**YU OneCard Analytics Network (YU C.A.N.)**

*Moshe Goldring & Mark Gotesman*

Stage 1: User Centered Design

**Who uses it?**

* YU Students
* YU Parents

**Why does he use it? What is he trying to accomplish when he uses it?**

* Tracking YU OneCard
* Budgeting YU OneCard

**What functions does your software provide to the user that help him achieve his goal(s)?**

* Alert system:
  + Overspending
  + Unusual spending (may indicate accidental charge 🡪 override
  + Underspending
* *Automatic* budgeting and tracking
  + Planning for daily/weekly/monthly spending, sent to user, calculated automatically based on cash remaining
  + (Indicate user spending relative to average YU spending habits)
  + Overbudget response:
    - 1. Recommended amount to refill café card with if running low
    - 2. Recommendations for how to adjust spending habits
* *Intermediate* presets for user types
  + E.g., go out twice a week and eat two meals a day at the café, be in for Shabbos 2/4 times a month, etc.
* *Advanced* manual mode
  + Customizations/preferences:
    - Desired budget
    - Locations
    - Times of purchases
    - Budget calculations based on time in YU
    - Notifications frequency & device
    - Send to parents
* Display/Data visualization
  + Purchases time plots
  + Pie charts
  + Locations bar graph
  + Location heat map (?)

**How does he use it? What steps does he go through in order to achieve his goal(s)? What are the workflows he progresses through when using it?**

* Automatic Users
  + No sign up required (integrated into YU backend system)
  + Or, small signup process (providing OneCard log-in information)
  + Receive emails at preset intervals (TBD), automatic recommendations for decision making (e.g. lower spending or add more money)
* Manual Users
  + Customize all the settings for preferences (see “advanced manual mode” above)
  + Receive more in-depth data visualizations and with customized setting and on customized devices

Stage 2: Data Model

**Input**

* JSON Files
  + Data will be scraped from the YU OneCard website and stored in JSON files to be read by the program.

**Data Storage/Access**

* “Purchase” *Objects*
  + Lines of data from the JSON file will be stored as purchase objects, which have four attributes:
    - Date
    - Time
    - Location (Enum will be used to represent the different locations)
    - Price
  + Date and Time may themselves be objects that have helper methods associated with them (e.g., for date, passing in another date and comparing them).
* “Student” *Class*
  + A student class will perform the main functions of analyzing the data and storing the purchase hash maps for the given student.
  + Potentially, we will develop a “student store” class to manage all the students, which could be used to find data on the whole student body as a whole (e.g. average purchase value at Nagel Bagels, hour of the day when most money is spent).
* *Hash Maps* (x4)
  + The purchase objects will be created once, and each attribute will be represented in a separate hash map as a key – lookups are then in O(n) time.
  + The value stored for each attribute will be a collection of purchases that match that specific key (e.g., the key of 4/12/2022 will have a collection of all purchases that were done on that date).
  + The hash value of each attribute is easily determined, as they all exist in a fixed range.
  + Queries for multiple attributes will collect multiple sets, and then the sets will be compared to arrive at the final result.