



NALANDA COLLEGE

**Constituent Unit of Magadh University**

**BIHARSHARIF, NALANDA-803101**

**PROJECT REPORT**

**ON**

**DAIRY BILLING**

***SUBMITTED BY***

### JAINENDRA KUMAR

**REG NO*. :- 1726300924/2017***

**ROLL NO. *:- 2030721846***

**DATE OF SUMBITION *( 11/12/2020)***



***UNDER THE GUIDENCE OF***

***PARMANAND PRASAD***

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR**

**QUALIFINIG BACHELOR DEGREE OF COMPUTER APPLICATION.**



**CERTIFICATE**

# This is to certify that ~~miss/mrs./~~ mr. JAINENDRA KUMAR ~~daughter/ wife~~/son of ASHOK KUMAR having College Roll No.*12* University Roll No. NAL-2030721846 has completed the project work for BCA Part – III (Paper – VIII) for the Academic session 2017-2020.

His/her conduct has been good.

***Signature\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***Co-ordinator Signature Of External Examinar***

***BCA Department***

***Nalanda College, Biharsharif***



|  |  |
| --- | --- |
| **Name** | Parmanand Prasad |
| **DOB** | 20th MAY,1963 |
| **Address** | Chaudhary Colony,  Naisarai, Bihar Sharif,  Nalanda(803101) |
| **Sex** | Male |
| **Status** | Married |
| **Educational Qualification** | 1. MCA, M.Sc. (MATH) 2. Total Information System   Management From LCC   1. B.Ed. From Gawhati.Univ |
| **Experience** | 1. Faculty in LCC, Deoria(U.P) 2. Faculty in LCC, Bihar Sharif 3. Faculty in Nalanda College BCA   &MCADEPT (MU) |
| **Hobbies** | Programming in C++,java,oracle |

### UNDER THE GUIDENCE OF

### *PARMANAND PRASAD*

***NALANDA COLLEGE, BIHARSHARIF (M. U.)***

**BIO-DATA OF GUIDE FACULTY**

|  |  |  |
| --- | --- | --- |
| *Slno* | *Description* | *Page no* |
| 01 | **Acknowledgement** | **05** |
| 02 | **Introduction** | **7-8** |
| 03 | **Project CAtegory** | **09** |
| 04 | **System Analysis** | **10-14** |
| 05 | **Feasibility Study** | **15** |
| 06 | **Design** | **17-25** |
| 07 | **Module Design and Process Logic */* Flow Chart** | **25-31** |
| 08 | **Coding** | **32-158** |
| 09 | **Testing Approach** | **158-161** |
| 10 | **Software Maintainance** | **161-163** |
| 11 | **Limitation of the Project** | **163** |
| 12 | **Future Scope Of The Project** | **163-164** |
| 13 | **Bibliography** | **164** |

*INDEX*



ACKNOWLEADGEMENT

I have the honor submitting project report on  **(DAIRY BILLING)** for the purpose of successful completion of my Bachelor Degree in Computer Application (B.C.A.) from Magadh University, Bodh Gaya.

AT FIRST, I thank to my respectable guider Sri Parmanand Prasad (Faculty BCA M.U.) Nalanda College Bihar Sharif, who gave me valuable time, and guidance in preparing the project report.

I express my gratitude to my parents, brothers, sisters and friends; without whose co-operation my work would not have been completed.

I am very thankful to Parmanand Prasad, which help me in the preparation and completion of this project.

***Sincerely***,

Name:- ***JAINENDRA KUMAR***

Roll No:-NAL- ***2030721846***

Reg.No.:- ***1726300924-17***

BCA-III (Hons.)

Session:-2017-2020

***Title***

***Dairy billing system***

INTRODUCTION

**NALANDA DAIRY FARM**

Nalnada Dairy Farm is engaged in the processing of milk and milk…..Dairy goods is sold in the market under the brand name of ‘Nalanda’ and Ice cream under the Nalanda Dairy Farm deals with sale of the Dealing in production and marketing of Milk and Milk products since 2008. It produces all types of milk product came into existences in the JANUARY, 2008. It is situated in the heart of the city of Bihar Sharif at Nala Road, Bihar Sharif in Bihar. Since its establishment the organization is expanding its horizon each year. With a major stock of milk when established organization has expanded manifold since then and has reached a stock limit of over five times the no. of milk products including go down. When started, the organization had a small no. of employees which now has stretched to nearly100 in order to ensure lubricate the work of shows the managerial skill of the organization and helps put its name shinning in the stardom of business

**OBJECTIVE**

After just few party registered in the organization After just few years, when it came in existence, the Anuj dairy, Home Patna has attend maturity with expert manager, staffs, worker well equipped place etc. now a day, it plays a pivotal in the production of milk in Patna districts.

