

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590 018



A Mini -Project Work on

Game House Shop

A Dissertation work submitted in partial fulfillment of the requirement
for the award of the degree

Bachelor of Engineering
In
Information Science & Engineering

Submitted by

Abhinav Anand
Abhishek Prasad

1AY18IS004
1AY18IS006

Under the guidance of

Prof. Arun K.H

Assistant Professor



DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
ACHARYA INSTITUTE OF TECHNOLOGY

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI. APPROVED BY AICTE, NEW DELHI,
ACCREDITED BY NAAC, NEW DELHI)

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107

2020-21

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ACHARYA INSTITUTE OF TECHNOLOGY

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI. APPROVED BY AICTE, NEW DELHI, ACCREDITED BY NAAC, NEW DELHI)

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107



Certificate

This is to Certify that the Mini-Project work entitled **"Game House Shop "** is a bonafide work carried out by **Abhinav Anand (1AY18IS004)** and **Abhishek Prasad (1AY18IS006)** in partial fulfillment for the award of the degree of **Bachelor of Engineering in Information Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project has been approved as it satisfies the academic requirements in respect of Project work prescribed for the Bachelor of Engineering Degree.

Prof. Arun K.H
Guide

Prof. Marigowda C K
HOD

Name of the Examiners

1. _____
2. _____

Signature with date

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this mini-project would be incomplete without the mention of the people who made it possible through constant guidance and encouragement.

We would take this opportunity to express our heart-felt gratitude to **Sri. B. Premnath Reddy**, Chairman, Acharya Institutes and **Dr. Prakash M R**, Principal, Acharya Institute of Technology for providing the necessary infrastructure to complete this mini-project.

We wish to express our deepest gratitude and thanks to **Prof. Marigowda C K**, Head of the Department, Information Science and Engineering and the project coordinator **Prof Pakruddin B** and **Prof Raushan Kashypa** for their constant support.

We wish to express sincere thanks to my guide **Prof. Arun K.H**, Assistant Professor, Department of Information Science and Engineering for helping us throughout and guiding us from time to time.

A warm thanks to all the faculty of Department of Information Science and Engineering, who have helped us with their views and encouraging ideas.

Abhinav Anand (1AY18IS004)

Abhishek Prasad (1AY18IS006)

ABSTRACT

The purpose of Game House Shop application is to automate the existing manual system by the help of computerized equipments and full-fledged computer software. By this application an administrator can store game details, retrieve game information, update and delete the game details and do the same operations with the data of employee. The customer here can search for game details and see the available game list and their quantity.

It's purpose is to fulfill the requirements of customer and shop owner, so that their information can be stored for a longer period with easy accessing and manipulation of the same. This application avoids user to maintain manual register information of the games details and the employee details. The objective and scope of this project is to record the various games available in the market and also manage employee data. It simplifies the task and reduce the paper work.

As manual computing system becomes more numerous, complex the need for the systematic approaches development becomes increasingly apparent. The objective of project work-Game House Shop is designing a convenient framework including game to the user on one platform. The primary goal of this project is to develop good software to overcome the existing problem caused by manual systems. Basically, the project describes how to manage for good performance and better services for the clients.

TABLE OF CONTENTS

Acknowledgement	i
Abstract	ii
1. Introduction	1
1.1 Introduction to File Structure	1
1.2 Fundamental Operation on File	
1.3 Services Provided to the USER (CUSTOMER)	2
1.4 Services Provided to the ADMINISTRATOR (OWNER)	3
2. System Requirements	6
2.1 Hardware Requirements	6
2.2 Software Requirements	6
3. System Overview	7
3.1 Problem Definition	7
3.2 Sequential Search	8
4. Design	10
4.1 Data Flow Diagram	10
4.2 Class Diagram	11
5. Implementation	12
6. Testing	14
7. Snapshots	18
Conclusion & Future Enhancements	27
Bibliography	28

CHAPTER 1

INTRODUCTION

1.1 Introduction to File Structures

File Structures is the Organization of Data in Secondary Storage Device in such a way that minimize the access time and the storage space. A File Structure is a combination of representations for data in files and of operations for accessing the data. A File Structure allows applications to read, write and modify data. It might also support finding the data that matches some search criteria or reading through the data in some particular order.

“**File organization**” refers to the logical relationship among the various records that constitute the file, particular with respect to means of identification access to any specific record. “File structure” refers to the format of label and data blocks and of any logical record control, information. The organization of the given file maybe sequential relative, or indexed.

Co-Sequential Processing

Co-sequential Processing is the coordinated processing of two or more sequential lists to produce single output list. Sometimes the processing results in a merging, or union, sometimes it results in a matching, or intersection and other times the operation is a combination of matching and merging of the items in the lists.

Aim of the File Structure

The need for file structures

- Ideally, we would like to get the information we need once access to the disk
- If it is impossible to get what we need in one access, we want structures that allow us to find the target information with as few accesses as possible.
- We want our file structures to group information so we are likely to get thing We need with only trip to the disk.

1.2 Fundamental Operations on File

Open: A function or system call that makes a file ready for use. It may also bind a logical filename to a physical file. Its arguments include the logical filename, the physical filename and may also include information on how the file is to be accessed.

Close: A function or system call that breaks the link between the logical filename and the corresponding physical filename.

Create: A function or system call that causes a file to be created on secondary storage and may also binds a logical filename to the file's physical filename.

Read: A function or a system call used to obtain input from a file or device. When viewed at the lowest level, it requires three arguments: a source file logical name corresponding to an open file; the destination address for the address and the size or amount of data to be read.

