CMPT475/477 – CS/IT/IS Capping Project

Vida Life Coach App

Andrew Arrigo, John Eletto, Jenna Ficula, Hunter Postiglione, Jack Ryan

12/12/2018

Table of Contents:

Executive Summary 3

UML Diagram 4

Project Plan 5

Financial Analysis 7

- Pre-Deployment 7
- Post-Deployment 7

ER Diagram 8

IT Requirements 9

User Interface 11

Test Cases 26

User Manual 31

Developer Documentation 31

- Database 31
- Setting up API 33

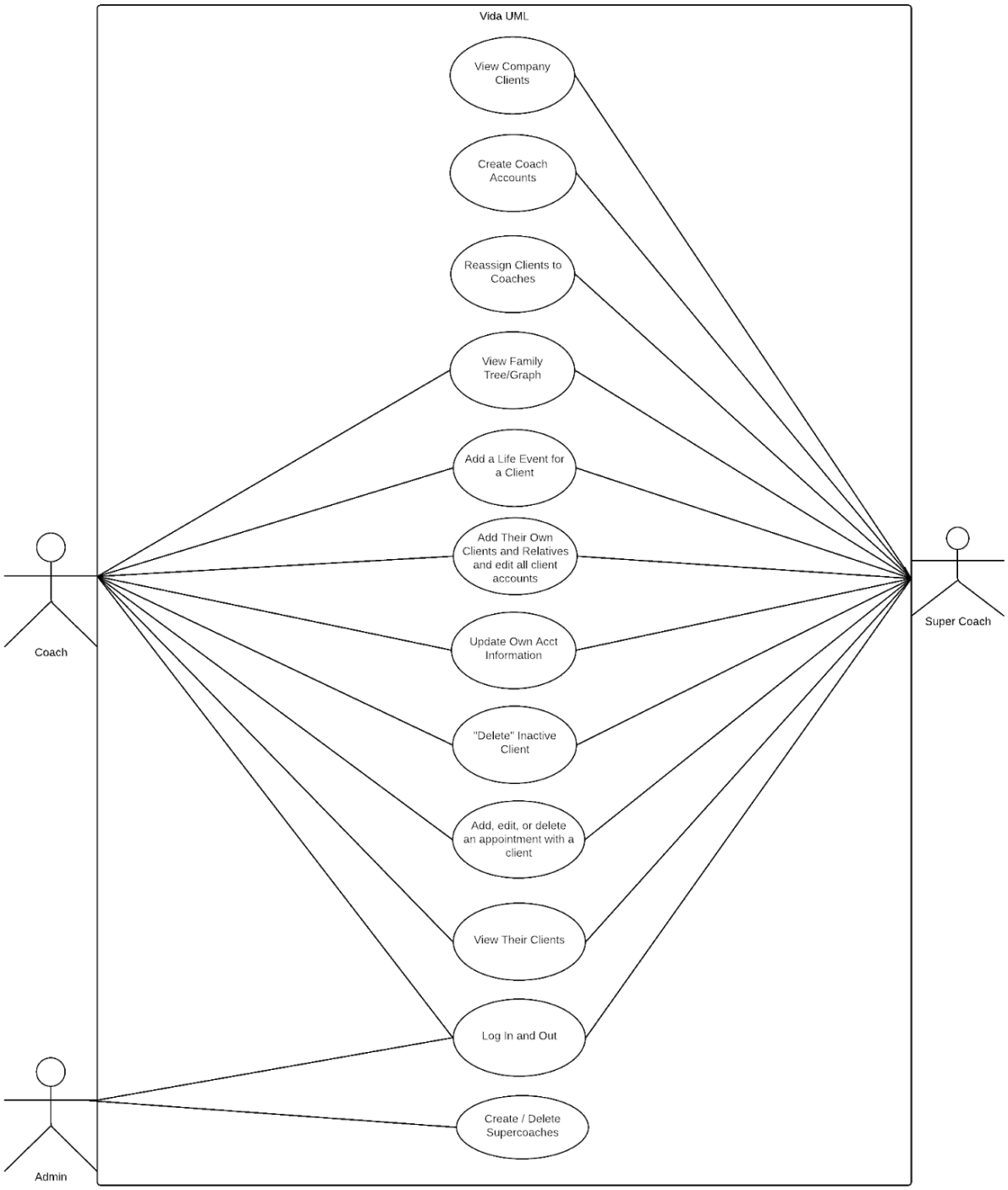
Ethics Essay 36

GitHub Link 36

Executive Summary

In 2015, the life coaching industry was made up of over 53,000 coaches and its market value was estimated to be worth over $1 billion. The task at hand was to develop an iOS application to be used for a life coach company. The company at hand had been using Microsoft Excel spreadsheets to manage client information. The application Vida was developed to manage clients in a more effective manner. Vida allows for coaches to do many of the different tasks such as add clients, set up meetings, and manage information. Due to the fact that some sensitive data will be collected from clients, it was important that security concerns were taken into careful consideration during the development and deployment stages. The Vida app displays a simple, easy, and responsive environment for its users. The ability to use add/edit/remove actions help to make for a manageable experience. Likewise, we have implemented the principle of least privilege in order to give normal coach accounts minimal access to unnecessary resources. Supercoach profiles have the ability to access and manage all coach profiles, along with granting and revoking privileges. The implementation of a Linux, Apache, Postgres, and PHP server allows for a secure environment with many different processes being achieved. Throughout the development process, the database was created with the intention to be scalable and support an growing user base. The Vida app is expected to be licensed and distributed to other coach companies. As the industry is expected to increase in market value and job opportunities, Vida will provide an easy and effective way to manage clients across many different companies at a low enterprise cost.

UML Diagram


Vida UML
View Company Clients
Create Coach Accounts
Reassign Clients to Coaches
View Family Tree/Graph
Add a Life Event for a Client
Add Their Own Clients and Relatives and edit all client accounts
Update Own Acct Information
"Delete" Inactive Client
Add, edit, or delete an appointment with a client
View Their Clients
Log In and Out
Create / Delete Supercoaches
Coach
Admin
Super Coach

Coach

A Coach, upon logging in, is responsible for maintaining the profiles of their clients, as such they have the ability to create/edit/delete these client profiles as well as update their own account information. They may also use the application to create/edit/delete appointments they have made with their clients.

Super Coach

A Super Coach has all of the abilities and permissions of a coach. In addition to all of the coach actions mentioned above, a super coach has the authority to create additional coach accounts, view all company clients from all of the coaches, and assign clients to other coaches when necessary.

Admin

An Admin is responsible for the very basic task of creating the initial Super Coach. Past creating that account, their only responsibility is deleting unnecessary Super Coach accounts. The Admin will not be responsible for any clients or interaction with clients.

Project Plan

9/5/2018	Team Meeting; 1st homework started	Started Homework #1 . Begin to discuss user requirements for the system and write down the main features for the UML Diagram. We began to brainstorm questions to ask the client to ask during class. We wrote down the user requirements on GitHub.	All
9/6/2018	*Second Team Meeting*	*Finalize user requirements and UML Diagram and Project Plan. Also, discuss the database and go over the ER diagram.*	All
9/8/2018	*Complete Homework #1*	*Produce Diagrams (UML and ER)*	Andrew / Jack
9/10/2018	*Third Team Meeting*	*Figure out IT requirements and Finalize ER diagram. Review HW1*	All

9/19/2018	*Create VMs*	*Create the two proposed VMs and download Debian Linux on both. Then download Postgres and Apache*	Andrew / Jack
9/19/2018	Homework #2 Due	E-R diagrams complete with supporting documentation if an explanation is required for anything you document in your diagrams.	Andrew / Jack
9/26/2018	Homework #3 Due	Complete mock-ups (ie. wireframes) of your user interface.	John / Hunter / Jenna
10/3/2018	*Build out Database*	*Create tables and keys. Populate data and query to ensure it works*	Andrew / Jack
10/3/2018	*Team Meeting*	*Start programming the front end of the application - home page and client list page*	John / Hunter / Jenna
	Team Meeting	*Debug and continue to build out the graph for the client relationships*	John / Hunter / Jenna
	Team Meeting	Plan out Graph Implementation; Decide what technology to build it from. D3, etc.	John / Hunter / Jenna
10/10/2018	Database design draft	Find out what should actually go here, the database should be drafted already by then	All
	Front End Dev	*Front end for Coaches built out. Rough demo of the graph with static data is in place. Mostly the view screens are created but there are no real form screens at this point*	John / Hunter / Jenna
10/17/2018	Demo	Complete initial demo	All
10/24/2018	Front End Dev	Profile Screen Completed and Models built out for connection to DB	John
10/31/2018	Connecting Front and Back End	Cleaned up front end UI. Began to build back-end API. Back-end API Authentication using JWT. Client List connected to back-end.	All
11/7/2018	First Demo of Prototype	Complete project prototype first-pass demo ready. The coach can now add and edit clients. The graph pulls data from the database but it is not yet displayed. Meetings are linked to coaches but you cannot add and update them yet	John / Hunter / Jenna
	Mid Week Updates	The user can now change their password and email credentials	John
11/28/2018	Draft Documentation	All documentation required for the project in draft form	All
11/28/2018	Mid Week Updates	Supercoach capabilities completed; Graph loads data from the back end	All
	Database Moved	Database moved over to the production server	Jack

