**Functional Requirements**

1. **Customer Management**
   * **Add/Remove/Edit Customer**: Ability to create new customer profiles, delete existing ones, and update information.
   * **Customer Types**: Support for different customer subclasses such as Individual, Company, Bank, Brokerage, Crypto Wallet, and Government.
2. **Account Management**
   * **Account Creation/Deletion**: Ability to create and delete accounts linked to customers.
   * **Edit Account Information**: Modify existing account details like account number, balance, and currency.
   * **View Account Details**: View details of all accounts or a specific account, including transaction history.
3. **Transaction Handling**
   * **Record Transactions**: Log every transaction, including amount, date, source, and destination accounts.
   * **Update Balances**: Automatically update account balances after each transaction.
   * **Transaction Validation**: Ensure that transactions are valid under business rules, such as sufficient balance and account status.
4. **Reporting**
   * **Account Statements**: Generate statements for any account over a specified period.
   * **Customer Reports**: Report on customer activity and account summaries.
   * **Transaction Logs**: Detailed logs of all transactions for auditing and troubleshooting.
5. **User Interface**
   * **Dashboard**: Main dashboard displaying key metrics such as total balance, recent transactions, and alerts.
   * **Navigation**: User-friendly menus and navigation to access various parts of the application.
6. **Security**
   * **Authentication and Authorization**: Manage user access based on roles and permissions.
   * **Data Encryption**: Encrypt sensitive data in transit and at rest.
   * **Activity Logging**: Log user activities for security auditing.

**Non-Functional Requirements**

1. **Performance**
   * **Response Time**: System should handle transactions and queries within a few seconds.
   * **Concurrency**: Support multiple users accessing the system simultaneously without performance degradation.
2. **Reliability**
   * **Availability**: System should be operational during defined hours with minimal downtime.
   * **Data Integrity**: Ensure data is accurately recorded and maintained without errors.
3. **Usability**
   * **User Experience**: Intuitive and easy-to-navigate user interface.
   * **Accessibility**: Design to be accessible to users with disabilities.
4. **Scalability**
   * **Load Handling**: Ability to scale up to handle growth in data volume and user load.
   * **Hardware Compatibility**: Ensure compatibility with Raspberry Pi configurations and performance limitations.
5. **Security**
   * **Local Network Security**: Implement measures to secure the application within a local network environment.
   * **Compliance**: Ensure compliance with relevant legal and regulatory requirements for data handling and privacy.
6. **Maintainability**
   * **Modularity**: Design the application with clear modularity to facilitate maintenance and future updates.
   * **Documentation**: Provide detailed documentation for system setup, configuration, and day-to-day operations.
7. **Deployment**
   * **Simplicity**: Easy deployment process on a Raspberry Pi, including an automated setup script.
   * **Recovery**: Implement backup and recovery procedures to handle system failures.

**Final Steps**

* **Review and Validation**: Have stakeholders review the requirements for completeness and feasibility.
* **Prioritization**: Prioritize the requirements based on business needs and development resources.

These requirements will provide a comprehensive foundation for the development of your Transaction Tracker app, guiding both technical specifications and user expectations.

**Follow-up Questions:**

**Q1:** How will you ensure that your system remains responsive and fast as the number of transactions and users grows?

**Q2:** What specific technologies and tools do you plan to use for data encryption and security measures in your application?

**Q3:** How will you handle version control and feature branching during the development of your application?