****

**Project 1**

**Application Software as well a Software: Next Generation AI-Pay (NGA Pay)-** An enhanced application software or a software allowing users to make online transactions between people, business and (other entities – Organization and operations) using AI techniques.

**Core value of the application or the software:** Pay with Freedomand 100% transparency.

**Key features of the application software: (10 Features)**

>Money Transfers – Effortlessly send and receive money transfer in real time basis from any region around the world.

>Availability of multiple payment methods – Credit card, Debit cards, international debit and credit cards and bank transfer.

>Availability of multiple currency – Users are allowed to transfer money in any currency based on the country and the region.

>Bio metric authentication – Facial, voice and finger prints analysis replacing the traditional fashioned Pin or password style to make the application more user friendly.

>Real Time transaction tracking – Tracking of user payments in real time basis.

>Intelligent Fraud Detection – Preventing Fraudulent activities with higher accuracy compared to other payment application available in the market

>Voice Activated payments – Making Transaction hands free using NLP technologies (For example user can say: NGA please pay my electricity and gas bill)

>Big Data Analytics platform: An application software which can remember more than 100 years of transaction history for any user.

>High level security: Enhanced level security to protect the application software from any vulnerabilities, threats and attacks.

>Intrusion Detection Systems to find any threats once installed in any operating system before configuring the user’s details

**Potential Benefits: (5 benefits)**

Faster payments: Using low internet data usage but making payments in less than a second while connected with WIFI, mobile and satellite network connection.

Increased security: Using Biometric authentication, enhanced level security and IDS system to find any threats.

Compatible with various operating system: Windows, Mac, Android, IOS, Chrome OS and Linux.

Reduce Cost and earning coins: No Fees for doing any transaction which usually gets charged while doing the payment from any banking or other payment application. Earning coins to get cash back in the user’s bank account instead of promotional gifts.

Low battery consumption: Usage of minimal battery or CPU and memory consumption for any operating system.

**UI of the application software:**

**Dashboard:**

>Overview to view the balance: To view the available balance in the user’s account

>Quick action using natural language and machine learning methodologies (Voice Interactive system)

>Transaction details: Recent and old transaction performed by the user

>Notification Alerts: About the payments, important updates and coins earned but the user

Transaction screen:

>Contact selection: User can choose the contact, enter phone number and email address by the virtual number pad.

>Amount Input: A field to enter the amount by the virtual number pad

>Currency selection: User can select the currency based on the region

>Payment Method: User can select the source (Like Bank account, Debit and credit card)

Payment confirmation:

>Transaction review screen: User can review the existing and new transaction

>Transaction completion screen: User can receive a confirmation once the transaction gets completed.

Account Management:

>Linked account and cards: User can see the linked bank account and cards. Option to edit the payment methods.

>Fund transfer: Allows user to move funds to their exiting and new bank accounts

>Transaction history: A user can have detailed overview of the exiting banking transaction for more than 100 years and new transaction done

Bill payments and recharge options:

>Bill categories: User can pay the utilities, mobile, internet, electricity, gas, water meter, maintenance and other charges.

>Saved Billers: Existing service provider

>Recharge option: User can recharge their mobile, internet data plan and utilize other prepaid service

QR code and Voice Recognition Payment technology:

>Scan to pay option

>Generate QR code and Using voice recognition payment technology

**Security features:**

>Two factor authentication Pin and Biometrics protection

Settings and support:

>Profile settings: Option to update and edit user information such as Name, phone number and email address)

>Security settings: To enable and disable biometrics, changes in the Pin or password and manage two factor authentication

>Customer support or Help: Availability of knowledge-based articles, live chat option and helpline to contact the customer care.

>Language and Region settings option.

**Epic: Set up JIRA project and define milestones**

**Story 1: (Developing a Flexible payment system that supports various payment methods)**

Defining the Payment Methods:

Identifying and prioritizing the payment methods (Example: credit cards, debit cards, digital wallets and bank transfers).

Consideration of regional preferences – Making of the choice of the payment methods. (Example: credit cards, debit cards, digital wallets, bank transfers and dealing with different currencies).

Choosing the Payment Gateway:

At first evaluating and selecting a reliable payment gateway that supports the desired payment methods and offers features like fraud prevention, chargeback management, and global reach. Consider factors such as fees, integration complexity, and customer support.

Integration of the Payment Gateway:

Following the payment gateway's integration guidelines to integrate its update API into the application. Implementing secure communication channels to protect sensitive payment data and users.

Handling Payment Requests of the users:

Developing a user interface to allow users to input payment information. Validating the input data to ensure accuracy and prevent errors. Forwarding the validated payment request to the payment gateway and to the bank

Store Payment Information:

Storing payment information (Example: Existing contact and new contacts) for future transactions.

Implement Security Measures:

Use encryption to protect sensitive payment data both in transit and at rest. Implement fraud prevention measures to detect and mitigate fraudulent transactions. Regularly updating the security patches and following best practices for web application security.

Testing of the payment application and validating the accuracy of the application software

Thoroughly test the payment system with various payment methods and scenarios before releasing the software in the market. Validate the accuracy of transactions and ensure compliance with payment regulations. Conducting security audits to identify and address vulnerabilities.

**Story 2 : Implementing efficient transaction processing capabilities to handle daily payment volumes and ensure timely settlements. Integrate with existing contact databases or creating a new contact management system to allow users to easily find and interact with contacts. Integrate with popular payment gateways (Example:Stripe, PayPal, Brain tree, Authorize.Net) to provide secure and convenient payment options.**

Transaction Process System:

Designing the transaction processing system to handle high volumes of transactions without compromising performance.

Optimizing the database queries, indexing, and caching to minimize latency.

Improvement of the application software efficiency:

Implementing the batch processing for recurring or non-urgent tasks to improve the application efficiency.

Reliable Process Framework:

Implementing robust error handling mechanisms to prevent system failures and ensure data integrity is Prioritized.

Developing recovery procedures to handle system outages or data corruption.

Automated settlement process to ensure timely payments to merchants and acquirers.

Develop mechanisms to synchronize contact data between your system and existing contact databases

Data consistency needs to be maintained and avoiding of the duplication.

Optimized Search:

Efficient search functionality and advanced search techniques to allow users to quickly find contacts

Providing features for users to interact with contacts, such as sending messages and making calls

Communication tools need to be integrated within the application for better communication purpose

Implement fraud prevention measures to protect against unauthorized transactions.