Health is most important all living creature. For human beings, it is the key of success. Of course, one can never compromise with it. The Raj diary, Patna has made it easy with latest technology. Here, machine, product and even complete production area are in neat and clean environment.

Since there are a large number of parties and distributers have faith at this industry, therefore, it is not specially, manipulation and maintenance of records are not easy. So it was requirement to develop an automates system for it to assist the organization to work easily and efficiently. Automated system provides electronic speed and accuracy. And so, it takes less time for manipulation and maintenance of records of all the party and distributor manually.

My job is to develop such type of automated system. The primary work of this organization is to cure party and distributer. For this, the organization manual registers, such as

1. **Product details:** This is to store the details of Product which are unavailable in the stock.
2. **Party detail:** This is to store the details of Party and Its Information working in the industry.
3. **Distributer Detail:** This table is used to store the information about Distributor who are registered in the organization to work in industry.
4. **Retailer Detail:** This Table is used to store the details of the Retailers for the works in the organization.
5. **Sale Detail:** This is to store the details of sale to be entered to organization.
6. **Party Registration Detail:** This register is maintained for storing information about the Distributor Registration details in the industry their area.
7. **Transaction Details:** This register is used to store the details of transaction of organization. This register is used to store the details of transaction.

Apart from these register the industry has to publish weekly reports about the organization

|  |  |
| --- | --- |
| * Party name * Party address * Party phone no * Party\_id | * Party state * Party email * Party fax * Party dob |

**PROJECT CATEGORY: OBJECT ORIENTED PROGRAMMING STRUCTURE**

**OBJECT ORIENTED PROGRAMMING CONECEPT:** object-oriented Programming (oop) is an approach to program organization and development, which attempts to eliminate some of the pitfalls of conventional programming methods by incorporating the best of structured programming features with several new concepts. It is a new way of organizing and developing programs and has nothing to do with any particular language. However, not all languages are suitable to implement the OOP concepts easily. The latest one added to this list is Java, a pure object-oriented language. The general concepts of OOP which form the heart of C++ Language.

**OBJECTS AND CLASSES: -** Objects are the basic runtime entities in an object-oriented system. In this project they may be news information, party detail, advertisement order monitoring by employee that the programs handle while coding. Objects should be chosen such that they match closely with the real-world objects. When a program is executed, the objects interact by sending massages to one another.

**DATA ABSTRACTION AND ENCAPSULATION: -** Data encapsulation is the most striking feature of a class. The wrapping up of data and methods into a single unit (called class) is known as encapsulation.

**INHERITANCE: -** Inheritance is a process by which objects of one class acquire the properties of objects of another class. Inheritancesupports the concepts of hierarchical classification. In OOP, the concept of inheritance provides the idea of reusability.

**POLYMORPHISM: -** Polymorphism is another important property of OOP. Polymorphism means the ability to take more than one form.

**OBJECT-ORIENTED PARADIGM: -** The features of object-oriented are:-

* + Emphasis is on data rather than procedure.
  + Programs are divided in such a ways and that are known as Objects.
  + Data structures are designed such that they characterize the objects.
  + Methods that operate on the date of an on object are tied together in the data structure.

Data is hidden and cannot be accessed by external function.

* **System Study**

In this stage, studding the current to find out how it works and where improvement should be made. System studies results in evaluation of how current methods are working and where adjustment is necessary and impossible.

* **Existing System**

The existing system for dairy is not computerized it is handled manually. It is very time consuming & need no. of people to do the jobs of the system. The existing System has many problems like; users have to do lots of paper work for data entry, report generation. The retrieval of data from the register is very tough task and every calculation on that data in not very simple as manual calculation is very slow in comparison of computerized system. Here in this case to many times consume, even a single mistake existing system not reliable for the organization. Hence it requires a perfect computerized system.

* **System Analysis**

In this phase, software developers carry out a detailed study of user’s requirements. They then arrive at the proposed system to be built. This phase generates fictional specifications, which contain:

* Output to be produced.
* Inputs that need to be received.
* Procedure that will get the output from the input.
* Adult and control the requirements that the user can use to monitor the proper functioning of the system.
* Retrieving the data, calculation on the data and the report generation will take less time.
* Acceptance test to list the tests that user can carry out to ensure that the system is acceptable.
* **Scope of this document**

Describe the scope of the requirement definition effort. Introduces the requirements elicitation team, including users, customers, system engineers, and developers.

This section also details any constraints that were placed upon the requirements elicitation process, such as schedules, costs, or the software engineering used to develop requirements.

**BUSINESS CONTEXT**

Provide an overview of the Hospitality. This overview should include the business’s mission statement and its organizational objectives or goals. This section describes the set of objective and requirement for the system from the user’s perspective. It may include a “wish list” of desirable characteristics’, along with feasible solutions that are in line with the business objectives.

