TraX
X marks the spot

Team Do Your Code Work | Softdev Final Project
PM: Joel Ye | Jerry Chang, Shamaul Dilmohamed, Richard Wang

Abstract:

<Name> is a map creation web app where users can upload images (floor plans, for example) and produce maps in a guided process. These maps are published for others to access, where they can specify two points on the map and instructions for precise navigation are given.

Uses: Exploring all those annoying floor plan maps on billboards - now digitized!

(Inspired by Ms. Genkina's Stuyvesant Marauder's Map)

Project Description/Walkthrough:

From the guest user's end:

Guests arrive at the home page, where they can search for maps by name (or tags), or explore the gallery.

Upon choosing a map to use, they can pick out two nodes (start and end) by clicking, or by choosing from a list, and generate a path.

From the non-guest user's end:

Creators log in - they have all the options of a guest user, and have their own maps page (for saved pages) and a map creation page. Here they can start a new map or resume work on an old map. (+ A settings page for account management)

They can also choose to publish a map, provided it clears utility checks (optional).

Mapmaking page:

Note: There's a concept of floors/different areas. (This is designed for navigating buildings, but if there are moving sidewalks, say, in an airport, users can move from one area to the next and creators would indicate this by connecting the images.)

Users have several options on the mapmaking page. They can save, delete, or publish entire maps. They can upload or remove images, and clear the working map of markers.
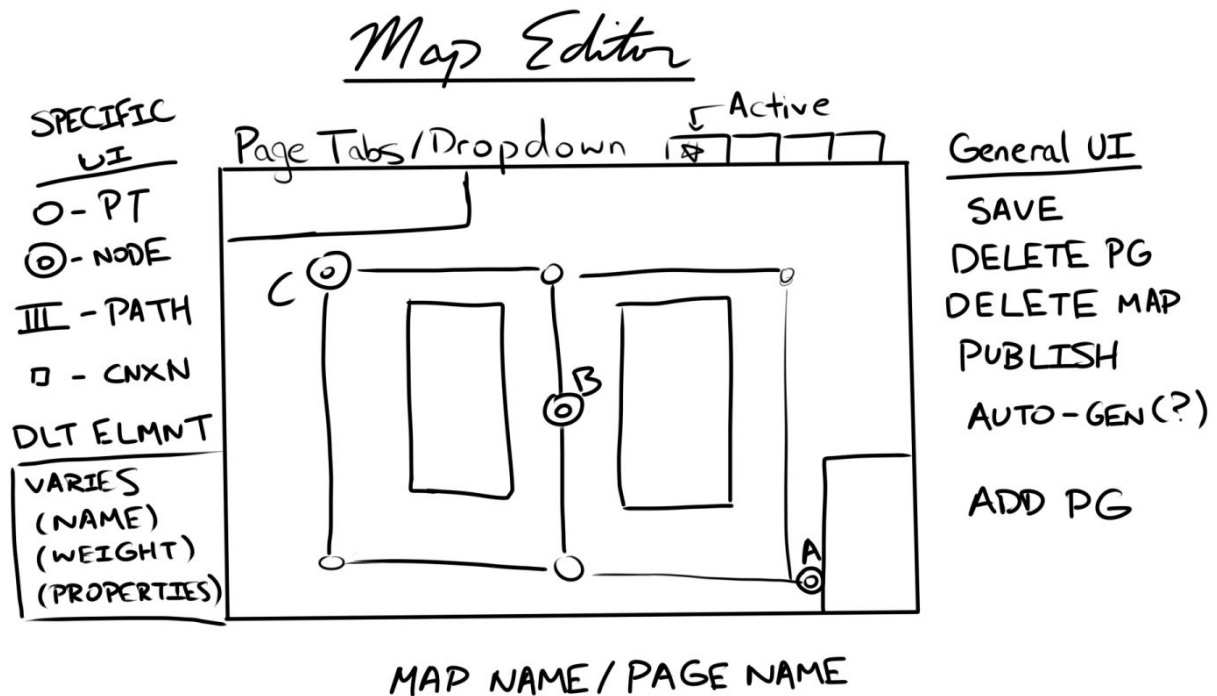
Types of markers:

Points - anchors for paths

Nodes - special type of point, destinations, waypoints, whatever you like to call them (these would be doors, for example) - specifies location and name.

