Cobb Connect

Project Requirements and Specifications

GCISL

Cobb Connect

Gabriel Righi, Joshua Frey, Anthony Salvione

9/28/2023

TABLE OF CONTENTS

I. Introduction

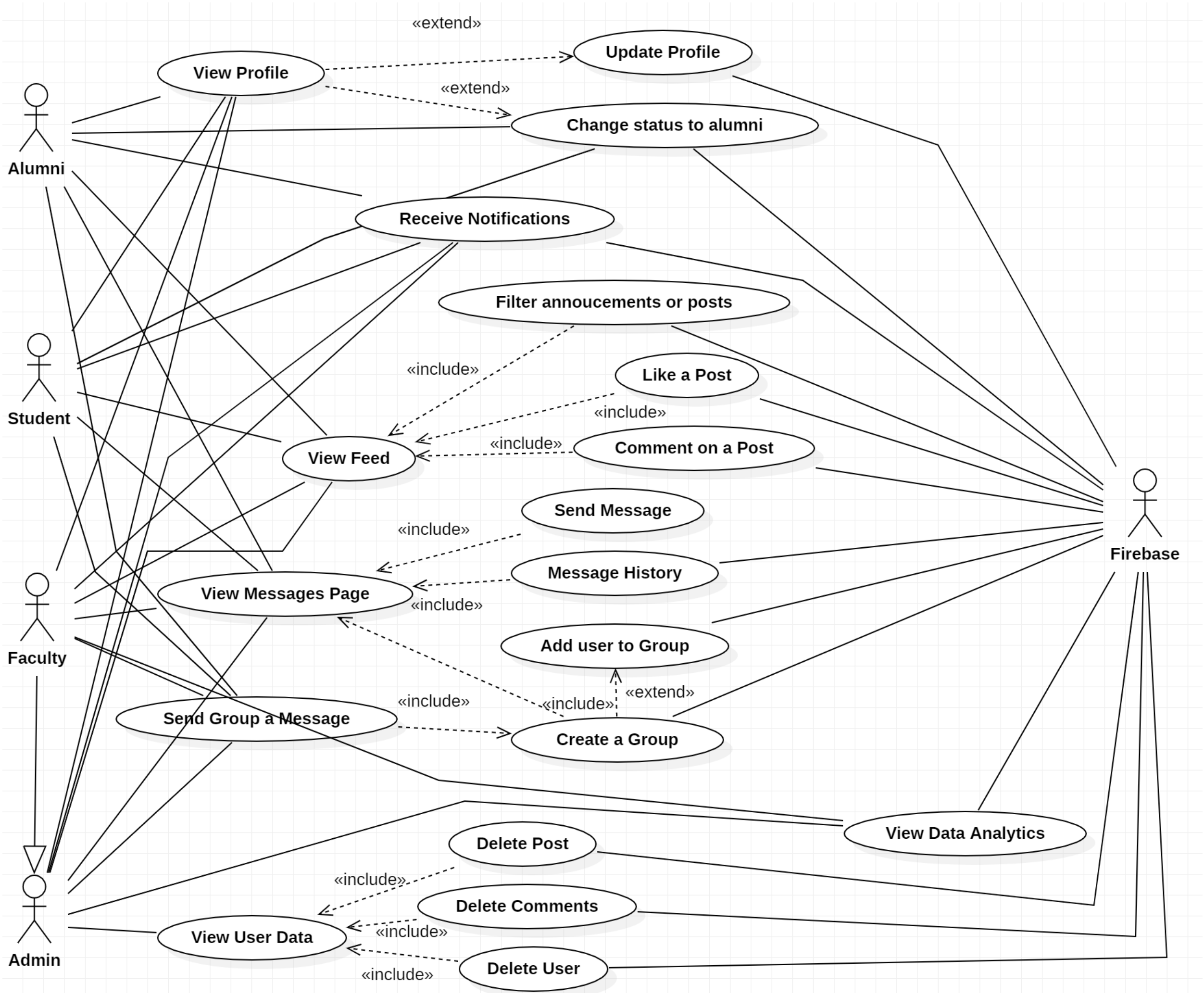
Provide a brief description of your project and summarize the objectives. Yes, this is somewhat lifted from the first writing assignment, but this document needs to be self contained and so a new reader needs a 1-2 paragraph introduction to know what the project is about.

Our team has been tasked with taking over and finish developing, testing, and deploying a multi-cross platform application for the Granger Cobb Institute for Senior Living (GCISL). This application will allow the faculty at GCISL to track alumni and students in their careers and provide a way for everyone to stay connected. This application will allow users to message each other as a way to connect with each other. The application will have a posting feature so alumni and students can be updated with the program and job postings.

