|  |  |  |
| --- | --- | --- |
| **C:\Users\Yash\Desktop\eclipse\Logo.png** | **Devil’s Vault** | **C:\Users\Yash\Desktop\eclipse\Logo.png** |

**Online Banking System**

**Submitted by:** Yash Aggarwal

041396403115

**Project link:** github.com/Yash773/SpringMVC-OnlineBankingSystem

**DECLARATION BY STUDENT**

I, **Yash Aggarwal** student of B. Tech in the institute of **Maharaja Agrasen Institute of technology**, for the **session (2015-2019)**, hereby declare that the project entitled **“Online Banking System”** has been completed by me at **HCL Info system Ltd., Noida.**

The work reported in this Project is the result of my own efforts.

**DATE:**

**PLACE:**

**(Yash Aggarwal)**

**CERTIFICATE FROM ORGANISATION**

**Acknowledgement**

It is not until you undertake a project like this one that you realize how massive the effort it really is, or how much you must rely upon the selfless efforts and goodwill of others. There are many who helped me with this project and I want to thank them all from the core of my Heart.

The training opportunity I had with **HCL Infosystem Ltd., Noida** was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this training period. I express my deepest thanks to **Miss Geetika Gupta** for taking part in useful decision & giving necessary advices and guidance and arranged all facilities to make life easier. I choose this moment to acknowledge his/her contribution gratefully.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

Sincerely,

**(Yash Aggarwal)**

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Topic** | **Page No.** |
| 1 | Declaration | 2 |
| 2 | Certificate | 3 |
| 3 | Acknowledgement | 4 |
| 4 | Chapter-1: Introduction | 6 |
| 5 | Chapter-2: Java | 8 |
| 6 | Chapter-3: Spring | 9 |
| 7 | Chapter-3: Project Monitoring System | 10 |
| 8 | Chapter-4: System Study | 14 |
| 9 | Chapter-5: Introduction To Project | 18 |
| 10 | Chapter-6: System Analysis | 22 |
| 11 | Chapter-7: Software Design | 32 |
| 12 | Chapter-8: Coding | 43 |
| 13 | Chapter-9-System Testing | 48 |
| 14 | Chapter-10: Conclusion | 52 |
| 15 | Bibliography | 53 |

**Chapter 1: Introduction**

**About Organisation:**

HCL Technologies Limited (Hindustan Computers Limited) is an Indian multinational IT services company, headquartered in Noida, Uttar Pradesh, India.

It is subsidiary of HCL Enterprise. Originally a research and development division of HCL, it emerged as an independent company in 1991 when HCL ventured into the software services business.

HCL technologies help global enterprises reimagine their business for the digital image.

HCL Technologies offers services including IT consulting, enterprise transformation, remote infrastructure management, engineering and R&D and business process outsourcing (BPO).

HCL also provide services such as Cyber security and Digitals and Analytics.

The company has offices in 34 countries including the United States, France, Germany and the United Kingdom.

It operates across sectors including aerospace and defence, automotive, consumer electronics, energy and utilities, financial services, government, industrial manufacturing, media and entertainment, public services, retail and consumers, server and storage, transportation and travel.

HCL is among the top 20 largest publicly traded companies in India with a market capitalisation of $18.7 billion.

HCL Technologies is a next-generation global technology company that helps enterprises reimagine their businesses for the digital age. Our technology products, services, and engineering are built on four decades of innovation, with a world-renowned management philosophy, a strong culture of invention and risk-taking, and a relentless focus on customer relationships. We offer an integrated portfolio of products, solutions, services, and IP through our Mode 1-2-3 strategy, built around digital, IoT, cloud, automation, cyber security, analytics, infrastructure management, and engineering services, among others. With a worldwide network of R&D, innovation labs and delivery centres, and 120,000+ ‘Ideapreneurs’ working in 39 countries, HCL serves leading enterprises across key industries, including 250 of the Fortune 500 and 650 of the Global 2000.

Businesses across industries stand at an inflection point today. Far-reaching disruption fuelled by technologies like digitalization, analytics, cloud, IoT, and automation means these technologies lay at the core of any enterprise that is trying to reinvent itself. As the demand for these services increase, HCL Technologies is accelerating its evolution into a next-generation technology company, driven by a unique business model, Mode 1-2-3, and many unmatched core competencies which bring this model to life.

The company’s DNA of grassroots innovation, its ingrained culture of co-innovation, and its tradition of going far beyond what is expected, to create customer value, puts HCL on a different plane from its competition. This gives it a distinct advantage in creating value for businesses in the digital and connected world.

**Chapter 2: What is Java?**

JAVA was developed by Sun Microsystems Inc. in 1991, later acquired by Oracle Corporation. It was developed by James Gosling and Patrick Naught on. It is a simple programming language. Writing, compiling and debugging a program is easy in java. It helps to create modular programs and reusable code.

Java is defined by a specification and consists of a programming language, a compiler, core libraries and a runtime (Java virtual machine) The Java runtime allows software developers to write program code in other languages than the Java programming language which still runs on the Java virtual machine. The Java platform is usually associated with the Java virtual machine and the Java core libraries**.**

**Main Features of Java**

**Web Application** - Java is used to create server-side web applications. Currently, Servlet, JSP, Struts, JSF, etc. technologies are used.

**Standalone Application** - It is also known as the desktop application or window-based application. An application that we need to install on every machine or server such as media player, antivirus, etc. AWT and Swing are used in java for creating standalone applications.

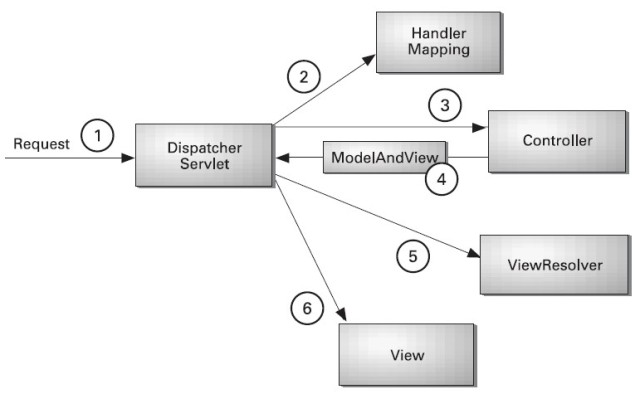
**Enterprise Application** - An application that is distributed in nature, such as banking applications, etc. It has the advantage of the high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.

**Mobile Application** - Java is used to create application software for mobile devices. Currently, Java ME is used for building applications for small devices, and also Java is a programming language for Google Android application development.

**Chapter 3: What is Spring?**

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to, or even replacement for the Enterprise JavaBeans (EJB) model. The Spring Framework is open source. It uses Model–view–controller to design and link web pages (called views) with logic controllers.

Model–view–controller is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user. The MVC design pattern decouples these major components allowing for efficient code reuse and parallel development. Traditionally used for desktop graphical user interfaces (GUIs), this architecture has become popular for designing web applications and even mobile, desktop and other clients. Popular programming languages like Java, C#, Ruby, PHP have MVC frameworks that are used in web application development straight out of the box

. 

Spring is an open source framework created to address the complexity of enterprise application development. One of important advantage of spring framework is its layered architecture, which allows you select component you want while others are ignored. The spring framework is both comprehensive and modular. It is an ideal framework for test driven development. Spring’s main aim is to make J2EE easier to use and promote good programming practice. It does this by enabling a POJO-based programming model that is applicable in a wide range of environments.

Chapter 4: Project Monitoring System

**Gantt chart:-**

A Gantt chart is a horizontal bar chart developed as a production control tool in 1917 by Henry L. Gantt, an American engineer and social scientist. Frequently used in project management, a Gantt chart provides a graphical illustration of a schedule that helps to plan, coordinate, and track specific tasks in a project.

A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (for example, days, weeks, or months) and a vertical axis representing the tasks that make up the project (for example, if the project is outfitting your computer with new software, the major tasks involved might be: conduct research, choose software, install software). Horizontal bars of varying lengths represent the sequences, timing, and time span for each task. Using the same example, you would put "conduct research" at the top of the vertical axis and draw a bar on the graph that represents the amount of time you expect to spend on the research, and then enter the other tasks below the first one and representative bars at the points in time when you expect to undertake them. The bar spans may overlap, as, for example, you may conduct research and choose software during the same time span. As the project progresses, secondary bars, arrowheads, or darkened bars may be added to indicate completed tasks, or the portions of tasks that have been completed. A vertical line is used to represent the report date.

**When to Use**

Gantt charts can be used in managing projects of all sizes and types. These charts are utilized in several industries and for a range of projects, such as building dams, bridges and highways, software development, and development of other goods and services. Project management tools, such as Microsoft Visio, Project, SharePoint and Excel, or specialized software, such as Gantto or Matchware, can help in designing Gantt charts.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Research Field** | **Duration** | | | | | |
| **Market Research** |  |  |  |  |  |  |
| **Define Specifications** |  |  |  |  |  |  |
| **Overall Architecture** |  |  |  |  |  |  |
| **Project Planning** |  |  |  |  |  |  |
| **Detail Design** |  |  |  |  |  |  |
| **Software Development** |  |  |  |  |  |  |
| **Test Plan** |  |  |  |  |  |  |
| **Testing and QA** |  |  |  |  |  |  |
| **Documentation** |  |  |  |  |  |  |
| **Duration** | **Week : 1** | **Week : 2** | **Week : 3** | **Week : 4** | **Week : 5** | **Week : 6** |

**Pert Chart**

A PERT chart is a project management tool used to schedule, organize, and coordinate tasks within a project. PERT stands for Program Evaluation Review Technique, a methodology developed by the U.S. Navy in the 1950s to manage the Polaris submarine missile program. A similar methodology, the Critical Path Method (CPM) was developed for project management in the private sector at about the same time.