Paths - connects points. Default weight is physical length, can be custom weighted by user (extension)

Connection - connects nodes between pages (similar to roads, different icon)

Map Editor

SPECIFIC UI

O - PT
◎ - NODE
Ⅲ - PATH
❑ - CNXN

DLT ELMNT

VARIES
(NAME)
(WEIGHT)
(PROPERTIES)

Page Tabs/Dropdown — Active

General UI

SAVE
DELETE PG
DELETE MAP
PUBLISH
AUTO-GEN (?)

ADD PG

MAP NAME / PAGE NAME
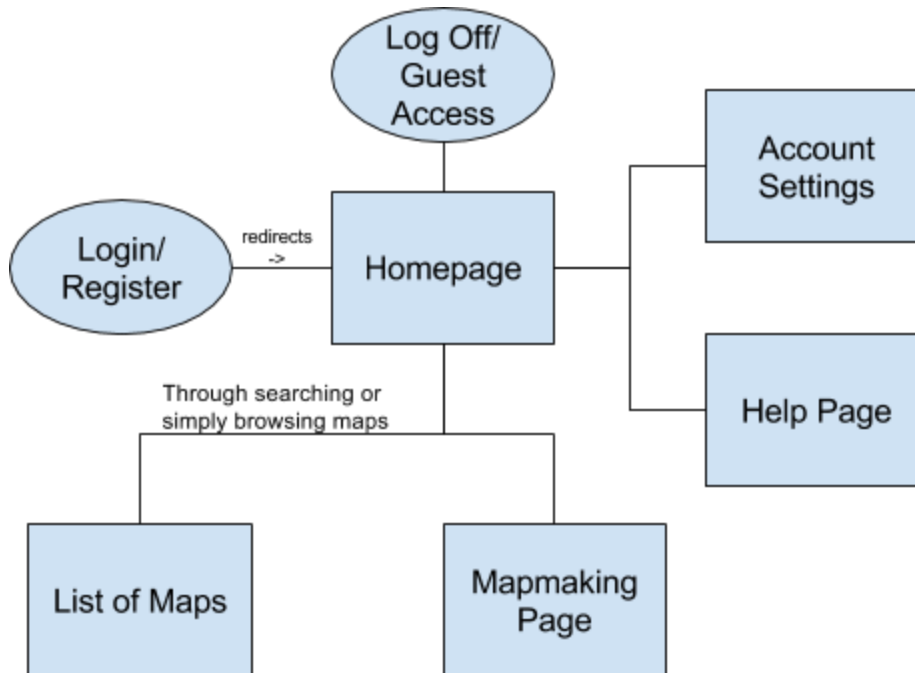
Database Schema:

Table *Users*
- uN: username
- uID: id
- pass: hashed password
- starredIDs: list of starred mapIDs
- ownedIDs: list of user-created mapIDs
- (optional) sharedIDs: list of private mapIDs user has access to

