



Student Loan Approval System

Overview:

The client for the project is a decade old company comprising of a team of senior experts. They specialize in providing various customized services and products in the Finserv space. Asset management and administration, capital markets advisory services, loan acquisition and loan program development. Their clients include colleges, universities, banks, investors, insurance companies. They have invested over \$150 million of their capital in investing portfolios and are managing over \$30 billion of private and federally-guaranteed student loans.

They needed our services to implement a highly secured student loan origination system including an admin portal for the underwriters to certify and approve student loans using various third party integrations like Transunion, getting schools' verification data etc.

Client details:

Name: Confidential | **Industry :** Finserv | **Location:** USA

Technologies:

Laravel 5, PHP, jQuery, Gulp, supervisor, AWS, EC2, S3, Docker, Jira, CodeCommit, CodeDeploy, CodePipeline, REST APIs, PDFtk, TCPDF, DocuSign, Transunion API.

Student Loan Approval System

Project Description:

The loan origination system makes calls to the Transunion API to determine the credibility of the student based on their credit history. If credit check passes successfully against the different identified criteria, the student is shown loan offer. A dynamic pricing algorithm based on market interest rates then gives the various pricing options to the student. Then a school certification process is undertaken by the admins which is partly automated.

Key Features:

Multiple loan products, apply with or without a cosigner, DocuSign electronic signature integration, secure document uploads to AWS S3, complex algorithms to calculate loan rate, generating dynamic PDFs, AWS configuration setup and deployment, automated deployments using code deploy and code pipeline, automated test scripts.

Key Success Factors:

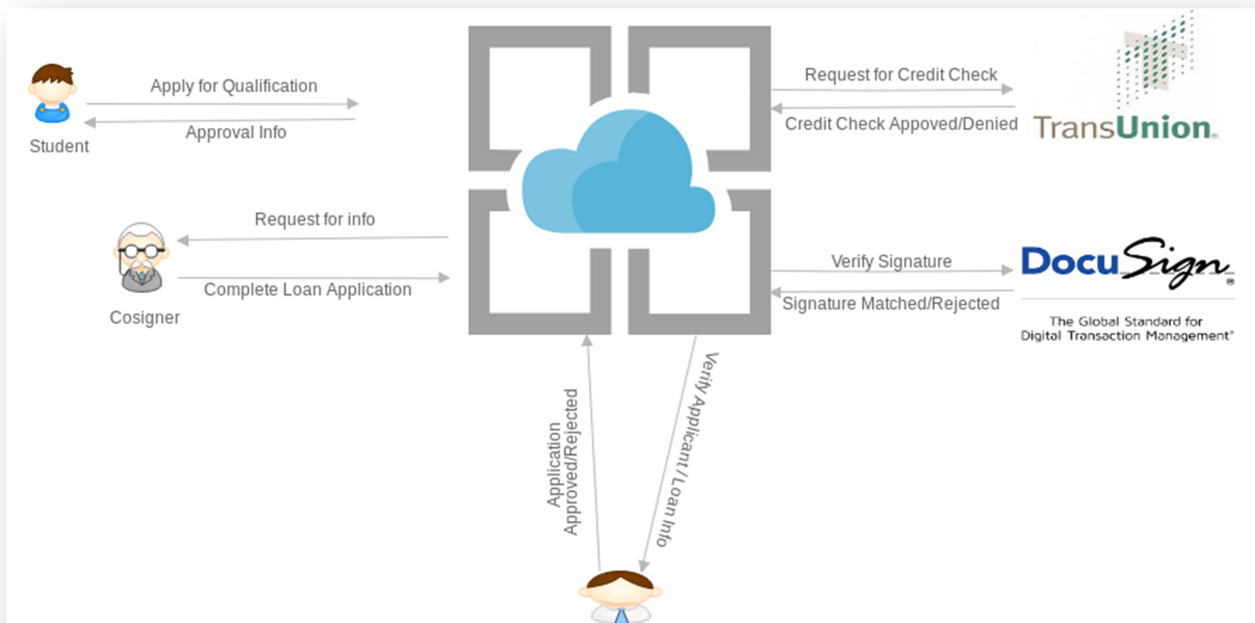
- Using a project management system like Jira
- Effective planning and execution of sprints
- Adhering to Scrum processes like daily standups to discuss progress and issues
- Regular review meeting with product owners/stakeholders and incorporating feedbacks
- Using Docker to ensure consistency across development environments
- Effective Git branching strategies for developers and releases
- Dedicated QA to test featured branches before pushing to UAT
- Continuous skilling of developers on web tools and techniques
- Dedicate adequate time to understand requirements and documenting them
- Using SonarQube and peer to peer code reviews for code quality

Processes followed:

1. Used Docker to locally setup code in environment as identical to production.
2. Used code-deploy along with code-pipeline to automate deployments. So any changes made on staging branch were automatically updated on the staging/testing servers, saving deployment time.
3. Used Jira to effectively plan sprints, track tickets/bugs and track work progress.

