Chi Alpha: Christian Fellowship

Full-stack Mobile Application

Project Solution Approach

Chi Alpha WSU

Web Ready

Vincent Yen, Justin Lee

10/05/22

TABLE OF CONTENTS

I. Introduction

The intent of this project is to create a resource in the form of a mobile application for Chi Alpha WSU. The original project was to build a more flexible website builder for the web administrator; however, after careful deliberation and time spent in requirements building with the client it was proposed that building a mobile application would be most beneficial to both the client and the development team. This document serves as an overview of the intended work for this project.

II. System Overview

The project is intended for the organization to increase their presence and access to and for current members of the fellowship. The mobile app will allow the organization to make announcements, advertise events, serve as a medium for individuals to make donations, and communicate with the organization.

III. Architecture Design

III.1. Overview

This project follows most closely with a MVC model- model-view-controller, of system architecture. The application closely resembles most website MVC models in that it serves content to pages for viewing, information about users and media are modeled for the database schema, and a controller service controls routes to the database and authentication service based on various requests from the user.

The diagram given by Appendix A figure 1 serves to show how the system is decomposed. The system proposed for the project is decomposed into 4 major parts: pages, user menu, server API, and database.

Each page is composed with an application page that a user can view to obtain various content provided by the content creator. The announcement page allows the content creator to serve announcements that the organization would like the fellowship community to be aware of. The events page allows the content creator to publish events that are happening in the community that users can sign up to attend. The giving page allows members of the community to donate to the organization through a third-party payment API service. The contact us page allows users to contact the organization via email service. The about us page gives information about the organization and its staff members. The media page allows the content creator to publish different forms of media such as videos or pictures for users to browse and consume.

The user menu and its subcomponents are necessary specifically for all components that require user differentiation during their session. Currently the only component that requires a user login is the chatroom so that each user can be distinguished from other others in a simultaneous view session.

The server API serves as middleware between the mobile application and the database to aid in serving different media stored in the database as well as user authentication.

The diagram in Appendix B figure 1 serves to show the different dependencies and interactions between each package in the system. The diagram given in Appendix B figure 2 serves as an overview of how the application will interact with the various package components.

The pages require the user interface to serve information for viewing. The user may interact with various pages for donations or to submit forms for things like event sign ups. As such the forms require pages and an API server to serve the form for viewing and to save the information into the database for the content creator to view later. Donations also require pages as a way for users to interact with and submit their payments. The donations also require a payment service which will be provided by a third-part payment API.

The chat package requires the user interface in order for other users to interact with and submit their messages. The user may or may not login in order to identify themselves or remain anonymous.

The server serves as the middleware between the forms and pages as well as the user authentication service, Forms being submit from a page will be sent to the server for validation and saving into the database. Users logging into the application will need to be authenticated with the user authentication service which will be validated against the database using the server.

The database package serves to save all information about users or store different forms of media based on the models provided in the server.

III.2. Subsystem Decomposition

III.2.1 User Interface

a) Description

The user interface is as its name implies. It serves as a graphical interface for the user to interact with the various components within the application.

b) Concepts and Algorithms Generated

The user interface is built using the React-Native framework which uses components derived from ReactJS. This framework allows the mobile application to be developed for both of the popular mobile operating systems- iOS and Android, simultaneously without having to work specifically with the native code for those operating systems.

The drawback for using React-Native is having to use the specific syntax used in that framework as well as not being able to perform more specific functions that are only available in the native language for those operating systems.

III.2.2 Pages

a) Description

The pages serve as the view component for users to view different content provided by a content creator.

b) Concepts and Algorithms Generated

Since each page is likely not change in its form each page in the page component will hard coded based on the requirements provided by the client. A basic template will allow serving of various content from the database in the media page, announcement page, or the events page.

A drawback of this would be that it would limit the flexibility of the content creator to form the view of the given pages if in the future it is decided that the view of the page is not to the liking of the content provider.

c) Interface Description

Services Required

1. Service Name: RESTful service
   Service Provided to: Server
   Description: The server will provide all content that is requested from each page that is consumed by the user.

III.2.2 Forms

a) Description

Each form will provide various ways for users to submit information that the content creator is asking for.

b) Concepts and Algorithms Generated

Each form will be created from a template which should allow some flexibility for the content creator to enter their own specific information or questions for user submission.

If the content creator requires a more specific model of how the forms should be represented it may require more tailored forms that would deviate from templating and more towards hardcoding.

c) Interface Description

Services Provided

1. Service Name: Form template
   Service Provided to: Forms
   Description: Provides a template for all forms to be used for the content creator.

