Software System Specification for

Workshop System

**Prepared by M.Muteeb Masood**

**Table of Contents**

**Table of Contents ii**

**1.** **Introduction 3**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2.** **Overall Description 3**

2.1 Product Perspective 1

2.2 Problem Domain 1

2.3 Problem 2

2.4 Solution Domain 2

**3.** **System Features 4**

**4.** **System Use Cases 5**

**5.** **System Structural Model 25**

**Appendix A: Glossary 25**

# Introduction

## Purpose

The product we are developing is a workshop system , to be used in a regular workshop. The product has not yet been released . The product has intended usage in a regular workshop and will help them with their tasks.

## Document Conventions

There are no standard or typographical conventions used in this document.

## Intended Audience and Reading Suggestions

The document can be read by everyone managers, developers, testers etc. It contains basic details and also extensive details about the software. The document can help readers know more about the software and how it works and intended use. The document starts with basic details then goes into further details of the software.

## Product Scope

The software basically helps the workshop run smoothly by implementing and managing their daily tasks in a software. The software will be easy to use, as their work will become more efficient and easy. The software would be used to complete and manage daily and regular tasks and it will also keep track of the work done for each client. Completing and maintaining tasks, jobs and clients will be the main goal of the software.

## References

No references were used for this software.

# Overall Description

## Product Perspective

I have come up with a self-contained idea to develop a highly beneficial software for the workshops with mechanics for a much quicker repair of vehicles. This product is completely self-contained and will be implemented for the very first time.

## Problem Domain

Problem domain for the software will be the multiple mechanics and their outlets doing various jobs. A menu, through which the workshop member will locate and figure out the required items and do the required jobs for the client’s vehicle.

## Problem

The system will help workshops run more smoothly as of right now workshops are not automated. It is a total nuisance to travel in rush hours and getting late due to workshops incompetence (human errors, time management). Hence, this is a user-friendly software in order to make these issues less complicated.

## Solution Domain

The workshop system is preferably going to be achieved through python language. An input message will be displayed to the user of the software where after the car and client details will be inserted. A list of related items will be shown to choose from. Variety of payment methods. The jobs and services will be displayed through a database and a friendly interface through the help of python.

# System Features

## Adding a client This feature adds a client and their details to the system. It has a high priority in the system.

* 1. **Printing clients list**This feature prints the clients list in the system, telling us about the clients. It has a medium priority in the system.
  2. **Deleting a client**This feature deletes a client and their details from the system. It has a low priority in the system.
  3. **Adding a mechanic**This feature adds a mechanic and their details to the system. It has a high priority in the system.
  4. **Printing a mechanic list**This feature prints the list of mechanic in the database. It has a medium priority in the system.
  5. **Deleting a mechanic**This feature deletes a mechanic and their details from the system. It has a low priority in the system.
  6. **Repairing a vehicle**This feature will discuss the repairing to be done on the client's vehicle. It has a high priority in the system.
  7. **Making an invoice**This feature deals with the invoice part, basically of the jobs done by the client. It has a high priority in the system.
  8. **Payment method**This feature will deal with how the client would like to pay for the invoice. It has a high priority in the system.

# System Use Cases

## Adding client ACTORS, GOALS AND USE CASE BRIEFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Actors** | **Task-level Goal** | **Priority** | **Brief** |
| Actor is Receptionist in the Workshop and Secondary Actor is the Client. | Goal is to add the new client in the database | High | Client will ask receptionist to add his details for Workshop. Receptionist will use program to Add this client in the database. |

