Oregon State University
College of Engineering

CS Capstone Requirements Document

November 14, 2017

ConnectBasket

Prepared for

OSU MIME

Dr. Chinweike Eseonu

Prepared by

Group 39

ConnectBasket Development Team

Kailyn Hellwege

Abstract

This document describes the three main parts of the ConnectBasket project that Kailyn Hellwege will be responsible for and compares and contrasts different technologies that can be used to implement those parts of the project.

CONTENTS

1 Content Delivery Networks

1.1 Overview

A content delivery network, or CDN, is a group of servers that are distributed geographically that work together to quickly deliver Internet content, such as images, videos, and HTML pages [1]. Currently, most web traffic is served through CDNs, and CDN's can also help protect against some malicious attacks [1].There are four main reasons to use a CDN: to improve website load times, to reduce bandwidth costs, to increase content availability, and to improve website security [1].

1.2 Criteria

For this project, security is an important factor to consider. Since the ConnectBasket website will likely contain patient and owner information, it is important that the website is secure. Another important factor to consider for the Connect-Basket project is price. We want the cost to stay as low as possible, and so choosing a free or low cost CDN is important to the project.

1.3 Potential Choices

1.3.1 Cloudflare

The CDN that Cloudflare provides is designed to optimize security and performance, has multiple price levels, and is easy to set up [2]. Cloudflare's CDN provides a flat price per month that does not depend on bandwidth [2]. There are four pricing levels [2]. The free version is for personal websites [2]. The Pro version is twenty dollars per month for professional websites [2]. The Business version is two hundred dollars per month for websites and businesses requiring advanced security and performance[2]. Finally, the Enterprise version is for companies requiring enterprise-grade security and performance, and Cloudflare muse be contacted to determine a monthly price [2]. In addition, setting up Cloudflare is easy, taking less than five minutes to set up a domain.

1.3.2 Akamai

Akamai serves thirty percent of Internet traffic, and it optimizes networks for content delivery [3]. Akamai's CDN has three components [3]. The Aura Edge Exchange allows operators to deliver video content to customers [3]. The Aura Control System is a set of tools that allows operators to manage components such as performance and CDN security [3]. Lastly, Akamai Federeation allows operators to work with the Akamai Intelligent Platform, which is a cloud-computing platform [3].

1.3.3 Amazon CloudFront

The Amazon CloudFront CDN is built on the Amazon Web Service (AWS) infrastructure, and there are forty-four Availability zones in sixteen regions, and there are plans for further expansion [4]. This large network provides high performance for customers all over the world [4]. CloudFront is a highly secure CDN, and for no additional cost, all customers benefit from the protections of AWS Shield Standard [4]. Amazon CloudFront is easily integrated and works best with other AWS services, such as Amazon Simple Storage Service and Amazon API Gateway [4]. For pricing, there are no fixed fees or long term contracts; payment is only for the data transfer and requests used to deliver content to customers [4].

1.4 Discussion

Cloudflare and Amazon seem to have opposite pricing systems. Cloudflare has a fixed pricing system, where the amount of traffic per month does not affect how much is paid. In contrast, for Amazon CloudFront has a variable pricing system, where payment is only for the amount of data transfer and requests. Akamai does not provide pricing information, and so a price comparison is not possible. All three have a high level of security, and so they would all be good choices for that factor.

1.5 Conclusion

The best CDN for this project is Cloudflare because it has a fixed cost, which means there will not be any spikes in cost in any given month. Amazon CloudFront would be a good CDN, especially since it integrates well with other AWS services, but it seems to be the best choice only if other AWS services are being used, and we are not using other AWS services.

2 Web Development Frameworks

2.1 Overview

A web development framework is considered to be tools and resources used by software developers to create and manage websites [**?**]. Web development frameworks extend the capabilities of a language and provide libraries so developers do not have to start from scratch and hand-code everything [5]. A framework may include libraries, APIs, security, and compilers, among other things [5].

2.2 Criteria

One important criteria for this project for web development frameworks is that it needs to be easily scalable to many different devices and sizes of devices. This is important because the ConnectBasket website will need to be accessible from staff's desktop or laptop computers as well as a variety of smartphone devices. Another key factor is it needs to be simple as opposed to cluttered with a large number of features that will likely go unused. This project is relatively simple, and so tons of features are not necessary. Additionally, a framework that is flexible would be best.

2.3 Potential Choices

2.3.1 ASP.NET

ASP.NET was created by Microsoft, and it is used to create interactive and data-driven web applications[6]. ASP.NET works with HTML to create dynamic web pages [7]. One advantage of ASP.NET is that it provides built-in Windows authentication, so it is secure [7]. In addition, an ASP.NET application can be written in a variety of languages, including C# and Visual Basic [6].

2.3.2 AngularJS

AngularJS is a JavaScript based web application framework that is open source and currently maintained by Google [8]. AngularJS uses HTML as a template and extends HTML syntax to describe a website or application's components easily [8]. AngularJS provides data binding capability to HTML, provides reusable components, and is unit testable, among many other features [8]. In addition, AngularJS is available on all major browsers and smartphones [8]. One disadvantage of AngularJS is that it is not secure, and so server side authentication is needed to keep an application secure [8].