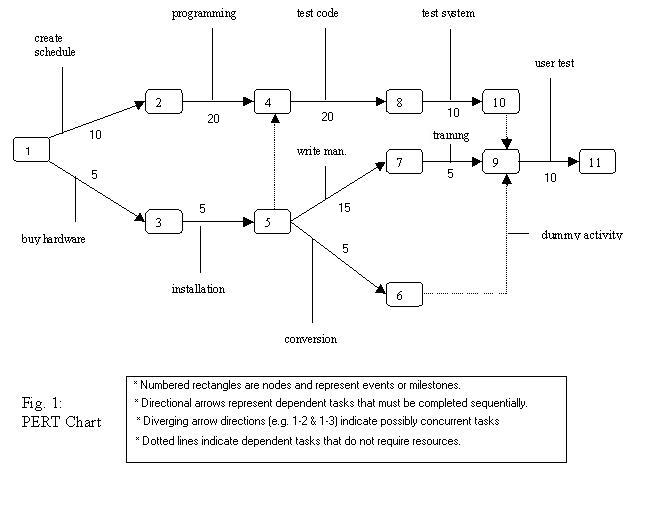
A PERT chart presents a graphic illustration of a project as a network diagram consisting of numbered nodes (either circles or rectangles) representing events, or milestones in the project linked by labelled vectors (directional lines) representing tasks in the project. The direction of the arrows on the lines indicates the sequence of tasks. In the diagram, for example, the tasks between nodes 1, 2, 4, 8, and 10 must be completed in sequence. These are called dependent or serial tasks. The tasks between nodes 1 and 2 and nodes 1 and 3 are not dependent on the completion of one to start the other and can be undertaken simultaneously. These tasks are called parallel or concurrent tasks. Tasks that must be completed in sequence but that don't require resources or completion time are considered to have event dependency. These are represented by dotted lines with arrows and are called dummy activities. For example, the dashed arrow linking nodes 6 and 9 indicates that the system files must be converted before the user test can take place, but that the resources and time required to prepare for the user test (writing the user manual and user training) are on another path. Numbers on the opposite sides of the vectors indicate the time allotted for the task.

**Benefits of PERT Charts**

A PERT chart allows managers to evaluate the time and resources required to manage a project. This includes the ability to track assets needed during any stage of production in the course of the entire project. PERT analysis incorporates data and information from multiple departments.

**Disadvantages of PERT Charts**

The use of a PERT chart is highly subjective and subject to management’s experience. For this reason, PERT charts may include unreliable data or unreasonable estimations for cost or time. PERT charts are deadline-focused and may not fully communicate the financial positioning of the project. Finally, because a PERT chart is labour intensive, the establishment and maintenance of the information require additional time and resources. Continual review of the information provided, as well as prospective positioning of the project, is required for PERT charts to be valuable.



Chapter 5: System Study

* **Existing System**
* Existing system is a manual system it will not provide the online system
* Existing system does not provide the separate login for the user (customer).
* Existing system does not provide the online transaction facility.
* This system does not give the update account information for the customer
* Existing system is having many problems such as security problems
* Existing system has more human involvement which is a time consuming process with many manual calculations.
* It even includes the machine damage and signature verification process for secured transactions which allows the customers and banks to waste their valuable time and resources.
* The major problem in online banking system is unauthorized user access with fake passwords.
* **Proposed System**
* The proposed system is highly computerized in which the data related to user accounts will be secured high with high accuracy that even reduces the machine damage and human made errors and this existing system is highly efficient to offer best services to the customers as well as banks because it has user friendly access that consumes less time when compared with normal banking system.
* The development of this new system contains the following activities, which try to develop on-line application by keeping the entire process in the view of data base integration approach.
* Customer must have a valid user ID and password to login to the system
* User password id not stored anywhere; rather, md5 code of the password is generated and used. Therefore no one can login with a fake password or no one can get the password even if they hack the database.
* After the valid user login in he is shown the account he is taken to a page which shows the present balance in that particular account number. User can view his monthly as well as annual statements. He can also take print out of the same.
* User can transfer funds from his account to any other account with this bank.
* If the transaction is successful a notification should appear to the customer, in case it is unsuccessful, a proper message is given as to the customer as to why it failed.
* **Feasibility Study**

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. A feasibility study could be one of the best investments you ever make in your organization. According to Compass Point’s article, what are Fundraising Feasibility Studies and are they worth? “A feasibility study is an objective survey of the community that assesses the likelihood of success for a fundraising project, and identifies strategies and specific individual givers for the campaign”. We would add that a feasibility study also assesses the internal readiness of an organization to take on a major fundraising project and recommends a plan to build organizational capacity if your organization is not ready. Would you start a new business without, first, creating a business plan? The results would most likely be disastrous. A feasibility study, like a business plan, provides a roadmap for the future so you can move into a fundraising campaign with confidence.

**Some of the specific benefits of a feasibility study are to:**

* Test the interest and commitment of current and potential donors and investors for your cause
* Provide an opportunity to test different messages with donors, investors and the community to determine the best case for the campaign. Remember, people respond to the need in the community, not the dollar amount you want to raise, so the stronger and clearer your message, the better!
* Allow yourself the ability to reaffirm or adjust campaign financial and timeline goals based on community feedback
* Begin to generate enthusiasm and interest for your cause and campaign
* Set the stage for continuing cultivation with the stakeholders you engage during the feasibility study process
* Generate new names and organizations who might be interested in contributing to the campaign
* Show the strengths and opportunities for improvement within an organization, specifically within the development area (at both a staff and board level)
* Provide an opportunity to develop board knowledge around fundraising and campaigns
* Create a roadmap and timeline for campaign implementation

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

* **ECONOMICAL FEASIBILITY:**

The economic analysis checks for the high investment incurred on the system. It evaluates development and implementing charges for the proposed “Metro Smart Card System Project”. The software used for the development is easily available at minimal cost & the database applied is freely available hence it results in low cost implementation.

* **OPERATIONAL FEASIBILITY**

A proposed system is beneficial only if it can be turned into an information system that will meet the operational requirements of an organization. A system often fails if it does not fit within existing operations and if users resist the change.

**Important issues a systems developer must look into are:**

* Will the new system be used if implemented in an organization?
* Are there any major barriers to implementation or is proposed system accepted without destructive resistance?

The whole purpose of computerizing the Merchant Management is to handle the work much more accurately and efficiently with less time consumption. There will be additional work to be completed, because now the Pay Commerce Company will have to maintain database of both their merchants as well as their Customers.

Compared to the semi-computerized system the chances of avoiding errors in a computerized system is much higher because the user need not stress himself unnecessarily resulting in recklessness. Unlike the semi-computerized system there would be backup data for all the information concerning the daily transactions occurred within the organization.

Another important fact to be regarded is the security control, which is handled by the system. Since data regarding each Customer and the Organization is confidential, security is a key issue. Information falling into the wrong hands could jeopardize the entire organization. Unlike in semi-computerized systems

* **TECHNICAL FEASIBILITY:**

This aspect concentrates on the concept of using Computer Meaning, “Mechanization” of human works. Thus the automated solution leads to the need for a technical feasibility study.

The focus on the platform used database management and use for that software.

The proposed system doesn’t require an in depth technical knowledge as the system development is simple and easy to understand. The Software used makes the system user friendly (GUI). The result obtain should be true in the real time conditions.

* **BEHAVIORAL FEASIBILITY**

People are inherently resistant to change and computer has been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the organization.

**Chapter 6: Introduction to Project**

**Online Banking System**

This system provides an online solution to the customer by providing facilities such as balance enquiry, funds transfer to another account in the same/different bank etc. Many consumers today are turning to the ease and convenience of Internet banking to take care of their financial needs. With the new levels of access made possible by the Internet, people can now check the status of their finances with the click of a button. The history of Internet banking has evolved from simply allowing customers to check balances online, to now being able to trade stocks and bonds from the comfort of their own home!

The Traditional way of maintaining details of a user in a bank was to enter the details and record them. Every time the user needs to perform some transactions he has to go to bank and perform the necessary actions, which may not be so feasible all the time. It may be a hard-hitting task for the users and the bankers too. The project gives real life understanding of Internet banking and activities performed by various roles in the supply chain. Here, we provide automation for banking system through Internet. Internet banking system project captures activities performed by different roles in real life banking which provides enhanced techniques for maintaining the required information up- to-date, which results in efficiency. One of the main reasons that people decide to begin using Internet banking over regular banking is in reference to convenience. Everybody wants methods that are going to save them time and make daily tasks easier on them, and that is exactly what Internet banking does.

**Identifying the Entities**

* **User:** The user is the main entity here i.e. the end person who will the using the product.
* **Account:** Every user will have either a Savings account or a Current account or both.
* **Transaction:** Every transaction that occurs interbank or intra-bank must be stored for display or further use.
* **Admin:** A super user or admin to oversee the operations and to disable/enable a user and view all users and transactions that occur in the bank.
* **Bank Details:** There are various branches of the bank and thus maintaining the details of every bank is required.

**Identifying the Tables**

Oracle 11g database is used to store the data. Proper Standards as per the company were followed to properly create the database and the required tables.

* **User Table:** This table stores the user details like name, last name, DOB, username, encrypted password, type of user etc.
* **Account Table:** This table stores the account details like account number, account type, date opened, registered to user etc.
* **Transactions Table:** This table stores data like sender, receiver, purpose, date and time etc.
* **UIDAI Table:** This table stores the details of AADHAAR card and PAN card.
* **IFSC Table:** This table stores all the bank details.

**Services Available**

* User may login and view his account
* New User may register for creating an account.
* Post login user sees a menu wherein he can choose from a variety of services offered.
* User can add money (although a redundant feature as the only way to add money is through offline by going to the bank but it still exists to make the app work).
* User can pay / transfer funds through direct money transfer in case of same branch or via NEFT / RTGS in case of different branch.
* User can apply for a loan.
* User can check his User detail and/or Account Details and modify them.
* User can change his password if he does not feel secure.
* User can link his PAN and AADHAAR cards with the account.
* User can check his transaction history and can order mini statement. He can also save the transaction as PDF or Excel file.

