Oregon State University
College of Engineering

CS Capstone Technology Review

November 20, 2017

ConnectBasket

Prepared for

OSU MIME

Dr. Chinweike Eseonu

Prepared by

Group 39
ConnectBasket Development Team

Taylor Kirkpatrick

Abstract

This document describes in detail three parts of the ConnectBasket software that will be developed by the team during the 2017-2018 Capstone year. These parts are the user login and profiles framework, the issue tracker framework, and the SMS service. For each part in this document, three technologies are compared to find the best choice for the project. These technologies are the HUGE framework, Request Tracker, and manual emailing respectively.

CONTENTS

1 Introduction

1.1 Purpose

The purpose of this document is to give an overview of what parts of the ConnectBasket project Taylor Kirkpatrick will have responsibility over, and analyze the technologies that may be used for each part. Each part will choose a specific technology that may be used in the final project, along with justification for why.

1.2 Scope

ConnectBasket will be a web based portal for the OSU Veterinary Hospital that will streamline the way messages are moved around the hospital and provide a better way to track the route that messages have taken. When an owner calls the hospital, receptionists will be able to type a message into the computer, assign it a category, and route it to the necessary staff members, which may be service specific vet techs, house officers, or faculty. Those staff members will be alerted, either by email or text, that they have a message waiting for them in ConnectBasket, and once viewed, notes can be added and the message can be re-routed to other staff members. After a message has been addressed by the necessary staff, the message can be closed preventing further notes or comments. The audit trail of when the message was created and closed will be visible, along with all the notes, timestamps of notes, and staff who added the notes.

1.3 Definitions, Acronyms, and Abbreviations

- Capstone - CS 461 class at Oregon State that the developers of ConnectBasket are in
- OSU - Oregon State University
- Hospital - referring to the OSU Veterinary Hospital
- Owner - a person who owns a pet that is a client of the hospital
- Patient - an animal who has/is going to be treated at the hospital
- VetHosp - the hospital management system used by the OSU Veterinary Hospital for keeping patient and owner records
- Mobile Devices - smartphones or tablets
- CPaaS - Communications Platform as a Service

1.4 References

References

[1] ”Userfrosting — Modern user management framework for PHP”, *Userfrosting.com*, 2017. [Online]. Available: https://www.userfrosting.com. [Accessed: 14- Nov- 2017].

[2] panique and others, HUGE, (2015), GitHub repository, https://github.com/panique/huge

[3] Microsoft, ”Introduction to ASP.NET Identity”, *Docs.microsoft.com*, 2017. [Online]. Available: https://docs.microsoft.com/en-us/aspnet/identity/overview/getting-started/introduction-to-aspnet-identity. [Accessed: 14- Nov- 2017].

[4] Trello, ”Trello”, *Trello.com*, 2017. [Online]. Available: https://trello.com/. [Accessed: 14- Nov- 2017].

[5] Best Practical, ”Request Tracker”, *Best Practical Solutions*, 2017. [Online]. Available: https://bestpractical.com/request-tracker. [Accessed: 14- Nov- 2017].

[6] Twilio, ”Communication APIs for SMS, Voice, Video and Authentication”, *Twilio.com*, 2017. [Online]. Available: https://www.twilio.com. [Accessed: 14- Nov- 2017].

[7] Nexmo, ”Nexmo - APIs for SMS, Voice and Phone Verifications”, *Nexmo.com*, 2017. [Online]. Available: https://www.nexmo.com/. [Accessed: 14- Nov- 2017].

[8] M. Grech, ”Twilio vs. Nexmo: The Head to Head Showdown for Developers — GetVoIP”, *Getvoip*, 2017. [Online]. Available: https://getvoip.com/blog/2017/01/05/twilio-vs-nexmo/. [Accessed: 14- Nov- 2017].

1.5 Overview

The rest of the document will contain three sections with an overview of the parts of the ConnectBasket project Taylor Kirkpatrick is responsible for. Each section will contain five subsections, three to analyze specific technologies, one to compare them, and one to state the chosen technology with project-specific reasoning.

