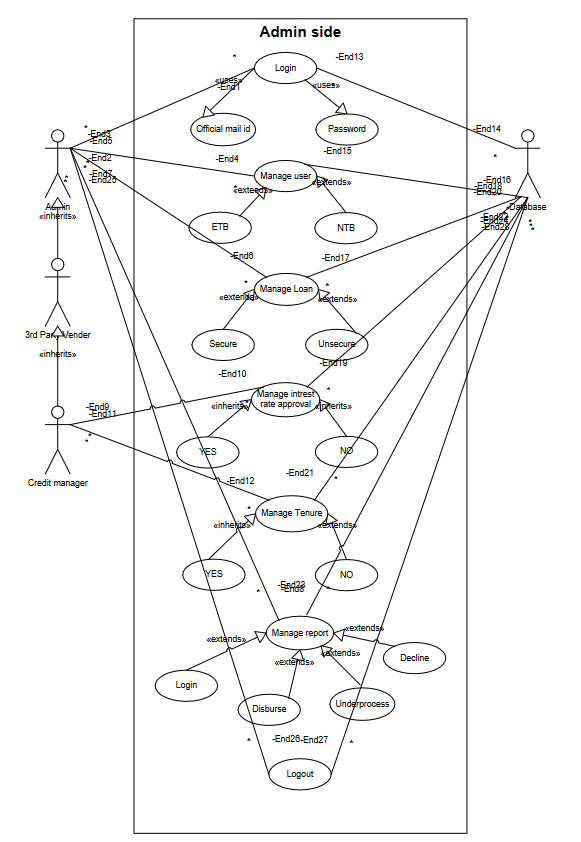
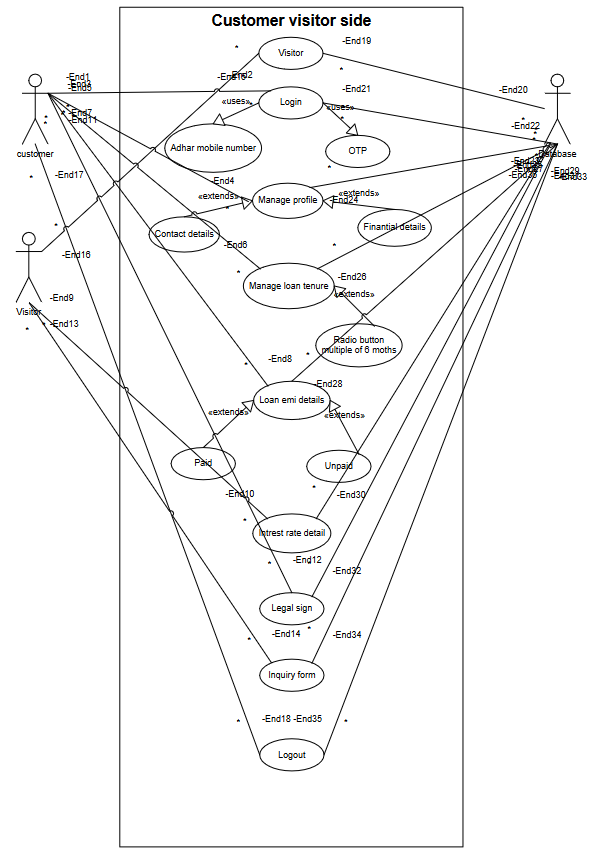
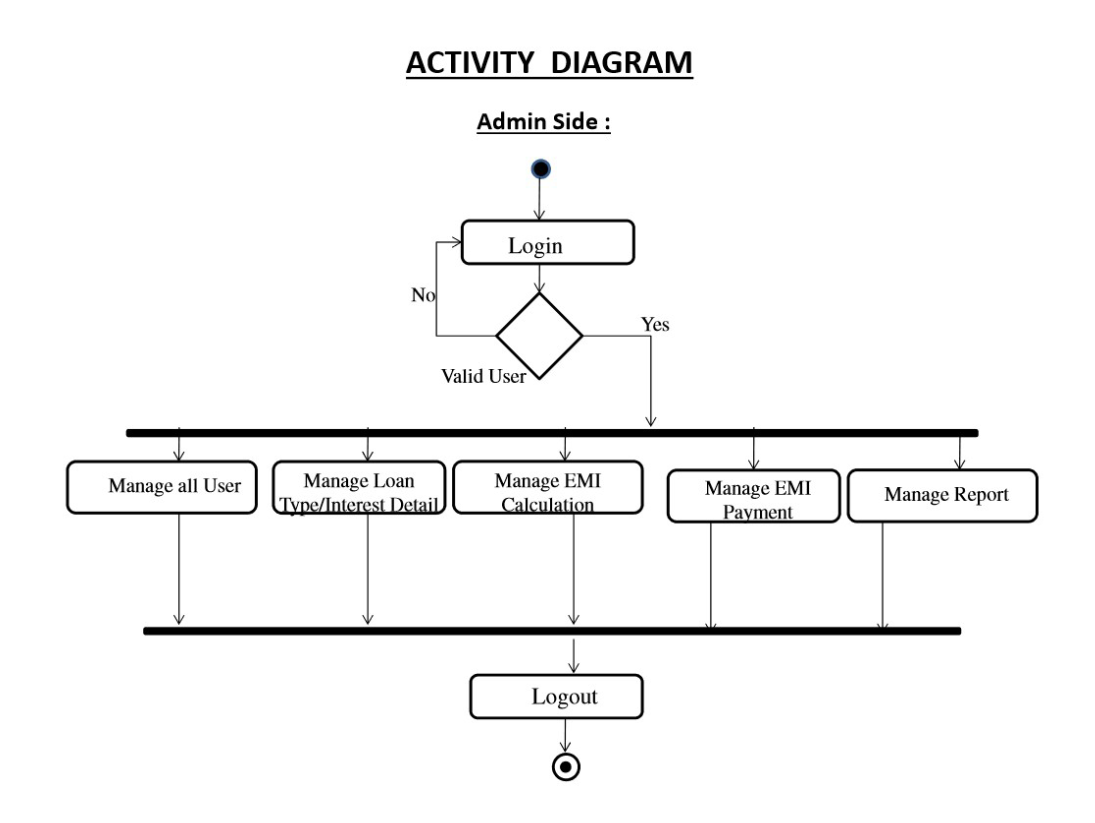
use case diagram



