Project 3 Problem Analysis

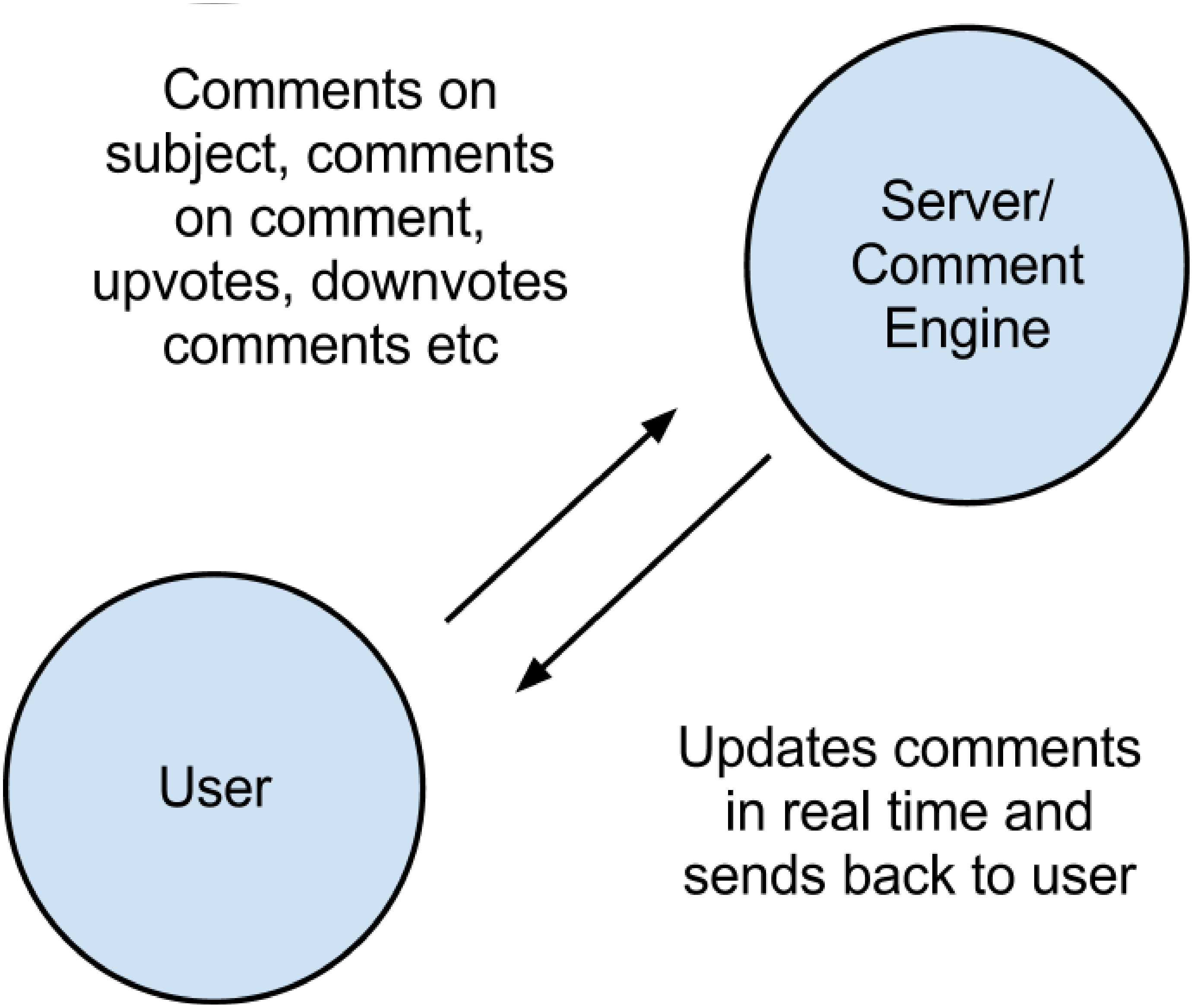
Overview

The goal of this project is to build a working comment thread app. I’ve thought long and hard about what the “essence” of a comment app like this is. I’ve spent some time playing with sites like reddit, quora, and stack overflow.