Our team's objective is to take over this project from a previous team and finish implementing the features that haven't been completed. Our team will first focus on hosting a beta test on what the previous team has completed to receive feedback from beta testers on what functionalities can be improved and added. Another objective for our team to complete is to deploy the application on a WSU hosted server. Once our application is completed we will perform testing to ensure it is ready to be used by our end users and teach the faculty at GCISL on how to use this application.

II. System Requirements Specification

II.1. Use Cases



View Profile

Actors:	Student, Alumni, Faculty, Admin, Firebase
Goal:	For users to view their profile information for the application.
Preconditions:	● User has an account. ● User is logged in.
Scenarios:	● A way for current faculty members to keep track of their alumni's career status.
Exceptions:	● None

Edit Profile

Actors:	Student, Alumni, Faculty, Admin, Firebase
Goal:	For users to be able to change their profile information for the application.
Preconditions:	● User has an account. ● User is logged in. ● Users must click on the profiles button to go to profiles page first.
Scenarios:	● A way for an alumni to update their career status so faculty can keep track of their progress in their career.
Exceptions:	● None

Change Status to Alumni

Actors:	Student, Alumni, Firebase
Goal:	If a student graduates, then they can change their status from student to alumni.
Preconditions:	● User has an account. ● User is logged in. ● User is a student at first in order to change status. ● User must navigate to the profiles page first.
Scenarios:	● A way for students who graduated to change their status to an alumni so faculty members can then keep track of their career status and progress.
Exceptions:	● None

Receive a Notification

Actors:	Student, Alumni, Faculty, Admin, Firebase

Goal:	A User will be notified by a red indicator in the messages icon to show how many new messages they received.
Preconditions:	● User must be logged in. ● User must have an account.
Scenarios:	● A way for every user to be up to date with their messages so communication between each other is efficient.
Exceptions:	· None

View Feed

Actors:	Student, Alumni, Faculty, Admin, Firebase
Goal:	Users can view the feed for posts.
Preconditions:	● User must have an account. ● User must be logged in.
Scenarios:	● Allows graduates to be up to date with the program when faculty members post.
Exceptions:	● None

Filter Announcements or Posts

Actors:	Student, Alumni, Faculty, Admin, Firebase
Goal:	Users can filter to either only see posts or only see announcements in their feed.
Preconditions:	● User must have an account. ● User must be logged in. ● User must be in the home page.

Scenarios:	● Allows for users to be able to filter between seeing only posts or only announcements.
Exceptions:	● There are no posts or announcements, so nothing was filtered.

Liking a Post/Commenting on a Post

Actors:	Student, Alumni, Faculty, Admin, Firebase
Goal:	Users can like or comment under a post or announcement.
Preconditions:	● User must have an account. ● User must be logged in. ● User must be at the home page where the feed is to view posts and announcements.
Scenarios:	● When faculty members post updates about the program the faculty member can see what graduates are keeping up to date as well when they like or comment.
Exceptions:	· None

View Messages Page

Actors:	Student, Alumni, Faculty, Admin
Goal:	User can view messages on the messages page.
Preconditions:	● User has an account. ● User is logged in. ● User clicks on the messages icon to go to messages page.
Scenarios:	● Everyone can see there messages to be up to date with one another.

Exceptions:	- User sees no messages if no message has been sent to the user or the user hasn't sent out a message yet.

Send Message

Actors:	Student, Alumni, Faculty, Admin, Firebase
Goal:	Users can send a message to another user privately.
Preconditions:	- User has an account. - The other user has an account. - User is logged in. - User is in the messages page.
Scenarios:	- Allows for faculty members to reach out to alumni to keep up and track their progress and also to receive feedback from them. - Allows for faculty members to better coordinate with each other for program events.
Exceptions:	- None

Create a group

Actors:	Student, Alumni, Faculty, Admin, Firebase
Goal:	A user can create a group with multiple users.
Preconditions:	- All Users have an account. - The user creating the group is logged in. - The user creating the group must go to messages page.

Scenarios:	- Faculty members can create a group chat to message and coordinate program events. - A way for everyone to stay in touch with each other and message multiple people if - A way for all students to communicate with each other under one group. - A way for all alumni to communicate with each other under one group.
Exceptions:	- None

Add user to a Group

Actors:	Student, Alumni, Faculty, Admin, Firebase
Goal:	A user in the group can add more users to the group.
Preconditions:	- The user adding and the user being added must have an account. - The user creating the group is logged in. - The user creating the group must go to messages page. - The group needs to be created already.
Scenarios:	- If a new faculty member is hired they can be added to a group chat with just faculty members.
Exceptions:	- None

Send Group Messages

Actors:	Student, Alumni, Faculty, Admin, Firebase
Goal:	A user can send messages to a group of users.
Preconditions:	- All Users must have an account. - The user sending the message must be logged in. - The user creating the group must go to messages page.

