TABLE OF CONTENTS

MAIN REPORT

INTRODUCTION

The system that was developed is an Equipment Hiring Information System (EHIS) for Greenhill Hire firm, a business that stocks general equipment for hiring by their customers. The system possesses a database of the firm's inventory, and each equipment in the inventory has their own attributes including category, item name, brand, quantity, and price per unit. Hired equipment have 2 additional attributes which are the start date and return date of hiring. With this system, customers will be able to make hiring more efficiently because various useful functions are provided for the users' convenience. Additionally, the system's menu and user interface are also neatly organized and simple to use.

FUNCTIONALITY OF THE SYSTEM

Before the user can access the main functions, the system welcomes and prompts the user to enter their name, e-mail, and phone number. Then, the system displays the main menu. In the main menu, the main functions are displayed and made available for the user to choose, including create new hiring, list equipment to be returned on a specified date, count the number of hiring for each category, and list all hired equipment by category. The usage of each function is explained below:

- **Create new hiring**

This function allows users to create a hiring for the equipment available in the firm's inventory. Upon selecting this function, an equipment category submenu is displayed, and the system prompts the user to enter the category that they want to browse in. Based on the chosen category, a list of tools is displayed in another submenu, and the user is prompted to select the tool that they want to hire. Next, a list of brands available for the selected tool is displayed together with their price and remaining quantity, and the user is prompted to select a brand. After identifying the specific tool that the user wants to hire, the system asks the user to enter the quantity of the tool to hire, the hiring's start date and return date.

Then, the system displays the summarized information of the new hiring for the user to check and asks the user for confirmation on hiring the tool. If the user enters 'Y', the system records the details of the hiring, and displays that the process was successful. Conversely, if the user enters 'N', the system cancels the hiring process and redirects the user to the main menu. Throughout the hiring process, all inputs entered by the user must be valid with the data expected by the system, otherwise the system will repeatedly prompt the user for new inputs until a valid input is received.

Additionally, this function is only usable if the hiring limit has not been reached. The system will notify the user that they reached the hiring limit if the limit is reached after creating a new hiring.

- **List equipment to return on a specified date (Search equipment)**

This function displays all equipment hiring based on a specified return date. In this function, the system prompts the user to enter a return date. Then, the system checks all existing hiring records for a matching return date. If a match is found, the system displays the details of all the equipment that are to be returned on the specified date. If no match is found, the system notifies the user that no hiring with the specified return date were found. Regardless of the result, the system asks the user if they want to continue searching for equipment. If the user accepts, they will be prompted for another return date; if the user rejects, they will be redirected to the main menu.

- **Count number of hiring for each category**

This function counts and displays the number of hiring made by the user for each equipment category. In this function, the system first counts and records the number of hiring made for each category. Then, the number of hiring for every category are displayed to the user. After that, the system redirects the user to the main menu.

- **List all equipment by category**

This function lists all equipment that have been hired based on their category. Upon selecting this function, the system displays 5 tables representing each equipment categories, and each table displays the details of hired tools that were categorized under it. If no hired tools can be found in a table, the system displays that there are no hiring records for that table. If the user has not hired any tools, the system shows that the user has not made any hiring record. Finally, the system redirects the user to the main menu.