The goals of the comment system are:

- users asynchronously add comments to an existing page,
- comments are then upvoted (and maybe also downvoted) by other users.
- Typically comments are sorted according to their votes, so that the most popular are seen first.
- Users can see comments being updated in real time

Context Diagram


Comments on subject, comments on comment, upvotes, downvotes comments etc
Server/ Comment Engine
User
Updates comments in real time and sends back to user

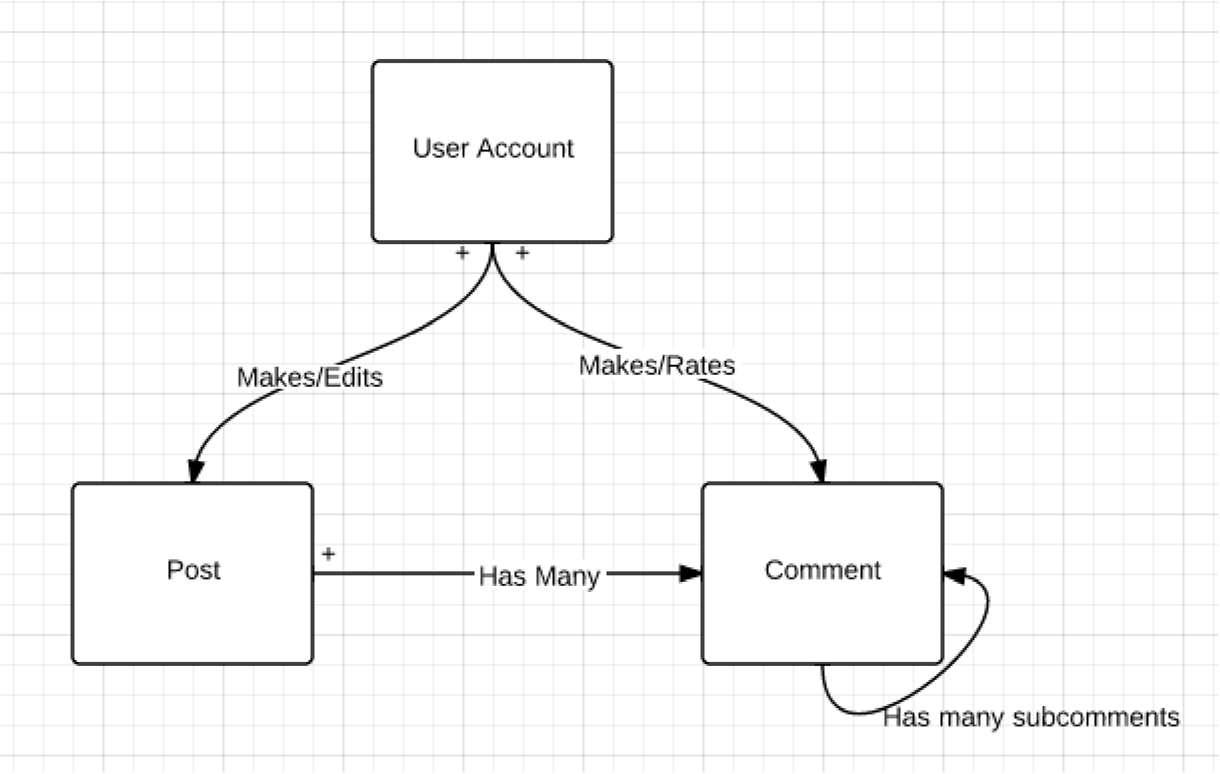
Unusual Requirements

The only thing I can think of for this is that we have to update the comments in real time. One could achieve this by having the client constantly poll the server for updates - but thats an ugly solution. Instead I'll use Pusher.com to push data to the clients in realtime. Pusher.com is basically a websocket api.

Key Features

- Basic Comment functionality
    - comment on posts
    - comment on comments
    - upvote comments
    - downvote comments
    - see updates in real time

Object Model



Event Model

User Account ::= (create (login |logout)* close)
Post ::= (create) (edit)*(destroy))
Comment ::= (create (upvote | downvote)* delete)

Feature description

I am creating a working comment thread app. Basically users can post about things, and then comment on the posts, comment on the comments and then rate the comments. The updates are pushed to the users in real time.

Security Concerns

I’m not too concerned with security holes in this app.