Computing Project Technical Plan

Name: Alexander Gogl
Course: Computing
Mode: Full Time
Supervisor: Chris Casey

Title

Proprietor's Assembly

Summary

Proprietor's Assembly makes collaborating in a house or tenement easy. The main task is to develop a feature-rich single page application which provides a digital blackboard, forum capabilities, a maintenance log system and polls for decision making. As SEO plays a big part in finding and indexing pages online and as there are limited options for integrating such techniques into spas, one big challenge will be to find a viable solution for this problem.

Deliverables

The deliverables for this project include the finished product in form of a singe page application which will be made available for beta testing on a highly available server. Delivered functionality includes: rudimentary forum capabilities for creating threads and comments, a digital blackboard for announcements, a log and maintenance system which facility management can utilise to track and review repair or maintenance requests. In addition, it will be evaluated which option is most feasible and sensible for the integration of SEO and then will be integrated into the final product.

Constraints

Deadlines

Deadline	Form of Assessment	Size of Assessment / Duration
5th October 2018	Project Proposal	Around 2 pages
14th October 2018	Technical Plan	Around 4 pages
10th December 2018	Project Deliverable 1	As agreed with supervisor
18th March 2019	Project Deliverable 2	A project artefact
25th March 2019	Draft Project Report	Maximum 50 pages
14th April 2019	Project Report	Maximum 50 pages
May 2018 (TBC)	Video	Video demo of project deliverable
May 2019 (TBC)	Poster	A1 poster, interview
May 2019 (TBC)	Interview at Poster Session	10 minutes

Technology Constraints

As I already have started implementing some parts of the application in Vue.js and a potential project partner is also interested in having the application developed with Vue.js this will pose an external constraint.

This however doesn't change the fact that a thorough evaluation of frameworks has to be executed to identify possible shortcomings of Vue.js in the future. Furthermore, the database system (PostgreSQL) and REST API (Django rest framework) have already been developed by the project partner and are therefore provided to me at no cost.

Key Problems

If the product fails to deliver on usability or functionality, users are likely to refrain from using it. Additionally without providing persistency in terms of storing authentication details and other data securely the application will become very bothersome and insecure to use. Another key problem is how and where the application "lives", questions such as "Which underlying server structure should be used?", "SSD or HDD storage?" or "Is the use of load balancers sensible?" will have to be answered.

