Senior Project

MC-EMR

Student: Steven Berlanga, Florida International University
Mentor: *Steven Luis, Mentor*
Instructor: Masoud Sadjadi, Florida International University

FIU School of Computing & Information Sciences

Orant CHARITIES the adventure of giving.

Problem

Doctors in Malawi need a client server system to allow them to triage, track, store, and provide metrics on patients they assess in rural areas of Africa.

My specific problem space is:

- Provide an API to allow the Server application to communicate with the Cloud application
- Implement the process of moving batched information to the cloud API from the Server
- Create callbacks on the Server and Cloud applications that will handle API responses

Current System

What they offer

Current Systems are either standalone applications or applications that require constant connection to the internet to sync.

What We Need

- A system that do not require internet connection
- A system that allows multiple application users (Client) to stay in sync with each other
- A system that allows all information gathered by the clients to be stored in a central location
- A system that allow users in a remote location to view the data gathered in the field

Requirements

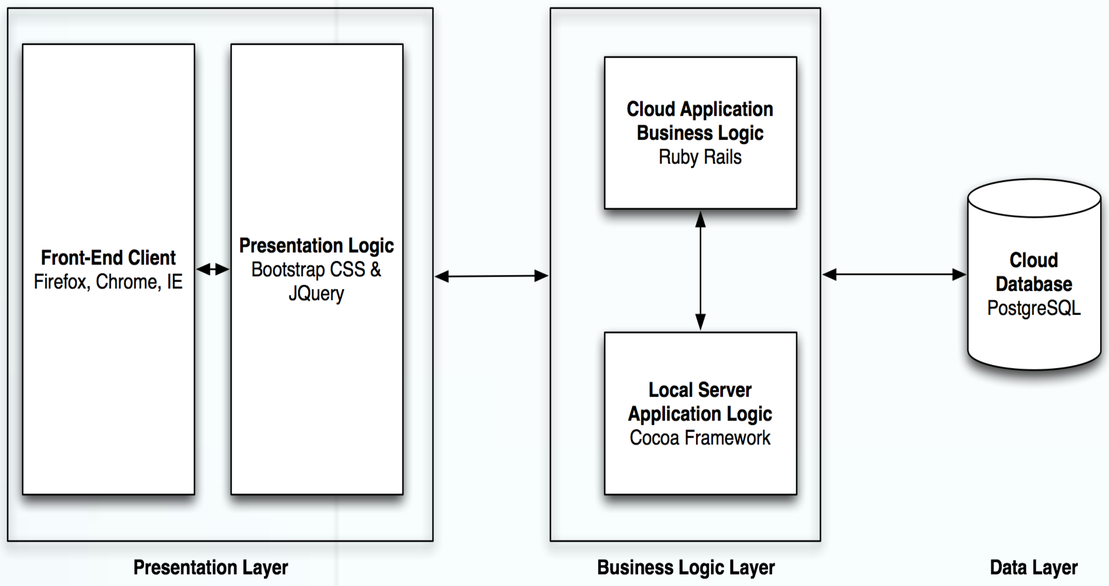
US-39:
As an application manager I want to remotely sync to any Mobile Clinic so that I can get current information.

US-110:
As an application administrator I want to be able to download information from the Cloud to the Server

