Duty Reporting - Database Technical Documentation

Luke Carlson, Connor Swords, Devin Stein, Alec Olesky

Database Schema and description

We are using a NoSQL database with Parse (www.parse.com). Our UserTable keeps track of which users exist in the system and what their role is. Our other table is HourEntry which is used to keep track of resident hours.

UserType

The schema for our UserType table is:

UserType(objectId,name,isSupervisor,residents,createdAt,updatedAt,password)

objectId [String]- unique identifier for the row in the database (primary key)

name [String]- account username

isSupervisor [Boolean] - indicates if they have supervisor permissions

residents [Array] - list of resident names that this account is directly responsible for.

updatedAt [Date] - timestamp to keep track of changes

createdAt [Date] - timestamp to keep track of creation

password [String] - the password associated with the user account

HourEntry

The schema for our HourEntry table is:

HourEntry(objectId, startTime, EndTime, hours, inLocation, username, createdAt, updatedAt)

objectId [String]- unique identifier for the row in the database (primary key)

startTime [Date] - timestamp to indicate when a resident logged in

startTime [Date] - timestamp to indicate when a resident logged out

username [String]- resident associated with the entry

updatedAt [Date] - timestamp to keep track of changes

createdAt [Date] - timestamp to keep track of creation

Database Connection

To connect with our Parse database we use Parse’s Android package. This involves creating a query and iterating through results. For example, to get a list of names that match the string “ExampleName” you would write:

```
ParseQuery<ParseObject> query = ParseQuery.getQuery("UserType");
query.whereEqualTo("name", “ExampleName”);
```

To use the output of this query, you would just ‘get’ the needed information from the row. To continue the example, we get the inLocation boolean from each row who has “ExampleName” as their name:

```
List<ParseObject> pObj = query.find();
for (ParseObject p : pObjs) {
                boolean inLocation = p.getBoolean("inLocation");
}
```

Cloud Code

In order to handle sending program directs alerts about resident duty violations, we set up a js script hosted on Parse that checks for violations and notifies the program director. The script can be found in Parse’s cloud code section.

The general idea behind the code is as follows:

```
For a given username:
    for each of their hour entries for the past month:
        check if any of the violations are breached
    end
Notify the program director of any breaches
end
```