COMP 421 Project deliverable 2: Tree Management System

Camilo Garcia La Rotta, Elias Homsi, Shiqiao Zhu, Siqi Liu

March 2, 2019

1 Database description

1.1 Entities

- **Park:** A park is an entity representing any park in the map. Its primary key is its name. Its other attribute is its location represented by a polygon.
- **Tree:** A tree is an entity representing any planted tree in the map. Its primary key is its ID TID. It also has attributes for its species, planted date, and geographic location.
- **Assessment:** An assessment is a report on the status of a tree at a given time. It is made by an Environmental Scientist. It is a weak entity that requires the date-time of the assessment, the id of the environmental scientist and the id of the tree in order to be uniquely identified. The assessment also has a status and a taken action attribute.
- **Civic Location:** Whenever a citizen orders a tree to be planted, the user must specify the destination civic location of the new tree. It has a type (residential/commercial/farm) and it is uniquely identified by a civic address attributes.
- **Municipality:** A municipality is an entity representing any municipalities covered by the map. It contains trees and parks. It belongs to a city and is uniquely identified by its name. It also has population and polygon attributes.
- **City:** A city represents any city in the map. It is comprised of municipalities and has workers associated with it (urban planners and environmental scientists). Its uniquely identified by a name and has a polygon attribute.
- **Transaction:** Residents can order trees to be planted at their civic location. The order is represented by the Transaction entity. It is uniquely identified by a TID and it has an attribute representing the type of tree wanted to be planted and another one representing the status of the transaction (pending review/approved/refused). It must be approved by an urban planner for parameters such as biodiversity, carbon intake and pest control of the sector.
- **Urban planner:** An urban planner is itself a city worker which in the system will itself be a user. It is uniquely identified by a UID and also has a name, phone number, and email attributes.
- **Environmental Scientist:** An environmental scientist is itself a city worker which in the system will itself be a user. It is uniquely identified by a UID and also has a name, phone number, and email attributes.

- **Resident:** A resident is a citizen who does not work for the city but lives in a civic location within the city's polygon. It is itself represented in the system as a user. It is uniquely identified by a UID and also has a name, phone number, and email attributes.
- **User:** A user is the most general entity representing a user of the system. It can be either a city worker or a standard citizen. It is uniquely identified by a UID and also has a name, phone number, and email attributes.
- **Worker:** A worker entity represents the city workers. They can be either an environmental scientist or urban planners. It is itself a user entity in the system. It is uniquely identified by a UID and also has a name, phone number, and email attributes.

1.2 Updated Relations

After reading the feedback for last deliverable, we updated our relations accordingly, removing the redundant "Tree located" relation.

- **is inside:** A tree is inside a park. This is a one-to-many relationship. A tree can be located in at most 1 park. A park can contain many trees.
- **belongs to:** A tree belongs to a municipality. This is a one-to-many relationship with key and participation constraints because every tree must belong to exactly one municipality. A municipality can have multiple trees.
- **is within:** A park is within a municipality. This is a one-to-many relationship with key and participation constraints because every park must belong to exactly one municipality. A municipality can have multiple parks.
- **is comprised of:** A city is comprised of municipalities. This is a one-to-many relationship with key and participation constraints because every municipality must belong to exactly one city. A city can have multiple municipalities.
- **targets:** A transaction targets a civic location. This is a one-to-many relationship with a key constraint. Multiple transactions can be made for the same civic address. A given transaction can target only one civic location.
- **orders:** A resident orders a transaction. This is a one-to-many relationship with a key constraint. Multiple transactions can be ordered by the same resident. A given transaction can be made by only one resident.
- **reviews:** An urban planner reviews a transaction. This is a one-to-many relationship with a key constraint. Multiple transactions can be reviewed by the same urban planner. A given transaction can be reviewed by only one urban planner.
- **lives in:** A resident lives in a civic address. This is a one to many relationships with key and participation constraints. A user is considered a resident only if they live within the polygon of the city. A resident must have one civic location. A civic location can house multiple residents.
- **works for:** A worker works for a city. This is a many-to-many relationship with participation constraint. A worker must work for at least one city to be a worker. A city can have multiple workers. The attribute "since" allows to gather metrics on seniority of the employee.

- **undergoes:** A tree undergoes an assessment by an environmental scientist. This is a weak ternary relationship. A date-time attribute in the assessment entity requires both the ID of the tree and the environmental scientist to be uniquely identified. This is because a tree can be assessed multiple times during a day by different scientists.

2 Updated Relations

After reading the feedback for last deliverable, we updated our relations accordingly, removing the redundant ”Tree located” relation.

- TREE(tid, geographic_location, planted_date, species, mname, pname, civic_address)
  - Mname foreign key referencing MUNICIPALITY
  - Pname foreign key referencing PARK
  - Civic_address foreign key referencing CIVIC_LOCATION
- PARK(pname, ppolygon, mname)
  - Mname foreign key referencing MUNICIPALITY
- MUNICIPALITY(mname, population, mpolygon, cname)
  - Cname foreign key referencing CITY
- CITY(cname, cpolygon)
- CIVIC_LOCATION(civic_address, location_type)
- WORKER(uid, salary, worker_type)
  - Uid foreign key referencing USER
- EMPLOYMENT(cname, uid, since)
  - Cname foreign key referencing CITY
  - Uid foreign key referencing WORKER
- USER(uid, uname, email, phone_number, civic_address)
  - Civic_address foreign key referencing CIVIC_LOCATION
- RESIDENT(uid)
  - Uid foreign key referencing USER
- URBAN_PLANNER(uid)
  - Uid foreign key referencing WORKER
- ENVIRONMENTAL_SCIENTIST(uid)
  - Uid foreign key referencing WORKER
- TRANSACTION(transid, tree_type, tstatus)

- REVIEW(transid, uid)
  - Uid foreign key referencing URBAN_PLANNER
- ORDER(transid, uid)
  - Uid foreign key referencing RESIDENT
- TARGET(transid, civic_address)
  - Civic_address foreign key referencing CIVIC_LOCATION
- ASSESSMENT(assess_date, tid, uid, astatus, action)
  - Tid foreign key referencing TREE
  - Asssess_date foreign key referencing ASSESSMENT
  - Uid foreign key referencing ENVIRONMENTAL SCIENTIST

Note: Each worker should works for at least one city (as per our ER diagram) is not capture in the relation model.