Q&A Web Application

Adam Frisby

December 13, 2013

Advisor: Hongli Lou

CPET 491 Instructor: Paul Lin

1

Project Outline

- Problem Statement and Solution
- Application Overview
- Programming Language and Software
- Application Infrastructure
- Software Architecture
- Security Concerns
- Ranking Algorithms
- Testing
- Conclusion

2

Problem Statement and Solution

- Problem: When large amounts of users are involved, contributions by users are disorganized. With most social media platforms, it is difficult to gauge the general vibe of the audience.
- Solution: New platform which allows ranked communication. Users will voluntarily host Interviews. Other users of the application will have the ability to vote up or down on questions and questions submitted.

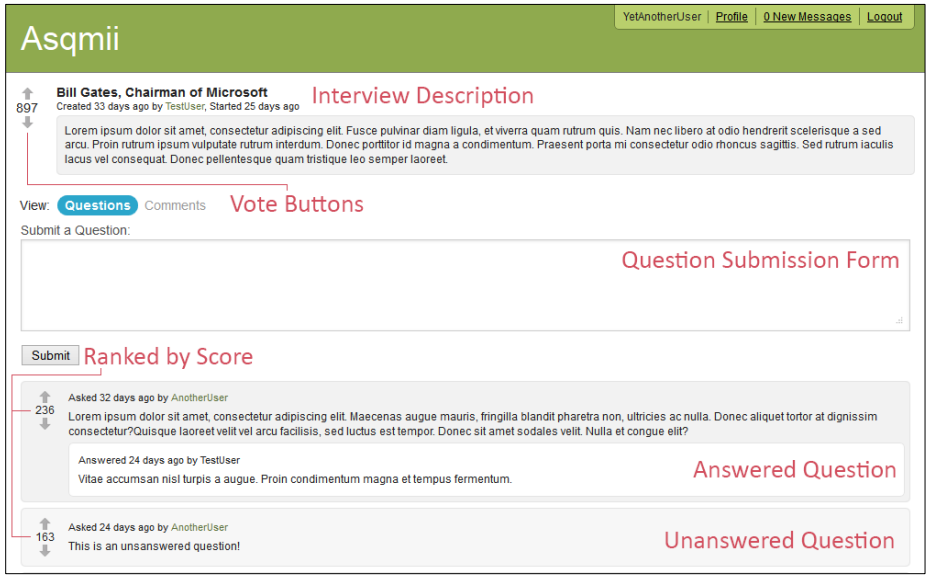
3

Application Overview

- User System
  - Profiles
  - Moderators
- Interview System
  - Questions & Answers
  - Comments
- Voting Systems
- Interview Categories
- Messaging System

4

Application Overview



5

Programming Language and Software

- Server
  - Go (golang) – Garbage collected, compiled language developed at Google
  - Revel – Go web framework
  - PostgreSQL – Open Source Database Management System
- Frontend
  - jQuery – JavaScript library used to simplify JavaScript

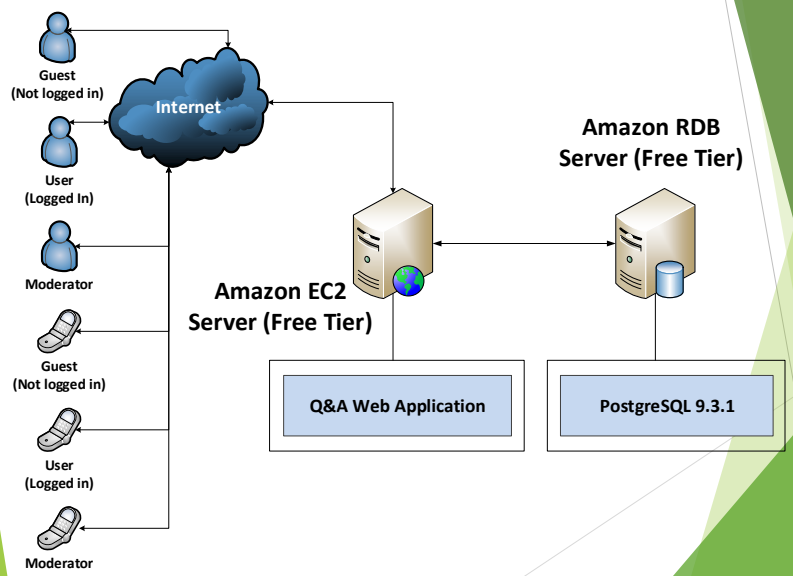
6

Cloud-Based Application Infrastructure

- Amazon EC2 t1.micro (free tier) instance [1]
  - OS: 64 bit Linux (Amazon Linux AMI)
  - Memory: 613 MB
- Amazon RDS db.t1.micro (free tier) instance [1]
  - Engine: PostgreSQL 9.3.1
  - Storage: 5 GB
  - Memory: 613 MB
- Shared Bandwidth: 15 GB/Month