**This project has following modules:-**

* **Home Screen**: This screen will show the home screen with the sign-in page, register page; about us section and locate a branch feature.
* **Login Page :** This Page takes in Username, Password, Account number and a captcha to sign-in the user and direct to its respective user profile page (or admin profile in case of admin sign-in). This page also directs the user to register for an account in case he doesn’t have one or gives an option to reset password if he forgets his.
* **Register Page:** If a user does not have an account or wants to create a new account he can do by providing the relevant details here.
* **About us:** This page briefly describes the bank, our Moto, or patrons and the details one may require.
* **Locate a Branch:** This feature lets the user locate a branch near him by entering the city / state / pin code.
* **User login:** User login has following features:
  + **Check/Update Account Details**
  + **Check/Update User Details**
  + **Pay via NEFT/RTGS/Account Transfer**
  + **Add money**
  + **Link Pan/Aadhaar**
  + **View Transactions**
  + **Change password**
  + **Apply for loan/Calculate EMI**
  + **Validate an IFSC Code**
* **Admin login:** Additional features that an admin has are:
  + **View All Transactions**
  + **View All Accounts**
  + **View All Users**
  + **Disable / Enable User(s)**

**Constraints**

* + **Login and password is used for identification of customer’s account and there is no facility for OTP to login.**
  + **This system works only on a single server.**
  + **GUI is only in English.**
  + **Limited to HTTP/HTTPS protocol.**

**Assumptions and Dependencies**

* + **The details of customers such as username, password, account type and their corresponding authority details should be manually entered by the administrator before using this system and there is no way to verify them.**
  + **Every user should be comfortable of working with computer and net browsing.**
  + **He should be aware of the banking system.**
  + **He must have basic knowledge of English too.**

**Chapter 7: System Analysis**

**DESIGN AND FEATURES**

The client will have client interface in which he can interact with the banking system. It is a web based interface which will be the web page of the banking application. Starting a page is displayed asking the type of customer he is whether ordinary or a corporate customer. Then the page is redirected to login page where the user can enter the login details. If the login particulars are valid then the user is taken to a home page where he has the entire transaction list that he can perform with the bank. All the above activities come under the client interface. The administrator will have an administrative interface which is a GUI so that he can view the entire system. He will also have a login page where he can enter the login particulars so that he can perform all his actions. This administrative interface provides different environment such that he can maintain database & provide backups for the information in the database. He can register the users by providing them with username, password & by creating account in the database. He can view the cheque book request & perform action to issue the cheque books to the clients.

**USAGE**

The aim of this project is to develop a secured online banking system with the following objectives:

* Create a banking system that is easily accessible by customers from the comfort of their homes, offices etc.
* Reduce the flow of human traffic and long queues at banks
* Reduce the time wasted in going to banks to stay on queues.
* Promote efficient and effective banking for the banks by focusing on those services that still require physical presence at the banking hall.

**Hardware Requirements**

|  |  |
| --- | --- |
| **Processor** | 800 MHz or faster |
| **RAM :** | 1 GB or more |
| **Java Version** | 1.8 |
| **Free Disk Space** | 500 MB or more |
| **Internet Connection** | Required |

**Software Requirement:-**

|  |  |
| --- | --- |
| **UI** | HTML, CSS, JSP, JSTL |
| **DB** | Oracle 11g, Sqldeveloper |
| **Server side** | JSP |
| **IDE** | Eclipse |
| **Server** | Apache Tomcat v9 |

**Front-end**

In this project dynamic approach was used to create the web project.

A server-side dynamic web page is one where its construction is controlled by an application server processing server-side scripts. In server-side scripting, parameters determine how the assembly of every new web page proceeds, including the setting up of more client-side processing.

A dynamic web page is then reloaded by the user or by a [computer program](https://en.wikipedia.org/wiki/Computer_program) to change some variable content. This updating information could come from the server, or from changes made to that page's DOM. This may or may not truncate the browsing history or create a saved version to go back to, but a dynamic web page is updated using [JSP](https://en.wikipedia.org/wiki/Ajax_(programming)) technologies that will neither create a page to go back to, nor truncate the [web browsing history](https://en.wikipedia.org/wiki/Web_browsing_history) forward of the displayed page. Using JSP technologies the end [user](https://en.wikipedia.org/wiki/User_(computing)) gets one dynamic page managed as a single page in the [web browser](https://en.wikipedia.org/wiki/Web_browser) while the actual [web content](https://en.wikipedia.org/wiki/Web_content) rendered on that page can vary. The Server engine sits only on the browser requesting parts of its DOM, the DOM, for its client, from an application server.

**Back-end.**

In this project Oracle 11g was used as a backend to store the data.

Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is a multi-model database management system produced and marketed by Oracle Corporation. It is a database commonly used for running online transaction processing (OLTP), data warehousing (DW) and mixed (OLTP & DW) database workloads. It is available on-prem, on-Cloud, or in a hybrid-Cloud environment. It may also be deployed on Oracle Engineered Systems (e.g. Exadata) on-prem, on Oracle (public) Cloud or (private) Cloud at Customer (e.g. Exadata Cloud at Customer). At Openworld 2017 in San Francisco, Executive Chairman of the Board and CTO, Larry Ellison announced the next database generation, Oracle Autonomous Database.

**Data Flow Diagram**

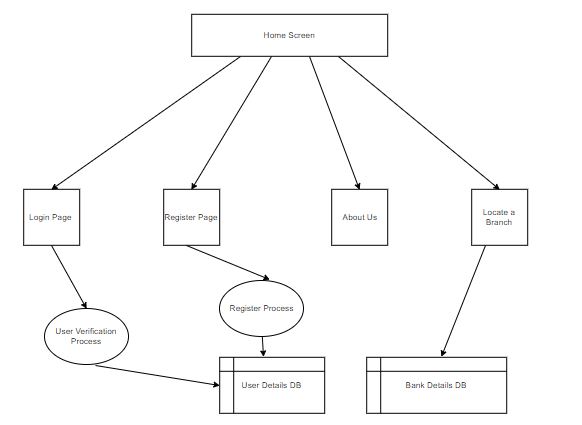
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years.

**DFD Symbols**

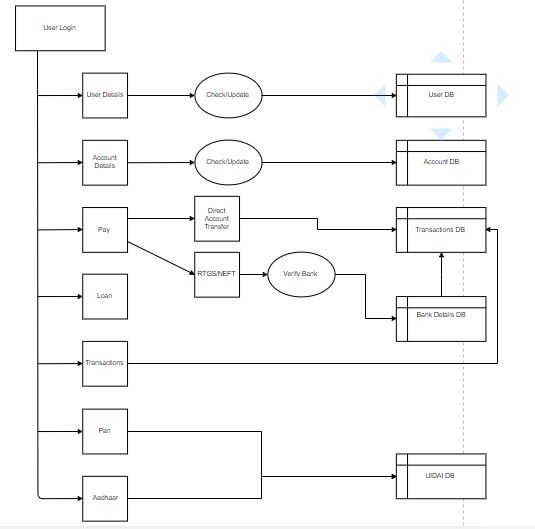
In the DFD, there are four symbols,

|  |  |  |
| --- | --- | --- |
| **Name** | **Symbol** | **Description** |
| **Square** |  | External Entity, Source or Destination or data |
| **Arrow** |  | Identifies the flow and direction of data |
| **Circle** |  | Represents a process that transforms incoming data flow(s) into outgoing data flow(s) |
| **Open Rectangle** |  | Store Data |

**Diagram for Home Page**

****

**Diagram for Login Page**

****

**Entity Relationship Diagram**

The ER diagrams define the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases**.**

**Importance of ERDs and their uses**

Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

However, while an ERD can be useful for organizing data that can be represented by a relational structure, it can't sufficiently represent semi-structured or [unstructured data](https://searchbusinessanalytics.techtarget.com/definition/unstructured-data). It's also unlikely to be helpful on its own in integrating data into a pre-existing information system.

**Main components of an ERD**

ERDs are generally depicted in one or more of the following models:

* A conceptual data model, which lacks specific detail but provides an overview of the scope of the project and how [data sets](https://whatis.techtarget.com/definition/data-set) relate to one another.
* A logical data model, which is more detailed than a conceptual data model, illustrating specific [attributes](https://searchmicroservices.techtarget.com/definition/attribute) and relationships among [data points](https://whatis.techtarget.com/definition/data-point). While a conceptual data model does not need to be designed before a logical data model, a physical data model is based on a logical data model.
* A physical data model, which provides the blueprint for a physical manifestation -- such as a relational database -- of the logical data model. One or more physical data models can be developed based on a logical data model.
* **Entities:** An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in our database, \_User is an entity. All these entities have some attributes or properties that give them their identity. Entities are something about which data is collected, stored, and maintained.

* **Attributes:** Entities are represented by means of their properties, called attributes. All attributes have values. For example, a user entity may have name, and age as attributes. Attributes are the properties of entities. Attributes are directly connected to its entity. There exists a domain or range of values that can be assigned to attributes. For example, a user name cannot be a numeric value. It has to be alphabetic. A user age cannot be negative, etc.
* **Relationship:**  The association among entities is called a relationship. An example of a relationship would be employees are assigned to projects. Here, assigned is called relationships.

**Why Would You Use an ERD?**

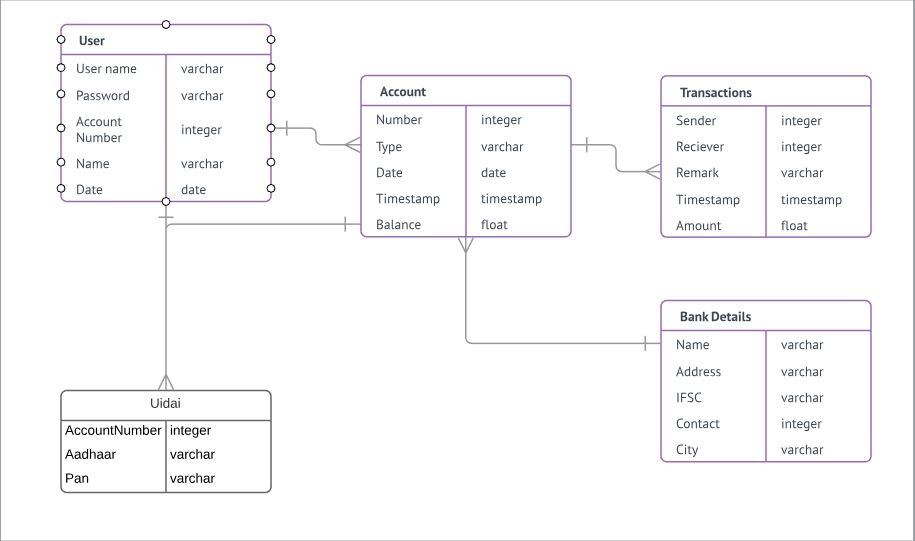
When it comes to business analysis, ERD is a critical part of designing your software systems or the way you implement your [business intelligence solutions](https://www.sisense.com/solutions/bi).

