

Scope and Charter



University of British Columbia Okanagan
COSC 499 - Summer 2019

Prepared by

Document Owner(s)	Role
Jae Ung Kim	Developer
Jan Reisler	Developer
Stephen Okanlawon	Developer

Project Overview

Project Name	VRK Solutions Inc. Website design		
Project Scope & Charter Author	Jae Ung Kim & Jan Reisler & Stephen Okanlawon		
Creation Date	May 19th, 2019	Last Revision Date	May 31st, 2019
Project Requestor	VRK Solutions Inc.	Project Manager	Stephen Okanlawon
Project Charter Status (Pending / Approve / Reject)	Pending	Date of Project Approval	
Proposed Project start & End Date	Start: May 2019 End: August 2019		

Approved By

Scott Fazackerley

Date

Client

Date

Project

Purpose and Objectives

The purpose is to build a website that serves as a portal where landlords can advertise their available home rental units which tenants can then view and contact landlords about. Users can login as either a tenant or a landlord depending on if they're looking to sell or rent out a suite/house or seeking residence. The website will be simple and intuitive as to avoid clutter and to make the site more appealing to users so they do not feel that the site is too complicated and overwhelming. It will allow to filter out listings to find specific listings that the user is interested in, as well as providing contact information and a wide range of information for each listing. Navigation between pages will also be simple and easy.

Related Success Criteria

The project will be successful if the website meets all the requirements and features that the client has listed. The website should allow users to search through the listings efficiently using features like; map searching via distance, scroll bar searching for different price ranges, and searching through other filters like smoke-friendly, pet-friendly, amongst others. We also want the website to be agile so that the website updates in real-time and can also be modified over time.

Project Team

- Jae Ung Kim
- Jan Reisler
- Stephen Okanlawon

Stakeholders

- Team
- Community members
- Customers
- Valentin Koch/ Sven Stein
- UBC

Requirements

Functional Requirements

- System will allow users to sign up and store the sign up information
- System will allow users to log in
- System will allow users to sign up as tenant or landlord and switch between them effortlessly
- System will allow users to view listings on map based on location and distance
- System will allow users to enter property information and post ad
- System will allow users to filter properties to find specific listings
- System will offer admin login via an admin portal
- Admin portal gives admin ability to delete user account and/or ads
- UI is clear and only includes relevant info on each page and groups all relevant data together
- System will offer password recovery methods
- System allows general search of listings without login

Non-Functional Requirements

- Listings must load and be displayed in less than two seconds when a user searches for specific listings or is filtering them
- Website has responsive design
- Users only need email, name, and password to register an account
- Logging into the website must take less than one second
- When creating listings or an account, all fields must be correctly handled of any range of characters and empty fields

Technical Requirements

- The website will be running on an online server
- The website front end will be created with CSS, HTML, and JavaScript/JQuery
- The website backend will be created with PHP and AJAX connected to a MySQL database
- Website can be accessed and correctly viewed with any mobile and desktop updated browser
- MVC design pattern

User Requirements

- User will be able to search all listings without login.
- User will be able to log in

- User will be able to sign up
- User will be able to sign up as tenant or landlord and switch between them
- User will be able to view listings on map based on location and distance
- User will be able to enter property information and post ad as a landlord
- User will be able to filter ads to find specific ads as a tenant
- Admin will be able to delete user account and/or ads via the admin portal
- User will be able to recover password

Admin Requirements

- Allow admins to add users upon request
- Allow admins to delete users upon request
- Allow admins to update users upon request
- Allow admins to delete listings upon request
- Allow admins to recover user account
- Allow admins to recover user listings
- Allow admins to recover user email and passwords
- Allow admins to view user and listing

Assumptions and Constraints

Assumptions

- Developers are comfortable programming in PHP.
- Developers are comfortable using GitHub.
- Developers will be provided with necessary materials from the client to complete the project.
- Developers will complete the project in a timely manner during the duration of the project (13 weeks).
- Client will respond to the developers requests or concerns as soon as possible.
- Developers will not run into technical problems.

Constraints

- The developers and clients will be away on vacation during the summer.
- Tight time constraints - the developers only have 13 weeks to complete the project.
- Knowledge constraints - the developers are 4th year University students.
- Some of the features clients require might be hard to implement for the developers.
- User testing difficulties - After the deployment of the website, getting users to participate in user testing can be difficult.

Project Description and boundaries

Description

- Once the project is completed, we will have a complete free and working website similar to trivago and airBnB but for rental properties. Our website will allow landlords to first rent out their properties but they can later decide to list the properties for sale if they want to. There will be lots of features that are helpful for all users of our website. For example, similar to trivago we will allow users to search listings via map so users can see the exact locations of listed properties. There will be more features implemented to this map search to guide users to find the best possible find for them.

Boundaries

- The project has to be finished prior to August 9th, 2019. However, the code has to be finished before, so developers can run testings and fix bugs if any.