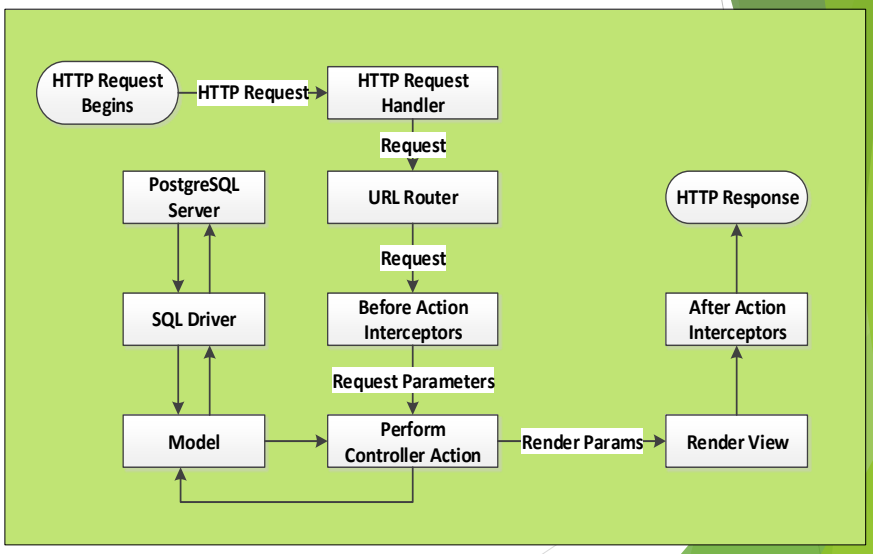
7

Network Diagram



8

Software Architecture



9

Controllers & Actions

- Each controller in the application contains several actions
- Examples:
  - "Register.Submit" – Validates the user's input and creates a new user account if user input is valid
  - "Interviews.View" – Pulls interview data from database and returns a view of the to the user
- Most actions return views, but some actions return data objects (in JSON format)

10

Interceptors

- An interceptor is a function which is called before or after a controller action is invocated.
- There are four times at which interceptors can be registered to run: BEFORE, AFTER, PANIC, FINALLY
- Interceptors are used to being/committing/rolling back database transactions and enforcing user permissions.

11

URL Router

- URL Router routes certain URLs to certain controller actions
- For example http://site.com/register routes to the "Register.Index"

```
GET     /logout          Logout.Index
GET     /register        Register.Index
POST    /register        Register.Submit
```

12

Models

- Models handle the interaction with the database.
- InsertUser(tx *jet.Tx, username string, password string, email string, ipAddress string)
- When a row is requested from the database through the model it is returned as a struct

13

Views

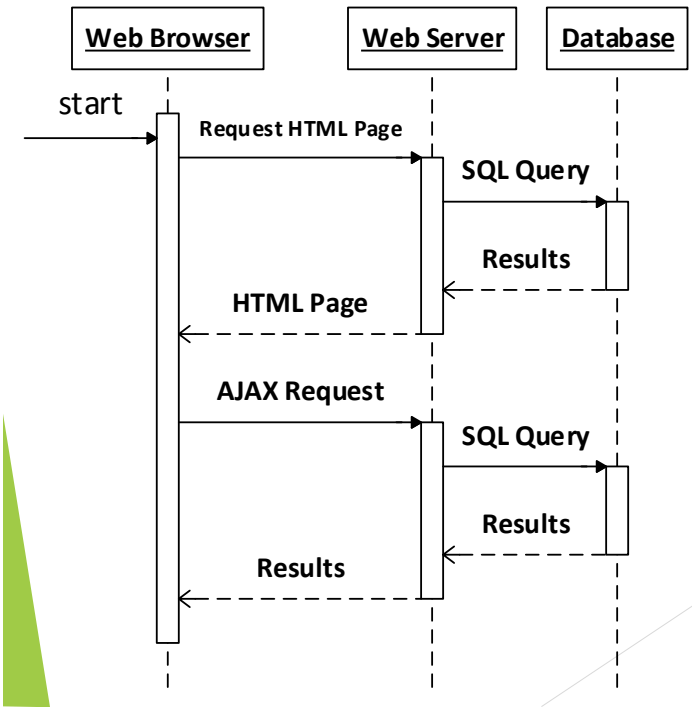
- In the context of the web application, views are the web pages which are displayed in the user's web browser.
- The application makes use of Go's built in template library.

