**Informatics Institute of Technology**

Informatics Institute of Technology

Department of Computing

Module: ECSI410 Software Development Principles 01

Module Leader – Mr. Guhanathan Poravi / Ms. Dinithi Jayaratne

Coursework

Date of submission: 23rd of March 2015 12:59 pm

* Student ID - 2014217
* Student First Name - Isuru
* Student Last Name - Siriwardena
* Student ID - 2014206
* Student First Name - Manith
* Student Last Name - Hasthaka

**Table of content**

* + List of Figures and List of Tables ----------- 3
  + Introduction ------------------------------------- 4
  + Analysis ----------------------------------------- 4
  + Design ------------------------------------------- 5
  + Implementation -------------------------------- 10
  + Screenshots ------------------------------------- 14
  + Evaluation (Testing) -------------------------- 17
  + Conclusion ------------------------------------- 20

**List of Figures and List of Tables**

**List of Figures**

|  |  |
| --- | --- |
| Figure Number | Description |
| Figure 1 | Functional and non-Functional requirements |
| Figure 2 | Flow chart for adding CDs to the database |
| Figure 3 | Flow chart for CD renting |
| Figure 4 | Flow chart for displaying the rented CD list |
| Figure 5 | Flow chart for displaying the reminder list |

**List of Tables**

|  |  |
| --- | --- |
| Table Number | Description |
| Table 1 | Data types |
| Table 2 | Adding CDs to the database |
| Table 3 | CD renting information |
| Table 4 | List of rented CDs |
| Table 5 | White box Testing |

**Introduction**

CD Universe, this software is developed for monitoring and controlling the routine in a CD shop. This software is a very user friendly software, user or admin able to add new customers, add customer’s details, add new CDs to list, searching CDs, removing CDs from listed. And the special feature in our software called CD Universe is user can get notifications about non-returned CDs after due date with customers details.

Our software is easy to use for both beginners and advanced users. It features a familiar and well thought-out, an attractive interface, enriched with fine searching insertion and reporting features. Owner of the CD shop can get lots of benefits via using this software rather than maintaining handmade reports. Customer can save their time, energy, and money by using this software.

**Analysis**

Figure 1

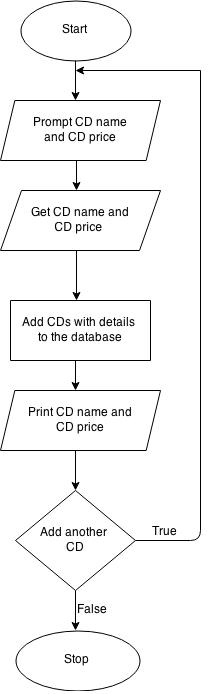
|  |  |
| --- | --- |
| Functional | Non Functional |
| System secured with user name and password. | User friendliness. |
| Unlimited CDs can enter to the system. | Efficiency. |
| Display the entered CDs. | Effectiveness. |
| Can delete the entered CDs from the system. | Maintainability. |
| Can rent the CDs to the customers through the system. | Reliability. |
| Display the rented CDs. | Security. |
| Can return CDs through the system. | Quality. |
| Can search the entered CDs to the system. | Supportability. |
| Can edit the CD details. | Relevant. |
| Can check the due dates. | Accuracy. |

**Design**

* Flow chart for each functionality identified

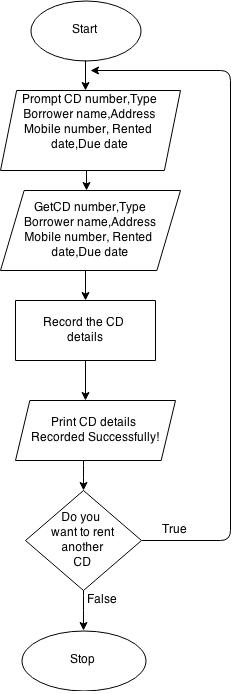
1. Flow chart for adding CDs to the database

