CS302 Python Project Indicative Marking Checklist 2019

|  |  |  |
| --- | --- | --- |
| **Grade** | **Task/Feature Description** | **Done?** |
| C | **Application runs following README instructions on Ubuntu Linux** | Y |
|  | **User can authenticate against the login server (using /api/ping)** | Y |
|  | **User can see who is currently online (using /api/list\_users)** | Y |
|  | **User can generate a public/private keypair (and submit to /api/add\_pubkey)** | Y |
|  | **User can report connection info (to /api/report)** | Y |
|  | **User can send and receive broadcasts to/from login server and other clients** | Y |
|  | **User participates in network health checks by regularly calling**  **/api/client\_ping on other clients and by serving /api/client\_ping requests** | Y |
| B-/B | Automatically refreshing page (or refreshing content) and/or notifications | N |
|  | Unicode support (including emojis) | Y |
|  | (Good) auto content filtering via lists of blocked words or phrases | Y |
|  | Good use of database(s) | Y |
|  | Use of local encryption/hashing/data security  (e.g. if passwords saved, they are encrypted/hashed) | Y |
|  | User can send/receive private messages | Y |
|  | User can search public broadcasts in some way (e.g. display only broadcasts from certain users, between certain times, that contain certain words ...) | Y |
| B/B+ | Graceful error handling (No ugly 500 error pages) | Y |
|  | Rate limiting on API | N |
|  | Private message interface (e.g. only show messages to and from a certain user, order by timestamp, mechanism to reply) | Y |
|  | (Good) page templating, e.g. using Jinja2 | N |
|  | Good inter-app security, including checking signatures and loginserver\_records to ensure message authenticity | P |
|  | Use of API keys with Login server instead of HTTP BASIC on all requests (i.e. use /api/load\_new\_apikey) | Y |
|  | Manage user status i.e. online/busy/away, including the sending of ‘offline’ to /api/report on sign out/application close | Y |
| A-/A | Retrieve and retransmit “offline” broadcasts and privatemessages (i.e. those sent while not online; implement and call /api/checkmessages) | P |
|  | Local favouriting/blocking of broadcasts/usernames/pubkeys | P |
|  | Markdown support in messages, including display of hotlinked external images (e.g. via ​![A test image](https://………/image.png)​ ) | Y |
|  | High standard of user experience (e.g. no lagging, awkward refreshing) | P |
|  | Attractive, cross-browser UI (e.g. looks the same in chrome/firefox) | P |
|  | 2FA (Two factor authentication) e.g. for keeping private keys safe | N |
| A/A+ | Multiple sessions(users) supported simultaneously | Y |
|  | Group conversations, including creating a group and inviting members, and sending and receiving messages | N |
|  | Receiving and transmitting meta messages for distributed meta information sharing  (e.g. displaying other users favourite messages, blocking a message because your friend blocks it) | P |
|  | Saving/loading private data to the login server for seamless cross-client  compatibility (​ encrypt/save/load/decrypt private data (e.g. keys/etc) to other student’s implementations; implement and call /api/add\_privatedata, /api/get\_privatedata) | Y |
|  | Defence against injection attacks | Y |

**Additional features:**

Cleaning messages to remove HTML tags, some are whitelisted and will not be removed such as <p>,<br/>, etc

User can change their encryption password from account