Write: A function or system call used to provide output capabilities. When viewed at the lowest level, it requires three arguments: a destination filename corresponding to an open file; the source address of the bytes that are to be written and the size or amount of data to be written.

Seek: A function or a system call that sets the read/write pointer to a specified position in a file. Languages that provide seeking functions allow programs to access specific.

1.3 Services provided to the USER (CUSTOMER)

The functions that the Game House Shop provides to the user (customer) are as follows:

1. DISPLAY: This operation is performed when the customer enters into the customer section and the clicks 1. When the customer selects option 1, it displays all the games available in the shop along with the details of the games and their quantity.

2. SEARCH: This function is used to search particular data from the System. By clicking on option 2 in the customer section, the customer can search for a particular game by using the game id and it will display that particular game details.

3. BACK: This option is used to go back on the options menu on the homescreen. This is performed by clicking on option 3 in the customer section.

4. EXIT : By clicking 4, the customer can exit from the application.

1.4 Services provided to the ADMINISTRATOR (OWNER)

The functions that the Game House Shop provides to the administrator (owner) are as follows:

1. LOGIN : The Admin can login into the Administrator section by providing the username and the password. There are totals five attempts for login, if suppose the admin gives the incorrect username or password five times then the application will terminate immediately.

After login two sections are available:

- i. **Game House stock management** - In this section the all the stock of game is managed by the admin. This section can be entered by selection option 1 after login.
- ii. **Game House Employee management** – In this section all the details of employee working in the GameHouse is managed by the admin. This section can be entered by selecting option 2 after login.

Game House Stock Management:

1. ADD : The Admin can add the details of the game in the application by clicking on option 1 . The details provided are game id, game name, game developer name, and the no. of copies of the games available. Here every game id is unique and duplicate id cannot be added.

2. DISPLAY : The Admin can display the list of games available in the store with the details of the game. This opertion is performed by selection option 2.

3. SEARCH : The Admin can search for a particular game details using this function. The admin has to provide the game id for searching and it will display the details of that particular game.

If the game id provided is wrong or not present in the file it will display game id not found.

4. UPDATE : The Admin can update the details of any existing game in the file, for example to update the quantity of the game that is already present in the file. This opertaion is done by

selecting option 4 and providing the game id of an existing game in the file and then updating the details. If the game id provided is wrong or not present in the file, it will display game id not found.

5. DELETE : The Admin can delete any game from the list which is out of stock or is outdated and is not available in the shop. This operation can be done by clicking option 5 and then providing the game id which has to be deleted. If the game id provided is wrong or not present in the file it will display game id not found.

6. BACK: This option is used to go back on the options menu of GameHouse stock management or GameHouse employee management. This is performed by clicking on option 6 .

7. GO TO HOMEPAGE : This option is used to directly go to the homepage of the application by clicking on option 7.

8. EXIT : By clicking 8, the Admin can exit from the application.

Game House Employee Management:

1. ADD : The Admin can add the details of the employee in the application by clicking on option 1. The details provided are employee id, employee name, game developer name, employee address, employee contact no., gender and age. Here every employee id is unique and duplicate id cannot be added.

2. DISPLAY : The Admin can display the list of employees working in the store with the details of the employee. This operation is performed by selection option 2.

3. SEARCH : The Admin can search for a particular employee details using this function. The admin has to provide the employee id for searching and it will display the details of that particular employee. If the employee id provided is wrong or not present in the file it will display employee id not found.

4. UPDATE : The Admin can update the details of any existing employee details in the file, for example to update any mistakes in the details of the employee . This operation is done by selecting option 4 and providing the employee id of an existing employee detail in the file and then updating the details. If the employee id provided is wrong or not present in the file it will display employee id not found.

5. DELETE : The Admin can delete any employee from the list who has left the job and is not working anymore in GameHouse shop. This operation can be done by clicking option 5 and then providing the employee id which has to be deleted. If the employee id provided is wrong or not present in the file it will display employee id not found.

6. BACK: This option is used to go back on the options menu of GameHouse stock management or GameHouse employee management. This is performed by clicking on option 6 .

7. GO TO HOMEPAGE : This option is used to directly go to the homepage of the application by clicking on option 7.

8. EXIT : By clicking 8, the Admin can exit from the application.

CHAPTER 2

HARDWARE AND SOFTWARE REQUIREMENTS

2.1 Software requirements

- **Operating System :** Windows® Vista 32/64-Bit / Windows®7 32/64-Bit / Windows® 8 32/64-Bit, Windows® 8.1 32/64-Bit, Windows® 10 32/64-Bit
- **Documentation Tool:** MS Word
- **Development Tool:** Dev C++, Turbo C++

2.2 Hardware requirements

- **Processor:** Intel Core2 Quad @ 2.4Ghz
- **RAM:** 4GB RAM
- **Memory:** 16 GB Hard disk

CHAPTER 3

SYSTEM OVERVIEW

3.1 Problem Definition

Currently many shops follow the process that they manually keep record of the employee details in a register and also see the stock manually by checking the storage of the store . This process involves a lot of manual tasks and it is very hectic to do the same. Maintaining every details of employee by manually writing in a register book and checking every time the storage of the shop for stock is a very hectic and slow process. Hence the possibility of getting lost of the register of employee details or tampering with the details of employee increases , also no proper stock of games can be maintained by manually doing this work. The amount of time taken for checking the stock would require a lot of time and effort if the quantity of games in the store is very high. . Overall the effort required is much more as compared to what it should be and the amount of human touch is a lot and that can cause a lot of problems as the chances of error is quite high relative to what a system can do.