Figure 2



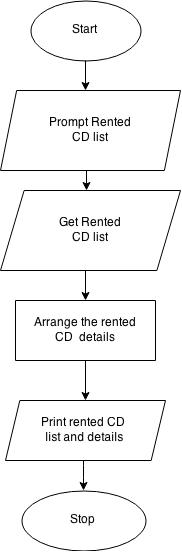
1. Flow chart for CD renting

Figure 3

****

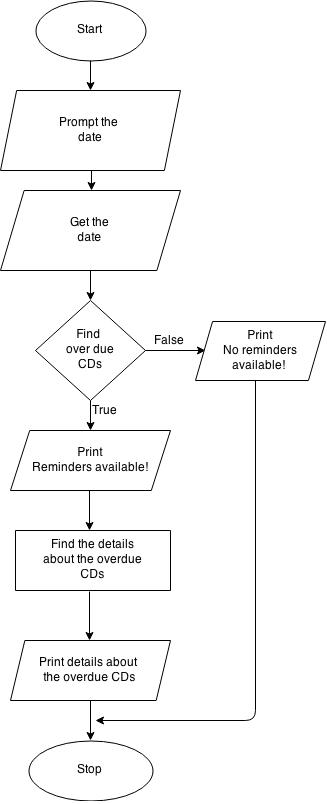
1. Flow chart for displaying the rented CD list

Figure 4

****

1. Flow chart for displaying the reminder list

Figure 5

****

* Algorithm for each functionality identified

1. **Add CD details to the database**

Start

Prompt user’s choice

Get user’s choice

Prompt Enter CD name, Enter CD price

Get CD name, CD price

Print CD name, CD price

If user want to add another CD

Do while user’s choice equals no

End.

1. **New CD rental**

Prompt user’s choice

Get user’s choice

Prompt CD number, Type, Borrower name, Address, Mobile number,

Rented date, Due date

Get CD number, Type, Borrower name, Address, Mobile number,

Rented date, Due date

Add CDs to the database

Print CD recorded successfully

If user want rent another CD

Do while user’s choice equals no

End

1. **Show Rented CD details**

Prompt Rented CD list

Get Rented CD list

Arrange the rented CD list

Print Rented CD list and details

End

1. **Display the reminder List**

Prompt the date

Get the date

If overdue CDs in the database

Print Reminders available

Find the details about the overdue CDs

Print details about the overdue CDs

Else

Print No Reminders Available

End

**Implementation**

1. Adding CD details to the database

**public** **static** **void** cdDetails() { // this method is about the adding CDs into the menu

// the data base

String confirm ="yes";

System.***out***.println("");

System.***out***.print("\t\t +-+-+-+ +-+-+-+ +-+-+ +-+-+-+ +-+-+-+-+-+-+-+" + "\n"+

"\t\t |A|d|d| |C|D|s| |t|o| |t|h|e| |d|a|t|a|b|s|e|"+"\n"+

"\t\t +-+-+-+ +-+-+-+ +-+-+ +-+-+-+ +-+-+-+-+-+-+-+" +"\n");

**while** (confirm.equalsIgnoreCase("yes")){

**int** itemPrice;

System.***out***.print("Enter CD name : ");

*sc*.nextLine();

*itemName* = *sc*.nextLine();

System.***out***.print("Enter CD price(Rs): ");

**do**{

**if**(!*sc*.hasNextInt()){

System.***err***.print("Enter an Integer!\n");

System.***out***.print("Enter CD price(Rs) : ");

*sc*.next();}

}**while**(!*sc*.hasNextInt());

itemPrice = *sc*.nextInt();

System.***out***.println("CD Number \t CD Name \t CD price ");

System.***out***.println("");

System.***out***.println(*x* + ")\t\t " + *itemName* + "\t\t " + itemPrice);

*cdList*.add(*x* + "\t" + *itemName*);

*x*++;

System.***out***.println("CD \"" + *itemName* + "\" added succesfully to the database!");

**do** {

System.***out***.println("Do you want to add a new CD to the system?(Yes or No)");

confirm = *sc*.next();

**if** (!(confirm.equalsIgnoreCase("yes") || confirm.equalsIgnoreCase("no")))

System.***out***.println("Invalid input. Enter (Yes/No)");

} **while** (!(confirm.equalsIgnoreCase("yes") ||confirm.equalsIgnoreCase("no")));

}

}

1. CD renting information

**public** **static** **void** rentCd() { // This method about rent a CD from the database

**int** y = 0;

System.***out***.println(" ");

System.***out***.println("\t\t +-+-+ +-+-+-+-+-+-+ +-+-+-+-+-+-+-+-+-+-+-+"+"\n"+