	Functionality Added	Notes added to profiles, Adding family members of clients, Graph is completed	All
12/8/2018	Bugs Fixed	Most bugs figured out and all screens fully functional	All
12/12/2018	FINAL PRESENTATION!	Presentation and Final Paper Due	All

Financial Analysis

Pre-Deployment

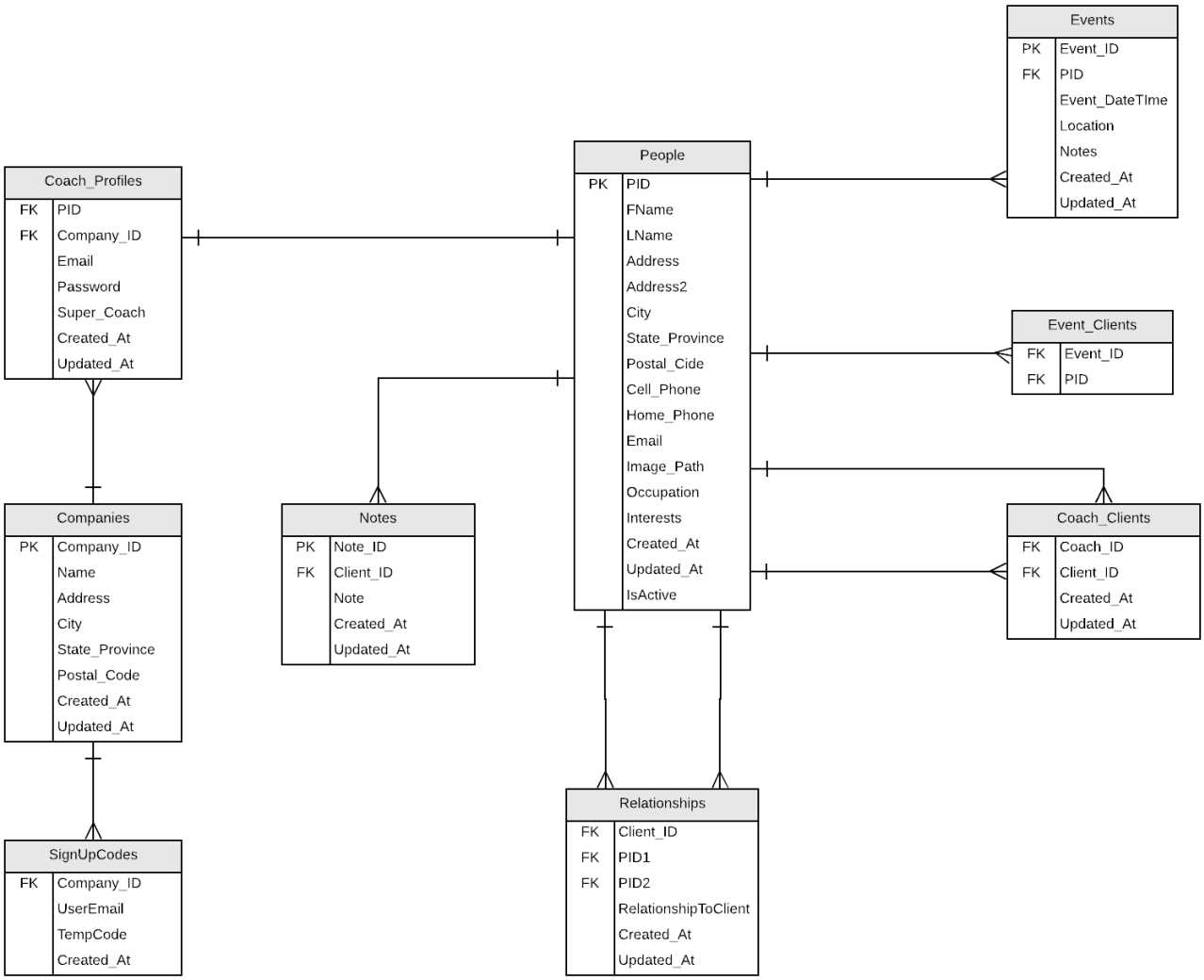
Hourly Software Development Rate:	$38
Estimated Hours Per Team Member	180
Labor Rate:	$34,200
iOS Developer License:	$100
Server Deployment (Pre-Purchase):	$45 ($15/month)
Total:	**$34,345**

Post-Deployment

Server Deployment:	$180/year**
iOS Developer License:	$100/year
Back End Admin:	$600/year
Total:	**$880/year**

**This is based on the assumption of only one small company, with less than 50 employees, using the system. Scaling up based on need.

ER Diagram


Events
PK Event_ID
FK PID
Event_DateTIme
Location
Notes
Created_At
Updated_At
People
PK PID
FName
LName
Address
Address2
City
State_Province
Postal_Cide
Cell_Phone
Home_Phone
Email
Image_Path
Occupation
Interests
Created_At
Updated_At
IsActive
Coach_Profiles
FK PID
FK Company_ID
Email
Password
Super_Coach
Created_At
Updated_At
Event_Clients
FK Event_ID
FK PID
Companies
PK Company_ID
Name
Address
City
State_Province
Postal_Code
Created_At
Updated_At
Notes
PK Note_ID
FK Client_ID
Note
Created_At
Updated_At
Coach_Clients
FK Coach_ID
FK Client_ID
Created_At
Updated_At
SignUpCodes
FK Company_ID
UserEmail
TempCode
Created_At
Relationships
FK Client_ID
FK PID1
FK PID2
RelationshipToClient
Created_At
Updated_At

IT Requirements

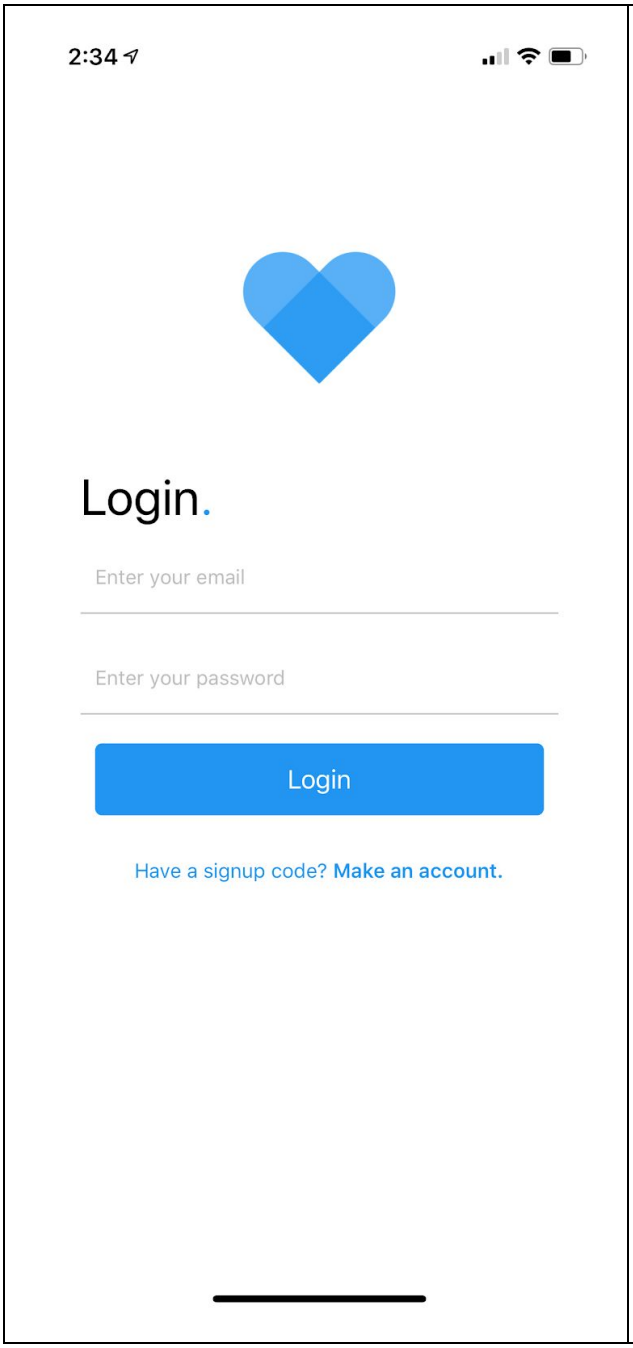
1. Application Server
    1.1. Physical Requirements
        1.1.1. 2 Gigabytes of Memory and 30 Gigabytes of Hard Disk
        1.1.2. Once the App goes live and users increase, a cluster of servers will be put together along with proper load balancers in order to not overwhelm the system
        1.1.3. A database designed to support multiple companies along the back-end
    1.2. Virtual System Requirements
        1.2.1. Linux: Ubuntu
        1.2.2. One backup image of the application
    1.3. Connectivity
        1.3.1. Will have a static IP assigned to this server
        1.3.2. This server will be connected on the back end to the Main Database Server and the internet
2. Postgres Server
    2.1. Physical Requirements
        2.1.1. 2 Gigabytes of memory and 30 Gigabytes of Hard Disk
        2.1.2. Once the number of users and data increases, there will be multiple data centers in order to handle all of the new data
    2.2. Virtual System Requirements
        2.2.1. Linux: Ubuntu
        2.2.2. One image per day on a separate server, each day the prior image is deleted
    2.3. Connectivity
        2.3.1. The database will have a static IP assigned to it
        2.3.2. The server will only be connected to the web server and also the backup server
3. Reliability
    3.1. Service Level Agreements
        3.1.1. Hosted privately through IBM Mainframe allows for 99.999% uptime
        3.1.2. A response time of less than 20ms
4. Recoverability
    4.1. The Database will be backed up weekly to another server that it is connected to.
    4.2. The only person with access to the backup database would be a system admin. If the main database goes down, there will be the option to switch the back end to the backup while the primary is recovered.