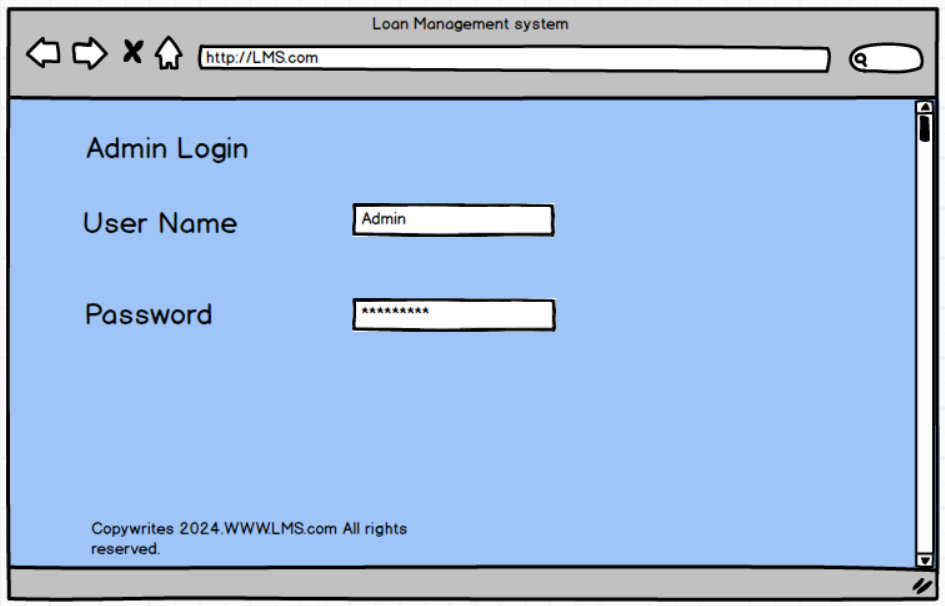


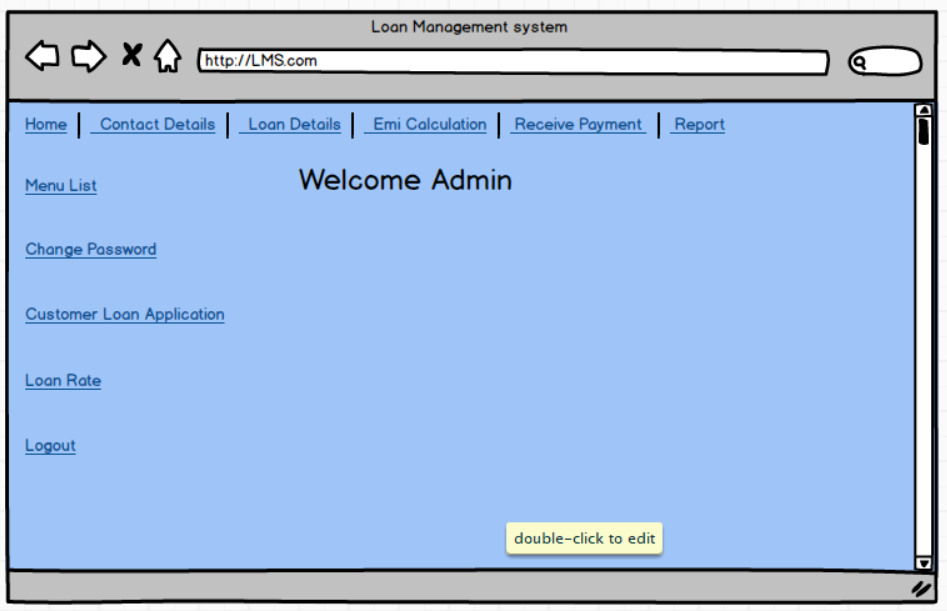
activity diagram

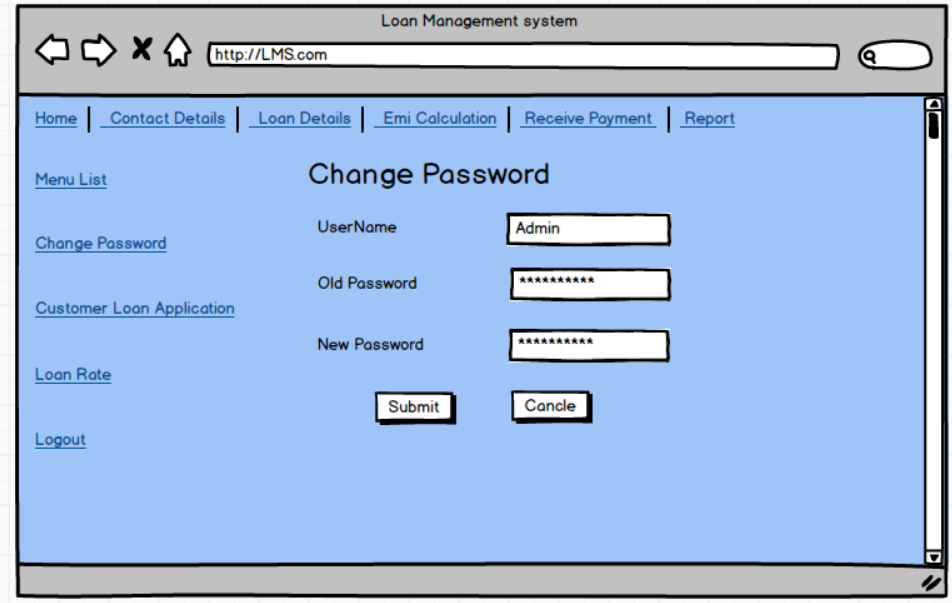


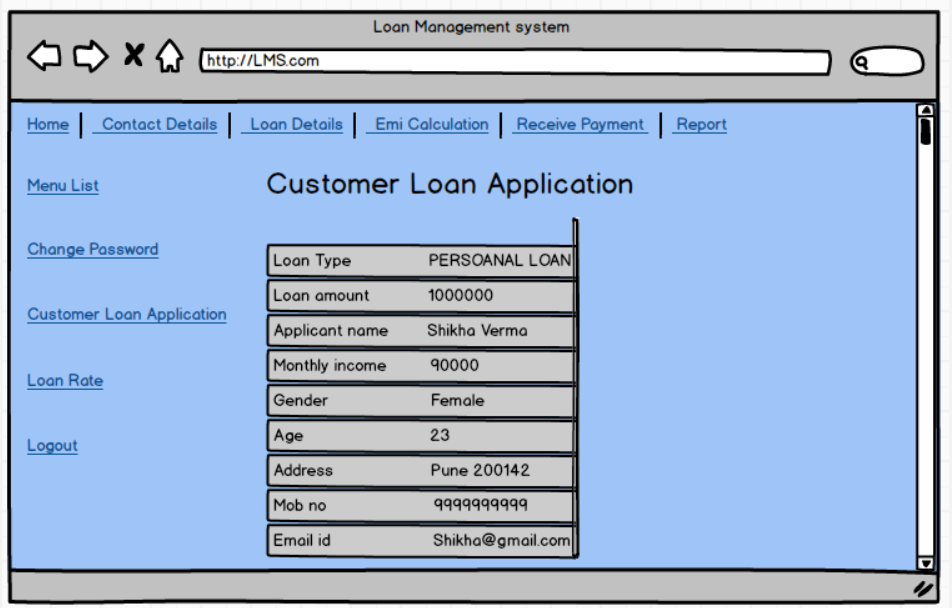


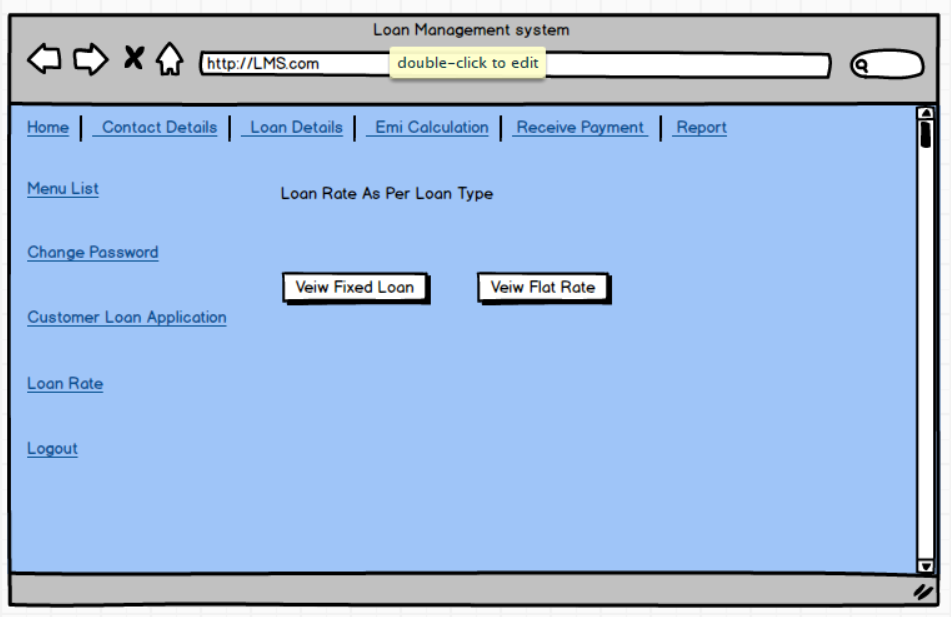
Document 7- Screens and pages

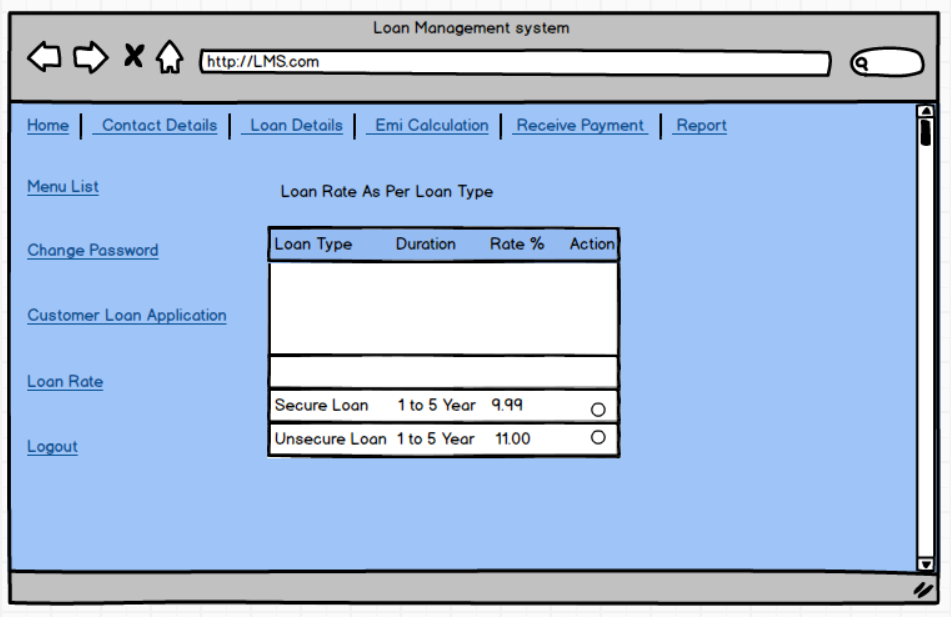


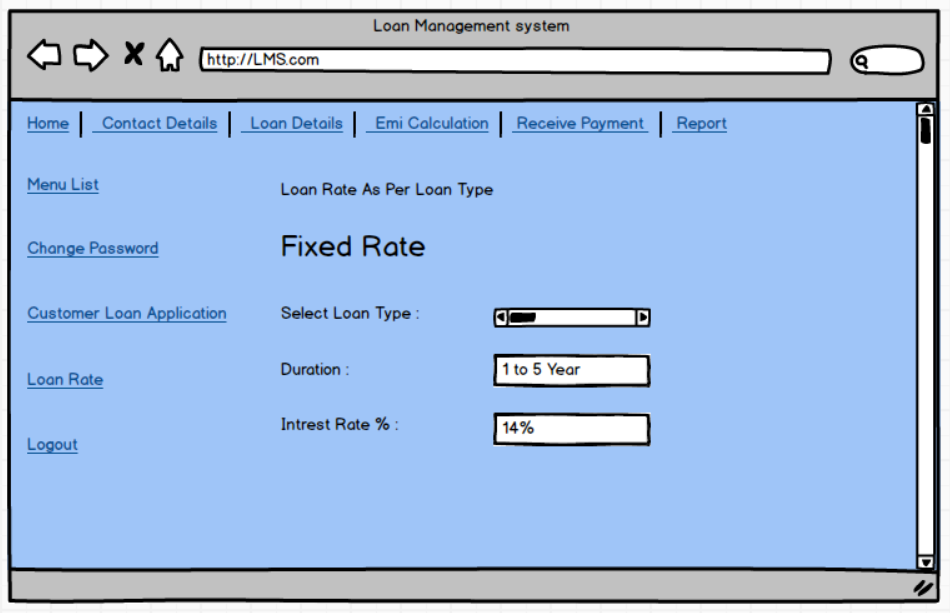


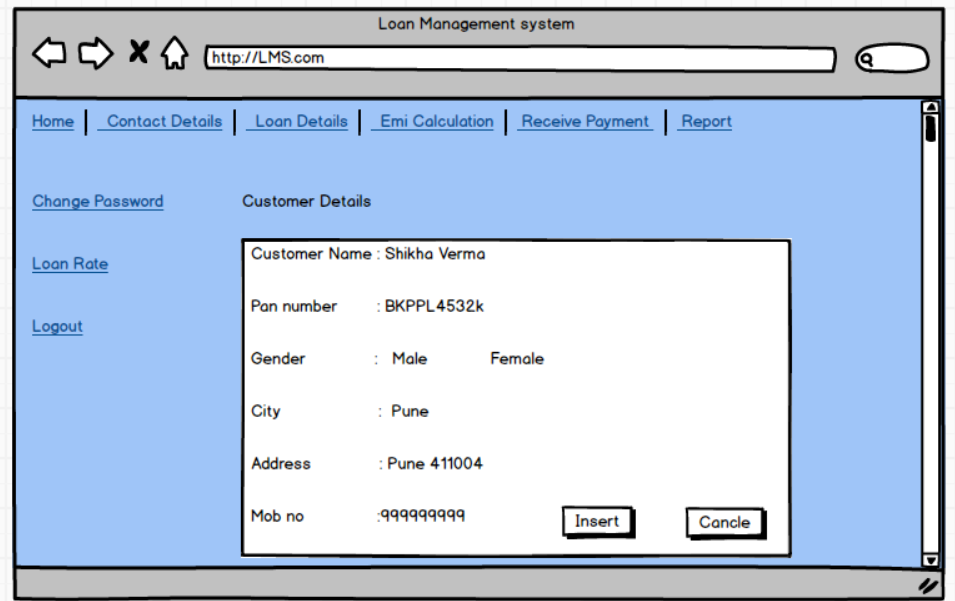


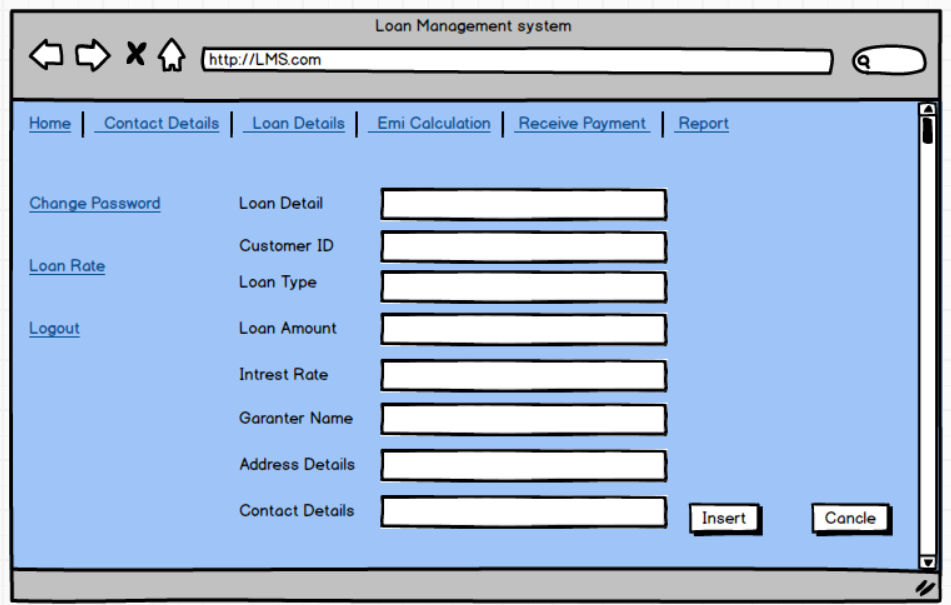


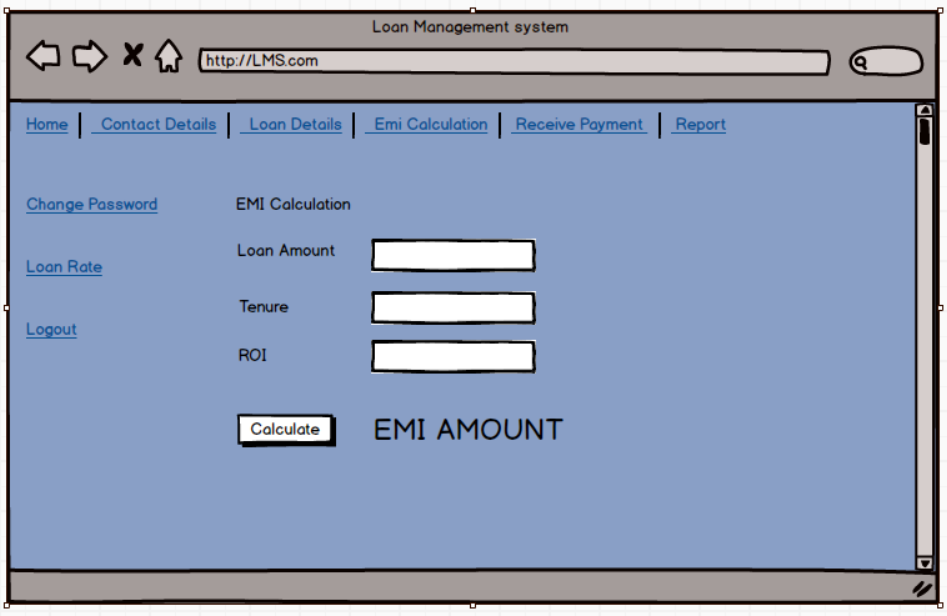


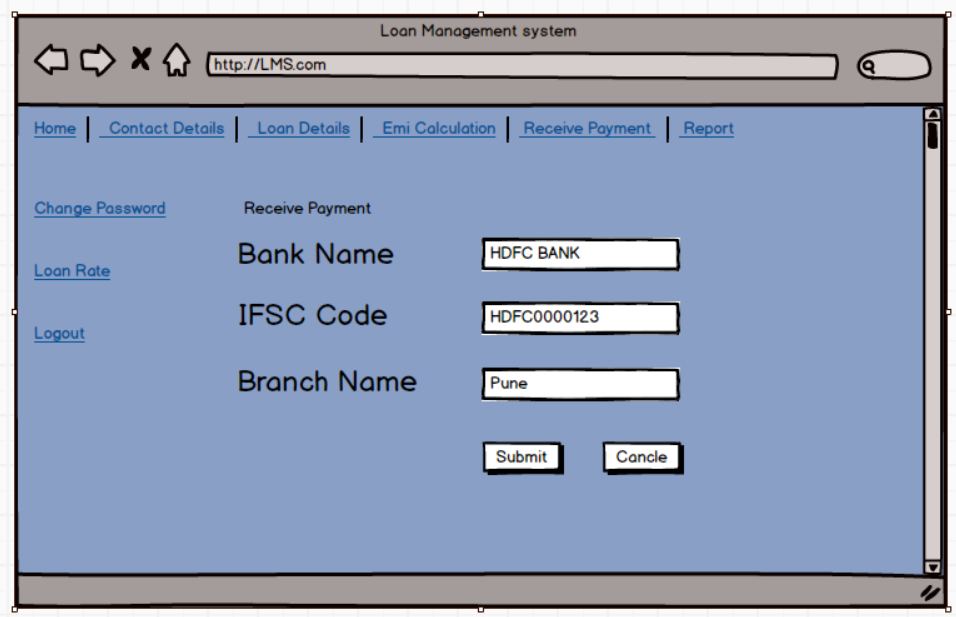












**Document 8- Tools-Visio and Axure**

MS VISIO :  
I used Visio to create diagrams that helped simplify and explain complex processes. Visio allowed me to turn system workflows, use cases, and process steps into clear visuals that everyone could easily understand. Whether I was working with developers, managers, or other stakeholders, these diagrams made it easier to communicate ideas and ensure that everyone was on the same page. I often used it to draw flowcharts and use case diagrams, which showed how different parts of the system connected and how data flowed through them.