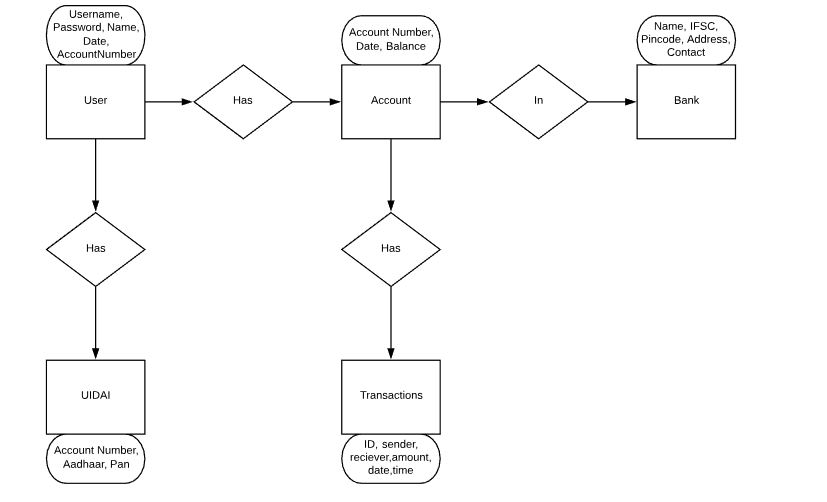
Businesses with complex databases – in other words, lots of data – might use ERD.

It’s mainly used by analysts when designing a database. The database helps them communicate the landscape of the business to different teams, and this overview will help you build the applications needed to support the business.

Using a visual representation of your business will help you understand its structure and this information is useful for business management and formulating strategies – and ultimately making efficiencies.

Understanding the relationships between entities can also help you root out any ambiguities or unnecessary processes within your organization. If you can do all that, you’re well on the way to streamlining your business.

****

****

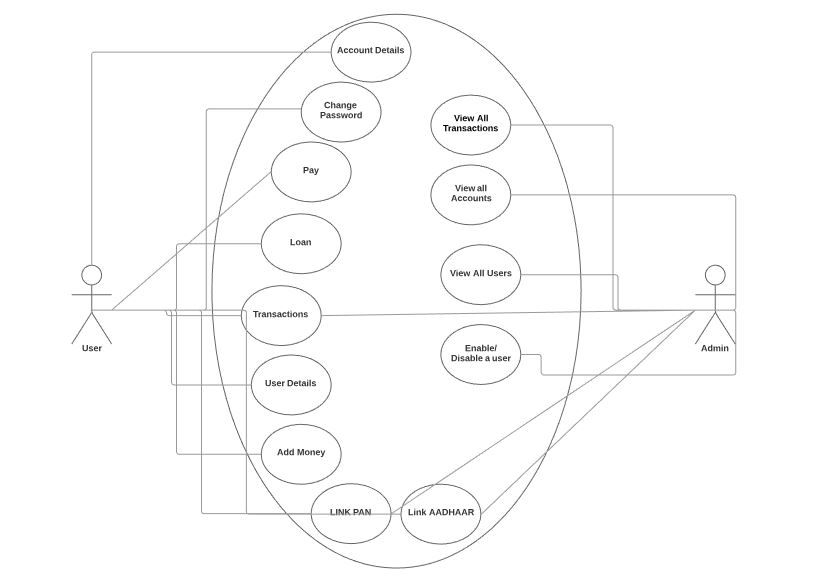
**Use Case Diagram**

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that "Use case diagrams are the blueprints for your system". They provide the simplified and graphical representation of what the system must actually do.

Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed. Siau and Lee conducted research to determine if there was a valid situation for use case diagrams at all or if they were unnecessary. What was found was that the use case diagrams conveyed the intent of the system in a more simplified manner to stakeholders and that they were "interpreted more completely than class diagrams".

The purpose of the use case diagrams is simply to provide the high level view of the system and convey the requirements in layman's terms for the stakeholders. Additional diagrams and documentation can be used to provide a complete functional and technical view of the system.

****

**Chapter 8: Software Design**

**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 analysed, 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.

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

For assessing user requirements, an SRS (Software Requirement Specification) document is created whereas for coding and implementation, there is a need of more specific and detailed requirements in software terms. The output of this process can directly be used into implementation in programming languages.

Software design is the first step in SDLC (Software Design Life Cycle), which moves the concentration from problem domain to solution domain. It tries to specify how to fulfil the requirements mentioned in SRS.

## Software Design Levels

Software design yields three levels of results:

* **Architectural Design -**The architectural design is the highest abstract version of the system. It identifies the software as a system with many components interacting with each other. At this level, the designers get the idea of proposed solution domain.
* **High-level Design-**The high-level design breaks the ‘single entity-multiple component’ concept of architectural design into less-abstracted view of sub-systems and modules and depicts their interaction with each other. High-level design focuses on how the system along with all of its components can be implemented in forms of modules. It recognizes modular structure of each sub-system and their relation and interaction among each other.
* **Detailed Design-**Detailed design deals with the implementation part of what is seen as a system and its sub-systems in the previous two designs. It is more detailed towards modules and their implementations. It defines logical structure of each module and their interfaces to communicate with other modules.

**Database Design**:

Oracle SQL Developer is an integrated development environment (IDE) for working with SQL in Oracle databases. Oracle Corporation provides this product free; it uses the Java Development Kit.

MySQL is the most popular Open Source Relational SQL Database Management System. MySQL is one of the best RDBMS being used for developing various web-based software applications. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. This tutorial will give you a quick start to MySQL and make you comfortable with MySQL programming.

MySQL is written in C and C++. Its SQL parser is written in YACC, but it uses a home-brewed lexical analyser MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, macOS, Microsoft Windows, NetBSD, Novell NetWare, Open SD, Open Solaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO Open Server, SCO UnixWare, Samos and Tru64. A port of MySQL to OpenVMS also exists.

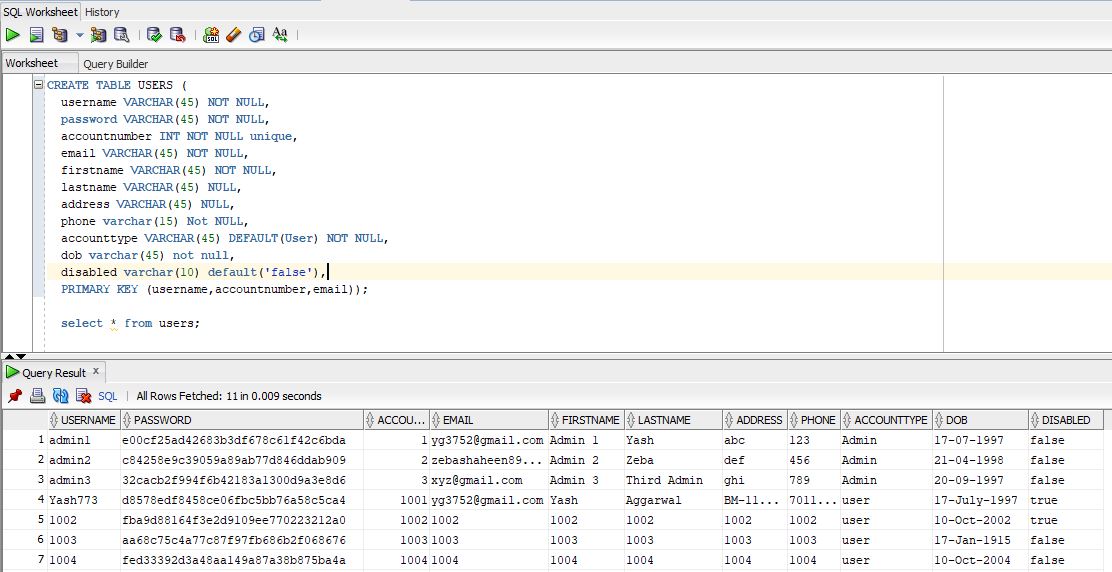
The MySQL server software itself and the client libraries use dual-licensing distribution. They are offered under GPL version 2, beginning from 28 June 2000[20] (which in 2009 has been extended with a FLOSS License Exception) or to use a proprietary license.

Support can be obtained from the official manual. Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organisations exist to provide support and services, including MariaDB and Percona.

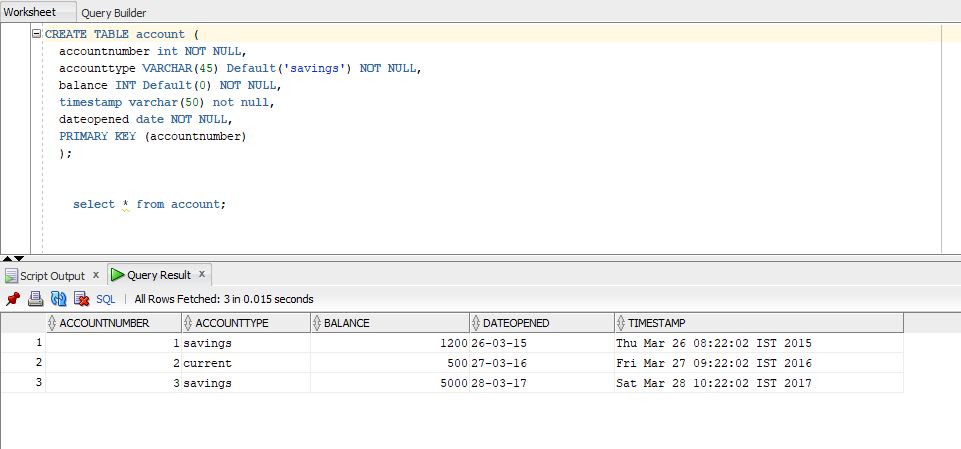
MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case" and that the "developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good”. It has also been tested to be a "fast, stable and true multi-user, multi-threaded SQL database server".

