

## **AGILE PRACTICES**

During the project lifecycle, as a group we decided to take an agile approach to the development process using the scrum method. What made this approach successful for us was the iterative nature within the project lifecycle. In comparison with the waterfall method where you must tackle each phase in the development lifecycle from the requirements all the way to the maintenance/transition phase in sequence, the scrum method allow us to be more flexible by making smaller iterations where we could incrementally add more functionality which allowed us to review any changing requirements within the project and update/fix errors during the project lifecycle.

The waterfall method for us was too rigid and did not give the option of going back to a previous phase. This meant that once you have identified the requirements of the system and have moved onto the design and build phases, if the requirements change you are unable to re-evaluate these until the project has reached the end of the cycle. As specifications can be mis-interpreted the scrum method allows you to identify these changes and within each iteration during the lifecycle they can be reviewed and acted upon much earlier. Due to the fact sections of functionality are being implemented iteratively the changes to be made to the system should be more manageable and easier to implement in comparison to the waterfall method where integrating new functionality from changing requirements could be more problematic with a fully built system.

From the outset of the project we broke down the specification given to us into small chunks which could be implemented either as functions or even more broadly as whole classes. We then identified and created an excel spreadsheet containing the sprints to be used in the project. These contained a time duration and priority level and catalogued the structure in which we were to undertake the project from beginning to end. As a team we identified areas within each sprint to tackle and collectively worked on individual elements. The initial phases were for system design and test case specifications. This included Use case diagrams for the actors interacting with the system, class diagrams for the system structure and functionality and a sequence diagrams to show the relationship with the actor and the system in a sequential manner. The test case specification was created to check that all functionality identified from the project specification was implemented correctly with the relevant error messages provided and would be utilised to debug towards the end of the project lifecycle.

After this the database, lease data class, main model classes including hall, room and student, and GUI base class were created to allow for the overall main functional components to be implement from an early stage. This allowed us to ensure the database was connected correctly and gave us a design for the GUI for which we could use to implement further windows/panels for display. The various windows for admin panel, browser, view permissions, view lease were created utilising the GUI base class along with the CSS stylesheets. From here the table view was created for the browser to display the information stored in the database for the end user. This completed the GUI element of the

main entry window. Further elements were added to complete the GUI aspects for admin panel and view lease windows next.

Once the main GUI elements were implemented the next phase was to implement the functionality between the GUI and the database data by creating handlers which linked with the lease data class. This gave the ability to create, edit and delete lease data. After this we added user permissions to give specific privileges to our end users using the system. Once complete the rest of the project and final sprint was dedicated too adding extra user functionality with the addition of items such as, right click button to access system options and search panel etc. Testing the system for any bugs that may have been missed also happen in this sprint as well as ensuring the code was fully commented for any future changes that may be made.