4.3. All of the data stored in the database needs to be stored long term. In order to comply with federal and state regulations and also for the system to be able to go back in time for each user.

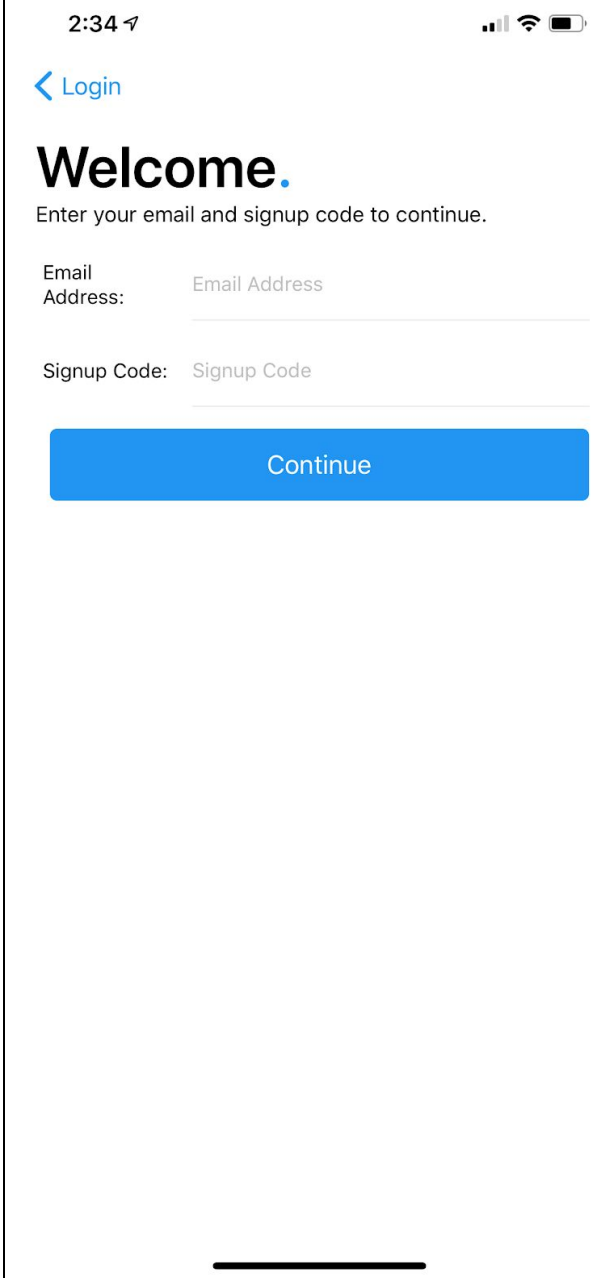
5. Security and Privacy
    5.1. Database
        5.1.1. The Admin can only add and update the Coach profiles in the database. The Super-Coach can update and view coach profiles, view and update client profiles and everything in the database. For Coaches, they can update and access their own coach profile along with their clients' information.
    5.2. Account Information
        5.2.1. User data
            5.2.1.1. Personal data saved to Postgresql database and will be protected through private web hosting. Registered clients information will only be accessible to a client's assigned coach. No other coach has the ability to view an unassigned clients information.
            5.2.1.2. Communication through API provides HTTPS protection
            5.2.1.3. Saved information will consist of notes, events, basic personal information, and any information disclosed clients.
        5.2.2. Privacy Considerations
            5.2.2.1. Client/coach information stored may contain sensitive data thus making it necessary for encryption on the backend of the database.
            5.2.2.2. Using Let's Encrypt to provide certificates for encryption
    5.3. Admin Access Controls
        5.3.1. The original Admin only needs to create a single SuperCoach. After that, there is a single Admin user to maintain the database. For adding and removing users after that, the Super-Coach creates other Super-Coaches and also Coaches. Then Coaches create their client accounts.

6. Maintenance
    6.1. Planned downtime requirements
        6.1.1. Database maintenance will occur bi-monthly to keep up-to-date with new clients entering the system and current clients leaving the program
        6.1.2. Maintenance will be done between 12AM-5AM, once every four months (Seasonally)
        6.1.3. The system should be up and running year round

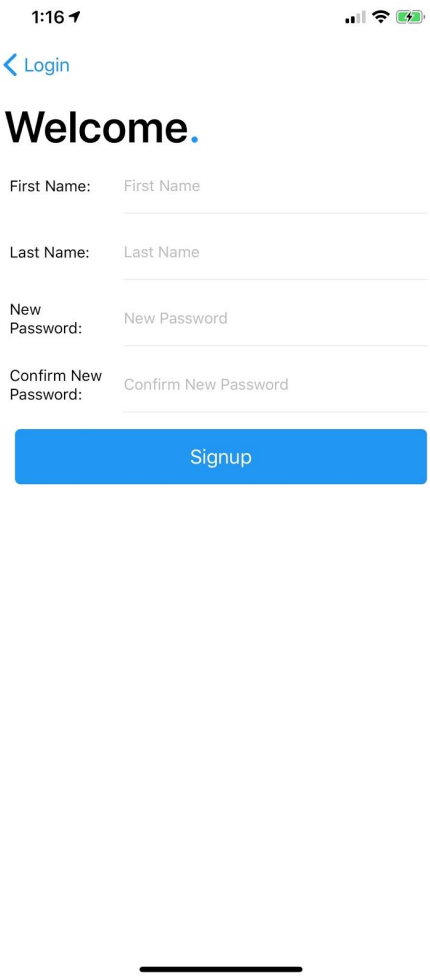
User Interface



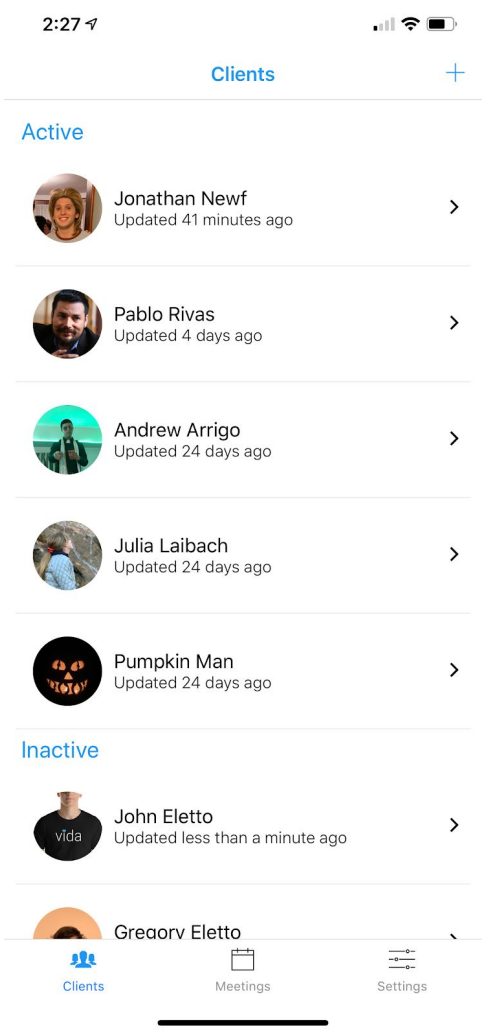
This is the first screen you come upon when entering the app. It prompts the user to enter their email along with their password. Then, the user clicks “Login” to get into the app. If it is a user’s first time logging in, they press the “Make an account” text below the “Login” button.