III.2.3 Chat

a) **Description**

The chat subsystem allows various users to chat with other users in the app.

b) **Concepts and Algorithms Generated**

The chat subsystem will be a socket interface using session tokens for each user that logs into the chat system during their view session. A user that spends a certain amount of time idle in the chat will be removed from the system in order to alleviate resource usage. Users may login with prior created login credentials to identify themselves in the chat or enter chat without logging in as an anonymous guest.

A particular draw-back of allowing anonymous guests is the possibility of traffic flooding or a form of DDoS attack. Furthermore, by remaining anonymous various individuals may be able to act in other malicious ways in the chat service without repercussions because the individual cannot be easily traced and stopped.

c) Interface Description

<u>Services Required</u>

1. Service Name: Server
Service Provided to: Chat
Description: The chat system requires user information for each session user in the chat. The server also serves as the middleware for all information exchanged in the chat so that a single user may broadcast to other session users or receive information that was broadcast from other users.

III.2.4 Server

a) **Description**

The server serves as a middleware between the database and the resource application. The server will route various requests to various functions in the server such as user authentication, saving information submitted from forms into the database, or serving various media requests from the Media page.

b) **Concepts and Algorithms Generated**

Given that the server is middleware between the database and the mobile application, this system serves as the controller component in the MVC architecture. The server will contain routes for which various requests will be routed to functions. The database schema will be built based on models provided in the server. Any forms generated for user submission will be built from models provided in this server as well.

A draw-back of this design is that if changes need to be made in database schema it may be difficult to change without affecting the information already being stored in database. Possible ways to overcome this is to make a new schema model to meet the new needs of the content creator while maintaining the old model for previously store information.

c) Interface Description

Services Provided

1. Service Name: User Authentication
   Service Provided to: Chat
   Description: The user authentication service verifies the credentials of a user that is logging into the system. This allows the user to be identified distinctly in the chat system.

2. Service Name: Database Model
   Service Provided to: Database
   Description: The database model provides the database with the template needed to form the database schemas in which each storage item will be follow.

3. Service Name: Routes
   Service Provided to: Pages/Forms
   Description: Any page that requires particular content will require requests from the server which the server will determine which function it should be routed to based on the type of request. Requests may include: get, put, and post.

Services Required

1. Service Name: Object Storage
   Service Provided to: Database
   Description: The server requires storage for all the information that it receives from the front-end.

III.2.5 Database

a) Description

The database serves a single point of storage for all information that is needed for serving various content or saving content. Some of this information includes: media storage, user credentials and identification, organization and staff bios, and information submitted from forms.

b) Concepts and Algorithms Generated

The chosen database structure will be MongoDB which will be paired with the Mongoose JavaScript library. MongoDB uses a non-relational type of storage which provides more flexibility and quicker data retrieval when compared to SQL type databases. Currently there are no forms of data being saved that requires indexing therefore this lends itself well to this form storage.

c) Interface Description

Services Provided

1. Service Name: Data Storage
   Service Provided to: Server
   Description: Data storage allows the server to obtain pertinent information that was previously saved for various purposes such as user validation, form data storage, and media storage.

Services Required

1. Service Name: Models
   Service Provided to: Server
   Description: The database requires models that are specified by the server in order to determine how it should form its schema templates.

IV. Data design

The database will be using a non-relation type of database called MongoDB. MongoDB represents data in the form of JSON as opposed to tables as seen in SQL type databases. This form of database does not enforce the use of schema's and gives it more flexibility in how objects can be stored. Furthermore, this flexibility improves the performance read/write as it does not require information to be inserted into a rigid table schema [5].

V. User Interface Design

The picture provided in Appendix C figure 1 serves as a general image of how the graphical user interface should appear. A menu with clear descriptions serves as a means for the user to switch between the different page views for consumption. Future renditions should include a user menu, not shown in the figure, which provides options to login and perform various functions that require user credentials such as the chat system.

User should be able to obtain the application via the Google Play store for Android devices or the Apple App store for iOS devices. In order to obtain the application from these stores the user must already have an account or otherwise register to create an account with the relative store in order to gain access to the store which allows the user to download the application onto their device.

VI. Glossary

Android – The world's most popular mobile operating system [1].

API – Stands for application programming interface. A third-party application that may be used in conjunction with another application for its specific services [4].

iOS – A mobile operating system for Apple mobile devices [6].

JavaScript - A scripting language used in web development to implement complex features and logical algorithms [7].

JSON – Stands for JavaScript Object Notation. It is used as a format for data transfer on the web [9].

Middleware – Software that allows two or more applications or application components to communicate with each other [10].

MongoDB – A non-relational database that stores information in the form of objects similar to JSON structure [5].

Mongoose – An Object Data Modeling JavaScript library used in conjunction with MongoDB and a JavaScript library called NodeJS. Mongoose allows relational database management and schema validations while still maintaining some flexibility in the non-relational structure of MongoDB [2].