**General constraints**

Lists general constraints placed upon the design team, including speed requirements, industry protocols, hardware platforms, and so forth.

***Requirement analysis***

This section lists the functional requirements in ranked order. Functional requirements describe the possible effects of a software system, in other words, what the system must accomplish. Other kind of requirements (such as interface requirement, performance requirements, and reliability requirements) describes how the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

* Short imperative sentence stating highest ranked Functional Requirement.
* **Description:** A full description of the requirement.
* **Criticality:** Describes how essential this requirement is to the overall system.
* **Technical-Issues:** Describes any design or implementation issues involved in satisfying this requirement.
* **Cost-and-schedule:** describes the relative or absolute costs associated with this issue.
* **Risks:** describes the circumstances under which this requirement might not able to be satisfied, and what actions can be taken to reduce the probability of this occurrence.
* **Dependencies with other Requirements:** Describes interaction with other requirements.
* Others as appropriate

1. **Interface Requirements**

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include library routines, token streams, shared memory, data streams, and so forth.

* **User-interfaces:** describes how this product interfaces with the user.
* **GUI:** Describes the graphical user interfaces if present. This section should include a set of screen dumps or mockups to illustrate user interface features. If the system is menu-driven, a description of all menus and their components should a provided.

1. ***Design constraints***

Specifies any constraints for the design team using this document.

* Standards compliance
* Hardware Limitations
* Others as appropriate
* **OTHER NON-FUNCTION ATTRIBUTES**

Specifies any other particular non-functional attributes required by the system. Examples are provided below.

* Security
* Binary compatibility
* Reliability
* Maintainability
* Portability
* Extensibility

Requirement Specification

An information system is intended to meet the need of the organization. So this step is to specify the needs or requirements. The software developer to gather the information should evolve a clear strategy. The strategy consist of

* Identifying the system resources.
* Forms and documents used by concern.
* Various queries from the management and the user.
* Various report used in the organization.

The requirements of purposed system are:

* The system should generate different type of report to support the management.
* It is also accepted that information maintained should be secured and not accessible in an unauthorized manner.
* It is also accepted that proper manual should be provided including how to work with the system.
* System should be user friendly.

***System Analysis***

***Identification of need***

To identify the need for software we use the principle of Requirement Engineering. Requirement engineering provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirement as they are transformed into an operational system . the requirement engineering process can be described in five distinct steps:-

* **Requirement elicitation.**
* **Requirement analysis & negotiation.**
* **Requirement specification.**
* **System modeling.**
* **Requirement validation.**
* **Requirement management.**

In other words we can say that requirement analysis is a software task that bridges the gap between system level requirement engineering and software design. Requirement analysis allows the software engineering refine the software allocation and build models of the data, functional and behavioral domains that will be treated by software. Requirement analysis provides the software designer with a representation of information, function and behavior that can be translated into data, architectural, interface and component level design; finally the requirement specification provides the developer and the customer with the means to assess quality once software is built.

The most commonly used requirement elicitation technique is to conduct a meeting or interview. The first meeting between a software engineer and customer can be likened to the awkwardness of a first date between to adolescents. Neither person knows what to say or ask, both are worried that what they do say will be misinterpreted, both are thinking about where it might lead(Both likely have radically different exception here) Both want to get the think over with, but at the same time, both want it to be a success. Here according to this principle the analysts stats by asking context-free-question. That is set of question that will lead to a basic understanding of the problem, the people who want a solution, the nature of solution that desired ant the effectiveness of the encounter itself. The first set of Context-free question focuses on the customer, the overall goals, and the benefits. For example, the analyst might ask:-

* Who is behind the request for this work?
* Who will use solution?
* What will be economic benefit of a successful solution?
* Is there another source for the solution that you need?

The next set of the questions enables the software engineer to gain a better understanding of the problem and the customer to voice his or her perceptions about solution:-

* How would you characterize “good” output that would be generated by a successful solution?
* What problems(s) will this solution address?
* Can you show me (or describe) the environment in which the solution will be used?
* Will special performance issues or constraints affect the way of the solution is approached?

**The final set of question focuses on the effectiveness of the meeting:**

* Are you the right person to answer these questions? Are you answers? Official”?
* Are my questions relevant to the problem that you have?
* Am I asking too many questions?
* Can anyone else provide additional information?
* Should I be asking you anything else?

***Preliminary Investigation***

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the System. The purpose of the preliminary Investigation is to evaluate project requests. It is neither a design study nor does it includes the collection of details top describe the business system in all respect. Rather, it is collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.

Here for the **“Dairy billing System”,** I have worked on the preliminary investigation that accomplished the following objectives:

* Clarify and understand the project request.
* Determine the size of the project.
* Assess costs and benefits of alternative approaches.
* Determine the technical and operational feasibility of alternative approaches.
* Report the finding to management with recommendation outlining the acceptances or rejection of the proposal.