3.1.1 Solution on problem

Creating new software we should analyze what is the basic need of the software. Analysis is nothing but a planning of creation of software to get proper output from it. Analysis is detail study of protects that you want to show in your software solving problems. The basic need of the software is to save the time of the user with the help of all useful information. And also to maintain the collection of data in your computer systematically" so that it's easy to understand. The proposed application provides lots of facility to the admin to store information of the games and the employees and it provides information in quick time in a systematic manner.

The processing time on the data is very fast. It provides required data quickly to the user and also in specified manner to the user. All the information of stored and updated is given to the user and also the the user can retrieve information according to the requirement of the user. Today it is very difficult to maintain these details manually. This is where our project Game House shop

comes in to replace all these manual tasks. It also removes a large amount of human task handling and thus decreases the chances of any sort of errors taking place.

3.2 The Sequential Search

Data items are stored in a collection such as a list, we say that they have a linear or sequential relationship. Each data item is stored in a position relative to the others. In Python lists, these relative positions are the index values of the individual items. Since these index values are ordered, it is possible for us to visit them in sequence. This process gives rise to our first searching technique, the **Sequential search**.

3.2.1 Analysis of Sequential Search

To analyze searching algorithms, we need to decide on a basic unit of computation. This is typically the common step that must be repeated in order to solve the problem. For searching, it makes sense to count the number of comparisons performed. Each comparison may or may not discover the item we are looking for.

We make another assumption here. The list of items is not ordered in any way. The items have been placed randomly into the list. In other words, the probability that the item we are looking for is in any particular position is exactly the same for each position of the list.

To check the item is in the list, the only way to know it is to compare it against every item present. If there are n items, then the sequential search requires n comparisons to discover that the item is not there. In the case where the item is in the list, the analysis is not so straightforward. There are actually three different scenarios that can occur. In the best case we will find the item in the first place we look, at the beginning of the list. We will need only one comparison. In the worst case, we will not discover the item until the very last comparison, the n th comparison.

On average, we will find the item about halfway into the list; that is, we will compare against $n/2$ items. However, that as n gets large, the coefficients, no matter what they are, become insignificant in our approximation, so the complexity of the sequential search, is $O(n)$.

