

Individual contribution report – SAW Accommodation

Ashley Pearson - 20001030

From the outset of the ASD project our team agreed a clear outline as to the areas we would individually focus on as a priority based on our previous experiences, strengths and areas of interest. My primary focus was class design and the implementation of object-oriented programming (OOP); throughout however, team members were in open communication and working across all areas.

Initially I built upon the classes found within our UML diagrams and began creating outlines of classes within a python file, many functions simply 'passed' but I believe it was critical to put the classes into code to see how they would interact together and if the basic paradigms of OOP were being met. A small number of changes were made at this point from our initial proposals, for example I decided that given both 'TheWarden' and 'HallManager' inherit from 'User' it would be redundant for both classes to have separate ID and name attributes given they are both the same values, likewise, the 'getHalls' function was moved to 'User' as both classes would return the same list, again reducing redundant code.

Once a structured outline of classes was available this was shared with the team for feedback, who had themselves made progress in their respective areas. A small number of semantic errors were highlighted but the plan was given the go-ahead to continue. At this point I began to flesh out the created classes and work alongside the barebones GUI to ensure that appropriate functions such as 'getters' worked as required, having rarely worked with GUI programming before this was both challenging and insightful, the need to have instances and attributes created, displayed, edited and removed in real time required knowing how to correctly assign data to the mentioned objects and how to link them together in meaningful ways.

Creating and improving the Treeview of the GUI was an interesting and time-consuming part of the project undertaken by a colleague, it was however initially filled with dummy data as proof of concept, for us to then work together to combine our understandings to populate using class data. This required using several Tkinter modules, rethinking getters and creating bind functions so that data live updated without the need for button clicks by the end user. All team members worked on this concurrently, with my focus being on the interactivity between attributes such as 'room number' and 'student name', as these belonged to separate classes, it was necessary that when a 'room' object had a 'lease' object assigned to it a 'student record' object was also instantiated, for the GUI to function as desired this needed to happen in real time and required constant communication to ensure one team members changes didn't break other implementations.

As well as my focus on class implementation I also assisted with the multi-platform experience of the GUI, as Tkinter utilises platform assets, the Mac experience can be quite different to that of Windows, creating a jarring experience between the platforms. To counter this, my colleague built the GUI in Windows and I worked fixing graphical errors for the Mac platform, creating as uniform of an experience as possible. I also felt that we

needed to add an element of privacy to the password field and implemented a barebone password reveal system.

Due to my deeper understanding of the classes within the system I implemented self-updating room info displayed upon the selection of a hall in the upper right-hand corner of the GUI. We felt as a team that if the system were to scale to hundreds of rooms per hall, calculating the number of free rooms could be a time consuming task for the end user and as such implemented this, this text is automated behind the scenes using class setters and getters and works on creation and deletion of rooms as well as the occupancy status of a room (which itself is automated based on the 'hasLease' Boolean check.

Due to working throughout the Christmas period and across the globe, I feel that the most invaluable experience gained from this project is the that of working as part of a remote scrum team with daily meetings. The experience of coordinating and merging multiple daily code changes into a comprehensive final project was a very different experience to a solo project and introduced a new host of challenges and tactics.