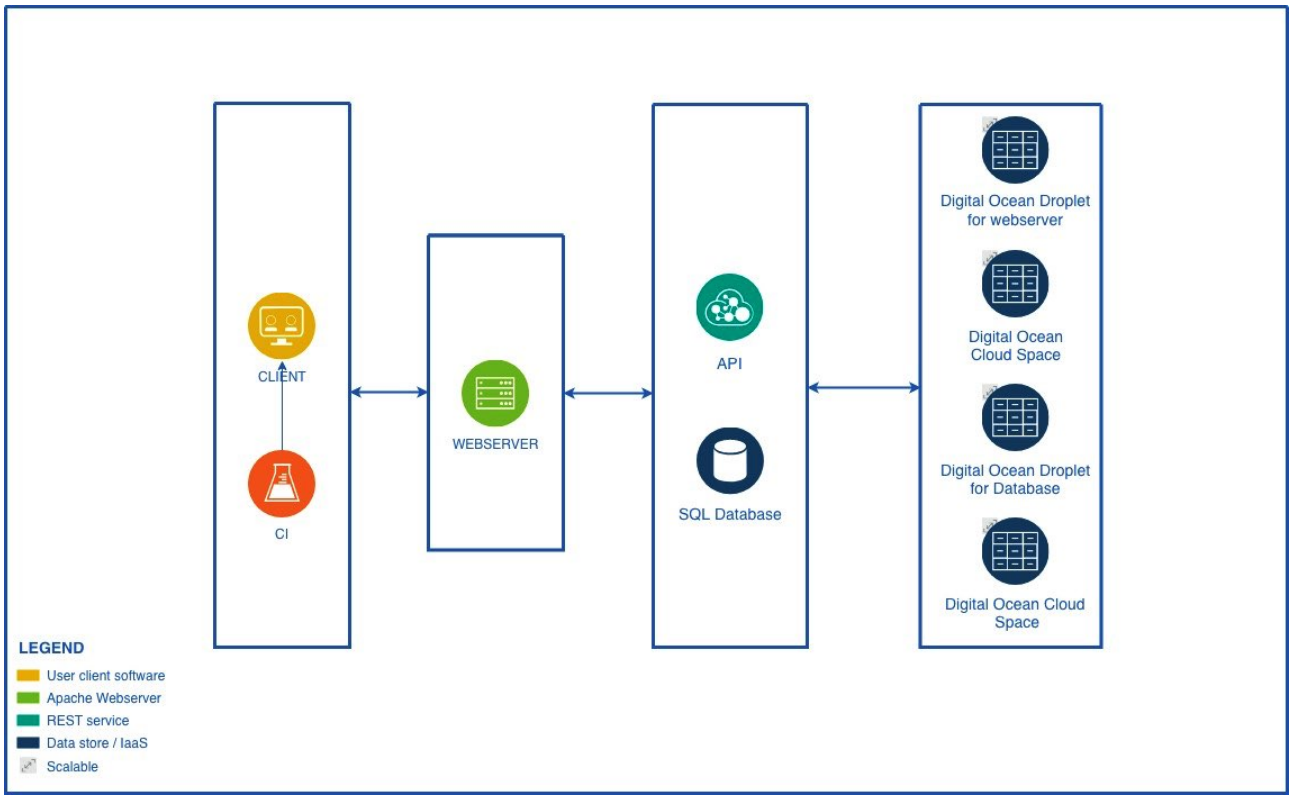
System and Work Outline

Key requirements

1. Comparison of existing technologies for building spas (including CSS / JavaScript etc.)
2. Implementation of functionality in form of a single page application
    1. Login / Register site
    2. Digital blackboard for making announcements
    3. Forum for discussing topics
    4. Log / maintenance systems
    5. Poll feature for guiding decisions
3. Writing tests for CI / CD
4. Evaluation of viability and reasonability of SEO techniques for spas
5. Integrating SEO into final product
6. Releasing production ready version

Technical outline

Systems diagram



This section describes how the project will be implemented in a more detailed manner, whereby the API and database are considered as a given. The following points are by no means a complete outline for developing the final product but rather give an overview of the most important steps - a lot of additional steps will have to be covered while developing.

Comparison of existing technologies

In order to build a fully functional single page application which is **scalable**, **efficient**, **secure** and **flexible**, different technologies have to be compared against each other. Frameworks like React and Angular are very well supported and backed by a big community, whereas Vue.js - a newcomer in this field has been very popular amongst developers in recent times. I will limit myself to comparing these three frameworks. The emphasis lies in identifying advantages and possible caveats of using Vue.js over React or Angular.

Implementation of functionality in form of a single page application

Authentication

Since authentication is one of the most bothersome parts and also a very crucial one in most projects, I will need to pay additional attention to this topic. There will need to be different components: a middleware to check if a user is authenticated between page changes, an interceptor which can intercept requests with the status code “401” for handling situations where the access token is expired and a persistent store such as cookies for storing token data so that the user will still be authenticated even if they reload the browser window. Furthermore, as the API utilises a custom token mechanism, no existing library can be used.

Data structure: ['token', 'token_expiry', 'user', 'refresh_token', 'refresh_token_expiry']

UI

As UI and UX are very important factors for user acceptance, I will need to carefully select a UI framework, preferably one that supports components for the chosen technology (React, Vue, Angular) as this makes creating a usable product drastically easier and iterating much faster. Bulma has been a safe bet for earlier projects.

Handling access rights

One aspect of the application will be how users interact with each other, therefore access rights are needed. There will be three different classes: “Proprietor”, “Tenant” and “House Management”.

Proprietors can create polls, blackboard entries, forum entries or add a new maintenance issue. Tenants can do everything what proprietors can except creating polls.
The House Management can create blackboard entries and edit maintenance issues (e.g mark as resolved)

Depending on which class the user is assigned to, the layout of the application has to dynamically change. If the user is a tenant with no rights for accessing polls then there is no point in showing them a link to the poll site.