**Story 3: Implement the strong authentication methods (Example: Multi-factor authentication and using of Biometrics) to protect user accounts from unauthorized access. Establish authorization, authentication rules and permissions to control user access to different features and data within the system. Validate user-provided banking details against real-time data from banks or through manual verification processes. (A process like Active Directory Domain Services)**

Multifactor authentication and use of biometrics:

Implementing a Multi factor authentication system like user’s password or biometrics.

Enforcing strong password policies, including minimum length, complexity requirements, and password expiration.

Grant users only the minimum permissions necessary to perform their job functions. Regularly review and update permissions to ensure they remain appropriate.

Setting up alerts in the application software which will help to monitor the key events like performance metrics and issues in real-time.

Following the payment gateway's integration guidelines to integrate its update API into the application. Implementing secure communication channels to protect sensitive payment data and users.

Handling Payment Requests of the users:

Developing a user interface to allow users to input payment information. Validating the input data to ensure accuracy and prevent errors. Forwarding the validated payment request to the payment gateway and to the bank

Store Payment Information:

Storing payment information (Example: Existing contact and new contacts) for future transactions.

Implement Security Measures:

Use encryption to protect sensitive payment data both in transit and at rest. Implement fraud prevention measures to detect and mitigate fraudulent transactions. Regularly updating the security patches and following best practices for web application security.

Validating the accuracy of the application software

Thoroughly test the payment system with various payment methods and scenarios before releasing the software in the market. Validate the accuracy of transactions and ensure compliance with payment regulations. Conducting security audits to identify and address vulnerabilities.

**Story 4 : Thoroughly testing the integrated systems to ensure they work seamlessly together and meet performance requirements. Design the system to handle future growth and increasing transaction volumes.**

Testing of the application software:

Testing plans: Integration, application and user acceptance testing.

Perform the testing of the application plan: How the system performs under various loads using tools like Appium and Load runner

Conduct vulnerability assessments and testing to identify potential security flaws.

Monitor the application software and set Alerts :

Continuously integrate feedback and improve the system.

Scheduling regular performance and security testing to ensure the application software remains robust .

Regularly update the payment system based on user feedback and evolving security threats.

Documentation and Review Process:

Documentation of testing procedures and results for every stages.

Conduct regular review sessions with stakeholders daily for improvement of the application software

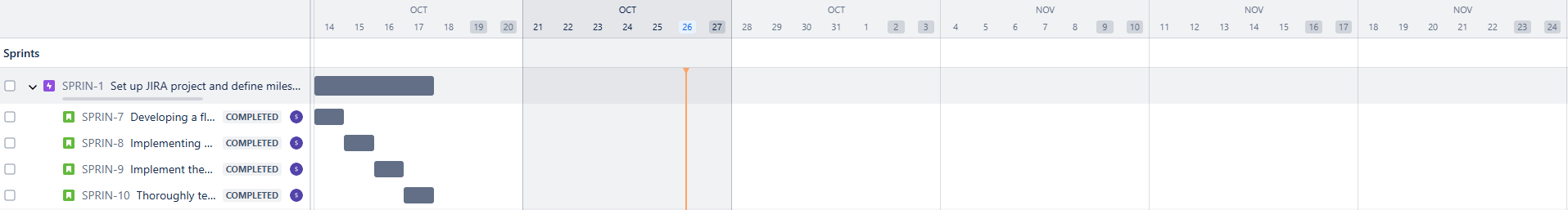
Final step:

Achieving the best performance of the application software before releasing in the market to avoid complains and get best feedback to increase transaction volumes and customer satisfaction.

Customer contact centre:

24\*7 help for the customers for any transaction issues and feedback for improvement of the application software.

**Screen shot:**



**Epic: Create user stories for user authentication**

**Story 1: Registration of the user and creating the account:**

Usage of email address and the phone number. User should get a verification code both in email address and phone number to link up the payment account as well as to recover the pin or the passwords.

**Story 2: Login with mobile number and email address. Addressing a device to be trusted and managing the devices:**

User can login to the application software using the email address and phone number after receiving an OTP. User can opt for the option to make a device trusted for future device recognition. Option for the user in the application to manage the devices from the application software after logging in to the account.

**Story 3: Enabling and disabling the Two factor authentication and Biometrics authentication:**

User should enable two factor authentication so that an extra layer of security gets added after receiving an OTP both in email address and phone number. Enabling the Biometric Authentication (Finger Print or Face ID). User can also opt for the option to disable two factor authentication and Biometrics from the application or the software settings.

**Story 4: Session Management:**

User can stay logged in after closing the application software but an alert will appear as a notification. User can also opt for the option to automatically log out from the application while exiting the app. Automatically log out option will be always be enabled if found the user is inactive for certain period of time (Time period - five minutes).

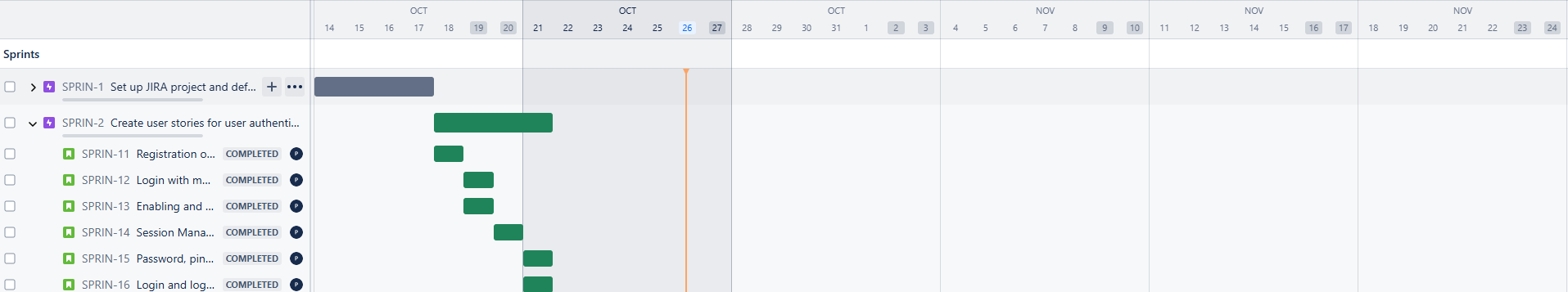
**Story 5:** **Password, pin recovery and edit request. User account lockout scenario.**

User should always get the verification code via email or phone to recover or edit the exiting password but the edit option will not work if the user is already signed in or in inactivity period. Same process will be done for account recovery scenarios but with a bank authentication within twenty- four hours.

