

Detailed Project Specification

Project Name: BESTAT

Project Member: weid1, ziqil1, danh

Project Proposal

Project descriptions

Before you go to any new cities, the first question you want to ask is where should I live. You want to find a hotel to enjoy your vacations. You want to seek an apartment to spend your summer. Or you want to buy a house to move in. People make decisions by collecting information from Google, Zillow, Bookings and so on. However, it is so easy to be overwhelmed and ignore some critical information. As a result, most people regret after moving in.

To solve this problem, we propose to develop a web-based GIS application to assist people in site selection. When users click a point, we will be able to offer a living index by aggregating information from multiple APIs. The living index will be calculated based on the business, crimes, transportation, and the living index of different categories will also be demonstrated since different users might have different preferences.

The data source includes the Google place API, the Spotcrime API, the DataGov API which provides some demographic information, and Zillow neighborhood boundary and Google Map API as the Map layer of the projects. If time permits, we can scale up the project by adding more features. First, the data granularity and service coverage will be determined based on the working progress and accessibility of the data. Also, we might add some survey questions for the user and recommend places to them. Hopefully, a user feedback system will be integrated into our recommendation system.

Technologies

For backend, we will use Django and GeoDjango with python3.6. To be more specific, we probably will use some cache services to avoid too much API calls.

For frontend, we will use Bootstrap or Materialize and JQuery, some JavaScript library such as Leaflet and even some client side libraries are also likely to be used depending on the detailed design.

Product Backlog

User module (Owner Dan Hou)

Registration: The users should be able to register with their email and verify its email.

Sign in: The users can sign in with their email.

Forget password: The users can reset password.

Log out: logged-in users can sign out.

User can log in with their google account.

User need to use Captcha.

Data Service module (Owner Ziqi Liu)

Obtain demographic data from US censurs.

Obtain place data from Google place API. The API call need to be run periodically to update.

Obtain data from spot crime API. The API call need to be run periodically to update.

Obtain data real time from Google map API.

Map the data in different granulite to the neighborhood we define.

Map Service module (Owner Dan Hou)

Users can choose city.

Users can see a map after choosing a city.

Users can see the neighborhood in the city they choose.

Users can place pin on the map.

Users can see the living index of the point they choose.

Users can choose multiple places and check the living index without reloading the page(AJAX).

Dashboard Service module (Owner Ziqi Liu)

Users can check details measurements of the place they choose.

Users can check the user reviews on this neighborhood.

Users can post reviews in this neighborhood.

Users can check the trend of a particular neighborhood.

Recommendation Service module (Owner Wei Dai)

Users can fill in preferences form.

Users can edit preferences form.

Users can check the recommended region in the city.

Users can check the neighborhood details by clicking the top recommendations.

Users can give feed backs on the recommendation (like, dislike).

The system will use the user generated contents (comments, feedbacks).

Testing (Owner Wei Dai)

Unit test: every version will be tested automatically before deploying.

Continuous Integration: The project will be integrated with Travis to automate the testing.

Data Models

Implementation

Our data models are implemented with Django built-in models, with SQL as database (This is only a draft design, we might shift to MongoDB if it has better performance regarding spatial data).

Data and APIs

Most of our data of models will be collected from different API or scrapy wrapper. Therefore, the fields in models will correspond to our APIs. See API Wrapper part for more details.

API Wrapper

Factfinder wrapper

```
def fetch(zip_code):
```

- :param zip_code:
- :return: dict

Fields

- POPULATION // *Population number*
- AGE // *Average age*

- EDUCATION // *Educational attainment: percent high school graduate or higher*
- HOUSING // *Total housing units*
- INCOME // *Average income*
- POVERTY // *Individuals below poverty level*

Spot crime wrapper

0. type of crime

- Theft
- Robbery
- Burglary
- Vandalism
- Shooting
- Arson
- Arrest
- Assault
- Other

1. neighborhood

```
neighbor = Neiborhood (name='Garfield', url='/pa/pittsburgh/garfield')
```

Fields

- neighbor.name
- neighbor.data # *dict containing crime data in last six month, where key is type of crime*

2. city

```
city = City (state='pa', city='pittsburgh')
```

Fields

- city.state
- city.city
- city.neighbors # *list of neighborhood instance*

3. crime

```
class Crime:
```

```
def fetch (self, lat, lon, radius=0.006): # Return a list of crime records and each crime record is a dict
```