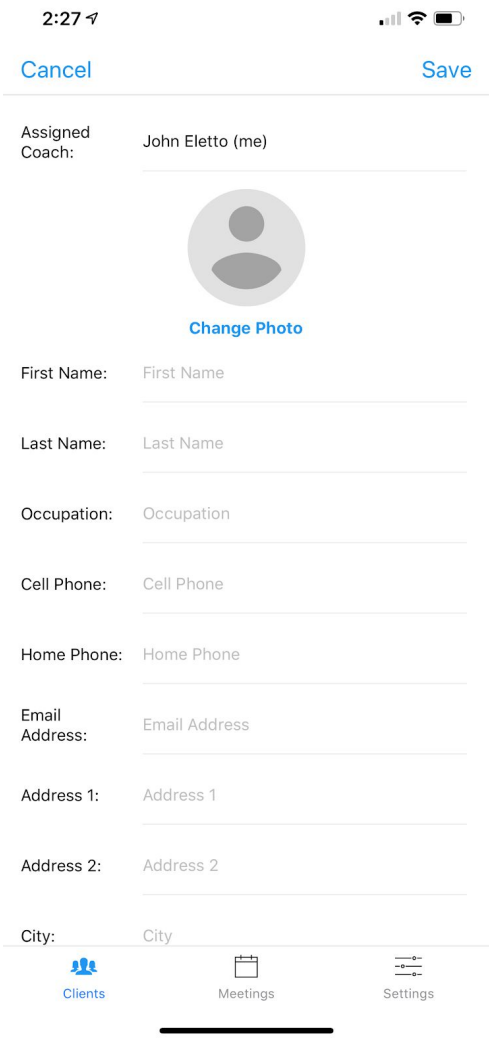
This is the screen seen by each coach logging in for the first time. It prompts the coach to provide their email address along with the signup code that was sent to them via email by a super coach in the company. A coach then presses “Continue” to proceed to the page for them to enter their information.



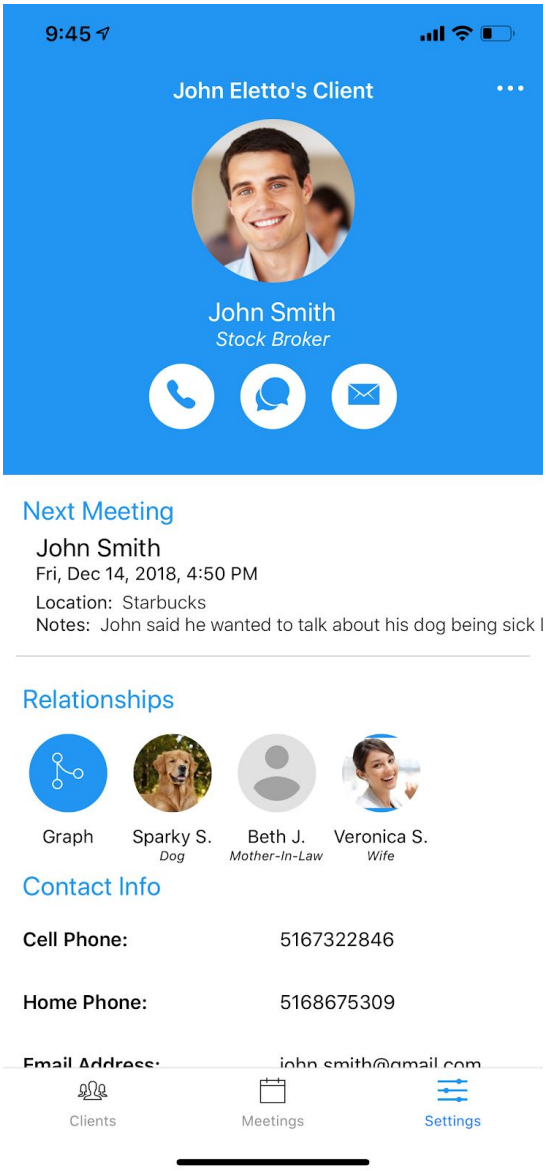
This is the screen for when a coach is first creating their account after their email and activation code is verified. It allows a user to edit his/her first and last name. Then, it allows the coach to change their password by entering a new password and confirming it.



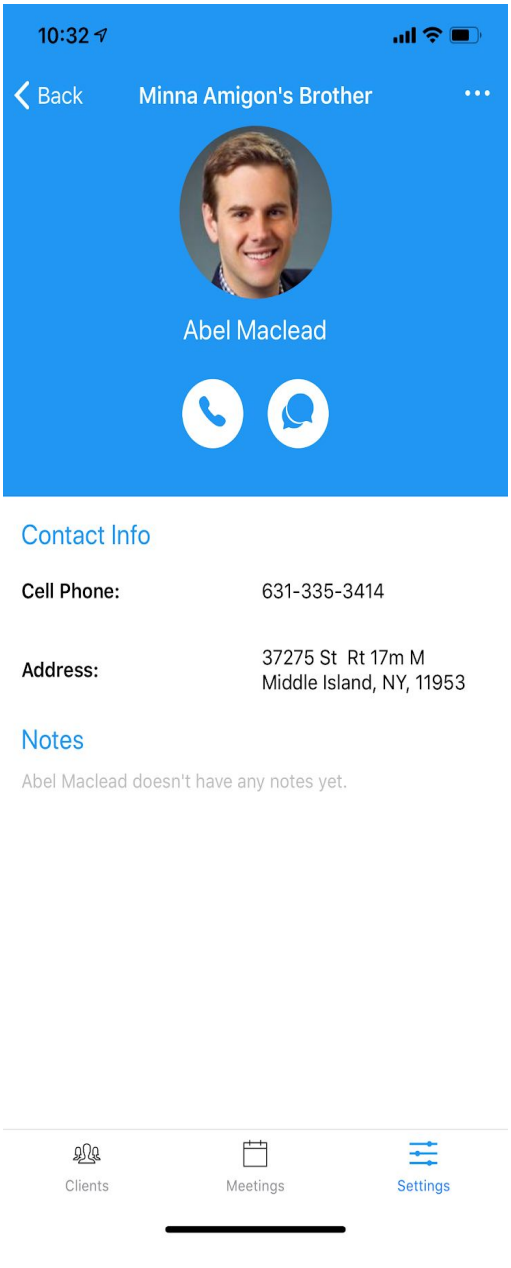
This is the first screen you are given when logging into the app as a coach. There is a list sorted with active clients first, then inactive clients underneath. Both lists are sorted by the last interaction with their profile. For each client, their full name, a small thumbnail picture, and the date or time that you last changed their profile is shown. You can click on the “+” button on top of the screen to add another client. Then if you click on a profile, it brings you to their individual profile.



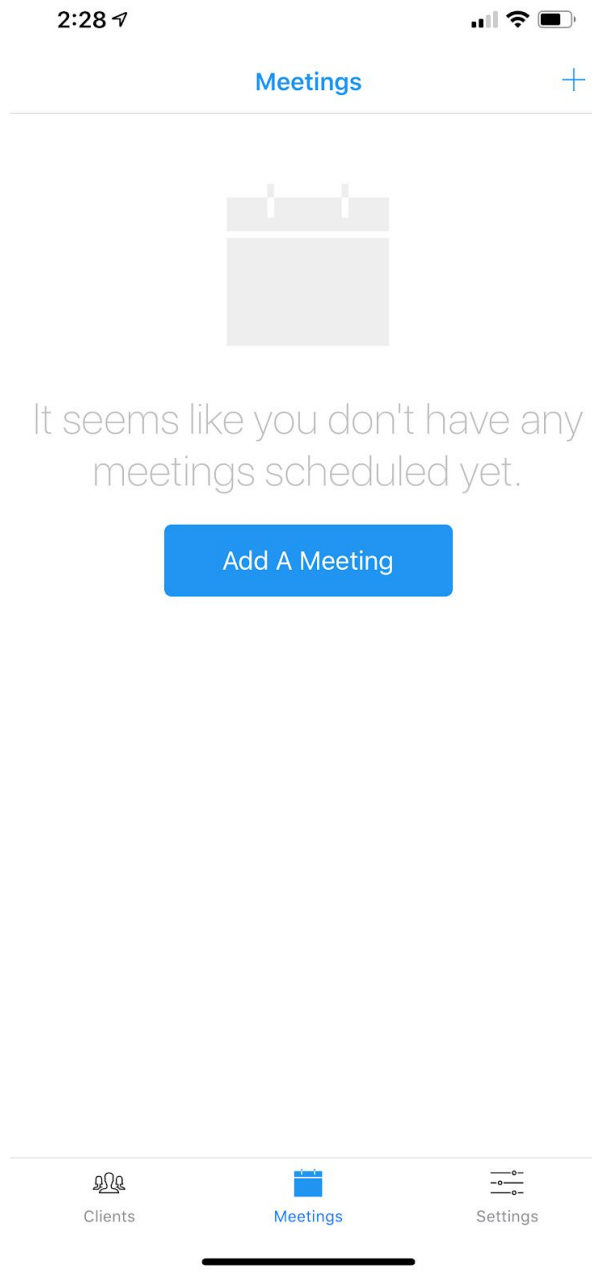
This screen is for when a coach wants to add a client. This particular screen, if you are a super coach, also allows you to assign the client to a coach in your company. It allows you to choose a picture for this client from the camera or photo roll and enter all of the necessary information for the new client.



