Cardinal Consulting

As requested by Animikii Indigenous Technologies:

YikeSite 2.0

1 Project Summary

1.1 Client Company Background

Animkii is an Indigenous Technology organization based out of the traditional territory of the Lekwungen (Songhees Peoples) of the Coast Salish Nation in Victoria, BC. Animikii has been offering services in web/software development, graphic design, and digital marketing to the Indigenous communities in Canada and the United States since its inception in 2003. The Animikii team has expanded over the past few years from 4 to 11 members including 5 software developers working on-site and remotely. The team is continuing to expand as they tap into more markets North America wide.

1.2 Project Purpose

Animkii has requested an analysis, and proposal for a potential solution, for their content management system, YikeSite. YikeSite has been used for all of Animikii's website development projects to date and is currently in use by many of Animikii's clients for keeping their respective websites up to date. The YikeSite CMS is also utilized by licensees outside of the organization and the licensee's respective client bases.

The following requirements document outlines the requirements and specifications needed for the new YikeSite system to be utilized by Animikii Indigenous Technology. All further mentions of the new system will be referred to as YikeSite 2.0. YikeSite 2.0 will aim to retain the features useful to clients and licensees while streamlining the process of updating system info and removing bottlenecks for the development team.

2 Team Members and Roles

2.1 Ceilidh Torrance

Position: Team Leader/Coordinator
Contact: ceilidhtorrance@gmail.com

Has worked with group members to create the RFP and Project Charter documents. Assisted in creating both presentations as well as presenting a section for the Charter presentation. Recently has been working with team to create the Requirements Document.

2.2 Darian Sampare

Position: Client Contact/Developer
Contact: dsampare04@gmail.com

Established the group's client to be Animikii and has been in contact with the company for the duration of the project by conducting interviews and gathering feedback on progress. Answered audience questions and presented slides in various presentations. Worked extensively on all milestone documents.

2.3 Gulshan Lalari

Position: Customer Relations
Contact: gulshanlalari@gmail.com

Assisting team members in completion of Request for Proposal and Project Charter documents and well as the construction of the in-class presentations. Presenting speaker for a portion of Project Charter presentation and collaborating with team on the current Requirements Document.

2.4 Hannah Bishop

Position: Back End Developer
Contact: hannahbananabishop@gmail.com

Work so far has included pitching the client to the class, creating the slides for the analyst charter pitch, working with the team to create RFP and Charter documents, and assisting team direction by creating skeleton documents.

2.5 Jacob Jones

Position: Researcher
Contact: jacobjones3877@gmail.com

Project work has included working on RFP and project charter documents as well as research on other possible market solutions. Working with the team on presentations and providing feedback on requirements have also been included in the workload.

2.6 Ryan Woodward

Position: Front End Developer
Contact: rwoodward84@gmail.com

Contributed heavily to pitches by creating informative slides and presenting to the class. Participated in writing all project deliverables including the RFP, project charter, and the following requirements document.

3 System Features

3.1 Client Editability of Website Content

3.1.1 Overview and Priority

The client should be able to edit their website's content after the construction of their website has been completed by the development team. This feature should allow the client to easily navigate their website to locate and edit all content within the website without having to access the code-base.

Priority: High

3.1.2 Functional Requirements

REQ-1.1: None

3.1.3 Non-Functional Requirements

REQ-2.1: The client must be able to edit their website after it has been completed and launched by the development team.

Rationale: Animikii does not have the resources to make changes to a client's website every time the client has an update to make. Therefore, the client must be able to independently maintain and update their site after some brief training period hosted by Animikii.

REQ-2.2: The content management system must be laid out in a way that flows easily and makes intuitive sense to a user in terms of where to find their data.
Rationale: Clients need a simple way to find data which needs to be updated or changed in order to make those changes.

REQ-2.3: Client cannot have access to the code-base in order to maintain system integrity.
Rationale: Giving clients access to the code-base could result in changes that cause a site to crash or appear wrong. Allowing a client to make changes to their site without giving them access to the code-base allows for safe and controlled editing of the website.

3.2 Client Editability of Website Layout and Design

3.2.1 Overview and Priority

The client should be able to edit the layout of their website in regards to page structure, menu navigation, and layout of individual components on each page. Furthermore, the client should have the ability to dictate design for some design aspects for their site. The scope of design aspects available to the client must be able to be dictated by the development team on a project-to-project basis.

Priority: High

3.2.2 Functional Requirements

REQ-3.1: It will take a client less time to manage their website using YikeSite 2.0.
Rationale: Clients currently complain about the amount of time it takes them to update and manage their time. Maintenance should be made simpler so that clients can focus on other aspects of their business.

