JusticeAI Iteration Summary

Iteration 10

Team Members

Name	Student ID	GitHub ID
Lance Lafontaine	26349188	lancelafontaine
Arek Manoukian	21710389	arekmano
Sylvain Czyzewski	27066333	vynny
Mihai Damaschin	27177895	mihaiqc
Samuel Campbell	26457959	samuel-campbell
Taimoor Rana	26436110	taimoorrana1
Zhipeng Cai	21346482	choitwao

Project Summary

JusticeAI (ProceZeus) is a web chat bot that aims to facilitate access to judicial proceedings involving specific domains of law. Users will have the ability to converse with the chatbot, describing in detail the situation for which they wish to pursue litigation. The system, which will leverage the power of machine learning and natural language processing, will guide the user through a process wherein they'll be prompted with a series of questions relating to their potential case allowing the system to ultimately determine, based on provincial jurisprudence, whether the user has a valid case worth pursuing in the judicial system. Alternatively, the system may also suggest remedies in lieu of legal action if it is deemed unlikely to be in the user's best interest.

Velocity

The primary focus of iteration 10 was implementing measures to increase prediction accuracy. This was done by giving the user the option to answer more questions after getting the fist prediction. In addition, improving user experience by adding a progress bar indicating how many questions are left to answer was also added. On the ML side, additional classifiers were added, including the 'tenant expulsion' and 'retaking the rental' classifiers. However, despite being completed, they were not able to be integrated into the NLP system in time and were not able to be signed off on, and were thus pushed to iteration 11.

During this iteration, we were able to complete **13** story points.

The following is a list of user stories that were completed in Iteration 10:

- #44 - Show Conversation Progress % (5 pts)
- #346 - I want to answer more questions to obtain a more accurate prediction (8 pts)
- #345 - DEV STORY: Fix CD (bug)

Plan for Next Iteration

In iteration 11, we will be focusing on adding functionality that allows the user to see why they got a particular prediction. This will revolve around the dashboard feature, which will be implemented in this iteration. The dashboard will provide the ability to see how the similar cases that lead to the prediction compare to the user's input, allowing them to gauge how their responses lead to the bots prediction. In addition we will be adding questions that improve the prediction by gathering date range data from the user, allowing the machine learning classifiers to take narrow down the dataset to a specific time period, potentially leading to a better prediction in certain cases. Finally, we will be integrating the new claim classifiers into the NLP system, allowing the bot to handle cases wherein a landlord wants to retake a rental from a tenant, and cases wherein a landlord wants to expel a tenant. In addition to this, we will be adding legal definition tooltips to the bots questions, giving the user some well needed definitions for certain legal terms.

Shown below are the stories that we will be working on for Iteration 11 for a total of **45** points:

- #186 - Claim: Retaking the rental (8 pts)
- #184 - Claim: Tenant Expulsion (8 pts)
- #346 - Legal Definitions (3 pts)
- #43 - Dashboard Reporting of Results (8 pts)
- #371 - See Similar Cases (5 pts)
- #370 - Similarity Claim Table (5 pts)
- #369 - Ask when the tenant paid you last (8 pts)
- #368 - Fix Progress Bar Bug (bug)

Noteworthy Achievements

- A progress bar now indicates to the user how close the bot is to giving a prediction.
- Which questions to ask for a claim category are no longer hardcoded and are instead fetched from an endpoint on the ML service.

- Anti-facts are now automatically retrieved from the ML endpoint and mapped by the NLP service prior to requesting a prediction, greatly increasing prediction accuracy.
- User now has the option to answer additional questions to improve their predictions. Batches of five additional questions are asked before giving a new prediction.

Technology, Architecture and Library Changes

The stakeholder has decided against having OCR functionality in the application as it is unlikely that the value desired will be reached in time for iteration 13. The code for the OCR components in the task service are being kept in the repository as proof of concepts for future contributors to pick up. However, for the purposes for the project due by iteration 13, the following libraries are not being integrated with the remainder of project. Aside from this, no major architecture or library changes have been made in the other components of the application.

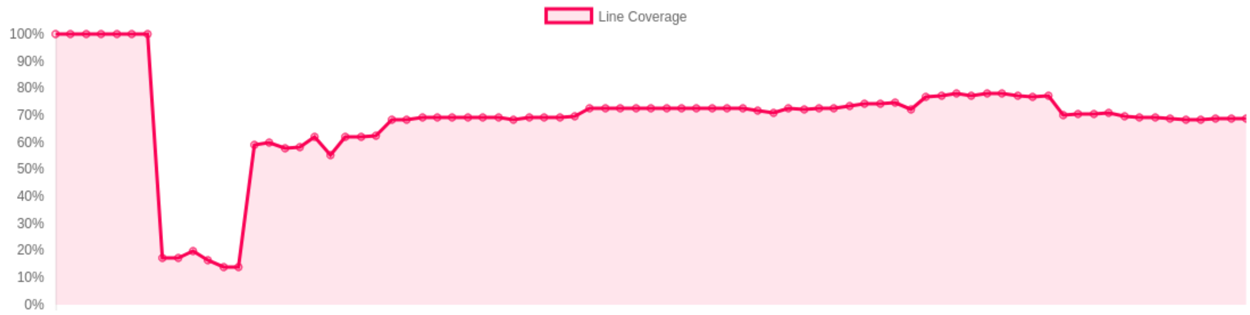
Component		
Tool	Language	Purpose
Task Service		
~~pytesseract~~	Python	Convenient Python bindings to the Tesseract library
~~tesseract~~	C++	An open-source multi-language OCR engine
~~OpenCV-Python~~	Python / C++	A computer vision and image processing library

Continuous Integration Processes and Naming/Coding Changes

We had previously determined that the automatic deployment of our application to our prod environment on pull request merges to master was not working properly. It was discovered that a regression was introduced in our server's SSH daemon configuration which prevented our continuous integration build service from being authenticated over SSH. Additionally, given that database migrations are currently not being used, the schemas for the database on the production server were not being updated, leading to a non-functional application. These continuous deployment issues were resolved during this iteration.

Unit Tests and Code Coverage

Line coverage after iteration 10 is currently at 68.63%.



Coverage has dropped by around 2% in this iteration due to the introduction of various new OCR-related machine learning additions. As these were a proof of concept of our system's lease and involved the creation of test mock for image data structures, we did not write unit tests for this changed code.

Iteration 10 Retrospective

What went well

- No significant blockers encountered during the iteration.
- Despite everyone being busy, features that improve user experience (progress bar, additional question prompt) have been implemented and work well.
- ML and NLP improvements have increased prediction accuracy. The related cases returned are now very similar to the user's input.

What went less well

- Development velocity slowed down significantly due to the midterms that all team members had.
- Machine learning resources are restrained due to lack of financial assistance from our stakeholder.
- Lack of assistance of our stakeholder is also restraining a cleaner more concise questioning phase which is affecting the polishing of the product and its quality.
- Bugs encountered and project did not build for demonstration to our project owner.

What we can do to fix it

- Manage the product owner's expectations in terms of the system's features at planning time.
- Put more pressure on the stakeholder to uphold their end of the bargain in terms of project resources.
- Increasing test coverage and accuracy, and having the team be reasonable for ensuring that every release is demoable.