Bankbazaar - Documents

Software Engineering (Lovely Professional University)

-

CASE STUDY

ON

BANKBAZAAR.COM

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

-

DECLARATION STATEMENT

I hereby declare that the case study entitled " BANKBAZAAR.COM"submitted at Lovely Professional University, Phagwara, Punjab is an authentic work and has not been submitted elsewhere.

I understand that the work presented herewith is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this case study represents authentic and honest effort conducted, in its entirety, by me. I am fully responsible for the contents of my case study report.

Signature of Candidate

Andhavarapu akhil

R.No: B35

Acknowledgement

I have taken efforts in this case study. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Ms. Shailja sharma Mam for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.
I would like to express my gratitude towards my parents & friends for their kind co-operation and encouragement which help me in completion of this project.
I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.
My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

-

Table of Contents

LIST OF FIGURES

FIGURE NO.	FIGURE DESCRIPTION	PAGE NO.
Figure 1.1		8

-

1.Introduction

1.1 Purpose

This document describe the software requirement and specification for Consumer Finance company (Bank Bazaar.com)

1.2 Document Conventions : font: TNR 12

1.3 Intended Audience and Reading Suggestion

The document is intended for all the stakeholders customer and the developer (designers, testers, maintainers). The reader is assumed to have basic knowledge of banking accounts, loan services and bak transactions. Knowledge and understanding of UML diagrams is also required.

1.4 Definitions, Abbrevations

1.4.1 Definitions

- **Bank**

A financial institution that holds accounts for customers and that issues cash cards authorizing access to accounts over the ATM network.

- **Bank computer**

The computer owned by a bank that interfaces with the ATM network and the bank's own cashier stations. A bank may actually have its own internal network of computers to process accounts, but we are only concerned with the one that interacts with the network.

- **Customer**

The holder of one or more accounts in a bank. A customer can consist of one or more persons or corporations, the correspondence is not relevant to this problem. The same person holding an account at a different bank is considered a different customer.

- **Transaction**

A single integral request for operations on the accounts of a single customer. We only specified that ATMs must dispense cash, but we should not preclude the possibility of printing checks or accepting cash or checks. We may also want to provide the flexibility to operate on accounts of different customers, although it is not required yet. The different operations must balance properly.

-

- **Loan**

A thing that is borrowed, especially a sum of money that is expected to be paid back with interest(A loan is the act of giving money, property or other material goods to another party in exchange for future repayment of the principal amount along with interest or other finance charges).

- **Insurance**

An arrangement by which a company or the state undertakes to provide a guarantee of compensation for specified loss, damage, illness, or death in return for payment of a specified premium.

- **Cash Card**

A card assigned to a bank customer that authorizes access to accounts using an ATM Machine. Each card contains a bank code and a card number, coded in accordance with national standards on credit cards and cash cards. The bank code uniquely identifies the bank within the consortium. The card number determines the accounts that the card can access. A card does not necessarily access all of a customer's accounts. Each cash card is owned by a single customer, but multiple copies of it may exist, so the possibility of simultaneous use of the same card from different machines must be considered.

- **Mutual Fund**

An investment programme funded by shareholders that trades in diversified holdings and is professionally managed.

1.4.2 Abbrevations

Throughout this document following abbreviations are used :

: input provided by the user at the run time .
i/p : output received on screen
o/p

- u_name : username entered for login process
- u_pass : password entered for login

success : return authenticated
: return Invalid usename/password
failure : return output
result : EMI Value
E : Principal value
P R : Rate of Interest
T SI : Time in months
: Simple Interest
CI : Compound Interest

-

1.5 Project Scope

The software allows you to compare policies , loans online among different vendors in the market , and gives you the most appropriate results . It also offers you to apply online for any kind of loan to any vendor . It tells you about the eligibility criteria for all the services which you want to apply for.It provide you all the information related banks at your doorsteps . The software identifies its customer by a username and password which you get at the time of online registration of this software. It analyse the i/p provided by user , search for it and provide the most relevant details . The software must handle multiple requests from the same user and keep the record correctly . It gives a very useful and wonderful feature “Track Application” .After applying for any service user can get the status of its application at any point of time .

