Software Design Specifications for IPhO-2015 Server system

Kumar Ayush, Sandesh Kalantre, Sharad Mirani

Contents

1 Introduction

The server system is used to present an ip based website on which tasks such as setting a question and voting over it can be performed.It also includes features to upload files to a user specific directory from where the user has the liberty to print the files.

2 Database

MongoDB was used as data server for the server system.The data is stored in the server in collections:

- **users** This collection contains the user data which is used for authentication during login. The data is organised as the following fields:
  - `ip`: IP address of the client
  - `pass`: Password
  - `logged`: Boolean which stores the current status(logged in/logged out) of the user
  - `type`: The type of the user.The system has two kinds of users:0 for a super-user who can set questions;1 for normal user who can vote and upload files as well as print them.
- **voteq** This collection contains the body of the questions asked and the options. The data is organised as the following fields:

  - `id`: A 5 lettered randomly generated id of the question.
  - `op_i`: The body of the ith option.

- **votec** This collection contains the id of the questions asked and number of votes received for each of the options. The data is organised as the following fields:
  - `id`: A 5 lettered randomly generated id of the question.
  - `op_i`: Number of votes received for the ith option.

3 HTML Pages

The system contains two homepages for the users and superusers located at `/u/index.html` and `/su/index.html`.Also a `auth.html` file is present at `/auth.html`.

4 The IO:Server and Client

The io between the server and the client is managed by socket.io signals Signals used in the program:

- **syn**

  Signal Cause: 'syn' signal is sent when the client is synchronised with the server and the user submits a password in auth.html.

  Signal Action: On receiving this signal,the server connects to the MongoDB server and retrieves the 'users' collection.The user's ip sent with the signal is first checked of its precense in the database.If the ip is not present **no response** is sent. If the ip is present,the true-pass(password) is checked with the pass sent by the user.If the password is correct,then the logged property of the user is set to true and a 'fin' signal is sent to signify that the syncing is complete.If the password is incorrect a 'syn-err' error signal is emmited.

- **syn-err**

  Signal Cause: This signal is sent when the user's ip is present in the database but the password submitted by the user is incorrect.

  Signal Action: On receiving this signal,auth.html adds to the password input box,the class "has-error".

- **fin**

  Signal Cause: This signal is sent when the user's ip is present in the database and the password entered by the user is correct.

  Signal Action: On reciving this signal,auth.html redirects the browser to the '/' page.

- **end**

  Signal Cause: This signal is sent when the logout button on the index.html page is clicked.

Signal Action: On receiving the signal,the server connects to the MongoDB server and retrieves the 'users' collection.The server upadates the logged property of the ip of the user to false.In the end the server emits a 'end-ack' signal.

- **end-ack** This signal denotes that the end was acknowleged by the server.

  Signal Cause: This signal is sent when an 'end' signal is received by the server.

  Signal Action: On receiving this signal,the index.html page redirects the user to the '/auth.html' page.

- **setvote**

  Signal Cause: This signal is sent by the su's index.html page.Along with the signal, the body of the question,the options and time are sent as well.

  Signal Action: On receiving this signal,the server generates a random 5 alphanumeric id of the question on which the question is stored in the database.The server then connects to the MongoDB server and retrieves the 'voteq' collection.It inserts the body and options of the question in the collection.The server then **broadcasts** a 'govote' signal along which it sends the id,body,options and time of the question.

- **govote**

  Signal Cause: This signal is **broadcasted** by the server after storing the question in 'voteq' collection on receiving the 'setvote' signal.

  Signal Action: On receiving this signal,the index.html of the user runs a timer for the time recieved with the signal.On end of the counter,the 'logvote' signal is emitted along with the id of the question and the options selected(1 for selected and 0 for not selected).

- **logvote**

  Signal Cause: This signal is sent by the index.html of user when the timer for a question has expired.

  Signal Action: On receiving this signal,the server connects to the MongoDB server and retieves the 'voteq' collection.It updates the corresponding options in the collection according to the users' responses.

- **voteresults**

  Signal Cause: This signal is sent from the server after a delay of 5000ms after the time of the question has expired.It signifies that the results must be now displayed.Along with the signal,the number of votes received and the options themselves are also sent.

  Signal Action: This signal causes the respective index.html pages of both users and super-users to display the results in the form of a histogram using chartjs.

5 File Upload

File uploads are managed by the npm package `multer`.When the user uploads a file, the file is first uploaded to `/tmp` folder with a time-stamp and the client sends a POST request for `/uploaded`. The server on receiving this request copies the file to the user directory located at `/home/ip`

6 Directory Listing

Directory listings are managed by the `serve-index` package.