**MoSCoW Task Analysis**

**Must**

*(Mission 5)*

\* Allow users to add regular and one-off tasks

\* Allow users to add relevant information regarding tasks, including type, duration, importance and frequency

\* Allow user to edit information regarding tasks.

\* Safeguard against common errors, such as accidental duplicate events and tasks

\* Present sortable list of existing tasks

*(Mission 6)*

\* Method of persistent data storage

\* Methods allowing for querying and updating of data

*(Mission 7)*

\* Allow tasks to be assigned to caretakers allowing for expected competition time, preferences, tales and special conditions.

\* Provide clear statement on daily tasks to each caretaker

\* Sortable list of tasks which clearly indicates unassigned tasks, as well as other key information i.e task priority

\* Safeguards to prevent errors, such as assigning the same task to several members of staff

*(Mission 8)*

\* Provide interface for users to log in using their username and password

\* Ability to add new users to the system

\* Ability to remove existing users from the system

\* Ability to edit an existing users information

\* Ability to categorise users, restricting access to features based upon their provided role(s)

\* Provide a software interface that allows other components of the program to check the current user’s credentials

\* Allow users to update security features such as passwords

**Should**

*General*

\* User friendly interface

\* Reliability - should not encounter any crashes

\* Maintainability - Well-structured, commented and documented code

\* Aesthetic - System should have a neat, modern and consistent layout

*(Mission 6)*

\* Efficiency - Duplicate or unnecessary data should be avoided through the use of normalisation (to third normal form) and formatting of queries

\* Interoperability - Data accessibility features should be easily implementable across all components of the solution

**Could**

*(Mission 5)*

\* Task filtering based on criteria

*(Mission 6)*

\* Robust data security

\* Allows for concurrency

\* Use a flexible software package

\* Make use of views, triggers and/or complex constraints

*(Mission 7)*

\* Filtering/sorting based upon completion time/preferences etc.

*(Mission 8)*

\* Undo Changes Features

\* User Filtering

\* Ensure that passwords meet a criteria to ensure greater standard of security is met

\* Use encryption techniques to ensure passwords can not be accessed from the database, thus jeopardising the integrity of the system

**Won’t**