**REPORT**

# **C Programming on Multiple Platforms - Project Based Learning Approach**

NAME: Yukta Kulkarni

PS NO: 99007673

**Project Title: Tourism Management system**

[**C Programming on Multiple Platforms - Project Based Learning Approach** 1](#_Toc97290175)

[**1. Requirements** 3](#_Toc97290176)

[**1.1 Description** 3](#_Toc97290177)

[**1.4 High level requirements** 4](#_Toc97290178)

[**Table 1.4** 4](#_Toc97290179)

[**1.5 Low level requirements** 4](#_Toc97290180)

[**1.6 SWOT Analysis** 5](#_Toc97290181)

[**1.7 4W's-1H** 6](#_Toc97290182)

[ **WHO -** 6](#_Toc97290183)

[ **WHAT -** 6](#_Toc97290184)

[ **WHEN -** 6](#_Toc97290185)

[ **WHERE -** 6](#_Toc97290186)

[ **HOW -** 6](#_Toc97290187)

[**1.8** **Best Method Followed** 7](#_Toc97290188)

[**4. Test plan and Output** 14](#_Toc97290189)

[**4.1 High level Test plan** 14](#_Toc97290190)

[**4.2 Low level Test plan** 15](#_Toc97290191)

[**4.3 Best Practices** 16](#_Toc97290192)

# **1. Requirements**

# **1.1 Description**

* The Tourism management system allows user to access all the details regarding complete tour package such as Traveller details, amount, events, etc. The main purpose of this system is to help tourism companies to manage customer details and hotels.

**1.2 Identifying feature**

1. Choice of international and India tour.
2. Display of different places based on selected tour type.
3. Traveler details are stored.
4. Generating nodes based on number of Traveler.
5. Display of booked tour receipt.

**1.3 State of Art/Research**

Tourism management project helps to automate all the process of travel and tourism, that deals with booking, display of visiting places with package, user details etc. This design is built using simple C programming structures. The main purpose is to help tourism companies to manage customer details and hotels. This design is very user friendly, easy to access the details regarding tour package with confirmation details displayed on generated receipt.

## **1.4 High level requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ld** | **Description** | **Category** | **Status** |
| HLR1 | User menu display regarding tour | Technical | Implemented |
| HLR2 | Choose of tour package | Technical | Implemented |
| HLR3 | Details of Traveler | Technical | Implemented |
| HLR4 | Generation of booked package receipt | Technical | Implemented |