CHAPTER 4

DESIGN

4.1 DATA FLOW DESIGN

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

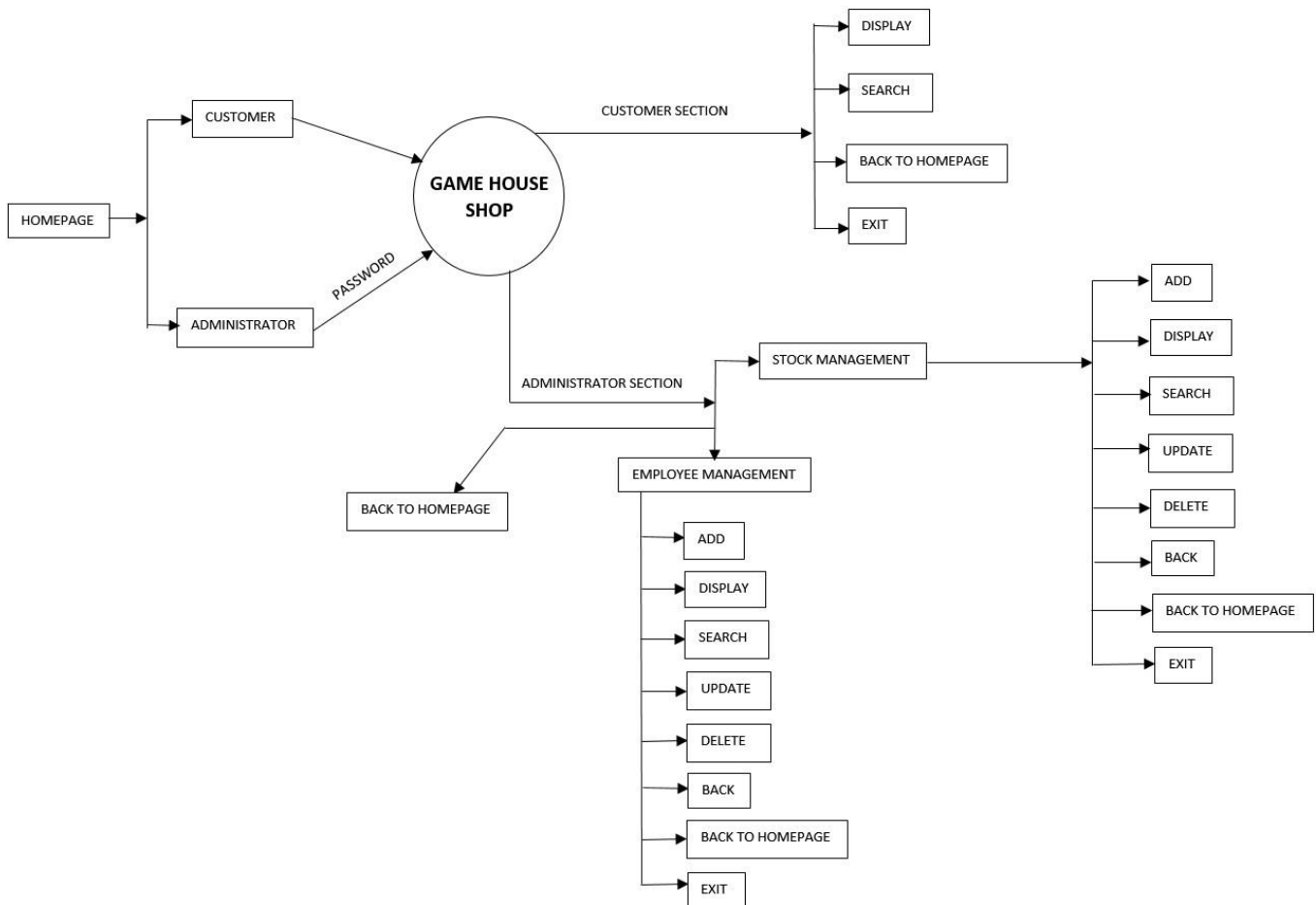


Fig 4.1: Data Flow Design

4.2 CLASS DIAGRAM

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

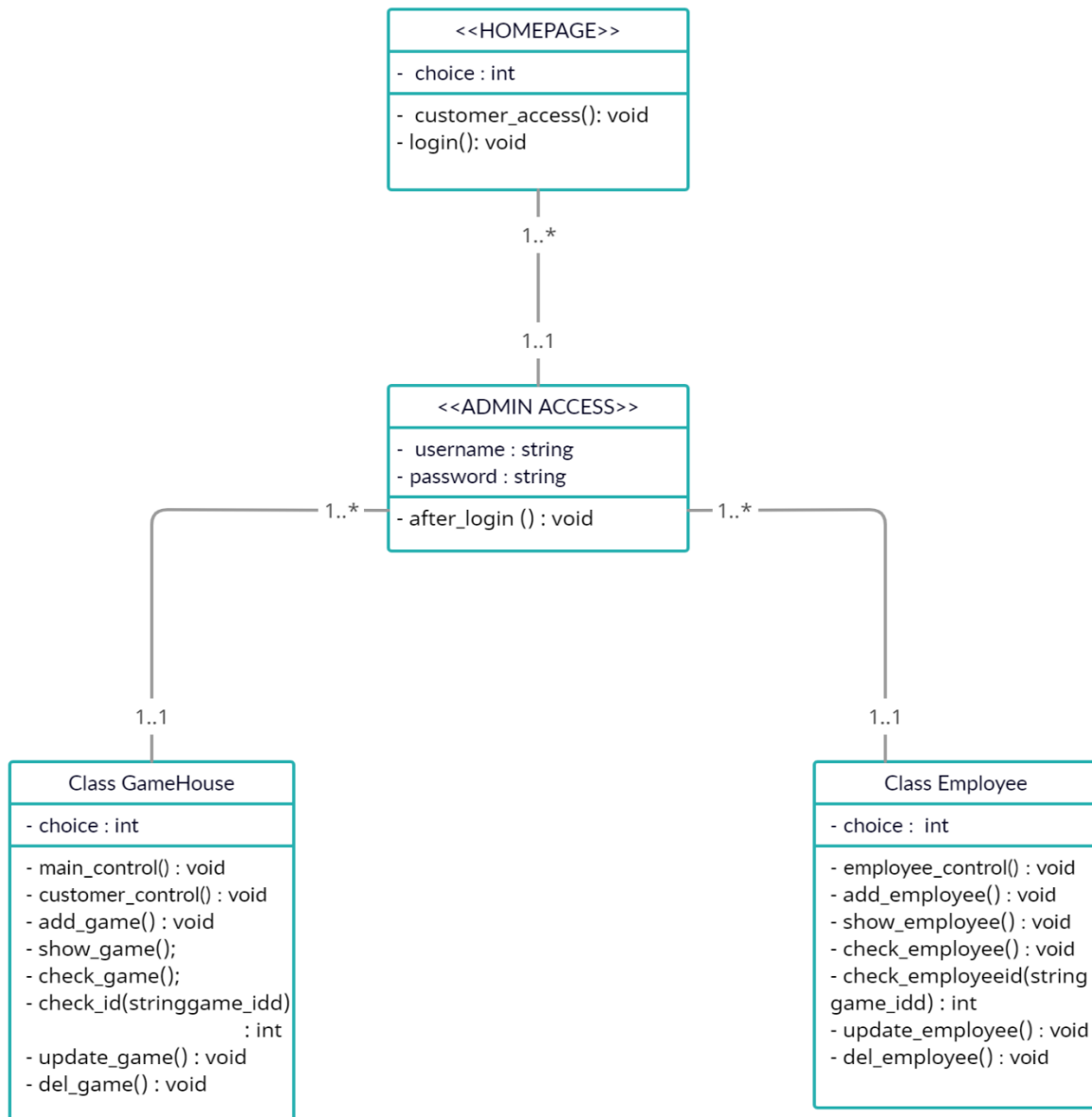


Fig 4.2: Class Diagram

CHAPTER 5

IMPLEMENTATION

Introduction:

The project is completely command based with proper implementation of File Structure concepts and its implementation is done using C++. We have used the object oriented concept in the implementation of design. The concept of class and methods is used to a large extent. The file structure concepts used in this project are File handling property and Sequential Search. Open, Read and Write operations are performed on the file to store and retrieve the details.

Functions and the members involved:

Design the class gamehouse. Each function of this class represents operations being performed on with the details of game file .

Members functions in gamehouse class includes :

- void main_control();
- void customer_control();
- void add_game();
- void show_game();
- void check_game();
- int check_id(string game_id);
- void update_game();
- void del_game();

Design the class gamehouse_employee. Each function of this class represents operations being performed on with the details of employee details file .