**Tables Used**:

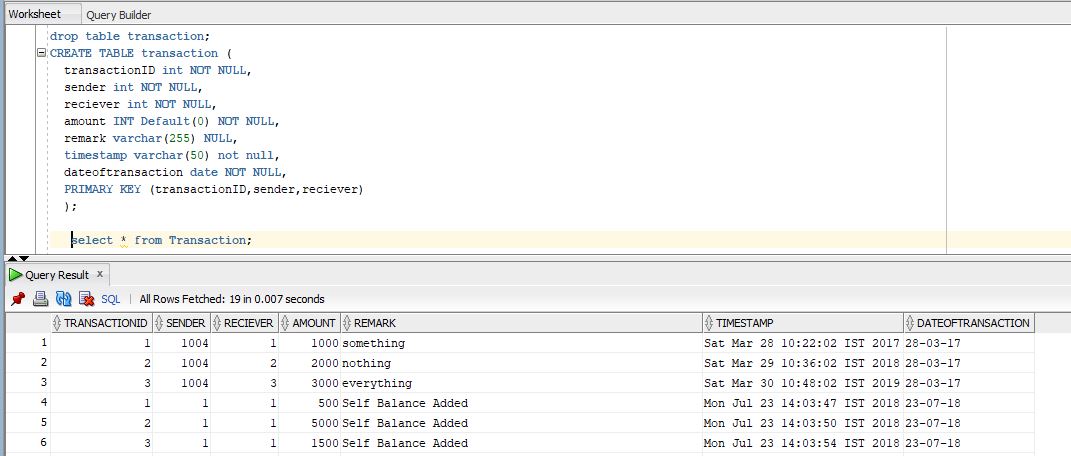
* **User Table:**



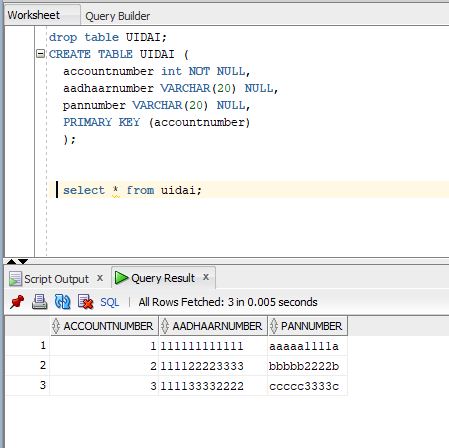
* **Account Table**



* **Transactions Table**



* **UIDAI Table**



**Program Design**

**Introduction**

**Online Banking System using Spring Framework in Java.**

**Requirement Scope**

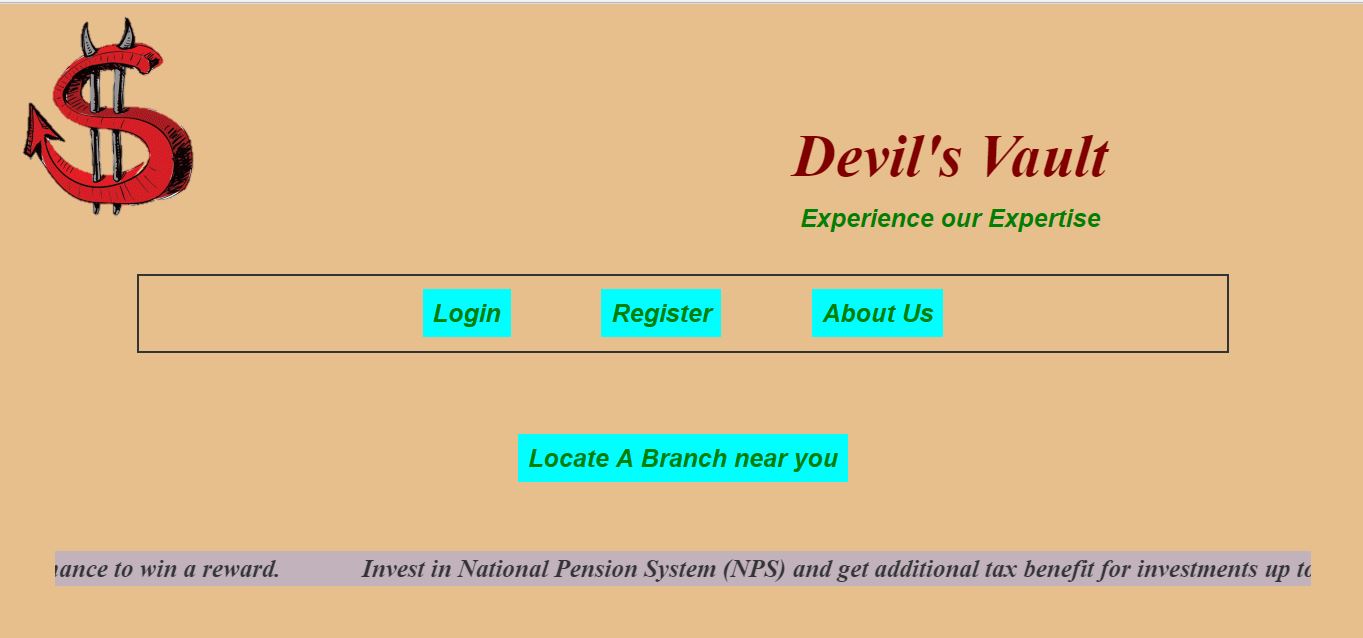
1. An Online Domain name and server space to store and manage the data.
2. API from UIDAI to check and validate the AADHAAR and PAN card details.
3. API to send OTP to registered Email and Mobile number.
4. A service to manufacture and implement Credit/Debit Cards.

**This Design is not production ready.**

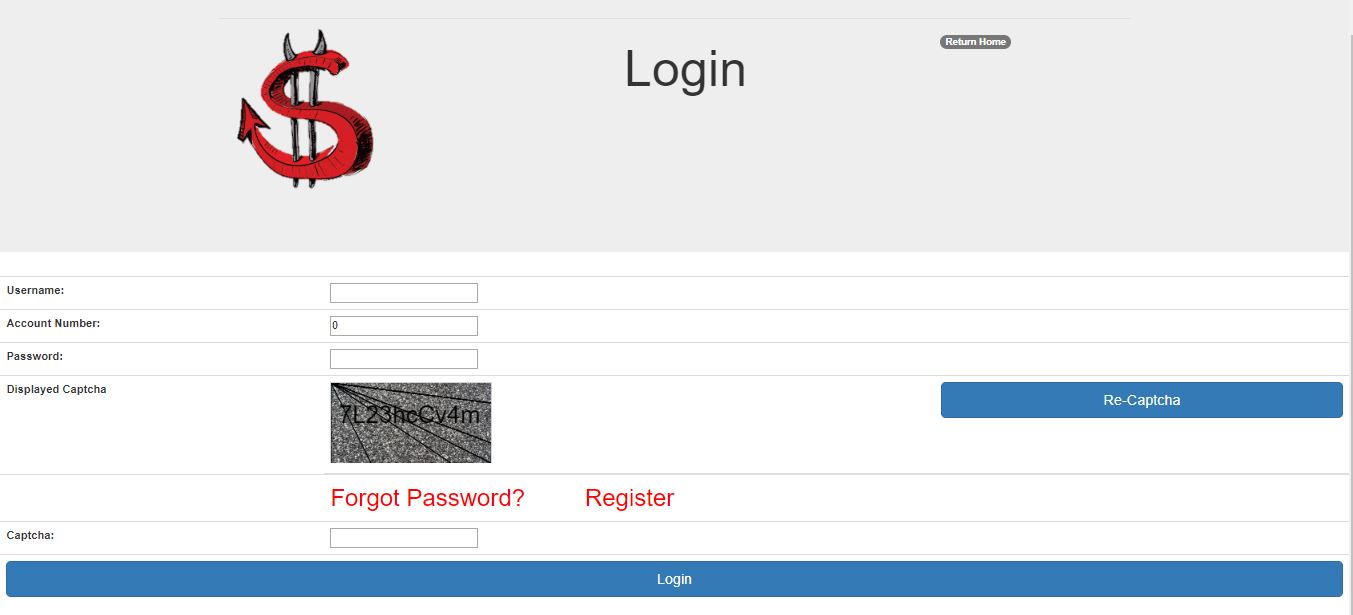
The motive of this exercise is to evaluate candidate skills rather than developing production ready software for an Online banking system. In a real production environment, the technology stack may differ significantly. In real life scenario, for example,

1. Lot of hardware interaction will be required for card sensor devices connectivity with servers.
2. Caching needs to be implemented for quick Transactions.
3. There will be a RDBMS for storing all the data that needs to be maintained by on-site Developer.
4. Real-time Money adding system.
5. Real production application may be hosted on a cloud to meet changing infrastructural requirements of the software, for example morning and evening office hours could need much more processing power than off hours. Thus some kind of Elastic Computing solution will be required in real scenario.