AXURE  
With Axure , I created interactive prototypes and wireframes. These were clickable models of websites or applications that stakeholders could interact with to see how the final product would function. By using Axure, I could gather feedback early, before the development process even started, to ensure the design was exactly what the business needed. The tool made it easy to adjust the design based on feedback and show different versions of the user interface. Overall, Axure was great for turning business requirements into visual, testable prototypes that improved communication between the design, development, and business teams. Both tools played a key role in improving project outcomes by providing clear visuals and functional mockups.

**Experience as Business Analyst**

1 Requirement Gathering

MOSCOW Technique:

During the requirement gathering phase , I used the MOSCOW technique to prioritize requirements. This method helped me break down the requirements into four categories: Must-Have, Should-Have, Could-Have, and Won’t-Have. It made it easier to focus on the essential features that had to be delivered while understanding which requirements could be flexible or left out. This approach ensured that everyone was clear about what was critical to the project and what could be adjusted based on time or resources.

Handling Client Unavailability:

At times, the client wasn’t available during the requirement gathering phase, which could slow things down. As a Business Analyst, I took the initiative to find other contacts within the client’s team who could provide the information we needed. This way, I could keep the project moving by getting the necessary details quickly. I reached out to the right people, built good communication with them, and ensured we didn’t lose time waiting for the main client contact to be available.