2 User Logins/Profiles

A feature highly anticipated by the client, users must have persistent, personalized states to distinguish them from other users when tasks are to be passed on. These user profiles must be private and secure, one to each user. Messages may be user-specific and carry information that is confidential or otherwise not to be made accessible to everyone, even within the company. This part specifies what framework or project will be built off of to create these users and their logins and profiles. Of note is the possible overlap this system may have with the issue tracking technology, as those are usually distributed as whole projects, including user logins. While we do expect to be using one of these frameworks, it may clash with what the issue tracker framework brings to the table.

2.1 Userfrosting

Userfrosting is a complete and easy to use framework for PHP. It depends on the PHP tool Composer and also makes use of Node.js and Bower for client-side packages. The framework fully automates and handles persistent user sessions without requiring the dev team to handle sessions manually between many potentially dynamically updating pages. Userfrosting also adds easily customizable and rather nice looking user profile pages, allowing a dev to simply present some information to users under the conditions that they desire. In addition, this framework provides simple methods for a dev to create user roles for role based control, and assign users to them. Lastly, the framework allows the creation of a user administration page for user operations. This interface can be easily built for non-technical users, and ease the process of user administration for them.[1]

2.2 HUGE

HUGE is a PHP framework for doing simple user authentication. HUGE came about in a time when PHP was lacking many quality login solutions, and was built primarily off of the password_compat library once it was released. HUGE is no longer in active development, but has remained an excellent choice when facing up against many other PHP frameworks. HUGE makes use of the official PHP password hashing functions, with many other nice features that one would expect in a login framework like email account verification, form encryption, a ”remember me” function, etc. HUGE also offers many other nice features like URL rewriting (masking possibly sensitive info), and supports user portraits already. HUGE requires PHP 5.5 or above, a MySQL database, a handful of PHP extentions, a few developer tools, and software for sending mail if desired.[2]

2.3 ASP.NET Identity

ASP.NET includes a relatively robust user login API for use in ASP.NET projects. As this project may be done in ASP.NET, this is something worth checking. As the web became wider and demanded even better user management support, ASP.NET created the framework ASP.NET Identity to keep up with alternatives, offering full support with other ASP.NET packages along with the expected features in a login framework. Playing well with other ASP.NET packages is not to be underestimated, however, as a project built in ASP.NET is sure to have numerous other packages. The framework does, of course, support simple login features, along with more advanced things like social media integration and a persistent ”remember me” cookie login, and role based control.[3]

2.4 Comparison

At a first glance, it appears that ASP.NET Identity lacks features compared to the other two options, and this observation is correct. However, it does boast the reliability of Microsoft’s products, and development with it in Visual Studio would be easier than the other two. Between the other two, Userfrosting is much more hands-off than HUGE, and while doing less complex and error prone work is nice, it also restricts the freedom of the developers. HUGE and Userfrosting have very similar features available, but HUGE may even boast more features. However, HUGE was never really formally released as anything other than a simple open source framework. Identity and Userfrosting are both more formal projects, and may not be as bug prone.

2.5 Selection

The HUGE framework is the tentative winner in this category. It is a happy medium between the hands-off GUIs of Userfrosting and the mostly manual configuration of ASP.NET Identity. HUGE has roughly the same number of features as the other two combined, is much better documented than both, and has many years of fellow users for support that Identity may not have, and Userfrosting is unlikely to have. This victory comes with a caveat, however, as it is dependent on what language the software is in. Should the dev team be unable to use PHP, or otherwise choose not to in favor of ASP.NET or something else, HUGE would be useless for the project. Of the three present here, however, it is the better of them, while the others provide options should other development decisions invalidate HUGE as an option.

3 Issue Tracker

The primary goal of this project is, in essence, to create a create a custom issue tracker, personalized for the veterinary hospital. The bulk of the requested features for this project are in tune with making a non-IT bug or feature tracker. The features requested are both standard issue tracker features and more specialized items as well. Thus, finding a framework or at least a simple, freely licensed issue tracker to build off of would simplify the development of the bulk of the features and reduce room for error.

