TEAM

Matan Gans

Strengths:

- Algorithms
- Data processing and querying
- Good at explaining code and logic

Weaknesses:

- Prone to lack of patience under stress

Matteo Lunghi

Strengths:

- Leadership
- Software design
- Algorithms and data structures

Weaknesses:

- Sometimes rushes into writing code too quickly.

Hossam Zaki

Strengths:

- Collaboration
- Strong work ethic
- Front-end design
- Well informed on biology and medicine topics

Weaknesses:

- Easily frustrated

Mohamad Abouelafia

Strengths:

- Friendly ; Works well with others
- Organized and likes planning
- Good visualizer
- Experience in imaging and computer vision

Weaknesses:

- Gets annoyed easily

Project Idea

1. **Futurosis (Currently Preferred Idea):**
    - Overview: This app will allow users to pinpoint developing medical conditions and offer likely diagnoses based on the inputted symptoms. At the moment, when we don't feel well and aren't ready to make the trip to the doctor's office just yet, our first step is the internet. However, a Google search is likely to freak you out, sending you to a WebMD result that can take an upset stomach and turn it into appendicitis, or create arthritis from a sore muscle. DiagnoseMe will allow a user to select a specific body part from our GUI, and then input specific symptoms that correspond to that body part. We will make use of big data to calculate likelihood of multiple diagnoses and give the user several options of what their symptoms may indicate, as well as a ranking of how likely it is. By using data and statistics, we hope to help users freak out a little less before going to the doctor when there is really not much to worry about.
    - Requirements: GUI (makes it easier for users to specify what symptoms they have), database with lots of data
    - Results from users: we asked around and our personal experiences were confirmed - WebMD usually freaks people out unnecessarily. One example story we heard is somebody had pain below the belly button, and WebMD only gave two options: the patient had either a kidney stone or appendicitis. This person went to sleep and was fine the next day.
    - Challenges: finding good data, deciding how to rank factors and decide on likelihood of diagnoses, making an interactive GUI

2. **MyRoutine (Title Tentative):**
    - Managing your time is hard, especially when you have to juggle courses, clubs, and (hopefully) food and sleep. It's easy enough to plug time blocks into your schedule when you know how long something will take (i.e. a lecture or a club meeting), but it gets trickier when you need to quantify the time you need for homework, studying for exams, long term projects, etc. We want to develop a program that will help users feel more productive in allocating their free time effectively.
    - Requirements: nice front-end interactive schedule design, algorithm that takes work out of the user's hands and develops a schedule for them (don't want to make the user work too much)
    - Results from users: we asked around if people would find this useful, and the the responses indicated that there is a need for this project - users feel unproductive

and would be more secure in their productivity if they actually had a rigid schedule, but don't actually end up making it on their own

- Challenges: Algorithm to fill free blocks of time with our suggestions for when a user should work on other tasks in the to-do list according to how much time they specify each should take (some sort of malloc-type functionality), interactive GUI with options for the user to play around with suggestions and make an optimal schedule.