### BANKING SYSTEM

### SYSTEM DESIGN DOCUMENT (SDS)

### TABLE OF CONTENTS 1. INTRODUCTION .................................................................. 2 1.1. PURPOSE OF THE SYSTEM ........................................................2 1.2. DESIGN GOALS ...................................................................2 1.3. DEFINITIONS, ACRONYMS AND ABBREVIATIONS ...............................3 1.4. REFERENCES ......................................................................3 1.5. OVERVIEW.........................................................................4 2. CURRENT SOFTWARE ARCHITECTURE................................. 4 3. PROPOSED SOFTWARE ARCHITECTURE .............................. 4 3.1. OVERVIEW.........................................................................4 3.2. SUBSYSTEM DECOMPOSITION ...................................................5 3.3. HARDWARE/SOFTWARE MAPPING ...............................................6 3.4. PERSISTENT DATA MANAGEMENT................................................8 3.5. ACCESS CONTROL AND SECURITY ...............................................8 3.6. GLOBAL SOFTWARE CONTROL ................................................. 10 3.7. BOUNDARY CONDITIONS ....................................................... 11 4. SUBSYSTEM SERVICES...................................................... 11 5. GLOSSARY ........................................................................ 14

BANKING SYSTEM –

1. Introduction

1.1. Purpose of the system  
The Requesting Bank is a banking institution with services in South  
Florida, with more than 50 branch offices and serving more than two  
hundred thousand clients.  
The bank still uses an obsolete, terminal-oriented banking system that  
has limited the operation expansion and growth, due to  
telecommunication limitations, lack of interactivity between users and  
system, high system maintenance costs, poor or inexistent support  
from some product vendors, as well as a very expensive learning curve  
for the new users.  
The new administration’s strategic plan for the next five years focuses  
on a statewide coverage expansion, but such expansion will be  
unreachable if the bank does not count with a new system strong  
enough to support the expected high volume operations.  
The purpose of this Banking System will reengineer the current  
system, bringing new technology to the existing one, and adding new  
capabilities such as the new internet banking interface for the bank’s  
customers.

1.2. Design goals  
 The design goals represent the desired qualities of Banking System  
and provide a consistent set of criteria that must be considered when  
make design decisions. Based on the nonfunctional requirements, the  
following design goals will have to be achieved in order to qualify the  
system as successful:  
Dependability criteria:  
• Robustness: The system has to be robust enough to manage any  
invalid input from the users.  
• Reliability: The system has to perform the Banking operations  
with no errors or discrepancies.  
• Security: The system security is one of the most important  
nonfunctional requirements.  
BANKING SYSTEM – System Design Document 3  
Cost criteria:  
• Deployment cost: The system has to be easy enough to have a  
cheap training cost.  
• Upgrade cost: The new system implementation has to deal with  
the former legacy system, in order to achieve a cost effective  
database migration.  
• Administration cost: The system must have a low administration  
cost.  
Maintenance criteria:  
• Extensibility: The system must support that new functionalities  
can be added.  
• Modifiability: The system has to be highly modifiable, so the  
Information Technology Department of the bank can maintain it.  
• Readability: The system has to be readable enough to assure its  
modifiability.  
• Traceability of requirements: The code should be easy to be  
mapped to specific requirements.  
End user criteria:  
• Utility: The system has been conceived specifically to support  
the bank employee’s work. The new system will also support the  
customers operations through its web-banking interface.  
• Usability: The system will be designed in a user-friendly fashion,  
both for bank employees and customers.

1.3. Definitions, acronyms, and abbreviations  
Bank Teller: The bank employee attending behind the windows, in  
charge of attending the customer’s transactions, such as deposits and  
withdrawals.  
Bank Officer: The bank employee in charge of the direct customer  
service. Her duties include, but are not limited to, opening new  
accounts, as well as updating and closing existing accounts.  
Customer: The account holder who uses the bank services.  
System Administrator: The system maintainer, in charge of creating  
usernames for bank employees.

