**Table of Contents**

[INTRODUCTION 1](#_Toc72724268)

[TEST PLAN 1](#_Toc72724269)

[Incorrect user selection: (input not in the menu) 1](#_Toc72724270)

[Main menu functions: 2](#_Toc72724271)

[Staff menu functions 5](#_Toc72724272)

[Member menu functions 11](#_Toc72724273)

[Tool categories menu. 13](#_Toc72724274)

[Algorithm 17](#_Toc72724275)

[Algorithm design 17](#_Toc72724276)

[Algorithm analysis 19](#_Toc72724277)

CAB301 PROJECT REPORT

# **INTRODUCTION**

This project works on a library system of tools that has 9 main categories and each has unique tool types for member to borrow. All the transactions/actions/updates from “staff” and “member” is systematically record by the application. By using a given criteria and detail design scenario, I have implemented all needed interfaces and made use most of the given functions. Furthermore, a test plan of the application and a detail analysis of algorithm have been finished which are demonstrated below.

# **TEST PLAN**

## **Incorrect user selection: (input not in the menu)**

\***note**: this check user’s inputs method applied to all menu in the program that requires user input to demonstrate next interface. (ex: Tool’s category menu, staff menu, member menu, tool’s type menu, tool option menu, member list menu, member’s borrowing tool list menu, etc.)

Text

Description automatically generated**Outcome**: (take main menu as an example)

## **Main menu functions:**

1. Login as staff:
2. **Text

   Description automatically generatedCorrect staff login:** using account name ”staff”, password: “today123”

**Outcome**: staff menu

Text

Description automatically generated

1. **Wrong staff** **login**: use wrong account name or password.

**Text

Description automatically generatedText

Description automatically generatedOutcome**:

1. Member Login
2. Success member login: input correct username and password (and the account must be existed)

Text

Description automatically generated- **Outcome**: displays member menu

1. Fail to login as a member:

**Outcomes**:

Text

Description automatically generated- Case: input wrong password(existed account)

Text

Description automatically generated- Case: Input account that is not existed in the system

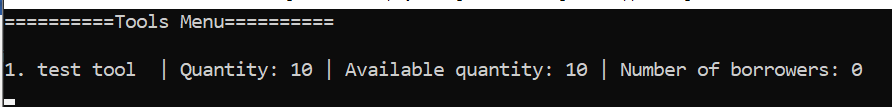
1. Exit program.
2. press “0” to exit the program instantly.

**A screen shot of a computer

Description automatically generated with low confidenceOutcome**:

## **Text Description automatically generatedStaff menu functions**

1. Add a new tool:
2. Successfully added: (valid quantity input)

**Outcome**: 

1. Fail to add (invalid quantity input):

Text

Description automatically generated**Outcome**:

1. Successfully added 2 tools with the same name:

Text

Description automatically generated**Outcome**:

1. Add new pieces of an existing tool:
2. Successfully added more pieces. (the tool must be existed to add)

Text

Description automatically generated**Outcome**:

Text

Description automatically generated with medium confidence

1. Fail to add more pieces.

**Outcomes**:

Case: No tool available to add more pieces:

Text

Description automatically generated

Text

Description automatically generatedCase: Wrong tool option

Text

Description automatically generatedCase: Invalid quantity input

1. Remove some pieces of a tool:
2. Successfully remove pieces:

Text

Description automatically generated**Text

Description automatically generated with medium confidenceOutcome**:

1. Fail to remove pieces:

**Outcomes**:

Text

Description automatically generatedCase: No tools available

Text

Description automatically generatedCase: wrong tool option

Text

Description automatically generatedCase: wrong quantity input

1. Register a new member:
2. Successfully added a new member. (provide valid data)

Text

Description automatically generated**Outcome**:

1. Fail to add new member.

**Outcomes**:

Text

Description automatically generatedCase: Account already existed.

Text

Description automatically generatedCase: Wrong phone number input.

Text

Description automatically generatedCase: Account Name first name or last name is empty (will repeatedly ask until got the input)

1. Remove a member:
2. Successfully remove a member (The user must be added before to remove)

Graphical user interface, text

Description automatically generated**Outcome**:

1. Fail to remove a member:

**Outcomes**:

Text

Description automatically generated with medium confidenceCase: account is not existed.

Case: the user is currently holding tool(s) Text

Description automatically generated

1. Find the contact number of a member:
2. Successfully found a member’s phone number: (user is existed and phone number is provided-unless return none)

Text

Description automatically generated**Outcome**:

1. Fail to find user’s phone number.

**Outcome**:

Text

Description automatically generatedCase: Account name is not existed in the library system

1. Back to main menu:

Press “0” to turn back to main menu. (while in staff menu)

Text

Description automatically generated**Outcome**:

## **Text Description automatically generatedMember menu functions**

1. Display tools of a tool type
2. Choose any tool type (in this screenshot I choose tool type: “Line trimmers”)

**Outcome**:

Text

Description automatically generatedCase: there are tools inside the tool type. (the tools in the screenshot are added into this tool type before)

A screenshot of a computer

Description automatically generated with low confidenceCase: there is no tools inside the tool type.