SCREENSHOTS OF THE SYSTEM

- **Customer registration**

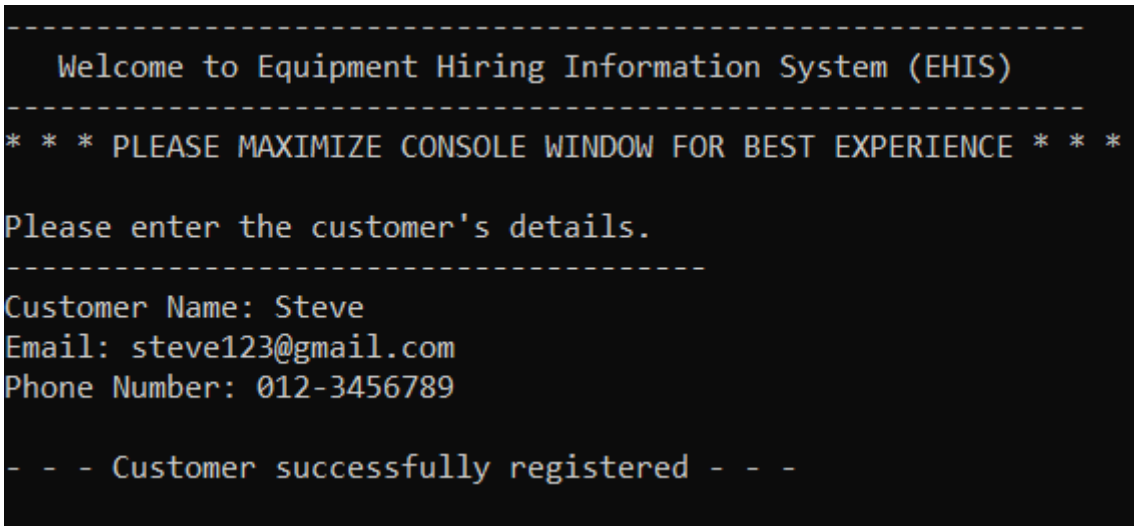

Figure 1.0: System welcomes user and prompts user to enter their details

- **Main menu**

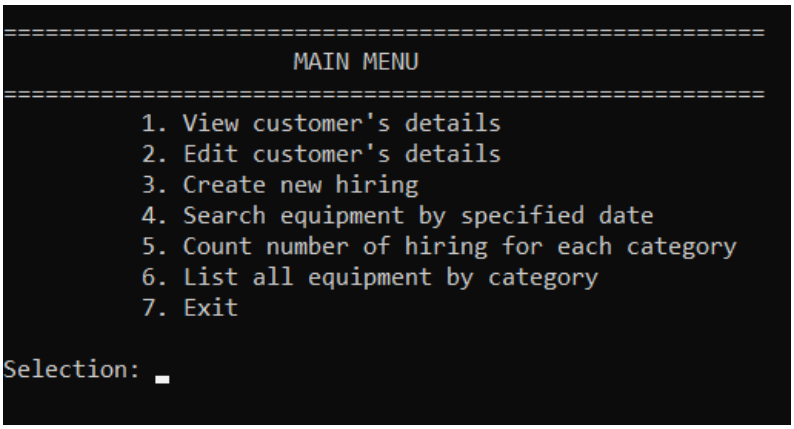

Figure 2.0: System displays the main menu, which shows the main functions. The user is then prompted to select a function.

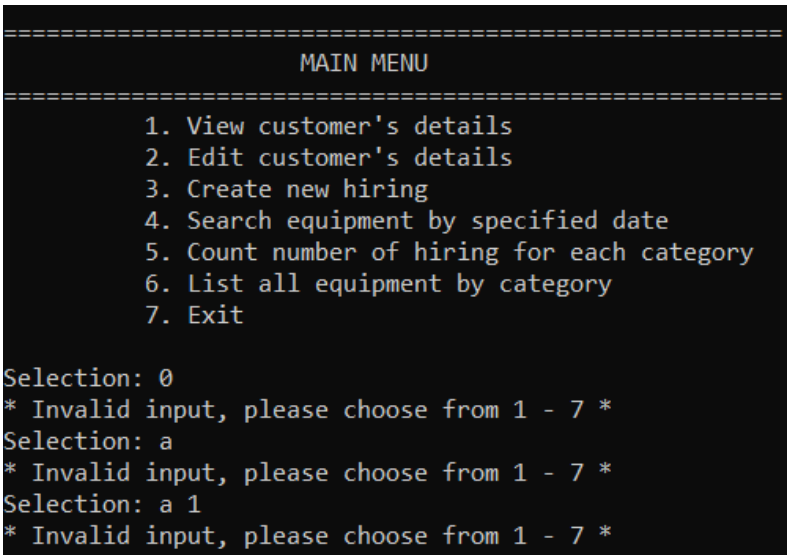

Figure 2.1: System rejects all invalid selection inputs.

- **Create new hiring**

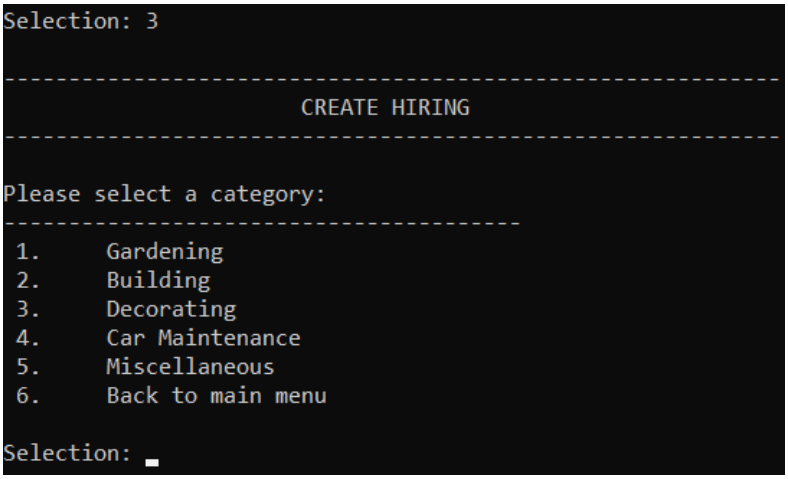

Figure 3.0: User enters “3” in main menu and is redirected to create hiring function, where they are first prompted to select an equipment category.

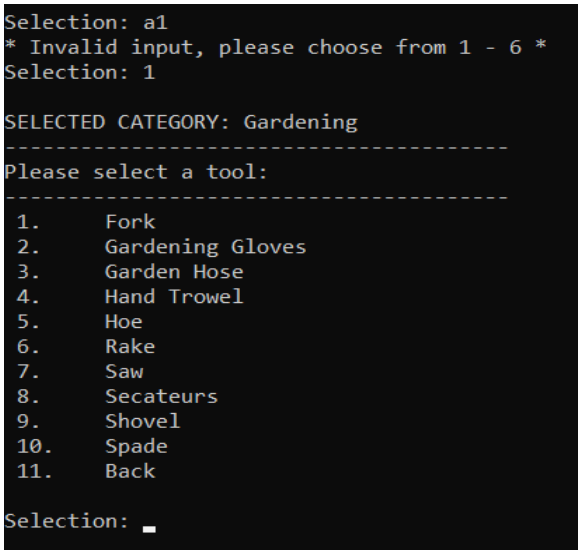

Figure 3.1: User enters "1" to select "Gardening" category. System displays a list of gardening equipment and prompts user to select a tool.

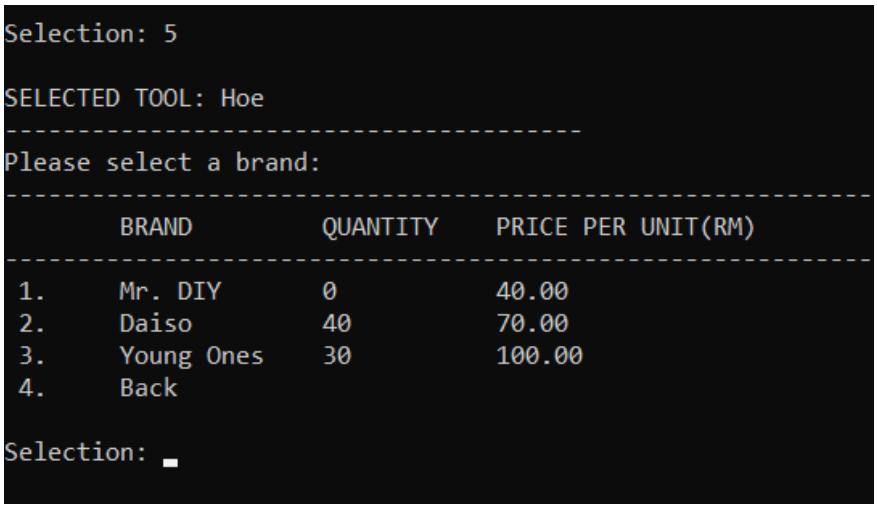

Figure 3.2: User enters "5" to select "Hoe" equipment. System displays a list of brands for the user to select.

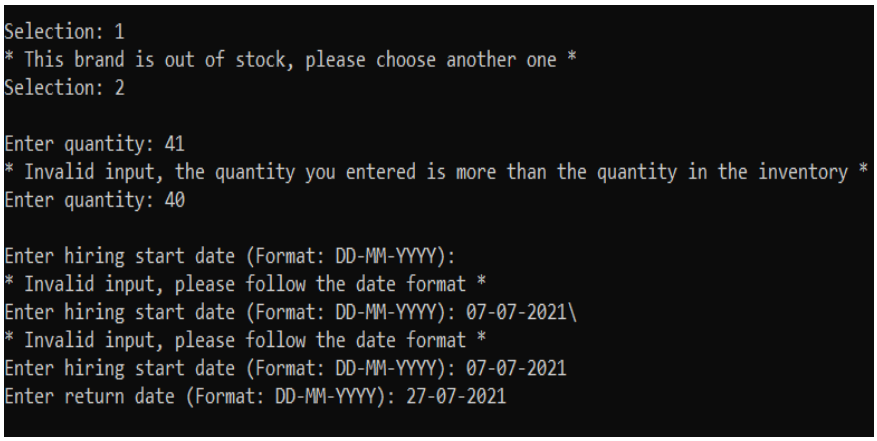

Figure 3.3: After selecting brand, user is prompted to enter quantity, start date and return date of hiring.

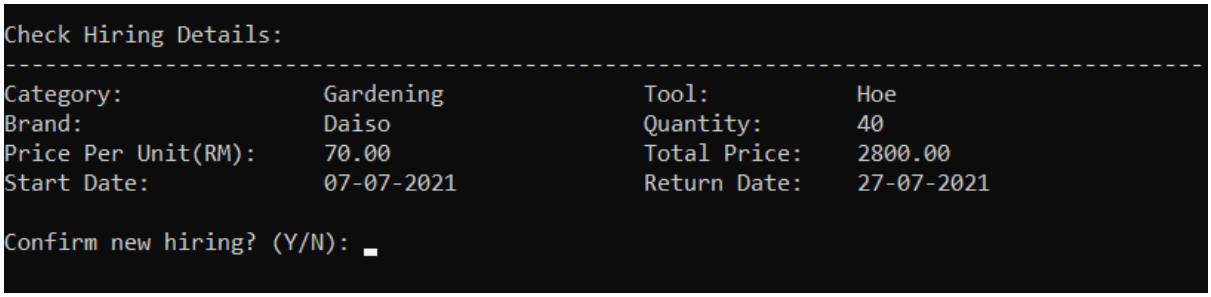

Figure 3.4: System displays a summary of the hiring details and asks user to confirm new hiring.

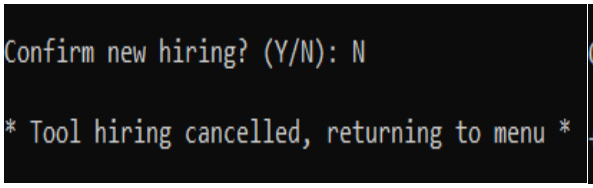

Figure 3.5: User enters 'N' to cancel hiring

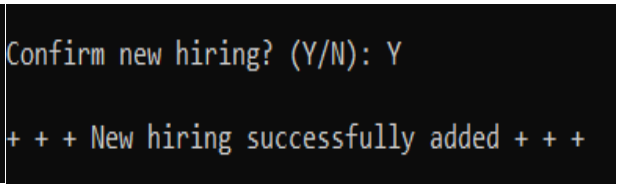

Figure 3.6: User enters 'Y' to confirm new hiring

- **List equipment to return on specified date (Search equipment)**

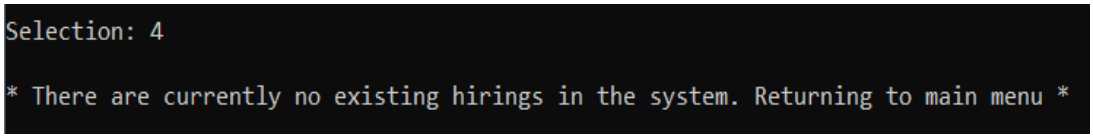

Figure 4.0: User enters "4" to select "Search equipment by specified date", system does not allow this function to be used as there are no existing hiring.

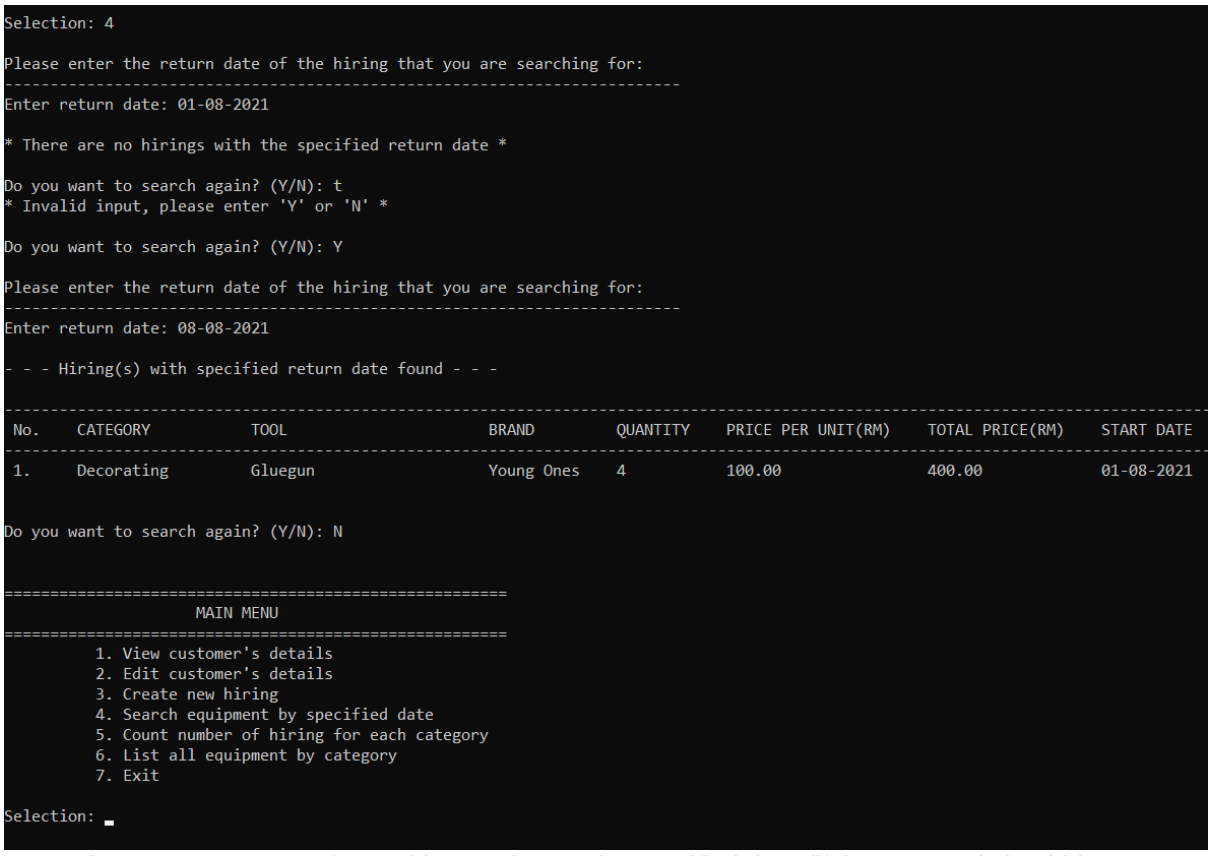

Figure 4.1: Process of searching equipment by specified date (if there are existing hiring to search).hiring to search).

- **Count number of hiring for each category**

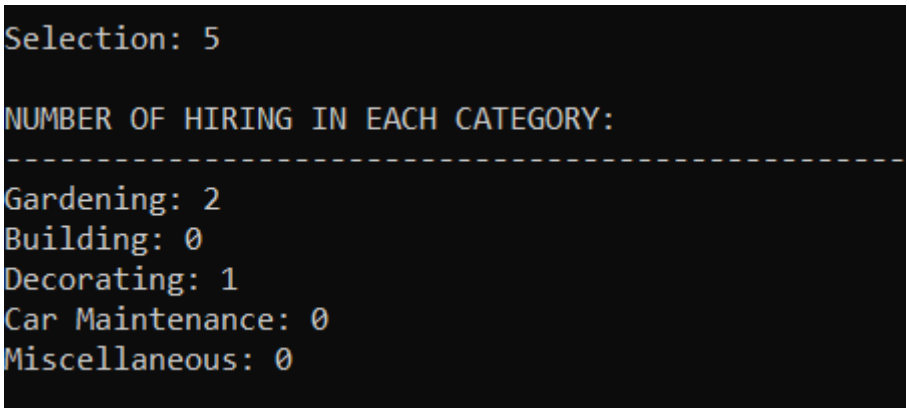

Figure 5.0: User enters “5” to select “Count number of hiring for each category”. The system counts and displays the number of hiring in each equipment category.hiring in each equipment category.

- **List all equipment by category**

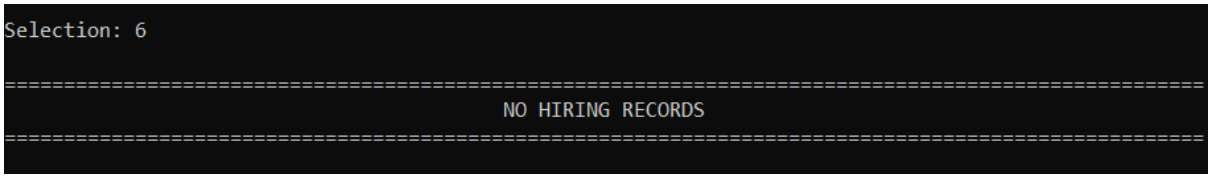

Figure 6.0: User enters “6” to select “List all equipment by category”. System displays “NO HIRING RECORDS” as the user has not created any hiring.hiring.

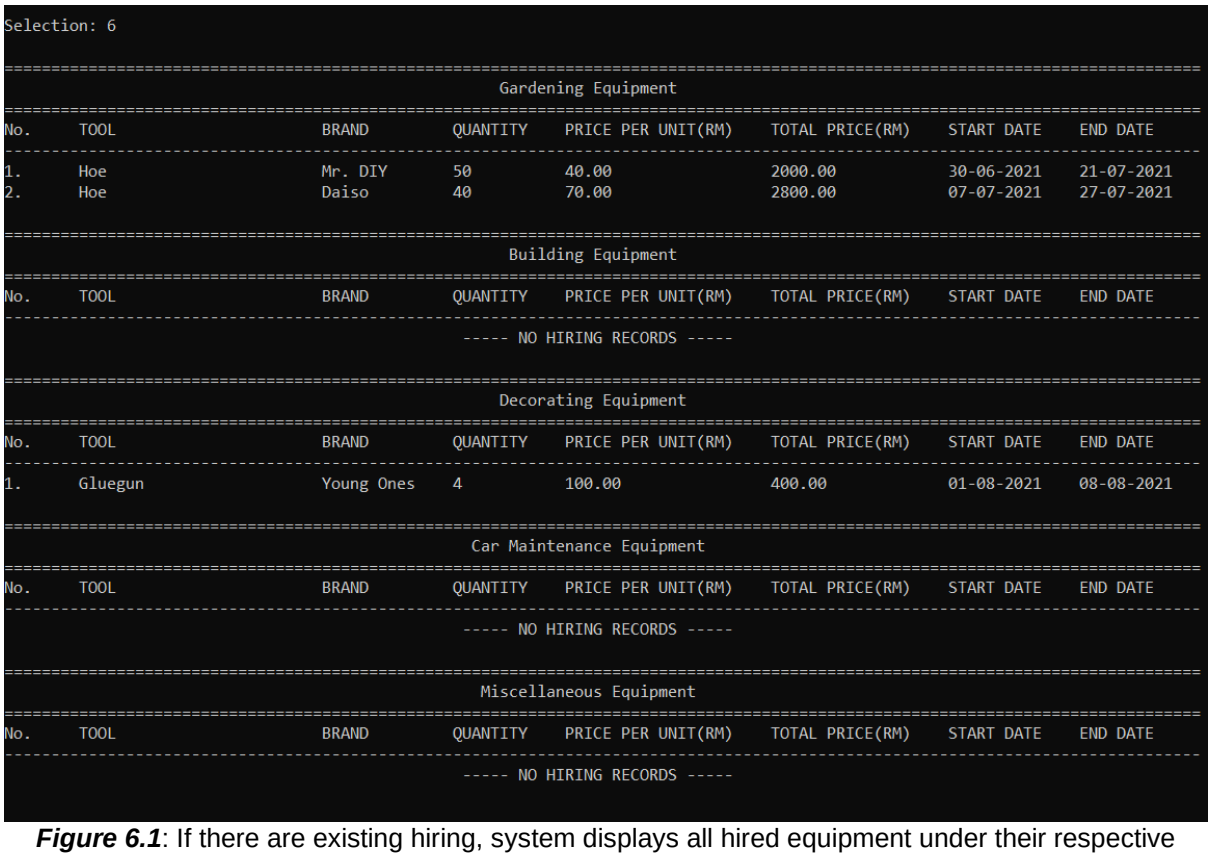

Figure 6.1: If there are existing hiring, system displays all hired equipment under their respective tables according to their category.ystem displays all hired equipment under their respective tables according to their category.

- **Exit**

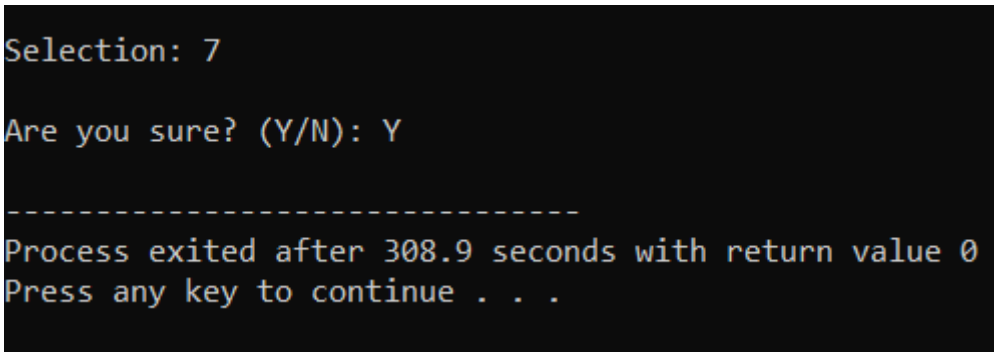

Figure 7.0: User selects “Exit” function and enters ‘Y’ to confirm exit from system.

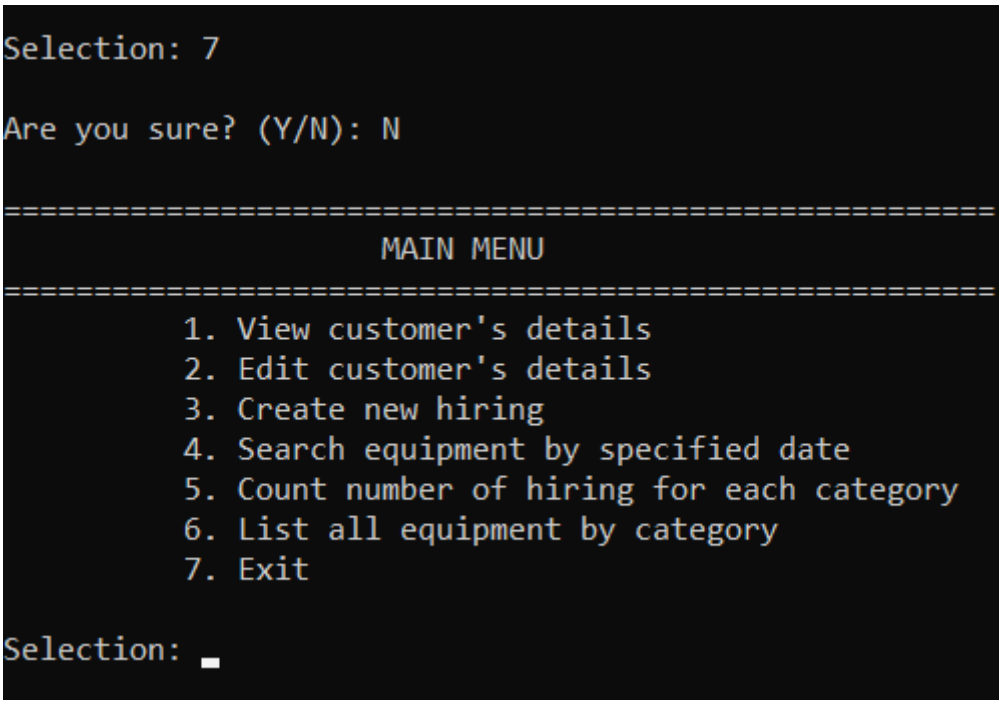

Figure 7.1: User selects “Exit” function but enters ‘N’ to cancel exiting.

STRENGTHS AND WEAKNESSES OF THE PROGRAM

- **Strengths**

One of strengths of the program is efficiency, as the user can easily identify and select the tools that are available for hiring. The program also mimics graphical user interface elements, which has higher usability when compared to text-only applications. The system also provides features that guide users in entering input, help prevent certain human errors, and ensure that most errors are reversible.

- **Weaknesses**

One flaw of the program is that the user interface may get messy after a certain amount of usage as the screen does not clear previous outputs. Another flaw is that the system's equipment database currently can only be modified by directly changing the program's code. The system must be upgraded in the future to integrate with another system that can manage and modify the equipment inventory.

INDIVIDUAL SELF-REFLECTION

- **Lim Zhe Yuan**

In the beginning of the assignment, I discussed about the concepts of our program using language that was too complicated. This leads us to little understanding of our program concepts and the way we write our code. To overcome this, I had to spend more time clearing unnecessary doubts and examining alternate solutions given by my teammates to determine the logic behind each operation and see whether they are relevant to the assignment questions. Thinking back, I should have used more time to determine a concrete program concept first before giving in-depth walkthrough on how we should write the program. Doing this ensures that we will not have ideas that conflict with the main concept that we have decided on when we write our program.

- **Thor Wen Zheng**

The final system utilizes structures to record information of different data types about every equipment and stores them in one-dimensional arrays of equipment struct elements. This method makes it easier to manage data compared to previous methods that we have tried. For example, using parallel two-dimensional arrays, each storing information of different data types for every equipment, or using classes and objects. Throughout the development of the system, I have faced several challenges including heavy college workload and busy schedules due to the shorter semester duration. After facing these challenges, I have improved my time management skills by properly utilizing my free time and learning how to appropriately prioritize my tasks. I also improved my team communication skills as system development in a team requires good teamwork and communication.

- **Tan Peng Heng**

The system was fully functional with all those requirements and some additional functional functions, such as login system, record, search, display and more. The system was using structure array to save all the data with different variables. When I was developing the system, I have more understanding about the logic of coding and learned a better coding way and to fix my bad coding logic. Next, I have also improved my time management skills to make my time arranged wisely and having more free time to do other things since I am having less time to work on my own tasks. Last, by processing this assignment I learned to improve my coding quality and having good user interface design to improve user's experience of the system and also shorting some coding process by shorting the code and make functional functions and I am glad we worked it as a group so that we could help each other when we were having troubles.

- **Adam John Simpson**

During the entire project, I found time management was important and over toward the end did my ever best to improve it. Alongside time management, I had improved my coding skills as I had previously learned how to program in C++ and utilized this skill. Furthermore, communicating with the team helps a lot as it allows for an understanding of the status of each member so the program can be worked on in the most efficient way possible to get it done on time. Being able to compare my work shows that there are different ways and that improvements can be made on my own coding and being able to compare my own work and get help from the other members can help understand how to shorten the code. I had found that my code could be shortened significantly, and this allowed for me to make improvements. I absolutely appreciate the work my group members have done to help with the project and that we were able to help each other during challenging times. Using these experiences can not only improve my coding for future opportunities but also my communication skills can be utilized from this experience.

DDA1224 Data Structures and Algorithms
MARKING RUBRIC
ASSIGNMENT 1
(20%)

Student1(S1): 0204677 LIM ZHE YUAN
Student2(S2): 0205096 THOR WEN ZHENG
Student3(S3): 0204144 ADAM JOHN SIMPSON
Student4(S4): 0205430 TAN PENG HENG

Coding(70%)

Learning Outcome	MARKING CRITERIA	SCALE					
		Not Achieved (0-4.5)	Just Achieved (5.0 - 5.5)	Proficient (6.0 - 6.5)	Very Good (7.0 - 7.5)	Excellent (8.0-10.0)	YOUR MARKS/COMMENTS
CLO1: Implement basic data structures	1. Code Quality (15%)	Very poor attempt. Unstructured, no comment/remarks, non meaningful variables used.	A poor attempt; which may be several problems with structure, or very little use has been made of comments, or the naming of classes, methods and variables is unsatisfactory in a significant number of cases.	quality with several omissions of naming and use of comments.	of comments, and where the majority of classes, variables and methods have been appropriately named. However there may be several omissions of Javadoc comments, and the code.	Good use of commenting throughout, including Javadoc comments for the vast majority of classes, methods and variables.	
	2. Modularity (use of classes/functions, with parameters and returned value/ collection) (10%)	Non modular program. No function used.	Demonstrate some limited use of classes/ functions/ collection.	Demonstrates appropriate use of classes/ functions/ collection.	Demonstrates proficiency in use of classes/ functions/ collection.	Demonstrates mastery in the use of classes/ functions/ collection.	
	3.Use of appropriate data structure (15%)	No data structure used	Demonstrates limited use of data structure and algorithm.	Demonstrates reasonable use of data structure and algorithm, but with a few	Demonstrates proficiency in use of data structure and algorithm.	Demonstrates complete and proper use of data structure and algorithm.	

				shortcomings.			
	4. Functionality (Program execution and output quality) (20%)	Non executable program	Program executed with runtime error but achieve partial program requirements.	Program executed error free with limitations to achieve minimum program requirements.	Program executed error free with correct output and achieve all program requirements.	Program executed error free with excellent output with appropriate validation.	
	5. Readability of the program (10%)	Non readable program	readability with poor indentation, variable names, and remarks	readability with poor indentation, but meaningful variable names, and remarks	readability with good indentation, meaningful variable names, but less remarks	Excellent readability with good indentation, meaningful variable names, and remarks	
						Total: 70%	
Report (30 %)							
Learning Outcome	**MARKING CRITERIA**	**SCALE**					
		Not Achieved (0-4.5)	**Just Achieved (5.0 - 5.5)**	**Proficient (6.0 - 6.5)**	**Very Good (7.0 - 7.5)**	**Excellent (8.0-10.0)**	**YOUR MARKS/COMMENTS**
CLO1: Implement basic data structures	**1. Introduction (5%)**	Little or insufficient definition of a data structure	Basic definition of a data structure.One or two topics is/are left out.	Able to define a data structure and its services but some parts are without adequate explanation/support evidence.	Able to define mobile a data structure with adequate explanation /support but there are only minor parts need further clarification.	Author is able to define a data structure with adequate support of journal papers.	
	2. Quality of information (10%)	Information not clearly relates to the main Topic.	Information clearly relates to the main topic. Points are insufficiently developed. Analysis is weak	Information clearly relates to the main topic. Points are made, but analysis is minimal	Information clearly relates to the main topic. Points and analysis are made and related to the topic.	Information clearly relates to the main topic. Points are clearly made. Analysis is sophisticated	
	5.Style of writing (5%)	The report writing does not meet the criteria for the assignment (too short or incomplete, too	Many ideas require clarification and/or are off-topic or have marginal	Ideas are stated clearly and are related to the topic, with only adequate grammatical	Most ideas are stated clearly and are related to the topic, with only minor grammatical and/or spelling errors.	Writing is clear and relevant, with no grammatical and/or spelling errors – polished and professional.	

		long, and/or completely off-topic). Reference section is missing.	relevance to the assignment. Many grammatical and/or spellings errors throughout the paper. Improper reference section	and/or spelling errors. Reference section with minor flaws	Reference section is in minimal	Reference section properly formatted.				
	6. Individual self-reflection (10%)	Student unable to demonstrates accurate, extensive, and deep understanding of the topics, where connection to a wide range of context, and reflection are not evident.	Student strive to demonstrates accurate, extensive, and deep understanding of the topics, where connection to a wide range of context, and reflection are partly evident.	Student strive to demonstrates accurate, extensive, and deep understanding of the topics, where connection to a wide range of context, and reflection are evident.	Student demonstrates accurate, extensive, and deep understanding of the topics, where connection to a wide range of context, and reflection are evident.	Student demonstrates very accurate, extensive, and deep understanding of the topics, where connection to a wide range of context, and reflection are clearly evident.	**S1**	**S2**	**S3**	**S4**
						Total 30% :				