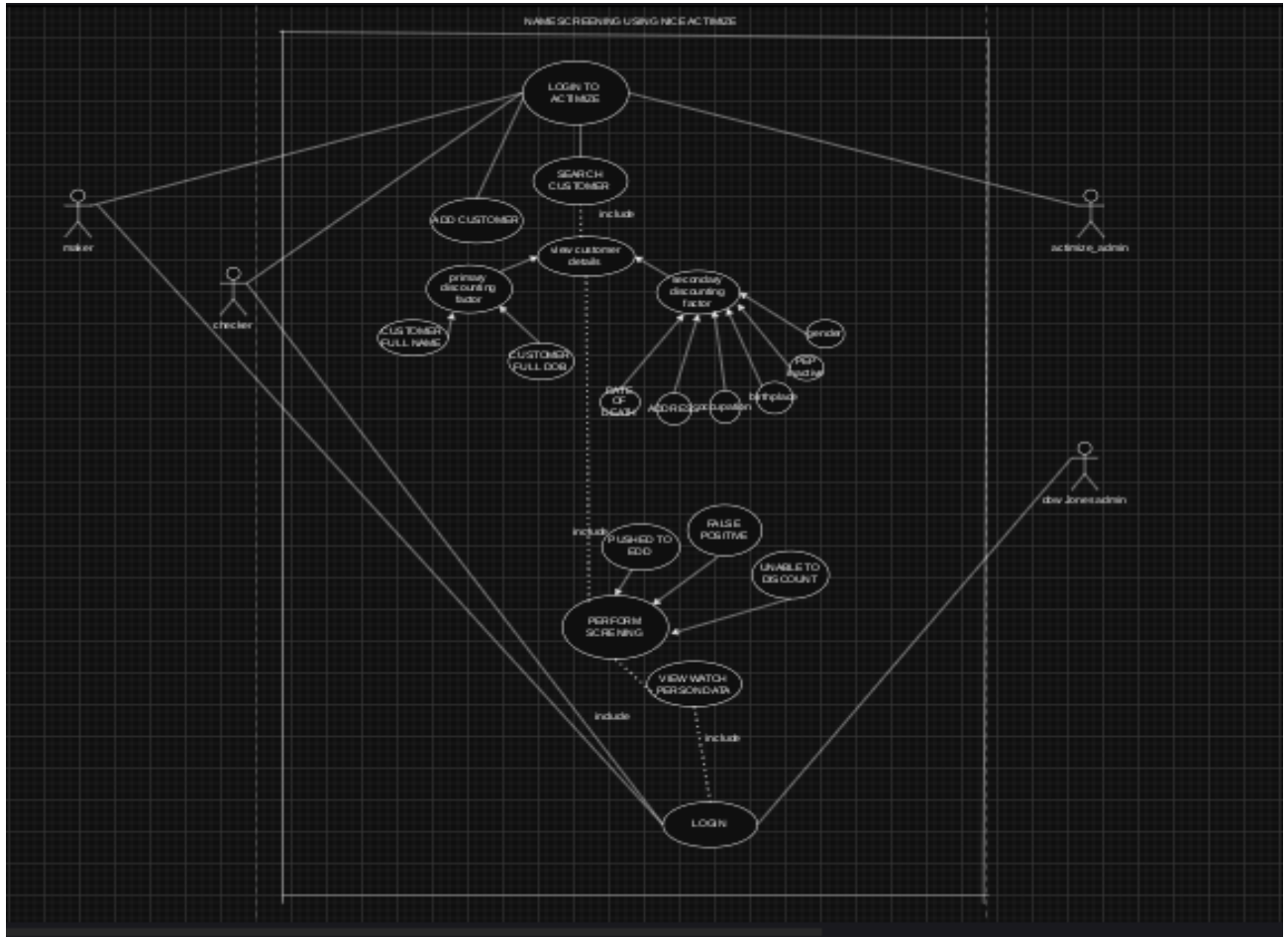


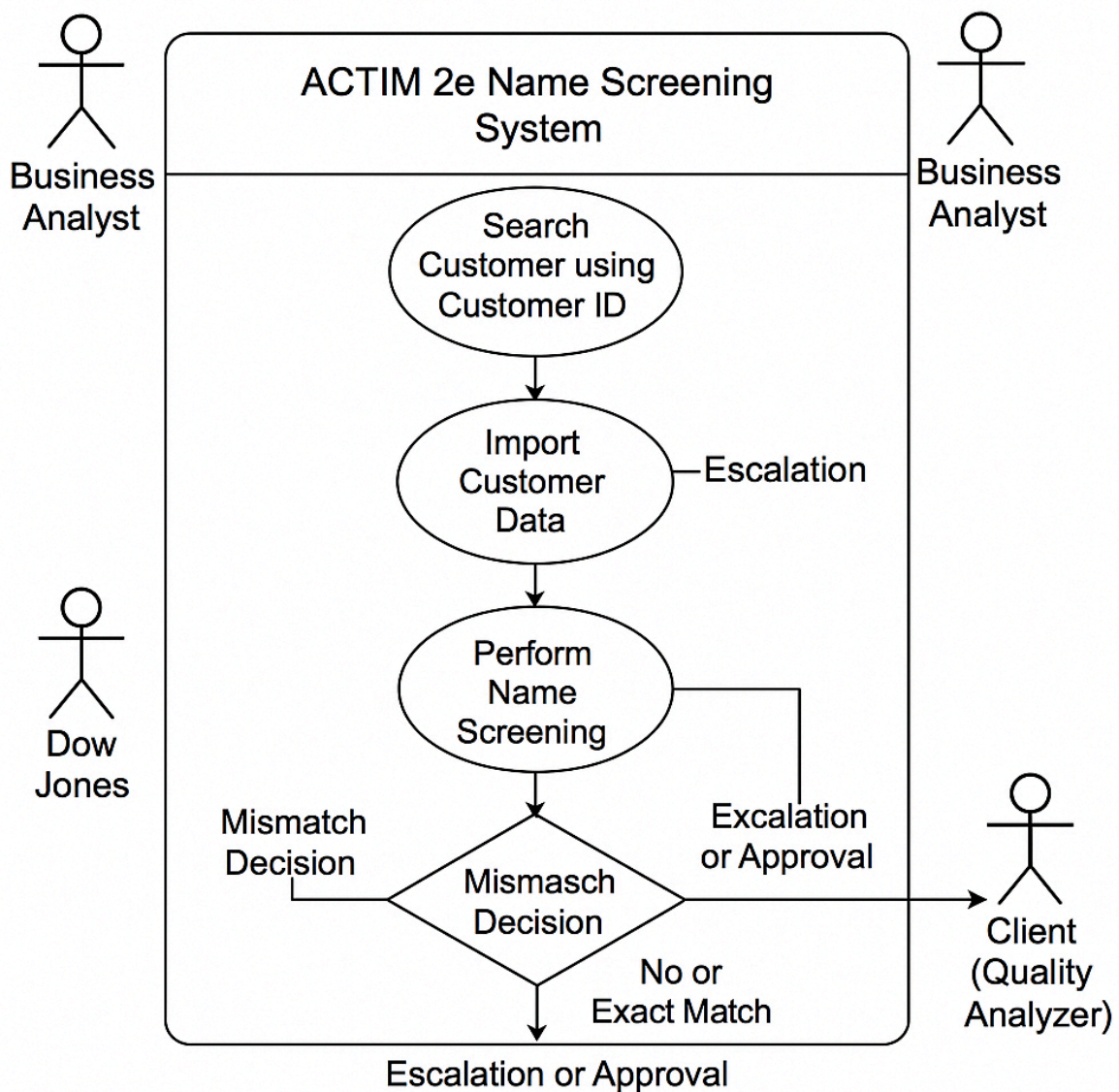
Waterfall Project Part 3/3

by DEBADRITA DAS

