## Capstone Project Proposal

### Overview

The goal is to make a fullstack CRUD app (provisionally called ‘Dress.me’) that allows someone to search for people willing to offer personal fashion shopping advice. Once scheduled they agree on a meeting time and place and the shopping assistant goes with the shopper.

#### Problem definition:

* Some people
  + Do not have fashion sense
  + Are not up-to-date to on trends
  + Do not know the best places to shop for what they want
  + Are weary of shot staff’s opinions (their job is to sell)
  + Don’t have access to someone’s opinion
  + Experience higher frequency of buyer’s remorse

#### Benefits:

* Shopping assistants:
  + Could be fashion students need experience working with people on fashion issues
  + People with good sense who need extra cash
* Shoppers
  + Access to unbiased opinions for a small fee
  + Ability to search for someone to help based on location, style and area of expertise
* Shops
  + Lower rate of ‘buyer remorse’ and therefore returns/exchanges
  + Data produced from the functioning app can be very valuable to identify customer trends

#### Main features:

* User registration
* Login
* SA login
  + Create and edit their availability schedule
  + Create and edit their profile
  + Upload a photo via webcam
  + View upcoming appointments
  + Request to edit/cancel upcoming appointments
* Shopper login
  + Create and edit their profile
  + Upload a photo via webcam
  + Search for a shopping assistant based on location, style/expertise
  + View upcoming appointments
  + Edit/Cancel upcoming appointments

### Goals

* **Technical**:
  + Utilize location-based searching – perhaps using a google maps API
  + Create a relational database in PostgreSQL with multiple tables
  + Provide the ability to create a new user and login using authentication (Bcrypt)
* **Phase 1**
  + User being able to search for SA and book the meeting
* **Phase 2**
  + Multiple users
    - Registration/login
    - Profile
* **Phase 3**
  + Reviewing SAs
    - Include in search filter

### Tech Stack

* **Front-end**
  + React
  + Axios
  + Bootstrap for styling
  + Material-UI lists
  + CSS
* **Back-end**
  + Node and express on the server
  + Axios to communicate with Google Distance Matrix API
  + Google Distance Matrix API for location-based searching
  + Bcrypt for hashing
  + Mongo and mongoose as the database

### Data

* Phase 1 – Table – Shopping assistants (SAs)
  + Availability schedule – string, number(dates/times) and Boolean (available)
  + Location – Number
  + Willing radius – number
  + Profile
    - Name – string
    - Photo – use browser to take photo – image
    - Bio – string
    - Expertise – string
    - Reviews – string
    - Rating – number
* Phase 1 – Table – Schedule
  + Shopper and SAs – foreign keys
  + Timestamp – number
  + Passed or upcoming – Boolean
* Phase 2 – Table – Shoppers
  + Location – Number
  + Profile
    - Name – string
    - Photo – use browser to take photo – image
    - Bio – string
    - Reviews – string
    - Rating – number

### Project Timeline

* ~~29~~~~th~~ ~~– 30~~~~th~~
  + ~~Learning the google API~~
  + ~~Set up a form on front end~~
  + ~~Send to back-end to go to google API~~
  + ~~Start with an array of a few SAs on the server~~
    - ~~Can create a database after I get it working~~
* ~~31~~~~st~~
  + ~~Build Search function and UI~~
* ~~1~~~~st~~
  + ~~Schedule UI~~
* ~~2~~~~nd~~
  + ~~Have schedule display based on SA availability~~
  + ~~Build function to book and have it display as booked~~
* 3rd
  + ~~Build user profile page to show booked meetings~~
  + ~~Integrated database~~
  + ~~Cancel function from profile page~~
* 4th-5th
  + ~~Create login~~
  + Create registration
* 6th – 7th
  + Styling

### CSS

* Z-index