**ABSTRACT**

This project is aimed at developing an Online Banking for customer. The system is an online application that can be accessed throughout the organization and outside as well with proper login provided.

. The database connectivity was planned using the latest “ Database connection” technology provided by Oracle. The authentication and authorization was crosschecked at all the relevant stages.

**Introduction**

**1.1 Overview**

Internet Banking is all about knowing our customer need and provide them with the right service at the right time through right channel 24\*7 day a week.

Being “electronic”, it not only provides its customers with faster and better facilities, it even reduces the manual overhead of accounts maintenance.

**1.2 ABOUT THE PROJECT**

APANA-BANK is one of the most prestigious BANKs .

APANA-BANK, C.P. is affiliated to the Central Board of Bank (CBB), which is the largest educational board in the country. It is recognized by the Department of Education, Govt. of NCT Delhi and the Ministry of HRD, Govt. of India. Over 5000 BANKs in India, with over 80,000 students, are members of the Board.

The BANK is also affiliated to the Indian Public BANKs' Conference (IPSC), and the National Progressive BANKs' Conference (NPSC). The members of these organizations include some of the premier BANKs in the country.

Life at DPSRKP centers on a shared commitment to academic excellence, intellectual growth, art, athletics, high standards of ethical awareness, sportsmanship, and community service. The BANK's traditions and accessibility to a broad curriculum add depth to each student’s life.  
The BANK upholds the founders' commitment to excellence in all fields, with emphasis on its motto Service Before Self.

**1.3 BANK PROFILE:**

APANA-BANK, C.P. is a co-educational day-cum-boarding BANK, with approximately 9,500 customer on its rolls. These children, in the Junior and Senior branches, study in the three different campuses at East of Kailash, Vasant Vihar and C.P.

The BANK is among the most distinguished members of the Ravi Public BANK, C.P.. It is a path breaker in the pursuit of excellence. Its endeavor of integrating quality with quantity is reflected in the pivotal role it has played in the setting up of DPS Vasant Kunj, DPS Faridabad and DPS Manali at the national level. It has also promoted three BANKs abroad in Kuwait, Nepal and Indonesia. As their Linking BANK it also co-ordinates their activities.

The BANK has also extended its expertise further and in collaboration with the Government of Haryana, has taken up 3 BANKs in the under-privileged area of Mewat, to augment and enhance their standards and make them more conducive to teaming.

The BANK considers education to be a life-long process which should have a strong foundation. The goal of the BANK is to inculcate in the customer a love for learning and a desire to excel at every level. The BANK also aims at equipping the customer with the intellectual and practical skills that are necessary to meet the challenges in the future.

To sum up, the mission of APANA-BANK, C.P. “to open doors and open minds” and prepare the ground for the future of the nation.

**1.5 Purpose:**

The Online Banking suite provides a global accounting foundation that provides the all private banks with electronic banking facilities. It allows client of private banks to carry out their day to day banking transactions.

**1.6 Scope:**

The Online Banking project is widely applicable with private banks. It can even be used in industries for their personal transactions (working).

**1.7 Functional components of the project:**

Following are the functional needs of the software:-

1. Customer must have a valid user ID and password to login to the system.

2. After the valid user logs in, the system shows the present balance in that particular account number.

3. Customer can perform transactions like deposit and withdrawal from his account.

4. Proper help to be provided as and when requested by the customer.

**System Analysis**

**Benefits of the system**

Quick, authenticated access to accounts via the desktop.

Easily scalable to grow with changing system requirement.

Enterprise wide access to information.

Improved information security, restricting unauthorized access.

Minimize Storage Space

In manual system, much storage space for data files is required so to overcome this problem, on automated well managed

database is developed for saving storage space. This s/w saves space and stores information efficiently. It ends the burden of having large manual filing storage system.

**Banking System can be used extensively**

Withdrawal of amount by the client.

Deposition of amount by the client.

Faster balance enquiry.

**FEASIBILITY REPORT**

**3.1 Understanding Feasibility**

Feasibility study means the analysis of problem to determine if It can be solved effectively. In other words it is the study of the possibilities of the proposed system it studies the work ability, impact on the organization ability to meet user’s need and efficient use of resources.

Three aspects in which the system has to be feasible are:-

**ECONOMICAL FEASIBILITY:**

The economical analysis checks for the high investment incurred on the system. It evaluates development &

implementing charges for the proposed “Banking Project”. The S/W used for the development is easily available at minimal cost & the database applied is freely available hence it results in low cost implementation.

**Software Requirement & Specification**

**Software Required:**

The project is implemented in ADV JAVA, hence the software’s required in the creation and execution of the project are J2EE ,NETBEANS or Eclipse .

**Technologies and Requriments**

**IDE:**

ECLIPSE NEON 3

**Front End:**

JSP, JDBC, Javascript, AJAX

**Programming Language:**

JAVA

**Back End:**

My SQL Database

**System Design**

**5.1 INTRODUCTION:**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

**5.3 E – R DIAGRAMS:**

* + **The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.**
  + **The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.**
  + **The set of primary components that are identified by the ERD are**

◆ **Data object** ◆ **Relationships**

◆ **Attributes** ◆ **Various types of indicators.**

**The primary purpose of the ERD is to represent data objects and their relationships.**

**5.4 DATA FLOW DIAGRAMS:**

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other

**DFD SYMBOLS:**

In the DFD, there are four symbols

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data

Process that transforms data flow.