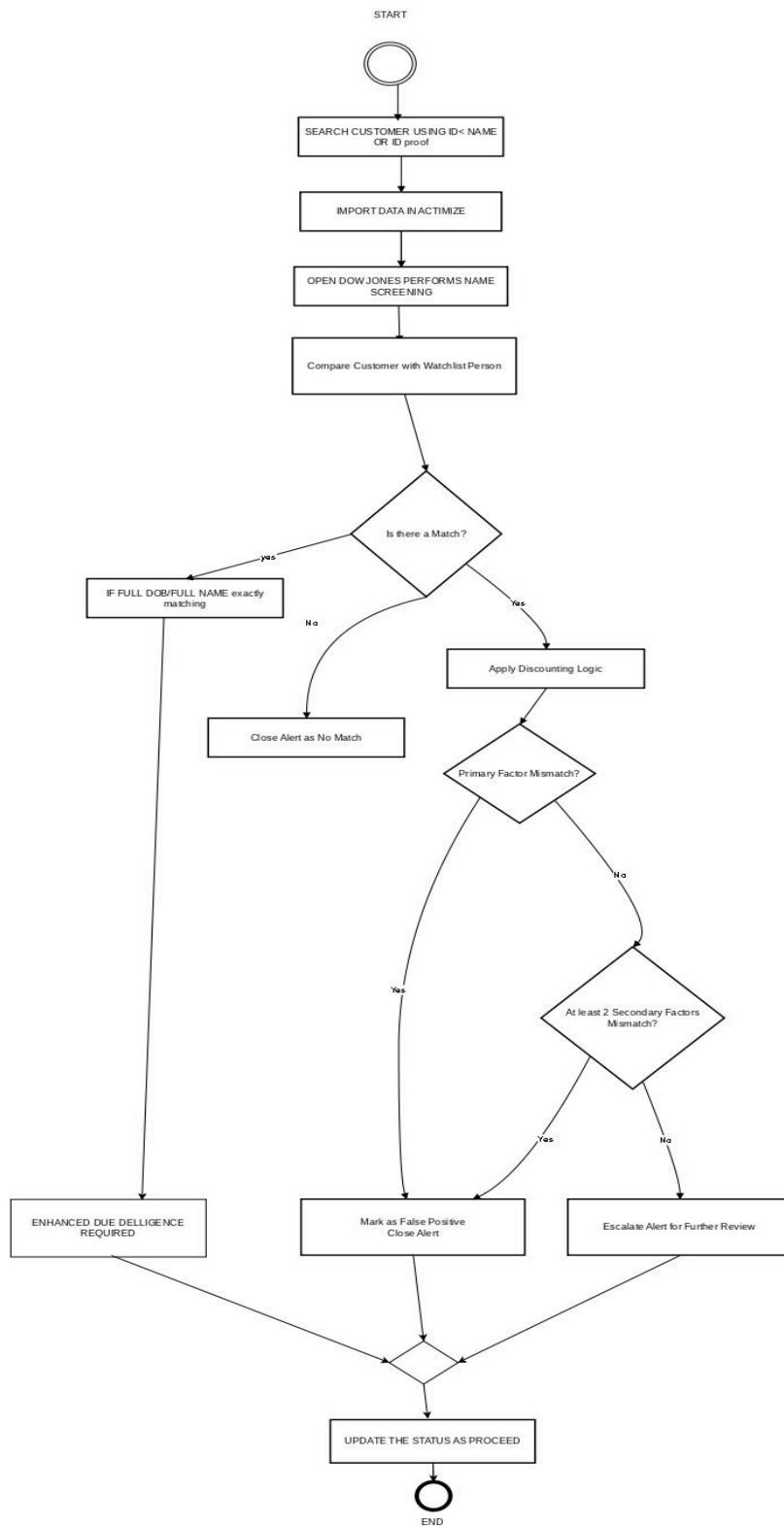
Document -6 Please prepare a use case diagram, activity diagram and a use case specification document.

