# **Statement of Purpose**

The proposed system under development is a Task Allocation System for the client, Capytec. The system intends to replace the paper-based system currently in use by the client for the same purpose, with the aim being to improve the efficiency of the caretaker’s roles by implementing an automatic task allocation system.

As per the client’s requests, the key features of the system should allow:

· Ability to maintain a list of tasks

· Allow caretakers to be allocated tasks, with support in the allocation process

· Tracking the completion of allocated tasks and the caretakers who completed them

· Provision of various performance reports regarding task completion and allocation

Similar to the current system, routine and, less commonly, one-off tasks should be allocated at the start of the day to the present caretakers. Minimising the time spent in the task allocation process is the main concern of management, so the process should be able to allocate tasks automatically, fairly spread across the present caretakers considering caretaker preferences, completion time, complexity and importance of tasks as to provide even allocation without any arguments/disagreements.

By taking several factors into consideration when performing task allocation, the caretaker’s requirements will be met as it is made apparent from the caretakers that many factors require consideration, hence why the current method has grown to take considerable time.

Tracking task completion is an important feature in this implementation, caretakers should sign off completed tasks and the system should retain this information for at least one month so that management may look back at previous tasks. It is noted that, for some tasks, proof of completion is of high importance for health and safety and as per property services’ request it should be possible to sign off important tasks as soon as they are finished. This is, however, not necessary for tasks of lesser importance.

Since there is only one shared PC, caretakers must be able to quickly change between logins, so they can sign off on their individual tasks without accidentally signing off as another user.

Management should have the ability to maintain user accounts so that new caretakers may be added in the case of expansion, or removing inactive users when they are no longer present in the business hence allowing longevity of the system.

**MoSCoW Task Analysis**

**Must**

*(Mission 5)*

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

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

\* Allow manager 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**