* 1. **USE CASE SPECIFICATIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Identification and History** | | | |
| **Use Case ID:** | PRJWRKSP 1.0 | | |
| **Use Case Name:** | Adding a Client | **Version No:** |  |
| **End Objective:** | This will add the client into the database. | | |
| **Created by:** | Muteeb | **On (date):** |  |
| **Last Update by:** |  | **On (date):** |  |
| **Approved by:** |  | **On (date):** |  |
| **User/Actor:** | Receptionist | | |
| **Business Owner Name:** | Workshop Owner  Kashif | **Contact Details:** |  |
| **Trigger:** | It is triggered by clicking add client in Client section | | |
| **Frequency of Use:** | 3-5 Times a week | | |

|  |
| --- |
| **Preconditions** |
| User clicks Clients in main menu and systems takes user to Clients section |

|  |  |  |
| --- | --- | --- |
| **Basic Flow:**   * **User initializes the program which takes user to a welcome screen and after that main menu will appear from where use selects ‘Clients’ option and then system will take user to Clients section. User selects the option of ‘Add Client’ in Client section. After adding a client program will prompt a message “Client has been successfully added” and take user back to main menu.** | | |
| **Step** | **User Actions** | **System Actions** |
| **1** | User initializes the program | System prompt with Welcome message and then displays the main menu. |
| **2** | User selects the option Client | System takes user to Client section |
| **3** | User clicks Add Client | System takes user to Add Client section |
| **4** | User types the client details | System stores client data in database and prompt the message “Client has been successfully added” |

|  |  |  |
| --- | --- | --- |
| **Alternate Flow** | | |
| **Step** | **User Actions** | **System Actions** |
| **3.1** | User clicks Client List | System prints the List of Clients |
| **3.2** | User clicks Delete Client | System takes user to Delete Client section to delete the desired client. |

|  |  |  |
| --- | --- | --- |
| **Exception Flow** | | |
| **1** | User leaves any slot empty | System prompts the message “Incomplete detail” and take back to client data input section. |

|  |
| --- |
| **Post conditions** |
| System prompts message and take user back to main menu. |

|  |
| --- |
| **Includes or Extension Points** |
|  |

|  |
| --- |
| **Special Requirements** |
|  |

|  |
| --- |
| **Business Rules** |
|  |

|  |
| --- |
| **Other Notes (Assumptions, Issues,)** |
| **Assumptions: N/A**   * **Issues: N/A** |

## Printing client list ACTORS, GOALS AND USE CASE BRIEFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Actors** | **Task-level Goal** | **Priority** | **Brief** |
| Actor is Receptionist in the Workshop and Secondary Actor is the internal database. | Goal is to print the client list from database | Medium | Receptionist will print the client list from the internal database |

## USE CASE SPECIFICATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Identification and History** | | | |
| **Use Case ID:** | PRJWRKSP 1.1 | | |
| **Use Case Name:** | Printing a client list | **Version No:** |  |
| **End Objective:** | This will print the client list form the internal database | | |
| **Created by:** | Muteeb | **On (date):** |  |
| **Last Update by:** |  | **On (date):** |  |
| **Approved by:** |  | **On (date):** |  |
| **User/Actor:** | Receptionist | | |
| **Business Owner Name:** | Workshop Owner  Kashif | **Contact Details:** |  |
| **Trigger:** | It is triggered by clicking Client List in Clients section | | |
| **Frequency of Use:** | Once a Month | | |

|  |
| --- |
| **Preconditions** |
| User clicks Clients in main menu and systems takes user to Clients section |

|  |  |  |
| --- | --- | --- |
| **Basic Flow:**   1. **User initializes the program which takes user to a welcome screen and after that main menu will appear from where use selects ‘Clients’ option and then system will take user to Clients section. User selects the option of ‘Client List’ in Client section. Systems prints the Client List and take user back to main menu.** | | |
| **Step** | **User Actions** | **System Actions** |
| **1** | User initializes the program | System prompt with Welcome message and then displays the main menu. |
| **2** | User selects the option Client | System takes user to Client section |
| **3** | User clicks Client List | System prints the Client List from the internal database |

|  |  |  |
| --- | --- | --- |
| **Alternate Flow** | | |
| **Step** | **User Actions** | **System Actions** |
| **3.1** | User clicks Add Client | System takes user to Add Client section |
| **3.2** | User clicks Delete Client | System takes user to Delete Client section to delete the desired client. |

