**Requirements**

**The application shall have the following components:**

**PROFILE** – app

1. Has the following attributes
   1. email
      1. No two users may have the same **email**
   2. password
      1. Hashed upon entering
   3. name
   4. location
      1. Generated from location services on app
   5. bio
   6. phone
   7. genres
      1. Can select multiple from a list
   8. instruments
      1. Can select multiple from a list
   9. profile\_picture
      1. Path in server

**PROFILE** – server

1. Has a “profile” table with the following attributes
   1. email
      1. primary key
      2. varchar(50)
      3. Cannot be NULL
   2. name
      1. text
   3. password
      1. text
      2. Should be hashed
      3. Cannot be NULL
   4. Location
      1. varchar(100)
      2. gps coordinates
      3. Cannot be NULL
   5. Bio
      1. Text
   6. phone
      1. varchar(12)
   7. profile\_picture
      1. A path to profile picture
      2. Text
2. Has a “profile\_genre” table with the following attributes
   1. profile\_genre\_id
      1. primary key – auto increment
      2. int(20)
      3. Cannot be NULL
   2. profile\_id
      1. foreign key
      2. Cannot be NULL
   3. genre
      1. text
      2. Cannot be NULL
3. Has a “profile\_instrument” table with the following attributes
   1. profile\_instrument\_id
      1. primary key – auto increment
      2. int(20)
      3. Cannot be NULL
   2. instrument
      1. text
      2. Cannot be NULL
   3. profile\_id
      1. foreign key – auto increment
      2. int(20)
      3. Cannot be NULL
4. Has a “profile\_ad” table with the following attributes
   1. profile\_ad\_id
      1. primary key – auto increment
      2. int(20)
      3. Cannot be NULL
   2. profile\_id
      1. foreign key
      2. int(20)
      3. Cannot be NULL
   3. Description
   4. looking\_for
      1. text
      2. Cannot be NULL
   5. genre
      1. text
   6. instrument
      1. text

**API – server only**

1. Implemented in python
2. The API accepts requests using the following
   1. URL = http://bmckalla.w3.uvm.edu/api/?action=”[see part b]”
   2. The action currently can be as follows
      1. create\_profile (POST)
      2. get\_profile (GET)
      3. update\_profile(PUT)
   3. A JSON object in the data field that contains parameters associated with the app side objects
      1. i.e. create\_profile contains email, name, password, etc.

**HTTP\_SEND**

1. Is a module for submitting requests to the API
2. Takes in a JSON encoded object
3. Has a method for **GET** requests to the API
   1. Returns JSON decoded object for success
   2. NULL if failed
4. Has a method to **POST** requests to the API
   1. Returns True for success
   2. Returns False if failed

**LOGIN**

1. Is a **ViewController**
2. Has two text boxes for username and password
3. Has a button to login with username and password entered
4. Has a button to start registering a user
   1. Segues into **REGISTRATION**

**REGISTRATION**

1. Is a **ViewController**
2. Prompts user for all variables associated with a **PROFILE** (app)
3. Validates/sanitizes all variables entered
   1. Hashes passwords
4. Generates a JSON object using the **JSONSerialization** class
5. Sends a **POST** request using the **HTTP\_SEND** method to try to create the profile
6. Verifies the request was valid
   1. Segues into home screen (Hub) if so
   2. Displays a warning screen if not

**HOME SCREEN**

1. Has a button to view profile with currently logged in user
   1. **GET** request for **PROFILE** (server) data from API using **HTTP\_SEND**
   2. Segues into **PROFILE\_DISPLAY**

**PROFILE\_DISPLAY**

1. Is a **ViewController**
2. Parses JSON to display associated **PROFILE** (app) attributes
   1. Contains a field for every attribute
3. **GET** request for **PROFILE’s** profile\_picture path on server to [url]/pics (TBD)
   1. Displays if successful
   2. Does not display if not successful