Chappy Use Cases

User Registration:

1. User loads app
2. App checks cache for existing account, finds none, requests user to login.
3. User enters e-mail address, toggles radio button indicating chaperone or client, hits ‘Register button.
4. App sends object to the web service, which checks whether the e-mail address is already registered in the server.
5. If not registered, inserts user information into the database and returns an object back to the app containing required data.
6. If registered, send error back to app indicating that the email address is already registered.

User Login:

1. User loads app.
2. User enter email address, toggles whether they are a client or chaperone, and presses ‘Login’ button.
3. App sends message to web service, which checks to see if the email address is in the chosen (client or chaperone) database list.
4. If the email address in found in the database, proceed to next use case based on whether use is a client or chaperone.
5. If the email address is not found in the database, alert the user that the email address was not found.

Client requests chaperone:

1. Client presses ‘Request Chaperone’ button.
2. Another view comes into focus, client selects what building they are in from a drop-down list and what building they want to go to from a drop-down list, and presses ‘Submit’ button.
3. App sends a new transaction request to web service which creates a new transaction in the database.
4. App polls the web service every x seconds to check if a chaperone has accepted the transaction.
5. When a chaperone is found, change view to display the distance in which the chaperone is away from the client.

Chaperone checks for open clients/transactions:

1. Chaperone opens client, logs in as a chaperone.
2. App sends message to the web service to check for any open transactions on the server and send back a list. Sends an empty list if there are no open transactions.

Chaperone accepts a client:

1. Chaperone selects an open client from the list of open client transactions.
2. App sends a message to the web service with the selected transaction and the web service updates the transaction on the server to indicate a chaperone has selected the client.

Chaperone arrives at client pickup location:

1. Chaperone and client meet up.
2. Chaperone presses ‘Arrived at Client’ Button
3. Client App sends message to web service telling it to up update current transaction arrival date in the server.

Chaperone/Client arrive at destination:

1. Chaperone/client arrive at destination.
2. Chaperone presses ‘Arrived at Destination’ button, sends a message to the web service.
3. The web service updates the transaction in the database.