Detail study and analysis of Requirements elicitation of Bank Management System:

The scope of a bank management system project defines the boundaries and limitations of what the project will deliver. A detailed project scope will help to ensure that the project is completed within budget and on time, and that it meets the needs of the stakeholders. The following is an example of the project scope for a bank management system:

1. Objectives: The main objective of the project is to develop a secure and user-friendly system for managing customer accounts and transactions for a bank.
2. Functionality: The system will provide customers with the ability to perform various transactions, such as deposit, withdrawal, and fund transfer. The system will also support loan management, reporting and analytics, and security features.
3. Platform: The system will be developed as a web-based application, accessible through a browser.
4. User interface: The user interface of the system will be user-friendly and intuitive, with easy navigation and a clear layout.
5. Integration: The system will integrate with existing bank systems, such as the customer database and payment gateways.
6. Security: The system will be designed with robust security features to protect customer information and prevent unauthorized access.
7. Standards and regulations: The system will be developed in accordance with relevant standards and regulations, such as Know Your Customer (KYC) and Anti-Money Laundering (AML) requirements.
8. Deliverables: The final deliverables of the project will include the completed bank management system, documentation, and training materials.
9. Timeline: The project timeline will be estimated based on the scope and complexity of the project, and will be agreed upon by all stakeholders.
10. Budget: The project budget will include all costs associated with development, testing, deployment, and maintenance of the system.

The project scope should be clearly defined and agreed upon by all stakeholders to ensure that the project is completed successfully. Any changes to the scope should be evaluated and approved through a formal change management process.

Functional Requirements:

1. Customer account management: The system should allow customers to create, manage and access their bank accounts, including viewing account information and transaction history.
2. Transactions: The system should enable customers to perform various transactions such as deposit, withdrawal, fund transfer, and bill payments.
3. Loan management: The system should handle loan applications, disbursements, and repayments.
4. Reporting and analytics: The system should generate reports on customer behavior, transactions, and performance indicators.
5. Security: The system should provide robust security features to protect customer information and prevent unauthorized access.

Non-functional Requirements:

1. User-friendliness: The system should have a user-friendly interface to make it easy for customers to perform transactions and access their account information.
2. Performance: The system should be fast and reliable, with minimal downtime.
3. Scalability: The system should be able to handle an increasing number of customers and transactions.
4. Data privacy: The system should protect the privacy of customer data and comply with relevant regulations and standards.
5. Accessibility: The system should be accessible to customers with disabilities, and comply with relevant accessibility standards.

User Interface Requirements:

1. Navigation: The system should have a clear and intuitive navigation structure, with easy access to all functions and features.
2. Design: The user interface should have a modern and attractive design, with a consistent layout and visual style.
3. Responsiveness: The system should be responsive and adapt to different screen sizes and devices.
4. Feedback: The system should provide clear and concise feedback to users, including error messages and confirmation messages.
5. Customization: The system should allow customers to customize their account settings and preferences.

These functional and non-functional requirements, along with the user interface requirements, will form the basis of the requirements specification for the bank management system. The requirements will be used to guide the development and testing of the system, and to ensure that the final product meets the needs of the stakeholders.

Level-0-Use Cases:

1. Customer Account Creation: The customer can create a new bank account by providing personal and contact information.
2. Login: The customer can log in to the system to access their account information and perform transactions.
3. View Account Information: The customer can view their account information, such as account balance and transaction history.
4. Deposit: The customer can deposit funds into their account.
5. Withdrawal: The customer can withdraw funds from their account.
6. Fund Transfer: The customer can transfer funds to other accounts within the same bank or to accounts in other banks.
7. Bill Payment: The customer can pay bills through the system.
8. Loan Application: The customer can apply for a loan through the system.
9. Loan Management: The system will handle loan applications, disbursements, and repayments.

Level 1--Use Cases:

1. Customer Account Management: This use case provides a more detailed description of the customer account creation and login use cases, including additional requirements such as password management and account security.
2. Transactions: This use case provides a more detailed description of the deposit, withdrawal, fund transfer, and bill payment use cases, including requirements for transaction history and statement generation.
3. Loan Management: This use case provides a more detailed description of the loan application and loan management use cases, including requirements for loan approval and disbursement processes.
4. Reporting and Analytics: This use case provides a more detailed description of the reporting and analytics functionality, including requirements for generating customer behavior reports and performance indicators.
5. Security: This use case provides a more detailed description of the security requirements for the system, including requirements for data encryption and user authentication.

These use cases provide a clear and detailed description of the functionalities to be implemented in the bank management system. They can be used to guide the development and testing of the system and to ensure that all requirements are met.

Pre-conditions and post-conditions are important components of use cases that describe the system state before and after an interaction with the system. For a bank management system, some common use cases and their pre and post conditions are:

Account opening:

Pre-condition: Customer provides personal and identification details.

1. Post-condition: Account is created and available for transactions.

Deposits:

Pre-condition: Customer must have a valid account.

1. Post-condition: Account balance is updated with the deposited amount.

Withdrawals:

Pre-condition: Customer must have a valid account and sufficient balance.

1. Post-condition: Account balance is updated with the withdrawn amount.

Fund transfer:

Pre-condition: Customer must have a valid account and sufficient balance.

1. Post-condition: Account balance is updated with the transferred amount, and the receiving account balance is updated with the received amount.

Loan processing:

Pre-condition: Customer must have a valid account, meet loan eligibility criteria and provide loan application details.

1. Post-condition: Loan application status is updated and loan disbursal is done if approved.

Note: The pre and post conditions may vary based on the specific business rules, assumptions and challenges of the bank management system.