"\t\t |C|D| |R|e|n|t|a|l| |I|n|f|o|r|m|a|t|i|o|n|"+"\n"+

"\t\t +-+-+ +-+-+-+-+-+-+ +-+-+-+-+-+-+-+-+-+-+-+");

System.***out***.println("");

System.***out***.print("Enter the CD number: ");

**int** itemNumber = *sc*.nextInt();

*cdList*.get(itemNumber);

System.***out***.print("Enter item type (weekly/overnight): ");

String itemType = *sc*.next();

System.***out***.print("Enter borrower name: ");

*sc*.nextLine();

String borrowerName = *sc*.nextLine();

System.***out***.print("Enter borrower address: ");

String borrowerAddress = *sc*.nextLine();

System.***out***.print("Enter borrower mobile Number: ");

String borrowerNumber = *sc*.next();

System.***out***.print("Rented date(dd/mm/yyyy): ");

String rentedDate = *sc*.next();

System.***out***.print("Due date(dd/mm/yyyy): ");

String dueDate = *sc*.next();

**int** rentNumber = y;

*rentList*.add(itemNumber + ";" + dueDate + ";" + itemType + ";" + borrowerName + ";" + borrowerAddress + " ;"

+ borrowerNumber +";"+ rentedDate+ ";" + rentNumber +";");

y++;

System.***out***.println("Your cd rental details was recorded successfully!");

String confirm;

**do** {

System.***out***.println("Do you want to rent a new cd? (Yes/No)");

confirm = *sc*.next();

**if** (!(confirm.equalsIgnoreCase("yes") ||confirm.equalsIgnoreCase("no")))

System.***out***.println("Invalid input. Enter (Yes/No)");

} **while** (!(confirm.equalsIgnoreCase("yes") ||confirm.equalsIgnoreCase("no")));

}

1. Edit the CD details

**public** **static** **void** editCd(){ //this method about editing a CDs in the database

System.***out***.println("");

System.***out***.println("\t\t +-+-+-+-+ +-+-+ +-+-+-+-+-+-+-+"+"\n"+

"\t\t |E|d|i|t| |C|D| |D|e|t|a|i|l|s|"+"\n"+

"\t\t +-+-+-+-+ +-+-+ +-+-+-+-+-+-+-+");

System.***out***.println("");

System.***out***.print("Enter the CD Name that you want to Edit : ");

String cdName = *sc*.next();

**for** (**int** i=0;i<*cdList*.size();i++){

String [] arr= *cdList*.get(i).split("\t");

**if**(arr[1].equals(cdName)){

System.***out***.print("Enter new CD Name : ");

String cdNameNew = *sc*.next();

*cdList*.set(i, arr[0]+"\t"+cdNameNew);

System.***out***.println("Updated CD details!");

**break**;

}

System.***out***.println("No CD Found with that number!");

}

}

1. Delete a CD from the list

**public** **static** **void** deleteCd() { // This method is about deleting CD from the

// database

String confirmation = "yes";

System.***out***.println("");

System.***out***.println("\t\t +-+-+-+-+-+-+ +-+ +-+-+-+-+-+ +-+-+-+-+ +-+-+-+ +-+-+-+-+-+-+-+-+"+"\n"+

"\t\t |D|e|l|e|t|e| |a| |M|o|v|i|e| |f|r|o|m| |t|h|e| |d|a|t|a|b|a|s|e|"+"\n"+

"\t\t +-+-+-+-+-+-+ +-+ +-+-+-+-+-+ +-+-+-+-+ +-+-+-+ +-+-+-+-+-+-+-+-+");

System.***out***.println("");

**while** (confirmation.equalsIgnoreCase("yes")){

System.***out***.print("Enter the CD Name that you want to delete : ");

String cdName = *sc*.next();

**for** (**int** i=0;i<*cdList*.size();i++){

String [] arr= *cdList*.get(i).split("\t");

**if**(arr[1].equals(cdName)){

*cdList*.remove(i);

System.***out***.println("CD \"" + cdName + "\" Deleted successfully!");

**break**;

}**else**{

System.***out***.println("No CD found");

**break**;

}

}

System.***out***.println("Do you want to delete another CD from the Database?(Yes or No)");

confirmation = *sc*.next();