Requirement Validation Using FURPS Technique:

Once I gathered the requirements, I used the FURPS technique to check if they met all the important standards. FURPS stands for Functionality, Usability, Reliability, Performance, and Supportability . I went through each requirement to ensure it was functional, easy to use, reliable, and could perform well. This validation process made sure that all the requirements were practical and fit the project’s goals.

Managing Duplicated or Repeated Requirements:

During the process, I often came across repeated or duplicated requirements. To keep things clear and organized, I removed these duplicates as soon as I found them. This helped avoid confusion and ensured that the requirements list was clean and easy to follow. I communicated with the stakeholders to make sure they understood why certain requirements were removed and kept everyone on the same page.

Using Prototyping to Refine Requirements:

To make the requirements clearer, I used prototyping . By creating visual models or mockups, I was able to show the client how the system would look and work. This helped them give more specific feedback and clarify their expectations. The prototypes made it easier for everyone to understand the design and functionality of the system before development started. It was a great way to refine the requirements and avoid misunderstandings later on.

2 Requirement Analysis

Drawing UML Diagrams:

During the requirement analysis phase , I created UML diagrams to visually show the system's structure and behavior. These diagrams, like use case diagrams and class diagrams, helped everyone understand how different parts of the system would work together. By using visuals, I made it easier for both the development team and stakeholders to grasp the requirements.

Using Activity Diagrams for Process Flow:

I also made activity diagrams to illustrate the process flow of various functionalities in the system. These diagrams clearly showed how users would interact with the system and how tasks would move from one step to another. This helped us identify any potential issues or areas for improvement early on.

Communicating and Handling Feedback:

After creating the diagrams, I shared them with the team for feedback. Sometimes team members had different opinions, and as a Business Analyst, I listened to their suggestions. If their points were valid, I updated the diagrams accordingly. This teamwork ensured everyone understood the requirements and any technical concerns were addressed.

Preparing BRS and SRS:

I also prepared two important documents: the Business Requirements Specification (BRS) , which outlines the business needs, and the Software Requirements Specification (SRS) , which details the technical requirements. These documents helped ensure that the project met business goals and provided clear guidance for the development team.