**Story 6: Login and log out history. Password, Account recovery and edition history.**

Option to see history for login, log out, Password and account recovery in the application or generate report which will sent to the user’s email address and phone number based on the request from the user.

**Screen shot:**



**Epic3: Design and develop front-end login page**

**Story 1: Creating of user-friendly application**

To create a user-friendly application page which can be easily understood for any users with multiple languages. Change of the language should appear in the front pagebefore signing in to the account .

**Story 2: Implementation of Robust security measures and Zero Trust security option.**

To implement robust security measures to protect the data of the user. Providing the best security as well as user can opt for Zero trust security based on their choice

**Story 3: Responsiveness of the application page**

The application page should be responsive for the compatible device. Should produce an error while accessing the options.

**Story 4: Designing of the application software**

Creating basic layout which should include the essential elements like username and password fields, login and sign out option, sign up and forgot password option. A layout to upload the user’s picture.

**Story 5:** **Development Phase**

The Front-end development team should build the HTML structure using elements header, main or body and footer. It should include primary buttons for typing the username and passwords. Secondary buttons should be also included like Sign up, login, logout and forget pin or password.

CSS styling should be done to apply the style for the HTML elements which should include futuristic colours like Neo-mint, electric blue, Graphite black etc

Java script functionality for interactivity and validation to check for the required fields once the data is submitted from the user’s end

**Story 6: Security features:**

Prompt should appear in time of login and logout and alerts should be thrown for any unsuccessful login or account locked scenarios

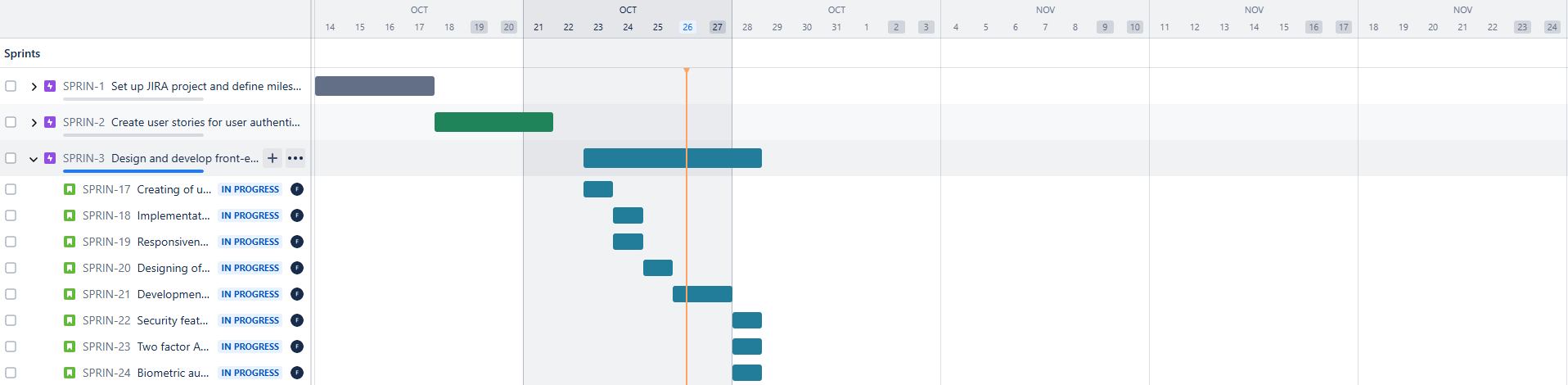
**Story 7: Two factor Authentication**

A secondary screen should appear once user enters the login credentials. If the option is enabled from the user’s end.

**Story 8: Biometric authentication and third -party login.**

User should see the option to login via Biometrics and using third party login like (Google, Twitter and face book)

**Screen shot:**

****

**Epic 4: Implement backend authentication with OAuth**

**Story 1:** **Configuring the application software with the Open Authorization:**

Configuring the backend application with the OAuth providers. Authorization from Multiple Oath providers can help a user to use their preferred account like (Google, Facebook, Apple, etc) .Users always trust familiar platforms for authentication and security which will enhance the trustworthiness of the application software .

**Story 2:** **Integration of the Oauth provider library and setting up of Oath in the backend of the application:**

Back-end Developer should integrate the Oath provider’s library into the backend application which provide the OAuth flow and communicate with the Oauth provider. Developer should obtain the client Id and client secret for each provide. Developer should ensure minimal scopes are requested from each provider for security and privacy purposes.

**Story 3: Creating of Oauth Authorization Flow:**

Redirecting the user to the provider’s content. The provider prompts the user to grant permission in the application. If the permission gets granted from the user the provider redirects the user with an authorization code in the application software. Exchanging the authorization code for an access token using the user ID and other details.

**Story 4: Storing of the provider’s access token:**

The access token securely gets stored in the server (NGA pay server), later this access token will be used to provide the API request to the payment provider and he will verify the token and grant access once a transaction is requested from a user.

**Story 5: Mechanism Implementation to refresh the access token:**

A mechanism will be implemented to refresh the access token once the token gets expired.

**Story 6: Implementation of Error handling:**

A mechanism will be implemented to handle the errors and the Oauth authentication gets failed.

**Story 7: Email, phone number, third-party account password and Biometrics authentication:**

Back-end Developer should implement a mechanism which allow the user to login to the application software based on their request and also allow user to reset and edit their login credentials. If an exiting account gets locked where the user has already configured his payment details with bank. The banker needs to provide the permission to user after the conduction of the verification within twenty-four hours.

**Story 8: Integration of the payment gateway:**

Developer should integrate at third-party payment gateway (Example Paypal , Razor pay , etc) which will handle the payments and transactions done by the user. Support for multiple payment methods like debit, credit and international cards as well as net banking and UPI.

**Story 9: Transaction history:**

Back-end Developer should design the data base so that the transaction history of a particular user for more than hundred years can be stored. Filter option needs to be implemented so that the user can see the transaction history for a particular time. Transaction history should reach the user via email and text once requested by the user. Developer should also implement a mechanism which will also generate automatic monthly reports of the user’s transaction and send to the users email address.

