Cobb Connect

Project Solution Approach

Granger Cobb Institute for Senior Living

Cobb Connect App

Gabriel Righi, Anthony Salvione,Josh Frey

10/5/2023

TABLE OF CONTENTS

b Introduction

The introduction begins by stating the purpose of the document. Explain the purpose for providing this design document and specify the intended audience for it. If this is a revision of an earlier document, please make sure to summarize what changes have been made during the revision (keep this discussion brief). Then provide a brief description of your project and state your project goal.

The purpose of this document is to provide our team's plan and approach to how we are going to implement our project. The purpose for providing a design document is to share to other programmers and teams on how we are designing and implementing our project. This document is also helpful for future teams that might work on this project if this project gets passed down, just like it was for our team. The previous team has documented their solution approach which allowed our team to gain a good understanding on how they implemented their code and their thought process. Reading the previous teams solution approach made it easier for our team to take over their code and make new changes or features that the previous team didn't complete.

Our team has been tasked with taking over and further developing a multi-platform application for the esteemed Granger Cobb Institute for Senior Living (GCISL). This application is designed to enhance the faculty's ability to track the progress of both alumni and current students in their respective careers, while also fostering a strong sense of community and connection among all users. Our application will serve as a powerful communication tool, enabling users to easily connect with each other through messaging. Additionally, it will feature a posting functionality that will keep alumni and students informed about program updates and job opportunities.

Our team's primary objective is to seamlessly pick up where the previous team left off, focusing on completing the remaining features. To ensure the application's success, we created a beta test, allowing us to gather valuable feedback from beta testers. This feedback will not only help us refine and improve existing functionalities but also identify new features that could enhance the user experience.

In addition to this, we are deploying the application on a dedicated WSU hosted server, which we have successfully been able to complete. Once the application is fully polished and ready for use, our team will conduct thorough testing to guarantee its functionality and reliability. As a final step, we will train the GCISL faculty on how to use the application.

b System Overview

The system overview contains a general description of the functionality and design of the project. The overview will only briefly describe the overall design considerations and the comprehensive explanations will be done in the sections to follow. The overview should serve as an introduction to these sections.

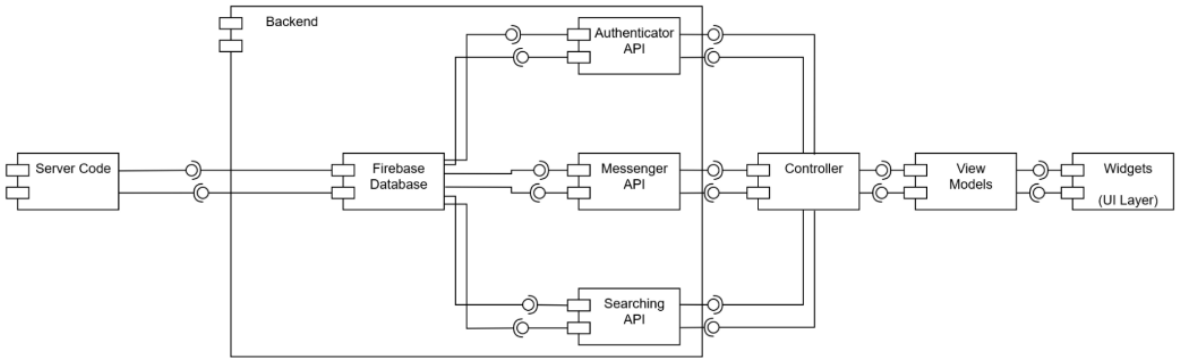
The previous team designed and implemented the project using a Client-Server architecture so our team will use their architecture to continue to implement new features. Also, the previous team implemented the project using Flutter to design the User Interface and Front-end and Firebase as our backend and API so our team will be using these as well. Flutter provides widgets that our team can use to implement how our pages are laid out and styled. We will use

these widgets to help fix the styling and how our users interact with each page and to add an admin page and new profile page. We will use Firebase's Authenticator and Realtime Database to communicate with the application and store all the data when users interact with the website. Our database will contain a user table, post table, and message table which will hold all the information about the user when they enter information in the profiles page, what type of users they are and what they can do on the application, meta data about posts and messages, and filters or settings the user wants. Users can delete their posts and faculty can delete users by deleting the data in the database. There will be a major focus on how the messaging, analytics, and home page look and function due to the user using these pages the most.

b Architecture Design

b.1. Overview

As a group we are coming in to work on a project that has already been designed. While we will most definitely make improvements to the project overall, we have no plans to make any fundamental changes to the architecture of the system. The previous cobb connect group documented their architecture system, the same one we will be using, in their Solution Approach. If you want that information, you should go view it in the github, as we are not designing a system for our project. Since we are not actually designing a system but upgrading and maintaining one, it would not make sense for us to speak about the architecture design, except in reference to what we have changed. Parts of our description might seem similar to the prior groups, for instance we will have very similar (if not identical) subsystem decomposition. This is because the prior group defined what they think the subsystems are, and we will just be stating what we will be changing in relation to those subsystems. Since this project is one continuous work it reasons that we should follow in the style of the previous documentation and try to keep as much similarity as possible, while not rehashing or copying their work.



