

RENTACTION

Charan Narayanan
Master's in Applied Computing
University of Windsor
 Windsor, Canada

Subuh Khan
Master's in Applied Computing
University of Windsor
 Windsor, Canada

Arif Ahmed Ayon
Master's in Applied Computing
University of Windsor
 Windsor, Canada

Muhaimeen Ahmed
Master's in Applied Computing
University of Windsor
 Windsor, Canada

Abstract— One can feel happy in their rented homes when not only the place is comfortable for living but also only when there is a smooth interaction between the property owners and the tenants themselves. A thing that is missing among these people i.e. the tenants, the property owners or the property managers is an effective interaction system. The aim of this project is to develop a simple but useful solution of a web application where the interaction between the property owners, property managers and the tenants is more effective and user friendly. The web application comes with the following features, space for rental marketing for property owners, a chat system where the owners, managers and tenants can message instantly with each other and a maintenance system where the tenants can request for maintenance if they find any issues at their homes.

I. INTRODUCTION

A House or Apartment rental is becoming so common these days. In countries like Canada, Around 32.2% of households are rented homes which are around 4.6 million of the total 14.1 million households.^[1] Around half a million of them are students which is a 20% increase from 2016.^[2] 90% of the international students prefer off campus residency which means there are high chances of increase in the number of people who are going to rent a home. Apart from students, there are so many immigrants every year. The federal government is aiming to bring a total of 310,000 newcomers to Canada in 2018. By 2020, the yearly total will hit 340,000.^[3] These proud migrants from other parts of the world are mostly are valuable people for the country. Initially, they do not look for a permanent settlement in a town or a city and one cannot expect them to own a house when they arrive to a new country. This makes them hunt for a temporary stay by renting a house.

Over the years landlords/property managers have had a problem in maintaining and managing their customers and their own records. Management has become difficult because of issues that include data increase day to day. Storing and maintaining all data manually is very difficult. Lack of these crucial requirements makes management of

the tenants and houses very difficult.

An increasing gap of communication persists among these renters and the property owners. Existing applications are having a management system for owners, managers and the renters but a proper two-way communication is missing. To overcome this problem, a more cozy way of interaction between the property owners, property managers and the tenants has been developed that is, a web application. This web application makes sure the communication is consistent and flawless right from a tenant renting a house till he/she moves out. To be short, RentRaction is a social media for tenants, property owners and property managers.

The main features of application will be:

1. Online payment integration
2. Easy communication via chat systems
3. Maintenance request and progress monitoring
4. Invoicing and billing
5. Reminders & Alerts
6. Lease Renewal
7. Tenant Screening

II. PROJECT DETAILS AND METHODOLOGY

A. Architecture

The Model View Controller Architecture has been used. The pattern encourages the development of modular systems, allowing developers to quickly update, add, or even remove functionality. The model represents the data. However, a model cannot simply be summed up as a database, or a gateway to another system which handles the data process. The Model must act as a gatekeeper to the data itself. The View is where data, requested from the Model, is viewed and its final output is determined. Traditionally in web apps built using MVC, the View is the part of the system where the HTML is generated and displayed. It also ignites reactions from the user, who then goes on to interact with the Controller. The view also

displays the model data, and sends user actions (e.g. button clicks) to the controller. The Controller can be summed up simply as a collector of information, which then passes it on to the Model to be organized for storage, and does not contain any logic other than that needed to collect the input. The controller provides model data to the view, and interprets user actions such as button clicks.^{[4][5]}