|  |  |  |
| --- | --- | --- |
| **Exception Flow** | | |
| **1** | User command to print Client List without any Client in database | System prompts the message of ‘Empty Client List’ |

|  |
| --- |
| **Post conditions** |
| System prints the Client List and take user back to main menu. |

|  |
| --- |
| **Includes or Extension Points** |
|  |

|  |
| --- |
| **Special Requirements** |
|  |

|  |
| --- |
| **Business Rules** |
|  |

|  |
| --- |
| **Other Notes (Assumptions, Issues,)** |
| **Assumptions: N/A**   1. **Issues: N/A** |

# 

## Deleting client ACTORS, GOALS AND USE CASE BRIEFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Actors** | **Task-level Goal** | **Priority** | **Brief** |
| Actor is Receptionist in the Workshop and Secondary Actor is the internal database | Goal is deleting the existing client in the database | Low | Receptionist deletes the desired client from the database |

## USE CASE SPECIFICATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Identification and History** | | | |
| **Use Case ID:** | PRJWRKSP 1.2 | | |
| **Use Case Name:** | Deleting a Client | **Version No:** |  |
| **End Objective:** | This will delete the client into the database. | | |
| **Created by:** | Muteeb | **On (date):** |  |
| **Last Update by:** |  | **On (date):** |  |
| **Approved by:** |  | **On (date):** |  |
| **User/Actor:** | Receptionist | | |
| **Business Owner Name:** | Workshop Owner  Kashif | **Contact Details:** |  |
| **Trigger:** | It is triggered by clicking Delete client in Client section | | |
| **Frequency of Use:** | 5-10 times a year | | |

|  |
| --- |
| **Preconditions** |
| User clicks Clients in main menu and systems takes user to Clients section |

|  |  |  |
| --- | --- | --- |
| **Basic Flow:**   1. **User initializes the program which takes user to a welcome screen and after that main menu will appear from where use selects ‘Clients’ option and then system will take user to Clients section. User selects the option of ‘Delete Client’ in Client section. After deleting a client program will prompt a message “Client has been successfully deleted” and take user back to main menu.** | | |
| **Step** | **User Actions** | **System Actions** |
| **1** | User initializes the program | System prompt with Welcome message and then displays the main menu. |
| **2** | User selects the option Client | System takes user to Client section |
| **3** | User clicks Delete Client | System takes user to Delete Client section |
| **4** | User selects the desired client | System deletes client data in database and prompt the message “Client has been successfully deleted” |

|  |  |  |
| --- | --- | --- |
| **Alternate Flow** | | |
| **Step** | **User Actions** | **System Actions** |
| **3.1** | User clicks Client List | System prints the List of Clients |
| **3.2** | User clicks Add Client | System takes user to Add Client section |

|  |  |  |
| --- | --- | --- |
| **Exception Flow** | | |
| **1** | User command to delete empty list | System prompts the message “Empty List” and take back to client data input section. |

|  |
| --- |
| **Post conditions** |
| System prompts message and take user back to main menu. |

|  |
| --- |
| **Includes or Extension Points** |
|  |

|  |
| --- |
| **Special Requirements** |
|  |

|  |
| --- |
| **Business Rules** |
|  |

|  |
| --- |
| **Other Notes (Assumptions, Issues,)** |
| **Assumptions: N/A**   1. **Issues: N/A** |

## Adding mechanic ACTORS, GOALS AND USE CASE BRIEFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Actors** | **Task-level Goal** | **Priority** | **Brief** |
| Actor is Receptionist in the Workshop and Secondary Actor is the Mechanic | Goal is to add the new Mechanic in the database | High | Mechanic will ask receptionist to add his details for Workshop. Receptionist will use program to Add this mechanic in the database. |

