These are five epics and corresponding user stories for a net banking web application project, following the Scrum framework:

Epic 1: User Registration and Authentication

1. As a new customer, I want to be able to register an account with the net banking web application so that I can access my account online.

2. As a registered user, I want to be able to securely log in to my net banking account using my username and password.

3. As a registered user, I want to be able to reset my password if I forget it, so that I can regain access to my account.

Epic 2: Account Management

1. As a customer, I want to be able to view my account details, such as my balance, transaction history, and personal information, so that I can keep track of my finances.

2. As a customer, I want to be able to update my personal information, such as my address or phone number, so that the bank has accurate contact details for me.

3. As a customer, I want to be able to manage my beneficiaries, such as adding or removing them, so that I can easily transfer funds to trusted recipients.

Epic 3: Fund Transfers

1. As a customer, I want to be able to transfer funds between my own accounts, so that I can manage my finances more efficiently.

2. As a customer, I want to be able to initiate fund transfers to other bank accounts, both within the same bank and to external banks, so that I can make payments or send money to others.

3. As a customer, I want to receive notifications or email confirmations for successful fund transfers, so that I can keep track of my transactions.

Epic 4: Bill Payments

1. As a customer, I want to be able to pay my utility bills, such as electricity or water bills, online through the net banking web application, so that I can avoid the hassle of physical payments.

2. As a customer, I want to be able to schedule recurring bill payments, so that I can automate the payment process for regular bills.

3. As a customer, I want to receive notifications or email confirmations for successful bill payments, so that I have a record of my transactions.

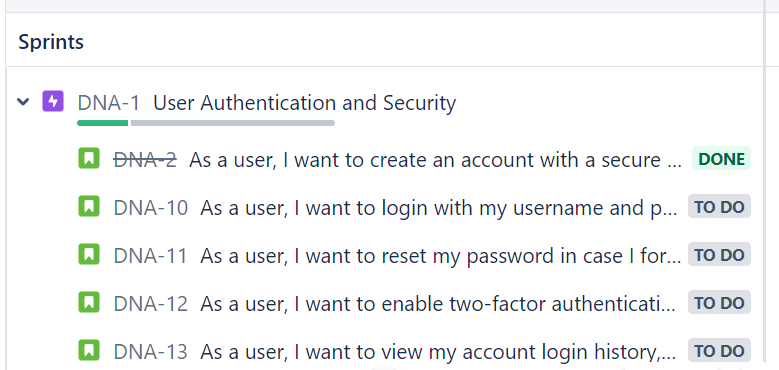
Epic 5: Customer Support

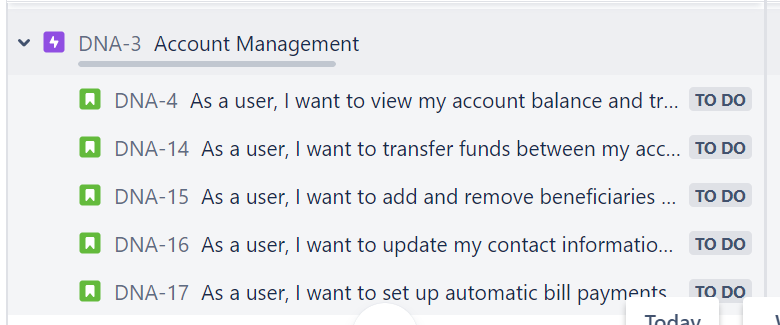
1. As a customer, I want to be able to access a knowledge base or FAQs section on the net banking web application, so that I can find answers to common queries.

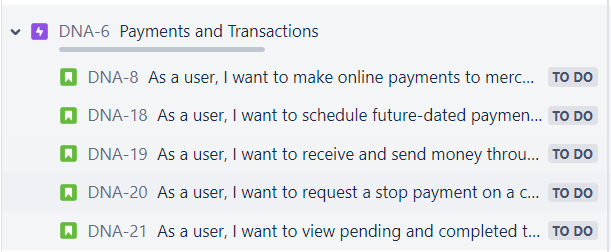
2. As a customer, I want to be able to initiate a live chat with a customer support representative for immediate assistance with any issues or concerns.

3. As a customer, I want to be able to submit feedback or complaints through the net banking web application, so that the bank can address my concerns.

These user stories cover a range of functionalities required for a net banking web application, focusing on user registration, authentication, account management, fund transfers, bill payments, and customer support. Remember that these are just examples, and the actual user stories for your project may vary based on specific requirements and priorities.







To create a backlog in Jira using the Scrum framework for a net banking web application project. Here's an example of how you can structure the backlog:

1. User Stories:

- As a user, I want to be able to create an account.

- As a user, I want to be able to log in to my account securely.

- As a user, I want to be able to view my account balance.

- As a user, I want to be able to transfer funds between my accounts.

- As a user, I want to be able to view transaction history.

- As a user, I want to be able to pay bills online.

- As a user, I want to receive notifications for account activities.

- As a user, I want to be able to update my personal information.

2. Epics:

- Account Management

- Authentication and Security

- Account Balance

- Funds Transfer

- Transaction History

- Bill Payment

- Notifications

- User Profile Management

3. Tasks and Sub-tasks:

- Account Management:

- Design account creation form

- Implement account creation functionality

- Create account validation rules

- Add account creation success message

- Authentication and Security:

- Design login page

- Implement login functionality

- Implement password encryption

- Implement two-factor authentication

- Account Balance:

- Retrieve account balance from the database

- Display account balance on the dashboard

- Implement real-time balance update

- Funds Transfer:

- Design funds transfer form

- Implement funds transfer functionality

- Validate available balance before transfer

- Send transfer confirmation email

- Transaction History:

- Retrieve transaction history from the database

- Display transaction history on the dashboard

- Implement pagination for transaction history

- Bill Payment:

- Design bill payment form

- Implement bill payment functionality

- Integrate with payment gateway

- Send payment receipt email

- Notifications:

- Set up email notification system

- Send account activity notifications

- Allow users to manage notification preferences

- User Profile Management:

- Design user profile page

- Implement profile update functionality

- Implement password change functionality

- Implement profile picture upload feature