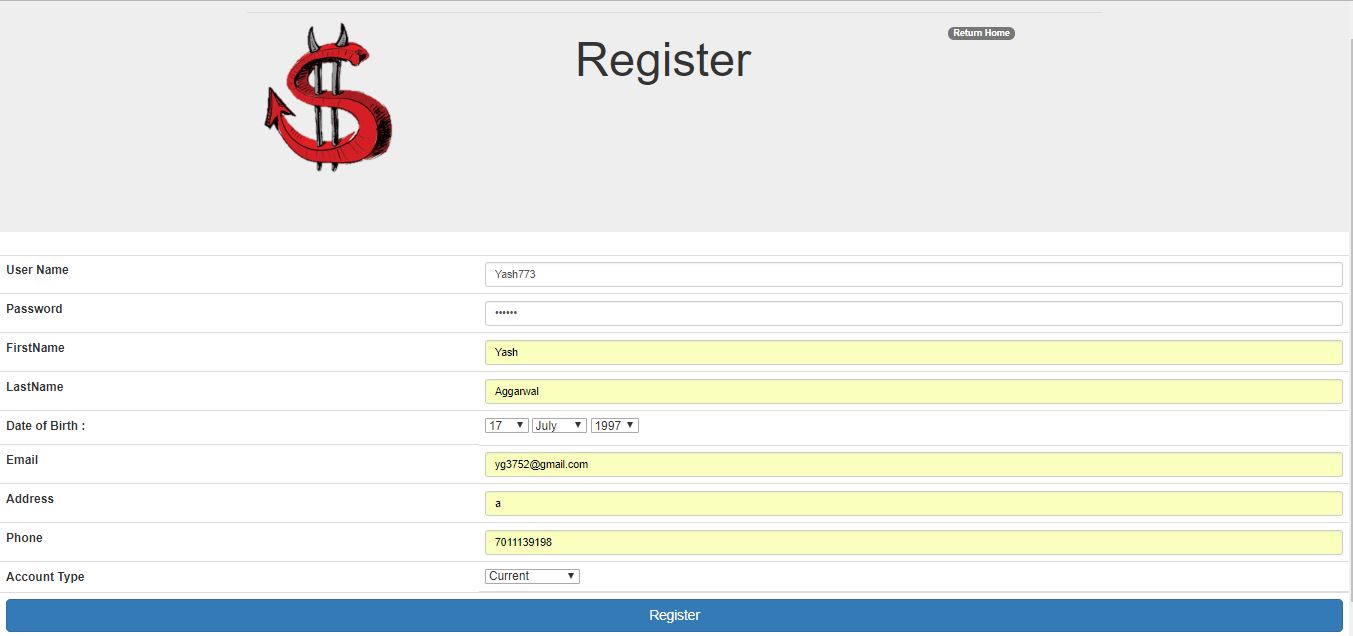
**Home Page**



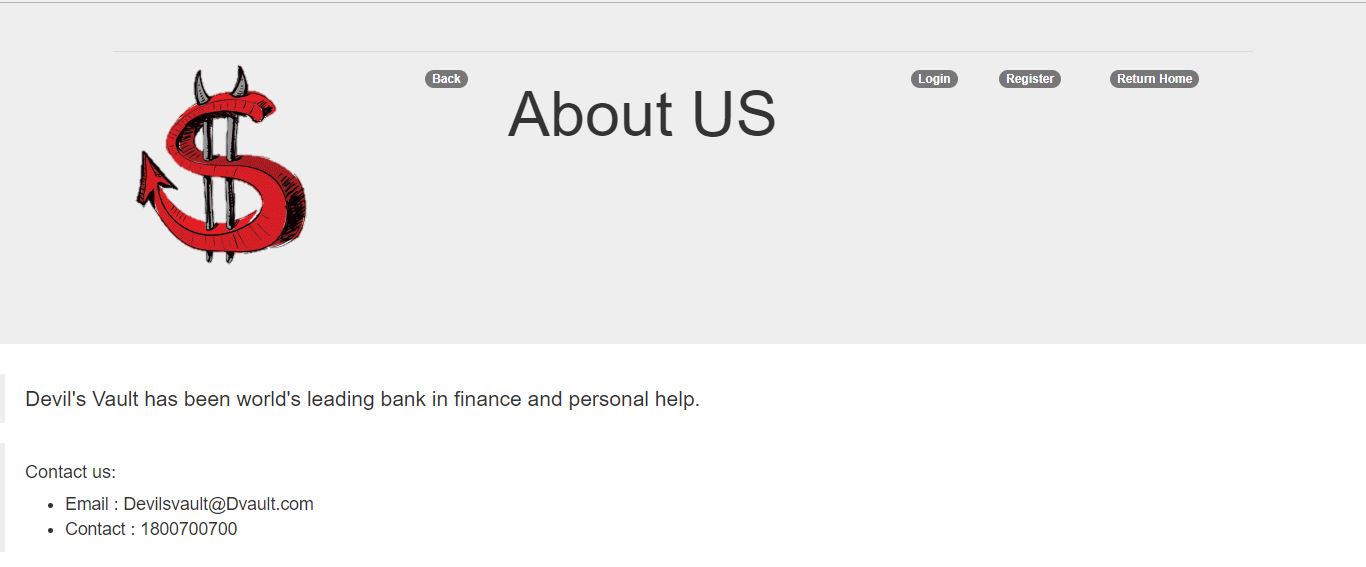
**Login Page**

****

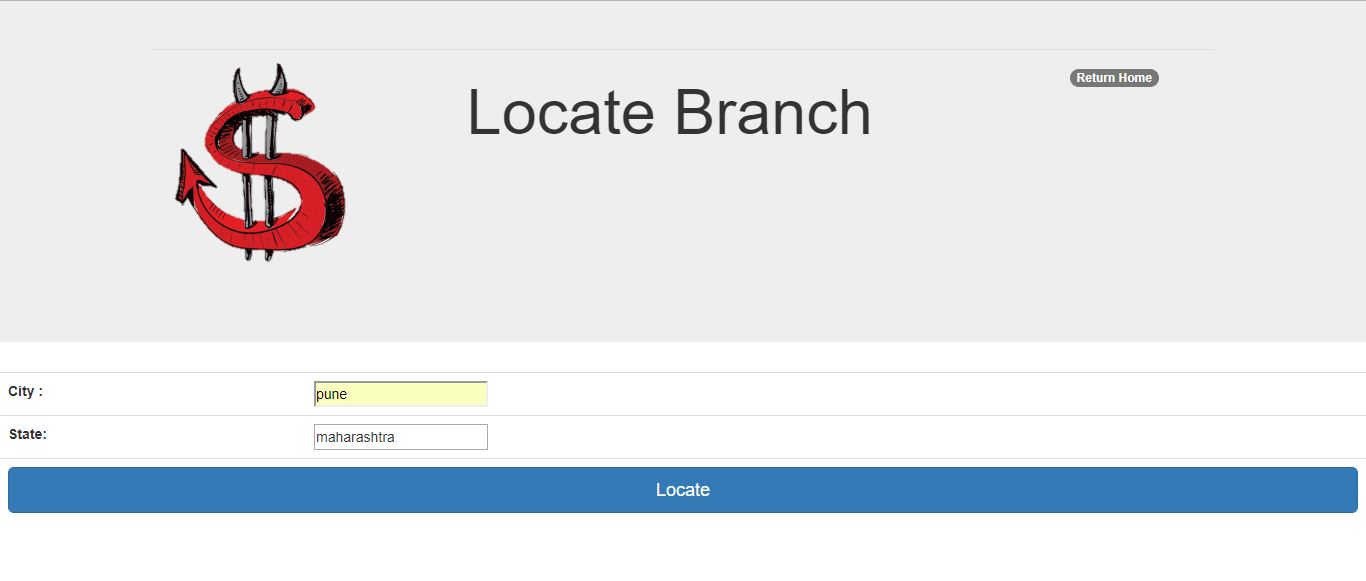
**Register Page**

****

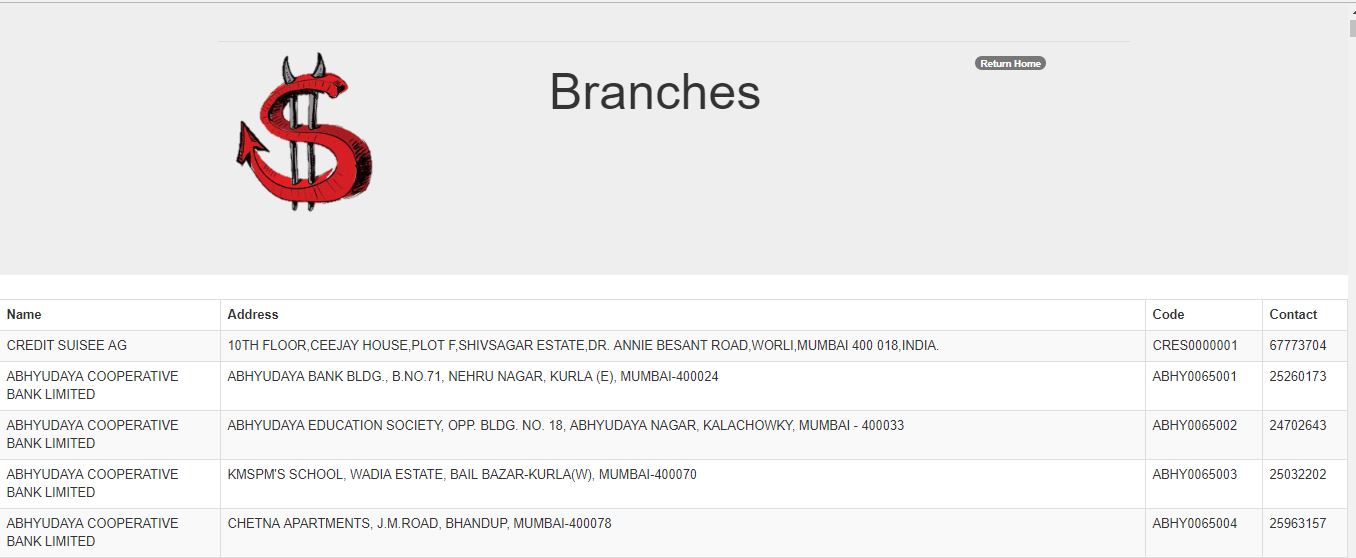
**About us Page**

****

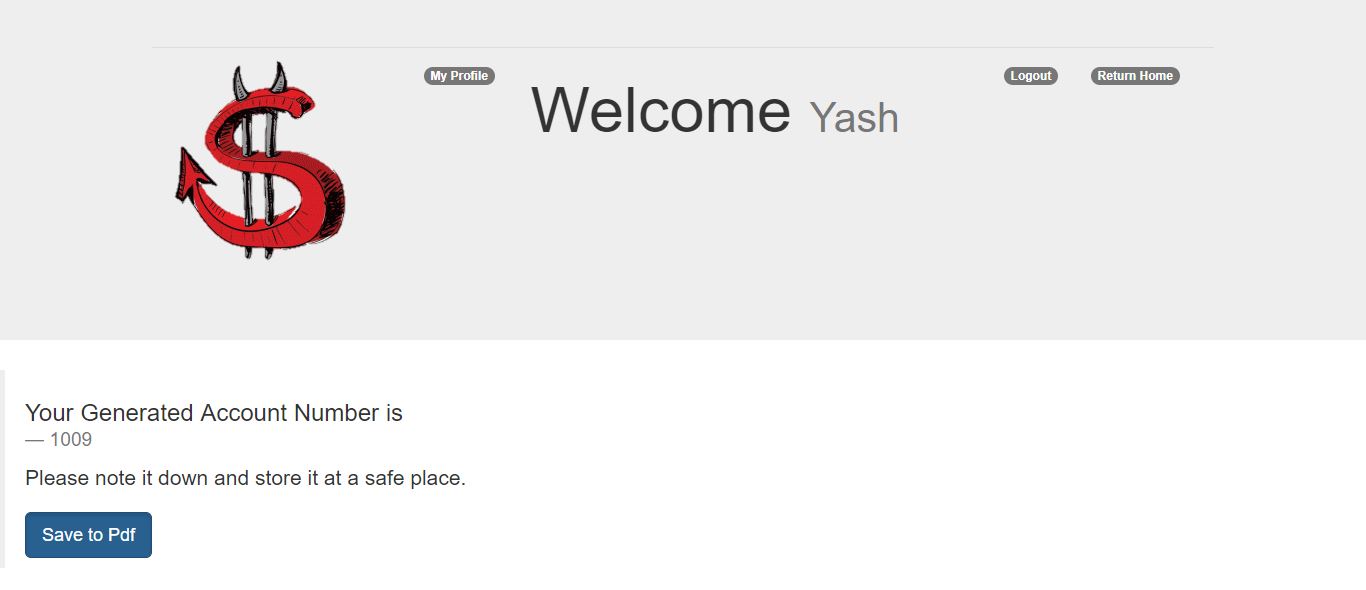
**Locate Page**

****

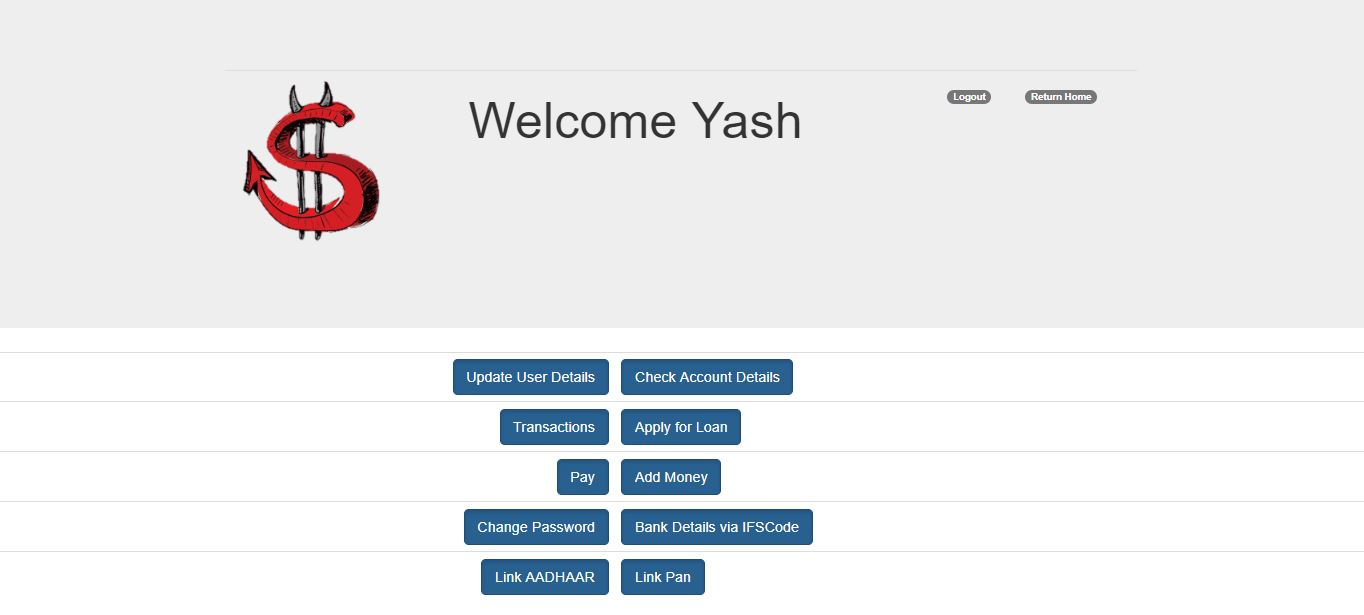
**Locate Successful Page**

****

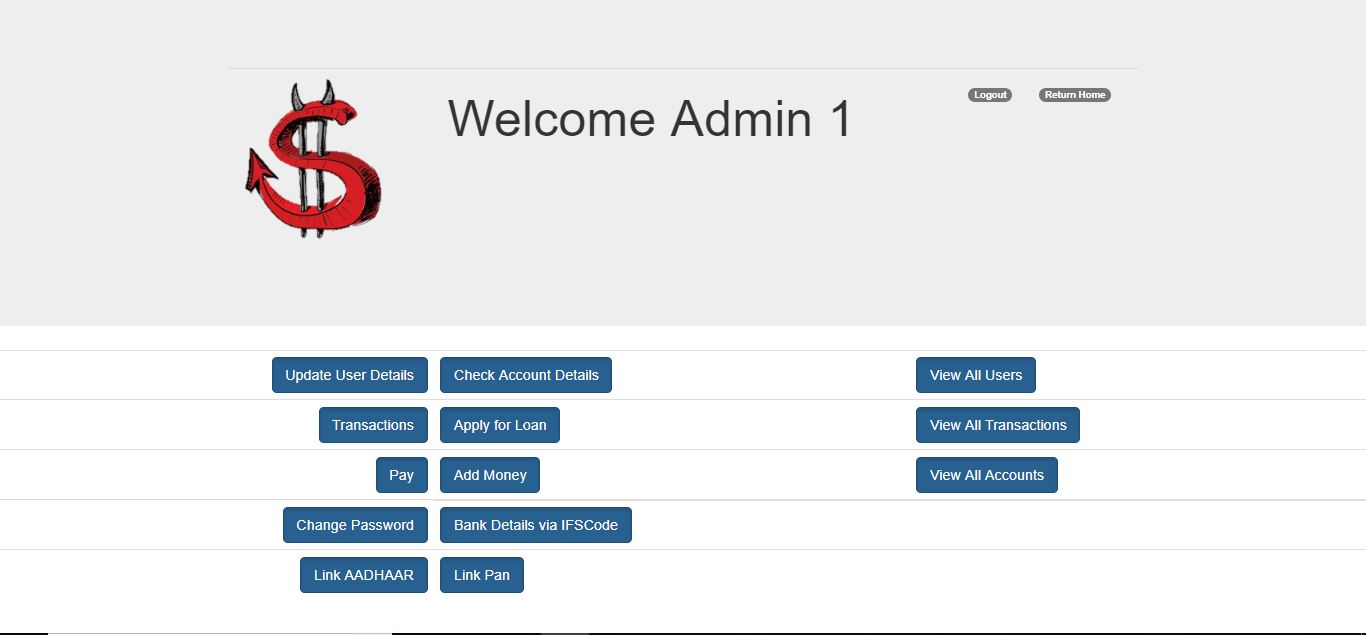
**Welcome Page**

****

**User Login**

****

**Admin Login**

****

**Chapter 9: Coding**

The input to the Design Phase is the design document. During the **coding phase**, different modules identified in the design document are coded according to the module specifications. The **Objectives** of the coding phase is to transform the design of the system, as given by its module specifications, into a high level languages and code and then unit this code. Software developers adhere to some well-defined and standard style of coding called coding standards. The reasons for adhering to a standard coding style are the following.

It gives a uniform appearance to the codes written by different engineers.

It encourages good programming practices.

It encourages code understanding.

**Coding Standards**

* Rules for limiting the use of global.
* Content of the headers preceding codes for different modules.
* Errors return conventions and exception handling mechanisms.

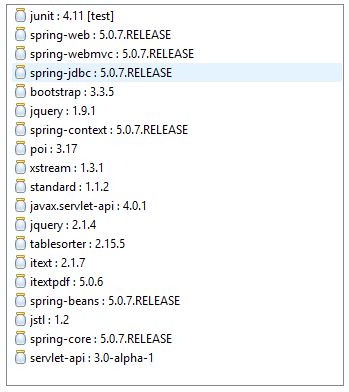
**Code Inspections**

They aim explicitly at the discovery of commonly made errors. During which, the code is examined for the presence of certain kinds of errors, in contrast to the hand simulation on code execution as done in code walk-through.

Once the design is complete, most of the major decisions about the system have been made. The goal of the coding phase is to translate the design of the system into code in a given programming language. For a given design, the aim of this phase is to implement the design in the best possible manner. The coding phase affects both testing and maintenance profoundly. A well written code reduces the testing and maintenance effort. Since the testing and maintenance cost of software are much higher than the coding cost, the goal of coding should be to reduce the testing and maintenance effort. Hence, during coding the focus should be on developing that are easy to write. Simplicity and clarity should be strived for, during the coding phase.

**Code Snippets to achieve various Specific Tasks**

* **Maven Dependencies**

****

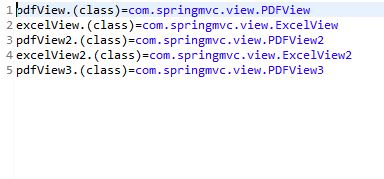
