

# CS302 Python Project Indicative Marking Checklist 2019

Grade	Task/Feature Description	Done?
C	Application runs following README instructions on Ubuntu Linux	<input checked="" type="checkbox"/>
	User can authenticate against the login server (using /api/ping)	<input checked="" type="checkbox"/>
	User can see who is currently online (using /api/list_users)	<input checked="" type="checkbox"/>
	User can generate a public/private keypair (and submit to /api/add_pubkey)	<input checked="" type="checkbox"/>
	User can report connection info (to /api/report)	<input checked="" type="checkbox"/>
	User can send and receive broadcasts to/from login server and other clients	<input checked="" type="checkbox"/>
	User participates in network health checks by regularly calling /api/client_ping on other clients and by serving /api/client_ping requests	<input checked="" type="checkbox"/>
B-/B	Automatically refreshing page (or refreshing content) and/or notifications	<input checked="" type="checkbox"/>
	Unicode support (including emojis)	<input checked="" type="checkbox"/>
	(Good) auto content filtering via lists of blocked words or phrases	<input type="checkbox"/>
	Good use of database(s)	<input type="checkbox"/>
	Use of local encryption/hashing/data security (e.g. if passwords saved, they are encrypted/hashed)	<input type="checkbox"/>
	User can send/receive private messages	<input checked="" type="checkbox"/>
	User can search public broadcasts in some way (e.g. display only broadcasts from certain users, between certain times, that contain certain words ...)	<input checked="" type="checkbox"/>
B/B+	Graceful error handling (No ugly 500 error pages)	<input checked="" type="checkbox"/>
	Rate limiting on API	<input type="checkbox"/>
	Private message interface (e.g. only show messages to and from a certain user, order by timestamp, mechanism to reply)	<input checked="" type="checkbox"/>
	(Good) page templating, e.g. using Jinja2	<input checked="" type="checkbox"/>
	Good inter-app security, including checking signatures and loginserver_records to ensure message authenticity	<input type="checkbox"/>
	Use of API keys with Login server instead of HTTP BASIC on all requests (i.e. use /api/load_new_apikey)	<input type="checkbox"/>
	Manage user status i.e. online/busy/away, including the sending of 'offline' to /api/report on sign out/application close	<input checked="" type="checkbox"/>
A-/A	Retrieve and retransmit "offline" broadcasts and privatemessages (i.e. those sent while not online; implement and call /api/checkmessages)	<input type="checkbox"/>
	Local favouriting/blocking of broadcasts/username/pubkeys	<input checked="" type="checkbox"/>
	Markdown support in messages, including display of hotlinked external images (e.g. via ! [A test image] (https://...../image.png) )	<input type="checkbox"/>
	High standard of user experience (e.g. no lagging, awkward refreshing)	<input checked="" type="checkbox"/>
	Attractive, cross-browser UI (e.g. looks the same in chrome/firefox)	<input checked="" type="checkbox"/>
	2FA (Two factor authentication) e.g. for keeping private keys safe	<input type="checkbox"/>
	Multiple sessions(users) supported simultaneously	<input checked="" type="checkbox"/>
A/A+	Group conversations, including creating a group and inviting members, and sending and receiving messages	<input type="checkbox"/>
	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)	<input type="checkbox"/>
	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)	<input checked="" type="checkbox"/>
	Defence against injection attacks	<input checked="" type="checkbox"/>

**All inter-application features must be well-tested. In your README you must list at least two other students (their UPI) who have clients that your client works with.**

**You may test transmission of broadcasts and privatemessages with the login server, but this alone may not guarantee your mark (as you must also be able to receive broadcasts and privatemessages, and the login server will never transmit a message or a broadcast).**

**You must complete all the requirements for the C grade (minimum requirements, in bold) to be eligible for consideration for any higher grade.**

Other grades will be awarded based on the number of completed features and their relative complexity, as indicated by the table. In general, completion of 4 features in each grade bracket will indicate eligibility for that (and below) grades. Items of a higher grade may be substituted for items of a lower grade. (I.e. if you completed all items in the A-/A and A/A+ grade section, it would not be necessary to complete items in the B-/B and B/B+ area).

Of course, how *well* a feature is implemented will also affect the final mark.

**Please note that this marking checklist is indicative only, and may still change even after the final deadline if there are other features identified or if rebalancing the checklist is needed.**

**You may implement things that are not on this checklist, consult with a TA to discuss what other features may be worth.**