KloudBook High Level Document

Version: 1.1

Project Team Pippin

Team Leader	David Soohoo
Secretary	Dave Burgess
Customer Liaison	Caleb Larson
Integration Liaison	Dylan Terry
Coders	Daniel Choi Robert Hromada
Project Manager	Timm Allman
Client	Mike Mullen Dave Mullen

1. Introduction

The purpose of this document is to detail the high-level design of the database controller and back end portions. This document provides diagrams that outline our classes and functions, as well as providing API further detailing interactions with the web application and external sources. We are working with Cloud 7 (the web application and iOS team) and Team Kilobyte (the Android team) to form a complete social network for KloudBook.

2. UML Class Diagram

See Page 4

3. Database Schema

See Page 5

4. API

4.1 RESTful API

Implementing a representational state transfer API will increase performance by making data available on-demand by reference. The client-side applications will connect to our back-end database by pushing information through JSON schemas.

4.2 Oauth API

Oauth 2.0 enables front-end users to establish authorized connections, via tokens, to third-party HTTP services such as Facebook, LinkedIn, and Google+, and request information such as name, email, and phone number. Clients will be able to import information from external websites, while others will be able to access information from our database via tokens.

4.3 Back-end API

Our API allows the front-end applications to interact with the back-end database in order to exchange, update, and search for information by making use of the classes and methods explained below:

4.3.1 Profile

This class pertains to each user by accessing the database with a unique ID and has methods for creating a profile, logging into that user's account, editing information, creating groups, and deactivating an account.

4.3.2 User

This class holds information about the client and other KloudBook users. User interacts with other User objects and is queried to display .

4.3.3 Search

This class holds functionality pertaining to the querying of Users by their fields.

4.3.4 Connections

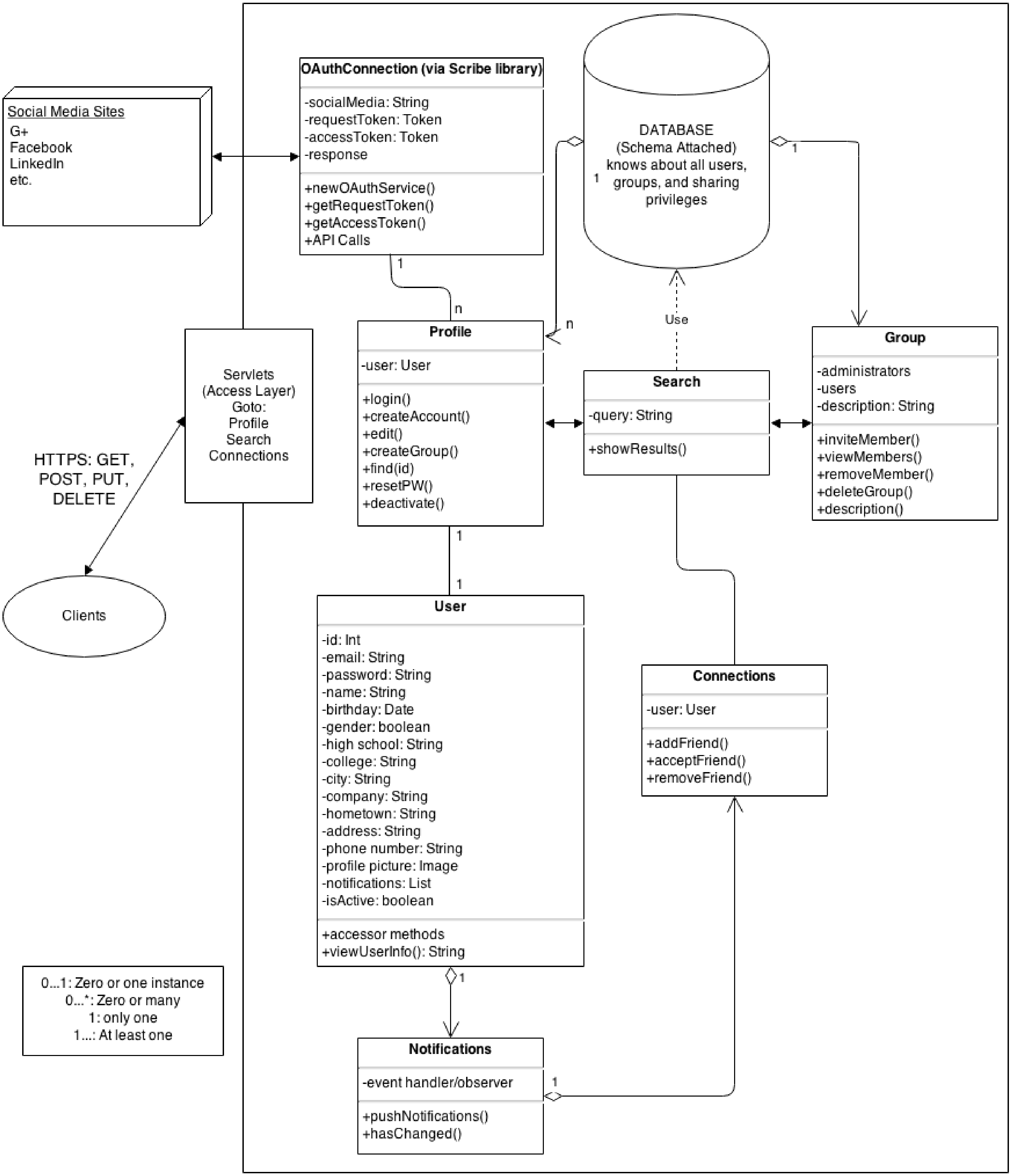
This class is responsible for requesting friends, handling whether a request was accepted or denied, and removing friends.