14

AJAX

- AJAX - Asynchronous JavaScript and XML
- XML is not required, JSON is used instead
- A request is made from the web browser in the background. The server will respond with a response and the page (view) is updated to reflect the results

15



16

Security Considerations

- Passwords are stored as cryptographic hashes using the bcrypt hash function
- Bcrypt has a "cost" which will increase the execution time of the function as well as using random salts
  - Longer time to brute force
  - Protected against rainbow tables

17

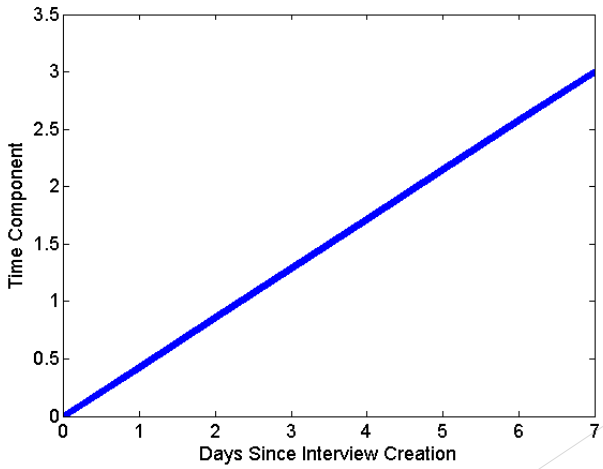
Ranking Algorithms

$$rank_{hot} = \begin{cases} \left(\frac{\Delta t}{k}\right) + log(1+s), & s \geq 0 \\ \left(\frac{\Delta t}{k}\right) - \frac{\left(10^{\left(\frac{|s|}{10}\right)}\right)}{10}, & s < 0 \end{cases}$$

Δt = Number of seconds between Jan 1, 2013 and interview creation [0, ∞)
k = 201570.0
s = Score [-∞,∞]