High Level Risks

- Cost risks: The project is done by University students (developers) as their graduation requirements; thus there is absolutely no cost. However, since it is also for actual company's website, developers might have problems with features that need subscriptions.
- Schedule risks: developers or client can get sick and delay the project process. Developers might need more time than expected time for certain features.
- Performance risks: Some of the website features might be too hard to implement for University students.
- Lack of communication between developers and the client.
- Poor team organization and team performance.
- Conflicts might arise between team members.
- Missing people

Budget Requirements

- Monetary value for this project is \$0 since all necessary materials will be provided from the client.
- Developers will each spend at least 16 hours per week for 13 weeks on the project.
- IDE we have chosen (atom) is free; thus no cost to implement it.

Environmental Constraints

- Extreme heat environment in Kelowna might cause developers to get ill.
- University WIFI might have connectivity issues resulting in delay in development.
- Website server might go down.
- Power outage in the middle of developing process might affect unsaved data. (If any)
- Website could be hacked or intercepted.

Development Process

Communication plan	When	What	Who
	May 22 2019	First meeting with the client	Jae & Stephen & Jan
	May 31 2019	Scope & Charter Requirements	Jae & Stephen & Jan
	June 7 2019	Design documents	Jae & Stephen & Jan
	August 8 2019	Final presentation	Jae & Stephen & Jan

Development Standards and tools

- During weekly meetings, the project team's integration lead, Jan, will make sure each team member made some progress with coding.
- During weekly meetings, integration lead will run a linter program to ensure that codes from the team follow the same style guidelines.
- Once everything checks out and merged to master branch, the project team will go through multiple testing (testing if certain features work with another) to ensure our latest version is working.
- As for version control, GitHub will be used. Every week, integration will make sure all of our codes are committed and pushed to the master branch at least once a week, and the master branch will always maintain the latest working version of the code.
- Different branches will represent different features of the website.

- Trello will be used to keep track of who is working on which features.
- Every merging to master branch will be documented by the technical lead.
- There will be a weekly meeting to ensure the project is making progress every week.

Work Breakdown Structure

Task List	Estimated Hours		
	Jae	Jan	Stephen
1.0 Documentation			
1.1) Scope and Charter	15	5	5
1.2) Weekly Reports	13	10	13
Total number of Hours Assigned Per Team Member	23	15	18
Weekly Average of Hours Assigned Per Team Member (13 weeks)	1.8	1.15	1.4
2.0 Learning			
2.1) Developers should review HTML / CSS	2	1	3
2.2) Developers should review PHP	2	2	6
2.3) Developers should review GitHub	1	0	3
2.3) Developers should learn how Bootstrap works	4	3	6
2.4) Developers should review Basic networking (server, XAMPP, etc)	2	1	5
2.5) Weekly Team meetings (Cumulative)	13	13	13
Total number of Hours Assigned Per Team Member	24	20	36
Weekly Average of Hours Assigned Per Team Member (13 weeks)	1.9	1.5	2.8
3.0 Designing			

3.1) Designing the layout of front page	2	2	2
3.2) Designing the layout of account pages	2	2	2
3.3) Designing name & logo of the company	5	1	
3.4) Designing the layout of map search page	2	2	5
3.5) Data models and diagrams (UML, ER, etc)	5	5	7
3.6) Designing the test plan	2	4	1
3.7) Finding correct/best tools to use for project	1	1	1
Total number of Hours Assigned Per Team Member	19	17	18
Weekly Average of Hours Assigned Per Team Member (13 weeks)	1.5	1.3	1.4
4.0 Developing the website			
4.1) Building front-end UI pages with bootstrap and make mobile compatible (Home page, browse page, admin page, account page, create listing page)	15	20	10
4.2) Building Database and make accessible to website	10	10	20
4.3) Integrate and display map engine on browse page	20	20	15
4.4) Implement functioning accounts and be able to login, register, post listings, and contact landlord correctly	20	25	20
4.5) Implement search function which displays listings and can be narrowed down with filters	15	25	15
Total number of Hours Assigned Per Team Member	80	100	80
Weekly Average of Hours Assigned Per Team Member (13 weeks)	6.1	7.7	6.1
4.0 Testing			
4.1) Integration Testing	5	10	5
4.2) User Testing	5	7	6

4.3) Usability Testing	5	7	5
4.4) Test plan for website	5	5	5
4.4) General	5	5	5
Total number of Hours Assigned Per Team Member	25	34	26
Weekly Average of Hours Assigned Per Team Member (13 weeks)	1.9	2.6	2.0
5.0 Other (Technical problems, missed details, change of requirements, variability, etc.)	30	30	30
Total number of Hours Assigned Per Team Member	30	30	30
Weekly Average of Hours Assigned Per Team Member (13 weeks)	2.3	2.3	2.3