4.3.5 Notification

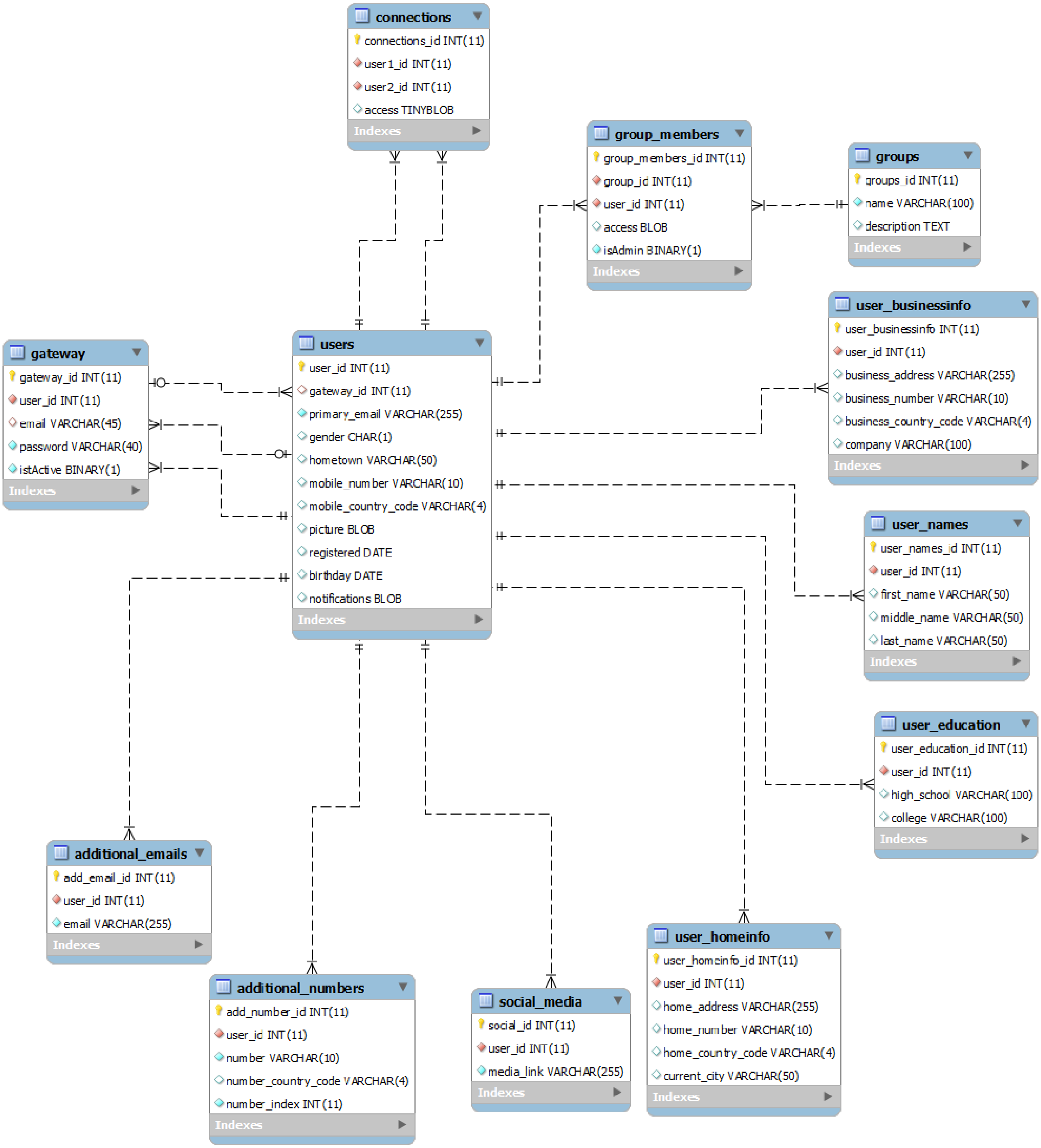
This class will forward events to a User’s notification list.