3.2.3 Non-Functional Requirements

REQ-4.1: The client will be able to edit the layout of their website.
Rationale: Users should be able to change the layout of their website to reflect company changes. This may include new logos or colour schemes, modernization to keep up with changing times, or the addition or removal of data fields as displayed information changes.

REQ-4.2: Developers will be able to change the specific design aspects available to a client based on the specific project.

Rationale: Developers should be able to specify which aspects of design are available to a specific client based on the complexity of the design and technical knowledge of a maintenance team in order to avoid site errors.

3.3 Creation of Reusable Components

3.3.1 Overview and Priority

The development team must be able to produce components for client websites that can be carried over to other sections of the site and furthermore to other projects entirely. This feature will allow the developer to create custom-coded website components through a medium of the system that allows the components to be used over again.

Priority: High

3.3.2 Functional Requirements

REQ-5.1: It must take developers less time to finish a project using YikeSite 2.0.
Rationale: Developers have timelines to finish projects which are often exceeded. The creation of YikeSite 2.0 should cut back on the amount of time it takes to finish a client site.

REQ-5.2: YikeSite 2.0 must reduce a projects overall expenses.
Rationale: Clients have budgets for projects which are often exceeded by projects being extended or having more difficult implementations than previously expected. Reducing the cost of a project will increase developer and customer satisfaction.

3.3.3 Non-Functional Requirements

REQ-6.1: Developers will be able to create components for a client's website that can be used again for another website.
Rationale: Allowing a developer to reuse pieces of other sites when building a new website saves time and money.

3.4 Content Uploading Before Site Development Completion

3.4.1 Overview and Priority

The non-technical members of the Animikii team must be able to upload content for the client's website before the development team has completed the project. This feature must allow members of Animkii's team to upload client website content to the site without interacting with the codebase so it is available for all members of the team to access throughout the project.

Priority: High

3.4.2 Functional Requirements

REQ-7.1: YikeSite 2.0 must allow content to be uploaded in modules that will then be merged upon website completion.
Rationale: Allowing non-technical members of the Animikii team to upload content in a modular fashion will allow version control of the website.

REQ-7.2: Developers must be able to test uploaded content to ensure it fits within current codebase guidelines
Rationale: Ensuring quality control will ensure that any uploaded content by non-technical staff will be able to be implemented.

REQ-7.3: Developers must be able to perform integration tests to ensure non-technical uploaded content will merge with existing build.
Rationale: Quality control must be ensured to allow uploaded content to be merged successfully without errors.

3.4.3 Non-Functional Requirements

REQ-8.1: Uploaded content from non-technical staff will backed up and secure
Rationale: Any content creation by non-technical staff must be able to be stored in the case of a system failure.

3.5 Website Domain and Hosting Establishment

3.5.1 Overview and Priority

Developers, non-technical team members, and clients must be able to dictate the domain to be used by any website. The hosting for websites on YikeSite 2.0 must be universally managed, this should not be accessible by the client.

Priority: High

3.5.2 Functional Requirements

REQ-9.1: Highest priority domain name suggested by clients will be tested to determine uniqueness and whether it can be secured.
Rationale: Domain names needed by client must be tested to determine whether or not they can actually be used.

REQ-9.2: Unique domain name will be secured and paired with a web host.
Rationale: Once a domain name has been secured for a client it needs to be paired with a web host.

3.5.3 Non-Functional Requirements

REQ-10.1: Clients will submit a list of acceptable domain names that they wish to be constrained to.
Rationale: Clients need to be able to choose a domain name that will represent their business in a positive way. To help alleviate potential collisions with existing domains a prioritized list is necessary.

REQ-10.2: Client web hosting information will be kept private and secure. Only developers will have access to it.
Rationale: It isn't necessary for non-technical stakeholders to have access to the web hosting information. As such, this information will only be available to developers.

3.6 User Access

3.6.1 Overview and Priority

An individual site should be accessible to the associated client, and any Animikii staff responsible for its creation. Individuals with administrative access to the site must have special privileges. User data must be encrypted securely. Users of the system can be created and deleted.

3.6.2 Functional Requirements

REQ-11.1: Individuals with access to a site on the system should be given different privileges based on their access level. Users may be of type 'normal' or 'administrator'.
Rationale: This prevents inexperienced or malicious users from creating destructive changes to any given site.

REQ-11.3: 'Normal' type users may create, modify, or delete any existing content, layout details, or pages.
Rationale: This is an appropriate level of access for any generic client or Animikii worker.

REQ-11.2: 'Administrative' type users should include 'normal' user privileges, as well as the ability to update hosting details, create or delete users of any type, and delete the site.
Rationale: These actions require a higher amount of responsibility, and should therefore only be available to a small subset of users.