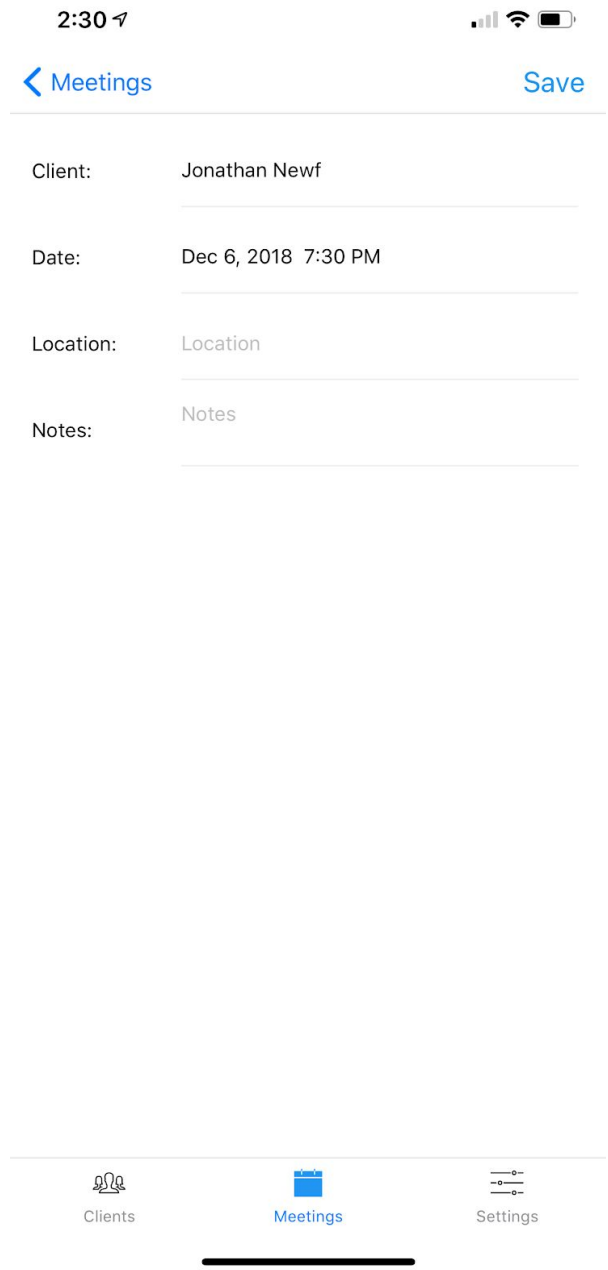
This is the screen to view clients once they have been created. If you are a super coach viewing a coach's clients, you can see whos client it is in the header. It shows a picture, clients name, and their profession, along with quick buttons to call, text, and email if available. The navigation button in the top right allows you to manipulate the client by editing, marking inactive, or adding a relationship. The next meeting object shows if you have any meetings with them. Relationships show family members of clients and also a button to view the graph. Then at the bottom, there is contact information and notes associated with the client available.



This screen for family members of clients is very similar to the main client screen. It says in the header who the main client is and how they're related (ex: Minna Amigon's Brother). It only does not contain the upcoming meetings tab and the relationships tab.

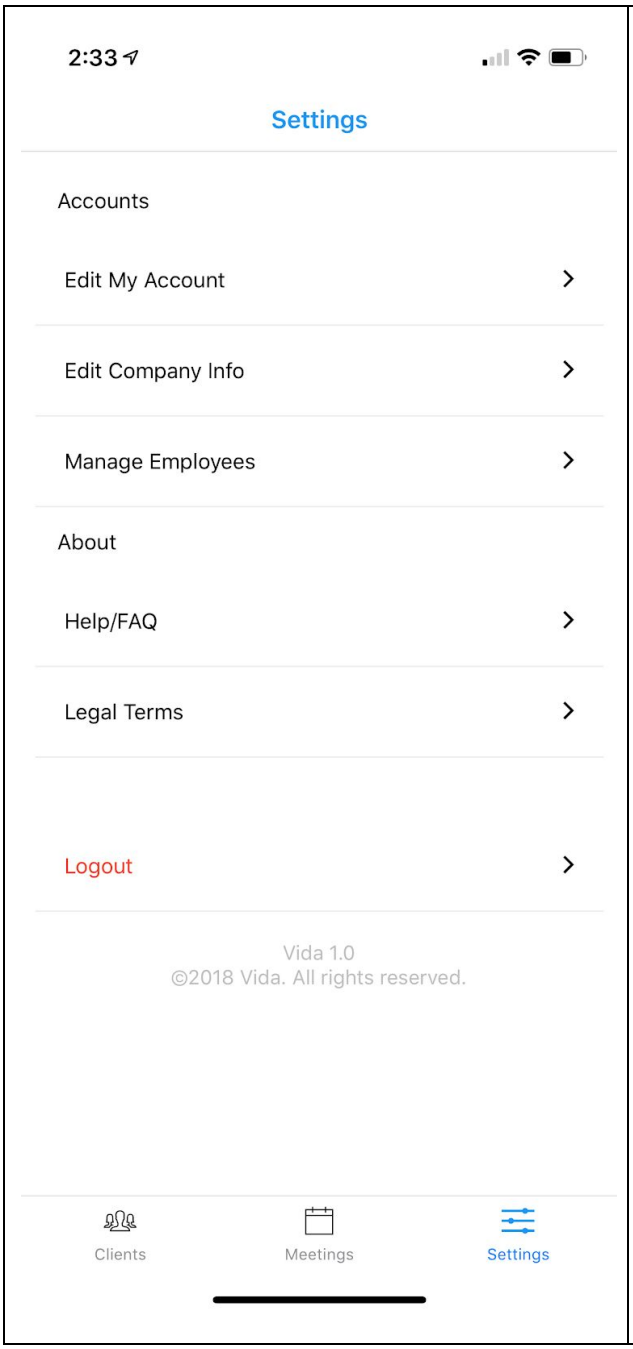


This screen comes up when you use the bottom navigation and click the meetings screen, but do not have a meeting to show. The “+” on the top right corner or the “Add A Meeting” in the center screen allows a coach to add another meeting. The icon on the top left is to toggle between future and past meetings in order to keep track of past interactions.

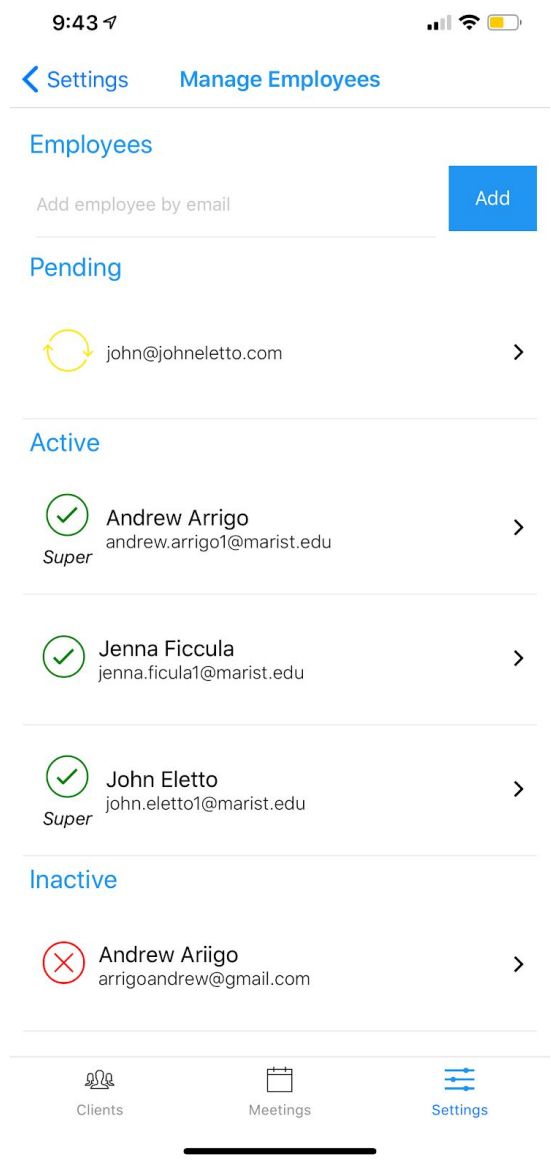


This screen is for a coach to add or update their meetings. The client field is a drop down based on the clients that are yours and the date is also a drop down. The location field is an open text field that allows you to put in any notes you have before, during, or after the meeting is over.