Student Loan Approval System

4. Separate EC2 instances for production, keeping the production code and database completely isolated from development and testing. Highly restricted access to production server on AWS so as to keep user data utmost secured.
5. We used SonarQube for code quality checks, ensuring the code getting pushed to production is following same basic standards, is having minimum duplicity and is scalable. Setup Jenkins for automated code quality check when pushing to master branch.



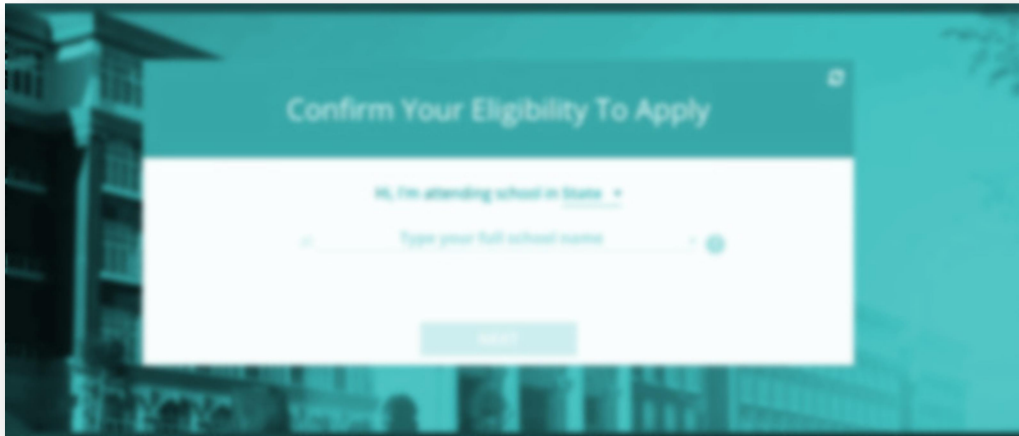
Security and Compliance:

The team decided to host the codebase in AWS knowing that they have 3rd party audits for security standards. The website uses https encryption using TLS and we deploy 3-4 times a week without downtime. Made sure that unit tests were written for each module. Functional testing over multiple browser and machine configurations is done to ensure no environmental issues are there. Data is encrypted in transit through Transport Layer Security, also all primary keys are encrypted when any form is submitted. Care has been taken to ensure that if one database is taken down, a secondary database is elected as primary. If there is an outage on the primary datacenter, the DNS immediately detects that and start redirecting requests to another datacenter in a different location. Provision is made to ensure the database is failsafe by taking snapshot every few hours. Written policies are in place for password protection and patch management. In addition to that only specific users are allowed to edit data in db. We also

Student Loan Approval System

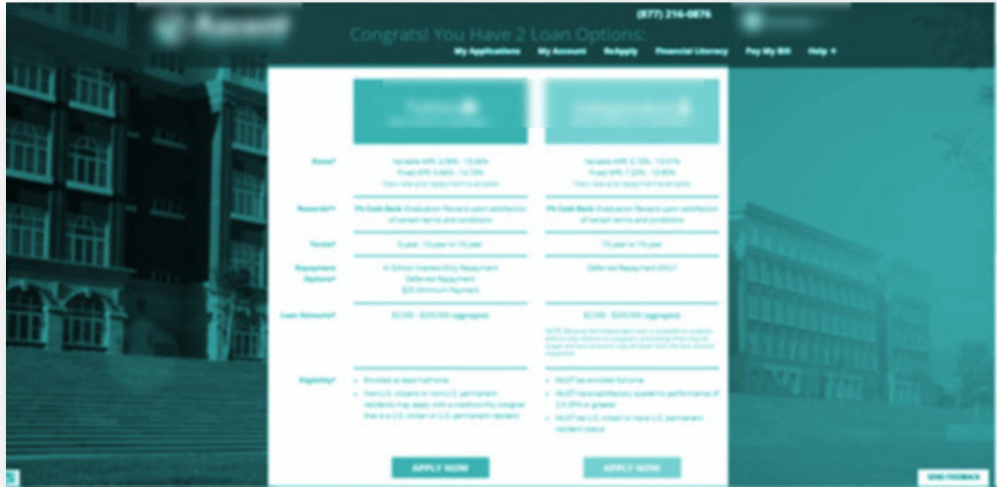
document exactly who has access to certain code projects. Try catch is extensively used to detect and log errors and a log monitor is used to send emails about errors with type. Enough data entry controls are in place to filter data. ACL is implemented to prevent unauthorized edit/deletion of data. PHP and Laravel versions are regularly upgraded to make sure the application is faster and secure. In addition to that we keep our customers up-to-date on the latest releases for any significant additions, fixes, or changes to the application.

Screenshots:



Screens 1: The student eligibility check page

Student Loan Approval System



Screen 2: Showing various loan product options on passing eligibility check.



Screen 3: Showing various loan product options on passing eligibility check.