Oregon State University
College of Engineering

CS Capstone Design Document

November 30, 2017

ConnectBasket

Prepared for

OSU MIME

Dr. Chinweike Eseonu

Prepared by

Group 39

ConnectBasket Development Team

Henry Fowler
Kailyn Hellwege
Taylor Kirkpatrick

Abstract

This document provides a detailed software design description for the ConnectBasket project. It is intended to serve as a guide for the development team to implement the project, and will provide a brief overview of the project as well as specific details about individual components. Different viewpoints of the project will be discussed as well as stakeholders of the project. The design plan for each component of the project will be covered with timelines and testing plans for each component.

CONTENTS

1 Introduction

1.1 Purpose

The purpose of this document is to layout a design plan for the ConnectBasket project being completed by the Capstone development team. It will provide design viewpoints as well as identify stakeholders in the project. Each component's design will be explained in detail, and a timeline and testing information will also be provided.

1.2 Scope

The ConnectBasket project is intended to be a stand-alone solution to help the Oregon State Veterinary Hospital communicate information between employees and to patients. The hospital's current system is heavily paper based and error-prone, and this project will allow for a more efficient workflow.

Employees of the hospital will need to be able to enter notes about patients, route the note to other employees to take some action, and close the note as being done. A notification system will be implemented to let employees know when they have a note to look at. There will also be a way for privileged users to view a history of notes and actions taken. This project will help the hospital quickly respond to patient inquiries and either answer their questions or schedule an appointment.

1.3 Intended Audience

The intended audience for this document is the Capstone development team working on the project as well as the client for the project. It is a plan for how the Capstone development team will implement the project. In addition to that, it will provide the client with information about the design and timeline for the development of the project.

1.4 Stakeholders

The stakeholders in this project include the client and the employees of the hospital. The client is working with both the Capstone development team and the hospital to complete the project. He will not be directly involved in using the product once it is developed. The employees of the hospital will be the end users of the project and interact with the finished product as a part of their daily jobs.

1.5 References

1.6 Glossary of Terms

Capstone - CS 461 class at Oregon State that the developers of ConnectBasket are in

Capstone development team - The developers of ConnectBasket

ConnectBasket - A web application being created for the OSU Veterinary Hospital

Hospital - referring to the OSU Veterinary Hospital

Message - Initial information taken down from a phone call with an owner

Mobile Devices - smartphones or tablets

Note - Information added to a message by a user the message was routed to

OSU - Oregon State University

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

Route - Send a notification to a user of the system that a note is waiting for them to look at

VetHosp - the hospital management system used by the OSU Veterinary Hospital for keeping patient and owner records

2 Design Viewpoints

2.1 Functional Viewpoint

2.1.1 Functional Elements

There are many functional elements that will make up the ConnectBasket web application. The four main pages of the web application are message entry, add notes to messages, message viewing, and an audit trail of messages. The other functional element of the system is a database that stores the information collected through the application.

2.1.2 Responsibilities

Each element has its own responsibilities. The message entry element will allow users to add a message to the database and route that message to other users. Add notes to messages will allow users to add more information to a message that has already been created and either route to other users or close the message. Message viewing will allow users to see all messages that have been routed to them and see all of the previous notes that are associated with that message. The audit trail of messages will provide certain users with the ability to look at a history of messages and their notes that were created during a given time period. The database will have to tables to store information about messages and their associated notes, and will have queries that return groups of messages that meet certain criteria.

2.1.3 Interfaces

Each element will have its own page in the application. The message entry and add notes to messages will be done through a web form, and the message viewing and audit trail of messages will be displayed through a table or grid. The database will be accessed through each page of the application.

2.1.4 Primary Interactions

Several of the pages will interact with each other. From the view messages page, a user will be able to go to the create message page or select a message and go to the add notes to messages page for that message. Adding messages and adding notes to messages will insert new information into the database, where message viewing and audit logs of messages will pull information out of the database.

2.2 Context Viewpoint

2.2.1 Relationships and Interactions

The ConnectBasket system has many relationships with its users. Users that work in reception and scheduling at the hospital, will need the system to add new messages as well as view messages and their associated notes to call owners and give them answers to their questions or schedule appointments. Vet techs and doctors will need to be able to view messages that have been routed to them and add notes to them. Administrators will need to be able to view the audit logs of messages.

There will also be several external systems that ConnectBasket will interact with. A user profile framework will be used for managing users of the application and keeping login information secure. ConnectBasket will also interact with

an email client in order to send notifications to users that a message is waiting for them. If proper access is provided to the development team, ConnectBasket will integrate with the VetHosp database to access patient and owner information.

