**Week 0**

Done:

* Created paper drawn images of all pages (total 4)
* Showed images to Heather
* Showed images to Keith
* Updated Trello board with project plan

Working on:

* Working on drawing out relationships between everything
* Math excel spread sheet

To do:

* Make fourth page (login and signup as one??)

**Week 1 – Day 1**

Done:

* Make login and signup – made whiteboard version
* Created a skeleton of:
  + Base html (with minimal bootstrap on base)
    - Should we download bootstrap (not necessary atm. Only downside to the way I am doing this now is that I’d need to have internet access to show people my demo)
  + All html pages w/ connections to base html
  + Routes from one html page to another
  + Creating server.py doc
  + **Result:** user can log in/subscribe and in session’s their cc info can be added to the database.
* Working on drawing out relationships between everything

Working on:

To do:

* Math excel spread sheet
  + Start with one card
  + Move up to two
  + Then three
  + Start thinking about how best to store all pieces of data for each step of the calculations
  + First do in python, then convert it into JS
* Create another table for the three main factors
  + Money
  + Time
  + sanity

Blocked on:

* Cannot push to github. May have altered something in github prior to pushing (deleting the .pyc file in github)
* Want to print out my server and model files and review them to make them more succinct. And join things like login and sign up

**Week 1 – Day 2**

Done:

* Ask my questions about:
  + Model.py
  + Server.py
* Update skeleton with features:
  + Get signup on one route. Not two
  + Figure out the spaces thing on line 91
* Make fourth page (login and signup as one
* Math excel spread sheet
* Math excel spread sheet
  + Start with one card
  + Move up to two
  + Then three

Working on:

* Update skeleton with features:
  + Make homepage button take you to login if session !=
* Math excel spread sheet
  + Figure out how to roll over payment amounts for one card to another once paid off (later stage? Add on??)
  + Start thinking about how best to store all pieces of data for each step of the calculations
  + First do in python, then convert it into JS
* Create another table for the three main factors
  + Money
  + Time
  + Sanity
  + Figure out how to do Boolean values?

To do:

Blocked on:

* Question about my data model, hw should I be storing the credit card inputs?
* How best to take data stored in database (Users, Cards, Values) and then do calculations in javascript

**Week 1 – Day 3**

Done:

* Update skeleton with features:
  + Make homepage button take you to login if session !=
* Create another table for the three main factors
  + Money
  + Time
  + Sanity
  + Figure out how to do Boolean values?
* Convert to PostgreSQL
* Add question to card-submission about min payment section.
  + Dropdown menu

Working on:

* Math excel spread sheet
  + Figure out how to roll over payment amounts for one card to another once paid off **(later stage? Add on??)**
  + Start thinking about how best to store all pieces of data for each step of the calculations
  + First do in python, then convert it into JS

To do:

* Figure out how to calculate payment plan with set amount per month in….. (later???)

Blocked on:

**Week 1 – Day 4**

Done:

* First do in python,
  + Just one card
    - All scenarios in excel spreadsheet worked when converted to python!!!!!!
  + Then make sure you can pass numbers into function. First have variables inside function

Working on:

* First do in python,
  + Then two
  + Then more
* Convert calculations into JS
  + Start thinking about how best to store all pieces of data for each step of the calculations (store as a dictionary)
    - Cards = {‘mastercard 12’: [100, 100, 94, 0], ‘visa 43’: [244, 198, 103, 0]}
    - Cards = [{name: ‘visa 34’, outstanding:1200, minPayment: 35}]

To do:

* + JS inserted into web app
  + Then store calculations into dashboard.html

Blocked on:

* How javascript works.
* Git commit and git push
* Getting calculations.py functions to work. SO LOST