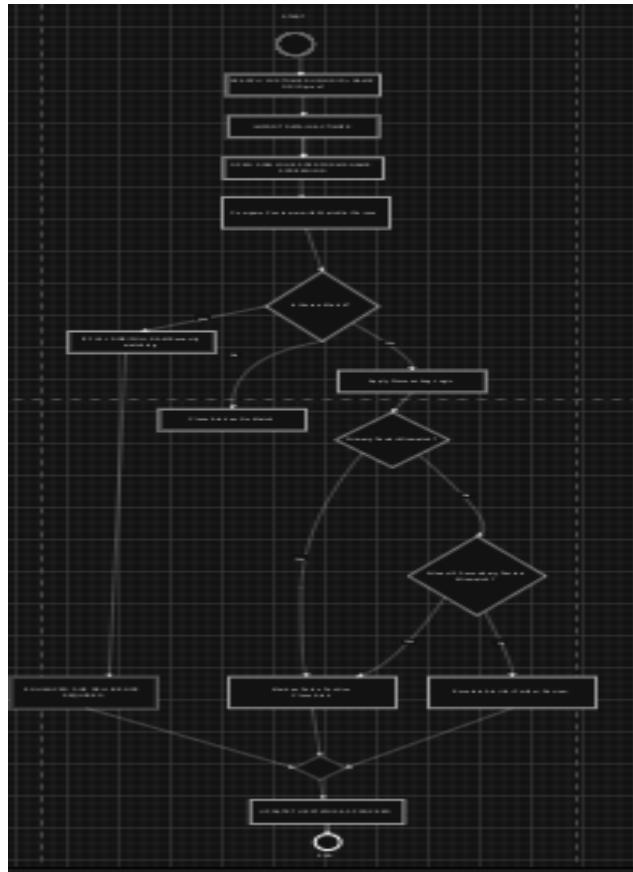


Simplified USE CASE diagram



Activity Diagram for name screening :





Use Case Specification Document:

Use Case Name	CUSTOMER ADDITION BY MAKER
Use Case Description	System allows makers to add customer's KYC document.
Actors	
Primary Actors	Compliance maker, ACTIMIZE SYSTEM ADMIN
Secondary Actors	Database
Basic Flow	1.Teller navigates to the KYC addition page.
	2.Teller enters Customer number.
	3.System validates Customer number.
	4.Teller able to add relevant documents of customer for unique identification of customer.
Alternate Flow	Invalid/Incorrect Customer number:

	System displays an error message for invalid or incorrect customer number.
	Teller tries to re-enter customer number or enters customer name.
Exceptional Flows	System downtime:
	Teller is unable to open KYC addition page.
	System displays maintenance message.
Pre-Conditions	Teller must be registered in the system and authorized to do operations.
	System must be online.
Post-Conditions	Teller is logged in and directed to the dashboard and able to navigate to KYC addition page.
Assumptions	Teller has valid credentials.
Constraints	High loading times during peak hours.
Dependencies	Teller database authentication service
Inputs and Outputs	Inputs: Username, Password, KYC Document information
	Outputs: Dashboard, KYC addition page, Message for successfully adding KYC documents
Business Rules	KYC document must among the government approved list of documents.
	KYC document must be valid for current financial year.
Miscellaneous Information	Ensure Customer name search functionality is available in absence of Customer number.