Fields

- cdid
- type
- date
- address
- link
- lat
- lon

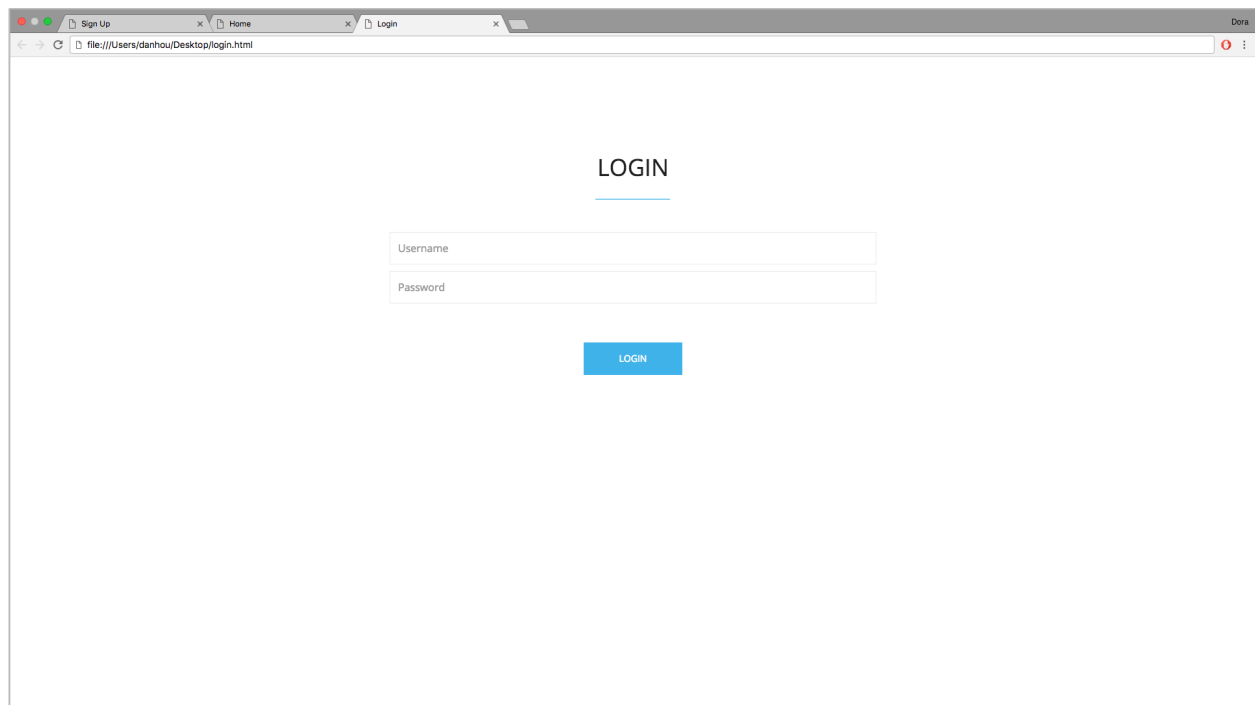
```
def crime_index(self, lat, lon, radius=0.06): # Return crime_index, calculated by crime. The higher the worse
```

Google Place

```
googleplace = GooglePlace()
```

```
def search_place (self, lat, lon, place_type, radius=1000): # Return a list of dict
```

Wireframe



A wireframe of a login page displayed in a web browser. The browser window has three tabs: 'Sign Up', 'Home', and 'Login'. The address bar shows the file path 'file:///Users/danhou/Desktop/login.html'. The page content is centered and features the word 'LOGIN' in a large, bold, black font, underlined with a thin blue line. Below the title are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. Both fields are white with a light gray border. Below the password field is a blue rectangular button with the word 'LOGIN' in white, uppercase letters.