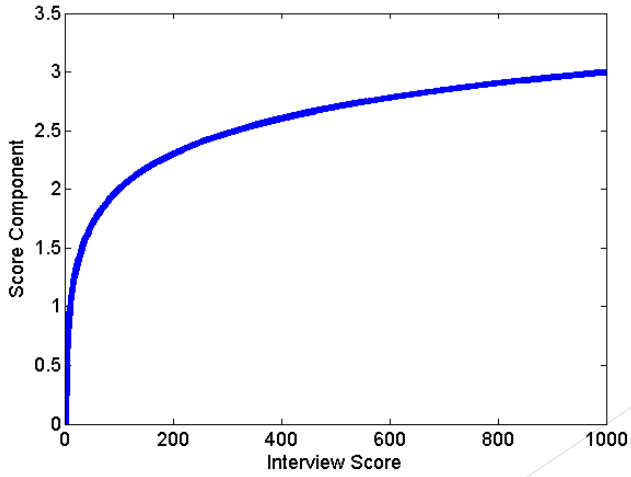
18

Ranking Algorithms – Time Component



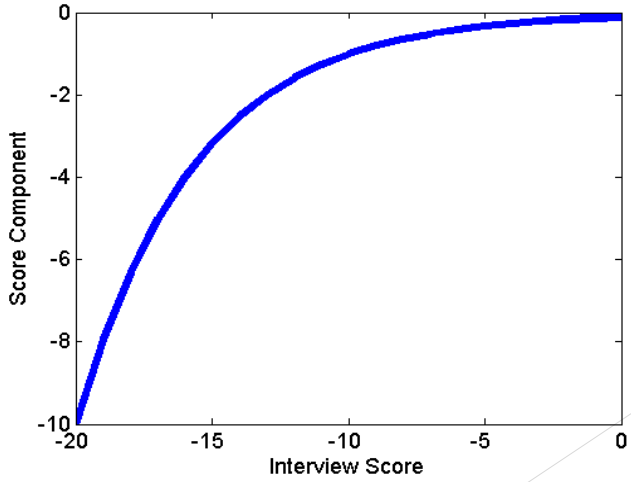
19

Ranking Algorithms – Positive Score Component



20

Ranking Algorithms – Negative Score Component



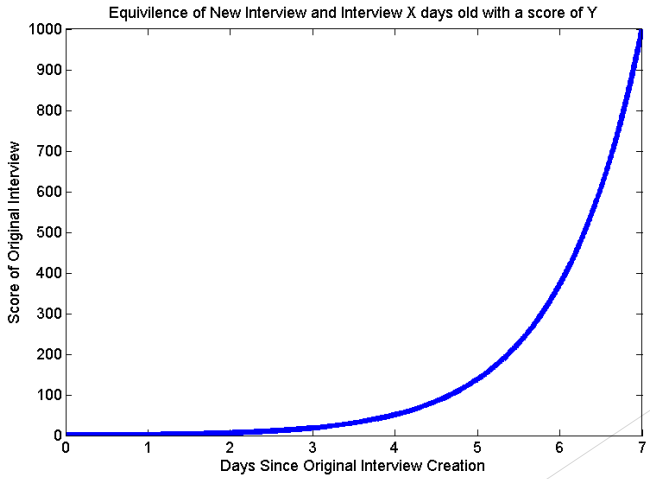
21

Ranking Algorithms

- Deriving k
  - Interview 1 week old with 1000 post is equivalent to a newly created interview
  - $\left(\frac{0}{K}\right) + \log(1 + 1000) = \left(\frac{604800\ s}{k}\right) + \log(1 + 0)$
  - $\log(1001) - \log(1) = \frac{604800\ s}{k}$
  - $k = \frac{604800\ s}{\log(1001) - \log(1)} = 201570.8342 \approx 201570$

22

Ranking Algorithms



23

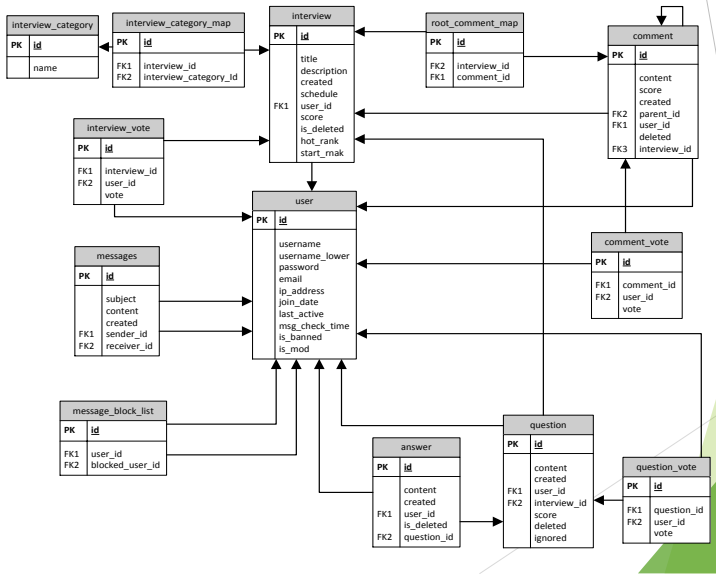
Ranking Algorithms

$$rank_{soon} = \begin{cases} \left(\frac{\Delta t}{k}\right) - log(1+s), & s \geq 0 \\ \left(\frac{\Delta t}{k}\right) + \frac{\left(10^{\left(\frac{|s|}{10}\right)}\right)}{10}, & s < 0 \end{cases}$$

Δt = Number of seconds between Jan 1, 2013 and scheduled time [0, ∞)
k = 201570.0
s = Score [-∞,∞]