2. Overall Description

2.1 Product Perspective

The software work independently has no collaboration with any public / private vendor . It works on the basic principle of Data Analysis . The whole platform operate on the World Wide Web. The data fetched form the survey , stores in the Database and respond back to the user when query is fired .

Software Interface : The software operates on World Wide Web and runs on a web browsers (Chrome , Firefox , Internet Explorer 9 or above etc).

Hardware Interface : The software can run on any internet enabled machine with browsers installed on it .(Browsers : It is a software that run web applications)

User Interfaces

Customer : The customer user interface should be intuitive, such that 99.9% of all new users are able to get all the information without any assistance.

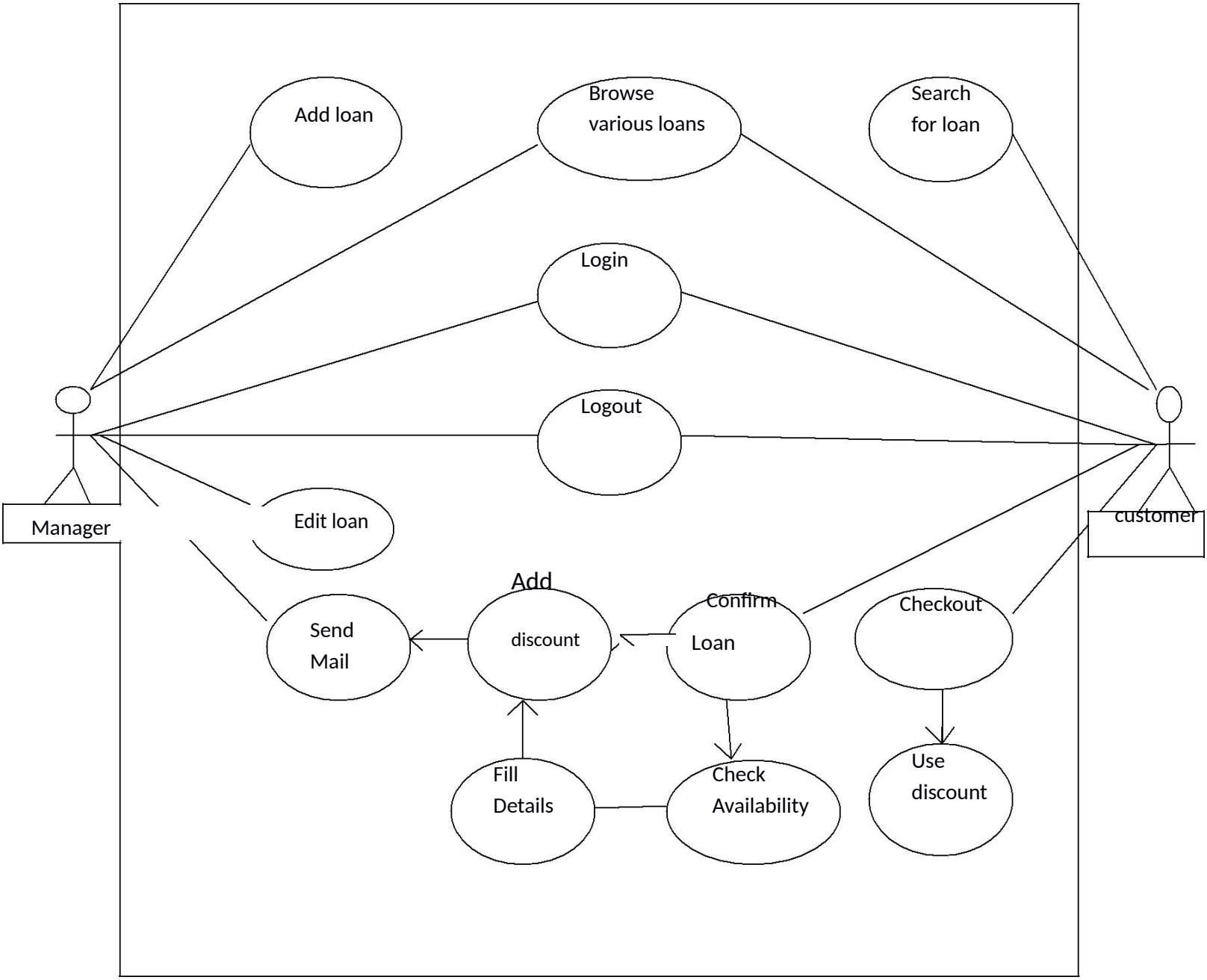
Administrator : The maintainer is responsible for adding new features to the software and servicing existing accounts present on software . A maintainer should update the software regularly for better customer satisfaction .