Members functions in gamehouse_employee class includes :

- void employee_control();
- void add_employee();
- void show_employee();

- void check_employee();
- int check_employeeid(string game_idd);
- void update_employee();
- void del_employee();

Functions defined outside the class:

- int login();
- void homepage();
- void customer_Access();
- void after_login();

Driver function:

- int main();

CHAPTER 6

TESTING

The integral part of any system's development life cycle is testing without which the system developed is sure to fail and result in loss of economic and manpower investments besides user's dissatisfaction and downfall of reputation. System testing is the stage of implementation, which aims at ensuring that the system works accurately and efficiently before actual operation commences. No program or system design is perfect, communication between the user and the designer is not always complete or clear. All this can result in errors.

Another reason for system testing is its utility as a user oriented vehicle before implementation. The application system is worthless if does not meet user needs, thus the system should be tested to see whether it meets the user requirements.

Testing here is conducted in bottom up approach as follows:

- Module testing: Here testing is done at each module level. Each case has been thoroughly tested to discover pitfalls.
- System testing: Here testing is done after all the modules have been integrated.

Test cases

Sl.No	Module	Test Case Description	Input	Expected Output	Actual Output	Status
1	Gamehouse Home Page	Verify whether game house home page displays.	Run the program.	Displays game house home page	Displays gamehouse home page	Pass
2	Menu Display	Displays the menu list.	Enter your choice.	Goes to the part of code you entered for to execute.	Goes to the part of code you entered for to execute.	Pass

Sl.No	Module	Test Case Description	Input	Expected Output	Actual Output	Status
3	Customer Access Menu Display	Displays all the options available for the customer	Enter your choice	Goes to the part of code you entered for to execute.	Goes to the part of code you entered for to execute.	Pass
4	Search for a particular game	Searches for the game customer wants	Enter the Game ID	Displays the info related to the game id entered	Displays the info related to the game id entered	Pass
5	Display all games available	Displays all the game present in the file	Enter choice 2 to display	Displays all the games	Displays all the games	Pass
6	Go back to homepage	Goes back to the homepage menu	Enter choice 3 to go	Displays the homepage menu	Displays the homepage menu	Pass
7	Administrat or options	Display the menu for gamehouse and employee record	Enter choice 2 on the homepage	Asks the admin to login first	Asks the admin to login first	Pass
8	Admin Login	Asks to enter Username and the password	Enter Username and password to get access	Displays menu for gamehouse management and employee record management	Displays menu for gamehouse management and employee record management	Pass
9	Admin Menu	Displays the admin menu list	Enter your choice	Goes to the part of code you entered for to execute.	Goes to the part of code you entered for to execute.	Pass
10	GameHouse Mgmt Page	Displays all the options available	Enter your choice	Performs action according to the selected choice	Performs action according to the selected choice	Pass

11	Add games	Add a new game to the file	Enter all the info asked for	Game gets added if not already present	Game gets added if not already present	Pass
12	Display all Games	Displays all the games available in the shop with their information	Enter 2 to display all games	Displays all the games	Displays all the games	Pass
13	Search for a particular game	Searches for the game admin wants to check	Enter the Game ID	Displays the info related to the game id entered	Displays the info related to the game id entered	Pass
14	Update game info	Updates all the fields of the selected Game Id	Enter the Game ID	Game gets Updated	Game gets Updated	Pass
15	Delete Game	Deletes the record of the game	Enter the Game ID	Game gets Deleted	Game gets Deleted	Pass
16	Go Back to Admin Menu	Displays the options on Admin Menu	Enter 6 to go back	Displays Admin Menu	Displays Admin Menu	Pass
17	Go Back to Homepage	Goes back to the homepage menu	Enter choice 3 to go	Displays the homepage menu	Displays the homepage menu	Pass
18	Employee Record Mgmt Page	Displays all the options available	Enter your choice	Performs action according to the selected choice	Performs action according to the selected choice	Pass
19	Add Employee	Add a new Employee to the file	Enter all the info asked for	Employee gets added if not already present	Employee gets added if not already present	Pass
20	Display all Employees	Displays all the Employees working in the shop	Enter 2 to display all Employees	Displays all the Employees	Displays all the Employees	Pass

21	Search for a particular Employee	Searches for the Employee admin wants to check	Enter the Employee ID	Displays the info related to the Employee	Displays the info related to the Employee	Pass
22	Update Employee info	Updates all the fields of the selected Emp Id	Enter the Employee ID	Employee gets Updated	Employee gets Updated	Pass
23	Delete Employee	Deletes the record of the Employee	Enter the Employee ID	Employee gets Deleted	Employee gets Deleted	Pass
24	Go Back to Admin Menu	Displays the options on Admin Menu	Enter 6 to go back	Displays Admin Menu	Displays Admin Menu	Pass
25	Go Back to Homepage	Goes back to the homepage menu	Enter choice 3 to go	Displays the homepage menu	Displays the homepage menu	Pass
26	Exit	Exits from the program	Select choice to exit	Program gets Terminated	Program gets Terminated	Pass

CHAPTER 7

SNAPSHOTS

This is the homepage of Gamehouse which displays options for customer and admin section