Use Case Name	Cardless cash withdrawal from ATM
Use Case Description	Allows customer to withdraw cash without debit card from ATM.
Actors	
Primary Actors	Customer, System Administrator
Secondary Actors	Database
Basic Flow	1.Customer opts for cardless withdrawal option on ATM screen.
	2.Customer enters Mobile number and PAN card details.
	3.System validates the entered details.

	4.Customer is asked to enter amount.
	5.System checks balance in customer account as well as cash available in ATM machine.
	6.Once done system dispenses cash to the customer.
Alternate Flow	Invalid mobile number or PAN card details:
	System displays an error message.
	Customer retries to enter mobile number and PAN card details.
Exceptional Flows	Cash unavailable or insufficient cash in the ATM:
	System displays appropriate error message
Pre-Conditions	Customer has linked mobile number and PAN card to bank account.
	Customer account balance and ATM cash availability must be maintained
Post-Conditions	Customer is able to get cash from ATM.
Assumptions	Customer has valid mobile number and PAN card details.
Constraints	Digital illiteracy among customers.
Dependencies	Customer database authentication service.
Inputs and Outputs	Inputs: Mobile Number, Pan Card details
	Outputs: Cash, Error messages
Business Rules	Mobile number, Pard card details, entered amount must meet security requirements.
	Service will be freeze after 3 failed attempts.
Miscellaneous Information	Ensure use of debit card functionality is available

Use Case Name	Unified Payment Interface (UPI) payment enabled
Use Case Description	Customers is able to make outgoing payments via UPI id or UPI linked mobile number of beneficiaries.
Actors	
Primary Actors	Customer, System Administrator
Secondary Actors	Database
Basic Flow	1.Customer opts for UPI to make outgoing payment.
	2.Customer enters UPI id or UPI linked mobile number of beneficiary.
	3.System validates the entered UPI details and displays beneficiary details.
	4.Customer is asked to enter amount.
	5.System checks balance in customer account.
	6.System asks to enter UPI pin.

	7.System validates UPI pin.
	8.Once validation successful, amount will be debited from customer account.
	9.Payment receipt will be generated.
Alternate Flow	Invalid mobile number or UPI details or UPI Pin:
	System displays an error message.
	Customer retries to enter mobile number and UPI details.
Exceptional Flows	System downtime:
	System displays appropriate error message
Pre-Conditions	Beneficiary has UPI linked mobile number and valid UPI Id.
	Customer account balance must be maintained.
Post-Conditions	Payment will be made to beneficiary with help of UPI.
Assumptions	Customer has valid UPI linked mobile number and valid UPI details.
	Beneficiary has valid UPI linked mobile number and valid UPI details
Constraints	Digital illiteracy among application users.
Dependencies	Customer database authentication service.
Inputs and Outputs	Inputs: Mobile Number, UPI details
	Outputs: Payment's success or rejection, Error messages
Business Rules	Mobile number, UPI details, entered amount must meet security requirements.
	Service will be blocked after 3 failed attempts.
Miscellaneous Information	Ensure use of net banking functionality is available.

Use Case Name	Daily UPI outgoing transaction limit
Use Case Description	Restricts daily UPI outgoing transaction above a predefined limit directed by the government.
Actors	
Primary Actors	Bank teller, Customer, System Administrator
Secondary Actors	Database
Basic Flow	1.Customer opts for UPI to make outgoing payment.
	2.Customer enters UPI id or UPI linked mobile number of beneficiary
	3.System validates the entered details.
	4.Customer is asked to enter amount.

	5.System checks balance in customer account as well as daily outgoing payment limit via UPI.
	6.If entered amount is exceeding daily limit then system displays appropriate error otherwise payment will be success.
Alternate Flow	Invalid mobile number or UPI details:
	System displays an error message.
	Customer retries to enter mobile number and UPI details
Exceptional Flows	System downtime:
	System displays appropriate error message
Pre-Conditions	Beneficiary has UPI linked mobile number and valid UPI Id.
	Customer account balance must be maintained.
Post-Conditions	Payment will be accepted or rejected based on daily limit validation.
Assumptions	Beneficiary has valid UPI linked mobile number and valid UPI details.
	Beneficiary has valid UPI linked mobile number and valid UPI details
Constraints	Digital illiteracy among customers
Dependencies	Customer database authentication service.
Inputs and Outputs	Inputs: Mobile Number, UPI details
	Outputs: Payment's success or rejection, Error messages
Business Rules	Mobile number, UPI details, entered amount must meet security requirements.
	Service will be blocked after 3 failed attempts.
Miscellaneous Information	Ensure use of net banking functionality is available.

Use Case Name	Capture Adhar enabled biometric into database
Use Case Description	Bank Teller is able to capture biometrics of customers from Adhar card to bank database.
Actors	
Primary Actors	Bank teller, System Administrator
Secondary Actors	Database
Basic Flow	1.Teller login into the system and navigates to customer information page.
	2.Teller clicks on add Adhaar details.
	3.Teller enters unique Adhaar number of customer.
	4.System validates the Adhaar Number and fetches the biometric data from the repository using API call.