4.3.6 Groups

This class creates and maintains groups by use of functions that can edit privileges of members, the inviting and removing of users, and a description.



Backend System


connections
connections_id INT(11)
user1_id INT(11)
user2_id INT(11)
access TINYBLOB
Indexes
group_members
group_members_id INT(11)
group_id INT(11)
user_id INT(11)
access BLOB
isAdmin BINARY(1)
Indexes
groups
groups_id INT(11)
name VARCHAR(100)
description TEXT
Indexes
gateway
gateway_id INT(11)
user_id INT(11)
email VARCHAR(45)
password VARCHAR(40)
istActive BINARY(1)
Indexes
users
user_id INT(11)
gateway_id INT(11)
primary_email VARCHAR(255)
gender CHAR(1)
hometown VARCHAR(50)
mobile_number VARCHAR(10)
mobile_country_code VARCHAR(4)
picture BLOB
registered DATE
birthday DATE
notifications BLOB
Indexes
user_businessinfo
user_businessinfo INT(11)
user_id INT(11)
business_address VARCHAR(255)
business_number VARCHAR(10)
business_country_code VARCHAR(4)
company VARCHAR(100)
Indexes
user_names
user_names_id INT(11)
user_id INT(11)
first_name VARCHAR(50)
middle_name VARCHAR(50)
last_name VARCHAR(50)
Indexes
user_education
user_education_id INT(11)
user_id INT(11)
high_school VARCHAR(100)
college VARCHAR(100)
Indexes
additional_emails
add_email_id INT(11)
user_id INT(11)
email VARCHAR(255)
Indexes
additional_numbers
add_number_id INT(11)
user_id INT(11)
number VARCHAR(10)
number_country_code VARCHAR(4)
number_index INT(11)
Indexes
social_media
social_id INT(11)
user_id INT(11)
media_link VARCHAR(255)
Indexes
user_homeinfo
user_homeinfo_id INT(11)
user_id INT(11)
home_address VARCHAR(255)
home_number VARCHAR(10)
home_country_code VARCHAR(4)
current_city VARCHAR(50)
Indexes