3.1 Trello

Trello is a popular kanban style scrum team management application. It is currently owned and maintained by Atlassian, who also own the similar software JIRA, which is more technical and is not free. Trello allows easy ”whiteboard style” organization for tasks, and can easily be applied to non-technical uses. Trello also has a robust API available to the public, and is what the developers use for further work on the product, proving the power of the API. Trello’s interface is easy to read and navigate, and interaction is simple and intuitive. Trello also works with mobile devices, supporting most modern mobile browsers as well as desktop.[4]

3.2 Request Tracker

Request Tracker, made by Best Practical, is an issue tracker designed for general use, instead of technical users. Request tracker has a very clean interface, with simple messaging and includes user login support by default. Request Tracker is an entirely open source project, and the community and developers seem to be quick to fix discovered bugs or issues. Request Tracker boasts email integration and allows users to interact fully with the system through email only, not even needing to open the application to resolve or interact with tickets.[5]

3.3 Manual Issue Tracking

This option involves creating a bare-bones, incredibly simple request tracker in-house. This option would have exactly the features desired and would integrate the easiest with the rest of the project, with the caveat of taking a significant amount of time. However, when parts of the project are still unknown, this is guaranteed to be the most fluid option, as the development team can change or customize anything they want at any time.

3.4 Comparison

Creating the request tracker manually is a very valid option, as a major problem some of the other issue trackers may have is the extent of the features they have that are mostly unneeded. Trello, while probably the most reliable and one of the easier options is especially guilty of this, as users would hardly benefit from the visual style of a board, and tickets being open and unsecured may prove to be problematic. Request Tracker is a good middle ground, as it is not bloated with unnecessary features, but does not require a custom built system.

3.5 Selection

Request Tracker would be the ideal here. It contains almost exactly the features the client is looking for, and has bells and whistles that the users might appreciate without being overwhelmed by them. Trello is much too bulky, and the kanban style is not very applicable in this case. If Request Tracker can be integrated with what the client wishes, or built off of into a new application, this would give the users the most functionality while reducing the risk of human error and more bugs that would be inherent in manually creating the issue tracking.

4 Notifications

The client has specifically requested smartphone notifications when notable activity takes place related to a specific user. Users should be able to be alerted to any activity related to their user profile or role on their smartphone, should they so wish. The client has not specified how this notification is to take place, and so could be up to which technology is chosen.

The initial idea for notifications, though alternatives were considered, was using SMS, as it does not depend on wifi connection, and anyone with a functioning phone would have the ability to receive such a notification, even without a smart phone. Push notifications, while pretty and useful, could be impractical or impossible as the ConnectBasket software will be a web application rather than a native mobile application.

4.1 Twilio

The Twillio tool allows an application to send SMS messages regardless of provider to a user, among other phone related features such as programmable calls and call redundancy allowing for more reliable calls and messaging. Twilio has a nice looking interface and easy to understand setup, and offers unique features like voice call support and robust message quality of life improvements like sticky sender and MMS support. Pricing for Twilio varies, and while they offer free API keys for limited use, many features even for small uses are not free.[6]

4.2 Nexmo

Nexmo, recently acquired by Vonage, boasts all of the features one could expect from a CPaaS tool. It allows a user to send messages through most providers and includes useful functions in the API for number screening, allowing a service to catch spam or malicious numbers before devoting resources to them. Nexmo also offers a real-time chat function to interact with users should such a situation be desired. Nexmo benefits from the resources of their parent company, able to operate in 80 countries, supporting many popular chat systems, and offers free inbound texts. Pricing is on a per-text basis, but does not contain a free trial for most features.[7]

4.3 Manual Email

This option forgoes the external tools and the use of SMS entirely. Most email providers allow for drafting and sending off a simple email to a target address. While this option would not directly solve the issue of mobile notifications, it could be used to take advantage of an existing feature most smartphones have, email notifications. Notifications could be delivered by the user’s email application, giving the user more control over when or what notifications they would like to see, and also should be free.

4.4 Comparison

Without the option of push notifications, which would be ideal, pricing and immediacy are the most important things. Twilio and Nexmo are rivals in the CPaaS industry, and have similar pricing plans that compete with each other. However, neither of them are free for the purposes of this project, as both lock messaging tools behind a ”pay per text” plan. Emailing is free and may likely even be easier than the other two to use, but lacks the immediacy that a SMS provides and also may not be received if the user lacks a wifi connection.[8]

4.5 Selection

Manual Emailing is probably the best choice here. While it is possible that the client or a member of the college would be willing to pay for either Twillio or Nexmo, the features they offer are overkill for this project, and these notifications may not even be used by many of the users. Those that do, however, would likely not appreciate texts every time there is any activity on a ticket.