Sign Up

file:///Users/danhoul/Desktop/signup.html

SIGN UP

Username

Email

Password

Re-enter Password

SIGN UP

Page 2 – Sign Up Page

Sign Up Home Login

file:///Users/danhoul/Desktop/home-page.html

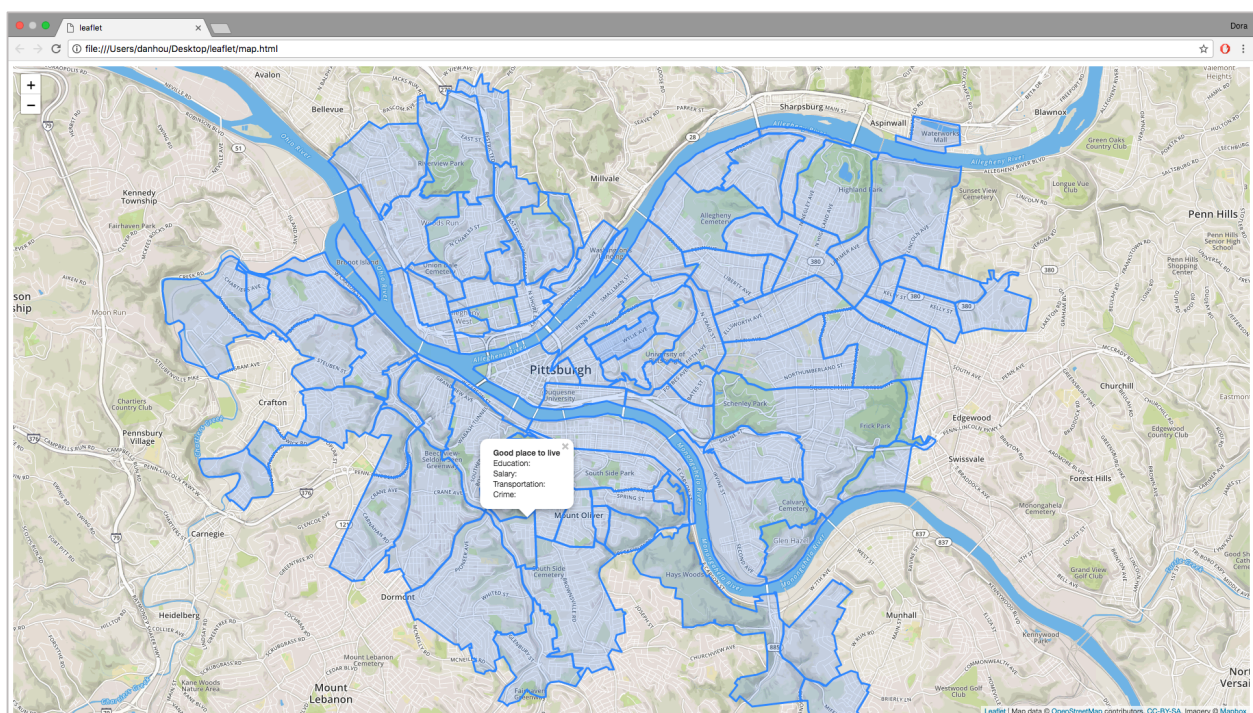
CHOOSE CITY

SUPPORT EDUCATION TRANSPORTATION SECURITY

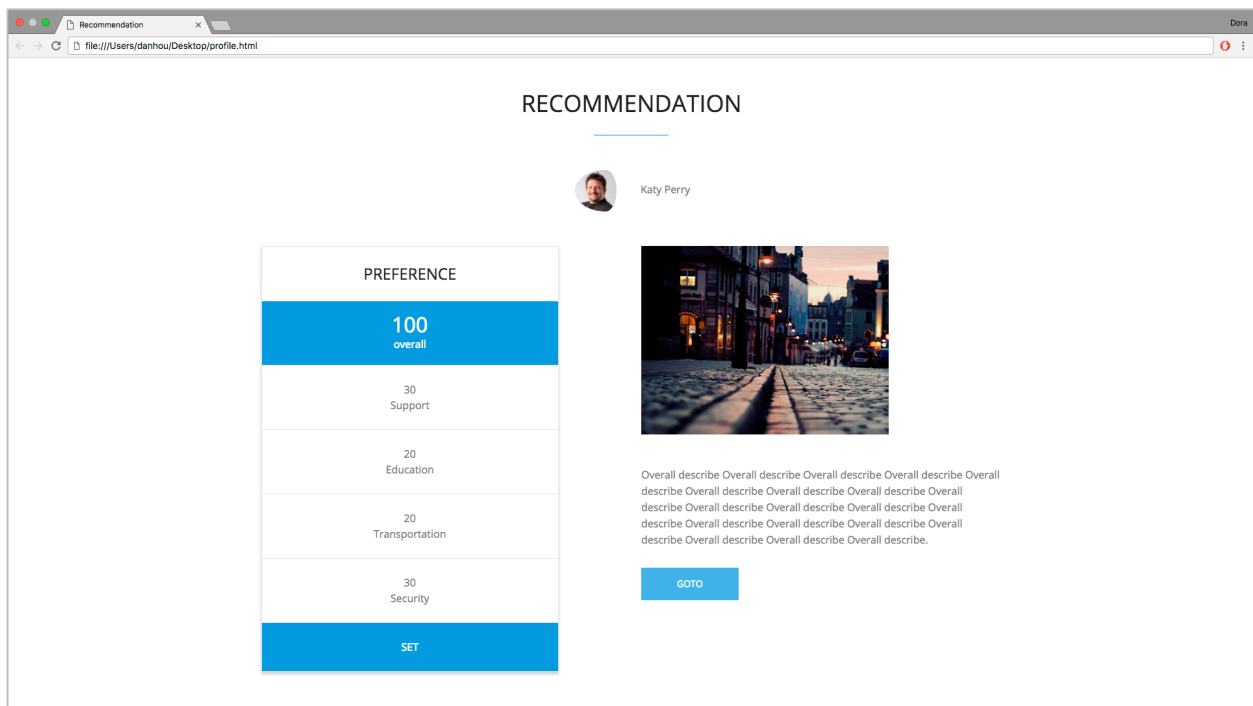
Start your new life...