Digital Blackboard

The Digital Blackboard is a way for making announcements like a tenants next grill party. What has to be created is: a site for creating board items and a site for viewing board items in a list.

Additionally the components for displaying a form, showing a single item and encapsulating items in a list have to be created.

Data structure: [‘title', 'content', 'tenement', 'created_date', 'owner', 'id']

Forum
The forum is used for discussing certain topics. There will have to be a site which shows all threads and a detail site for each thread. Additionally, like the blackboard a way of creating forum entries has to be provided - this can also be a normal html form.

Data structure: ['category', 'title', 'content', 'tenement', 'created_date', 'owner', 'id']

Log / Maintenance System
To provide the user an easy way to report issues (e.g “sink broken”) a maintenance system comes into play. As there will be more data involved than for the forum or blackboard use case, I will need to find a better way of displaying the data. I could imagine using an interactive table for that. What makes implementing this use case a bit harder is the fact that there is nested data: issue_status_changes[] is a list inside a list and I will manually have to get the most recent status change and pass it as a parameter for showing the actual HTML item.

Data structure: ['date', 'tenement', 'location', 'reference', ‘issue_status_changes[]’, 'owner', 'type', 'id', 'comments', ‘name’]

Polls
Polls are a means of deciding on certain options. If the houses facade needs a new colouring, proprietors can vote on an option. It is especially hard to implement a usable voting HTML block as I have to take various situations into account. When a user already voted on a poll, it needs to get propagated back to the backend and at the next page load, this poll block has to be flagged as alreadyVoted, as the user shouldn’t be able to vote twice. When it’s flagged there also needs to be vastly different styling showing a percentage, the number of votes and a progress bar which also indicates the percentage.

Data structure: ['tenement', 'created_date', 'owner', ‘poll_answer_options[]’, 'question', 'id', 'participation', ‘poll_given_answers[]’]

Consuming existing API
As the API and database have already been built for this particular project there should be no major caveats or problems, however, as there are some constraints in Django and the “Django Rest Framework”, some things might not work right away. For example when getting maintenance logs from the server in json format, the field “current_status” (e.g “Pending” or “Resolved") is not sent in the root object of a specific item - instead a list of all status changes is encapsulated inside a nested list, therefore I will need to iterate over these changes, sort them and add the field “current_status” post-getting.

Writing tests for CI / CD
As I have little experience in writing tests for JavaScript, I will need to research on clear instruction on how to do so. Furthermore I plan to use Jenkins as a CI system.

Evaluation of viability and reasonability of SEO techniques for spa's
SEO plays a big part in commercial success nowadays, therefore I want to bake SEO best-practices right into the project. Since there are very limited options available for single page applications I will have to find an appropriate solution. Especially server side rendering is really important and libraries like next.js (for React) and nuxt.js (for Vue) provide these capabilities out of the box. However, if a websites functionality is locked behind a login site, the best SEO won't get you very far. What I aim for in this project is to evaluate an appropriate approach of adding search engine optimisation to "Proprietor's Assembly". To do this, I will create 3 or more single page applications with (text) content and host them on a web server. Then I will use common website benchmarking tools to review the pages performance in terms of index-ability and SEO score.

Integrating SEO into final product
The findings of the prior evaluation will be incorporated into the by adding the most suitable approach to the final product.

