# COMPSYS302 Phase II Project Indicative Marking Checklist

|  |  |  |
| --- | --- | --- |
| **Grade** | Task/Feature Description | **Done?** |
| **C** | Application runs following README instructions on Linux | **yes** |
|  | User can log in | **yes** |
|  | User can see who is currently online | **yes** |
|  | User can connect with another user and the games start | **yes** |
|  | Game multiplayer functionality works correctly | **yes** |
| **B/B-** | Use of encryption when interacting with the login server | **yes** |
|  | Use of encryption/hashing/data security within the application |  |
|  | Celebration screen upon victory (in the browser) |  |
|  | Show opponent’s connection health (based on round-trip ping time) |  |
|  | Friends list | **yes** |
|  | Passphrase to play |  |
|  | Game message validation on slave side |  |
|  | Logs out from the login server upon application exit | **yes** |
| **B+/B** | Modular and Pythonic code, including commenting and documentation | **yes** |
|  | Some effort towards a nice web user interface and/or themes | **yes** |
|  | Automatically refreshing page (or refreshing content) and notifications | **yes** |
|  | Use of threading for communicating with login server regularly | **yes** |
|  | Good use of database(s) |  |
|  | Show user status (online, away, offline, etc.) |  |
|  | Different game modes/options/rules |  |
|  | Java game embedded in browser |  |
| **A** | Profile pictures, user profiles, and instant messaging |  |
|  | P2P Networking if login server goes down |  |
|  | Use of effective inter-app encryption/hashing/data security |  |
|  | Nice user interface (responsive and compatible cross-browser) | **yes** |
|  | Evidence of Unit Testing |  |
|  | User Analytics (statistics, rankings, reputation) |  |
|  | (Good) Contribution towards developing class protocol | **yes** |
|  | Fails graciously when interacting with substandard clients | **yes** |

**All inter-app features must be supported by at least one other node/client as well.  
You must list at least one supported client (give the group number) in your README.  
Any additional features must not interfere with minimum requirement features.**

**You must include a copy of this table with your submission**, with the items that you think you have completed marked as done.

**You must complete all the requirements for the C grade (minimum spec) to be eligible for consideration for any higher grade.** To get the grade indicated on the left, you must complete at least **four** of the items from the appropriate grade. Items from a higher grade can be used to substitute for items in a lower grade (i.e. if you have three items in B/B- category and five items in the B+/B category, one of the B+/B category items can be counted towards the B/B- category).

To qualify for the item in the higher grade, it must be beyond what is necessary for the minimum spec. For example, within-app hashing is required to login to the login server – you must demonstrate additional encryption/hashing/data security beyond that.  
Please note that this marking checklist is indicative only, and may still change after the final deadline if there are other features identified or if rebalancing of the checklist is needed.