**Meeting Agenda & Minutes**

Wed 14/02/18, 11.30, The Zone at Ellison Building

**Agenda**

* Review ERD and database system
  + Discuss and decide upon appropriate triggers
  + Concurrency issues
* Review all UI diagrams
* Develop database
* Review Case descriptions
* Review class diagram
* Talk about next steps, tasks and deadlines

**Attendees**

Peter Smith (Scribe), Sam Connelly (Chair), Max Walsh

**Apologies**

Michael Bulgrass

**Minutes**

Meeting opened at 11:40 with SC discussing progress from the tutors. Example data has been present on the discussion boards and had already been looked at and considered.

**UI Diagrams**

MW demonstrated the UI diagrams that he had sketched, asked for opinions and discussed his thoughts. He wanted to make use of 3 sub-windows within the main manager screen to limit pop-ups and to show the majority of the required data on one screen to reduce the need for clicking onto different screens. He suggested using filters to limit the displayed data, allowing the user some freedom to select and view the required data. There was some discussion over updates to the data, such as from issues raised by the caretakers and how this would be represented - but it was assured that through the use of filters it would work.

MW had made use of the term “stalled” in his UI which was ambiguous. It was suggested changing this to “postponed” or to an alternate descriptor.

There was further discussion as to the layout, including the number of buttons and their effects. Each window would have a “search” feature that would allow the manager to run a query and find the data on the caretaker and/or the tasks. There should also be checkboxes or something similar used at the “sign off” stage of the system, where a manager can “select all” and sign them all off with one click.

There was some discussion over the frequency of tasks, and how to implement it. SC wanted to postpone discussion of this matter as it was his problem to look at and wasn’t affecting the system at this stage, but despite being the chair the group continued with the discussion. It was decided that the task frequency should be stored as a string to allow for maximum description, rather than a set value (0 being a “one off” task, 1 being daily, 2 being twice weekly) as there were just too many parameters to consider. The java code could be tailored to search and report on any and all variables rather than trying to include problematic frequencies (such as “twice a day, three times a week”). Added to this a task that needed to be done multiple times a day would be split into multiple tasks for efficiency, ease of coding and practical use.

**Discussion of Filtering**

The group was in agreement to limit the number of pop-up screens to as few as possible, and keep as much relevant data on one screen without overcrowding it. The practicality of handing the assigned tasks out was discussed again, with a single “print all” button being the favoured approach. This then raised the question of what if tasks were re-allocated at the start of the day, and either one, some or all of the caretakers had their day’s timetables changed.

The decision was made to produce a search screen where the manager could search for a single caretaker and print out just their allocated tasks as necessary.

**Database**

There was a great deal of discussion over the layout of the tables, and what to include. SC introduced his database shell for discussion, and a number of additions and modifications were discussed and implemented.

The important feature was the addition of a table that allowed the storing of data regarding allocated tasks, rather than trying to store all the data in the “task” table itself. This would allow for more refinement of the data, including performance management and comparing tasks alongside the caretakers who had completed or been allocated them.

There was a brief discussion on adding a “comments” dialogue box to the system to allow caretakers to send messages directly to the manager, but with there being an “other” dialogue box where caretakers could inform the manager of any issues it was deemed that this was repeating itself.

The preferences table was looked at in detail and simplified, It was again agreed that there should be a system based calculation to determine a level of efficiency for each caretaker and how this would be calculated. It was agreed that there would be a level of “likes” and “dislikes”, with 0 being “cannot do”, 1 being “really do not like” and 10 being “love”. This feature would help allocate tasks to the correct caretaker.

It was decided that each task would have a “base time required” allocated to it which would represent how long it should take to complete a task. This would then be modified by an efficiency value for each caretaker which would be calculated by the system as an average of the time taken on previous allocations of this task.

This decision resulted in a number of short-duration conversations about of this would be implemented. It was ultimately decided that each task would have a set duration, and that the “completed tasks” table would have a field where the actual time taken would be recorded. Even though the manager wants the system to keep data for two months, it was determined that storing all of the data for performance management reasons would be beneficial, and just have a report showing the previous two months of allocated tasks prevent information overload or displaying too much.

SC asked about such things as environmental factors. The example he gave was cutting and clearing the lawns of grass and other things. In summer this task would be quite quick to complete, but in other seasons this would become problematic at best and would take a lot longer. In winter the cutting of the grass might not even happen, but clearing up might still take quite some time. It was asked how this would be represented in the system. And while environmental factors would affect the efficiency, it was decided that the calculation would average itself out over time especially if data was stored from previous years and not just the last two months.

This would also allow data to be stored for tasks that are to be assigned once every 6 months, or greater than the 2 month limit to reported data.

MW suggested that the team focus on creating a system that would work then consider how to access the optional features, and this was agreed in principle.

The team then discussed what was needed for the follow weeks, and tasks were distributed, since MB was not present and could not demonstrate the work that he had completed.

**Any Other Business**

* None raised

**Action Items**

**Next Meeting Agenda Items**

* SC to chair meeting 21/02/2018
* Review Use Case Descriptions
* Review start of UI construction
* Pseudocode algorithms
  + Task allocation
  + Password hash
  + Efficiency calculation
  + Report creation