2.2 Product Features

The software should work 24 hrs for better satisfaction . The software identifies a customer by a username and password. Here username can be anything like : mobile number or any social id . It collects information about various bank services (e.g., Loans, Insurances, Policies , bill payment), communicates the transaction information to the customer’s bank, and dispenses cash to the customer. The software provide their own software for their own computers. The software requires appropriate record keeping and security provisions. The software must handle concurrent accesses to the same account correctly and should be able to handle multiple search queries from multiple user at a same time .

-

The software provides the feature of comparing bank loans , bank policies , investment schemes , etc with respect to customer choice .., It analyze public reviews , different polices ,schemes over the internet . It makes u aware about the terms and conditions for particular scheme , eligibility criteria and all the required documents to apply for any service . It is a software which provide hassle free banking from any part of the world to any kind of people i.e even neophytes can access this software and get aware from all the banking processes .



3 User Class and Prespectives

Characterstics : There are various uses of this software ..

At first glance, price-comparison websites are an example of capitalism at its best. Savvy consumers can use them to hunt down the best available deal for insurance, loan and lifetime policies Firms providing such items, terrified of losing customers, feel an obligation to improve their offerings all the time. But recent theory and practice suggest the reality is more complex: comparison sites are simultaneously friends and foes of competition..No need to stand in large queues and wait for turn .

Users: are simply members of the general public with no special training.

Administrators: must be experienced network administrators, to be able to update new features to the system

2.4 Operating Enviornment

The hardware, software and technology used should have following specifications:

- Ability to maintain the confidientiality
- Ability to differntiate the currency
- Ability to validate user
- Search result must be available to user within notime
- Should not be distractive
- Should not contain irrelevant content
- User friendly
- Regular update
- Most secure
- Resources must be equally available to all users
- Credentials must not be shared
- Search information must not be disclosed

2.5 Design and Implementation Constraints

- Login
- Account Info / Account Session

Validate User Account

- Firstly validate that username must exsist in the database
- Validate that username must be entered valid
- Validate that password must be valid and consistent
- Validate that username/password must not be blank .

Validate Account Info

-

- Validate that if account is not expired
- Continously validate the session of user .

2.6 Assumptions and Dependencies

- Hardware never fails
- Software is unbreachable
- Limited number of transactions per day (sufficient paper for receipts)
- Limited amount of comparisons per day (sufficient information)
- Contain enough information to compare products

3 . Specific Requirments

3.1 Functional Requirments

Functional requirements of this software are organised in very simple and easy way . Need to pass the value at the run time . All the processes are done dynamically .

Functional Requirements 1

- **Description :** Initial Display (Home Screen)
- **Input :** Select the parameters form dropdown Lists and Search (without Login)
- **Processing :** Execute a query at the backend
- **Output :** Display the result .
- **Authorization :** NONE

Functional Requirements 2

- **Description :** Initial Display (Home Screen)
- **Input :** Enter u_name and u_pass
- **Processing :** Check if the username and password entered is valid or not. If valid result=success else result = failure
- **Output :** Display the result .
- **Authorization :** starts after the customer entered the details

Functional Requirements 3

- **Description :** If the result is failure
- **Output :** Home Screen with error prompt .

Functional Requirements 4

-

- **Description** **:** If the result is success
- **Output** **:** Display the user Home Screen

Functional Requirements 5

- **Description** **:** User screen
- **Input** **:** Click on the Loans and choose relevant choice(ex : Education loan)
- **Output** **:** Related window

Functional Requirements 6

- **Description** **:** Education Loan screen
- **Input** **:** Enter degree , country , course duration , College name, etc
- **Processing** : Query will be fired at backend and all the details according to filled data will be shown to the user .
- **Output** **:** User get the response related to loan vendors .
- **Authorization :** Check whether all the fields are correctly filled or not ….if not prompt an error to the user to refill the form .