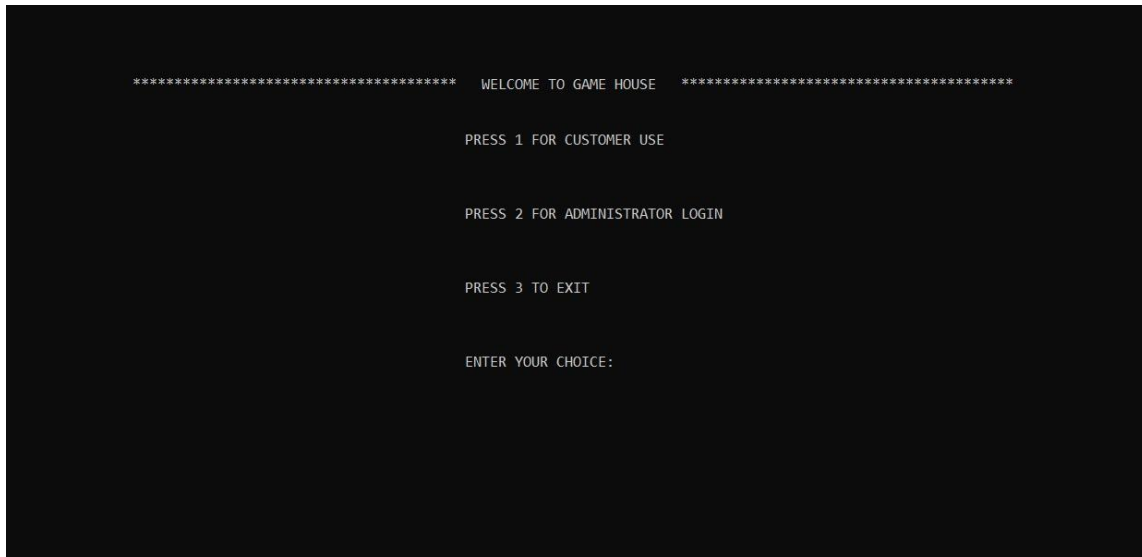


Fig 7.1 : Game House Homepage

This is the customer section of Gamehouse

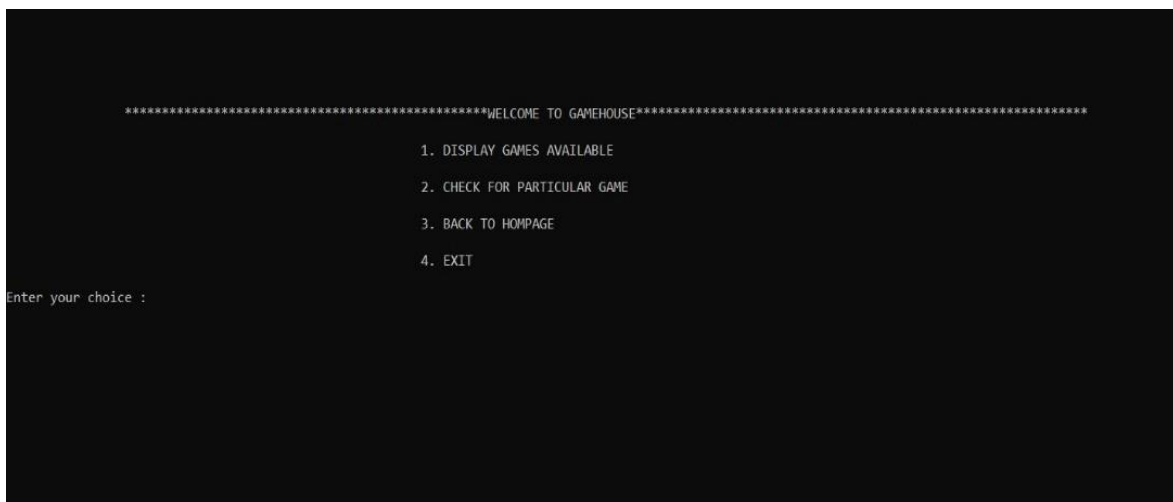


Fig 7.2 : Customer section

This is the display page of game list for the customer section

```
THE GAMES AVAILABLE

GAME ID : #1102
GAME NAME : DYING_LIGHT
GAME DEVELOPER NAME : TECHLAND
No. of copies of GAME : 15

GAME ID : #1103
GAME NAME : METRO_EXODUS
GAME DEVELOPER NAME : 4A_GAMES
No. of copies of GAME : 30

GAME ID : #1104
GAME NAME : RESIDENT_EVIL_BIOHAZARD
GAME DEVELOPER NAME : CAPCOM
No. of copies of GAME : 10

GAME ID : #1105
GAME NAME : NEED_FOR_SPEED
GAME DEVELOPER NAME : EA_SPORTS
No. of copies of GAME : 35

GAME ID : #1106
GAME NAME : FIFA_2021
GAME DEVELOPER NAME : EA_SPORTS
No. of copies of GAME : 25

GAME ID : #1107
GAME NAME : BATTLEFIELD
GAME DEVELOPER NAME : EA_GAMES
No. of copies of GAME : 12
```

Fig 7.3 : Display for customer

This the search screen for customer where a particular game details is shown to customer

```
THE GAME DETAILS PROVIDED BELOW

GAME ID : #1102
GAME NAME : DYING_LIGHT
GAME DEVELOPER NAME : TECHLAND
No. of copies of GAME : 15

Press any key to continue . . .
```

Fig 7.4: Search screen for customer

This is the admin login section where the admin has to login using username and password

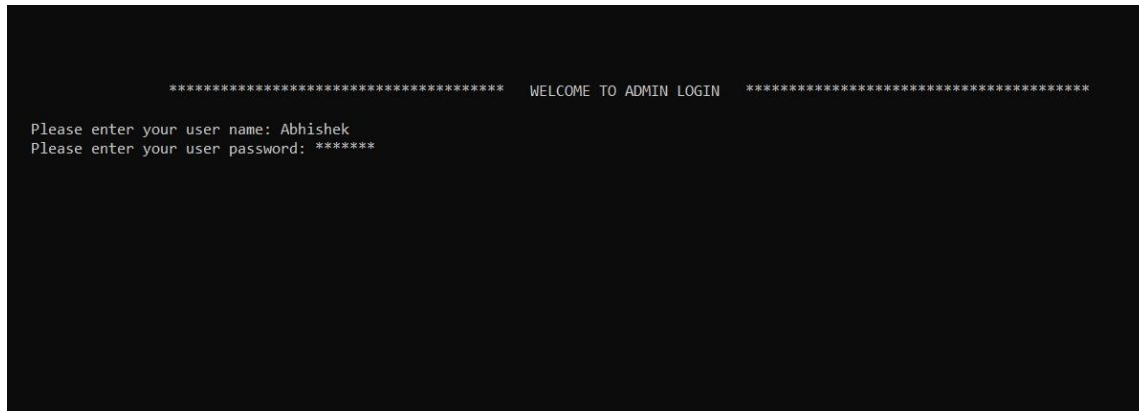


Fig 7.5: Admin login

This is the admin section after login. Two options are available one is for gamehouse stock management and second is for gamehouse employee management.