React-Native – React Native is an open-source mobile framework using JavaScript to design applications for IOS and Android. React Native is derived from ReactJS, a JavaScript framework used for web application development [3].

ReactJS – A JavaScript library which is used to build reusable UI components in Web development [11].

SQL – A relational database which stores data using tables and rows and enforces “referential integrity” [5].

RESTful – REST stands for Representational State Transfer. An API that uses this architectural design is said to be RESTful. REST is used to across various forms of network protocols to transfer information [8].

VII. References

[1] Brown, C. Scott. *What is Android? Here’s everything you need to know.* Feb. 19, 2022. Accessed: Sep. 29, 2022. [Online]. Available: https://www.androidauthority.com/what-is-android-328076/

[2] Karnik, Nick. *Introduction to Mongoose for MongoDB.* Feb. 11, 2018. Accessed: Oct. 4, 2022. [Online]. Available: https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/

[3] Shilpa S. *What is React Native?*. Jul. 01, 2021. Accessed: Sep, 29, 2022. [Online]. Available: https://www.tutorialspoint.com/what-is-react-native

[4] *Application Programming Interface (API)*. Accessed: Sep 18, 2022. [Online]. Available: https://www.ibm.com/cloud/learn/api

[5] *MongoDB vs. MySQL Differences.* Accessed: Oct. 4, 2022. [Online]. Available: https://www.mongodb.com/compare/mongodb-mysql

[6] *iOS.* Aug. 28, 2012. Accessed: Sep, 29, 2022. [Online]. Available: https://www.techopedia.com/definition/25206/ios

[7] *What is JavaScript?.* Sep. 14, 2022. Accessed: Oct. 4, 2022. [Online]. Available: *https*://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

[8] *What is RESTful API?.* Accessed: Oct. 4, 2022. [Online]. Available: https://www.mulesoft.com/resources/api/restful-api

[9] *JSON Defined.* Accessed: Oct. 4, 2022. [Online]. Available: https://www.oracle.com/database/what-is-json/

[10] *Middleware.* Mar. 5, 2021. Accessed: Oct. 4, 2022. [Online]. Available: https://www.ibm.com/cloud/learn/middleware

[11] *ReactJS – Overview.* Accessed: Oct. 4, 2022. [Online]. Available: https://www.tutorialspoint.com/reactjs/reactjs_overview.htm