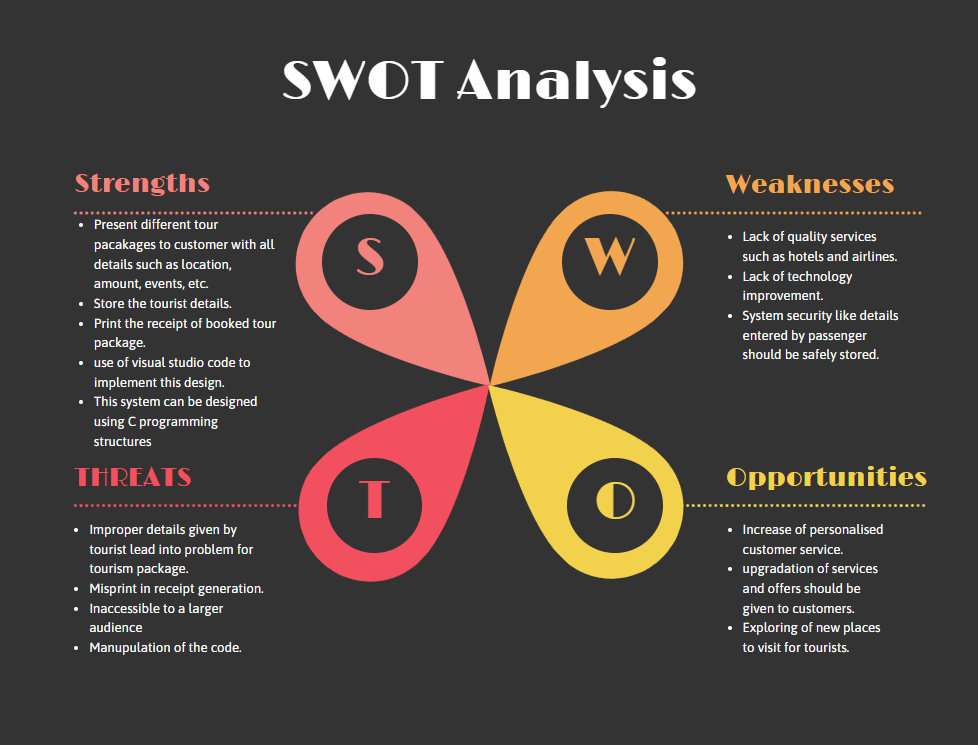
## **Table 1.4**

## **1.5 Low level requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ld** | **Description** | **Category** | **Status** |
| LLR1 | user choice to select international or national tour | Technical | Implemented |
| LLR2 | user choice of different tour package | Technical | Implemented |
| LLR3 | Details of Traveler such as date, gender, name, age | Technical | Implemented |
| LLR4 | Receipt if generated regarding booked package with details | Technical | Implemented |

**Table 1.5**

# **1.6 SWOT Analysis**



**Figure 1.6**

# **1.7 4W's-1H**

## **WHO -**

This system will help the tourism company members to manage customer details and hotels.

## **WHAT -**

Provide easy actions that can be done by the customers to book their tour.

## **WHEN -**

Online management is trending in today's era and Tourism management system plays an important role in present century.

## **WHERE -**

This is implemented in tourism companies and hotels.

## **HOW -**

It can be used by everyone, its user friendly, ease of use.

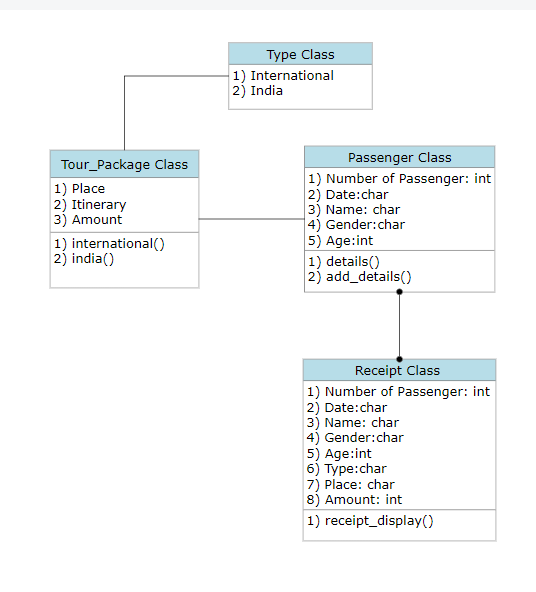
# **1.8** **Best Method Followed**

* To improve the readability of code there is usage of modular functions in this design.
* Structures are used to combine the data of different data types together.
* Usage of arrays so that similar elements can be stored at contiguous memory locations.
* Pointers are used to locate memory dynamically.
* Header-files are included for standard input/output operations.
* Unit testing is done to meet the quality standards before it is deployed.