2.2.2 Dependencies

The notification feature of ConnectBasket will depend on users having access to their email through a mobile device or desktop computer that they check frequently. The user profile system will depend on users keeping their own information up to date in order to receive notifications and see the proper messages.

2.3 Information Context

2.3.1 Information Storage

The information for ConnectBasket will be stored in a relational database. There will be tables that store information about messages, notes, owners, patients, and possibly users. The different pages of the web application will allow for the information to be managed and displayed.

2.3.2 Database Design

Put a diagram here

3 COMPONENTS

3.1 Server Setup

3.1.1 Details

This project will require having a server configured to host the web application ConnectBasket. The application will not be supporting heavy traffic and does not have any specific speed requirements so the server requirements are not very strict.

The server will need to work with the web server software and database to make the application accessible to the users. Users will interact with the server through the API created as a part of the ConnectBasket application.

3.1.2 Design Decisions

The server operating system that will be used for this project is Linux Ubuntu Server. It was chosen for its low cost, short time to set up, and well written documentation. The rest of the components of the project rely on the server being up and running so the decision was primarily based on the amount of time it will take to get set up.

3.1.3 Timeline

The setup of the server will be the first priority in the development of this project. For this reason, the completion of this component will be done by the third week of winter term.

3.1.4 Testing

There will be relatively little testing done for this particular component. If the web application is able to successfully be accessed, then this component will be considered successfully implemented.

3.2 Web Server

3.2.1 Details

In order for ConnectBasket to work, the server must have web server software running on it. This software will allow users to access the web application running on the server from any mobile device or desktop computer that has internet access.

3.2.2 Design Decisions

For this project, Apache HTTP will be the web server software that is used. It was chosen because of its cost, documentation, and flexibility. It is free software that can be used by the ConnectBasket development team for no cost. There is a large amount of well written documentation that will help during the setup period. Due to the short development window for the project, speed of implementation was a key factor in deciding which web server software to use. It is also a very flexible web server that can run on almost any hardware and with almost any server operating system.

3.2.3 Timeline

The web server software properly working is critical to the rest of the components being able to work and be tested. This component will be implemented as soon as the server component is completed. The web server software will be completely implemented by the end of week four of winter term.

3.2.4 Testing

The testing of the web server software will be done by attempting to access the ConnectBasket application after it has been partially or fully created. This component will be considered complete when the web application is able to be used from all required devices.

3.3 Database

3.3.1 Details

ConnectBasket will have a database that stores the information collected by the application. Notes that are created will need to be stored in tables as well as all actions that are taken with those notes. There will also need to be tables that store information about the patients of the hospital and their owners. If the information is provided to the Capstone development team, the patient tables will be filled with information from the current VetHosp system. Otherwise, the tables will be filled with fake data for testing purposes, and patient information will have to be manually added by a hospital employee through ConnectBasket the first time an owner calls after the system is being used.

The data gathered through the application will be used for many different purposes. First, the notes entered and actions taken will need to be immediately available through the web application for other users to view. Second, users will need to be able to see a chain of all actions that are associated with a certain note. Finally, a select group of users will be able to view an audit trail of all notes created and actions taken over a given time period.

3.3.2 Design Decisions

The database used for this project will be a MySQL database. It is important to have a relational database to store the data needed for the project. MySQL is also very flexible and can work well in almost any system. It is not the fastest database that could be used, but this application will not require any large amount of data manipulation, so the simple and easy to use interface provided by MySQL is the best choice.

3.3.3 Timeline

The database will be critical for testing many other parts of the application, but will also be a large part of the project. Initial setup of the database and creation of tables will start around week two or three of winter term and finish by week five. The finished version week five is still subject to change, if the need for different or more tables or queries is found during the development of the rest of the project.

3.3.4 Testing

The database will be tested in several ways. Queries will be run on their own to make sure that the correct data is being returned, in addition to using the web application to test and make sure data can be added and updated.

3.4 Mobile Notification System

3.4.1 Details

ConnectBasket should have the ability to notify users. Users will have unique profiles that can contain information about them, including their email. This email should be used, according to user specifications, to alert them on their mobile device when a message is received or some other event designated as important by the user occurs. ConnectBasket will rely on the user's email application on their mobile device to deliver alerts and notifications. The body of this email should contain a link to the relevant page appropriate to view the specific event. The email body should also contain identifiers for any other users involved in the activity, along with a short description of the activity.