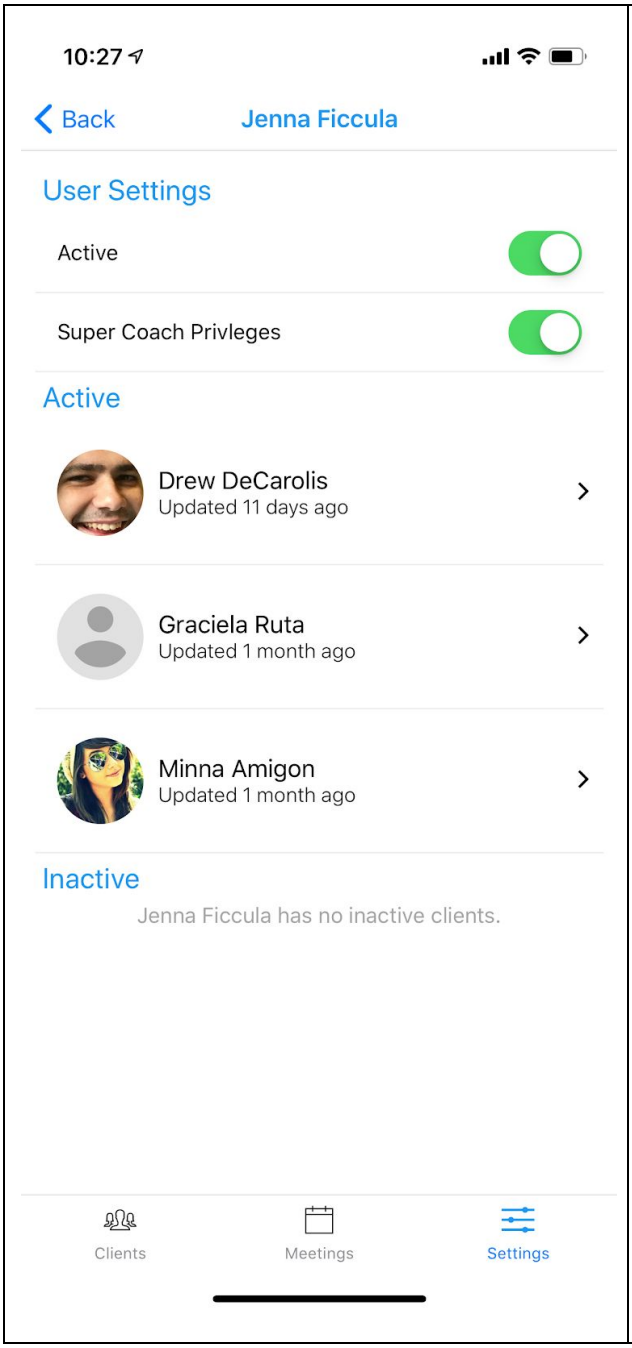
This is what the meetings screen looks like with data populated. It displays the meetings, from the one that is coming next down to the one furthest away. It provides the client, date and time, location and then any notes that were provided with the meeting.



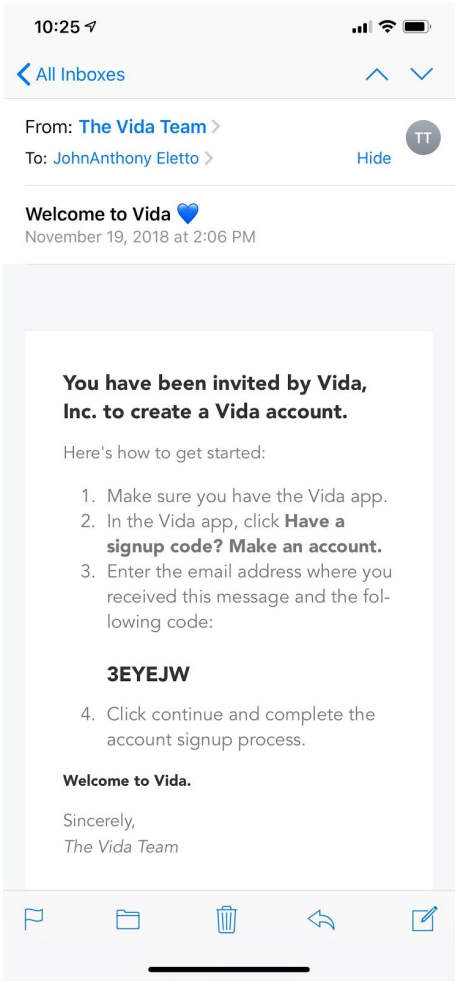
This is the settings page when you click settings in the bottom navigation. It allows you to edit your account and also manage employees if you are a super coach. There is also a log out button at the bottom of the screen.



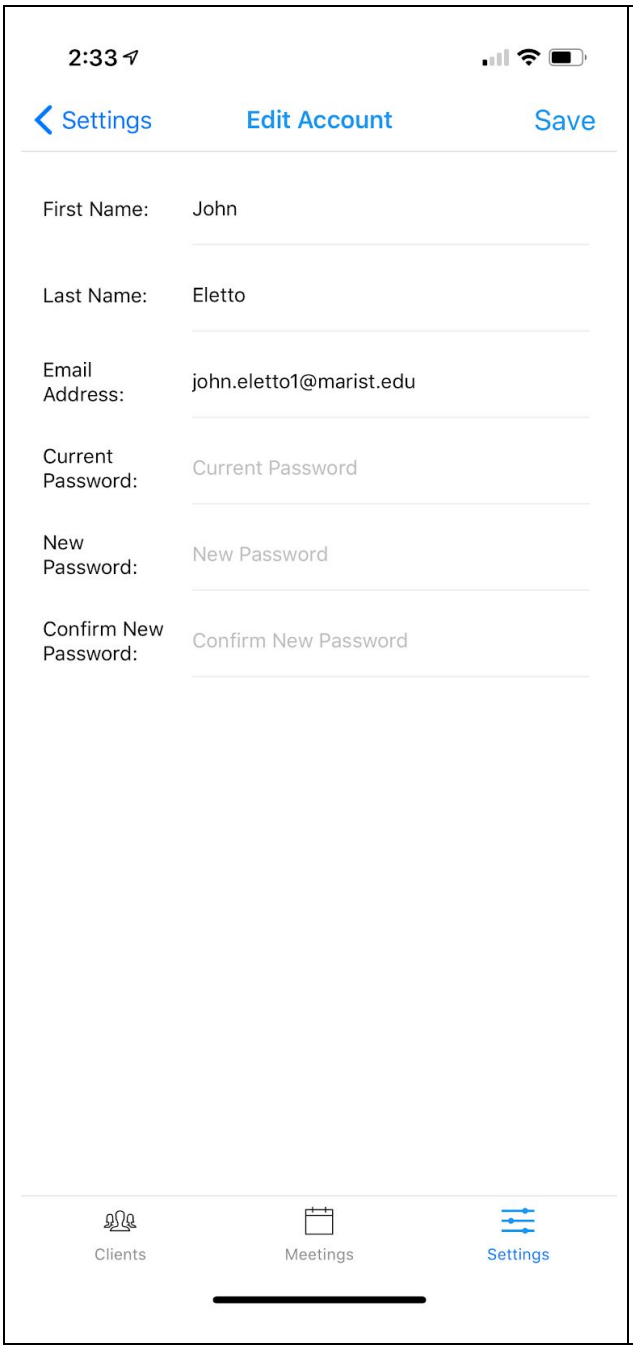
This screen is for super coaches to manage their employees. You are able to add an employee by supplying their email on top and then hitting add. The rest of the screen is dedicated to pending employees, active employees, and then inactive employees. For active employees, there is the word “Super” under their green check to indicate that they are also a super coach. Each employee is clickable to manage their account.



This is the screen a super coach is given when clicking on a coach's account. It allows them to mark a coach active or inactive and also give or revoke super coach capabilities. Then, there is a list of clients that allow you to manage their clients just as they can.



This is the email a coach receives when they are invited to the app. It provides instructions along with a code to access their account.



This is the screen a coach sees when editing their profile. It allows them to change their basic information along with changing their password.

Test Cases

- Most test cases assume you have logged in to the application already.
- This **does not** include **TC01** and **TC05** which involve logging in, and creating an account.
- **TC02 & TC03** require you to be logged in as a Super Coach as only they have the authorization to do those tasks.

Super Coach Credentials

Username: john.ryan4@marist.edu
Password: Pablo1960!

TC01: Login

The user should be able to login with their username & password.
(If you're using our database you can use the ones provided above, otherwise, you'll have to follow the steps in the documentation to create an account in the database that has super coach authorization.)

Steps:

1. Input your username into the username field
2. Input your password into the password field
3. Press login

Expected Result: You will be logged in and brought to the client page.

TC02: Edit Company

The user should be able to modify their company name. Requires Super Coach account.

Steps:

1. Navigate to the settings page (rightmost selection in bottom navigation).
2. Tap "Edit Company"
3. Change your company name
4. Press "Save"

Expected Result: New company name will be saved, when going back to "Edit Company" you will see the new name.

TC03: Invite Coach

The user should be able to invite other coaches to join the application. Super coach status required.