	5.Teller adds customers signature into the database for identification.
Alternate Flow	Invalid Adhaar number or Customer details:
	System displays an error message.
	Teller retries to enter Adhaar number and Customer details.
Exceptional Flows	System downtime:
	System displays appropriate error message
Pre-Conditions	Customer has valid Adhaar card number.
	Customer is registered account holder in the bank.
Post-Conditions	Customer signature is saved into bank's database.
Assumptions	Bank has working API system to fetch the details of customers based on Adhaar card number.
Constraints	Unavailability of technology to capture biometric details.
Dependencies	Customer database API and authentication service.
Inputs and Outputs	Inputs: Adhaar Card Number, Customer details
	Outputs: Signature addition into database, Error messages
Business Rules	Adhaar card number, Customer details and biometric details must meet security requirements.
	Bank Teller should be authorized personal to do the bank operations.
Miscellaneous Information	Ensure all data handling and processing comply with the relevant provisions of the Indian Aadhaar Act.

Document -8 Tools-Visio and Axure

Using Visio and Axure for this comprehensive banking upgrade project allowed for a streamlined approach in designing and implementing both the technical and user-centric aspects of the solutions.

In Visio, I developed intricate system architecture diagrams to illustrate the FLEXCUBE Universal Banking Solution's patch-set upgrade and data migration process. These visualizations ensured that stakeholders and technical teams clearly understood dependencies and workflows, particularly when implementing the Oracle Banking Payments module to offload existing payment interface loads.

On the other hand, Axure was essential in creating interactive prototypes to demonstrate new features like KYC integration, electronic signatures, and mobile banking enhancements, including UPI functionality and cardless ATM withdrawals.

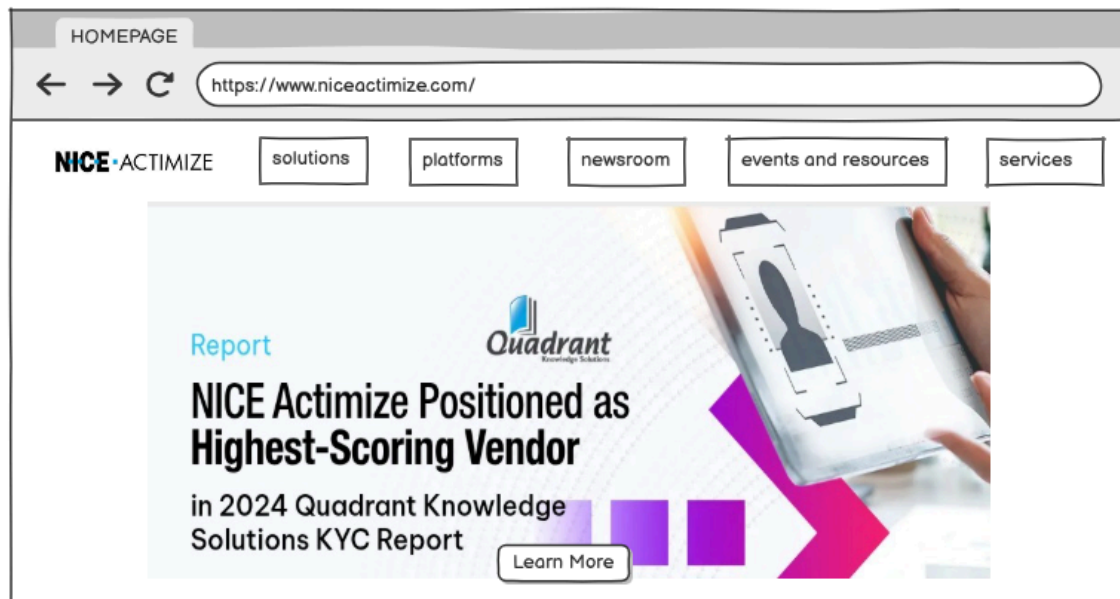
These prototypes enabled iterative feedback, facilitating improvements in usability and compliance alignment. This dual-tool approach fostered effective collaboration across teams, minimized ambiguities, and ensured that project objectives were met efficiently and effectively.

Along with Visio and Axure RP, Balsamiq played a crucial role in the initial design and wireframing phases of the FLEXCUBE Universal Banking Solution upgrade and Oracle Banking Payments implementation. Using Balsamiq, I was able to create intuitive, low-fidelity wireframes that effectively captured the functional requirements of various system modules, ensuring a clear visualization of the user experience before development. The tool facilitated rapid prototyping of key features such as the Know Your Customer (KYC) interface, electronic signature integration, mobile banking application layout, and cardless ATM withdrawal functionality. By leveraging Balsamiq's drag-and-drop elements, I efficiently structured the payment screens to include UPI ID functionality and daily transaction limit settings, ensuring compliance with security guidelines. The iterative design approach allowed stakeholders to review, provide feedback, and make necessary modifications early in the development cycle, reducing potential rework.

Document -7 Screens and Pages

In the attached page design, I have illustrated the flow of a NAME SCREENING, starting from the **Home** page, navigating through the **Login** screen, and proceeding to the **screening** screen, culminating in the transaction authorization process. The design has been created using **Balsamiq**.