* **Create Custom Server and Set index File**

****

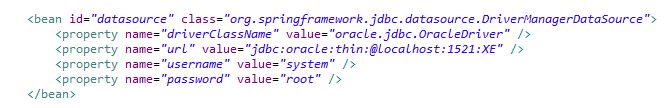
* **Create View Resolvers and set priority**



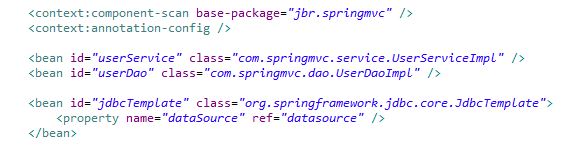
* **Map Custom Views**

****

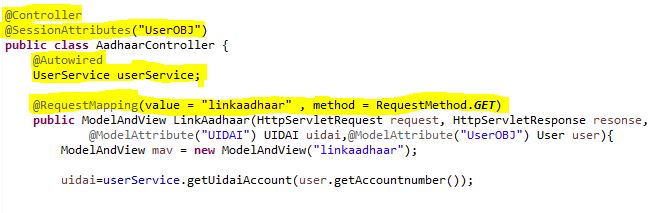
* **Set a data source and provide driver details**

****

* **Dao Mapping**

****

* **Basic Controller Syntax**

****

* **Basic Excel Writing**

****

* **Basic PDF Writing**

****

**Chapter 9: System Testing**

System Testing is the testing of a complete and fully integrated software product. Usually software is only one element of a larger computer based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer based system.

The focus of the system testing is to evaluate the compliance of the entire system with respect to the specified requirements. System testing helps in approving and checking the business, functional, technical, and any non-functional requirements of the application concerning the architecture as a whole.

The basic goal of the software development process is to produce software that has no errors or very few errors. In an effort to detect errors soon after they are introduced, each phase ends with verification activity such as a review.

As testing is the last phase before the final software is delivered, it has the enormous responsibility of detecting any type of error that may in the software. Software typically undergoes changes even after it has been delivered. And to validate that a change has not affected some old functionality of software regression testing is performed

The scope of system testing is not only limited to the design of the system but also to the behaviour and believed expectations of the business. In accordance with the software test cycle, system testing is performed before acceptance testing and after integration testing. Independent users or testers are given the tasks to perform in the system testing phase.

* In accordance to the software development lifecycle, system testing is considered as the first level of testing where the entire system is checked or tested.
* Proper evaluation of the system meeting the functional requirements is done in system testing.
* Validation, verification and testing of business requirements and application architecture are done during the system testing phase.
* System testing provides users with an effective environment which more or less resembles the live or production environment. As such any testing done provides more reliable and efficient results.

