BOSTON
UNIVERSITY

Project Status Report 6

Team1

CS-673 (Software Engineering)

Fall 2024

By

Sergio Khalil

Yuhang Zhang

Asma Asiri

Brad Nissenbaum

Shivaang Kumar

Team Meeting Minutes:

Date: 12/03/2024

Attendees:

- Sergio Khalil
- Yuhang Zhang
- Asma Asiri
- Brad Nissenbaum
- Shivaang Kumar

Discussion Points:

- Discuss the effectiveness of using the ApplicationsContext for managing and passing application progress and task data.
- Evaluate the current implementation of the task statuses and the progress bar.
- The View Details button functionality could be a point of discussion.
- Discuss the page's accessibility, including keyboard navigation, screen reader support, and compliance with accessibility standards.
- Discuss whether role-based permissions (e.g., admin, agent, client) are necessary to control who can manage users.
- Reflect on the value of enabling agents to send tasks back to clients with comments. Does this enhance collaboration or introduce delays if used excessively?
- Explore extending the task collaboration feature to include real-time updates (e.g., notifications or activity feeds).
- Are the current search filters (address and ID) sufficient, or should additional filters (e.g., application status) be added?
- How should archived applications be handled—should they remain searchable?
- Are backend API endpoints optimized for dashboard needs?
- Should advanced filters or export capabilities be added if time permits?
- Does the layout support agents' workflows effectively?

Capture Decision:

- We will continue using the ApplicationsContext for managing and passing application progress and task data, monitoring scalability for future needs.
- Enhance the current implementation of task statuses and the progress bar by adding animations or hover tooltips.
- Expand the View Details button functionality to display task-specific information in a modal or separate section.
- Implement role-based permissions to control critical actions like user management, archiving, and deletion.
- Retain agents' ability to send client tasks with comments, introducing notifications to avoid excessive delays.
- Explore extending task collaboration features to include real-time updates, such as notifications or activity feeds.
- Add search filters, such as application status, to complement existing criteria like address and ID.
- Keep archived applications searchable but marked to prevent confusion, with guidelines introduced for archiving processes.
- Optimize backend API endpoints for search, archiving, and collaboration to improve performance and align with front-end needs.
- Consider implementing advanced filters and export features as "nice to have" options if time permits.
- Refine the layout and workflows to support agents better, prioritizing usability improvements like more apparent feedback mechanisms.

Action items:

- Share the implementation with team members for feedback and ensure it aligns with project requirements and standards.
- Incorporate suggestions from the team to refine the Application Progress Page and related features.
- Update the routing logic to include the Application Progress Page as a standalone route or as part of a multi-page flow.
- Test routing functionality to ensure smooth navigation between pages.
- Implement proper validation and error handling for file uploads, including checks for file size and format restrictions.
- Add real-time feedback during uploads, such as progress bars or success/failure notifications.
- Provide real-time notifications to clients when a task is returned with comments.

- Prepare contingency plans, such as transitioning to alternative state management solutions.
- Expand the "View Details" button to display task-specific information in a modal or separate section.
- Ensure the new feature integrates seamlessly with the overall workflow.
- Implement role-based permissions for critical actions like user management, archiving, and deleting applications.
- Test search functionality to ensure accuracy and performance with expanded filters.
- Mark archived applications and ensure they remain searchable with proper labeling.
- Review and optimize backend API endpoints for search, archiving, and collaboration to improve performance.
- Ensure APIs are aligned with the front-end's current and future requirements.
- Focus on user-friendly interfaces for configuring filters and managing exports.

Team Contribution:

-Asma Asiri: I developed the Application Progress Page to track and display the progress of tasks within the application. The page includes a visually dynamic progress bar and a detailed task table with columns for task names, notes or instructions, file uploads, submission statuses with color-coded indicators, and a View Details action button. To maintain simplicity and consistency with the existing project structure, I incorporated all styles as inline styles using JavaScript objects within the component. Additionally, I addressed potential issues like missing npm scripts with step-by-step resolutions. This work ensures the Application Progress Page is functional, user-friendly, and ready for integration, providing a clear and effective way to track task progress within the application.

-Sergio Khalil:

- Added functionality to manage users (add and delete) inside an existing lease application.
- Implemented file upload within tasks to attach relevant documents.
- Enabled agents to send tasks back to clients with comments for better collaboration.
- Provided agents with the ability to download uploaded files for offline access.

- Brad Nissenbaum:

- I worked on the Agent dashboard page. I added functionality to enable agents to search for applications by address and ID.
- Started writing code to archive and delete applications.
- Started coding search capability by applicant name.

- Held meetings with the backend team to assess our progress. List what is complete, what remains to be done, and a list of “nice to have” items to work on if time allows.
- Tested functionality of finalized components to make sure they were behaving correctly.

- Shivaang Kumar

- I refined the Add Application Page layout to ensure a user-friendly design and consistent visual structure.
- Enhanced the overall UI/UX of the platform by aligning styles across pages for a polished and professional look.
- Conducted testing to validate the usability and responsiveness of the front-end changes, ensuring they met project requirements.

- Yuhang Zhang

- I added the front end of the final success page to provide clear feedback to users after the operation is completed.
- I optimized the design of some non-core front-end elements that can enhance the user experience to ensure the unity of the overall style and further enhance the user experience.
- The layout of the form and the prompt information of the input box have been adjusted to improve the convenience and accuracy of users filling in information.

GitHub group project repo:

GitHub repository:
https://github.com/sergioBU2024/CS673Team1-Real-Estate-Checklist-Tracker