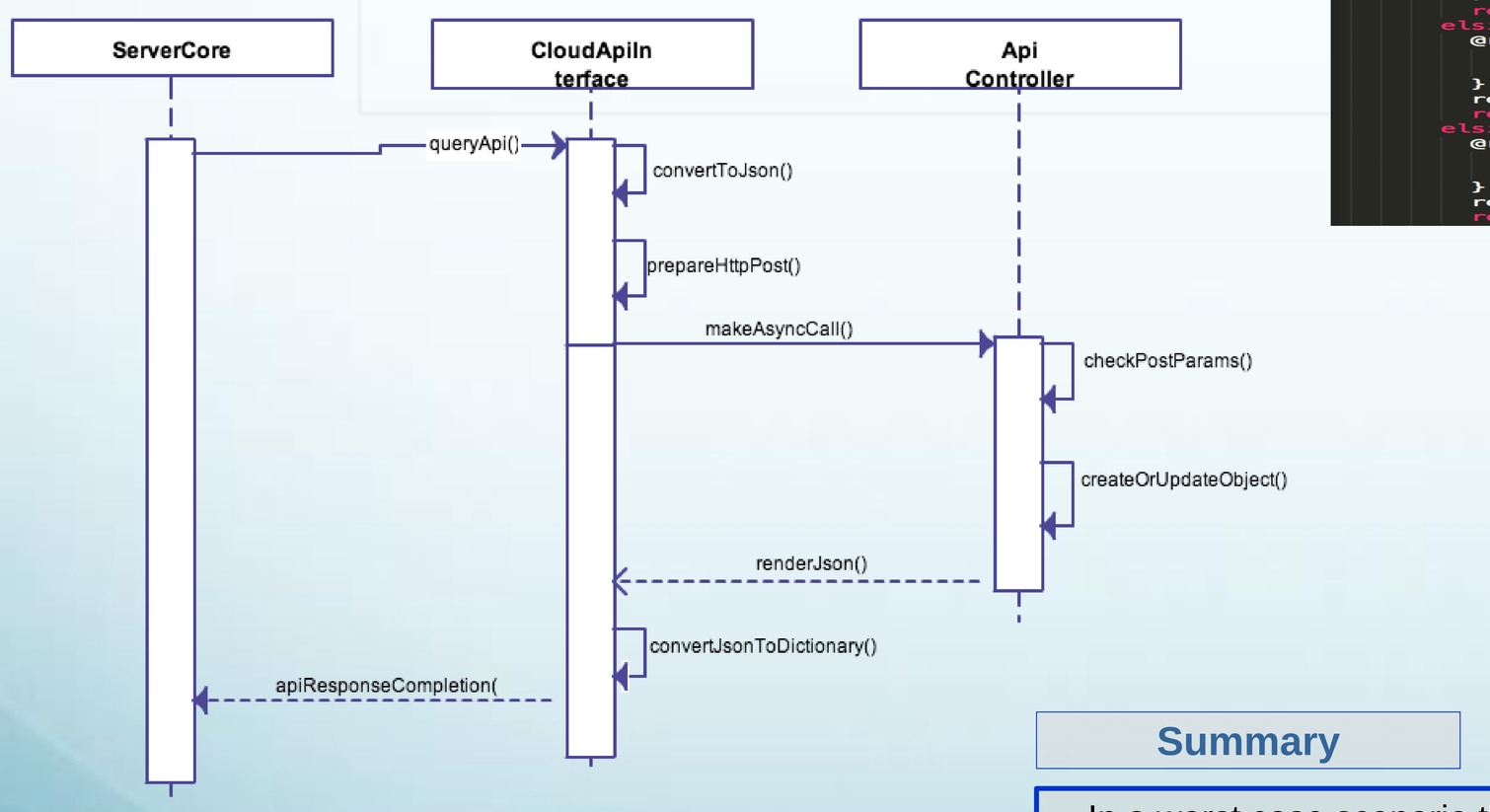
US-109:
As an application administrator I want to be able to upload information from the Cloud to the Server

US-21:
As an application administrator I want to verify when the local system backs up to the Cloud so that I can verify the data's status

System Design



Object Design



Implementation

- The api followed the REST api architecture.
- All POST and GET functions response to JSON format only and respond with a JSON encoded response.
- A two layer authentication process was put into place to secure the API.
- A standardize response was put into place in order to make the API response most easily to read and interpret.

```
def update_user
#try to find user for that id
begin

  #extract post params
  postParams = JSON.parse(params[:params]);

  if(!postParams.has_key?('access_token'))
    @response = {
      :result => 'false',
      :data => 'missing access_token'
    }
    render json:@response
    return
  end
  isAuthenticated(postParams['access_token'])

  @users = postParams['Users']
  @users.each do |user|
    #check to see that POST has all the appropriate parameters to
    #create a user. If post does not, return json response with a
    #result of false

    if(!user.has_key?('userName'))
      @response = {
        :result => 'false',
        :message => 'missing user\'s username - Param ["userName"]'
      }
      render json:@response
      return
    elsif(!user.has_key?('password'))
      @response = {
        :result => 'false',
        :message => 'missing user\'s password - Param ["password"]'
      }
      render json:@response
      return
    elsif(!user.has_key?('firstName'))
      @response = {
        :result => 'false',
        :message => 'missing user\'s first name - Param ["firstName"]'
      }
      render json:@response
      return
```

Verification

- OCUnit was used to check the response time/accuracy from the API on the Local Server. These tests were written to trigger every negative response form the API for more branch coverage; a coverage percentage of 96.5% was achieved.
- In the cloud application, both Unit testing and subsystem testing (the cloud app being our subsystem) was done using Rspec. Over 23 tests were written with a coverage of 89%.
- Apache

Summary

- In a worst case scenario the system will store data on the client, even if the server is down.
- Stable data is accessible to all client applications and the locking mechanism prevents users from overwriting each other's information
- The server can Export and Import to and from a JSON file as well as the Cloud App which allows overseas users to access the data

Acknowledgement

I am thankful for the help that I received from my group members, my mentor, and advisor.