3.4.2 Design Decisions

Rather than email, using SMS was an alternative option. However, due to the small scale of this project, paying for a VOIP service to handle SMS to most providers is unreasonable. While it is possible to use email to send a SMS message, it requires that one know the user's cell phone provider. In addition, messages sent in this method can be ugly and cluttered, possibly resulting in the user ignoring the message, or even suspecting it as spam. On top of this, messages are generally for personal communication, and an automated message relayer may confuse the users. Because of all this, the team decided against using SMS here. The ideal choice would be to use native notifications for mobile devices, but this is also an unreasonable option, as ConnectBasket will be a web application and will lack native mobile versions. Notifications through a mobile browser are technically possible, but will only take effect while the page is open on the users phone, which is likely to be very rare and is not a good solution. Sending emails from a service email or a noreply account is easier than both, free when done directly like this, can reach users even when they do not have ConnectBasket open, and easy to avoid confusion with custom body content and a more appropriate medium than texting for automated messages.

3.4.3 Timeline

Notifications should be a more simple element to the project, and should not take longer than a week to develop and test completely. While a primitive version can be created ahead of time if needed, actual use-case testing will not be possible until user profiles and user messages exist to a certain degree. Notifications will need to make use of user information, at minimum a user's contact information and their identifying username. Notifications will also need to make use of a message sent to a user, requiring the message system to be implemented and to the point where it can send at least basic messages to specific users.

3.4.4 Testing

Testing notifications should be very simple. Basic functionality can be tested immediately, by sending a dummy message to a specific email. If the basic function works, it will alert the email's owner on their mobile device, should the settings be enabled. Otherwise, testing will occur after both profiles and messages are implemented, with success criteria being: notification sent automatically, notification sent upon user receiving a message, notification sent using contact information from the user's profile, and user being able to navigate to the message via the notification. If all of these things are possible, this feature will be considered finished.

3.5 User Profile Framework

3.5.1 Details

Users of the ConnectBasket software should be able to securely log in to accounts containing potentially confidential information. These accounts should be unique and linked to a single employee of the hospital, though not every employee of the hospital must have an account. These accounts should have identifying information, for both the ConnectBasket backend to use, and for other employees to use. The user accounts should have profiles that contain their information. User accounts should have the ability to be part of one or multiple groups, which can also contain other users. These groups will be used by the issue tracker system.

3.5.2 Design Decisions

The focus for this feature is, rather than the functionality of the users, the security of the accounts and the ability to log in from various devices. Users should have profiles, but these profiles will not need to contain much more information than a username, an email, some settings, and a list of groups the user is a member of. Thus, since there is not a lot of user information, the PHP user login framework HUGE will be used, or at least referenced, for the user logins for ConnectBasket. HUGE is a simple framework without many fancy features, perfect for use in ConnectBasket. Other options researched by the team, like Userfrosting, were also simple and easy to use, but generally had more functionality than needed, which could lead to confusion or compatibility issues.

3.5.3 Timeline

Users should be one of the first things implemented in the system. While the message tracking functionality of ConnectBasket does not necessarily depend on users to exist, it is not functional and cannot be tested without it. Thus, user profile functionality is likely one of the first parts of the project software that should be done, once the web server is ready. It will need to be mostly implemented to allow other features to be started or tested.

3.5.4 Testing

Testing a basic user login function is as simple as seeing if a login is possible. This can be tested early in the project, possibly even offline, before the webserver is prepared on a developers personal computer. Once the webserver and database, a basic user-facing front-end is prepared, and user logins are implemented, they can be tested with success criteria being: User can, without cookies in place (or in an incgonito mode or a new device) navigate to the ConnectBasket webpage and log in to their unique account using a username and password, user can view information that the issue tracking system provides for them, user cannot see messages unrelated to them, user can change contact information, user can log out, a different unique user can log in and do all of the above, and a user can close the browser or navigate

to another page and be able to return to the ConnectBasket site without needing to log in again (that is, cookies). All of these tests should also be verified on a mobile device as well.

3.6 Issue Tracker Framework

3.6.1 Details

The core functionality of ConnectBasket is similar to many issue trackers or Kanban boards, without the graphical interface or board layout. ConnectBasket will include features one could expect from any issue tracker, but personalized for the hospital. The core of the issue tracking is the messages sent between users, which should be secure, accurate, and relatively fast. These messages should be directed to users and to groups. Users, as mentioned in the user profile framework section, can be part of groups, and can select messages sent to these groups. These messages can be forwarded to other locations or closed. These messages should have a history, which will always contain and report the activity of a message, with all potentially relevant details. Messages should have multiple states such as closed, open, duplicate, etc. that can be changed on demand.