(Since a diagram was required, this is the previous groups diagram which we aren't changing)

b.2. Subsystem Decomposition

I.1.1. Widgets

a. Description
The widgets represent the front end of our project. This encompasses the layout of pages, style, and overall theme of the website.

b Concepts and Algorithms Generated
Dart uses forms as its method to generate css,html and javascript code. We will be using these forms to describe the layout of pages we create. These forms will handle all user interaction from button presses to keyboard input.

c Interface Description
Provide a description of the subsystem interface. Explain the provided services in detail and give the names of the required services.

Services Provided:
1. Service name: Admin Page
   Service provided to:Admin User
   Description: This page will display detailed information about all users on the system
2. Service name: Profile Page
   Service provided to: User
   Description: This page will display a user's information in a profile style

Services Required:
1. Service Name: View Model
Service Provided From: View subsystem
Needs updated information from the view model to display the most accurate data.

I.1.2. View Models

1. Description
The view model describes the model of how data will be gathered and stored in our project. When someone enters information about themselves, or makes a post about something, a model defines how the data they provide will be grouped in code.

2. Concepts and Algorithms Generated
The algorithm behind this lies in how we store the data in the firebase database. We will group relevant pieces of information together with unique identifiers to reference at later points.

3. Interface Description

Services Provided:
1. Service name: Student Model
   Service provided to: Widgets
   Description: This model will help define one of the three kinds of users. Students will have reduced permissions as they are still in school so they will not need to provide certain pieces of information such as their location and job information.

2. Service name: Alumni Model
   Service provided to: User
   Description: This page will help define one of three kinds of users. Alumni will be required to provide information about their current company, and location.

3. Service name: Faculty Model
   Service provided to: User
   Description: This page will help define one of three kinds of users. Faculty, or admins, will be able to control system wide information through their accounts. They will have special permissions that allow them to delete users, their posts, and other elevated actions.

Services Required:
Service Name: Controller
Service Provided From: the controller subsystem

I.1.3. Controller

a) Description

The controller defines the actual routes the user goes through to complete tasks. This will generally be in the form of web requests. The requests will generally be calls to various firebase applications to update the database

b) Concepts and Algorithms Generated

The algorithm behind this lies in how we store the data in the firebase database. We will group relevant pieces of information together with unique identifiers to reference at later points.

c) Interface Description

Services Provided:

d) Service name: Delete (DELETE)
Service provided to: View Model, Backend - Authenticator
Description: This command will delete a post from the firebase, and remove it from the page.

e) Service name: Get Lat/Long (GET)
Service provided to: View Model, Backend - Authenticator
Description: This command gets the geolocation latitude and longitude based on the information provided to the webpage.

f) Service name: Retrieve Users (GET)
Service provided to: View Model, Backend - Authenticator
Description: This command gets all the users' information from the database.

g) Service name: Delete User (DELETE)
Service provided to: View Model, Backend - Authenticator
Description: This command removes a user from the database.

Services Required
Service Name: Firebase
Service Provided From: Datastorage

I.1.4. Firebase

a) Description

All information from the program will be stored on the firebase database.

b) Concepts and Algorithms Generated

Through web requests we will be sending and receiving information from the database

c) Interface Description.

Services Provided:

1. Service name: Users Table
   Service provided to: Controller Subsystem
   Description: This database table contains all of the users information. This is notably different from the users login information, its just their account information such as firstname lastname location etc…

2. Service name: Posts Table
   Service provided to: Controller Subsystem
   Description: This database table contains all the information about every post made by users. It stores text, images, and user

3. Service name: Messages Table
   Service provided to: Controller Subsystem
   Description: This database table contains all messages sent by users.

Services Required:
1. Service Name: Controller
Service Provided From: Controller subsystem
2. Service Name: Model
Service Provided From: Model subsystem

4. Data design

[You may skip this section if your project doesn't require any data manipulation or storage]

Describe all data structures (including the internal and temporary data structures), and the database(s) created as part of the application. This information is important from the design point of view as it will help the team in properly understanding all the data structures and databases which will be required for the coding.

There will be three primary data groups of data that are being recorded using Google Firebase; user profiles, user posts and user messaging. Through these three sets the website

can store all of the appropriate data needed to present the analytics page. All of these data sets will be internal data structures which will be saved permanently unless manual removal is done.

The user profile contains the bulk of the information on the user. The information included is the following: city, state and country location, company the user works for, time of creation, experience, first name, last name, usertype, latitude and longitude coordinates, phone number, company position and zip code. Depending on privacy settings and preferences some of the information will not be displayed for the user requesting it.There also should be a dataset that holds the users preferences. If they want to show something or hide something from their profile it should save to the database.