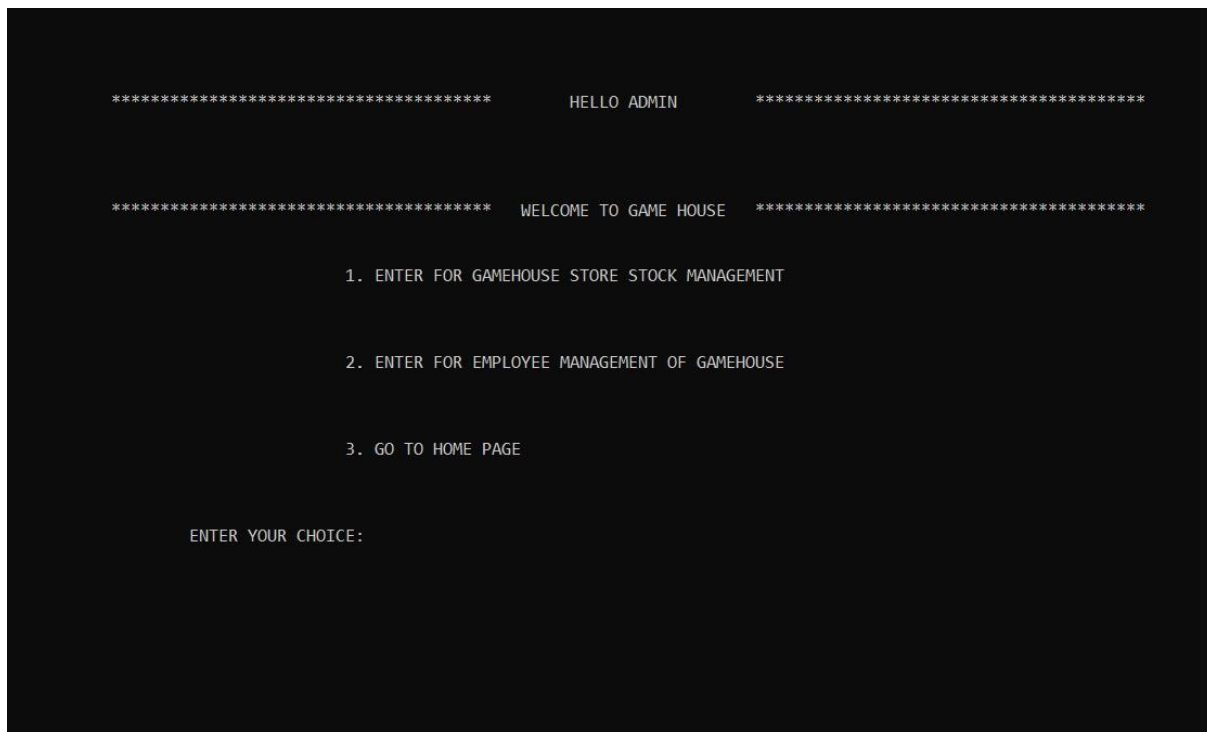


Fig 7.6: Admin section

This the gamehouse stock management section for admin

```
*****WELCOME TO GAMEHOUSE*****

1. ADD GAME
2. DISPLAY GAMES AVAILABLE
3. CHECK FOR PARTICULAR GAME
4. UPDATE GAME LIST
5. DELETE GAME FROM LIST
6. BACK
7. GO TO HOME PAGE
8. EXIT

Enter your choice :
```

Fig 7.7 : Gamehouse stock management menu

This is the add game option for admin

```
ADD GAME

GAME ID : #1116
GAME NAME : GOD_OF_WAR
GAME DEVELOPER NAME : SANTA_MONICA_STUDIOS
NO. OF COPIES OF THIS GAME : 32
Want to add another game? (y/n) :
```

Fig 7.8 : Add game section for admin

This is the display game list section for admin

```
THE GAMES AVAILABLE

GAME ID : #1102
GAME NAME : DYING_LIGHT
GAME DEVELOPER NAME : TECHLAND
No. of copies of GAME : 15

GAME ID : #1103
GAME NAME : METRO_EXODUS
GAME DEVELOPER NAME : 4A_GAMES
No. of copies of GAME : 30

GAME ID : #1104
GAME NAME : RESIDENT_EVIL_BIOHAZARD
GAME DEVELOPER NAME : CAPCOM
No. of copies of GAME : 10

GAME ID : #1105
GAME NAME : NEED_FOR_SPEED
GAME DEVELOPER NAME : EA_SPORTS
No. of copies of GAME : 35

GAME ID : #1106
GAME NAME : FIFA_2021
GAME DEVELOPER NAME : EA_SPORTS
No. of copies of GAME : 25

GAME ID : #1107
GAME NAME : BATTLEFIELD
GAME DEVELOPER NAME : EA_GAMES
No. of copies of GAME : 12
```