1. Borrow a tool.
2. Successfully borrowed a tool. (The tools must be first added to system to borrow)

Text

Description automatically generated**Outcome**:

1. Fail to borrow a tool.

**Outcomes**:

Case: there is no tools to borrow.

A screenshot of a computer

Description automatically generated with low confidence

Text

Description automatically generatedCase: The user already got 3 tools inside the inventory.

A picture containing text, indoor, black, dark

Description automatically generatedCase: The tool’s available quantity is 0.

1. Return a tool.
2. Successfully return a tool (The user must have tools in inventory to return)

**Outcome**:

Text

Description automatically generated

1. Fail to return a tool.

**Outcome**:

Case: user have not borrowed any tools.

Text

Description automatically generated

1. List all the tools being rented.
2. Success to show.

**Outcome**:

Text

Description automatically generated

1. Fail to show. (when there is no tools)

**Outcome**:

Text

Description automatically generated

1. Display top 3 most frequently rented tools.
2. Success to show (When there are at least 3 tools in the system)

**Outcome**:

1. Text

   Description automatically generatedFail to show.

**Outcomes**:

Text

Description automatically generatedCase: not enough tools in the library system. (Still show ranking but not fully top 3 tools – for example the screenshot below demonstrates the case where there’s only one tool in the system.).

1. Back to main menu (while in member menu)

**Outcome**:

Text

Description automatically generated

## **Text Description automatically generatedTool categories menu.**

1. Choose gardening tools.

**Text

Description automatically generatedOutcome:**

1. Choose flooring tools.

**Outcome:**

**Text

Description automatically generated**

1. Choose fencing tools.

**Text

Description automatically generatedOutcome:**

1. Choose Measuring tools.

Text

Description automatically generated**Outcome:**

1. Choose cleaning tools.

**Outcome:**

**Text

Description automatically generated**

1. Choose painting tools.

**Outcome:**

**Text

Description automatically generated**

1. Choose electronic tools.

**Outcome:**

Text

Description automatically generated

1. Choose electricity tools.

**Outcome:**

Text

Description automatically generated

1. Choose automotive tools.

**Outcome:**

Text

Description automatically generated

1. Turn back to menu.

**Outcomes:**

Text

Description automatically generatedCase: if the user is the staff

Text

Description automatically generatedCase: if the user is the member

**Algorithm**

## **Algorithm designText, letter Description automatically generated**

Text, letter

Description automatically generated

**ALGORITHM** MaxKeyDelete(A[0…n-1], size)

//This method delete the maximum key and rebuild the whole heap

//Input: An array of orderable items A[0..n-1] and size of that array (a integer value)

//1. Exchange the root’s key with the last key K of the heap;

//2. Decrease the heap’s size by 1;

//3. Heapify the complete binary tree

**While not** heap **and do**

//the left child of k

**If**  **do** //k has 2 children

**If**  **do**

//j is the larger child of k

**If**  **do**

**else do**

**ALGORITHM** displayTopTHree()

//This method rearrange a toolcollection object inside the tool library system then choose out 3 most frequently rented tool. Or in another hand, this method rearranges a list of orderable items then return 3 items with highest values.

//**step 1.** Create an empty array to store all existed values inside the system.

//empty array

//**step 2.** Scan through the system to spot out existed tool and add to the created array. By using this block of code the array will have value v[1…n]

**For each** ToolCategory [1…n] inCategories [1…9] **do**

**For each** ToolType [1…n] **in** ToolCategory [1…n] **do**

**For each** Tool **in** ToolType [1…n] **do**

//add found tools into the empty array

//**step 3**. Sort the array in ascending order using HeapSort function

//**step 4**. Return 3 items with highest values. The array will now be v[1…n] with ascending values.

// “n” is the number of items inside the array

## **Algorithm analysis**

STEP 1: (C1)

Take: 1

STEP 2: (C2)

* Text

  Description automatically generatedAccording to the pseudocode provided above. I can identify the **basic operation** of the function is.
* This basic operation is nested in 3 for-each loops (loop of each category in categories, tool type in each category and tool in each tool type).

- From the most inner-loop the worst case is when there’s n tools available in each tool type (No tool type is empty). Therefore, the basic operation will happen “n” times. => **n times worst case**

- Middle-loop (“ **For each** ToolType [1…n] **in** ToolCategory [1…n] **do**”), based on the category and tool type structure, each category will have between minimum 5 and maximum 6 tools. Therefore, the worst-case for this loop will be a category which has 6 tool types. => **6 times worst case**

- Outer-most-loop has the fix value of 9 categories. => **9 times run**

**Total: n x 6 x 9 = 54n** => step 2 will take 54n

STEP 3: (C3)

using Heapsort() method will have **O(n Log n)**

STEP4: (C4)

Take: 3

The worst-case time complexity in big-O notation of DisplayTopTHree() method:

**T(n) = C1 + C2 + C3 + C4 = 1 + 54n + O(n log n) + 3 = 4 + 54n + O(n log n)**

* Worst case **O(n log n)**