

CS5200 Final Project

Milestone Description

Group Number: 6

Ting Li, Siyuan Liu, Yunyu Guo

Overall aim for this project

We are going to help the non-profit *Trees for Oakland* by submitting a database design. More specifically, this applicant should allow residents to submit requests to plant a tree, the volunteers to sign up, and the admin team to process the planting requests and do the tree planting site visit. Besides, we should help to generate both required and optional data retrieve reports by writing SQL queries.

Class and Justification

Until March 22, 2025, we proposed 8 entities.

First, we deal with the aspect of the user. We noticed the "resident", "volunteer", and "admin" share similar attributes, like first name, last name, email, password, neighborhood and zip code. Besides, only Volunteer requires more attributes than the others. We naturally came up with inheritance: User is a super class, and Volunteer are its subclasses.

With the suggestion from Professor Monge, only the Volunteer class is kept as a subclass because it has extra attributes -- applicationStatus, since a registered user can apply to be a volunteer and see their status. Resident and Admin do not have additional attributes or unique behaviors, so we decided not to create separate subclasses for them. Instead, we use a role attribute in the User class to distinguish between Resident, Volunteer, and Admin. This simplifies the structure, possibly saves space, and makes it easier to manage user roles in our applicant. **Overall, the classes are *User* and *Volunteer*.**

Second, this is about the activities in different users' workflows. A resident user can request trees, but first, they need to upload their permit. The *Permit* class is used for this. Then, the resident can apply for trees, so we created the *TreeRequest* class to collect related information. After that, an admin should visit the site to gather more details, which is why we have the *SiteVisit* class. *PowerLine* (boolean) and *bedWidth* attributes are in *SiteVisit* class, which corresponds to the attributes in *Tree* class (discussed in next paragraph): plantable under power lines, minimum bed widths. These are used to recommend suitable trees to the planting site by admin. Next, the volunteer(s) will help with the tree planting work assigned by an admin. For this, we create the *TreePlanting* class. *TreePlanted* and *zipCode* attributes are in the *TreePlanting* class, these will be used to keep track of the tree species and number planted in different neighborhoods. **Overall, the classes are *Permit*, *TreeRequest*, *SiteVisit*, and *TreePlanting*.** *Neighborhood* class is associated with *User* class and *treeRequest* class to facilitate report that summarizes requests.

Lastly, it's tree. We need each tree species information for report and recommendation of suitable tree option aligned with planting site. Naturally, *Tree* class is generated with attributes based on the City of Oakland Street Tree List from. We use common names as tree species since common names are unique. Besides, after carefully reviewing the Official Street Tree List, we add the class *PlantingZoneFactor* to keep the design scalable. Tree inventory attribute is in the *Tree* class for admin to manage. **Overall, the classes are *Tree* and *PlantingZoneFactor*.**

Assumptions and Justifications

- Suppose Admin would do the site visit first, then assign (which is our interpretation of “leading”) one or many Volunteers to do the tree planting job.

- Volunteer can only see the latest status of its own application. Only store the latest availability of volunteers when they sign up for tree planting event.
- Resident can require a tree to be planted at residential address, and can require a tree to be planted at commercial address, so resident can submit more than 1 requests, a request only links to the specific resident.
- An Admin can go to multiple siteVisit locations, a siteVisit location only relates to an Admin.
- An Admin can observe many TreePlanting events, a TreePlanting event only has one Admin.
- A Tree has at least one PlantingZoneFactor, each PlantingZoneFactor has at least one corresponding Tree.
- Each TreePlanting event only plants 1 tree, each TreeRequest can only request 1 tree
- Each user lives in 1 neighborhood, a neighborhood can have many users; each treeReqeust has 1 neighborhood, a neighborhood can be associated with many treeRequests.

Reference

<https://www.oaklandca.gov/resources/official-street-tree-list>

<https://www.treesforoakland.org/how-to-get-a-tree.html>

<https://docs.google.com/forms/d/e/1FAIpQLSccg6WxSEgPx-epsUc4OKMJuGkzMUuCTZP2xw1nyW-6p-7kug/viewform>

<https://localwiki.org/oakland/Neighborhoods>

<https://sqldatagenerator.com/generator>