**Story 10: Security features:**

Encryption of the user’s data both in transit and rest while a transaction is been done by a user. Implementation of the refresh token for frequent user login and logouts. Expiry of those tokens when a user is idle for more than five minutes**.**

**Story 11: Error handling for any transaction:**

Back-end Developer should implement a mechanism which can auto handle any error generated while a transaction happens and resolve the bug without making the user to wait for a long time. Reports needs to be shared to the user for a bug fix so that a complete transparency report can be shared to the user for making the application completely Trustworthy. Notification needs to be throwed from the application for the fixit of the error.

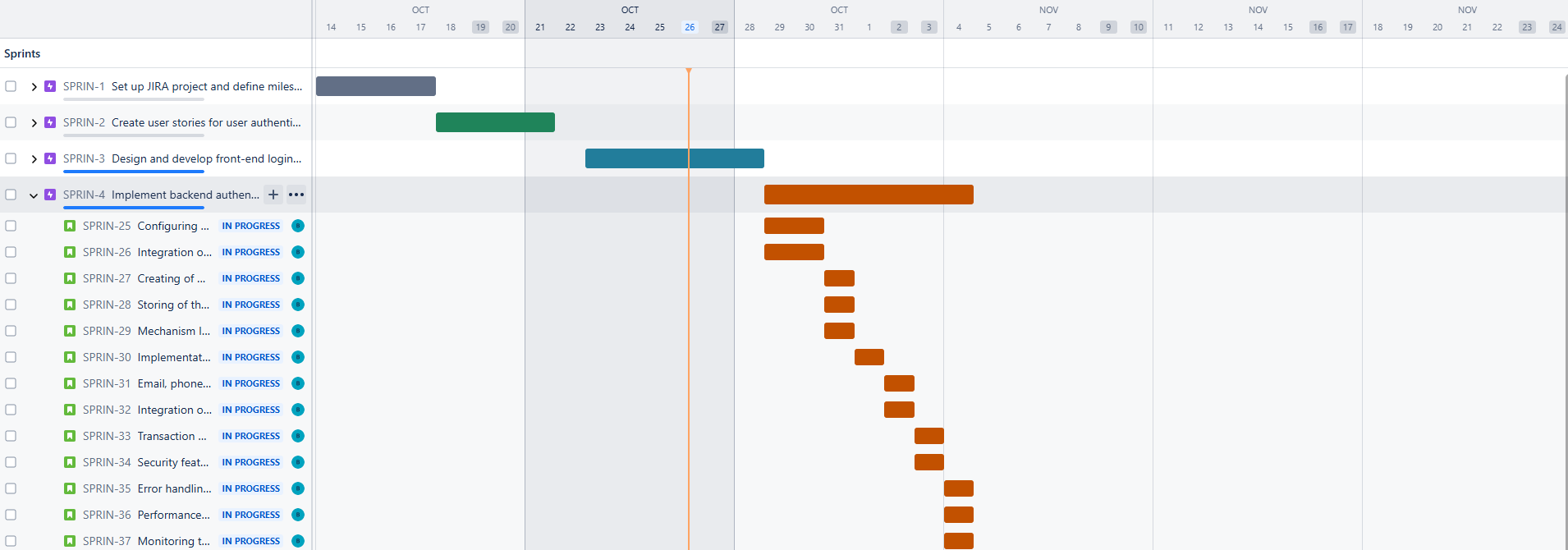
**Story 12: Performance Optimization of the application software:**

Back-end Developer should keep in mind for such mechanism that helps the user for a smooth performance while user is frequently the important fields like checking transaction history, existing banker details, user’s details and contacts. Setting up a cache expiration time if the user is inactive. If large amount users are accessing the application software from different devices sluggish performance should not appear and handle high traffic. Mechanism should be also implemented that by chance any issue occurs which is affecting multiple user’s immediate reports should be sent to the backend team before the user contacts the contact centre.

**Story 13: Monitoring the application software codes**

Back-end Developer should monitor the software codes so that if any changes are required then the same can be implanted in the next version of the software or implement an update while the user is updating the application software in daily basis.

**Screen shot:**



**Epic 5:** Write unit tests for authentication module

**Story 1: Testing the OAuth Provider Authorization**

QA Engineer should verify the authorization URL is generated for the multiple providers with appropriate client ID and scopes. Engineer should also test that if any error gets produced with Oath credentials and if something is missing from the application. Engineer should create a dummy user to cross verify whether the authentication process gets passed.

**Story 2: Dummy user account test**

QA Engineer should test the Oath provider’s a dummy user account whether the data is correctly accessible from the application software. Engineer should also test with invalid user account whether the error is getting produced from the application software.

**Story 3: Verification test**

QA Engineer should test whether a JWT token gets created after user signs in with the email address, phone number and other third-party account. Verification of the JWT token needs to be done whether the token allows access to secured routes. For expired tokens and incorrect credentials, the application software should deny the user to login.

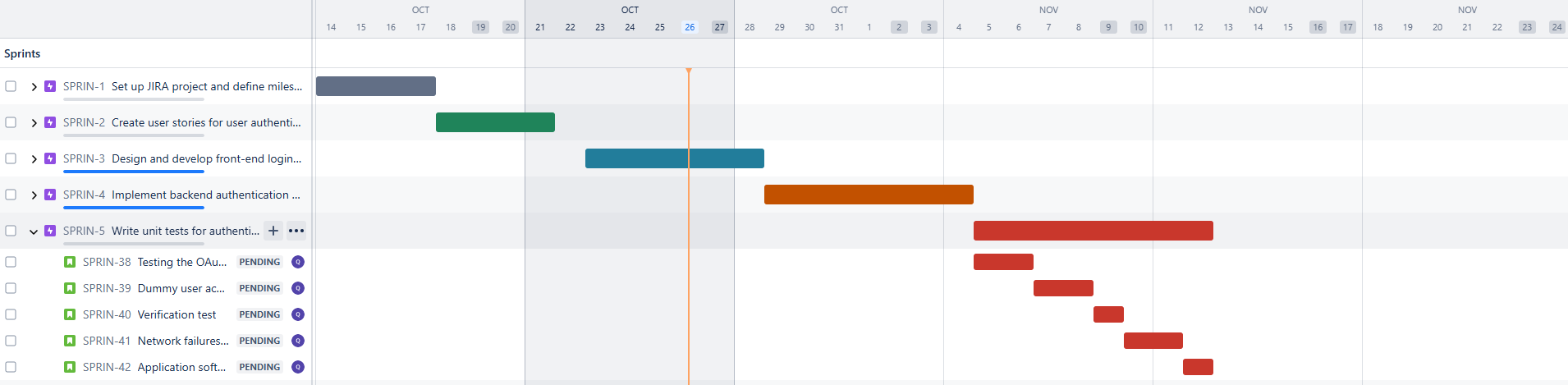
**Story 4: Network failures and user permission for the application**

QA Engineer should check the network errors when there is a down time from the ISP and how the application responds when the errors appear. Engineer should also check if a user denies permission or permits a permission then how the application responds with disrupting the user flow.

