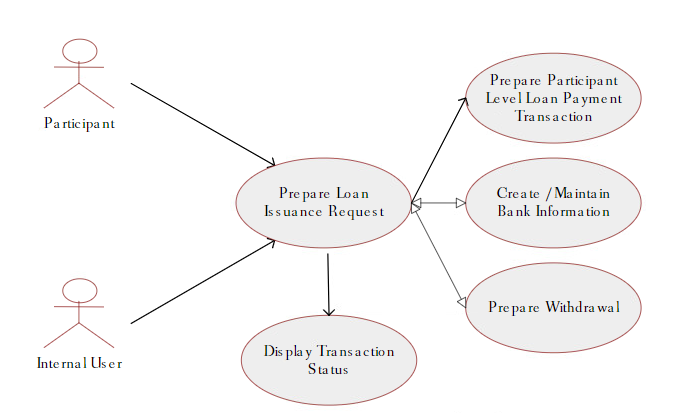
PROJECT DESCRIPTION

Distribute Product Team is a project team under VANGUARD GROUP where in this project deals with Participant Experience (PE) and Retirement Withdrawal Coach (RWC), which are developed based on technologies mentioned below:

* Java/J2EE
* Spring Framework
* Angular
* DB2 and Postgres

Participant Experience is categorized into two different version as below

|  |  |
| --- | --- |
| **Actor name** | **Description** |
| External version (IPE)/ Participant | An external user to Vanguard who has the authority to view and maintain information. In this case it is the participant with Institutional relationship with Vanguard. |
| Internal version (III) | An internal user to Vanguard who has the authority to view or maintain information. In this use case, it is participat services and recordkeeping services. |



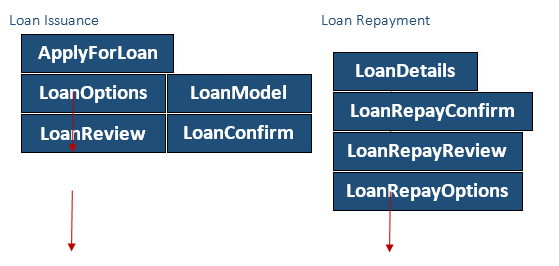
Distribute Product Team own these modules which are mentioned and explained below

* Withdrawals
* Loans
* Terminations
* Retirement Withdrawal Coach

**WITHDRAWALS:**

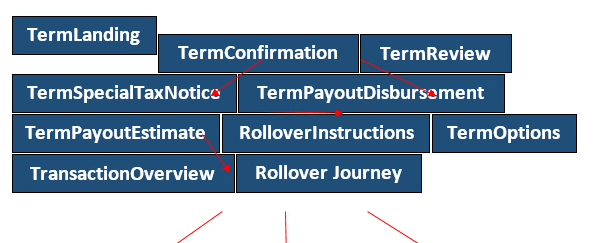
* Withdrawals is a recently build module under DPT, which deals with two different types hardship and non-hardship withdrawals.
* Hardship withdrawals requires that user can withdraw their money only for specific reasons.
* Internal version has special “Withdrawal by Option” flow and more options for the user to how to withdraw their money.
* External version has a very simpler flow with general or basic options for the user to withdraw their money.
* Withdrawals module technically made up of three main components.
  + WDL – NGA UI for withdrawal transaction
  + WDS – NGA Service (withdrawal-process) that deals mostly with data needed for hardship withdrawals.
* Hardship withdrawals are again classified into two types
  + Summary hardship allows the use to submit answers for hardship reasons online
  + Non-summary hardship does not allow users to submit answers online, they must send in documentation by mail

**LOANS:**

* Loans is applicable for both the External and internal participant.
* Loans flow is classified into two types as **TAKE LOAN** and **REPAY LOAN**
* Below flow diagrams will explain how to navigate over the Loan issuance and Loan repayment flow.
* 

