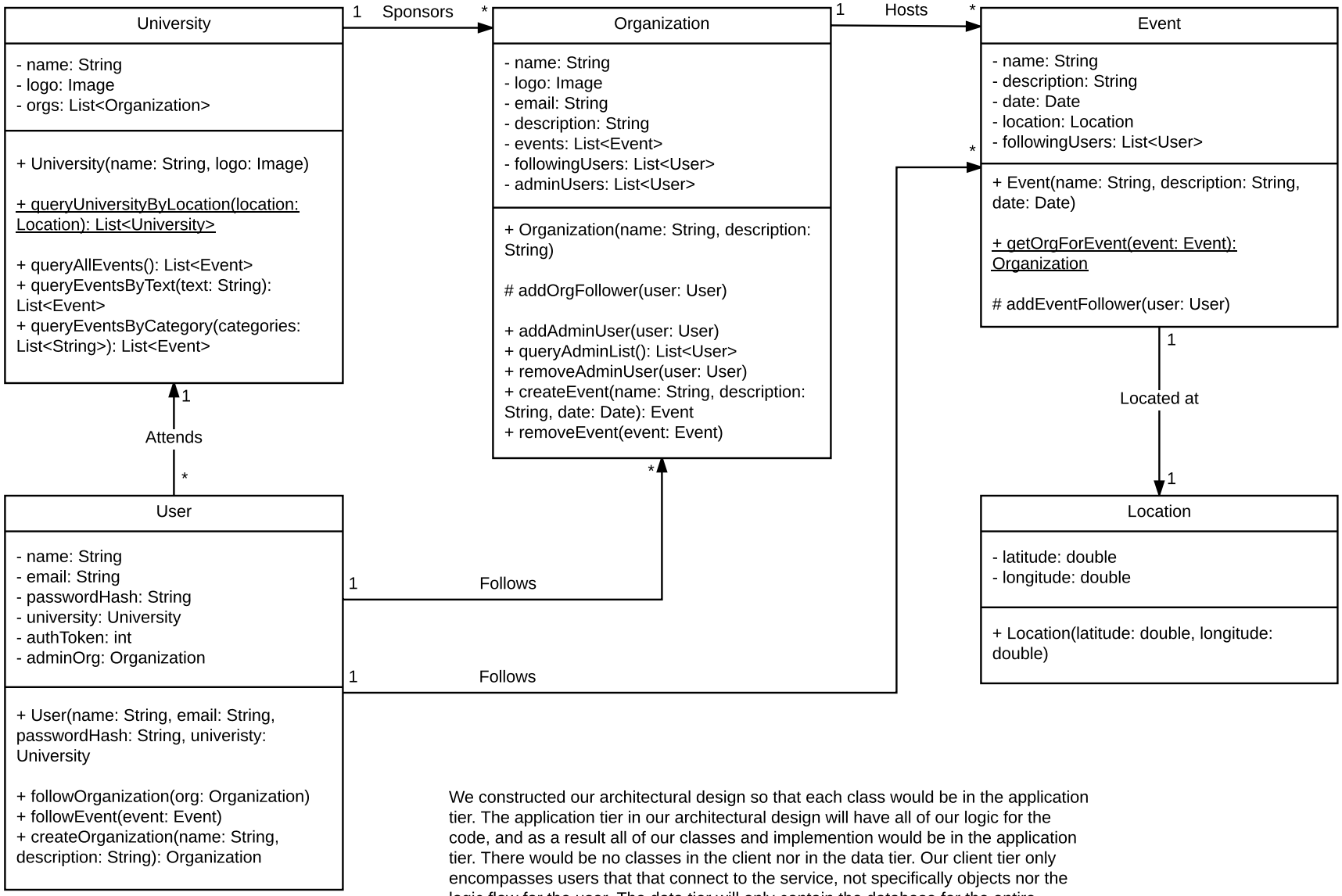
Deliverable 4

Group #5 - Event Planner
11/9/2016
CS 3304 Fall 2016



We constructed our architectural design so that each class would be in the application tier. The application tier in our architectural design will have all of our logic for the code, and as a result all of our classes and implemention would be in the application tier. There would be no classes in the client nor in the data tier. Our client tier only encompasses users that that connect to the service, not specifically objects nor the logic flow for the user. The data tier will only contain the database for the entire application.