## USE CASE SPECIFICATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Identification and History** | | | |
| **Use Case ID:** | PRJWRKSP 2.0 | | |
| **Use Case Name:** | Adding a Mechanic | **Version No:** |  |
| **End Objective:** | This will add the Mechanic into the database. | | |
| **Created by:** | Muteeb | **On (date):** |  |
| **Last Update by:** |  | **On (date):** |  |
| **Approved by:** |  | **On (date):** |  |
| **User/Actor:** | Receptionist | | |
| **Business Owner Name:** | Workshop Owner  Kashif | **Contact Details:** |  |
| **Trigger:** | It is triggered by clicking Add Mechanic in Mechanic section | | |
| **Frequency of Use:** | Once a month | | |

|  |
| --- |
| **Preconditions** |
| User clicks Mechanic in main menu and systems takes user to Mechanic section |

|  |  |  |
| --- | --- | --- |
| **Basic Flow:**   1. **User initializes the program which takes user to a welcome screen and after that main menu will appear from where use selects ‘Mechanics’ option and then system will take user to Mechanics section. User selects the option of ‘Add Mechanic’ in Mechanics section. After adding a Mechanic program will prompt a message “Mechanic has been successfully added” and take user back to main menu.** | | |
| **Step** | **User Actions** | **System Actions** |
| **1** | User initializes the program | System prompt with Welcome message and then displays the main menu. |
| **2** | User selects the option Mechanic | System takes user to Mechanic section |
| **3** | User clicks Add Mechanic | System takes user to Add Mechanic section |
| **4** | User types the Mechanic details | System stores Mechanic data in database and prompt the message “Mechanic has been successfully added” |

|  |  |  |
| --- | --- | --- |
| **Alternate Flow** | | |
| **Step** | **User Actions** | **System Actions** |
| **3.1** | User clicks Mechanic List | System prints the List of Mechanic |
| **3.2** | User clicks Delete Mechanic | System takes user to Delete Mechanic section to delete the desired Mechanic |

|  |  |  |
| --- | --- | --- |
| **Exception Flow** | | |
| **1** | User leaves any slot empty | System prompts the message “Incomplete detail” and take back to Mechanic data input section. |

|  |
| --- |
| **Post conditions** |
| System prompts message and take user back to main menu. |

|  |
| --- |
| **Includes or Extension Points** |
|  |

|  |
| --- |
| **Special Requirements** |
|  |

|  |
| --- |
| **Business Rules** |
|  |

|  |
| --- |
| **Other Notes (Assumptions, Issues,)** |
| **Assumptions: N/A**   1. **Issues: N/A** |

## Printing mechanic list ACTORS, GOALS AND USE CASE BRIEFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Actors** | **Task-level Goal** | **Priority** | **Brief** |
| Actor is Receptionist in the Workshop and Secondary Actor is the internal database. | Goal is to print the Mechanic list from database | Medium | Receptionist will print the Mechanic list from the internal database |

## USE CASE SPECIFICATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Identification and History** | | | |
| **Use Case ID:** | PRJWRKSP 2.1 | | |
| **Use Case Name:** | Printing a Mechanic list | **Version No:** |  |
| **End Objective:** | This will print the Mechanic list form the internal database | | |
| **Created by:** | Muteeb | **On (date):** |  |
| **Last Update by:** |  | **On (date):** |  |
| **Approved by:** |  | **On (date):** |  |
| **User/Actor:** | Receptionist | | |
| **Business Owner Name:** | Workshop Owner  Kashif | **Contact Details:** |  |
| **Trigger:** | It is triggered by clicking Mechanic List in Mechanic section | | |
| **Frequency of Use:** | Once a Month | | |

|  |
| --- |
| **Preconditions** |
| User clicks Mechanic in main menu and systems takes user to Mechanic section |

|  |  |  |
| --- | --- | --- |
| **Basic Flow:**   1. **User initializes the program which takes user to a welcome screen and after that main menu will appear from where use selects ‘Clients’ option. System takes user to Client section where user selects ‘Mechanics List’. System prints the Mechanics list and take user back to main menu.** | | |
| **Step** | **User Actions** | **System Actions** |
| **1** | User initializes the program | System prompt with Welcome message and then displays the main menu. |
| **2** | User selects the option Mechanic | System takes user to Mechanic section |
| **3** | User clicks Mechanic List | System prints the Mechanic List from the internal database |