Two Category of Software Testing

* **Black Box Testing**-Black-box testing is a method of [software testing](https://en.wikipedia.org/wiki/Software_testing) that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: [unit](https://en.wikipedia.org/wiki/Unit_test), [integration](https://en.wikipedia.org/wiki/Integration_testing), [system](https://en.wikipedia.org/wiki/System_testing) and [acceptance](https://en.wikipedia.org/wiki/Acceptance_test). It is sometimes referred to as specification-based testing.
* **White Box Testing**-White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing [software](https://en.wikipedia.org/wiki/Software) that tests internal structures or workings of an application, as opposed to its functionality (i.e. [black-box testing](https://en.wikipedia.org/wiki/Black-box_testing)). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the expected outputs. This is analogous to testing nodes in a circuit, e.g. [in-circuit testing](https://en.wikipedia.org/wiki/In-circuit_test) (ICT). White-box testing can be applied the [unit](https://en.wikipedia.org/wiki/Unit_testing), [integration](https://en.wikipedia.org/wiki/Integration_testing) and [system](https://en.wikipedia.org/wiki/System_testing) levels of the testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

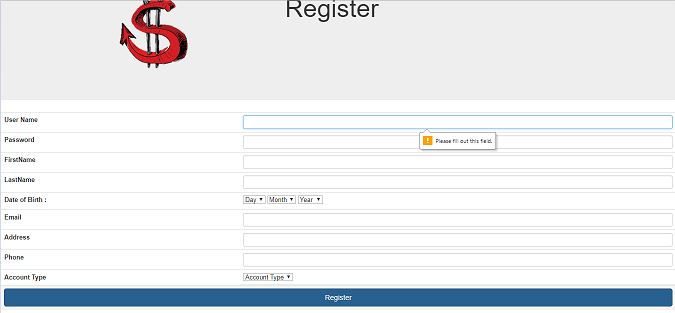
**Testing With Dummy Data:**

Dummy data is benign information that does not contain any useful data, but serves to reserve space where real data is nominally present. Dummy data can be used as a placeholder for both testing and operational purposes. For testing, dummy data can also be used as stubs or pad to avoid software testing issues by ensuring that all variables and data fields are occupied. In operational use, dummy data may be transmitted for OPSEC purposes. Dummy data must be rigorously evaluated and documented to ensure that it does not cause unintended effects.

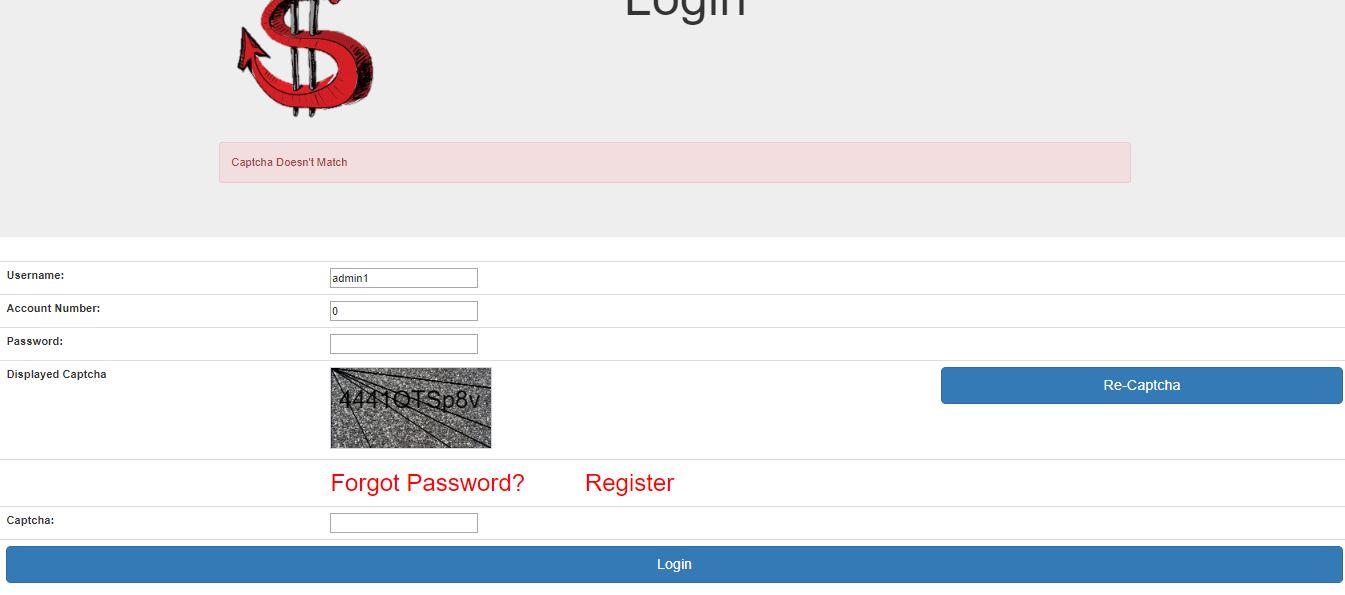
Testing with dummy data helps us find what happens if we mis-click or mis-type.

**Some of the cases are as below:**

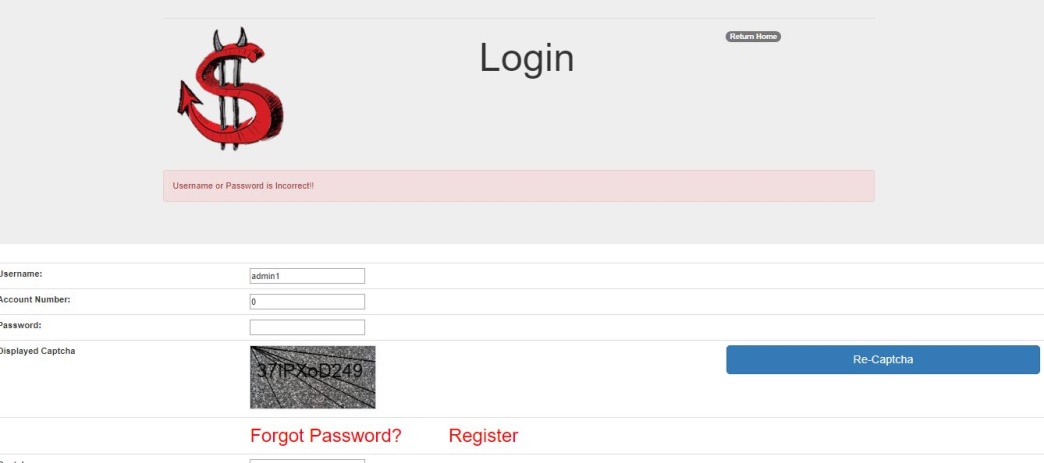
* + Not Entering all the information

****

* + Captcha Doesn’t match



* + Username Password is incorrect

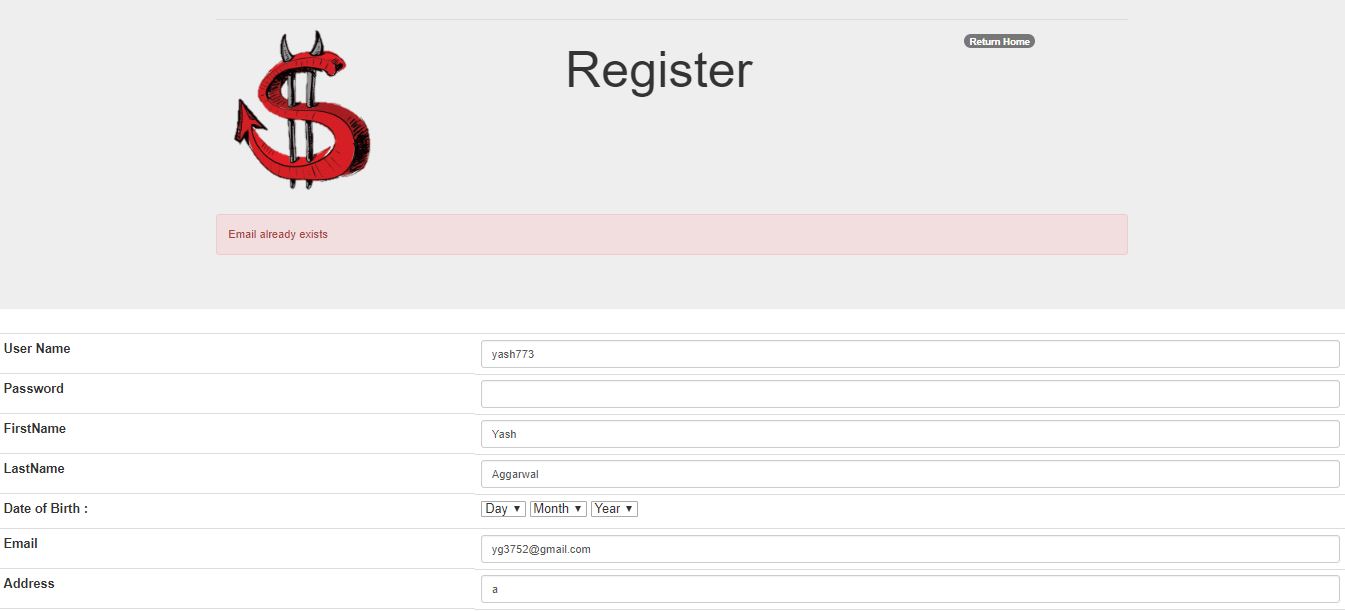
****

**Testing With Live Data:**

Real-time data (RTD) is information that is delivered immediately after collection. There is no delay in the timeliness of the information provided. Real-time data is often used for navigation or tracking. Such data is usually processed using real-time computing although it can also be stored for later or off-line data analysis. Real-time data is not the same as dynamic data. Real-time data can be dynamic (e.g. a variable indicating current location) or static (e.g. a fresh log entry indicating location at a specific time).

**Some operations on Testing with Live data are described below:**

* User Already Exists



**Chapter 10: Conclusion**

* Online Banking System is developed to fulfil client requirements with flexible and easy platform. So anyone can use this easily and maintain their Account.
* After doing this project, I learned variety of concepts frameworks and I would like to thank all the concerned individuals who have contributed to my precious learning.
* I have learned to prepare the requirement catalogue, functional specification, design specification, system development life cycle in practical development and also learned real life software development with tools having interactive features.
* I have also understood the importance of naming conventions and coding conventions in development process.
* Though the project was taking too long and I faced many difficulties which were resolved through the effective guidance of seniors of the organization.

**I am sure that I can perform better and better as I step upon the ladders of the experience**.

**Bibliography**

1. <https://stackoverflow.com/>
2. <https://github.com/>
3. <https://getbootstrap.com/>
4. <https://spring.io/>
5. <https://www.w3schools.com/>