Note: getter and setter methods for all classes are implied

Deliverable 4

Group #5 - Event Planner
11/9/2016
CS 3304 Fall 2016

University
- name: String - logo: Image - organizations: List<Organization>
+ University(name: String, logo: Image) + getName(): String + getLogo(): Image + getOrgList(): List<Organization> + setName(name: String) + setImage(logo: Image) + addOrg(org: Organization) + removeOrg(org: Organization)

Organization
- name: String - logo: Image - email: String - description: String - events: List<Event> - followingUsers: List<User> - adminUsers: List<User>
+ Organization(name: String, logo: Image, email: String, description: String, initialAdmin: User) + getName(): String + getLogo(): Image + getEmail(): String + getDescription(): String + getEventList(): List<Event> + getFollowingUserList(): List<User> + getAdminUserList(): List<User> + setName(name: String) + setLogo(logo: Image) + setEmail(email: String) + setDescription(description: String) + addEvent(event: Event) + addFollowingUser(user: User) + addAdminUser(user: User) + removeEvent(event: Event) + removeFollowingUser(user: User) + removeAdminUser(user: User)

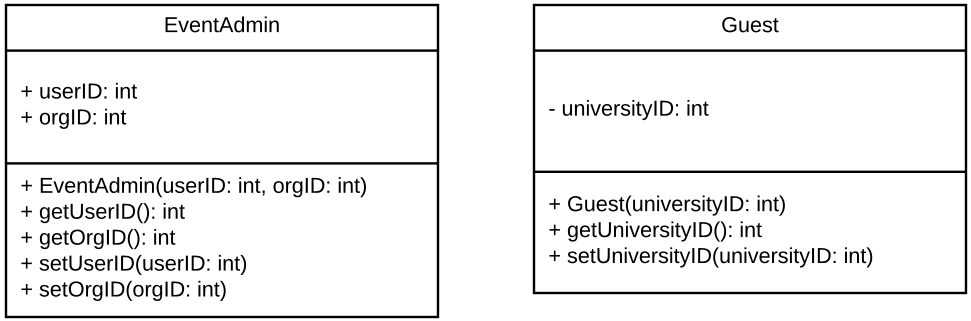
Event
- name: String - description: String - date: Date - location: Location
+ Event(name: String, description: String, date: Date, location: Location) + getName(): String + getDescription(): String + getDate(): Date + getLocation(): Location + setName(name: String) + setDescription(description: String) + setDate(date: Date) + setLocation(location: Location)

User
- userID: int - name: String - email: String - password: String - universityID: int - authToken: int - adminOrgID: int
+ User(userID: int, name: String, email: String, password: String, univeristyID: int, authToken: int) + getUserID(): int + getName(): String + getEmail(): String + getPassword(): String + getUniversityID(): int + getAuthToken(): int + getAdminOrgID(): int + setUserID(userID: int) + setName(name: String) + setEmail(email: String) + setPassword(password: String) + setUniversityID(universityID: int) + setAuthToken(authToken: int) + setAdminOrgID(orgID: int)

Location
- latitude: double - longitude: double
+ getLatitude(): double + getLongitude(): double + setLatitude(latitude: double) + setLongitude(longitude: double)

We constructed our architectural design so that each class would be in the application tier. The application tier in our architectural design will have all of our logic for the code, and as a result all of our classes and implemention would be in the application tier. There would be no classes in the client nor in the data tier. Our client tier only encompasses users that that connect to the service, not specifically objects nor the logic flow for the user. The data tier will only contain the database for the entire application.

Trash Bin


EventAdmin
+ userID: int
+ orgID: int
+ EventAdmin(userID: int, orgID: int)
+ getUserID(): int
+ getOrgID(): int
+ setUserID(userID: int)
+ setOrgID(orgID: int)

Guest
- universityID: int
+ Guest(universityID: int)
+ getUniversityID(): int
+ setUniversityID(universityID: int)