Functional Requirements 7

- **Description** **:** Two wheeler Loan screen
- **Processing** : Query will be fired at backend and all the details according to filled data will be shown to the user .
- **Output :** User get the brief detail on two wheeler loan and user reviews also .The windows shows result related to eligibility criteria , loan comparison , user details , rate of Interest and public discussions about the loan schemes .etc

Rest all others have the same features as two wheeler loan . Home loan , car loan , used car loan , personal loan etc .

Functional Requirements 8

- **D cription Output**
- **e**
- **s**

-

: : Click on Insurance and then Health Insuarance
Health Insurance screen will be displayed.

Functional Requirements 9

- **Description :** Health screen window
- **Input :** Enter check my eligibility , members to insure , age,
- **Processing** : query will be processed
- **Output :** result will be displayed to you .

Authorization : All the details must be filled …

Functional Requirements 10

- **Description :** Fixed Deposit screen
- **Input :** Enter age , Nationality ,Annual income , Employed/Not Employed
- **Processing** : query will be processed
- **Output :** result with all comparisons will be displayed .
- **Authorization :** All the field must be filled

4. External Interface Requirments

4.1 User Interface

The customer user interface should be intuitive, such that almost all the new users are able to complete their analysis without any assistance .

- Ability to read the database
- Ability to differtiate between currency
- User friendly for convenience
- Continuous power supply
- Ability to connect to bankend database
- Ability to take input from user

-

- Ability to validate user

4.2 Hardware Interface

-

- System with Internet facility
- System must have browser installed on it

5. Other Non-Functional Requirements

5.1 Performance Requirements

- It must be able to perform in adverse conditions like high/low temperature etc.
- Uninterrupted Interrupted connections
- High data transfer rate

5.2 Safety Requirments

- Data must be consistent .
- Data must be safe from all physical menaces like : steal,theft,etc
- Database must follow AAA rules of security
- Intergrity must be maintained
- There must be an emergency back up of data in case of system failure
- Database must be divided in fragments
- All the open protocols , ports must be kept closed from being Hacked .
- It should not respond redirections
- It must contain cache.
- Avoid traffic collision
- Platform swap must not be there
- Load Balancer should be there
- Protected with SSL layer
- Should be under HTTPs protocol
- Web Crawlers must be there for predictions .
- Database must be properly configured
- Database must not respond to irrelevant SQL Injections .

5.3 Security Requirements

- Users accessibility is censured in all the ways
- Users are advised to change their password on first use
- Users are advised not to tell their password to anyone
- The maximum number of attempts to enter paassword will be three

5.4 Software Quality Analysis

- **Availability**
- **Security**
- **Maintainability**

5.4.1 Availability: The software and its resources must be available to the user at each and every point of time .

-

5.4.2 Security: The software must be secured enough to keep user's details confidentiall .

5.4.3 Maintainability: The software must be maintained properly in order to avoid inconvenience. to the user .

6 . Other Requirements

6.1 Data Base

The software must be able to use several data formats according to the data formats that are provided by the data bases of different banks. A transaction should have all the properties of a data base transaction (Atomicity, Consistency, Isolation, Durability).

Atomicity

This property states that a transaction must be treated as an atomic unit, that is, either all of its operations are executed or none. There must be no state in a database where a transaction is left partially completed. States should be defined either before the execution of the transaction or after the execution/abortion/failure of the transaction.

Consistency

The database must remain in a consistent state after any transaction. No transaction should have any adverse effect on the data residing in the database. If the database was in a consistent state before the execution of a transaction, it must remain consistent after the execution of the transaction as well.

Durability

The database should be durable enough to hold all its latest updates even if the system fails or restarts. If a transaction updates a chunk of data in a database and commits, then the database will hold the modified data. If a transaction commits but the system fails before the data could be written on to the disk, then that data will be updated once the system springs back into action.

Isolation

In a database system where more than one transaction are being executed simultaneously and in parallel, the property of isolation states that all the transactions will be carried out and executed as if it is the only transaction in the system. No transaction will affect the existence of any other transaction.