1.4. References  
• Problem Statement  
• Requirements Analysis Document

BANKING SYSTEM – System Design Document

1.5. Overview  
This document is organized as follows: Section 2 presents a brief  
overview of the current software architecture, describing the current  
situation of the banking system. Section 3 presents the proposed  
software architecture, focusing on the subsystem decomposition,  
hardware and software mapping and the persistent data management.  
Section 4 presents the subsystems services.

2.Current Software architecture  
The current system is based on a Mainframe with several terminals  
attached to it. They have an AS400 mainframe that provides all the  
services to the branch office’s terminals.  
This typical Client/Server architecture has limited the bank’s  
expansion, due to the high costs of operations, the lack of hardware  
and software maintenance in some cases, and the difficulty of  
functionality of the system.  
Other products on the market implement all their operations and  
functionalities over Client/Server architecture, based on TCP/IP  
networks, using either Windows Applications or web-based interfaces  
to communicate the clients with the main hosts.

3. Proposed Software architecture  
3.1. Overview  
This project has the goal of implementing a reengineering to the  
current banking system. It will be implemented using the Client/Server  
architecture, but will renew the technology of the current system, by  
adding new functionalities and improving the existing one. The  
improvements will be basically the integration of more modern  
technologies, such as a TCP/IP network to interconnect all the branch  
offices to the host servers.  
All the applications running in the branch offices will be implemented  
with web-based front ends, to improve the usability of the system, as  
well as to optimize the computational resources of the main hosts. At  
BANKING SYSTEM – System Design Document 5  
the same time, this architecture allows the system to scale the server  
configuration according to the needs of the bank.

3.2. Subsystem decomposition  
During the subsystem decomposition of the Banking System we divide  
the system into smaller subsystems with a strong coherence. The  
different subsystems should have a loose coupling.  
The system will be decomposed based on the use cases and the  
different actors we have defined.  
The decomposition shows the existence of the following subsystems:  
• User Management Subsystem  
• Account Management Subsystem  
• Transaction Management Subsystem  
• Storage Subsystem  
• Database Subsystem  
The Database subsystem will be implemented by the Relational  
Database Management System (RDBMS) used to store the persistent  
data. The Storage subsystem will encapsulate the database, providing  
a common interface to the other three high-level subsystems.  
AccountManagement

UserManagement

TransactionManagement  
Storage  
Database

Figure 1: Subsystem Decomposition  
BANKING SYSTEM – System Design Document 6  
user management  
AccountManagement  
transaction management  
Storage  
Database  
Person  
SysAdmin User  
BankOfficer BankTeller  
Transaction  
Account  
Customer

Figure 2: Subsystem Decomposition with classes  
3.3. Hardware/Software mapping  
The banking system will be web-based, both for the banking users and  
for the customer access through the internet banking platform. The  
the functionality will all be performed in the main host and will be  
accessed through TCP/IP networks by all the users.  
The branch employees will access the system through the bank’s  
TCP/IP private network, and the customers will access through the  
World Wide Web.  
The proposed main host will be a Sun Microsystems Sun Fire E2900  
Server. It has been chosen due to its ability to scale up to 12 servers,  
which can be reached in the next 5 years, according to the expansion  
goals of the bank.