Scenarios:	- If a faculty member needs to message only faculty members then instead of posting they can just privately message a faculty member in a group message.
Exceptions:	- None

View Data Analytics

Actors:	Faculty, Admin, Firebase
Goal:	A faculty member can view data analytics.
Preconditions:	- Must be a faculty user. - The faculty member is logged in. - The faculty member is in the analytics page.
Scenarios:	- Faculty members can view a summary of the graduates data so they can make better decisions for their program.
Exceptions:	- None

Admin Privileges (view user data, delete posts, comments, users)

Actors:	Admin, Firebase
Goal:	An admin can have administrative power of viewing or overseeing other user's posts, announcements, comments. Admin can delete users, comments, and posts.
Preconditions:	- Must be an admin. - Admin must be logged in.
Scenarios:	- If a faculty member isn't working at the program anymore the admin can delete them as a user. - Any inappropriate post or comment can be deleted.

Exceptions:	● None

II.2. Functional Requirements

II.2.1 Firebase Backend

i) Firebase backend

a) Unread notifications

Display to the user how many messages were sent to their account that are currently unread. This will be tracked via the firebase backend. The way it will be displayed is so that the number of new notifications will be in the same area as the app bar so it indicates how many are left.

b) Add new account to database when created

Store the information in firebase so that they will always have information saved. A dictionary or stack will contain all of the user's information.

c) Messaging page and data

Create a clean and functional UI to hold and display all of the viewable messages for the user and store these messages in firebase so they may be viewed again later.

d) Saving feed data

Firebase will save all of the post data and display them when the user accesses the posts. The post data will include time of post, user associated with the post and the contents of the post itself.

i) Firebase backend

Priority Level 2

Source: Darcie and WSU Students

a) Unread notifications

Priority Level 1

Source: Darcie and WSU Students

b) Add new account to database when created

Priority Level 2

Source: Darcie and WSU Students

c) Messaging page and data

Priority Level 2

Source: Darcie and WSU Students and Administrators of the website

d) Saving feed data

Source: Darcie and WSU Students

Priority Level 1

II.2.2 Interface and Navigation

ii) Interface and navigation

a) Creating an interface to display the data

Present information in an accessible and clear way

b) Auto filling information onto the profile page

When the user is logged in and they already have an account with information connected to it, their profile page will be filled with their proper information.

c) App bar

A bar at the top of each page which allows the user to easily navigate the website

d) Settings page

Create a clean and functional UI to hold and display all of the user settings

e) Messaging page and data

Create a clean and functional UI to hold and display all of the viewable messages for the user and store these messages in firebase so they may be viewed again later.

ii) Interface and navigation

Source: Darcie and WSU Students and Capstone Counselor

Priority Level 3

a) Creating an interface to display the data

Priority Level 3

Source: Darcie and WSU Students

b) Auto filling information onto the profile page

Priority Level 3

Source: Darcie and WSU Students

c) App bar

Priority Level 3

Source: Darcie and WSU Students

d) Settings page

Priority Level 2

Source: Darcie and WSU Students

e) Messaging page and data

Priority Level 3

Source: Darcie and WSU Students

II.2.3 Main Functionality

iii) Main functionality

a) Feed page and ability to edit feed

Functionality to sort which updates and messages you want to view and a clean and functional UI for the user to navigate their updates and feed.

b) Creating a post

Certain users who have the access to post updates onto the feed, which can be viewed by all users

iii) Main functionality

Priority Level 2

Source: Darcie and WSU Students

a) Feed page and ability to edit feed

Priority Level 3

Source: Darcie and WSU Students

b) Creating a post

Priority Level 3

Source: Darcie and WSU Students

II.2.4 Network and Security

iv) Network and Security

a) Restrict access unless logged in

Some things will not be able to be viewed unless the user has access (logged in or not)

- b) Hosting the website
    - WSU servers will host this website
- c) Password hashing and encrypting
    - Users have their data secured (most likely provided by either firebase or WSU servers)

iv) Network and Security

- a) Restrict access unless logged in
    - Priority Level 2
    - Source: Darcie and WSU Students
- b) Hosting the website
    - Priority Level 2
    - Source: Darcie and WSU Students
- c) Password hashing and encrypting
    - Priority Level 2
    - Source: Darcie and WSU Students

II.2.1. [The name of the next module/component/part]

ii) Interface and navigation

- a) Creating an interface to display the data
    - Present information in an accessible and clear way. The correct information is shown when presented with a filtered search and/or permissions are granted to see the given posts.
- b) Auto filling information onto the profile page
    - When the user is logged in and they already have an account with information connected to it, their profile page will be filled with their proper information.
- c) App bar
    - A bar at the top of each page which allows the user to easily navigate the website. Every tab of the website will be included in this bar. The bar will add or remove different tabs depending on the access of the user.
- d) Settings page
    - Create a clean and functional UI to hold and display all of the user settings.
- e) Messaging page and data
    - Create a clean and functional UI to hold and display all of the viewable messages for the user and store these messages in firebase so they may be viewed again later. Prioritize a quick way to store and sort data for these messages and data.