Table *Maps*
- mapName: name of the map
- mapID: map id
- ownerID: uID of map creator
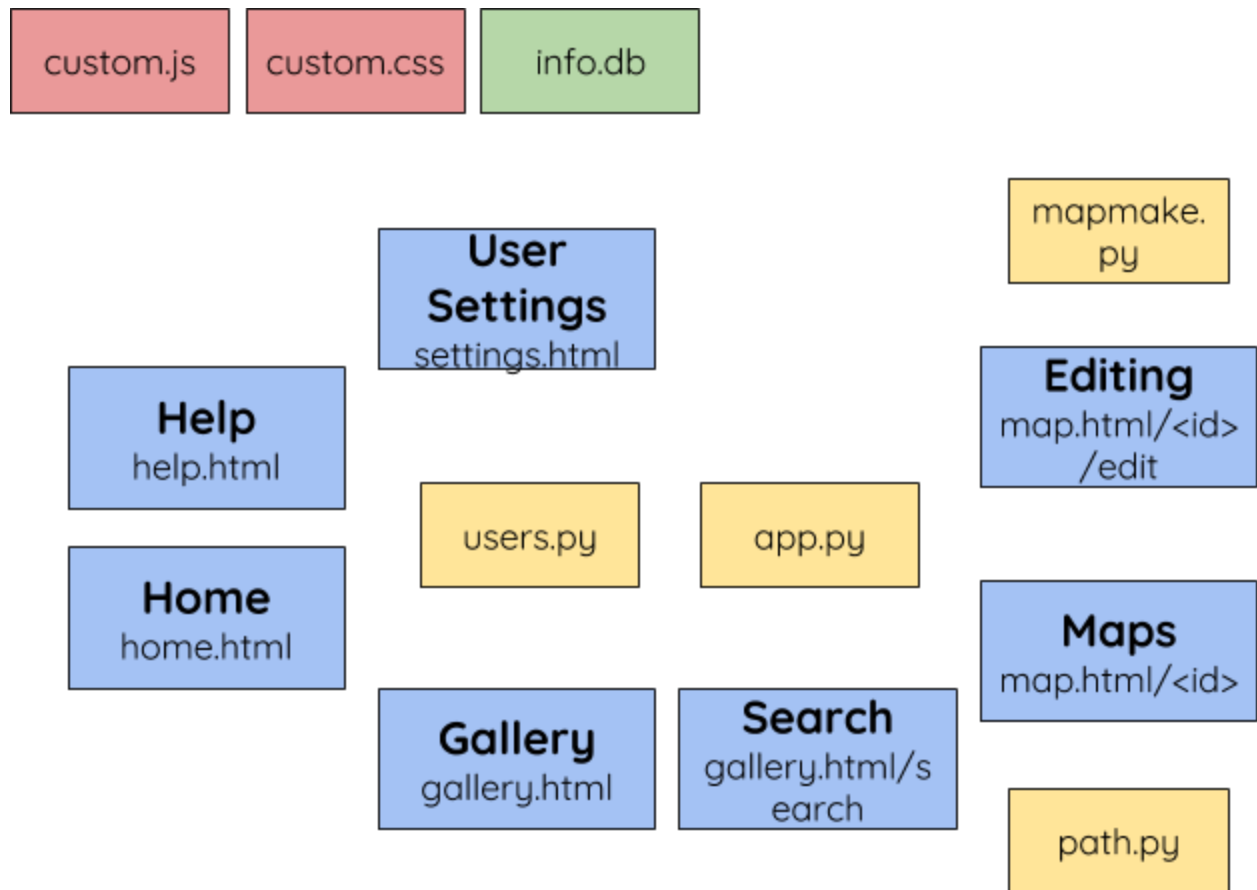- data: list of data containing all page info

- (optional) upvotes and downvotes

Site Map:

- Homepage: Introduction, about, credits, and search bar
- List of maps: Either browsing (by popularity) or with search queries
- Help page: Advice about mapmaking / ways to use the site

Component Map:

```
custom.js     custom.css     info.db

                                          mapmake.py

              User
            Settings                       Editing
           settings.html                 map.html/<id>
Help                                         /edit
help.html
              users.py      app.py
Home                                        Maps
home.html                                 map.html/<id>

              Gallery      Search
            gallery.html  gallery.html/s
                              earch         path.py
```

Component Description

- *home.html*
    - Big search bar, and general information
- *gallery.html*
    - Browsing public maps (optional search query)
- *help.html*
    - Advice on mapmaking, site usage
- *settings.html*
    - page where the user manages/changes their personal settings, like their passwords
- *map.html*
    - */<id>* : The unique page for an individual map
    - */<id>/edit* : The page for editing the specific map

- *users.py*
    - Functions dealing with user information
    - Login, authentication, user settings, user creation/registration, etc
- *app.py*
    - Flask app that deals with routing
- *mapmake.py*
    - Functions dealing with the editing of the map
    - Saving and retrieving data from the database
- *path.py*
    - Pathfinding algorithms, work with loaded maps
- *mapEditor.js*
    - JS that allows user interaction with map editor (d3) - interacts with mapmake.py
- *custom.js*
    - Blanket js file for general animation and everything else
- *custom.css*
    - Makes page purdy
- *info.db*
    - Holds all the data regarding users and maps

Task Delegation:

Joel: Flask routing/misc python, Editor - editing javascript
Jerry: CSS/Frontend
Shamaul: DB Management, File management
Richard: Pathfinding algorithm, everything else / interim frontend

Timeline (ROUGH):

5/15 - Basic login system, working templates, site hierarchy,
5/17 - Stuff unrelated to editor is done
5/19 - File upload/download working
5/26 - Users can put in points and submit, (algo not completed) and maps are makeable

6/02 - Working basic version complete

6/09 - Finish + Debugging + Extra features

Extensions:
- OCR Suggested path generation (making the creation process easier, carrying design from floor to floor)
- Mobile app to guide user precisely (google maps lite)
- Images of physical locations to make navigation easier
- Building the initial floor plan
- Statistics for creators, popularity of maps
- Tagging system for maps
- Collaborative mapping
- Maps that are shared with certain people