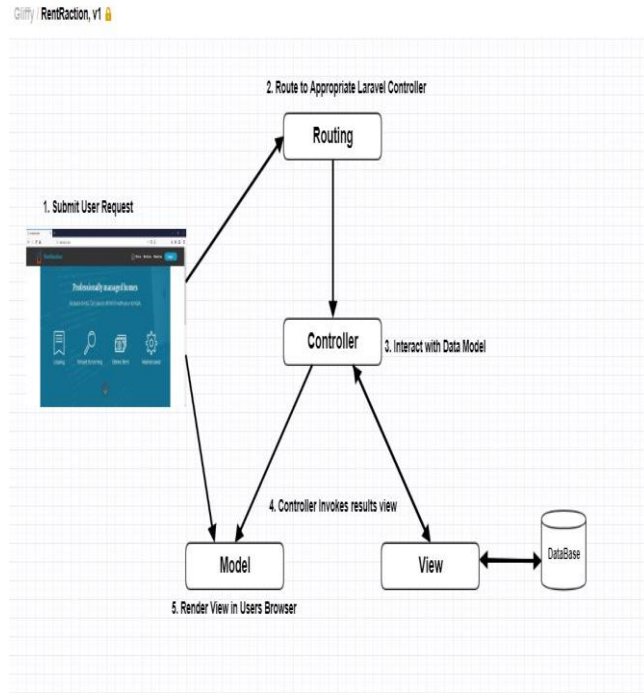


Figure 2.0 System Architecture ^[6]

B. Platform

We have used Laravel platform, a PHP web framework which follows the MVC pattern. It has the feature of a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance.^[7]

C. Methodology

The methodology explains what kind of way or approach the team has put forward to develop the application.

A Scrum methodology has been followed, which is a type of an agile methodology. Scrum follows a set of fixed length iterations in which the product is developed. Each of these iterations is called as Sprint.^[8] Typically, each sprint is fixed somewhere within 2 weeks to 1 month. In this project, a sprint every 2 weeks has been fixed and developed accordingly. The start of each Sprint happens with a Sprint Planning meeting which finalizes the backlog/work items planned for that sprint. An estimation of the Sprint is also declared in this phase.

D. Design

The Front end of the web application system has been designed using HTML, CSS and JavaScript. HTML

makes up for the text on the page, CSS makes it pretty, and JavaScript makes it interactive. For the backend, the designing has been done with PHP which is a server-side scripting language and MySQL for the database.

III. EXPERIMENTAL SETUP

A. Implementation details

Stage 1 - At the initial stage, the implementation proposal was reviewed and at first, an outlined framework was designed with some dummy data and some basic front end html pages.

Stage 2 - The entities are connected as required and the backend i.e. the database has been improvised.

Stage 3 - A more user friendly responsive design was implemented and a complete demonstration of the project with data was done and the final product was delivered with scope for future improvements.

B. User Scenarios

As mentioned before, the system has three main potential users, the property owners, the tenants and the property managers. Initially, anyone can view the home page of the web application where a brief introduction on the application is available and if he/she is interested, can sign up as a property owner, tenant or a manager.

Once a user is signed up, he/she can login to the web application based on their respective roles.

Owner – An owner can see the list of the properties he/she owns. The list can be updated by adding, removing or updating a property's details. The owner can also see if their tenants have paid the rent or it is due and send a reminder. Another feature is that, the owner can see the list of applicants who have applied to rent his/her property and can accept or reject the applicants based on their details. The owner can check if a tenant has requested for any maintenance activity in their respective homes and act on it by assigning the task to the manager or the owner himself. A track of the owner's expenses is also available and the owner can also chat with the tenants and managers under him/her.

Tenant – A tenant can see the list of available properties and can apply to any if available, the terms and documents for each property can be viewed. A tenant can also request maintenance if he/she finds any issues at their rented homes. The tenant gets notified when the rent is due and the rent can be paid online. The tenant can also chat with her/his owner.

Manager – A manager sees the maintenance request assigned by an owner to him/her. The manager should be able to work on the request and update the owner and tenant that whether the request has been done successfully or not. The manager can also chat with the tenants and owners with respect to the maintenance requests.

C. Testing

Unit testing has been made on the application during the each stage of development and as a whole. Unit testing is a level of software testing where individual units/components of the software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. After unit testing, integration testing has been carried out on each module, integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as the output as the integrated system ready.