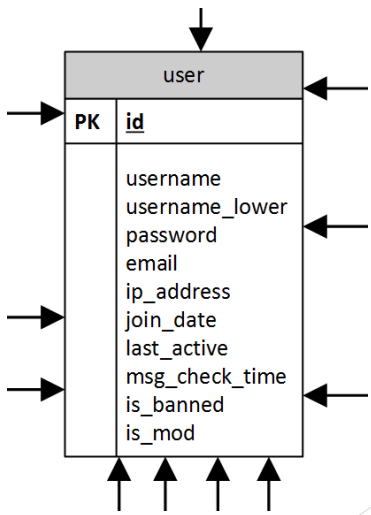
24

Database Schema



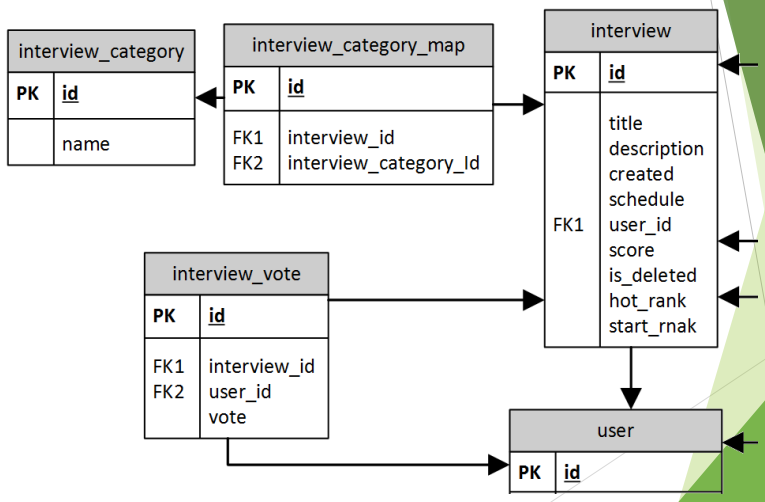
25

Database Schema - Users



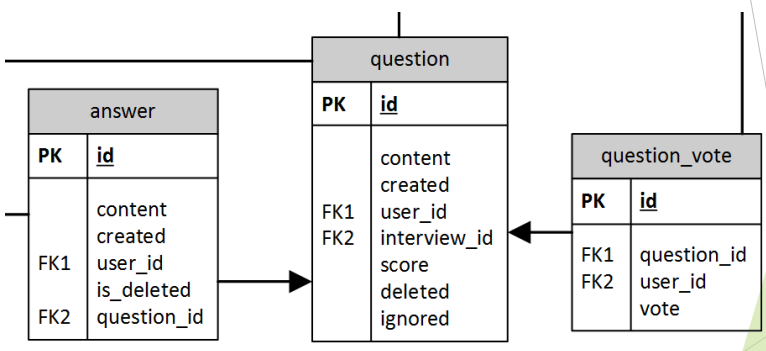
26

Database Schema - Interviews



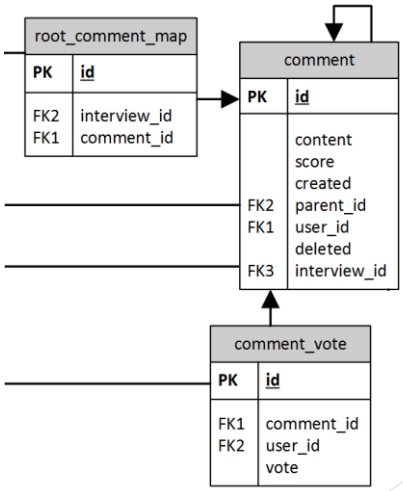
27

Database Schema - Questions



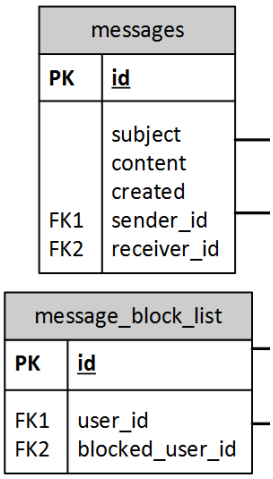
28

Database Schema - Comments



29

Database Schema - Messages



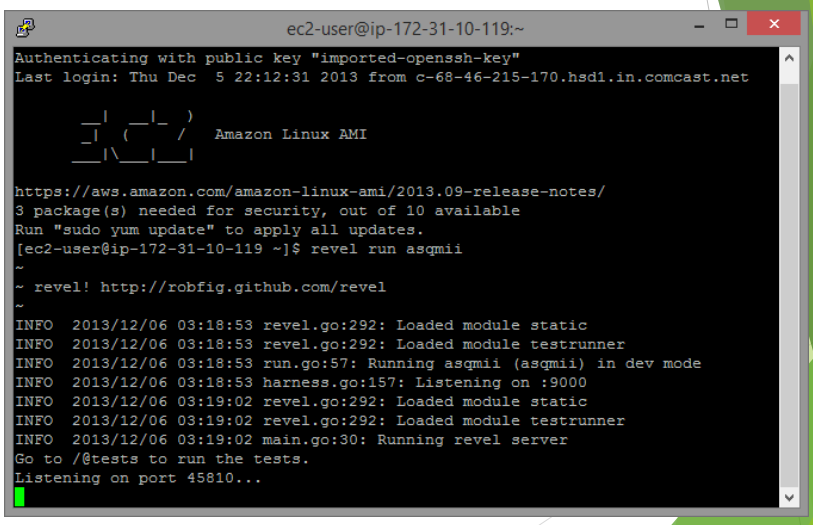
30

Select Requirements

ID	Requirement	Method	Result
F8	Application shall allow visitors to register user accounts	Demo	Pass
F9	Users shall have profile with questions/answer/comment history	Demo	Pass
F10	Application shall allow user to schedule a Q&A session	Demo	Pass
F18	Application shall allow user to submit questions before & during Q&A session	Demo	Pass
F21	Comments shall allow children comments	Demo	Pass
F23	Application shall rank Q&A sessions by votes	Demo	Pass
F24	Application shall allow user to vote up or down on questions	Demo	Pass
F26	Application should allow certain users to be moderators	Demo	Pass
F31	Application should allow users to private message each other	Demo	Pass
F32	Application should allow users to block members from messaging them	Demo	Pass

31

Running Web Server



32

Performance Testing

- Loading Home Page Performance Test
  - Script was used to open 5 instances of Firefox and load the homepage, delay 5 seconds, and repeat
  - Ran for 10 minutes
  - Analyze: Application should handle about 40 concurrent users reloading every 5 seconds.
  - Limiting factor: Web Server CPU usage.

33

Performance Testing

- Scheduling Interview Performance
  - Script was used to open 5 instances of Firefox and load the “Schedule an Interview” page, fill the form with valid data, submit the form, delay 5 second, and repeat