Steps:

1. Navigate to the settings page (rightmost selection in bottom navigation).
2. Tap “Manage Employees”
3. Input an email that you have access to and would like to use as the coach account’s email into the email field.
4. Press “Add”

Expected Result: A login code will be sent to the supplied coach’s email along with a prompt to create their account. They will show up in pending until they have done so.

TC04: Logout

The user should be able to log out.

Steps:

1. Navigate to the settings page (rightmost selection in bottom navigation).
2. Press “Logout”

Expected Result: You will be logged out and brought to the login screen.

TC05: Create Account

A coach invited to the platform should be able to create an account.

Steps:

1. After receiving an email to create your coach account press “Make an Account”
2. Enter the email you received the code on into the email field
3. Enter the code contained in the email into the code field.
4. Press “Continue”
5. Enter in your first name, Ex. “John”
6. Enter in your last name, Ex. “Smith”
7. Enter in your password, Ex. “Password!321”
8. Re-enter in your password from the last field
9. Press “Signup”

Expected Result: You will be logged in using your new account and brought to the landing page(Clients page).

TC06: Add a Client

The user should be able to add a client with details/photo

Steps:

1. Press the “+” on the top right of the client list screen
2. Fill out the First name field with your “client’s” first name, ex. “Richard”
3. Fill out the Last name field with your “client’s” last name, ex. “Tompkins”
4. Press the “Change Photo”
5. Select “Take a photo” **OR** “Select from Camera Roll”
6. Take a picture and accept it **OR** Choose a picture and accept it
7. These are the only required fields so now you may hit “Save”

Expected Result: After hitting save you should be brought to a client profile page containing the information for the new client.

TC07: Update a Client

The user should be able to update a client with details/photo.

Steps:

1. From the list of your clients choose one by tapping it.
2. Tap the “... ‘ on the top right of the client profile, and choose “Edit Client” from the options
3. Add in the Occupation “Application Developer” or any other position you want.
4. Press “Save”

Expected Result: After hitting save a refreshed client profile page will contain the updated information for the client.

TC08: Add a relationship to a Client

The user should be able to add a relationship to the client or one of the client’s relations. Ex. You can add a mother to the client, but also a sister to that mother after she has been created.

Steps:

1. From the list of your clients choose one, create one if none exist.
2. After pressing a client, tap the “... ‘ on the top right, and choose “Add Relationship”
3. Tap the “Connecting Relative” field and choose a name to serve as the other half of the relationship pair. If there are no other relations yet then it will just be the main client.
4. Supply a first name for the new Relation, Ex. “Janice”
5. Supply a last name for the new Relation, Ex. “Baldwin”
6. Name the relation that the two share, Ex. “Mother”
7. Press “Save”

Expected Result: After hitting save when looking at a refreshed client profile page it will now show “Janice Baldwin” or whatever name you have chosen in the relations field.

TC09: Add a note to a Client

The user should be able to add a note to a client.

Steps:

1. From the list of your clients choose one.
2. After pressing a client, tap the "... ' on the top right, and choose "Add A Note"
3. Fill out some text for the note, Ex. "This client has been getting defensive when asking about the progress with meditation he has made, consider alternative methods".
4. Press "Save"

Expected Result: Hitting save saves the note, you will now be able to view the note when looking at the client page.

TC10: Delete a Note

The user should be able to delete a note they have with on a client. Having a client with a note is required.

Steps:

1. From the list of your clients choose the one you have a note on.
2. Press "See All Notes"
3. Swipe a note of your choice to the left and select delete.

Expected Result: Note should now be deleted.

TC11: Add a new meeting.

The user should be able to add a meeting with a client. Having a client is **required**.

Steps:

1. Navigate to the meetings page (Middle button in bottom navigation).
2. Press either the "Add a Meeting" in the center of the page(Won't show if there are already meetings) or the "+" on the top right.
3. Tap Client and choose what client you want for your meeting. Ex. Richard Tompkins
4. Tap Date to open the date picker and choose when you want to schedule the meeting. Ex. Dec 25 4:06 PM
5. Optionally fill in the Notes/Location fields, not required for a meeting.
6. Press "Save" on the top right.

Expected Result: After hitting save you should be brought back to the meetings page and there will be your meeting.

TC12: Update a meeting.

The user should be able to update a meeting they have with a client. Having a meeting is **required**.

Steps:

1. Navigate to the meetings page (Middle button in bottom navigation).
2. Choose a meeting that you have with a client, and tap it.
3. Modify the Location to something new. Ex. “Starbucks”
4. Hit “Save”

Expected Result: After hitting save you should be brought back to the meetings page and there will be your updated meeting.

TC13: Delete a meeting.

The user should be able to delete a meeting they have with a client. Having a meeting is **required**.

Steps:

1. Navigate to the meetings page (Middle button in bottom navigation).
2. Choose a meeting and tap it.
3. Press “Delete” at the bottom of the meeting.
4. Confirm that you want to delete the event when prompted.

Expected Result: After hitting confirming via the alert you will be brought back to the meetings page and your meeting will no longer be there.

TC14: Mark a Client Inactive

The user should be able to set a client’s status to inactive.

Steps:

1. From the list of your clients choose one.
2. After pressing a client, tap the “... ‘ on the top right, choose “Mark Inactive”

Expected Result: Now when looking at the client list the previously selected client will be in the “Inactive” section.

TC15: Edit Account

The user should be able to modify their account details.

Steps:

1. Navigate to the settings page (rightmost selection in bottom navigation).
2. Tap “Edit Account”
3. Change your last name to “Arrigo”
4. “Save”

Expected Result: Account details will be saved, when going back to “Edit Account” you will see that your last name is Arrigo.

TC16: View Graph

The user should be able to view a graph that displays the relationships a client has.

Steps:

1. From the list of your clients choose one with at least one relationship
2. In line with the relationships there is a button representing the graph, press it. Should be labeled on the graph.

Expected Result: You will be brought to a screen that shows the relationships of the client.

User Manual

In order to get the app running in production, you will have to follow the following steps:

1. Go to the App Store and download Expo Client
2. Once downloaded, click *Sign in to your account*
    a. Username: team@getvida.app
    b. Password: Marist2019!
3. Once logged in, click Vida under the published projects
4. Use the following credentials to log into the app
    a. Username john.ryan4@marist.edu
    b. Password: Pablo1960!

Developer Documentation

Database

1. `Open SSH Connection to the CLI`
2. `Issue Commands:`
    a. **`Ifconfig`**

i. This will provide the IP address on the second line, after the word "inet"

b. Sudo ufw allow ssh

c. Sudo ufw allow 5432

d. Sudo ufw allow 80

e. Sudo ufw allow 443

f. Sudo ufw enable

i. Type "y" when prompted

g. NOTE: if there is a firewall enabled in the Google Cloud Platform, you must ensure there is a rule to enable incoming traffic to port 5432

h. sudo apt update

i. sudo apt install postgresql postgresql-contrib

3. When Prompted, type "Y"
4. In the VM Command line do the following:
   a. Cd /etc/postgresql/10/main
   b. Sudo nano postgresql.conf
      i. At the bottom of the file, change the listen address to show: listen_address = '*'
      ii. Write out then save the file and exit
   c. Sudo nano pg_hba.conf
      i. At the end of the file, add the following line: host all all 0.0.0.0/0 md5
      ii. Write out then save the file and exit
   d. Sudo nano /etc/init.d/postgresql restart
5. Issue Commands:
   a. sudo -u postgres psql
   b. create database VidaDB
   c. \password postgres
   d. *Enter password twice*
6. Go to your local machine
   a. Download PGAdmin 4
   b. Documentation provided in the link below based on your operating system
   c. https://www.pgadmin.org/download/
7. Once downloaded, open PGAdmin
   a. Click "Add A New Server"
   b. Type a name for the server, Such as "VidaDB"
   c. Go to the connections tab
   d. In the Hostname Field, type in the ip address of the Virtual machine

e. In the password field, use the password you changed the postgres user in step 5c-d
f. Click Connect

8. Select "VidaDB" in PGAdmin
    a. Right click on "Databases" and then hit "Create"
    b. Give a name for the database, use "VidaDB" and leave the owner as postgres
    c. Hit the "Save" button
    d. Right click "VidaDB" under the databases tab
    e. Select Restore
        i. Select File VidaDatabaseBackup.psql from our GitHub repository found here:
            1. https://github.com/johnanthonyeletto/Vida-app/blob/master/VidaBackupDev.psql
        ii. Hit Select and then change the Role Name to say postgres
        iii. Then hit Restore
            1. Note: If there are any errors, ignore them
    f. Right Click Login/Group Roles under the Server
        i. Click "Create" then 'Login/Group Role"
        ii. Name should be "App"
        iii. Under the definition tab, make the password "Marist2019"
        iv. Under Privileges, enable the "Can Login?" option
        v. Hit "Save"
    g. Right Click the "VidaDB" Database and click "Grant Wizard"
        i. Select all of the rows by clicking the box on the top left corner and hit "Next"
        ii. Click the "+" button and then select the App user under the grantee tab
        iii. Under privileges, click "Select", "Update", "Insert", and "Delete" amd then click "Next" and then "Finish"