3.6.2 Design Decisions

Much of the desired functionality for this project is visible in this feature, the message system. Since much of what the client is asking for is similar to a ticketing system or kanban board, the development team will be looking for examples in similar technologies. The team considered simply wrapping an existing solution with a different user interaction system, but found that such a solution would contain too many unneeded features, and would not be much effort for a senior capstone project. There are a few open source options that do similar or almost exactly what is requested by the client, but these are unlikely to fulfill the needs of this project for the reasons listed above and because the client has requested specific features that may not exist in these solutions. The team will be implementing an issue tracking message system from scratch, using the bestpractical Issue Tracker as a guide for what the final product may look like, and to use as reference should any roadblocks or issues be encountered.

3.6.3 Timeline

This feature depends on the physical hardware and the webserver existing before implementation. While it cannot be formally tested beforehand, development can be done offline on a development computer. This work may not function in the same way as it will when on a webserver, so development will also need to take place on the server regardless of how much is done offline. This feature depends on and cannot be completed until the webserver has been implemented. In addition, to view any meaningful output and for testing, the user login feature will need to be implemented to some degree.

3.6.4 Testing

A successful test of this feature denotes a working version of ConnectBasket. Early tests before full completion will likely be on whether or not the feature is capable of sending a message to a dummy user, and allowing users to enter and view messages.

3.7 Content Delivery Network

3.7.1 Details

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. Currently, most web traffic is served through CDNs, and CDN's can also help protect against some malicious attacks. Reasons to use a CDN include improving website load times, increasing content availability, and improving website security.

3.7.2 Design Decisions

The CDN chosen for the ConnectBasket project is Cloudflare. The CDN that Cloudflare provides is designed to optimize security and performance, has multiple price levels, and is easy to set up. Cloudflare's CDN provides a flat price per month that does not depend on bandwidth, which makes it a good choice because there will never be any unexpected spikes in cost.

3.7.3 Timeline

As this feature is not a required one for the project, its development will not be a priority. The addition of a CDN will be done after all other parts of the project are complete. This would likely be done in the final week of winter term or during spring term.

3.7.4 Testing

The testing of the CDN would be minimal. The only testing that would need to be done is to make sure all features of the web application were still working after the CDN had been set up. As long as it does not cause problems with the functionality of ConnectBasket, it would be considered working.

3.8 Web Development Framework

3.8.1 Details

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. ConnectBasket will have many basic features that most web development frameworks will provide libraries to easily implement. Using a web development framework to complete this project will help to speed up the development process and make the finish product aesthetically pleasing and user friendly.

3.8.2 Design Decisions

The web development framework that will be used for the ConnectBasket web application is AngularJS. AngularJS was created and is maintained by Google and provides a powerful framework for building quick and easy to use web applications. It is a Javascript framework that extends basic HTML, adding many useful features that will work nicely to provide the features needed for ConnectBasket. AngularJS was chosen for this project because it provides simple, easy to use tools that can build everything the project needs to have.

3.8.3 Timeline

The web development framework will be used to create the ConnectBasket application. Development will begin at the start of winter term and will continue until the end of winter term and into spring term until the project is complete. Development can be done on the development teams computers, but for full testing, the setup of the web server and database will need to be completed first.

3.8.4 Testing

The testing of the ConnectBasket application created by the web development framework will need to be very thorough. There will need to be tests done of all of the individual features that make up the web application. In addition to testing by the developers, testing will need to be done with the users of the application to receive feedback and make final changes to the project.

3.9 Software Design Pattern

3.9.1 Details

A design pattern is a solution to a general problem that can be reused in multiple situations. There are many different design patterns that can be used for a variety of situations, but often, there is one pattern that best fits a scenario.

3.9.2 Design Decisions

The design pattern that will be used for the ConnectBasket project is MVC, or Model-View-Controller. There are three components of MVC: Model, View, and Controller. The Model represents the data, the structure of the data, and how the data is represented in the application. The View is the component that makes the data displayable to the user in a useful way [**?**]. 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. Using the MVC design pattern for the ConnectBasket project makes sense, because it can be split into these three components. The Model component will be the patient and owner data in the database. The View will be the interface of the application. The Controller will be the way that the Data and View components communicate and are connected. Additionally, there are three developers on the development team, which makes it easy to assign roles.

3.9.3 Timeline

The software design pattern is used throughout implementation of the project and will be finished when the project is complete during spring term. It is closely associated with the web development framework and they will be completed at the same time.

3.9.4 Testing

The chosen design pattern will allow for easier testing of the finished ConnectBasket application through unit tests that can test each function that the controller provides.

4 Conclusion

4.1 Summary