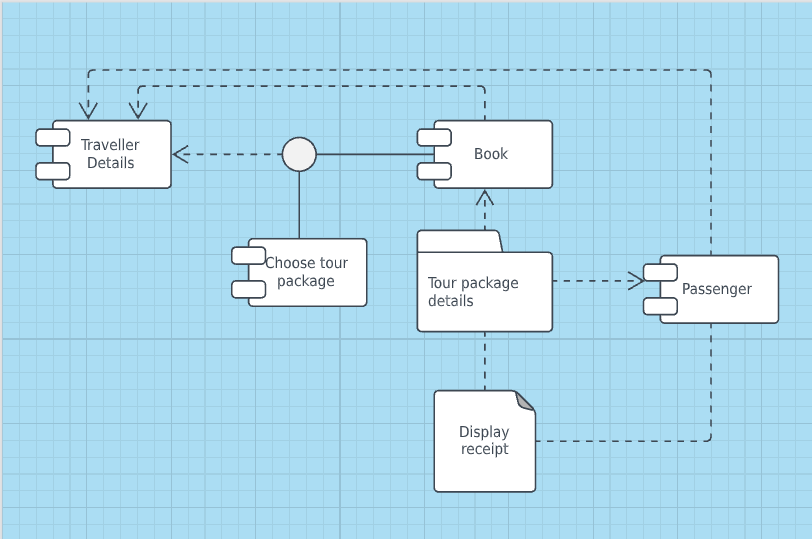
**2. Design**

**2.1 Structural diagram**

**2.1.1 Class diagram**

 **Figure 2.1.1**

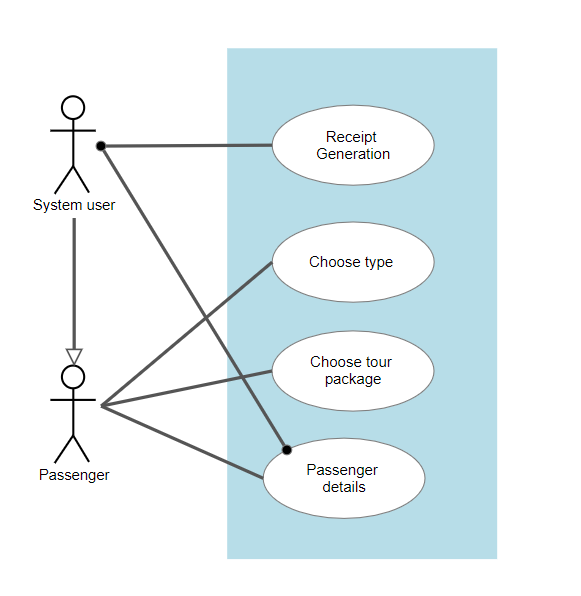
**2.1.2 Component diagram**



**Figure 2.1.1**

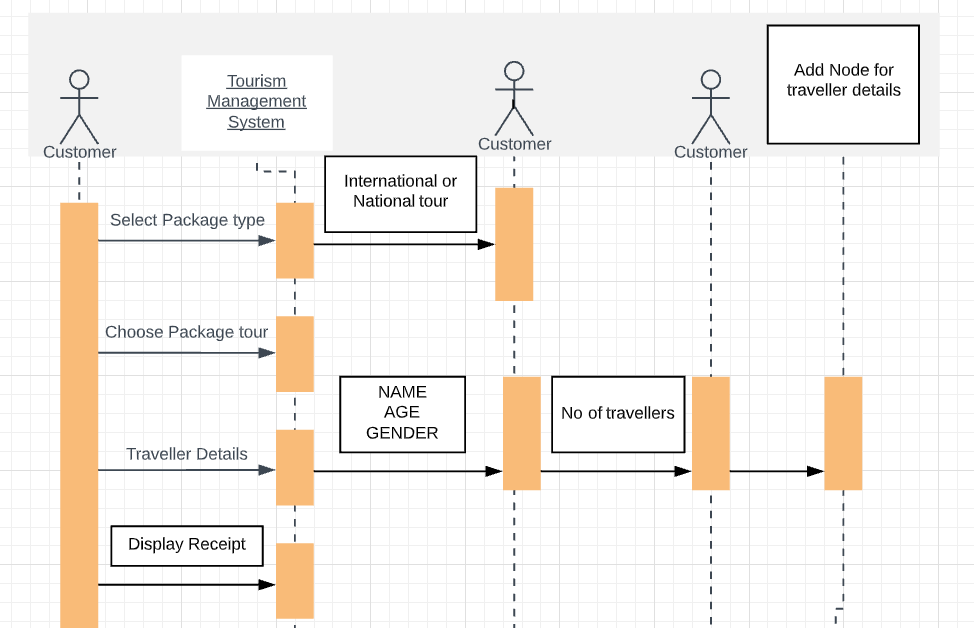
**2.2 Behavioral diagram**

**2.2.1 Use case diagram**



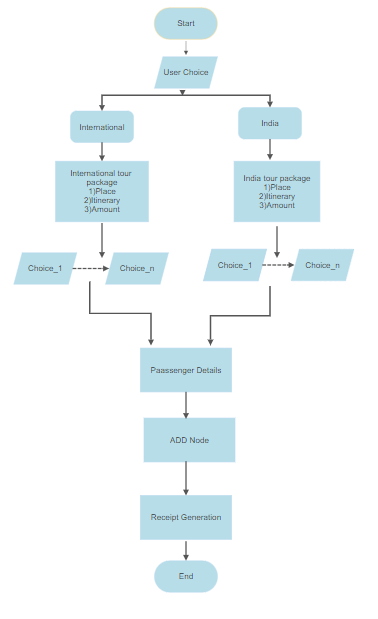
**Figure 2.2.1**

**2.2.2 Sequence diagram**



**Figure 2.2.2**

**2.3 Flowchart**



**Figure 2.3**

**3. IMPLEMENTATION**

This design is implemented in visual studio code using WSL Linux commands.

