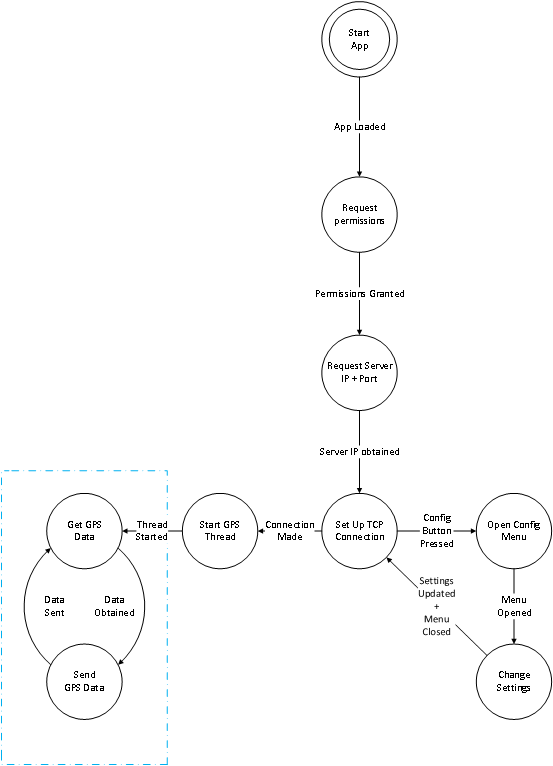
Android GPS Design

## Client FSM:



## Client Pseudo:

**START APP**

The entry point of the app. It will initialize our storage variables, and proceed onwards.

1. Load app
2. Initialize settings storage
3. Get location manager
4. Go to Request Permissions State

**REQUEST PERMISSIONS**

This state merely ensures that we can use the permissions in our manifest

1. Request GPS permission
2. Go to Request Server IP + Port State

**REQUEST SERVER IP AND PORT**

This state will function as a pair of textboxes that respond once the user clicks a submit button. It will check that the server exists via lookup, then proceed to the Set-up TCP Connection State

1. Extract IP
2. If IP is an invalid Host
   1. Display error message to user
   2. Prompt for re-entry
3. Extract Port
4. Go to Set up TCP Connection State with IP and Port

**SET UP TCP CONNECTION**

This establishes the connection to be used by the client.

1. Create Socket
2. Handle Errors
3. Go to start GPS thread
4. Wait for Config button press
   1. Go to Open Config Menu State

**START GPS THREAD**

This state creates a thread that runs a continuous loop of Getting the GPS data, then sending it to the server.

1. Create thread with start point at the GPS data state
2. Perform Error checking
3. Start Thread
4. Return to the Set up TCP connection State

**GET GPS DATA**

This state executes as a forever loop that reads the GPS data, then sends it to the server.

1. Subscribe to location updates
2. While true
   1. Request location update
   2. Go to Send GPS data State with current Location

**SEND GPS DATA**

This state takes the passed in location and echoes it to the server.

1. Parse location for desired fields
2. create message
3. append phone IP to message
4. append location data to message
5. write message to server.

**OPEN CONFIG MENU**

This state opens a simple config menu that allows the user to alter their settings for acquiring GPS data.

1. Open Menu fragment
2. Store user selections
3. wait for user to hit save button
   1. Go to Change settings state with Selections

**CHANGE SETTINGS**

This state takes the users selections and applies them to the GPS criteria.

1. Parse user selections
2. Update existing settings
3. Close menu

## Web Server FSM

C:\Users\Bobo\Downloads\AlexZielinskiWebServer.png

## Web Server Pseudo

**AUTHENTICATE USER**

When the user enters the web page they need to be authenticated. So when the user enters the web page they will be prompted to enter a username and password.

1. Prompt user with a username and password field
2. Compare with username and password credentials saved in webserver config file
3. Grant access if successful (else notify user of incorrect username or password and re-prompt)

**INIT GOOGLE MAPS API**

This state is responsible for initializing Google maps within the web browser.

1. Init Google Maps API key
2. Display Google Maps in web browser

**READ COORDINATE VALUES FROM DB AND PLOT**

This state is responsible for retrieving client coordinate values stored in a database. It will then plot the points on the map and link the points together to show the (real-time) movement history of a client

1. Read database client info entries
2. Check if client is a new client
   1. If new client then store the client ID locally on the web server and assign a color
3. Plot coordinate point with according client color
4. Link previous point with new point



**PROCESS COMMAND-LINE ARGS**

1. If more than two arguments were specified
   * 1. print error and usage message
     2. Exit
2. if one argument is specified
   * 1. set port as default
3. otherwise
   * 1. set port as argument 2

**SETUP TCP SOCKET**

1. Create a socket to listen to
2. Set socket options
3. Initialize address information
4. Bind address to the listen socket
5. Listen for connections
6. Set up an array of client descriptors to be used

**MONITOR CONNECTIONS**

1. While the server is alive
   * 1. If the listen socket is set
        1. Go to ADD NEW CLIENT
     2. Go to READ SOCKET

**ADD NEW CLIENT**

1. Accept the connection
2. Add the new socket descriptor to the client container
3. Add the new socket to the socket set
4. Make sure the max number of clients has not been reached

**READ SOCKET**

1. Go through each client in the client container
   * 1. If the socket is set
        + 1. read the socket
          2. If connection was closed

Go to Remove client

Exit state

* + - * 1. Go to STORE INFO IN FILE

**REMOVE CLIENT**

1. Set that clients value in the client container to an invalid value
2. close the socket
3. Increase the number of available clients

**STORE INFO IN FILE**

1. Get the file path to Apache’s default directory
2. Open a file for appending in the file path
3. Write the GPS client data to the file
4. Write a newline character
5. Close the file