Testing Summary:

Web: Login, Registration, Add Property, Search Property, Apply for rent, Request for maintenance, Dashboard, Profile, Sign Out;
 Total Bugs: 09 [Critical = 03, Major = 02, Minor = 04]
 Total Opened: 09
 Total In-Progress: 00
 Total Fixed: 09
 Total Verified: 09

The following modules of the system are tested as a part of unit testing and integration testing.

Unit(s)/Module(s)	Unit test	Integration test
Login	Done	Done
Registration	Done	Done
Add Property	Done	Done
Apply for rent	Done	Done
Request for maintenance	Done	Done
Dashboard	Done	Done
Profile	Done	Done
Sign Out	Done	Done

Table 3.1 Tested Modules

D. Findings and Challenges

It has been found that many existing systems have an interaction system but they are not effective. A system with both maintenance system and a chat system will be more effective. Also that chat system is designed in such a way that the owner can chat only with his tenants and managers under him/her.

A challenge that was faced was to give equal importance to all the users of the system. When there are three different potential users, each one should have their own freedom. This was overcome by designing separate dashboard for

each of the users respectively and defining a proper flow for each of them.

IV. FUTURE WORKS

A. Maintenance ticketing system

The owner will be notified when a tenant requests for maintenance and it will be taken care by a manager if he/she is available. But apart from this, a ticketing system would be more efficient where each time when a tenant requests for maintenance, a ticket with an incident number will be generated and it will be assigned to the respective owner, the owner will then assign it to the manager or he himself works on it. The ticket/incident will then be resolved when the maintenance work is done and the tenant will be notified.

B. To implement the manager part more efficiently

The manger part will be implemented more effectively along with the ticketing system and with prompt notifications when a task is assigned to him.

V. CONCLUSION

Effectively resolving the communication issues is important to an owner-tenant relationship, Rentracton web application will be a useful tool for simplifying the communication gap among the owners, managers and the tenants.

VI. REFERENCES

- [1]. BNN. (2018). *Census: Home ownership rates drop as more Canadians opt to rent - BNN Bloomberg*. [online] Available at: <https://www.bnnbloomberg.ca/census-home-ownership-rates-take-historic-dip-as-more-canadians-opt-to-rent-1.895239> [Accessed 5 Dec. 2018].
- [2]. Katem, E. (2018). *Canada is home to nearly half a million international students - Canada Study News*. [online] Canada Study News. Available at: <https://www.canadastudynews.com/2018/03/19/canada-is-home-to-nearly-half-a-million-international-students/> [Accessed 5 Dec. 2018].
- [3]. Scotti, M. (2018). *Canadian government wants 310,000 immigrants in 2018, 340,000 a year by 2020*. [online] Global News. Available at: <https://globalnews.ca/news/3836805/340k-immigrants-per-year-by-2020-government-unveils-new-immigration-targets/> [Accessed 5 Dec. 2018].
- [4]. Tomdalling.com. (2018). *Model View Controller Explained — Tom Dalling*. [online] Available at: <https://www.tomdalling.com/blog/software-design/model-view-controller-explained/> [Accessed 5 Dec. 2018].
- [5]. March 04, 2. (2018). *PHP Master | The MVC Pattern and PHP, Part 1*. [online] SitePoint. Available at:

<https://www.sitepoint.com/the-mvc-pattern-and-php-1/>
[Accessed 5 Dec. 2018].

[6]. Figure 2.0 created [online] using
<https://www.gliffy.com/> [Accessed 30 Nov. 2018].

[7]. En.wikipedia.org. (2018). *Laravel*. [online] Available
at: <https://en.wikipedia.org/wiki/Laravel> [Accessed 5 Dec.
2018].

[8]. En.wikipedia.org. (2018). *Scrum (software
development)*. [online] Available at:
[https://en.wikipedia.org/wiki/Scrum_\(software_developmen
t\)](https://en.wikipedia.org/wiki/Scrum_(software_development)) [Accessed 5 Dec. 2018].