1. Ask the user to choose between international or India tour package.

* This operation is implemented using switch case statements.

2. Display the particular tour packages with amount.

* This function is implemented in if-else condition statement after user’s choice.

3. Traveler details are entered like number of Travelers, name, age, gender.

* After the user entered details, we add nodes using structures based on number of Travelers.

4. Finally a receipt is generated of booked package tour, containing name, age, gender, tour package, date, etc.

# **4. Test plan and Output**

## **4.1 High level Test plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected i/p** | **Expected o/p** | **Actual o/p** | **Type of test** |
| H\_01 | Check for the tour menu display | Program execution | Formatted menu display | Formatted menu display | Requirement |
| H\_02 | check if all package tour is available | user choice | Formatted menu display | Formatted menu display | Requirement |
| H\_03 | Check the details to enter for Traveler menu is displayed | enter details | Proper display of details menu and read the input details | Proper display of details menu and read the input details | Requirement |
| H\_04 | display of receipt | Program execution | Formatted Package tour display | Formatted Package tour display | Requirement |

**Table 4.1**

## 

## **4.2 Low level Test plan**

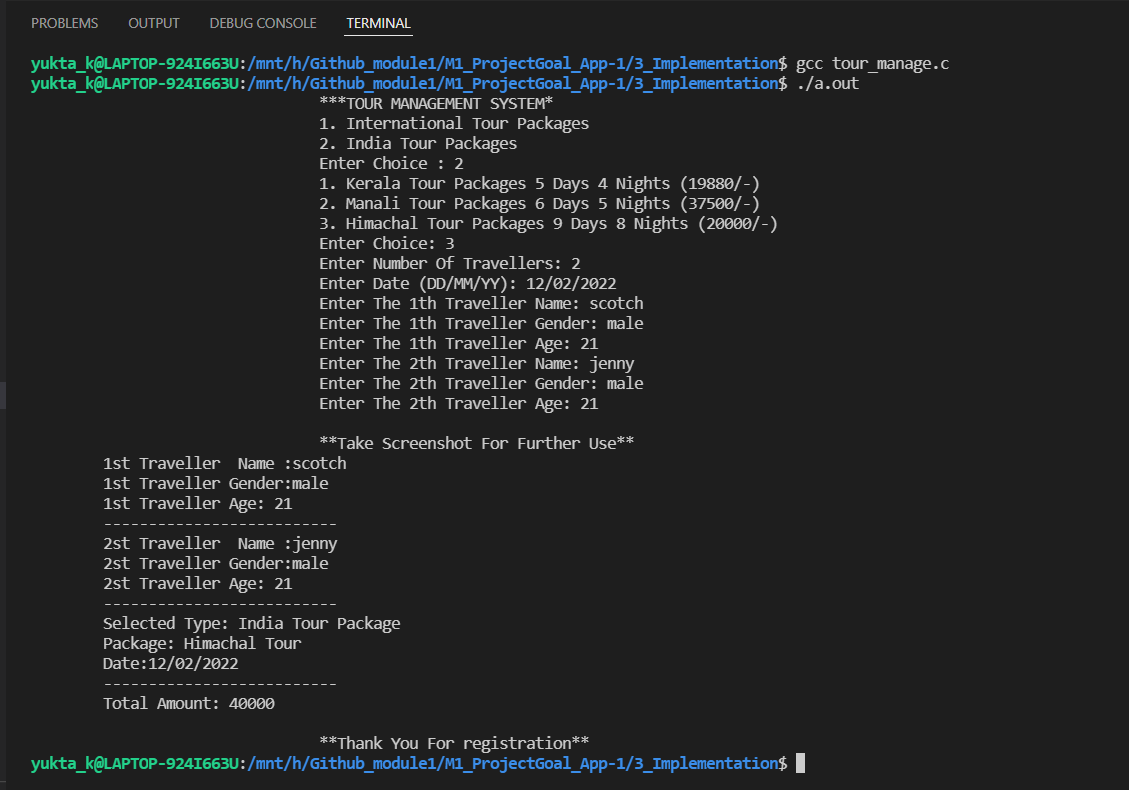
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected i/p** | **Expected o/p** | **Actual o/p** | **Type of test** |
| L\_01 | Display of choice between international and India tour | User choice | Based on the choice particular function is called | Based on the choice particular function is called | Technical |
| L\_02 | Display of different tour packages for the corresponding selected type | user choice | Reading of input and copying data to string | Reading of input and copying data to string | Technical |
| L\_03 | Traveler details should be entered by user | enter details | Reading of input and adding node function is called based on number of traveler | Reading of input and adding node function is called based on number of traveler | Technical |
| L\_04 | Booked tour receipt is generated | - | Detailed information of traveler is displayed with selected tour package | Detailed information of traveler is displayed with selected tour package | Technical |