Fig 7.9: Display gamelist for admin

This is the search screen for particular game detail for admin

```
THE GAME DETAILS PROVIDED BELOW

GAME ID : #1116
GAME NAME : GOD_OF_WAR
GAME DEVELOPER NAME : SANTA_MONICA_STUDIOS
No. of copies of GAME : 32

Press any key to continue . . .
```

Fig 7.10 : Search game detail for admin

This is for updating any game details by the admin.

```
UPDATE GAME LIST

GAME ID : #1105

UPDATED GAME LIST

New GAME Name : PUBG
DEVELOPER Name : KRAFTON
No. of copies of GAME : 20
Press any key to continue . . .
```

Fig 7.11 : Update game details

This is the delete game page by admin

```
Delete a Game

GAME ID : #1107

Delete a GAME

GAME is Deleted Successfully...
Press any key to continue . . .
```

Fig 7.12 : Delete game from file

This is the employee management section for which is handled by admin.

```
*****GAMEHOUSE EMPLOYEE SECTION*****

1. ADD EMPLOYEE
2. EMPLOYEE WORKING IN GAMEHOUSE
3. CHECK FOR PARTICULAR EMPLOYEE DETAILS
4. UPDATE EMPLOYEE LIST
5. DELETE EMPLOYEE FROM LIST
6. BACK
7. GO TO HOME PAGE
8. EXIT

Enter your choice :
```

Fig 7.13 : Gamehouse employee management for admin

This is the add employee details by admin

```
ADD EMPLOYEE

EMPLOYEE ID : 110011
EMPLOYEE NAME : Abhishek
EMPLOYEE ADDRESS : Siliguri
EMPLOYEE PHONE NO. : 7788994455
EMPLOYEE GENDER : M
EMPLOYEE AGE : 21

Want to add another employee details? (y/n) :
```

Fig 7.14 : Add employee details

This is the display employee details page for admin

```
THE EMPLOYEE WORKING IN GAMESHOP

EMPLOYEE ID : 1101
EMPLOYEE NAME : abhishek
EMPLOYEE ADDRESS : slg
EMPLOYEE CONTACT NO. : 9679631401
GENDER : M
AGE : 21

EMPLOYEE ID : 1102
EMPLOYEE NAME : abhinav
EMPLOYEE ADDRESS : ranchi
EMPLOYEE CONTACT NO. : 789461231
GENDER : M
AGE : 22

EMPLOYEE ID : 1103
EMPLOYEE NAME : wasim
EMPLOYEE ADDRESS : pune
EMPLOYEE CONTACT NO. : 123456789
GENDER : M
AGE : 30

EMPLOYEE ID : 1104
EMPLOYEE NAME : narin
EMPLOYEE ADDRESS : sweden
EMPLOYEE CONTACT NO. : 1236547899
GENDER : F
AGE : 27

EMPLOYEE ID : 1111
EMPLOYEE NAME : s
EMPLOYEE ADDRESS : s
EMPLOYEE CONTACT NO. : s
GENDER : s
AGE : 1
```

Fig 7.15 : Display employee details

This is search employee details by the admin

```
THE EMPLOYEE DETAILS PROVIDED BELOW

EMPLOYEE ID : 110011
EMPLOYEE NAME : Abhishek
EMPLOYEE ADDRESS : Siliguri
EMPLOYEE CONTACT NO. : 7788994455
GENDER : M
AGE : 21

Press any key to continue . . .
```

Fig 7.16 : Search employee details

This is the update employee details section by the admin

```
UPDATE EMPLOYEE LIST
EMPLOYEE ID : 110011

UPDATE EMPLOYEE LIST
NEW EMPLOYEE NAME : Rahul
EMPLOYEE ADDRESS : Bangalore
CONTACT NO. : 4455669875
GENDER : M
AGE :50
Press any key to continue . . .
```

Fig 7.17 : Update employee details

This is delete employee details by admin

```
Delete details of EMPLOYEE
EMPLOYEE ID : 1102

Delete Employee Details
Employee details is Deleted Successfully...
Press any key to continue . . .
```

Fig 7.18 : Delete employee details

CONCLUSIONS AND FUTURE ENHANCEMENT

Conclusions

The Game House shop application is used to overcome the entire problem which the shop owners are facing currently, and making the manual system to a computerized system. This application helps in processing the data easily. The customer can easily come and search for any game using the application, hence the time for manual searching in store is saved. Also the owner can keep a track record of all the stock of the games and also the data of the employees working in the shop. All these data can be accessed easily and is easier to handle. The files are created in the memory and can easily be accessed. Hence this application helps in maintaining and retrieving data easily.

Limitation

As the number of files and the data in the file increases, comparison processes also increase, hence maintaining the large files and many files at a time becomes very difficult as we cannot read. The user can search the details just using game id.

Future Enhancements

Given the File Structure design that we have built, we foresee a lot of areas that need enhancements.

1. First of all search the details using game name or the game developer is a major void that needs to be filled. Any database would require such facilities of record modifications and deletions.
2. Voice search feature can also be added in the future where the user can search any detail by just saying any key word like game name or the employee name to search for the data or display the data.

BIBLIOGRAPHY

Book references

1. File Structures An Object-Oriented Approach with C++ by Michael J.Folk, Bill Zoellick, Greg Riccardi
2. Software Engineering-by Ian Sommerville
3. Let Us C++ by Yashavant .P.Kanetkar

Web references

1. www.stanford.edu
2. www.awl.com
3. www.codeproject.com
4. www.sourceforge.com
5. www.w3cschools.com