2.3.3 ReactJS

ReactJS is a JavaScript library that makes it easy to build user interfaces by creating reusable UI components[9]. ReactJS involves thinking about everything as a component [9]. Advantages of ReactJS include that it can be used with other frameworks, and can be used on the client side or the server side [9]. One major disadvantage of ReactJS is that other tools will need to be combined with React to get a complete set of tools that is required for development [9].

2.4 Discussion

AngularJS and ReactJS are both JavaScript based, while ASP.NET is very flexible because it can be written in many languages, including C# and Visual Basic. Angular JS is useful because it is available on all major browsers and smartphones, but a downside is that it is not secure. In contrast, ASP.NET is secure and provides built-in Windows authentication. ReactJS would be good for creating reusable components, but it would possibly require additional tools for successful development. AngularJS also can easily create reusable components.

2.5 Conclusion

ASP.NET is the best framework for the ConnectBasket website. It is flexible and can be written in multiple languages, and it has useful built-in Windows authentication.

3 Design Patterns

3.1 Overview

Software design patterns are a very important part of any software application. One very important feature of this project is to provide a user interface that will be accepted by users who are reluctant to change. This means how users are able to interact with the data is very important and modeling a system that closely reflects their current system in a more efficient way is critical.

3.2 Criteria

Major factors to consider when choosing a software design pattern for this project is how well will the system be able to model data in a way that our system needs it to. We need to have users inputting data into the system which need to be able to display that information to other users. There are needs for data to be filtered for specific users as well as displayed as an overall summary for some users.

3.3 Potential Choices

3.3.1 MVC

MVC is a software design pattern for implementing a user interface [10]. It separates an application into three parts known as the model, view, and controller [10]. The model is the structure of the data in the application and how the data is represented in the application [10]. View is the component that makes the data displayable to the user in a useful way [10]. The controller supplies an interface between the model and view components, taking the data from the model and converting it into something that can be used by the view component to display to the user [10]. MVC supports the development of multiple different views for one model so the same information can be displayed in multiple ways if that is important for the application [10]. It is also a very fast development process because one person or team can develop the view component while another works on the controller and another works on the model [10]. This design pattern requires multiple developer for it to work properly [10].

3.3.2 MVP

MVP is a software design pattern that is based on similar concepts to the MVC design pattern [11]. It separates an application into four different components that are responsible for determining how a user can interact with the system [11]. Those components are the view, view interface, presenter, and model [11]. The view is responsible for determining how the user will see the information displayed while the view interface connects the view to the presenter [11]. The presenter connects the view to the model, which is responsible for the data that is displayed by the view [11]. A view will usually only be connected to one presenter [11]. Testing can be done easily with this design pattern because all interactions are done through an interface [11]. Development is speed up by the fact that multiple programmers can split up the components and work on them separately [11].

3.3.3 MVVM

MVVM is a software design patter that is an extension of the MVC design pattern [12]. MVVM has four components that are Model, View, Controller, and View Model [12]. The MVVM pattern has view models convert data from the model layer into something that can be usable by the view layer which keeps the code for controllers from growing too large [12]. In MVVM, the controller no longer depends on the model so it is much easier to test. Views are only used to present the data they are given [12]. The view controller interacts with both the view model and view layers and the view model also interacts with the model [12].

3.4 Discussion

MVVM and MVP are both similar to MVC, and provide some additional features compared to MVC. In contrast to the four components of MVVM and MVP, MVC only has three components which can make it a simpler and easier to use pattern. MVVM and MVP have an additional components which further modularizes the functionality of each component and makes it easier to test.

3.5 Conclusion

The best design pattern to use for this project is MVC. The user interface for this project will not be very complicated, so, while having additional components might improve testability, it will not be worth it for the increased complexity. Having three components also provides the option for dividing the components between the three person development team, with one team member working on each component.

REFERENCES

[1] "What is a cdn?." https://www.cloudflare.com/learning/cdn/what-is-a-cdn/.

[2] "Fast, global content delivery network." https://www.cloudflare.com/cdn/.

[3] "Cdn services." https://www.akamai.com/us/en/resources/cdn-services.jsp.

[4] "Amazon cloudfront." https://aws.amazon.com/cloudfront/.

[5] C. Wodehouse, "What is a framework?." https://www.upwork.com/hiring/development/understanding-software-frameworks/.

[6] "Asp.net - introduction." https://www.tutorialspoint.com/asp.net/asp.net_introduction.htm.

[7] A. Hasan, "Top 12 main advantages of asp.net framework." http://www.arpatech.com/blog/top-advantages-of-asp-net-framework/, 3 2017.

[8] "Angularjs - overview." https://www.tutorialspoint.com/angularjs/angularjs_overview.htm.

[9] "React." https://www.tutorialspoint.com/reactjs/reactjs_overview.htm.

[10] Jithin, "What is mvc? advantages and disadvantages of mv." https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/, 10 2016.

[11] T. Snyder, "Mvc or mvp pattern whats the difference?." https://www.infragistics.com/community/blogs/todd_snyder/archive/2007/10/17/mvc-or-mvp-pattern-whats-the-difference.aspx, 10 2007.

[12] B. Jacobs, "What are the benefits of model-view-viewmodel." https://cocoacasts.com/what-are-the-benefits-of-model-view-viewmodel/, 11 2016.