BANKING SYSTEM – System Design Document 7  
The system will run over the UNIX operating system, specifically over  
Solaris Version 3.2.5. The web server will run over Apache Web Server  
Version 1.3.33.  
The programming language used to develop this product will be Java,  
using JSP for the web interface.  
We have selected MySQL Version 4.1 as the Database Management  
System, and we will use JDBC drivers to connect the Java components  
to it.  
The Banking System consists of three independent components:  
WebBrowser, BankSystemServer and DatabaseServer.  
The following UML deployment diagram illustrates the hardware/  
software mapping for Banking System.  
aP C :P C  
:U n ixH o st  
:U n ixH o st  
:E x p lo re  
:U se rM angem e n t  
:A c c o u n tM anagem e n t  
:T ra n sa ctionM angem e n t  
:S to ra g e  
:D a ta b a se  
Figure 3: Package and Deployment Diagram Deleted: <sp>  
BANKING SYSTEM – System Design Document 8  
3.4. Persistent data management  
Based on the Requirement Analysis Document, we have identified the  
following Entity objects as persistent data objects:  
• Person: Information about all the system users. It inherits into  
System Administrator and User; at the same time, User inherits  
into BankOfficer, BankTeller and Customer.  
• Account: Information about the bank accounts. It includes the  
account holder’s information, as well as balances, average  
balances, and so on.  
• Transaction: Detailed information about the bank transactions  
performed against the accounts. It includes the type of  
transaction, date and time, amount, ending balance, etc.  
This persistent information will be stored in a Relational Database  
Management Subsystem (RDBMS). We have selected MySQL as  
RDBMS, due to its versatility, high performance and integration with  
the other products that constitute the new platform.  
3.5. Access control and security  
The access control for the Banking system is implemented through the  
capabilities list. This representation comes up to be more compact and  
efficient for the System. A capability associates a (class, operation)  
pair with an actor. A capability allows an actor access to an object of  
the class. Denying a capability is equivalent to denying access.  
Bank Officer CAPABILITY LIST  
CLASS OPERATION  
Function Select menu ManageAccounts(),  
updateLogin(),  
Login().  
Manage Account Menu Create(), createAccount(),  
updateAccount(), closeAccount().  
Create Account Form Create(), submit().  
Update Account Form Create(), submit().  
Close Account Form Create(), submit().  
Account createAccount(),  
updateAccount(),  
closeAccount(),  
BANKING SYSTEM – System Design Document 9  
Bank Teller CAPABILITY LIST  
Customer CAPABILITY LIST  
CLASS OPERATION  
Function Select menu ManageTransactions(),  
updateLogin(),  
Deposit(), transfer(), withdraw().  
Manage Account Report Create(),  
Print(),  
dismiss().  
Transaction Select Menu Create().  
Withdraw Form Create(), submit().  
Deposit Form Create(), submit().  
Transfer Form Create(), submit().  
Manage Transactions Report Create(),  
Print(),  
dismiss().  
Transaction Create().  
Person Update()  
Login form Create(), submit().  
Update Login Form Create(), submit().  
CLASS OPERATION  
Function Select menu ManageTransactions(),  
updateLogin(),  
Login().  
Account Deposit(),  
Transfer(),  
Withdraw().  
Transaction Select Menu Create().  
Withdraw Form Create(), submit().  
Deposit Form Create(), submit().  
Transfer Form Create(), submit().  
Manage Transactions Report Create(),  
Print(),  
dismiss().  
Transaction Create().  
Person Update()  
Login form Create(), submit().  
Update Login Form Create(), submit().  
BANKING SYSTEM – System Design Document 10  
Login().  
Account Deposit(),  
Transfer(),  
Withdraw().  
Transaction Select Menu Create().  
Withdraw Form Create(), submit().  
Deposit Form Create(), submit().  
Transfer Form Create(), submit().  
Manage Transactions Report Create(),  
Print(),  
dismiss().  
Transaction Create().  
Person  
Create Customer Login Form  
Update()  
Login form Create(), submit().  
Update Login Form Create(), submit().  
System Administrator CAPABILITY LIST  
CLASS OPERATION  
Function Select menu createLogin(), updateLogin(),  
Login().  
Create Login Menu createCustomerLogin(),  
createEmployeeLogin().  
Person  
Create Customer Login Form  
Create(), update().  
Create(), submit().  
Create Employee Login Form Create(), submit().  
Login form Create(), submit().  
Create Login Report Create(), print(), dismiss().  
Update Login Form Create(), submit().  
3.6. Global software control  
Procedure-driven control would be the one that will be suitable for our  
system. In our system, operations wait for user input whenever they  
need data from either a bank employee or a customer. For example,  