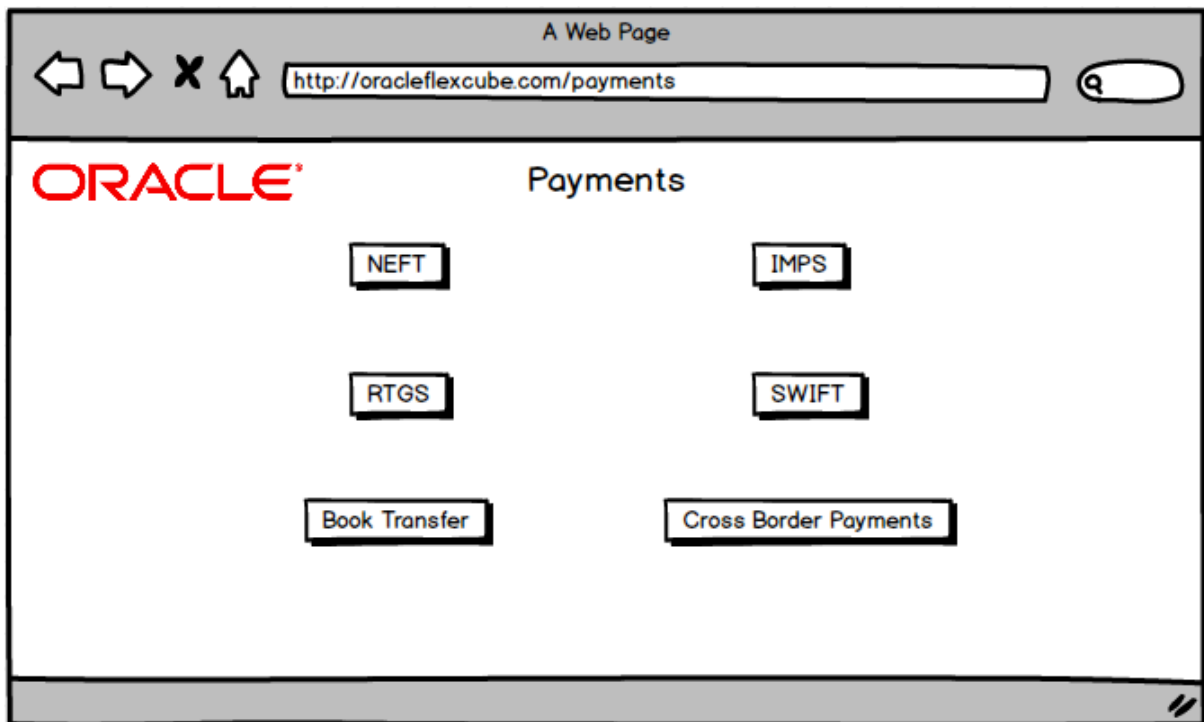
1. Home Page :-



2. Login page :-

The screenshot shows the login page of the NICE Actimize website. The browser's address bar displays <https://marketplace.niceactimize.com/login>. The page features the NICE Actimize logo, a search bar, and a login form with fields for EMAIL and PASSWORD. Below the password field is a "Forgot Password ?" link and a checkbox for "keep me logged in". The "LOGIN" button is located at the bottom left, and the "SIGN UP" button is at the bottom right.

3. Payments page :-



4. NEFT Outgoing Screen page :-

A screenshot of a web browser window titled "A Web Page". The address bar shows the URL "http://www.oracleflexcube.com/payments/neft". The page content features the Oracle logo on the left and the title "NEFT Outgoing Transaction Screen" in the center. Below the title, there is a form with the following fields and values:

Account Number	854237985212543	Branch Code	1003
Customer Id	1278536954	Customer Name	Shravari Das
Amount	12534	Beneficiary Account Number	854237987514238
Currency	INR	Beneficiary Branch Code	1005
Value Date	07/02/2025	Transaction Ref Number	NEFT1003875214398

At the bottom right of the form, there are two buttons: "Save" and "Cancel". The browser window has standard navigation buttons (back, forward, stop, home) and a search icon.

5. Authorization page :-

A Web Page

http://www.oracleflexcube.com/payments/neft/authorize

ORACLE NEFT Outgoing Transaction Screen

Account Number	854237985212543	Beneficiary Account Number	854237987514238
Customer Id	1278536954	Amount	12534
Value Date	07/02/2025	Transaction Ref Number	NEFT1003875214398

Alert

Do ou want to authorize this transaction ?

Yes No

Authorize Cancel

6. Success page :-

A Web Page

http://www.oracleflexcube.com/payments/neft/success

ORACLE NEFT Outgoing Transaction Screen

Account Number	854237985212543	Beneficiary Account Number	854237987514238
Customer Id	1278536954	Amount	12534
Value Date	07/02/2025	Transaction Ref Number	NEFT1003875214398

Alert

Transaction successfully processed.

Okay Cancel

As a Business Analyst, my experience on this project was multifaceted and dynamic, involving active participation across all phases to ensure the successful delivery of a robust solution.

