Team 1

Real Estate Checklist Tracker “RealList”

CS-673

The Idea

- Streamline communication between real estate agents and their clients
- Help efficiently track the progress of rental applications - including submitting necessary documents and payments
- Ensure transparency
- Reduce manual paperwork during the leasing process

The Need

Our market research shows:

- “Most of the communication was done in one long email chain (over 60 emails), which became hard to track.”
- “Being able to go through the application/approval/moving process all in one place vs various email and text threads would be helpful.”
- “A shared checklist of completed and incomplete requirements and “per-issue” communication would be handy.”

Scope

- **Checklist Creation:** Agents can customize checklists based on individual client needs.
- **Real-time Notifications:** Agents and clients receive notifications about task progress, including completed documents or payments.
- **Document Management:** Agents and clients can upload, store, and manage documents in a centralized repository.
- **Status Dashboard:** Agents and brokers can view client progress through a dashboard.
- **Integration:** The system integrates with third-party credit and background check tools, with results uploaded for decision-making.

Goals

The **Real Estate Checklist Tracker** project aims to create a system that improves communication and document tracking between agents, brokers, and clients during the leasing process. It enables agents to manage checklists, clients to track application progress, and both parties to handle documents efficiently, reducing delays and ensuring transparency.

Target Users

- **Brokers/Managers:** The owners/managers of the brokerage who manage the agents.
- **Real Estate Agents:** Professionals who manage property rentals.
- **Clients:** Individuals looking to rent real estate properties.

Team members’ roles and responsibilities

- Sergio Khalil: Backend Developer, Scrum Master
- Yuhang Zhang: QA Engineer, Front End Developer
- Asma Asiri: Design by using Figma, Front End Developer
- Brad Nissenbaum: Project manager, Backend Developer
- Shivaang Kumar: Front-end Developer, QA Engineer

High Level Requirements (Functional Requirements)

- User Authentication and Roles.
- Checklist Creation and Management.
- Document Upload and Management.
- Real-Time Notifications.
- Dashboard for Monitoring Progress.
- Integration with External Services.
- Search and Filtering Capabilities.
- Data Security and Privacy.
- Audit and Logging.

High Level Requirements (non-Functional Requirements)

- Performance Requirements
- Security Requirements
- Usability Requirements
- Reliability and Availability
- Maintainability and Support
- Compatibility Requirements
- Disaster Recovery
- Logging and Monitoring
- Scalability Requirements

GUI and API Updates

APIs:

- Login and Registration
- Application CRUD

GUI:

- Steps after Login and Registration Pages (DEMO)

Progress Updates

Frontend:

- Completed **Login and Registration Pages**.

Backend:

- We implemented **Login and Registration Functionality** with password encryption.
- **JWT Token Authorization** for secure user authentication and session management.
- Fully functional **CRUD operations** for:
  - **User Management** (Create, Read, Update, Delete users).
  - **Lease Application Management** (Create, Read, Update, Delete lease applications).

Security:

- Integrated JWT for securing all API endpoints.

Upcoming Milestones

- Add the capability for an agent to send a request to the client to create an account and upload documents
- Add step tracking/checklist functionality
- Add functionality for an agent to request documents and information
- Add functionality to upload documents
- Integrate lease signing services like DocuSign
- Add two-way communication/messaging?
- Add agent/broker dashboards?

Demo

Although a fully functional demo is yet to be completed, the groundwork is set, and we are prepared to integrate further features, such as lease applications and other critical components, in the next phase.

Risks, project timeline & plans, etc.

Risks:

- Competing deadlines with other classes/work.
- Unexpected absences - Sickness etc.
- Fixing bugs takes longer than expected.

Timeline:

- Start **testing** in early November.
- Finish the **backend** by mid-November.
- Finish **front** by the end of November.
- **Final product** and end-user evaluations will be done by early December.

Conclusion; Q&A

Progress Made:

- We have successfully implemented secure login and registration functionalities, including JWT token-based authorization and entire CRUD operations.
- Established the groundwork for future development, particularly around lease applications.

Next Steps:

- Integrating document uploads.
- Adding checklist functionality and agent-client communication features.

Open to Feedback:

- We appreciate any feedback on our progress and suggestions to help us refine our approach as we move into the next development phase.