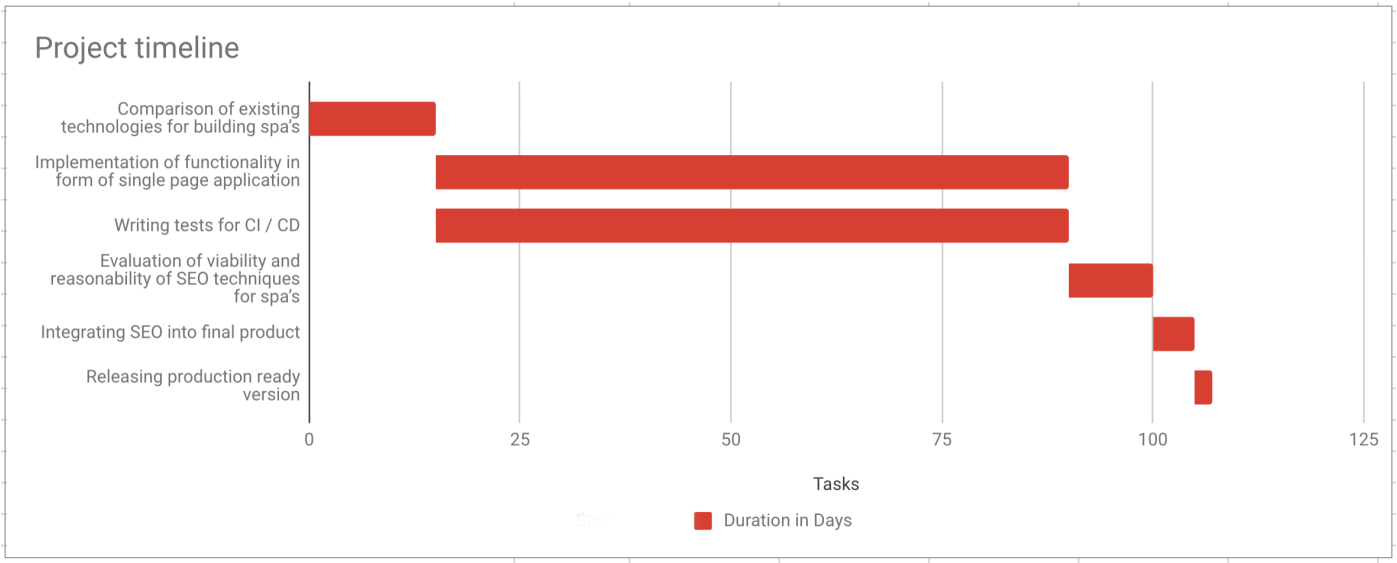
Releasing production ready version
For publishing the application online, I will use DigitalOcean as an IaaS.

Security Issues
Security issues are considered a very high risk for this project: if a hacker manages to gain full access to the backend they can alter user information or examine highly sensitive information. Two possible attack vendors in the application are the use of cookies and user generated input. Hackers could use these to their advantage by trying to do cross site scripting or hijack the cookies.

Cross Platform usability
Cross platform usability is not a big issue since single page applications are just a type of advanced static website. Every device that can utilise a browser will be able to use the service. However, special attention has to be payed to different device manufacturers and sizes.

Personal development
As I am already somewhat familiar with React, Angular and Vue but lack the advanced knowledge needed to build a fully functional, efficient and secure webapplication I will need to learn how to use these frameworks to my advantage. I also need to completely learn how to test JavaScript applications and integrate them into a production environment with tools like Jenkins.

Project Activities



Risk Analysis

Risk	Severity	Likelihood	Action
Work is lost due to unforeseen reasons.	High	Low	Use online version control system and additional offline backup.
Illness	High	Low	Complete tasks early prior to due date
Project too complex	High	Medium	Renegotiate project deliverables with project supervisor
Failure to integrate system as a whole	High	Low	Break project down into more achievable blocks
Failure to learn technologies used in project	High	Medium	Renegotiate project deliverables with project supervisor
Unforeseen application bugs in production	High	Medium	Use of CI system to identify errors during development
Cannot implement SEO into spa	Low	Low	Fallback to static pages for certain sites
Site is victim of a hack	Very High	Low	Mitigate security risks by implementing XSS and SQL injection defense mechanisms
Failure to understand logic for iterating over deeply nested items	Medium	Medium	Allow for this to happen by including an appropriate time buffer

Options

In terms of designing the user interface there are various options: Bootstrap, Vuetify, Foundation, Weex and Bulma, whereas the latter one is getting more popular everyday. Additionally in earlier projects I found it to be the easiest and most elegant framework.

The main options for project management are Scrum and Kanban, however, Scrum is designed for teams of three people and up so it is considered an inadequate solution for this project. Kanban on the other hand has excellent integrations with GitHub - my VCS of choice, making it the most suitable approach.