**Story 5: Application software Data base and User Session test**

QA engineer should perform a test that the user’s details is correctly stored in the database including the OAuth provider details. Engineer should test and confirm that the duplicate contacts does not gets saved in the data base and users session gets expired according to the time duration implemented in the application software.

**Screen shot:**

****

**Epic 6: Conduct sprint planning for sprint 1**

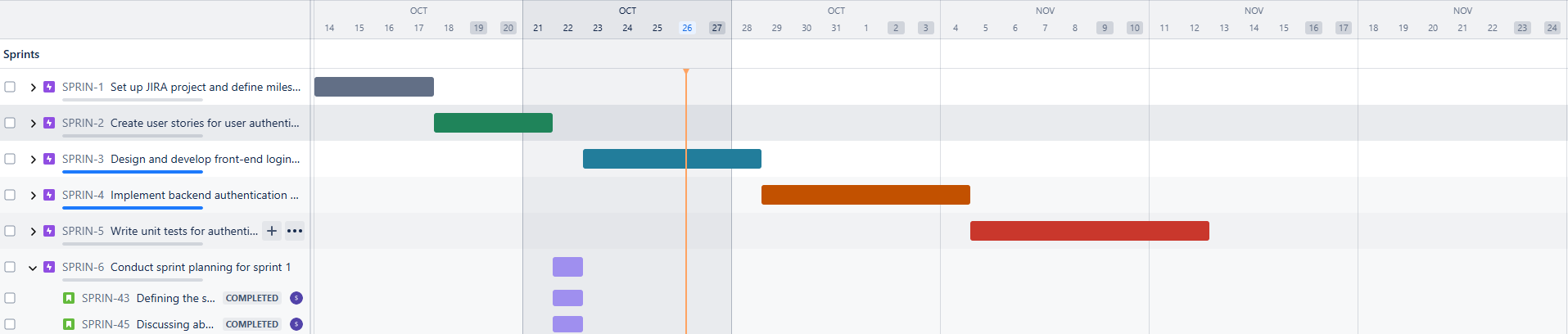
**Story 1: Defining the sprint goals and setting the roadmap of the project. Planning and break down of the task for each team. Potential risk from each sprint.**

Clarifying each team about the about the sprints and setting up the targets for each sprint. Defining the road of success for each sprint. Discussing about the potential risk for each sprint which might delay the task. Explaining about the Application UI and user’s authentication. Discussing about each team’s task. Clarifying with team members for any doubts to perform the task within the assigned time and set up the expectation for each sprint.

**Story 2: Discussing about the Tool usage for making the application**

Tools usage discussion with each team for having a clear vision to complete the project at correct time.

**Screen shot:**

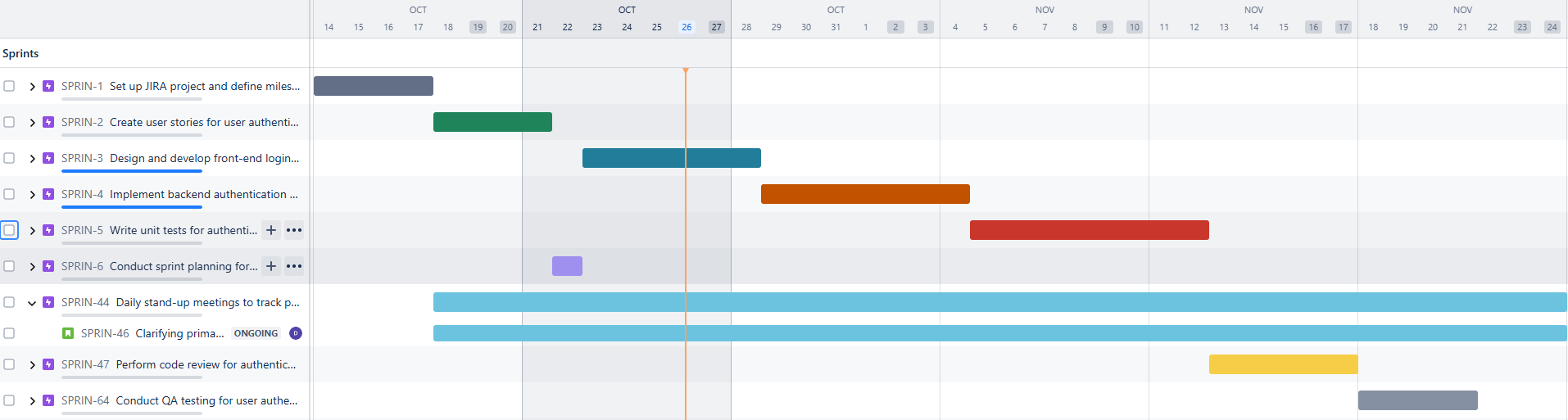
****

**Epic 7: Daily stand-up meetings to track progress**

**Story 1: Clarifying primary goal of the project when the sprint needs to be completed.**

Briefing the daily updates which keep everyone aligned and focused on the current sprint's goals. Team members can share their work and challenges while performing the project. Team members have a complete view of the project, leading to better coordination and problem-solving techniques. The entire team can track the overall project progress and identify areas that might need additional attention.

**Screen shot:**



**Epic 8: Perform code review for authentication module**

**Story 1: User authentication Flow.**

Backend development team should review the codes so that the authentication flow is clear and secure. Security features codes should be strong and effective while user storing the password and while a transaction happens there should not be any security breaches.

**Story 2: Oath Providers Authorization.**

Backend development team should verify the application software codes should completely function while a user is using a third-party account.

**Story 3: Avoiding the duplication of the application software codes.**

Backend development team should also verify that there should not be any duplication of the software codes and adhere to coding standards and guidelines.

**Story 4: Sensitive information of the user.**

Backend development team should make sure that the personal information entered by a user should not revealed so that the trustworthiness of the application increases.

**Story 5: Error messages produced by the application software.**

Errors produced should be user friendly and completely understandable. It should be informative to the users.

**Story 6: Performance of the coding structure modules.**

Backend development team should check the code paths and organize the codes in to well- structured modules. The team should check whether the codes are efficient and effective.

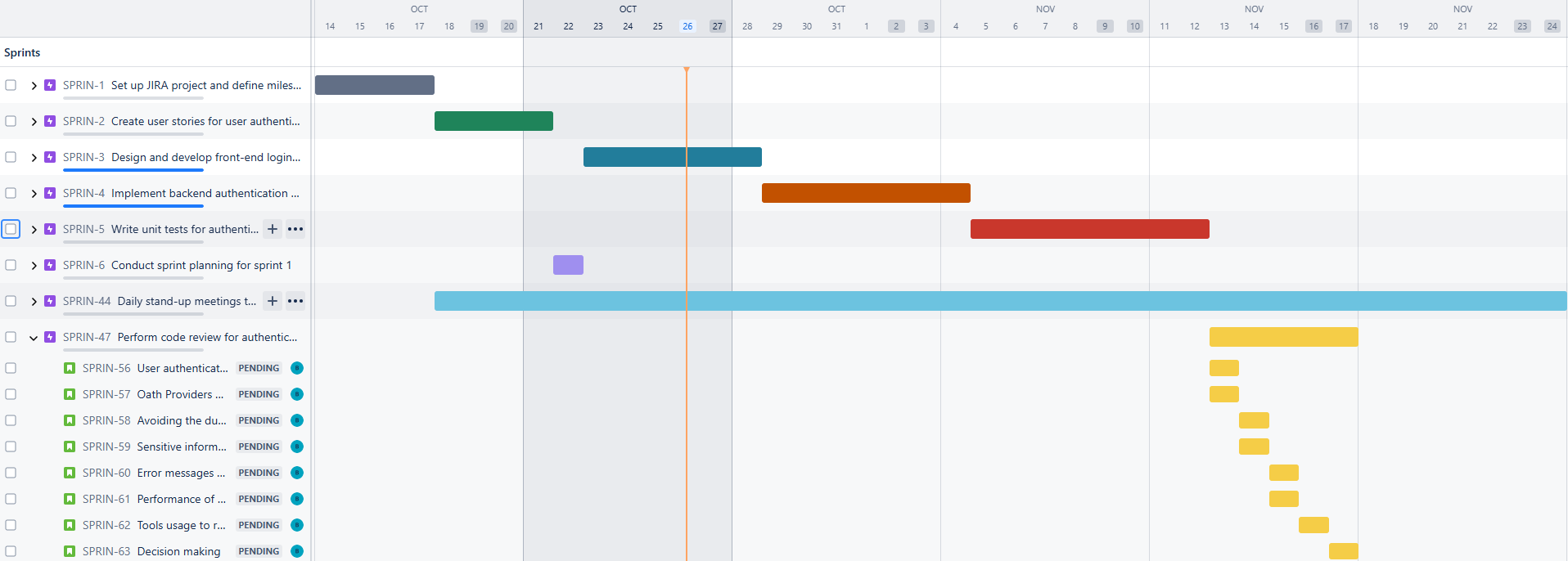
**Story 7: Tools usage to review the application software codes.**

Discussing about the tools for reviewing the codes.

**Story 8: Decision making.**

Making proper decisions so that the correct technologies can followed while developing the application.

**Screen shots:**

****

**Epic 9: Conduct QA testing for user authentication**

**Story 1: User login and logout testing:**

QA engineer should test with a dummy user account whether the is user can login with the email address, phone number, third- party account as well with Biometrics. Engineer should test whether that user account can be logged if requested as well as automatic logout methodologies works automatically when user is inactive for more than five minutes producing a notification in the application and the user should be notified via email and text.

**Story 2: Error login notification if the user login fields are empty:**

QA engineer should test whether an error gets produced while leaving the username or password fields is been left blank while trying to login. Case sensitivity methodologies should also be implanted so that user can be notified.

**Story 3**: **Managing the session time:**

Ensuring that the user remains logged in after a successful login attempt from the user

**Story 4: Logout testing:**

Ensuring that user account gets log out from the application software once the logout option gets selected.

**Story 5: User account details remembrance and device trustability.**

User account details should be remembered by the application if the remember me option is selected from the user’s end. Device which are marked as trusted should be stored properly in the application software data base.

**Story 6: Appearance of hint if user inputs invalid user login credentials.**

Hint should appear for the user if an invalid login credential entered by the user. It should also notify the user with knowledge-based article for the self-support.

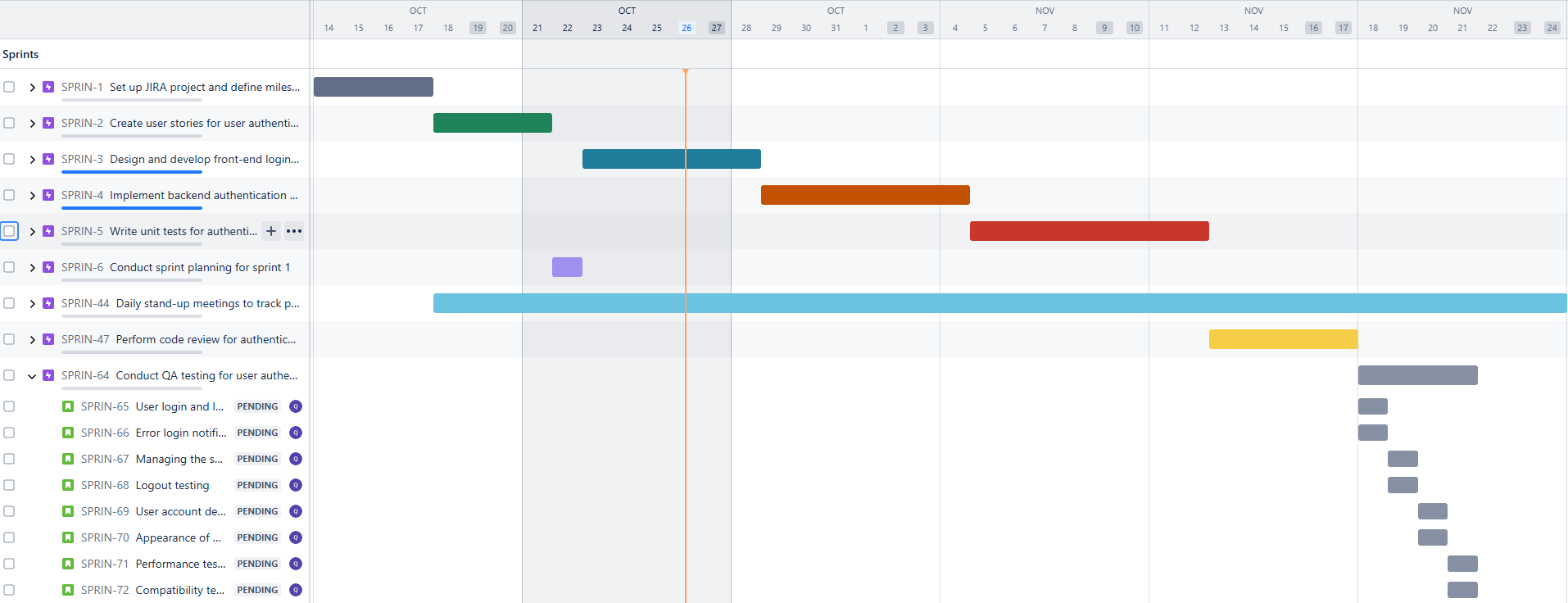
**Story 7: Performance testing of the application software.**