REQ-11.3: User data should be stored in a database with encrypted passwords.

Rationale: Encrypting passwords rather than storing plaintext strings is best practice, and prevents compromising users' data (especially individuals with similar passwords for multiple systems).

3.6.3 Non-Functional Requirements

REQ-12.1: None.

3.7 Resources

3.7.1 Overview and Priority

Animikii has limited resources that it can dedicate to the implementation of YikeSite 2.0. The project has a maximum budget set at $100,000 and a time limit of six months. During this time Animikii feels that they can spare 3-4 developers to implement the solution assuming that the project does not distract from development.

3.7.2 Functional Requirements

REQ-13.1: The implementation of YikeSite 2.0 cannot cost more than $100,000.
Rationale: The maximum budget for YikeSite 2.0 has been set by Animikii. The budget was decided based on the value of the project and the projected increase in business brought about by the overhaul.

REQ-13.2: The creation of YikeSite 2.0 cannot take more than six months.
Rationale: The client has limited development to six months as this is the amount of time they feel they can spare for the project.

REQ-13.3: The overhaul of YikeSite 2.0 cannot require more than 4 developers over the six month period.
Rationale: Over the six month implementation period for YikeSite 2.0, Animikii feels that they can spare up to 4 developers from other projects in order to implement the proposed solution.

3.7.3 Non-Functional Requirements

REQ-14.1: During the implementation of YikeSite 2.0 Animikii's focus must remain on serving their clients.
Rationale: Animikii cannot halt other operations, in order to update their content management system otherwise they will lose clients. For the same reasons, they cannot lose focus on the clients that they currently have, or new ones that they are trying to bring in.

3.8 Documentation

The type of content management system that will be implemented is partially constricted by when the data is processed. Content Management Systems on the market today build content into a public webpage though offline or online systems. The use of offline processing is for websites that can be preprocessed and then launched; launching a site using a static site generator can be time consuming while missing out on many features offered in other processing systems like accounts and real time maintenance. An online model makes more features available by using a web cache to retrieve data or by building html when a user visits the page. This model allows for more complexity and functionality of a website allowing for some of the previously mentioned features. A developer can also mix the two systems together to work for them depending on how they would like to structure their workflow. Using the right mix of offline and online processing can help with streamlining some of the bottlenecks faced in the current yikesite.

4 Alternative Solutions

The main alternative solution explored was to replace YikeSite with an existing modern CMS framework. Possible options included Wix, Squarespace, and Drupal. This suggestion was noted several times in class and was worth looking into. Certainly, an existing framework would provide fantastic features, documentation, and user experience to clients and developers. However, this approach had a significant drawback. There are over 200 existing YikeSite sites and a large number of licensees who extend YikeSite to their own clients. These clients provide word of mouth marketing for YikeSite, and provide Animikii with a stable source of revenue. Transitioning Animikii's workflow to include an external CMS would involve either transitioning those sites and licensees to use the CMS, or maintaining YikeSite as a legacy service and creating all new sites with a new framework. Disrupting 200+ clients wasn't a viable option, and moving away from YikeSite on future projects would mean losing YikeSite client referrals and competing for clients in a larger market. Ultimately, the decision to go with the current approach came from conducting stakeholder modeling. We itemized the project stakeholders and weighed their respective influence. It was decided that continuing to support the needs of existing clients/licensees and maintaining what has worked well for the company for many years justified the time and expense of modernizing YikeSite rather than switching to a different CMS.

5 Risks and Limitations

5.1 Risks

5.1.1 Unforeseen development delays / Exceeding Budget

The development team will rely heavily on the involvement of the original creator of the YikeSite CMS and any unforeseen absence could have an effect on the implementation schedule. This in turn, could have an impact on the overall budget of the project if more time/resources are required to complete the project.

5.1.2 Developer dissatisfaction with the new system

Bringing greater product value to Clients has been identified as a priority in Yikesite 2.0, which could result in the neglect of meeting developer needs within the new system.