|  |  |  |
| --- | --- | --- |
| **Alternate Flow** | | |
| **Step** | **User Actions** | **System Actions** |
| **3.1** | User clicks Add Mechanic | System takes user to Add Mechanic section |
| **3.2** | User clicks Delete Mechanic | System takes user to Delete Mechanic section to delete the desired Mechanic |

|  |  |  |
| --- | --- | --- |
| **Exception Flow** | | |
| **1** | User command to print Mechanic List without any Mechanic in database | System prompts the message of ‘Empty Mechanic List’ |

|  |
| --- |
| **Post conditions** |
| System prints the Mechanic List and take user back to main menu. |

|  |
| --- |
| **Includes or Extension Points** |
|  |

|  |
| --- |
| **Special Requirements** |
|  |

|  |
| --- |
| **Business Rules** |
|  |

|  |
| --- |
| **Other Notes (Assumptions, Issues,)** |
| **Assumptions: N/A**   1. **Issues: N/A** |

# 

## Deleting mechanic ACTORS, GOALS AND USE CASE BRIEFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Actors** | **Task-level Goal** | **Priority** | **Brief** |
| Actor is Receptionist in the Workshop and Secondary Actor is the internal database | Goal is delete the existing Mechanic in the database | Low | Receptionist deletes the desired Mechanic from the database |

## USE CASE SPECIFICATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Identification and History** | | | |
| **Use Case ID:** | PRJWRKSP 2.2 | | |
| **Use Case Name:** | Deleting a Mechanic | **Version No:** |  |
| **End Objective:** | This will delete the Mechanic into the database. | | |
| **Created by:** | Muteeb | **On (date):** |  |
| **Last Update by:** |  | **On (date):** |  |
| **Approved by:** |  | **On (date):** |  |
| **User/Actor:** | Receptionist | | |
| **Business Owner Name:** | Workshop Owner  Kashif | **Contact Details:** |  |
| **Trigger:** | It is triggered by clicking Delete Mechanic in Mechanic section | | |
| **Frequency of Use:** | 5-10 times a year | | |

|  |
| --- |
| **Preconditions** |
| User clicks Mechanic in main menu and systems takes user to Mechanic section |

|  |  |  |
| --- | --- | --- |
| **Basic Flow:**   1. **User initializes the program which takes user to a welcome screen and after that main menu will appear from where use selects ‘Mechanics’ option and then system will take user to Mechanics section. User selects the option of ‘Delete Mechanics’ in Mechanics section. After deleting a Mechanic program will prompt a message “Mechanic has been successfully deleted” and take user back to main menu.** | | |
| **Step** | **User Actions** | **System Actions** |
| **1** | User initializes the program | System prompt with Welcome message and then displays the main menu. |
| **2** | User selects the option Mechanics | System takes user to Mechanics section |
| **3** | User clicks Delete Mechanic | System takes user to Delete Mechanic section |
| **4** | User selects the desired Mechanic | System deletes Mechanic data in database and prompt the message “Mechanic has been successfully deleted” |

|  |  |  |
| --- | --- | --- |
| **Alternate Flow** | | |
| **Step** | **User Actions** | **System Actions** |
| **3.1** | User clicks Mechanic List | System prints the List of Mechanics |
| **3.2** | User clicks Add Mechanic | System takes user to Add Mechanic section |

|  |  |  |
| --- | --- | --- |
| **Exception Flow** | | |
| **1** | User command to delete empty list | System prompts the message “Empty List” and take back to Mechanic data input section. |

|  |
| --- |
| **Post conditions** |
| System prompts message and take user back to main menu. |

|  |
| --- |
| **Includes or Extension Points** |
|  |

|  |
| --- |
| **Special Requirements** |
|  |

|  |
| --- |
| **Business Rules** |
|  |

|  |
| --- |
| **Other Notes (Assumptions, Issues,)** |
| **Assumptions: N/A**   1. **Issues: N/A** |