**TERMINATIONS:**

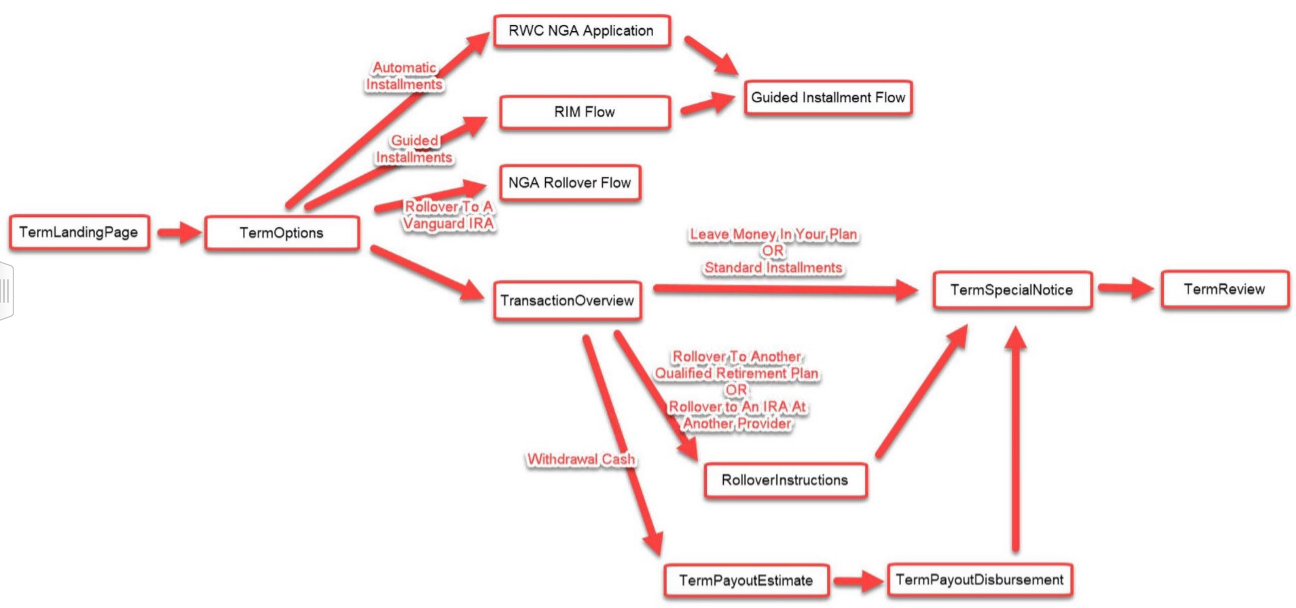
* Terminations might mean many things as stated below.
  + Set up Installments / Guided Installments (receive a certain amount at a certain frequency).
  + Guided installments.
  + Adjusts the installment amount for inflation every year.
  + User can be able to model via the old Retirement Income Modeler Tool.
  + User can be able to model via the new Retirement Withdrawal Coach Tool.
  + Submitting the installments for which system will provide a form to the user.
  + User must complete and send the form via email to complete the transaction.
  + Rollover money to another location
  + Payout Now
  + No action (Defer Payout)
* Below flow diagrams will explain how to navigate over Terminations flow.



**RETIREMENT WITHDRAWAL COACH:**

* RWC is UI treatment on the secure home page of PE.
* This application has a tool/flow that the user can navigate to calculate a good installment amount to take.
* Retrieves user data and user loan or any other information and display in the UI.
* RWC calls the Sustainable withdrawal calculator with those pieces of data to get results
* Connects to the Guided Installments flow in External/Internal.
* Users can choose to set up guided Installments based on the amount they modeled in the tool.

Flow chart mentioned below will walk through the RWC flow.



**ROLES AND RESPONSIBILITIES:**

* Leads and influences technical direction for large-scale, highly complex technical
* initiatives and/or projects requiring integration of cross functional systems.
* Provides technical guidance in evaluating applications systems or evaluating requests for
* proposals.
* Collaborates with the business to prioritize key business/technical initiatives.
* Utilizes expert knowledge of the customers business to recommend solutions and ensures
* business and technology objectives are met and maintained.
* Ensures developers produce comprehensive tests for all developed code. Lead system and
* integrated testing across sub-systems as the need arises.
* Help maintain code quality, organization, and automatization.
* Develop and maintain Java/J2EE, 12 -factor applications using Spring boot, Pivotial cloud foundry, AWS App Mesh, AWS EC2, EKS, S3 and Microservice design pattern.
* Develop and maintain Angular Applications using MEAN Stack approach and deploy them in PCF
* Generate, automate and execute test cases using Mock frameworks, Junit and perform
* Test Driven Development Strategy.
* Generate, automate and execute business test cases using Cucumber and perform
* Behavior-driven development strategy. Perform API integration like PACT testing.
* Work with existing Agile methodologies and participate and SCRUM calls.
* Participate in Agile SCRUM and application design meetings.
* Manage code builds using CI/CD tools such as GIT.
* Manage code coverage and comply to code coverage threshold using tools such as
* SonarQube. Execute and comply static security testing and vulnerability testing using
* standard pipeline tool integration in DevOps workflows.
* Analyzing angular code using ts-lint to maintain best coding practices across the team
* API integration with UI/UX.
* Manage logs using Splunk and other log management tools.
* Responsible for understanding the project from application level.
* Consuming RESTful web services using apache Http Client for the data coming from external systems.
* Use Spring Validation framework to implement the server-side validations and use Angular JS AJAX to get the data from the server asynchronously by using JSON objects.
* Adding increased security level to the code before deploying it in the cloud.
* Safe guarding the servers in any unexpected situations and making disaster recovery plans for securing the data.
* Configuring spring AOP and using Spring declarative transaction management for transactions.
* Worked one-on-one with client to develop layout, color scheme for his website and implemented it into a final interface design with the HTML and JavaScript.
* Implementation of Spring Framework for building the REST API and using MVC architecture to develop flexible and loosely coupled web application.
* Developing the DAO layer for the application using Spring, Hibernate and development of various business logic and reports using HQL and Hibernate Criteria
* Responsible for updating APIGEE code for masking the api's for security purpose.