1. Requirement Gathering:

- **Requirement Prioritization Using MOSCOW:**
During the requirement gathering phase, I employed the MOSCOW technique (Must have, Should have, Could have, and Would have) to prioritize and classify the client's requirements. This ensured clarity on the critical needs and helped manage stakeholder expectations effectively.
- **Managing Client Availability Challenges:**
In instances where the client was unavailable for an extended period, I proactively identified alternative points of contact from the client's team. This approach enabled me to obtain the necessary information promptly to avoid delays in the project timeline.
- **Requirement Validation with FURPS:**
I validated the gathered requirements using the FURPS framework (Functionality, Usability, Reliability, Performance, and Supportability). This process helped ensure the requirements were complete, feasible, and aligned with both the business objectives and technical constraints.
- **Handling Duplicated or Repeated Requirements:**
To maintain the integrity of the requirement documentation, I carefully reviewed and eliminated duplicate or redundant requirements. This not only reduced confusion but also streamlined the development process by providing a clear and concise set of deliverables.
- **Prototyping for Enhanced Requirement Clarity:**
I utilized prototyping as a visual and interactive tool to refine and capture more specific requirements. This method facilitated better communication with stakeholders, allowing them to visualize the proposed solutions and provide detailed feedback.

2. Requirement Analysis:

- **Visual Representation with UML Diagrams:**
To analyse and communicate the requirements effectively, I created UML (Unified Modelling Language) diagrams, including use case diagrams, sequence diagrams, and class diagrams. These diagrams provided a clear and structured visual representation of the system's requirements and behaviour, ensuring alignment among stakeholders.

- **Activity Diagrams for Process Flow:**
I also developed activity diagrams to illustrate the end-to-end process flow. These diagrams captured the sequence of operations, decision points, and parallel activities, helping the team understand the functional flow of the system.
- **Collaborative Diagram Review and Modifications:**
Once the diagrams were created, I shared them with the project team for feedback. During this collaborative review process, some team members raised concerns or suggested changes. As a BA, I actively listened to their perspectives, analysed the impact of the suggestions, and made necessary modifications to ensure the diagrams accurately represented the requirements while maintaining consensus.
- **Preparation of BRS and SRS:**
I prepared comprehensive Business Requirement Specification (BRS) and Software Requirement Specification (SRS) documents.
 - BRS focused on capturing high-level business needs, goals, and stakeholder expectations.
 - SRS detailed the functional and non-functional requirements, including technical specifications, constraints, and performance criteria.
 These documents served as a foundational reference for both the development team and stakeholders, ensuring clarity and alignment throughout the project lifecycle.

3. Design

- **From the use case diagrams, we prepare test cases -**
The initial phase of the project involved reviewing and understanding the use case diagrams. These diagrams were essential in identifying all functional and non-functional requirements of the system. Using these diagrams, I meticulously prepared test cases to cover every aspect of the system's behaviour. Each test case was crafted to validate specific scenarios, ensuring complete test coverage of both the migration process and the OBPM payments system configuration along with patchset upgradation.
- **Communication with Client on Design and Solution Documents -**
Frequent communication with the client was critical throughout the project lifecycle. This involved discussing the design and solution documents in detail to ensure alignment with the client's expectations and regulatory requirements. I scheduled regular meetings which

helped to address ambiguities, resolve queries, and incorporate feedback into the solution design and clear communication fostered a collaborative environment and reduced the risk of misunderstandings.

- Write negative test cases as well along with positive test cases -
To ensure robustness and reliability of the system, I wrote both positive and negative test cases. Positive test cases validated expected system behaviour under normal conditions, while negative test cases assessed the system's resilience to invalid inputs and edge cases. This dual approach ensured the system could handle real-world scenarios, including unexpected user actions and data anomalies, without failure.
- Ensuring Comprehensive Test Case Coverage -
Recognizing the critical nature of the project, I ensured that no test case was overlooked. Missing a test case could have led to significant issues during later stages of development or deployment, potentially causing delays or failures. I conducted peer reviews of test cases and collaborated with the Quality Assurance (QA) team to confirm comprehensive coverage of all requirements and scenarios.
- Preparing Test Data for Testing -
An integral part of the testing phase was preparing realistic and comprehensive test data. I created datasets to simulate various user scenarios, transaction types, and system states, ensuring the testing process replicated real-world conditions as closely as possible. Special attention was given to ensuring data integrity and compliance with banking regulations.
- Updating the Requirements Traceability Matrix (RTM) -
To maintain alignment between requirements, design, and testing, I regularly updated the Requirements Traceability Matrix (RTM). This document tracked the status of each requirement, linking it to corresponding design elements, test cases, and test results. Keeping the RTM up to date ensured that all requirements were met and provided a clear audit trail for stakeholders.

4. Development

- Organizing JAD Sessions -
Joint Application Development (JAD) sessions were a critical component of the project. I organized and facilitated these sessions to gather requirements, refine solutions, and ensure alignment among

stakeholders. These sessions were designed to encourage collaboration between the business and technical teams, resulting in a shared understanding of project objectives and deliverables.