since our project is a web-based project, the web server waits for  
requests from the web browser. Upon receipt of a request, the web  
server processes and dispatches it to the appropriate web page, thus  
resulting in an event-based control flow.  
BANKING SYSTEM – System Design Document 11  
3.7. Boundary conditions  
Banking system includes four run-time components: the WebBrower,  
the BankSystemSever (which include the subsystems  
UserManagement, AccountManagement, TransactionManagement), the  
Storage and for the second prototype, DatabaseSever. The WebBrower  
and DatabaseSever are off-the-shelf components and are started and  
shut down individually. The Storage is started and shut down by the  
BankSystemSever.  
The starting, shutdown and installing of the Banking System define the  
boundary conditions.  
• Installing. Since Banking System is web-based application, it  
does not need explicit installation execution.  
• Starting.  
• Shutdown.  
4.Subsystem services  
USER MANAGEMENT SUBSYSTEM:  
This subsystem is responsible for managing different users of the  
system by taking care of the login information of different users. It  
provides functions for Creating Login, Updating Login and Log-in. It  
manages the usernames and passwords of all users of the system for  
security purposes. This subsystem uses the services of storage  
subsystem to store and retrieve login information. System  
administrator and all users of the system communicate with this  
subsystem.  
Operations provided by this subsystem are:  
• createLogin()  
• updateLogin()  
• login()  
The createLogin() and updateLogin() services of this subsystem  
implement the print() service, to allow the customer to have a backup  
of her operation.  
ACCOUNT MANAGEMENT SUBSYSTEM:  
BANKING SYSTEM – System Design Document 12  
This subsystem is responsible for managing user accounts. It provides  
functions for Opening an account, Updating an account and Closing  
account. Bank Officer is the only actor who communicates with this  
subsystem. This subsystem uses login services of the User  
Management subsystem for authenticating the Bank Officer and also  
uses the storage subsystem for storing account’s information.  
Operations provided by this subsystem are:  
• createAccount()  
• updateAccount()  
• closeAccount()  
All the methods of this subsystem implement the print() service, to  
allow the customer to have a backup of her operation.  
TRANSACTION MANAGEMENT SUBSYSTEM:  
This subsystem is responsible for managing the transactions of  
accounts. This subsystem provides all functions for managing variety  
of transactions like deposit, withdrawal, transfer of funds, checking  
monthly balance and transactions. Bank Officer, Bank teller and  
customer are the actors who communicate with this subsystem.  
This subsystem also communicates with the storage subsystem for  
storing critical transaction data and uses the services of the user  
management subsystem for authenticating the users who perform  
transactions.  
Operations provided by this subsystem are:  
• deposit()  
• withdraw()  
• transfer()  
• checkTransactions()  
• checkBalance()  
All the methods of this subsystem implement the print() service, to  
allow the customer to have a backup of her operation.  
STORAGE SUBSYSTEM:  
This subsystem is responsible for specifying a common interface for  
the above subsystems for managing data. This subsystem is  
responsible for getting system-related data from different subsystems  
and issuing DBMS-specific calls for information storage and retrieval.  
BANKING SYSTEM – System Design Document 13  
Even if the DBMS changes, only this subsystem needs to be changed,  
without affecting other subsystems that access this subsystem.  
DATABASE SUBSYSTEM:  
This subsystem is a typical off-the-shelf component that is responsible  
for storage and retrieval of system data. A Database management  
system (DBMS) provides all the necessary functions for performing the  
tasks.  
BANKING SYSTEM – System Design Document 14  
Glossary  
Customer: A customer is the term used for referring a Bank customer  
who utilizes the Bank services by opening an account, depositing  
money, withdrawing and earning interest if any.  
Bank Officer: A Bank officer is the term used to refer to the officer of  
the Bank who takes care of opening the accounts, creating the logins  
for Bank Employees and closing accounts if any. He also can do  
transactions if necessary.  
Bank Teller: A Bank Teller is the term used to refer the teller of the  
system, who performs all the customer account transactions. He is the  
one who makes a deposit, withdrawal or checks the account’s balance.  
Administrator: An administrator is the term referring to a bank  
system maintainer who is in charge of creating usernames for bank  
employees and system maintenance.