QA engineer should test whether the application software responds once the user is trying to login with his login credentials and whether the application software is able to take the load when multiple users is trying to login at the same time. QA engineer should ensure that a stable performance is maintained when a user is trying login and logout.

**Story 8: Compatibility testing.**

QA engineer should test whether the application is compatible with multiple browsers (Example: Chrome, Edge, Firefox, Safari, Bravo etc). The application software should compatible with computers, tablet, mobile and ipads which run operating system like Windows, Mac, Chrome OS, Linux android and IOS). The application software should not be compatible with operating system version which has been stopped by the vendors. Potential bugs need to be fixed before the product gets launched in the market.

**Screen shots:**

****

**Epic 10:** Sprint summary

**Story 1: Achievements:**

Integration of a highly secured payment gateway, Oath authorization, making user friendly interface to attract most of the customers around the universe. Implementing next generation features like voice payment option, zero trust security etc. High quality performance while running the application software.

**Story 2: Challenges faced:**

Decline of some payments from the payment gateway while using the dummy account. Transaction limit error produced when multiple dummy account payments were processed on the same day. No error was produced for a certain period when there was a downtime from the ISP.

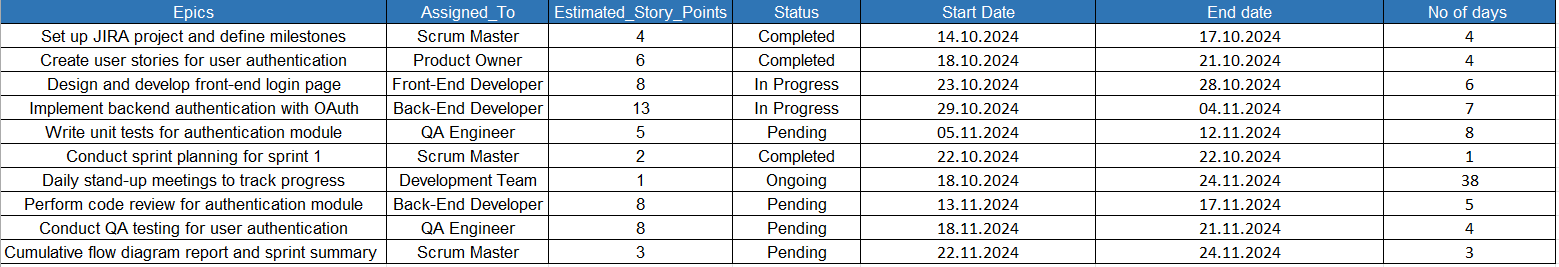
**Story 3: Recovery of the problems which occurred while building the application software**

Communication was done from the backend development team to the payment gateway team to resolve the issues asap to increase the payment limit for the application software. Instant resolution was provided from the payment gateway team to overcome the payments which were declined. Notification produced in the application software whenever there was a downtime from the ISP.