}

}

1. Reminder for overdue items

**public** **static** **void** dueDates(){ //This method about show a reminder about rented CDs in the database

**int** count=0;

System.***out***.println(" ");

System.***out***.println("\t\t +-+-+-+-+-+-+-+-+-+"+"\n"+

"\t\t |R|e|m|i|n|d|e|r|s|" + '\n'+

"\t\t +-+-+-+-+-+-+-+-+-+");

System.***out***.println("");

System.***out***.print("Enter the date in dd/mm/yyyy format: ");

String number = *sc*.next();

**for**(**int** i=0;i<*rentList*.size();i++){

String arr []=*rentList*.get(i).split(";");

String date=arr[1];

SimpleDateFormat f = **new** SimpleDateFormat("dd/MM/yyyy");

f.setLenient(**false**);

Date parse = **null**;

Date user=**new** Date();

**try** {

parse = f.parse(date);

user=f.parse(number);

} **catch** (ParseException e) { }

Calendar cal = Calendar.*getInstance*();

cal.setTime(user);

cal.add(Calendar.***DATE***, 7);

**int** diffInDays = (**int**)( Math.*abs*(cal.getTime().getTime() - parse.getTime())

/ (1000 \* 60 \* 24\*60 ) );

**if**(diffInDays>7){

count++;

System.***out***.println("Item number : "+arr[0]+ "\n" + "Rented Date: "+arr[6] +"\n" +"Due Date: "+arr[1] +"\n" +"Item Type: "+arr[2] +"\n"+"Borrower Name: "+arr[3] +

"\n" +"Borrower Address: "+arr[4] +"\n" + "Borrower mobile number: "+arr[5]);

}

}

**if**(count>0){

System.***out***.println(count+" Reminders Available");

}

**else**{

System.***out***.println("NO Reminders Available! ");

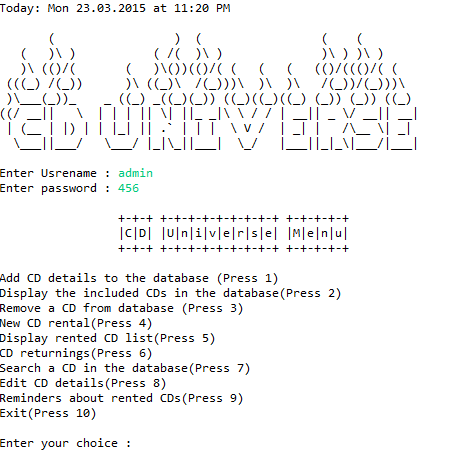
}

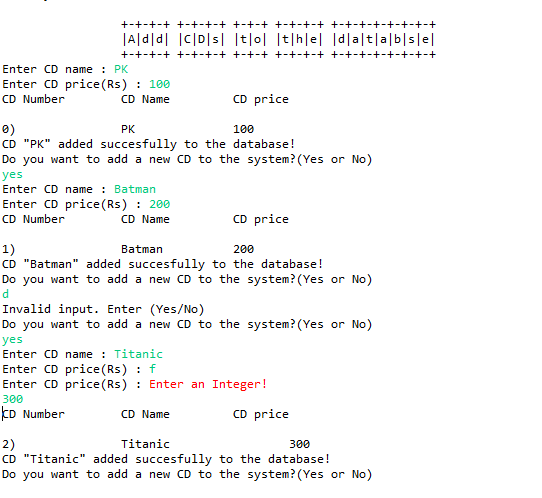
}

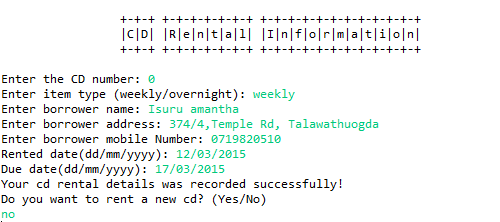
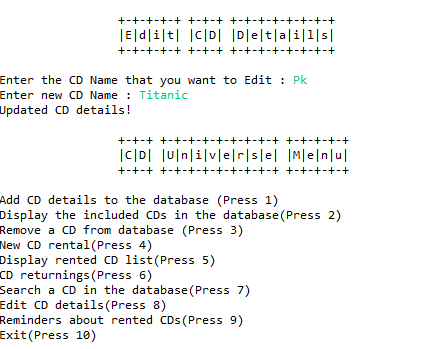
1. Other operation in this system –