- **Clarifying Queries of the Technical Team During Coding -**
Throughout the coding phase, I served as the primary point of contact for the technical team, addressing their queries and providing clarifications on requirements. This ensured that the development process remained on track and aligned with the business objectives. By promptly resolving ambiguities, I minimized potential rework and kept the team's productivity high.
- **Managing Team Dynamics During JAD Sessions -**
There were instances where team members disagreed with concepts or were hesitant to cooperate during JAD sessions. As a BA, I handled these situations tactfully by engaging in one-on-one discussions with those individuals. I explained the potential impact of their actions on the project's success and emphasized the importance of collaboration. My efforts created a healthy and cooperative environment within the team, fostering better teamwork and alignment.
- **Referring to Diagrams for Coding the Unit -**
To ensure the technical team had the necessary resources for development, I referred to detailed diagrams and other documentation that outlined system workflows, data flows, and architecture. These diagrams were instrumental in guiding the team during the unit coding process, ensuring that development adhered to the agreed-upon design.
- **Conducting Regular Meetings with Technical Team and Client -**
Regular meetings with the technical team and the client were an essential aspect of the project. These meetings helped maintain alignment, address challenges, and ensure progress transparency. While it was challenging to coordinate schedules due to some team members being unavailable, I mitigated this by recording the sessions and sharing them with those who missed the meetings. Additionally, I conducted one-on-one discussions with the absent members to bring them up to speed and address their concerns, ensuring that no information gaps existed.

5. Testing

- **Preparing Test Cases from Use Cases -**
The process begins with thoroughly analysing the use cases to derive comprehensive test cases. Use cases provided by stakeholders or derived from business requirements are translated into step-by-step scenarios. Each test case is:
 - Mapped to the corresponding business requirements.
 - Designed to validate the end-to-end functionality, covering both expected and edge-case scenarios.
 - Reviewed collaboratively with technical teams and stakeholders to ensure accuracy and completeness.

- **Performing High-Level Testing -**
The BA plays a critical role in conducting high-level testing to verify the overall functionality and integration of the system. During this phase:
 - Basic sanity tests are performed to ensure the environment is stable post-migration and patchset upgrades.
 - Payments system workflows are validated to ensure there are no interruptions in critical operations like transaction processing and reconciliation.
 - Any anomalies or gaps are documented and communicated to the development team for immediate resolution.

- **Requesting Test Data from the Client -**
Realistic and accurate test data is essential for effective testing. The BA liaises with the client to:
 - Understand the specific data requirements for different scenarios (e.g., various transaction types, user roles, account setups).
 - Ensure compliance with data privacy and security regulations.
 - Validate the provided data for accuracy and completeness before integrating it into the test environment.

- **Updating the Requirements Traceability Matrix (RTM) -**
As a BA , I ensured that all requirements are accounted for and validated through:
 - Maintaining an updated RTM to track the coverage of requirements during testing.
 - Mapping each test case to its respective requirement to ensure completeness.
 - Highlighting any missing or out-of-scope requirements and discussing them with stakeholders.

- Taking Signoff from the Client -
Once high-level testing is completed, As a BA, I:
 - Presented the test results, along with identified defects and resolutions, to the client.
 - Coordinated discussions to address any open issues or concerns.
 - Secured formal signoff from the client, indicating that the testing phase is complete and satisfactory.
- Preparing the Client for User Acceptance Testing (UAT) -
To ensure a smooth UAT phase, As a BA, I:
 - Conducted training sessions or provides detailed documentation to help the client's team understand the new environment and system changes.
 - Assisted in setting up the UAT environment and prepares test scenarios tailored to the client's use cases.
 - Provided ongoing support during UAT, addressing any issues and ensuring the client is confident with the changes.

6. Deployment

- Forwarding RTM (Requirements Traceability Matrix) to the Client:
 - I prepared the RTM, ensuring all requirements were accurately documented and linked to corresponding deliverables.
 - I collaborated with stakeholders to verify that all client requirements were met during the project lifecycle.
 - I also ensured the RTM was forwarded to the client as part of the project closure document, providing a comprehensive audit trail of project activities and deliverables.
- Coordination for End-User Manuals:
 - I worked closely with the technical team and subject matter experts to create comprehensive end-user manuals.
 - I reviewed the manuals to ensure they were user-friendly, detailed, and aligned with client requirements.
 - I also coordinated with the client to finalize and share the manuals, ensuring all stakeholders had access to the documentation.
- Planning and Organizing Training Sessions:
 - I developed a detailed training plan to equip end users with the necessary knowledge and skills for the new environment and systems.

- o I identified key training objectives and tailored the sessions to address specific client needs, such as system navigation, troubleshooting, and operational workflows.
- o I coordinated with trainers and SMEs to deliver engaging and interactive training sessions.
- o I also ensured training materials were comprehensive and aligned with the new systems and configurations.

- Ensuring Meeting Attendance:
 - o I organized regular meetings and training sessions, ensuring all relevant stakeholders were invited.
 - o I sent timely reminders and followed up with participants to confirm attendance.
 - o I monitored participation and addressed any challenges that could prevent stakeholders from attending.
 - o I provided meeting minutes and actionable items post-session to ensure alignment and accountability.