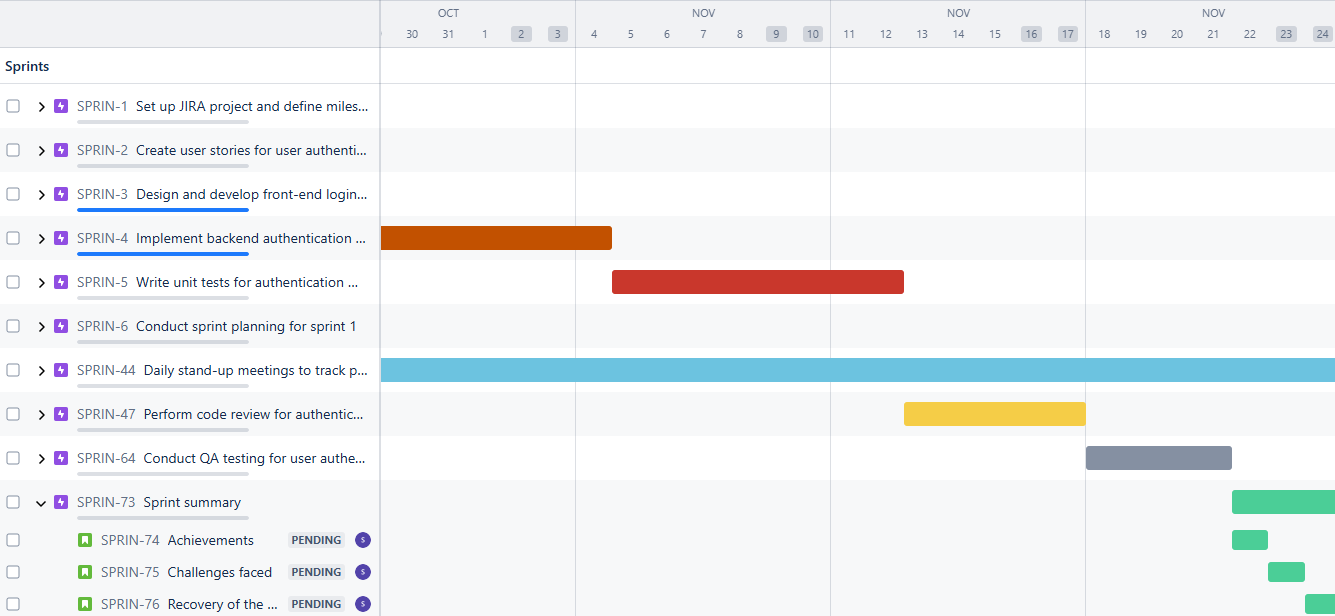
**Total number Weeks: 5 weeks**

**Number of Epics:10**

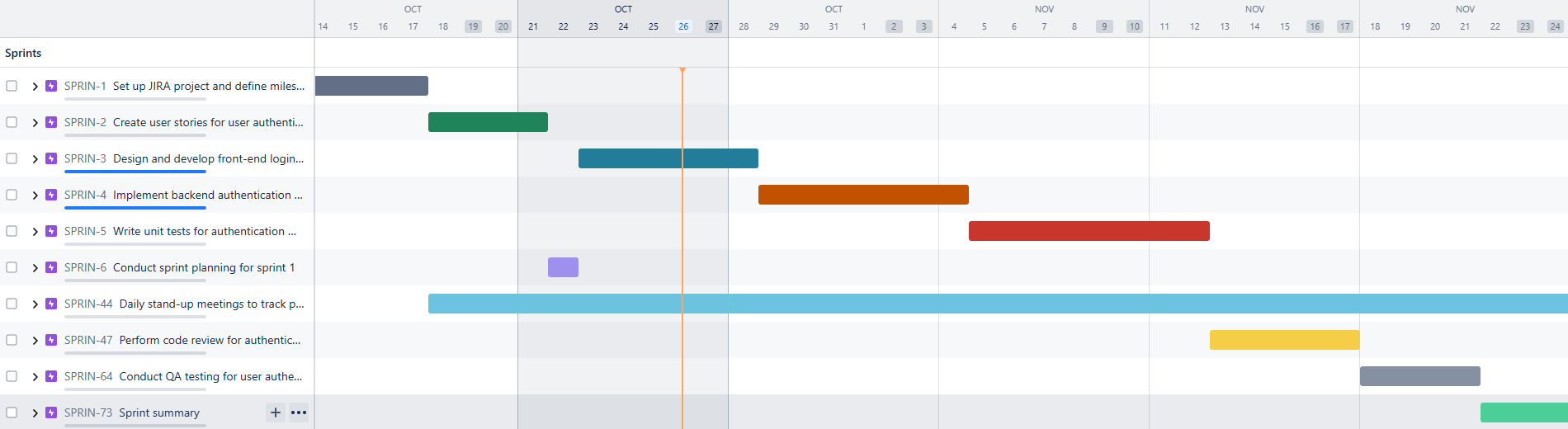
**Chid stories: 58**

****

**Screen shot:**

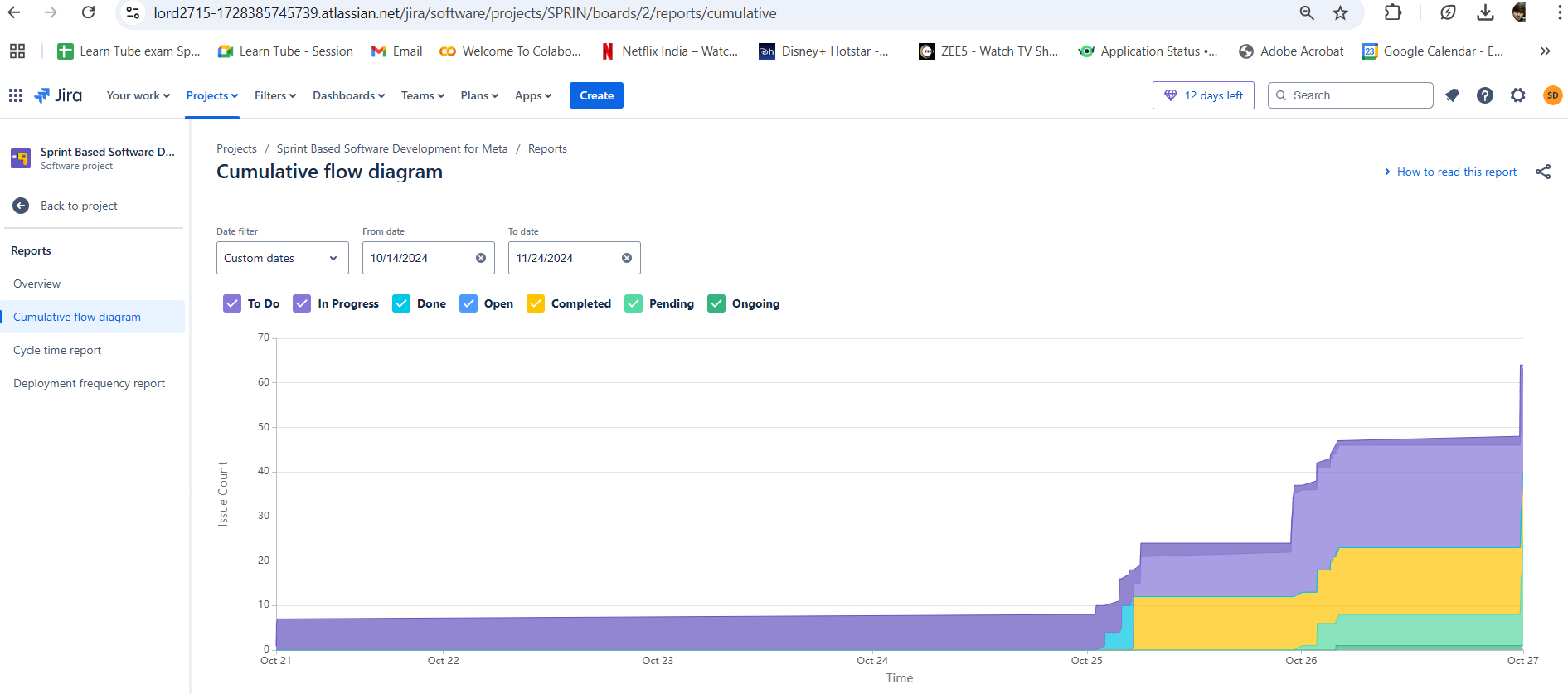
****

**Screen shot: (Time line for all the epics)**

****

**Report: Cumulative flow diagram:**

**Screen shot:**

****

.