**Table 4.2**

# **4.3 Best Practices**

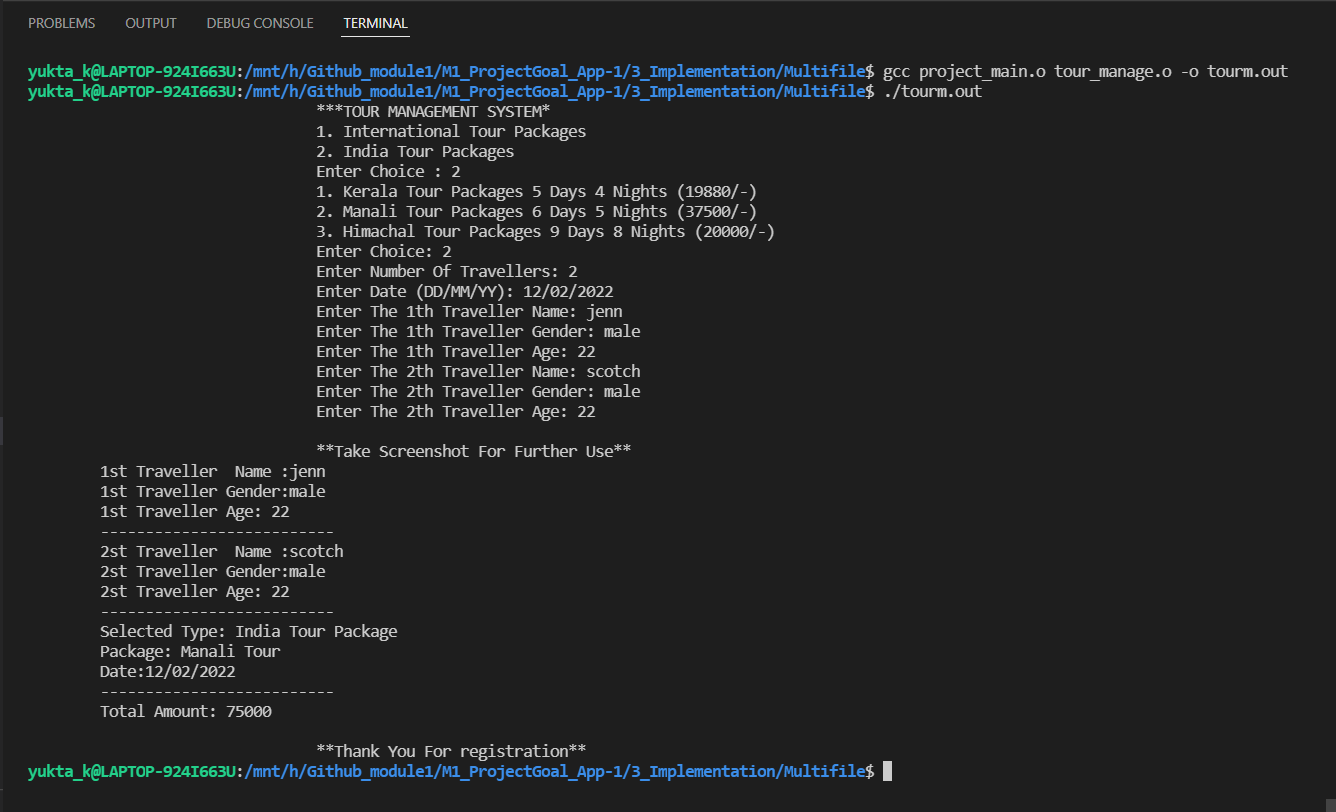
1. Presentation of both low level and high level test plan.
2. Testing of expected output with actual output.
3. Mentioned all possible inputs.
4. Requirement and Technical type of testing is done

**4.4 Output:**



**Figure 4.4**

**4.4.1 Multifile output:**



**Figure 4.4.1**