  - Ran for 10 minutes
  - Analyze: Application should handle about 20 concurrent users scheduling interviews every 5 seconds.
  - Limiting factor: Database CPU usage.

34

Cost

One-Time Costs

Item	Price
Software Tools	$0.00
Software Libraries	$0.00
Total Cost:	**$0.00**

Reoccurring Costs

Item	Price
t1.micro EC2 Instance [2]	$23.00/year
Elastic IP [3]	$0.00/year
Micro Standard Deployment RDS PostgreSQL Instance [4]	$23.00/year
Total Cost:	**$46.00/year**

35

Knowledge Gained

- Go - Learned much about the language including the syntax, common design patterns of Go, and development environment.
- Git - Use of command line Git to initiate, clone, pull, and commit as well as the use of Go's get command to clone Github repositories

36

Lessons Learned

- Time Management
  - Project encompassed a wide range of activities
  - Begin early
  - Keep at it

37

Conclusion

- Project meets all requirements set in Phase I of Senior Design
- Possible Future Improvements
  - Caching
  - Social Media Integration
  - Spam Protection & Email Verification
  - Private & Group Interviews
  - Notifications and Alerts

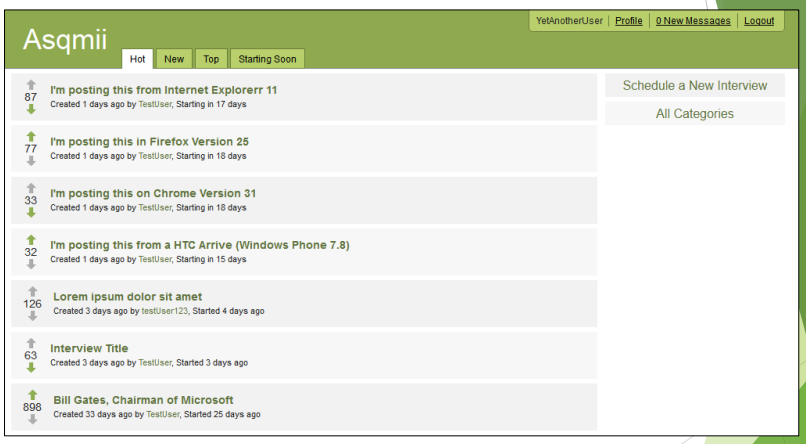
38

References

- [1] "AWS Free Usage Tier," Amazon, [Online]. Available: http://aws.amazon.com/free/. [Accessed 4 December 2013].
- [2] "Amazon EC2 Pricing," Amazon, [Online]. Available: http://aws.amazon.com/ec2/pricing/#reserved. [Accessed 1 December 2013].
- [3] "Amazon EC2 Pricing - Elastic IP Addresses," Amazon, [Online]. Available: http://aws.amazon.com/ec2/pricing/#elastic-ip. [Accessed 1 December 2013].
- [4] "Amazon Relational Database Service Pricing," Amazon, [Online]. Available: http://aws.amazon.com/rds/pricing/. [Accessed 1 December 2013].

Questions?

Demonstration



URL: http://54.213.46.161:9000/

41