VIII. Appendices

VIII.1. Appendix A

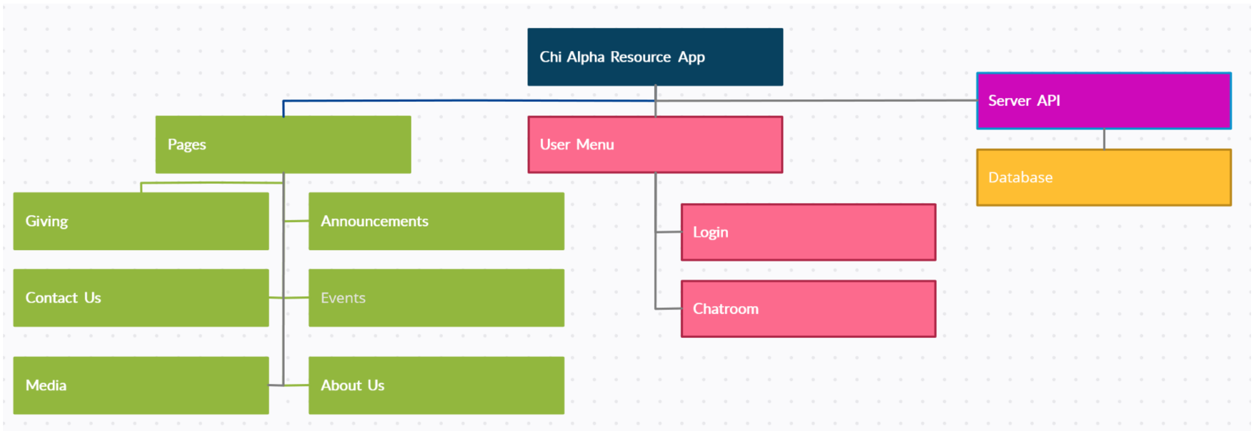


Figure 1

VIII.2. Appendix B

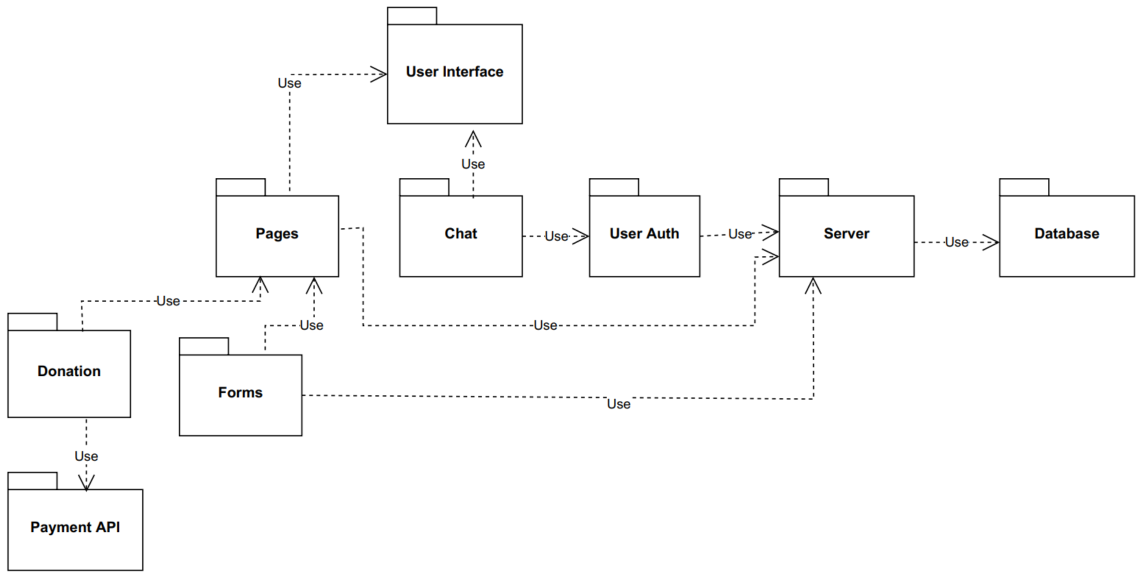


Figure 1

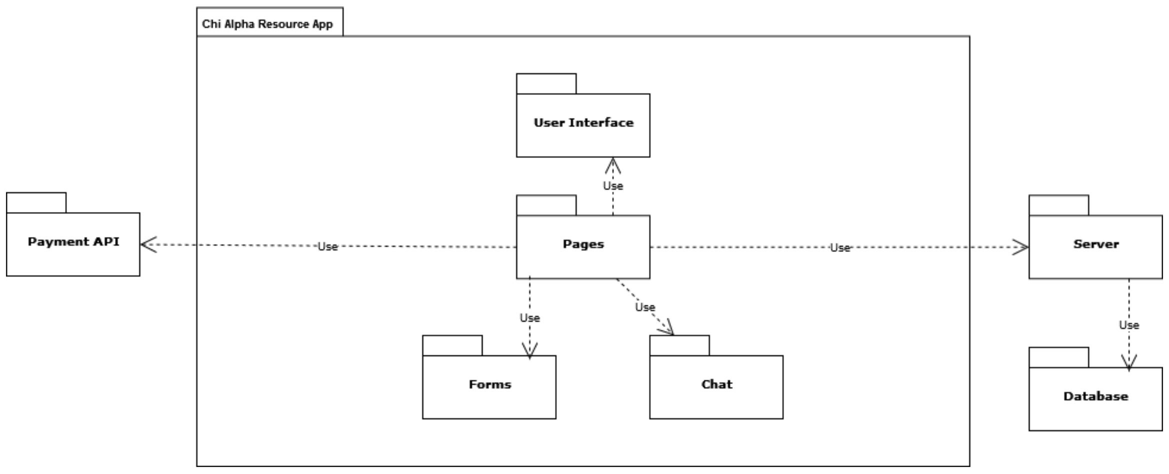
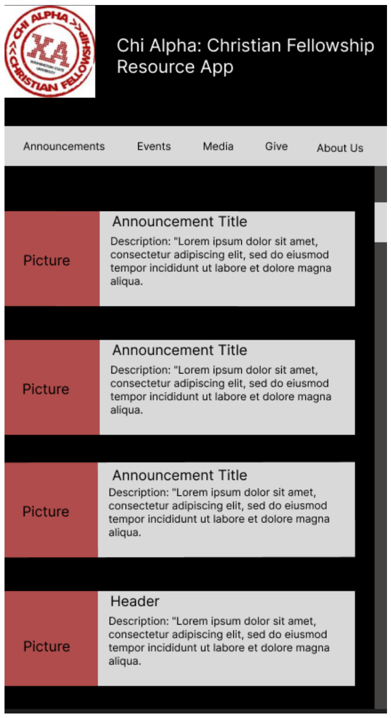


Figure 2

VIII.3. Appendix C



Chi Alpha: Christian Fellowship Resource App

Announcement Title

Description: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Announcement Title

Description: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Announcement Title

Description: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Header

Description: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Figure 1