iii) Main functionality

- a) Feed page and ability to edit feed
    - Functionality to sort which updates and messages you want to view and a clean and functional UI for the user to navigate their updates and feed. Different sorts of feed posts may include: Recent posts, posts flagged as football related, posts from one specific user, etc.
- b) Creating a post
    - Certain users who have the access to post updates onto the feed, which can be viewed by all users. The users that can post include staff, administrators, school alumni and others.

iv) Network and Security

a) Restrict access unless logged in

Some things will not be able to be viewed unless the user has access (logged in or not). This is important for the website to maintain its user integrity and requires that the user creates an account if most features are to be accessed.

b) Hosting the website

WSU servers will host this website

c) Password hashing and encrypting

Users have their data secured (most likely provided by either firebase or WSU servers). Firebase contains a feature and documentation to safely secure user passwords.

II.3. Non-Functional Requirements

The app overall shall be developed using Flutter as the website frontend, and firebase as a website backend. These choices were made by the previous developers, and we will continue in the same manner. To actually implement new features suggested by the Beta Testers, we will be using the agile software methods with a series of sprints. For each sprint we will outline a series of tasks that we want to complete, and we will get them done by the end of the sprint.

Hosting

The website shall be hosted on WSU servers. Having a locally hosted website allows for quicker error diagnosis, faster service, and easier maintenance overall.

Usability

An average non-technically inclined user should have no trouble navigating the website. All UI elements should be clearly defined, and adhere to common standards for icons and usage. The website should be intuitive and easy to access for all users.

Reliability (Robustness)

The website should never go offline due to software malfunction, except in the case of server failure. The code of the website should be written robustly enough so that no individual user can crash the website through misuse.

Reliability (Scalability)

The website should be able to sustain 100 concurrent users and store the data for 10x its concurrent limit. Its storage capability should scale with the user count, and it should not need any input to increase storage space.

Performance

There should be no noticeable lag between user actions responses on the webpage. Actions that interact with the database should appear within a second of the submission or changing of any data.

Supportability

There should be at least 1 admin account that has the ability to remove user content and remove users in general. This admin account should be passed over to GCISL and they will moderate their app as they see fit.

Packaging

The entire app should be delivered in a docker container. This container should start the website upon instantiation, and should work on any docker system. The system itself can be installed on as many servers as necessary as it shares a cloud database.

III. System Evolution

Our project is a continuation of a previous group's application. Their application met the criteria defined by the client at the time, but was never actually implemented for the client's use. Our goal is to allow the project to be transferred and hosted on WSU servers so that GCISL can use it. After that we will have beta testers determine what they like and dislike about the application. This is how we will adapt to changing user needs, by actually changing the website. We will gather feedback during the setup of the website and change it where necessary.

A key design aspect of the application is that it will be delivered in the form of a docker container. This has the benefit of being able to run on any system that has docker. Within the dockerfile, it specifies versions of every program that is necessary for the app to actually run. This means that as technology advances the dockerfile will always stay relevant as it has versions for every program. Since we decided to develop using docker, it means that our project is easy to transport and upgrade. We won't have to worry about system requirements as if the system is running docker, it can run our program. There are very few risk points in our current project as the project itself has already been created. We are merely adding features and transferring ownership. There could be an issue with us learning out the previous project functions, but that is unavoidable. We might have issues adding new features depending on how the previous group implemented certain functions, but we can't know that until we actually start implementing changes.

Google ending firebase as a service is a potential future problem for our website. Firebase obviously as a website database cannot be inside of a docker container, and thus our project is reliant on its maintenance. If Google ever decides to end the service, then the website will cease to function. This does not seem like a large worry as Google is a massive tech company and it would be unlike them to cancel a service like firebase.

Another potential issue could be actually transferring the docker container to WSU servers. We could have issues with the timeline of getting the data over to WSU. These things take a long time to happen and there are many logistics to getting WSU to host a website.

Google maps API might also be an issue. The API used to be free to use, but has since switched to a paying model. We will need to secure funding from WSU in order to secure proper functioning of the website. Not getting funding could be an issue. This issue might also crop up with the geolocation API that checks someone's address when they set that value in their profile. The API key seems to run out after some amount of time, so we will need a way to get a long term api.

IV. Glossary

Docker: A platform for packaging and deploying applications in containers

Flutter: Google's toolkit for building cross-platform mobile, web, and desktop apps.

Firebase: Mobile and web app development platform with various backend services.

V. References

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).