Setting up API

1. Issue the following commands in the command line:
    a. **Sudo apt-get update**
    b. **Sudo apt-get install apache2**
        i. When prompted, type "y"

2. Go to your browser and type in the external ip of the machine to ensure Apache installed correctly
    a. You should see a Apache2 Default Page
3. Type the following commands:
    a. Cd /var/www
    b. **Sudo git clone** https://github.com/johnanthonyeletto/Vida-app.git
        i. Type in your github username and password
    c. Cd
    d. sudo apt-get -y install php7.2 libapache2-mod-php7.2
    e. sudo service apache2 restart
    f. cd /var/www/Vida-app
    g. sudo chmod 755 -R api
    h. sudo chmod -R o+w api/storage
    i. cd /etc/php/7.2/apache2/
    j. sudo nano php.ini
    k. Remove the semicolon in front of the line "extension=pdo_pgsql"
    l. Write out then save the file and exit
    m. sudo service apache2 restart
    n. Cd /etc/apache2/sites-available
    o. sudo nano 000-default.conf
    p. Change the documentRoot to read: "DocumentRoot /var/www/Vida-app/api/public"
    q. Add: <Directory /var/www/Vida-app/api/public>
                Options Indexes FollowSymLinks
                AllowOverride All
                Require all granted
            </Directory>
        i. Write out then save the file and exit
    r. sudo service apache2 restart
    s. cd /var/www/Vida-app/api
    t. Sudo cp .env.example .env
    u. sudo nano .env
        i. Erase all contents of this file and then replace it with:

```
APP_ENV=production
APP_DEBUG=true
APP_KEY=base64:Pf8bqFhStKix4rtdt1FOgvPpudMqsYyw+Uv+dOvaF8w=
APP_TIMEZONE=UTC

LOG_CHANNEL=stack
```

```
LOG_SLACK_WEBHOOK_URL=

DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=5432
DB_DATABASE=VidaDB
DB_USERNAME=App
DB_PASSWORD=Marist2019

CACHE_DRIVER=file
QUEUE_DRIVER=sync

JWT_SECRET=076CE1F76B7457DB274C19BDA2B0F8660752614ADE7CF5C7B9D7E15C360E3B55

MAIL_DRIVER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_ENCRYPTION=tls
MAIL_USERNAME=johnanthony.eletto@gmail.com
MAIL_PASSWORD=pcdylgcpbczayrit
MAIL_FROM_ADDRESS=team@getvida.app
MAIL_FROM_NAME="The Vida Team"
```

1. Write out then save the file and exit

v. Cd

w. Sudo apt update

x. sudo apt install curl php-cli php-mbstring git unzip

i. When prompted, enter "y"

y. Cd ~

z. curl -sS https://getcomposer.org/installer -o composer-setup.php

aa. sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer

bb. Composer

i. To verify that composer installed

cc. Cd /var/www/Vida-app/api

dd. Sudo composer install

ee. sudo apt-get install php-mbstring php-dom

i. When prompted, enter "y"

ff. Sudo composer update

4. Go to the browser on your local machine and enter the external IP for the VM. It should read:

Welcome To The Vida API!

Please Follow This Documentation to set up Laravel/Lumen on Apache.
https://www.howtoforge.com/tutorial/install-laravel-on-ubuntu-for-apache/

Please Follow This Documentation to set up SSL on Apache.
https://certbot.eff.org/lets-encrypt/ubuntubionic-apache

Ethics Essay

Vida is a life coach application designed to solve the problem of a life coach agency needing to improve management of their client database and information. Vida allows life coaches to manage clients, meetings, and also other employees through a seamless interface. With all of this functionality, it becomes increasingly difficult to manage the data for the clients, coaches, and companies. The life coach can input all the data for clients necessary to create a full and complete profile. Client profiles can contain sensitive information of the goals, needs, and family relationships. Life coaches coach not only the client, but members of the family expanding the reach of data to other people. Since the application acts as a customer database, the constant collection of big data leads to countless ethical implications of maintaining and protecting client information. One of the key components of this application is the importance of the high profile and wealthy clients that are part of the organization receiving life coaching. These clients have sensitive personal information that is being disclosed to Vida life coaches and being inputted into our database systems. Although throughout the technology industry there are numerous ethical concerns regarding freedom of speech and jobs and labor issues, the data security and privacy ethical issues are the foundation of ethical concerns for Vida.

Since the Vida app upon initial development and deployment will be used primarily in the United States, this paper will seek to investigate the legislation applicable to United States Data Privacy Policies. As of December 2018, there is no single, comprehensive federal law regulating the collection and use of personal data by a private business or federal government. The United States has currently established a series of federal and state laws supplemented by industry standards and best practices for self regulatory data privacy procedures. Specifically, the two most regulated digital data privacy industries are the financial and healthcare industries using

Acts such as the Federal Trade Commission Act and Health Information Portability and Accountability Act. No such legislation or standards exist for the technology industry. Despite the combination of multiple policies regarding the collection, protection, and distribution of personal data the Pew Research Center conducted a study on Data Security and found that out of 1040 American adults surveyed in Spring 2018, 64% experienced a major data breach and 49% feel their personal information is less secure that it was 5 years prior to the survey. Only 12% of study participants have a high level of confidence that social media companies, governments, or technology entities can keep their information safe.

With these mounting societal concerns of data security, Vida is in a vulnerable ethical position since the company maintains possession of data from high net worth and influential individual clients. Therefore, it is important to ensure that all of the client data is secure. Such information can include possible family financial information, substance abuse issues, marital or family related problems, nondisclosed professional information about a clients work life etc. Security measures and data privacy protective measures such as distinction in permissions for super coaches and clients have been put in place to combat the loss of valuable information. It is imperative that Vida focuses on the reputation of the company and value it is providing to its stakeholders without the misuse and mishandling of user data. Vida seeks to build trust among the coaches and clients to obtain informed consent and ensure the proper handling of client data through transparent data policies and regulatory measures.

One of the biggest ethical issues in the United States technology industry has come to surface in recent months is with the Facebook data breach. Facebook's Mark Zuckerberg appearing before the United States Congress to address his company's involvement in the harvesting and improper distribution of approximately 87 million Facebook profiles. Harvesting user data and profiting of distributing data to third party vendors has become one of the most

debated ethical issues in the technology industry as of late. There is often a lack of candor with users as to where their public data on large scale big application platforms ends up. It is important that Vida places an emphasis on ethical data management and informs the clients about what is being done with their data. Although it has been proposed that stronger legislative and industry regulations should prevent this kind of unethical distribution of data, it is unreasonable to believe that these regulations will be strictly adhered to and nationally adopted. It is the ethical obligation of Vida to actively engage to fully protect client data. Despite the fact that even if it is possible to describe all of the ways data might be used, it is difficult to expect participants to fully understand. Clients should be educated on their data privacy rights to enable trust to build between the Vida brand and the user.

In order to combat these concerns of data privacy protection and sustain ethical business practices, it is important that Vida maintain transparency, establish appropriate use guidelines, and provide recourse and control for clients. These objectives can be obtained through first establishing a review committee guided by the Vida Mission and corporate stakeholders to approve proposed data policies. Vida can maintain transparency by establishing policies for collection and use of client data that is understandable, enforceable and continuously maintained. Additionally, it is important to clearly state what clients may expect Vida to use their private information to provide a service of life coaching and to serve the client needs. Guidelines can be established to protect the details of the client and create a framework to ensure data is ethically used and not at risk for manipulation or exploitation. Finally it is essential Vida provides clients permissions to control and understand the collection and use of their personal data. There should be clear opt out policies and provided means of recourse if they feel their data is being used inappropriately. This will provide Vida clients with the

confidence in the Vida App and to a certain extent more control over their privacy as to maintain these ethical data practices.

References

Barnes, S. B. (2006). A privacy paradox: Social networking in the United States. First Monday, 11(9).

Bollier, D. (2010) 'The promise and peril of big data', [Online] Available at:http://www.aspeninstitute.org/sites/default/files/content/docs/pubs/The_Promise_and_Peril_of_Big_Data.pdf(11 July 2011).

Olmstead, Kenneth. Smith, Aaron. (2017). Pew Research Center. January, 2017, "Americans and Cybersecurity." http://www.pewinternet.org/2017/01/26/americans-and-cybersecurity/

Pandiani, J. A., Banks, S. M., & Schacht, L. M. (1998). Personal privacy versus public accountability: A technological solution to an ethical dilemma. The Journal of Behavioral Health Services & Research, 25(4), 456-463.

Richards, N. M., & King, J. H. (2013). Three paradoxes of big data. Stan. L. Rev. Online, 66, 41.

Schadt, E. E. (2012). The changing privacy landscape in the era of big data. Molecular systems biology, 8(1), 612.

GitHub Link

https://github.com/johnanthonyeletto/Vida-app