**Problem**

Pinecrest Gardens features over 1000 varieties of rare and exotic tropical plants and palm trees in a native tropical hardwood and cypress setting. You can spend a memorable day exploring, pack a picnic, enjoying a lecture or a class or watching a movie. The Village of Pinecrest wanted to develop a mobile application that could be used as a marketing tool to promote the facilities at Pinecrest Gardens, integrating it with social media and compatible with iOS and Android devices.

**Current System**

Currently the City of Pinecrest has a mobile website that provides an overview of Pinecrest Gardens. However, when using a smartphone or tablet, a website is not as convenient to use as a mobile application.

**Requirements**

**CMS API:**

- Act as an intermediary between Mobile Application backend subsystem and the database or CMS.
- Handle http requests from Mobile Application backend.
- Receive request to get a specific page from the Mobile Application.

The client had a specific requirement list that included over 40 different required functions of the application. I was responsible for:

- Receive Pages Request (CMS011)
- Receive Multimedia Request (CMS012)
- Receive Subscriber Request (CMS 013)
- Send Contact Request (CMS 014)

**Implementation**

To build the API, I used PHP, a server-side scripting language designed for web development but also used as a general purpose programming language. The backend of the application communicate with the API using an HTTP Request. The API get the commands from the URL and send JSON file to the backend of the Application.

For implementing the Database I used MySQL to create the tables and PHP to create the different classes needed. The first step of implementing the database was to identify the different tables needed for our application and the CMS from that I was able to implement the ER diagram. The second portion of implementation was to create the different tables need and develop the different classes and functions need for the queries that will be run by the APP.

**System Design**

For the mobile application we decided to go with a Model-View-Controller architecture, enforcing the separation of logic, presentation and data. Also, we chose to go with a Client-Server architecture to communicate with the CMS.

**Verification**

Testing for this project was broken up into three stages:

- **Unit Testing:** All API specific functionalities were individually tested using PHP Unit to make sure that they worked as specified by the Integration Testing.
- **Integration Testing:** I tested that the CMS API was working as expected with the backend of the mobile application. Due to the small interdependence between our different modules we use the Big Bang approach.
- **System Testing:** This consisted primarily of exploratory test, making sure that the integrated system is in compliance with the specified requirements.

Another tool that I used to test the API is Postman. This tool is excellent for testing API, look at the figure below:

**Screenshots**

After three month of hard work, we have successfully created a solution for our client. The CMS API has been working as expected. Also, the front-end and back-end of the mobile application has been implemented and it is working as expected. Also, with the help of the CMS developed by Yoel Nunez and my API the content of the entire application will always be up-to-date. For the user to always receive the latest content of the Mobile Application they will need to close the Mobile Application and open again and then it will be updated. As of today, the application is ready to be downloaded from App Store on iOS devices or Google Play on Android devices.

**Acknowledgement**

The material presented in this poster is based upon the work supported by Gabriela Wilson and Pinecrest Gardens. I am thankful to the help that I received from my teammates Eric Weiterman, Jorge Lopez and Yoel Nunez.