

```

// DataDocumentation

// Comments on posts data scheme
const CommentSchema = new Schema({

  _post: { type: Schema.Types.ObjectId, ref: 'Post' }, // type :
comment
  _sender: { type: Schema.Types.ObjectId, ref: 'User' }, // user id
associated
  text: String, // The actual comment text
  dateGiven: Date // Date when the user comments

});

// Friendship (ie connected user to another user) scheme

const FriendshipSchema = new Schema({

  _sender: { type: Schema.Types.ObjectId, ref: 'User' }, //person
sending friend request
  _receiver: { type: Schema.Types.ObjectId, ref: 'User' }, // person
receiving friend request
  dateGiven: Date, // date stamp of request
  dateAccepted: { type: Date, default: null }, // date accepted
  status: { type: Number, default:0 } // confirmed (1) or denied (0)

});

// Post scheme

const postSchema = new Schema({

  _user: { type: Schema.Types.ObjectId, ref: 'User'}, // type: post,
OP (original poster) user id attached to post
  description: String, // description of post
  image: String, // panorama
  location: String, // location to of where picture was take
  props: [PropSchema], // prop type
  comments: [CommentSchema], // allow comments
  dateCreated: Date, // date stamp
  lastInteracted: Date // last edited basically *

});

// Prop scheme

const PropSchema = new Schema({

  _post: { type: Schema.Types.ObjectId, ref: 'Post' }, // type post
  _comment: { type: Schema.Types.ObjectId, red: 'Comment'}, // type

```

```

comment
  _sender: { type: Schema.Types.ObjectId, ref: 'User' }, // type:
user: sender
  dateGiven: Date // date stamp
})

// Recipient scheme

const recipientSchema = new Schema ({
  email: String, // recipient user email
  responded: { type: Boolean, default: false } //response
});

// Survey scheme

const surveySchema = new Schema({
  title: String, // title
  subject: String, // subject
  body: String, // body
  recipients: [RecipientSchema], // who will recieve survey
  yes: { type: Number, default: 0 }, // yes response
  no: { type: Number, default: 0}, // no response
  _user: { type: Schema.Types.ObjectId, ref: 'User'}, // user
  dateSent: Date, //date stamp
  lastResponded: Date // last interaction
});

// User scheme

const userSchema = new Schema({
  email: String, // user email
  facebookId: { type: String, default: null }, // if they signed
up with / linked facebook account
  googleId: { type: String, default: null}, // if they signed
up with / linked google account
  firstName: String, // user first name
  lastName: String, // user last name
  description: { type: String, default: null} , // bio basically
  language: String, // preferred language
  credits: { type: Number, default: 0 }, // credits (points,
think of reddit karma)
  private: { type: Number, default: 0 } // is this user private
or public
});

```