3 Design:

Preparing Test Cases from Use Case Diagrams:

During the design phase , I prepared test cases based on the use case diagrams. These test cases outlined specific conditions under which we would test the system to ensure it met the requirements. By directly linking test cases to the use cases, I made sure that every functionality was covered, facilitating a more organized and efficient testing process.

Communicating with Clients:

I also communicated with the client regarding the design and solution documents. This involved discussing how the proposed solution would work and ensuring that it aligned with their expectations. Regular communication with the client helped in obtaining feedback and made sure they were on board with the design choices before moving forward.

Writing Positive and Negative Test Cases:

I made sure to write both positive and negative test cases . Positive test cases checked if the system behaves as expected under normal conditions, while negative test cases ensured that the system correctly handles unexpected or incorrect input. This comprehensive testing approach helped us identify potential issues and made the system more robust.

Ensuring No Test Case is Missed:

I took great care not to miss a single test case, knowing that even a small oversight could have a significant impact on project development later on. By being thorough, I aimed to ensure that all aspects of the system were tested and validated before deployment.

Preparing Test Data:

I also prepared test data needed for testing. This data was crucial for running the test cases effectively and simulating real-world scenarios to see how the system would perform under different conditions. Well-prepared test data helped ensure that our tests were accurate and reliable.

Updating the Requirements Traceability Matrix (RTM):

Finally, I updated the Requirements Traceability Matrix (RTM) to track the status of all requirements. This matrix helped ensure that every requirement was addressed and met during the design and testing phases. It served as a valuable tool for maintaining alignment between the requirements and the testing process, making sure we didn’t overlook anything important.

4. Development:

Organizing JAD Sessions:

During the development phase , I organized Joint Application Development (JAD) sessions to bring together stakeholders and the technical team for collaborative discussions. These sessions helped gather insights and make decisions quickly, ensuring everyone was aligned on the project goals.

Clarifying Queries from the Tech Team:

I also clarified queries from the technical team while they were coding. When they had questions or needed further explanations, I provided the necessary information to keep the development process moving smoothly.

Handling Team Disagreements:

Sometimes, team members disagreed with the concepts discussed in JAD sessions or were uncooperative. As a Business Analyst, I handled these situations gently by having one-on-one discussions with them. I explained how their actions could impact the project and worked to create a positive, collaborative environment within the team.

Referring Diagrams for Coding:

Throughout development, I referred to the diagrams we created earlier to guide the coding process. These diagrams served as important references for the developers, ensuring that the implementation aligned with the design and requirements.

Conducting Regular Meetings:

I conducted regular meetings with the technical team and the client, which could be challenging. Sometimes, team members couldn’t attend, so I recorded the sessions and shared them with those who missed out. I also followed up with one-on-one discussions to address any questions or concerns they might have had.

5 Testing:

Preparing Test Cases from Use Cases:

During the testing phase , I prepared test cases based on the use cases we developed earlier. This ensured that we tested all the functionalities that the system was supposed to deliver.

Performing High-Level Testing:

I performed high-level testing to check the overall functionality of the system. This included testing key features to ensure they worked as intended before moving on to more detailed testing.

Requesting Test Data from the Client:

I requested test data from the client to conduct the tests effectively. This data was necessary to simulate real-world scenarios and validate the system’s performance.

Updating the Requirements Traceability Matrix (RTM):

I updated the Requirements Traceability Matrix (RTM) to track the status of all requirements throughout the testing process. This helped ensure that everything was being tested and met.

Taking Sign-off from the Client:

Once the testing was complete, I took sign-off from the client to confirm that the system met their requirements and was ready for deployment.

Preparing the Client for User Acceptance Testing (UAT):

I prepared the client for User Acceptance Testing (UAT) , ensuring they understood how to test the system and what feedback was needed. This step was crucial for validating that the system met their needs before going live.

6 Deployment:

Forwarding RTM to the Client:

During the deployment phase , I forwarded the Requirements Traceability Matrix (RTM) to the client as part of the project closure document. This ensured that all requirements were documented and met before finalizing the project.

Coordinating End-User Manuals:

I coordinated the completion and sharing of end-user manuals to help users understand how to operate the system effectively. These manuals provided essential guidance and instructions.

Planning Training Sessions:

I planned and organized training sessions for the end users to familiarize them with the system. These sessions were crucial for ensuring that users felt confident and comfortable using the new system.

Ensuring Attendance at Meetings:

I made sure that all candidates attended the training sessions and meetings. I followed up with them to confirm their participation and addressed any scheduling conflicts to ensure everyone was on board.