Y2S1 Client Project – Working Agreement

General

Communications

* All project-based communications will be done through Teams

Definitions

Ready

* Issue is populated with relevant Acceptance Criteria
* Issue name meets the format “As a <role> I want <feature> so that <reason>”
* Issue is not dependant on any, non-closed, issue
* Issue appropriately weighted based on team opinions and planning poker session(s)
* If issue has a corresponding UI element, it has a “UI Plan” created before development begins
* If Issue has a corresponding DB element, it has an ER Diagram created with corresponding changes before development begins
* All above points verified by a member of the team external to the main developer(s) of the issue

Done

* At least one automated test per major feature created during the process of the issue
* All acceptance criteria complete and met
* All relevant tests pass
* Appropriate Javadoc present throughout
* All above points verified by a member of the team external to the main developer(s) of the issue

Working Days

* Team meetings on Monday afternoons – after client/staff catch up meetings
* As many team meetings as possible will be done in-person, either in Abacws or NSA if available/based on preference
* Any team members who cannot make an in-person meeting will join remotely if available
* All team members will take at least one day off a week
  + In addition to weekends which are, optionally, time off for all

Version Control

Branching

* “master” branch will be for deployable code only
* Code from a completed issue, but is not yet deployable, will be placed into a “develop” branch
* Issue Branching
  + Code relating to each issue will be on its own respective branch
  + Branches are to be created off of “develop” only when the corresponding issue meets the requirements for “ready” status
  + Branches are to be merged back into “develop” only when the corresponding issue meets the requirements for “done” status
  + Branches are to be named based on the issue they correspond to - e.g. “Issue1”
* After an issue’s branch is merged back into “develop” the other team members will be updated so they can merge develop into their own branches – will reduce the likelihood for Merge Issues later on
* Branches are not deleted on merge, as they could be useful to go back over at a later point

Tagging

* Whenever a branch is merged into “develop”, a tag with a relevant name is created at that commit – e.g. “Issue1Complete”

Code Practices

Database

* Naming conventions
  + Table names start capitalised, and are plural nouns
  + Field names follow typical conventions (lower case with underscores)
  + Foreign keys are the same name as their respective key in their original table
* JDBC vs JPA
  + JPA will be used where possible, for easier readability of code
  + If a query is too complex, or JPA cannot perform the task, then JDBC can be used instead
* Security
  + All sensitive fields, such as passwords and addresses, will be encrypted before entering the database

Entity objects

* All tables have a corresponding Entity Class created and mapped appropriately
* Entity objects will be used wherever preferable to Domain – easier for linking to the database, and makes adapters not necessary

File Structure

* File names and structure will follow a similar format to that of Carl’s “Tutorial Companion” project from his Codelabs – easily readable as to where everything should be stored

Javadoc

* All classes and methods have appropriate Javadoc implemented to increase readability of code

Logging

* All errors logged
* Successful and failed login attempts, it is logged – “user\_id” value, IP Address, timestamp

Login Process

* If required, login will be completed with email and password
* Password will be encrypted, and then checked against encrypted value

UI

* All pages share relevant “nav-bar” and “footer” fragments
* If a framework is required, “Bulma” will be used as opposed to “Bootstrap”