The message and post datasets require that the user has an account, so some of the information in these sets are dependent on information in the user profile. Posts are viewable to the public and contain three pieces of information: the post contents, name of the user who posted (if no name inputted yet, it will remain anonymous), and the time of creation. Lastly, the private messaging data set. Each user will have its own branch, containing all received and sent messages. Each message contains the message contents, user identification number of sender (obtained through a hash function), and the time of creation.

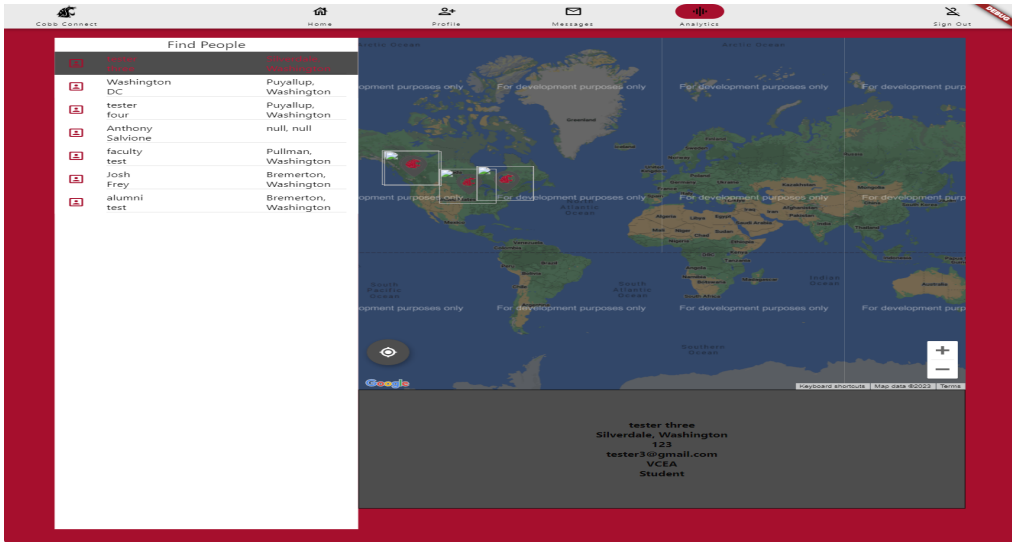
5. User Interface Design

[You may skip this section if your project doesn't have a GUI component] – but! If the tools is ever to be used by humans (even just starting and stopping it), there's some form of user interface design. It can be very simple, but it does exist. Make sure you document how you expect people to use your product, even if it's just:

- Installation
- Configuration file edits
- Launch daemon by running command [x]

Provide a detailed description of user interface. The information in this section should be accompanied with proper images showing how exactly you vision the interface to be like (for example mock-ups). Make sure to mention which use cases in your "Requirements Specification" document will utilize these interfaces for user interaction.

The primary feature of this website is the analytics map which displays all of the users and their locations using Google's API. This is where the primary interface design changes will take place. The current analytics interface design will be pasted below:



The prioritization of the interface should be in consideration of the clients. The information should be easily accessible, displayed in a neat and satisfying way and provide more specific ways to sort through information. A pulldown tab should be implemented allowing the user to filter information (such as by location, by company etc.). The information below should be displayed in a neater way.

The homepage of this project contains a text box to fill in order to post, and a list of posts underneath. A way to filter posts should also be implemented here. Tags should be added to posts as a way to sort posts easier and allow the poster to target their audience easier.

The user is expected to use the homepage, messaging and analytics page the most and this is where the user interface design efforts should be focused.

The messaging page currently only contains a pullout tab from the left side which allows the user to choose which other user they are planning to message. Before the pullout tab is pressed, the page is fairly empty. Messages and information should be displayed more clearly and organized in a way that is easy to understand. There may be no need of the pullout tab on the left side and instead display all the messages on one page and expand them when selected.

6. Glossary

Define technical terms used in the document.

7. References

(Dutoit, 2010), 3rd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice Hall, 2010.

Firebase, “Documentation | Firebase,” Firebase, 2019. https://firebase.google.com/docs

Prakhar Srivastav, “A Docker Tutorial for Beginners,” A Docker Tutorial for Beginners, 2014. https://docker-curriculum.com/

S. Pasupuleti, “How, When, and Why you should switch from Vercel to a different Hosting Provider (Especially for…,” Medium, Feb. 05, 2022. https://medium.com/@sushrit.pk21/how-when-and-why-you-should-switch-from-vercel-to-a-different-hosting-provider-especially-for-8ba25e439788 (accessed Sep. 29, 2023).

VIII. Appendices

Any larger images, charts, or external materials should be put into appendices. These are attached at the end of the document, so the main materials are kept closer together and the overall flow of the document is preserved. If you include 4 pages of spreadsheets in the middle of a section, it makes it very difficult to track the flow of your presentation. Instead, those sheets go in Appendix [X] and are referred to by the earlier document.

You may have as many appendices as you need for the document to make sense.