Property Management Platform Proposal

JAC FOREST LTD
12/01/17
David Mifsud

Proposed Solution

Overview: To design and build a mobile-first, web application to facilitate the management of multiple properties owned by different landlords. The platform will act as a digital record for all tenancy contracts of these properties meaning that tenancies can be tracked, edited, renewed and updated.

Outlined below are the user interfaces that are needed to fulfill the requirements of the application and their relevant functionality.

User Interface

1. **Login** - In order to protect sensitive information within the platform, user login flow must be added. This view is the gateway to the application and will contain username and password fields as well as a login button. Admin login details will be manually added to the database and issued to the primary user. If additional users need access to the platform, the primary user must enter the email of this user so that an activation link can be sent to them. The link will bring them to a page that will allow them to submit a username and password.
2. **Dashboard** - Once a user logs into the application they will be taken to the user dashboard. The dashboard will contain navigation to all key aspects of the platform which are: Landlords, Search.
3. **Landlords Page** - This page contains information on all of the current landlords on the platform. There will be functionality to add additional landlords to this page. If a landlord is selected then you are taken to their properties page.
4. **Properties Page** - This page contains a list of properties that are owned by the chosen landlord. As in the landlord's page, there will also be functionality to add more properties to the page. Once a property is clicked you are then taken to one of two places, the flats page if the property has multiple flats within it, or the tenancy page if the contract is for the whole property (e.g. houses).

5. **Flats Page** - This page is similar to both the landlord and property pages. It contains a list of flats within the chosen property. Upon clicking these flats the user is then taken to the tenancy page. Note: no functionality to add more flats to this page under the assumption that the properties won't change.
6. **Tenancy Page** - There is a tenancy page for each flat or property within the system. Functionality needed for this page is the ability to input when the tenants pay their rent. This will automatically determine whether or not it has been paid on time and then display this information on the page. A tenancy needs to have the functionality of renewing the contract as it stands, renew with a change of details and also to create a new tenancy if it has not been renewed by the previous tenants. This page will contain the following information: Address, Tenancy Dates, Rent, Deposit, Notes, Payment history, Tenants. All of this information is saved to the database so that it can be referenced at a later date. If a tenant is clicked then it you are navigated to the tenant page.
7. **Tenant Page** - The tenant page is a profile page for all tenants who have rented a property within the system. It will give a history of their tenancies to date which can be printed off for reference. This history will include a breakdown of their payment history - how often they paid on time and how much they paid.
8. **Search** - This page can be navigated to from the dashboard. It will hold all functionality for searching through the platform data for tenants and properties by name and rental amount. The search results will be displayed beneath the search bar after the search has been made. These results will be clickable which will take you to the relevant tenant or property page.
9. **Property History** - Each property will have its own tenancy history that can be reviewed. The history for each property will be accessible through a link in the property pages. They will contain information on all previous tenancies of the property.

Database

In order to save application data, the platform will need to be linked to a database. The following data will need to be saved:

- Landlords
- Properties
- Tenants
- User data (username & password)

Various options for databases will be explored and, following a detailed review and in consultation with the client, one will be selected.

Process

Stage one - Setup, training, design and scaffolding

- Wireframes - basic designs of the user interface
- Environment Setup - setting up hosting, domain, server and development tools
- Systems Architecture Planning
- Database Schema Design
- Database Setup
- Load Initial Database Data
- Setup Views

Stage two - Functionality

- Search functionality for Properties, Tenants and Landlords
- Hook up search to the database
- Landlords (+ ability to add more)
- Add Properties (+ ability to add more)
- Flats - Add tenants, rent information, tenancy dates
- Input rent payments from tenancies
- Renew tenancies as is, Renew tenancies with changes
- Add new tenancy to property

Stage three - Deploy

- Live deploy to domain
- Product launch
- Handover of codebase and platform information

Technologies

Proposed technologies:

- HapiJS (backend) + carefully selected Node.JS modules
- HTML5, CSS and Javascript - including frameworks (React, Riotjs)
- Redis database - ensures both fast page load times and searchability with location
- Amazon Web Services and heroku (hosting) - industry standard
- Development tools - Travic CI, linting, codecov

Fees

The breakdown for this body of work in terms of developer hours is as follows:

Step one: 40 developer hours (including wireframing)
Step two: 40 developer hours
Step three: 16 developer hours

96 developer hours at a rate of £25/hour

Total: £2400

Additional costs covered by the client:

The fees stated above do not include any payment or purchase of domain names or web hosting services (roughly £5/month). These will be covered by the client.

Project Tracking

Client inclusion is extremely important throughout the entirety of the process. There must be a continuous feedback loop between the work being done and the client.

The primary mode of tracking and approving work is done through https://github.com in the form of issues. (*Github is an online code repository that tracks progress of a project*)

Once an issue has been completed, it will be assigned to the client for approval by testing the assigned feature on the platform. An introductory call/screenshare will be necessary in order to familiarise the client with github and the issue process.