## Repairing vehicle ACTORS, GOALS AND USE CASE BRIEFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Actors** | **Task-level Goal** | **Priority** | **Brief** |
| Actor is Receptionist in the Workshop and Secondary Actor is the Client. | Goal is to repair the client’s Vehicle | High | Client will ask receptionist to repair his vehicle in Workshop. Receptionist will ask the issues and inform mechanic about that issues. |

## USE CASE SPECIFICATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Identification and History** | | | |
| **Use Case ID:** | PRJWRKSP 3.0 | | |
| **Use Case Name:** | Repairing a Vehicle | **Version No:** |  |
| **End Objective:** | This will gather the info about issue of vehicle and send the details to desired mechanic | | |
| **Created by:** | Muteeb | **On (date):** |  |
| **Last Update by:** |  | **On (date):** |  |
| **Approved by:** |  | **On (date):** |  |
| **User/Actor:** | Receptionist | | |
| **Business Owner Name:** | Workshop Owner  Kashif | **Contact Details:** |  |
| **Trigger:** | It is triggered by clicking Repair in Main Menu section | | |
| **Frequency of Use:** | 20 times a week | | |

|  |
| --- |
| **Preconditions** |
| User initializes the program and system takes user to main menu |

|  |  |  |
| --- | --- | --- |
| **Basic Flow:**   1. **User initializes the program which takes user to a welcome screen and after that main menu will appear from where use selects ‘Repair’ option and then system will take user to Repair section. User selects the desired client from client list. After selecting a client program will take user to Repair section where user will gather info about issues and send it to Mechanic.** | | |
| **Step** | **User Actions** | **System Actions** |
| **1** | User initializes the program | System prompt with Welcome message and then displays the main menu. |
| **2** | User selects the option Repair | System prompts user with the list of client to select from |
| **3** | User selects client | System takes user to Repair section |
| **4** | User selects from types of issue that has to be resolved | System gathers the data and send it to the Mechanic |

|  |  |  |
| --- | --- | --- |
| **Alternate Flow** | | |
| **Step** | **User Actions** | **System Actions** |
| **3.1** | User clicks Oil Change | System gathers the info and send it to the Mechanic |
| **3.2** | User clicks Air Refill | System gathers the info and send it to the Mechanic |

|  |  |  |
| --- | --- | --- |
| **Exception Flow** | | |
| **1** | N/A | N/A |

|  |
| --- |
| **Post conditions** |
| System prompt the message to take user to Invoice section |

|  |
| --- |
| **Includes or Extension Points** |
|  |

|  |
| --- |
| **Special Requirements** |
|  |

|  |
| --- |
| **Business Rules** |
|  |

|  |
| --- |
| **Other Notes (Assumptions, Issues,)** |
| **Assumptions: N/A**   1. **Issues: N/A** |

## Invoice ACTORS, GOALS AND USE CASE BRIEFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Actors** | **Task-level Goal** | **Priority** | **Brief** |
| Actor is Receptionist in the Workshop and Secondary Actor is the Client. | Goal is to make an Invoice of the client’s repair job | High | Client will ask receptionist to make invoice of the repair job. Receptionist will add the repair jobs in the invoice and calculate the total. |