CHOOSE

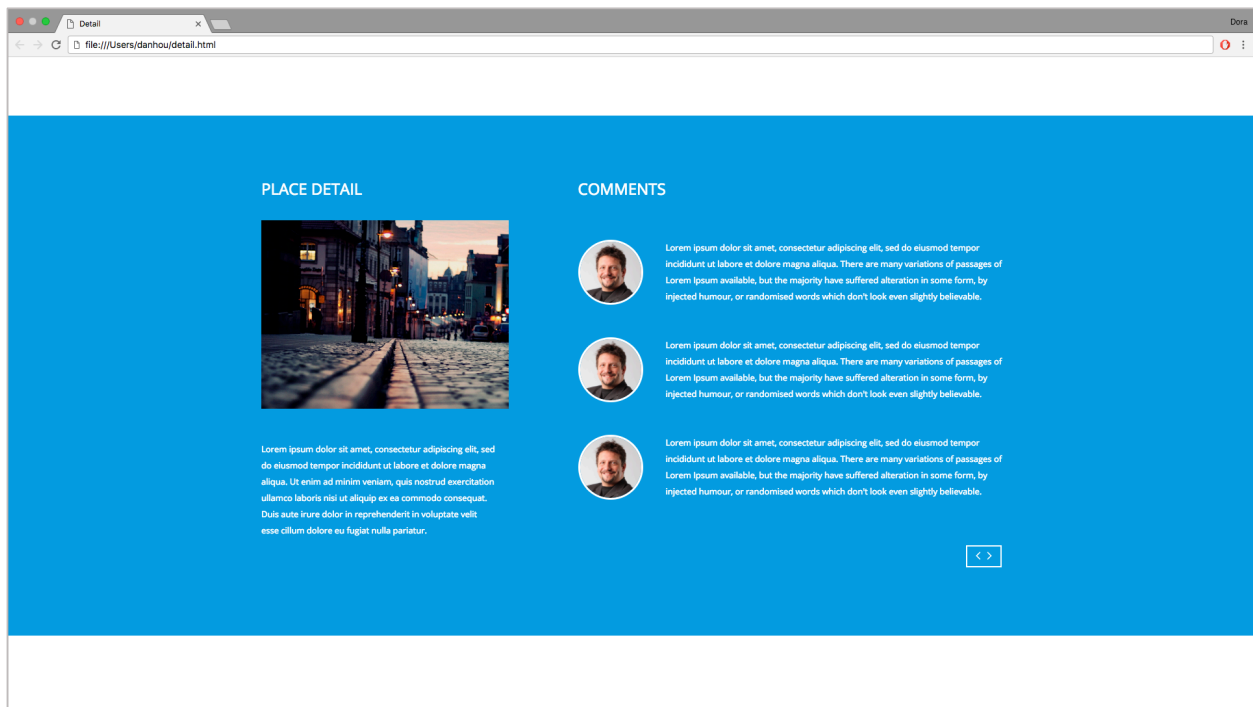
Page 3 – Home Page



Page 4 – Map Page



Page 5 – Recommendation Page



Page 6 – Detailed Page