement/transaction history. And rights to fetch

## data from database also, under normal user restrictions he/she

## will be able to perform insert/view/delete operations for new

## entries.

**2.5. Data Flow Diagram:**

* Data Flow diagram shows data flow of the system on every node.

data flow diagrams for every level are shown as below:

**🡪eBMS data flow Diagram: level 0**

**Diagram

Description automatically generated**

**🡪eBMS data flow Diagram for Transaction: Level 2**

**Diagram, schematic

Description automatically generated**

**2.5. Sequence Diagrams for eBMS:**

* The sequence Diagrams shows the interaction logic between the

objects in the system in the time order that the interactions take

place. For eBMS sequence diagram is as follows:

🡪**Sequence Diagram of Transaction:**

**Diagram

Description automatically generated**

**🡪Sequence Diagram of Login:**

**Diagram

Description automatically generated**

**[IMPORTANT NOTE: Here Given Webapp UI example is just for visual representation of eBMS system webpage, real system login page may or may not be the same.]**

**2.6. WebApp UI:**

* Here is a small example given for web app visual presentation of

eBMS login page.

**Graphical user interface, website

Description automatically generated**

**3. Access control and Security:**

* The Access control for the eBanking management system is used

through the capabilities and operation. A capability allows an actor

access to an object of the class. Here is the list:

Login().

Manage Account Menu()

Update\_Account()

Account

Transaction

Manage Transaction()

Select/View Transaction()

Show/Edit/Add Other\_info()

Transaction\_type\_code()

Account\_type\_code()

Customer\_type\_code()

**3.1. Security Goals:**

## All the available data in the database should be protected all the

## time and login credentials stored in the database should be

## encrypted, so that even if breach occurs data should not be

## leaked.

## Every interaction between user and system should happen in

## secure environment and should not be breached.

## The banking management system must be fully and only

## accessible to the authentic users only. Any third party

## software or another unauthorised user should not be able

## to access the system without active user or admin

## permission.

## If any of the security protocols are breached or unauthorised

## entries are logged, legal actions will be taken against such

## activities.

## Database backups and regular software updates, monitoring and

## logging should be on periodic intervals, and should be handled

## by admin or someone who has permission of admin to do this

## actions.