## USE CASE SPECIFICATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Identification and History** | | | |
| **Use Case ID:** | PRJWRKSP 3.0 | | |
| **Use Case Name:** | Making an Invoice | **Version No:** |  |
| **End Objective:** | This will gather the info about repair job and add it to Invoice with charges and Total Amount | | |
| **Created by:** | Muteeb | **On (date):** |  |
| **Last Update by:** |  | **On (date):** |  |
| **Approved by:** |  | **On (date):** |  |
| **User/Actor:** | Receptionist | | |
| **Business Owner Name:** | Workshop Owner  Kashif | **Contact Details:** |  |
| **Trigger:** | It is triggered by clicking Invoice in Main Menu section | | |
| **Frequency of Use:** | 20 times a week | | |

|  |
| --- |
| **Preconditions** |
| System sends the repair job info to mechanic |

|  |  |  |
| --- | --- | --- |
| **Basic Flow:**   1. **User initializes the program which takes user to a welcome screen and after that main menu will appear from where use selects ‘Repair’ option and then system will take user to Repair section. User selects the issues in Repair section. After sending the info to the Mechanic system prompts to make an Invoice and takes user back to main menu from where user selects Invoice option. Systems gathers repair job info and its charges and calculate the total and create an invoice in specific format** | | |
| **Step** | **User Actions** | **System Actions** |
| **1** | User initializes the program | System prompt with Welcome message and then displays the main menu. |
| **2** | User selects the option Repair | System prompts user with the list of client to select from |
| **3** | User selects client | System takes user to Repair section |
| **4** | User selects from types of issue that has to be resolved | System gathers the data and send it to the Mechanic |
| **5** | User clicks the option Make Invoice | System takes user back to Main Menu |
| **6** | User selects Invoice | System gathers info about repair jobs and its charges and total bill and create and invoice in specific format |

|  |  |  |
| --- | --- | --- |
| **Alternate Flow** | | |
| **Step** | **User Actions** | **System Actions** |
|  | N/A | N/A |
|  | N/A | N/A |

|  |  |  |
| --- | --- | --- |
| **Exception Flow** | | |
| **1** | N/A | N/A |

|  |
| --- |
| **Post conditions** |
| System prompt to select the Payment Method |

|  |
| --- |
| **Includes or Extension Points** |
|  |

|  |
| --- |
| **Special Requirements** |
|  |

|  |
| --- |
| **Business Rules** |
|  |

|  |
| --- |
| **Other Notes (Assumptions, Issues,)** |
| **Assumptions: N/A**   1. **Issues: N/A** |

## Payment method ACTORS, GOALS AND USE CASE BRIEFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Actors** | **Task-level Goal** | **Priority** | **Brief** |
| Actor is Receptionist in the Workshop and Secondary Actor is the Client. | Goal is to select the payment method the client desires | High | Receptionist will ask client for desired payment method and add it to the invoice |

## USE CASE SPECIFICATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Identification and History** | | | |
| **Use Case ID:** | PRJWRKSP 4.0 | | |
| **Use Case Name:** | Payment Method | **Version No:** |  |
| **End Objective:** | This will select the payment method and add it to the invoice | | |
| **Created by:** | Muteeb | **On (date):** |  |
| **Last Update by:** |  | **On (date):** |  |
| **Approved by:** |  | **On (date):** |  |
| **User/Actor:** | Receptionist | | |
| **Business Owner Name:** | Workshop Owner  Kashif | **Contact Details:** |  |
| **Trigger:** | It is triggered by clicking Payment Option in Invoice section | | |
| **Frequency of Use:** | 20 times a week | | |

|  |
| --- |
| **Preconditions** |
| System creates an Invoice |

|  |  |  |
| --- | --- | --- |
| **Basic Flow:**   1. **User initializes the program which takes user to a welcome screen and after that main menu will appear from where use selects ‘Repair’ option and then system will take user to Repair section. User selects the issues in Repair section. After sending the info to the Mechanic system prompts to make an Invoice and takes user back to main menu from where user selects Invoice option. Systems gathers repair job info and its charges and calculate the total and create an invoice in specific format. System then prompt for Payment Method. User selects the payment method as the client desires.** | | |
| **Step** | **User Actions** | **System Actions** |
| **1** | User initializes the program | System prompt with Welcome message and then displays the main menu. |
| **2** | User selects the option Repair | System prompts user with the list of client to select from |
| **3** | User selects client | System takes user to Repair section |
| **4** | User selects from types of issue that has to be resolved | System gathers the data and send it to the Mechanic |
| **5** | User clicks the option Make Invoice | System takes user back to Main Menu |
| **6** | User selects Invoice | System gathers info about repair jobs and its charges and total bill and create and invoice in specific format and prompt to select Payment Method |
| **7** | User selects Payment Method Option | System add the Payment Method option to the Invoice |