5.1.3 Client dissatisfaction with the new system

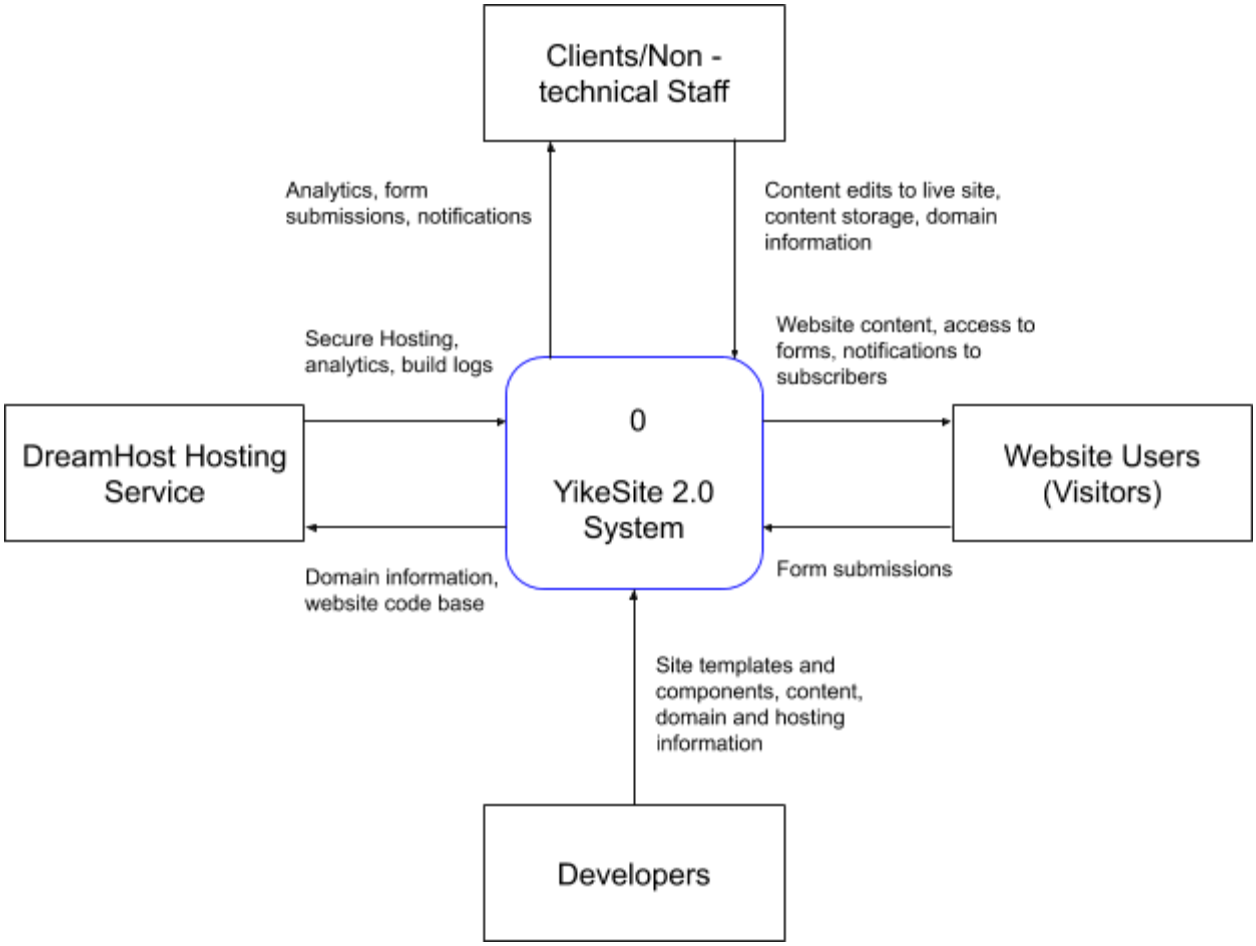
Although the main focus of YikeSite 2.0 is to add more functionality and create a better user experience for Clients, there is always a risk that the Client may not want to adapt to changes or have to learn a new system.

5.2 Limitations

As noted in REQ-13 the main constraints with the development of Yikesite 2.0 will be the budget, timeline and availability of knowledgeable developers for the 6-Month development period. The proposed functional requirements (REQ-13.1-3) should provide ample time and resources for developers to complete an overhaul of the existing system in the proposed time frame, given that the developers have a strong understanding of the current codebase. Knowing that Animikii only has one developer with extensive knowledge of the current codebase, the development team will rely heavily upon this individual during the development process which could cause problems in efficiency. Having a larger team to develop YikeSite2.0 will help to mitigate limitations in future iterations of the YikeSite CMS thanks to the number of new, more knowledgeable developers at Animikii.

6 Appendix

6.1 Context Diagram



7 Release Plan

Work Item	Description	Delivery Date
Project Charter	Project overview and plan	Oct. 15
Proposed Solution Meeting	Discuss proposed solution feasibility with Animikii	Oct. 18
Prototype Delivery	Presenting prototype to the client	Nov. 5
Prototype Feedback Meeting	Gathering in-depth feedback from Animikii on the delivered prototype and accepting any change requests	Nov. 5

Requirements Analysis	Requirements analysis section of the final report to be completed and submitted for feedback	Nov. 10
Final Project Presentation	Presentation of our solution findings	Nov. 19
Final Report	Detailed report of our proposed solution	Dec. 4