Source or Destination of data

Data flow

Data Store

**DATA FLOW**

1. A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The later is usually indicated however by two separate arrows since these happen at different type.
2. A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
3. A data flow cannot go directly back to the same process it leads. There must be atleast one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
4. A Data flow to a data store means update (delete or change).
5. A data Flow from a data store means retrieve or use.

A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

(a) **:**  User Details

(b) **:** Response

(c) **:** Personal details

(d) **:** Reply

(e) **:** Account Transaction entry

(f) **:** Transaction Details

(g) **:** Loan Application

(h) **:** Response

(i) **:**  Feedback

(j) **:**  Reply

(k) **:** Create ,Update ,Reply

(l) **:**  User data

(m)**:** Login Info

(n) **:** Invalid login

(f)

Reply

User

Access

Access

Valid user



Valid user

**Online Banking System**

**Database**

Administrator

(e)

(g)

(h)

(i)

(j)

Access

Registration Info

Valid Administrator

(k)

(l)

Login\_ Info

(m)

(n)

User Details

Reply

(a)

(b)

(c)

(d)

Fig. Level 1 DFD

Registration Info

Verify data

Valid User

Register Info

Update

Login\_Info

Fig . Level 2 DFD process-1

User Data

User Data

CUSTOMER

Personal Details

Account Request

Response

Verified\_ Appl

update

Retrieve

Fig . Level 2 DFD process-2

Valid user

Request for transfer

Valid user

Balance Enquiry

Deposit cash

Other Account

Other Bank

Money\_Transfer

Account\_tab

Branch

Fig . Level 2 DFD process-3

Fig . Level 2 DFD process-4

PROCESS 5

Response

Valid user

Loan\_ application

Sanctioned\_Loan

Response

Update

Response

Loan details

Valid user

Valid user

Valid user

Response

Reply

Feedbacks\_ Reply

Reply

Feed Back

FeedBack\_tab

User Feedback

Feedback

Reply

Fig . Level 2 DFD process-5

Verified

data

Response

User Details

Update Table

Retrieve Data

Login\_Info

New Password

Old Password

New Password

Customer\_ Info

Loan\_ Application

Sanctioned\_ Loan

Update

Customer Feed Back

Retrieve Feedback

Response

Feed Back

Money Transfer \_tab

Account\_ tab

Update Data

Update

Data

Retrieve Data

Confirm Password

User Feedback

Reply

Account\_Requests Tab

Application data

Account

Details

Account details

update

Fig . Level 2 DFD process-6

verify

update

Valid user

Money\_ Transfer

Account\_tab

Account\_tab

Status Info

Fig . Level 3 DFD

**5.5 DATABASE TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Data type** | **Nullable** | **Primary key** |
| ACCOUNTINFO | Number | No | Yes |
| USERNAME | Varchar2 | Yes | No |
| PASSWORD | Varchar2 | Yes | No |
| AMOUNT | Varchar2 | Yes | No |
| ADDRESS | Varchar2 | Yes | No |
| PHONE | Varchar2 | Yes | No |

**SYSTEM TESTING AND IMPLIMENTATION**

**8.1. INTRODUCTION**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

**8.2. STRATEGIC APPROACH TO SOFTWARE TESTING**

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

UNIT TESTING

MODULE TESTING

SUB-SYSTEM TESING

SYSTEM TESTING

ACCEPTANCE TESTING

Component Testing

Integration Testing

User Testing

**8.3. Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

**1. WHITE BOX TESTING**

This type of testing ensures that

* All independent paths have been exercised at least once
* All logical decisions have been exercised on their true and false sides
* All loops are executed at their boundaries and within their operational bounds
* All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

**2. BASIC PATH TESTING**

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

V(G)=E-N+2 or

V(G)=P+1 or

V(G)=Number Of Regions

Where V(G) is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

**3. CONDITIONAL TESTING**

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

**4. DATA FLOW TESTING**

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

**5. LOOP TESTING**

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

* All the loops were tested at their limits, just above them and just below them.
* All the loops were skipped at least once.
* For nested loops test the inner most loop first and then work outwards.
* For concatenated loops the values of dependent loops were set with the help of connected loop.
* Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

# CHAPTER- 9

**SYSTEM SECURITY**

# 9.1. Introduction

The protection of computer based resources that includes hardware, software, data, procedures and people against unauthorized use or natural

Disaster is known as System Security.

System Security can be divided into four related issues:

* Security
* Integrity
* Privacy
* Confidentiality

**SYSTEM SECURITY** refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

**DATA SECURITY** is the protection of data from loss, disclosure, modification and destruction.

**SYSTEM INTEGRITY** refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

**PRIVACY** defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

**CONFIDENTIALITY** is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

## 9.2. SECURITY IN SOFTWARE

System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employees two types of checks and controls:

**CLIENT SIDE VALIDATION**

Various client side validations are used to ensure on the client side that only valid data is entered. Client side validation saves server time and load to handle invalid data. Some checks imposed are:

* VBScript in used to ensure those required fields are filled with suitable data only. Maximum lengths of the fields of the forms are appropriately defined.
* Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.
* Tab-indexes are set according to the need and taking into account the ease of user while working with the system.

**SERVER SIDE VALIDATION**

Some checks cannot be applied at client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed is:

* Server side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
* User is intimating through appropriate messages about the successful operations or exceptions occurring at server side.
* Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled o the server side.
* Using server side validation, constraints on several restricted operations are imposed.

**CHAPTER- 10**