Potential Ethical or Legal Issues

When a user wants to use the service, they have to enter personal information. This includes the address of their home and their name - this can lead to potential issues related to the GDPR.

Providing access to the service for visually impaired users also poses an ethical issue, this however can be tackled fairly easily by providing aria tags to a big part of html elements and by letting the user choose the font colour and size.

Commercial Analysis

Factor name	Description	Is this a cost or a benefit	Estimated Amount	Estimate of when paid
CentOS	Operating System	Cost	£0	NA
Digital Ocean	IaaS	Cost	39 £ / month	Every month
JS Frontend Framework	Framework for building application	Cost	£0	NA
Working Time	Workforce	Benefit	200 * 60 * 1.5 = £18000	1/3 at the beginning, 2/3 when finished

Employability Contribution

Proprietors assembly is a way of connecting proprietors, tenants and house management firms. It provides an easy way to manage properties: Users can use a digital blackboard to make announcements, track issues with a maintenance log system or open a new forum thread. Proprietors can also see key metrics at a glance and use the advanced poll feature to help them guide their decisions.

Written with state-of-the art technology such as Vue.js and Nuxt.js, this is a highly advanced project which can also be exploited commercially.

References

Crispin, L. et al(2009). Agile testing : a practical guide for testers and agile teams. London: Addison-Wesley, 368p

Portney, D. (2018). An SEO's survival guide to Single Page Applications (SPAs). [online] Available at: https://searchenginewatch.com/2018/04/09/an-seos-survival-guide-to-single-page-applications-spas/ [Accessed 3 Oct. 2018.]

Samiksha, G. et al (2016). Continuous Integration for Single Page Application. [online] Available at: http://www.ijircce.com/upload/2016/february/132_55_Continuous.pdf [Accessed 3 Oct. 2018.]

Sébastien, C. (2018). Introduction - Nuxt.js [online] Available at: https://nuxtjs.org/guide [Accessed 3 Oct. 2018.]

Wilkie, M. (2006). Landlord and Tenant law. Basingstoke: Palgrave Macmillan, 355p

You, E. (2018). Introduction - Vue.js. [online] Available at: https://vuejs.org/v2/guide/ [Accessed 3 Oct. 2018.]

Ethical Assessment

This form is to be completed by the supervisor after discussion with the student and signed by both to confirm agreement.

<table>
<tr><th>Project Features</th><th>Action</th><th>Outcome (Tick 1)</th></tr>
<tr><td>1. This project does not use people as subjects or to evaluate a product and is not expected to raise any ethical or legal issues.</td><td>Any potential change with ethical implications to be discussed with the supervisor.</td><td>☐The supervisor approves the project under delegated authority of the Ethics Committee</td></tr>
<tr><td>2. This project has well known ethical issues, which can be dealt with through standard guidelines.

It does not involve vulnerable people (e.g. children) or animals</td><td>I have read the guidelines available on Blackboard. In carrying out this project, I will use relevant standard guidelines as indicated:
<table>
<tr><th>Issues</th><th>Guidelines</th><th>Student Initials</th></tr>
<tr><td>Uses questionnaires</td><td>Questionnaires
Data Protection</td><td>AG</td></tr>
<tr><td>Uses people e.g. to evaluate software</td><td>People
Data Protection</td><td>AG</td></tr>
<tr><td>Personal data</td><td>Data Protection</td><td>AG</td></tr>
</table></td><td>☐The supervisor approves the project under delegated authority of the Ethics Committee</td></tr>
<tr><td>3. This project has potentially significant or novel ethical issues. For example, it involves vulnerable beings, it uses deception, may cause potential harm (physical or mental), or may generate controversy because of its domain, methods or product</td><td>Explicit approval will be sought from the Ethics Committee.</td><td>☐The supervisor requires this project to be submitted to the Ethics Committee</td></tr>
<tr><td>4. This project is unsuitable due to ethical reasons</td><td>Revise or abandon this project</td><td>☐The supervisor does not support this project</td></tr>
</table>

I agree with the above classification and will take the appropriate actions

Student name: **Date:**

Student signature:

Supervisor name: **Date:**

Supervisor signature: