

1

Shared Power

Assignment Report

GROUP MEMBERS & STUDENT ID:

1. Marcus Bennet – 1818787
2. Arya Elpula – 1714603
3. Shehu Shehu – 1817622
4. Marian Terchila - 1819263
5. Shehu Abubaker – 1911192

**INTRODUCTION:**

Our group of 5 members have come together to create a software called “Shared Power” which would help tradesmen to share their tools. Users in the system will be able to hire tools for up to 3 days and can search for the availability of the tool for up to 6 weeks in advance. There would be options whether to be collected or delivered. They will also have the option to have the availability of the tool per day or per half day rates for each unit.

Nonetheless, if a tool/s have been returned late then there would be a fine which is double the day rate for every day it has not been returned. We, as a group have also understood that the system had to generate invoices for all the users on monthly basis. It was pragmatic to include all the rest of the expenses such as hire charges, delivery costs and late fee (if appropriate). It was necessary to include the compulsory £5 insurance bill in all the bills monthly.

The system that we are implementing will allow owners to take pictures/description of the tools when they are uploaded to the system instantly. When a tool has serious damage, the insurance company will determine who was at fault and who would have needed to pay for the repair of the tool/s.

Tasks have been set for each of our group member after finalising assumptions and constraints. We decided to split up the work to cover grounds much faster and helped each one of their designated tasks if assistance was needed.

**UML DIAGRAMS:**

Our group commenced by designing the sea level diagram and at start have identified 3 actors as Owners, Customers and Shared power administrator.

**Sea Level Diagram (See appendix 1**):

* The Owner once registered, logs in to his/her account and will add the tools, remove the tools, edit their profile, certify the condition of the tool once returned from the customer.
* The customer can log in when they are registered, Search for the tool in the menu, update their profile and ultimately hire the tool itself as well as return the tool.
* Shared Powers system will login, generate monthly invoices and most importantly manage user accounts so if a user has forgotten their details, they can be aided by the admin.

After the sea level diagram has been designed, we have started working towards designing comprehendible fish level diagrams which will go into more depth of what each use-case initiate.

**Fish Level Diagram – Creating account (See appendix 2):**

* The owner can add tools for publishing in their own menu as well he/she can also add the items name, Tool description, pick up address (will be shown once hired?) tool price which will give an option whether the tool will be half a day or whole day rate and have the chance to upload one or more pictures of the tool.
* The tool description will be filled with general model name, duration of usage and tool usage instructions which will be guided by set of pictures (if uploaded) also to verify if there are damages to the tool/s.
* The pick-up address will be revealed to the customer once the tool/s has been rented as the owners’ address is considered to be personal data.
* The owner will also add the pricing of the tool, we have generalised the price in between because the owner will have the ability to add half-day prices or full-day prices on a tool and then can publish the tool to the app.

**Creating account (See Appendix 3):**

This diagram involves both the users where they can create their accounts by firstly registering themselves as a new user, then adding all the details needed in the registry form. After submitting the form, they will be asked to sign in and eventually will have to verify their emails in order to secure their accounts.

**Logging in** **(See Appendix 4):**

Both owner and customer can login the system as an existing user or by creating a new account.

**Generate Monthly Invoices** (**See Appendix 5):**

The programmed administrator will generate monthly invoices by calculating the hire charges, the delivery cost (if applicable), add a late fee if the tool has not been returned (double the rate for every day it has not been returned) on time and add a compulsory £5 Insurance fee.

**Verify tool condition diagram** **(See Appendix 6):**

The owner will navigate to his/her home menu and will have the option to report damage of tool to the insurance company and the insurance will determine who is at fault by an investigation ,and once that is done they will communicate through via e-mail to both the owner of the tool and the customer who has hired the tool recently in order to decide who will pay for the damage.

**Edit profile diagram** **(See Appendix 7**):

This diagram involves both owner and the customer where they have the chance to update their profile by changing their first name, last name, address, post code, phone numbers and email address. For the username and password, the user will have to retype their old password to set a new one, if they do not remember their old password, they will be assisted by the shared power admin via e-mail.

**Clam Level- Existing user (See Appendix 8):**

All the actors from our system, the Tradesman, the Hirer and the Administrator adds a Username and a Password for accessing the system.

**Clam Level- New user (See Appendix 9):**

The Tradesman and the Hirer can create a new user, by adding the name, last name, age, address, phone number, email address username and password.

The activity diagram follows the sequence that place in these use cases.

**Activity diagram for Customer (See Appendix 10):**

The Customer starts by Login in the system which is followed by a New User or an Existing User. If the Customer is an existing user, he will write his username and password and login in. If the Customer is a new user, he/she will have to add all the details and then register.

Once entered inside the system the Owner has four options to choose from which are Return Tools, Select Tools, Give Review and Edit Profile. In returning tool the Owner first uploads the picture, then write the notes and can Logout. For selecting the tool Customer first chooses the quantity then chooses the distribution method which is either pickup or delivery. The Owner then choose date which is the purchase date and writes his address. When the order is complete, he/she can confirm order and Logout. After editing the profile, he/she can save it and Logout as well.

**Class diagram for customer (See Appendix 9):**

**ASSUMPTIONS AND CONSTRAINTS:**

The first assumption we as a group have made was that the owner /users were already registered in the software and they didn’t need to create an account before they log in. Unfortunately, this assumption was later proved wrong because after having a meeting with the group because we have realised that the owners/users were not already registered and a basic development for registration system was needed for us to understand the concepts and ideas of how an account is created in the system.

Another assumption the group has made was that insurance charge of £5 would be added to each rental that has been hired. For example, if the customer hired 5 tools in a month then there would be a charge of £25 on the invoice for the month. However, this assumption was cleared by a user; they informed us that only £5 was needed to add to every month regardless of how many tools were hired as this would increase effectiveness of the system and collect data much easier, as well as for our program to act accordingly.

Before the development phase to initiate we have made more several assumptions of how the system would be designed and developed, and how we believe the system would act.

* No middleman is there between owners and users.
* Insurance takes care of non-return/stolen tools
* Invoices only show tools that has been returned.
* Two users are separate, if either each one of them wants to be an owner/customer, they have to make a new account.
* Owner can’t publish the same tool twice
* Insurance claims are handled through emails (outside the system)
* A customer can book tools up to 6 weeks in advance beside seeing its availability.
* If an owner removes a tool, the customer will be notified in the system.

We could have not developed these scenarios as they were too complex and required a lot of different functions. Apart from this we also had to connect and merge our groups python file as each of us had different styles of coding.

**DEVELOPMENT PHASE:**

In the development phase we used global in every function and have created a Graphical User Interface with the Tkinter package. We also imported other packages such as OS in order to interface with underlying operating system where python is running which in this case is Windows/MacOS. Random package has been used for generating random numbers in the invoices. Time date packages has also been functionalised and included in the program for generating current dates in the invoices. File dialog packages to upload files to the database. Also, a message box package to include in appropriate settings such as in the complaint form section to the insurance claim option.

The development phase is divided into four sections starting with Login Interface.  We created two buttons for Tradesman and hirer. The Hirer and Tradesman needs to Create an account before Login inside the system. They can do this by adding personal information including username and password which gets saved on the database. Once the account is created the Hirer and Tradesman can Login inside the system by clicking the Existing User button. If the user inputs the wrong password and username, the python verifies it by comparing it to the file stored in the SQLite database and displays error.

The administrator is computerised and is already logged in the system before hand as the user is embedded within the system itself. For each heading for every window we used light subtle colours and appropriately sized font to create an elusive touch to the user’s environment and not use harsh colours. The screen size is adjusted according to the size of the buttons and labels on the window by the geometry size function.

Secondly, the welcome window for Owners can add tools, edit profile, give complaint to the insurance, verify tool condition and finally logout. The Customer has the same buttons as owners. Besides for the select tools and return tools. The group has used entry boxes for adding tool prices, text widget for tool specifications and tool description. The upload picture function uses file dialog to search for file on the desktop and print the path file which also is included in the customer section when the customer returns the tool. The publish tool button simply deletes the current screen and then saves all the specification to the system.

In the edit profile window, we have used strings and used them as labels. The entry box used for adding information and insert function is utilised for information that will or is currently typed. We have used the same python class as the one in the registry class as it will update from the registry class in the database.

The log out button uses message box which will confirm if you are sure to log out or not and then destroys the current screen when confirming log out. This function is also used in the Customers section too.

On the other hand, the customers section for selecting tools we defined \_ tools which will be navigated in the current window that the user has logged in. For each tool we have defined the same step for other tools. When clicked on any tool on the menu, the picture will be focused which we added by importing the \_ package form Tkinter. The tool description uses labels, select quantity uses spin boxes and \_\_\_\_ buttons for selecting the distribution method. For selecting dates, we used option-menu which selects year, month and day. When the customer completes this process, He/she will complete the order by clicking “\_\_\_\_” which will display a message box and delete all the previous windows.

The computerised administrator generates invoices and the insurance receives a complaint form as the complaint form is submitted to the insurances email and will be diligently investigated from email onwards back and forth to the customer and user to keep updated. The invoice number uses the random.randint function and generates a random number every time the button \_\_\_\_\_\_ is pressed. The date is updated every-time the window is opened.  There are \_\_\_\_\_ tools where the user can add the quantity and when the total cost button is pressed it multiplies the quantity with the tool price and display in entry boxes.

**SOFTWARE TESTING:**

In this project we have done manual testing and initial requirements were as follows:

* Requirement Analysis: We ensured that the application is accessible and user friendly to the people in every operating system.
* Defect Logging: Our group took notes on bugs and faults and then have been debugged.

The scenarios are shown below:

**LOGIN AS TRADEMEN**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario | Test Case | Pre-Condition | Test Step | Test Data | Expected  Result | Actual  Result | Pass  /Fail |
| Check  Login as Tradesman  Functionality | Check Respond on entering valid name and password | 1.Application must be installed  2. Register | 1.Launch Application  2.Enter  Name  3.Enter password  4.click Login | 1.Name  Arya  2.Password  321  1.Name  Marian  2.Password  123 | Login must be successful | Login  Fail | Fail |

* Login fail due to verify functions was not clear in the file
* Login failed due to the delete function not being globalised

**Referred back to developers and testers for debugging again.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario | Test Case | Pre-Condition | Test Step | Test Data | Expected  Result | Actual  Result | Pass  /Fail |
| Check  Login as Tradesman  Functionality | Check Respond on entering valid name and password | Application must be installed.  2.Register | 1.Launch Application  2.Enter  Name  3.Enter password  4.click Login | 1.Name  Arya  2.Password  321  1.Name  Marian  2.Password  123 | Login must be successful | Login  Successful | Pass |

**LOGOUT TEST CASE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario | Test Case | Pre-Condition | Test Step | Test Data | Expected  Result | Actual  Result | Pass  /Fail |
| Logout  Functionality | Checking if logout closes all the window. | Application must be installed.  2.Register | 1.Launch invoice.  2.Data input.  Generate invoice. | 1.Invoice input.  2.Data Input. | Invoice generated | invoice  Successful | Pass |

* **Log out functionality worked well**

**GENERATE MONTHLY INVOICE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario | Test Case | Pre-Condition | Test Step | Test Data | Expected  Result | Actual  Result | Pass  /Fail |
| Check  Invoice  Functionality | Check Invoice Respond entering valid Data. | Application must be installed.  2.Register | 1.Launch invoice.  2.Data input.  Generate invoice. | 1.Invoice input.  2.Data Input. | Invoice generated | invoice  Successful | Pass |

**LOGIN AS CUSTOMER**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario | Test Case | Pre-Condition | Test Step | Test Data | Expected  Result | Actual  Result | Pass  /Fail |
| Check  Login as Hirer  Functionality | Check Respond on entering valid name and password | Application must be installed.  2.Register | 1.Launch Application  2.Enter  Name  3.Enter password  4.click Login | 1.Welcome hire window.  2. Search tool.  3.check image and details | Show images, price, details. | Successful  Showing image, price and details | Pass |

**ISSUES IDENTIFIED AND HOW THEY WERE RESOLVED:**

We solved the login issue by creating separate verify function for each user because python was saving the account information as whole one user. Where python should have given error and it was not giving us error, we fixed that by making sure that the file on which account data is being saved is the same directory as IDLE file.

Furthermore, we have faced several problems in giving buttons some specific command. We couldn’t work out the late fee button which was supposed to double the rate for each tool instead we as a group manually set daily fee of a specified number if all tools are selected. Another issue was in trouble shooting. The main challenge was to combine delivery charges and late fee together. We had to do some additional research on Stack Overflow to create a function to make a

delivery charge button which can deduct and add charges from the invoice and can be calculated and displayed in the display window.

**APPENDIXES:**