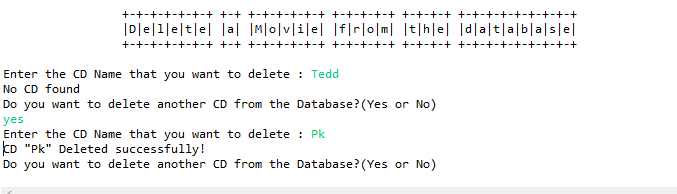
* User name and password security system – This CD system is secured with a predefined user name and password. Therefore no one can’t access to the system if they don’t know the password and the username.
* Menu list – This operation is useful for the user. This menu will guide the user to the correct path. If the user want to do anything with the system, all the essential information are provided in the menu list.
* Display the included CDs – This operation helps to find the index number or any other details in this system quickly. This operation display the whole list of CDs.
* Display rented CD list – This operation display the rented CD list quickly. Therefore you can find whether the CDs rented or available very easily.
* Exit function – If this operation is executed the system will terminated.

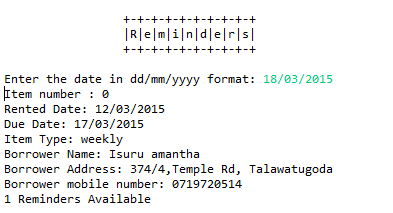
**Screen Shots**









****

**Data Types**

|  |  |  |
| --- | --- | --- |
| **Variables** | **Data type** | **Purpose** |
| * cdList * rentList * x * confirmation * itemName * itemPrice | * Array List * Array List * Int * String * String * String | * To maintain a list of CDs   Information(CD name, CD number)   * To maintain a list of CDs information(rented CDs name, CD number, rented date, due date, borrower’s details) * Assigning the size of cdList * To check whether user’s choice equals to “yes” * To assign a name for a CD * To assign a value for a CD |

Table 1

**Testing**

**Black Box Testing**

1. Adding CD details to the database (Table 2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial number | Input | Expected Output | Actual Output | Bug |
| 1 | CD name – PK  CD price(Rs) - 100 | CD PK added successfully to the database | CD PK added successfully to the database | no |
| 2 | CD name – PK  CD price(Rs) – w | Enter an integer | Enter an integer | no |

1. CD renting information (Table 3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial number | Input | Expected Output | Actual Output | Bug |
| 1 | CD number – 1  Item type – weekly  Borrower name – Isuru  Borrower address – 47, Temple Rd  Borrower mobile number – 07189898  Rented date – 12/12/2012  Due date – 19/12/2012 | Your cd rental details was recorded successfully! | Your cd rental details was recorded successfully! | no |

1. List of rented CDs (Table 4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial number | Input | Expected Output | Actual Output | Bug |
| 1 | User choice | Item number  Due date  Item type  Borrower name  Borrower Address  Borrower mobile number  Rented date | Item number  Due date  Item type  Borrower name  Borrower Address  Borrower mobile number  Rented date | no |

**White box testing (**Table 5)

|  |  |  |
| --- | --- | --- |
| Test no | Condition | Path |
| 1 | If user name and password entered by the user | True – Display the menu list  False – invalid user name and password |
| 2 | Add CD details to the database – CD price | True – Add CD price to the database  False – invalid integer |
| 3 | Remove CD from the database | True – remove particular CD from the database  False – CD not found |
| 4 | New CD rental – CD number | True – Process the rest of the code  False – invalid CD number |
| 5 | CD returning - CD number | True – return the CD  False – invalid CD number |
| 6 | Edit CD details – CD name | True – Edit CD details  False – invalid CD name |
| 7 | Reminders - Date | True – Display the reminders  False – No reminders available |

**Conclusion**

Making this program helped both of us to add many experiences to our lives. And also it helped us to blush up and widen our knowledge in coding, flow charts and algorithms etc. The most valuable thing that we achieved from this project is how to face to tough situations and how to work as a team. From project we learnt to how to manage time and how we should pre plain our duty.

But still there is something which we weren’t able to achieve. But we think this project will be a great privilege ourselves to reduce those weak points too. And we would like to thank our module leader Mr. Guhanathan Poravi for asked us to do this project as our Course Work 01 to improve our knowledge and blush up our skills.