|  |  |  |
| --- | --- | --- |
| **Alternate Flow** | | |
| **Step** | **User Actions** | **System Actions** |
| **7.1** | User selects Hard Cash | System add “Payment Via Hard Cash” to the Invoice |
| **7.2** | User selects Online Transfer | Systems Prompt with Workshop AC Number. |

|  |  |  |
| --- | --- | --- |
| **Exception Flow** | | |
| **1** | N/A | N/A |

|  |
| --- |
| **Post conditions** |
| System prompt to Print the Invoice |

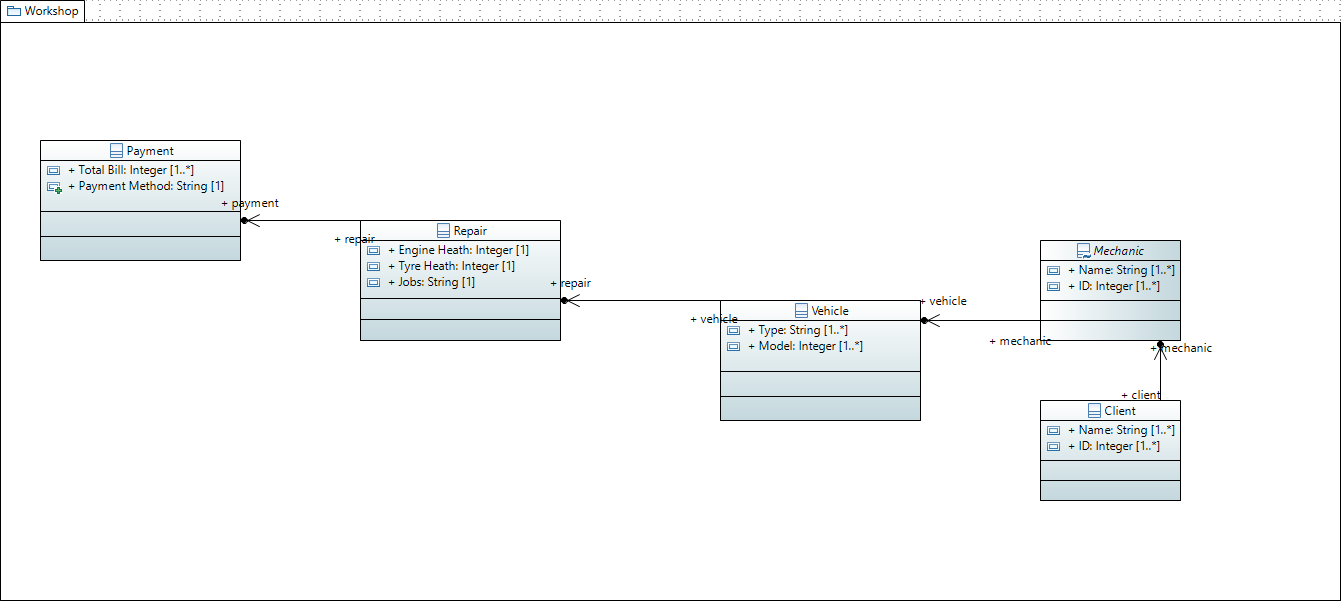
|  |
| --- |
| **Includes or Extension Points** |
|  |

|  |
| --- |
| **Special Requirements** |
|  |

|  |
| --- |
| **Business Rules** |
|  |

|  |
| --- |
| **Other Notes (Assumptions, Issues,)** |
| **Assumptions: N/A**   1. **Issues: Should there be PayPal Option?** |

# System Structural Model

**

**Appendix A: Glossary**