

(\*\*\* Evicted - CIS 330 \*\*\*)  
(\*\*\* Group 14 \*\*\*)  
(\*\*\* Tayler Mandel, Lucas Peña, Mary Xia \*\*\*)  
(\*\*\* Documentation \*\*\*)

Our group created a game with the Yelp dataset given to us. We decided to model this game off of the Sims by allowing players to move around cities and work to keep their stats at optimal levels. The features that we initially proposed were as follows:

1. Using the Google Maps API to show locations of businesses being used.
2. Using the Yelp API to get images of businesses for their individual pages.
3. Using the Facebook API to login or creating an account.
4. Saving user data in database for users to continue playing at a later date.
5. Each user has stats like health, hunger, happiness/satisfaction, fatigue, location, money.
6. Inventory can include pets, vehicle, clothing that can be bought at businesses.
7. Integrate weather into site using Weather Channel Data Feed.
8. Use reviews for job wages, food quality, happiness, and health.
9. Time changes based on actions, not real time, so hunger, health, and fatigue change based on number of actions/clicks.
10. Fatigue is based on vehicle type as well as distance traveled between businesses.
11. Use comments in reviews for random events at locations, like being closed randomly or having mice.

This is our new list of features at the time of submission:

1. We use the Google Maps API along with location markers that are dynamically loaded onto the map.
2. The user's location is a foreign key to a business, and information about that business is loaded with the image and a link to the yelp business site.
3. The user can login/register for an account.
4. The site saves state in our database so user can continue from where they were.
5. The user has stats (health, hunger, fatigue, etc.) that are displayed in bars that change as the game progresses.
6. The database is organized such that each business has a category that fits with those described in cats.txt.
7. We use a weather API to determine how comfortable the user is in his comfort bar.
8. A python script parses five random reviews of a business each time the user does an action there. This evaluates how stats change.
9. When the user navigates to new place, fatigue and hunger change based off distance travelled.
10. The user can both move to a location (changing his/her stats) or perform an action or work at that place.
11. When a user's health reaches zero, he dies and the page redirets to a death page. This deletes the user and closes the session.
12. Stats change based on steps, not time. As a player moves, works, or does an action at a place, his health and fatigue change slowly due to time changing.

**Completeness:**

We tried to follow our initial list of features to the best of our ability. We found that as we worked, some of the features that we proposed initially did not fit with the overall goal of our website. Below I will address some of these changes with their initial feature number following the description:

- We found that we did not need the Yelp API to get images of businesses, as we had a link to the photo URL in the Yelp dataset. We still use business images in our table that displays information about the user's current location, but we are not using the Yelp API for this. (2)
- Adding Facebook login capabilities would not match well with the fact that we had our own Users table that kept track of a username and password. We decided to just get rid of this in favor of using our own table to keep track of users. Since each user lives alone and there is no means of finding friends on our site, there is no need to have Facebook integration. (3)
- We decided to remove the concept of items entirely from our design. We found that items wouldn't really fit with what we wanted to do, as it complicated user stats and the categories that we assigned to each location. Since the dataset doesn't include information about what each business sells or the price of the item, we found it easier to just base pricing on reviews and scratch the idea of having an inventory. We instead broke businesses up into categories and allowed one to choose what stat he or she wanted to change based on these categories. (6)
- The initial plan was to have items and base fatigue on the type of vehicle the user had. This has changed, as we have removed the concept of items. We now have allowed for one to move to a location and work there to make up for the changed fatigue. (10)
- There are not random events at locations like proposed. We decided that our parsing was too slow to do on every timestamp more than once, so we got rid of this feature for more ease of use. We now allow one to die if his health reaches zero, causing his account to be deleted and logged out. (11)

#### **Documentation:**

We have a README that describes every main file in our project as well as this Documentation to describe our changes upon completion of the project. Our blog posts also discuss all of our changes as well as our features as we worked on the project. Our code is also mildly commented.

#### **Regularity of Blogging + Meeting TA Mentor:**

We blog every time we meet and work on the project. We have also met regularly with Saurabh to discuss our plans as well as progress. We have emailed Saurabh with questions as well, so we have been very actively taking advantage of Saurabh's advice and documenting our changes.

#### **Novelty of Idea + Usefulness:**

We find that our project is unique, as no other group seems to be using the data for any reason other than suggestions or statistical analysis. We decided when we first met as a group to discuss our ideas that we wanted something different, and this made talking to other groups about our problems nearly impossible, as everyone else was doing practically the same thing.

#### **Database Schema Quality + Indices:**

Our schema reflects the structure of both the Yelp dataset as well as our particular use of it. Our game stores information about users including a foreign key to a business in order to properly link the business that a player is at with a business tuple in the Business table. The Review table is built as you would expect, with a reference to the Business table. Reviews are used for our parsing, and they are queried by the businessid, so no index is needed. The Business table has an index on (xcoord, ycoord), as we often query the Business table to find other businesses that are in a certain radius from the user's

current location. This index allows for faster search in our database. Our other queries for the User table use username in the WHERE clause, so no index is needed since username is a primary key.

**Data Importing:**

We modified the Java code from our homework about importing data to comply with the format of the Yelp dataset. We do not use Yelp users in our project, so we did not feel that it was necessary to import this information. We also did not include all of the data about each business or review. We took advantage of the Yelp categories that describe each business in order to classify each business to our needs (based on the mapping in cat.txt). Our data was imported all at once initially, but as our schema changed, we continually reimported the Business table using the same, slightly modified, Java project.

**Performance, Architecture, Seamlessness of User Experience:**

We initially queried the database several times per update, but we found that this was slow and often gave us the wrong results after refreshing the page. This caused us to refactor our update.php code to only do a single update on the database after calculating stat changes and parsing reviews. This change helped us cut down the time needed to update the database. We also initially had four or five different python scripts that parsed each review (in update.php). This meant that we called each script five times (once for each review) in each if statement. This was fixed by concatenating all of the reviews together and creating a single python script. We now have only one python call for parsing that looks at each word only once as opposed to several times. We have seen performance increases due to this change. We also tried our best to avoid having the page refresh when buttons were clicked. We used JavaScript to dynamically update a table to avoid refreshing. The site now runs much more quickly and requires little wait time.

**UI Design:**

Our user interface is designed for ease of use. It may not be the prettiest thing, but it does what we need and does not waste space. It is mostly put together via html tables, as we found this to be the easiest way to organize our information. We dynamically update the locations table using JavaScript to avoid refreshing the page too often. We thought about our design often. We initially had the map in the middle of our page, but we found that this wasted space, as there was a lot of extra room under each table. This new setup allows us to have a larger map and have bars to show stats instead of just numbers.

**Presentation:**

We plan to present with gusto.

**Extra Credit:**

We feel as though our project was very different from everyone else's. This caused us to have to work entirely on our own without advice from other groups, as none had the same issues as we had. Also, our group has only three members, as our fourth member dropped the class after groups had already been set. That being said, the quality of our work being on par or even better than that of other groups that had the advantage of having four group members should be noted.