Considering above criteria I also keep in mind that the requirements are clearly understandable when the clarification of project request is enquired.

The data of **Dairy Billing System,** which are collected by me during preliminary investigation, are:-

* Reviewing organization Document
* Onsite observation and
* Conducting interviews

***Feasibility Study***

Not everything imaginable is feasible, not even in software, evanescent as it may appear to outsiders. Feasibility is the determination of whether or not a project is worth doing. On the contrary, software feasibility has seven solid dimensions are below:-

1. **Technical feasibility.**
2. **Operational feasibility.**
3. **Economic feasibility.**
4. **Social feasibility.**
5. **Management feasibility.**
6. **Legal feasibility.**
7. **Time feasibility.**

The process followed in making this determination is called a feasibility study. This type of study determines when project can or should be taken. Once it has been determined that the projects is feasible, and then analyst can go ahead and prepare the project specification, which finalizes project requirements. Generally, feasibility studies are undertaken with tight time constraints and normally culminate in a written and oral feasibility report. The contents and recommendation of such a study will be used as a sound basis for deciding whether to proceed, postpone or cancel the project. Thus, since the feasibility study may lead to the commitment of large resources, it becomes necessary that it should be conducted competently and that no fundamental errors of judgment are made.

In other words we can say that a feasibility study is conducted to select the best system that meets performance requirement. This contains and identification description, an evaluation of candidate system and the selection best system for the job. The system required a statement of constraints; the identification of specific system objective and a description of outputs define performance.

***Economical Feasibility***

Economical analysis is the most frequently used technique for evaluation of the effectiveness of a proposed system. More commonly known as Cost/Benefit analysis, the procedure is to determine the benefits and saving that are accepted from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. Other weight costs, a decision is taken to design and implement system. Otherwise, further justification or alternative in the proposed system will have to be made if is to have chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system lifecycle. Before establishing Economical Feasibility so as to justify the expenses that different organization working under this category has to bear for this Software. It is crystal clear that the user or customer of this Software has no need to invest heavy amount for this software. As my project has been written in such a language, which is platform independent. It has also the feature of WORM. It means posses the technology of write once & read many.

There is no any heavy expense on the maintenance of this Software. Also there is no need to expense a heavy Amount on the set of “Very Expert Personals” protection of **Secret password and User Name** unauthorized access of this software.

For different types of Process Operation, Report or Query and Output also, there is n need of heavy expenditure.

***Operational Feasibility***

My project is also fit in Operational Feasibility Study. As I have explained before this about Technical Feasibility and Economical Feasibility, this Software is very easy and user-friendly.

I have used a very easy menu system and also used some control buttons in the easy way by which one can choose the option on his desire. Though this is fully protected with the security by word (Password and user name), if both are matched only then one can get access of this software. Various Reports, Forms and Queries can be generated on the fingertips for the user. In nutshell we can say that it has following operational features:

1. **It is User-Friendly.**
2. **It is having less paperwork.**
3. **Efficient tractability.**
4. **Query can be generated.**
5. **Various reports and Forms can be generated.**
6. **Fully protected be Password and User Name for unauthorized access.**

***Software Engineering Paradigm***

**Liner Sequential Model** has been used in carrying out this project work. The linear Sequential Modal is the oldest and the most widely used paradigm for software Engineering. Linear Sequential Modal called sometimes the Classic Life Cycle of the waterfall Model.

The linear Sequential Model suggests a systematic, Sequential approach to software development that begins at the system level and progress through Analysis, Design, Coding, Testing and Support.

Linear Sequential Model contains the following activities:

***System / Information Engineering and Modeling:-***

Because software is always part of a larger system (or business), work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. This system view is essential when software must interact with other elements such as hardware, people and database. System engineering and analysis encompass requirements gathering at the system level with a small amount of top-level design and analysis. Information engineering encompasses requirements gathering at the strategic business and at the business area level.

***Software Requirements Analysis***

This requirements gathering process is intensified and focused specifically on software. To understand the nature of the program(s) to be built, the software engineer (“analyst”) must understand the information domain for the software, as well as required function, behavior, performance, and interface. Requirement for both the system and the software documented and reviewed with the customer.

***Design:-***

Software design is actually a multi step process that focuses on four distinct attributes of a program data structure, software architecture, interface representations, and procedural (algorithmic) detail. The design process translates requirements into a Representation of the software that can be accessed for quality before coding begins. Like requirements, the design is documented and becomes part of the software configuration.

***Code Generation:-***

The design must be translated into a machine-readable form. The code generation step performs this task. If design is performed in a detailed manner, code generation can be accomplished mechanistically.

***Testing:-***

Once code has been generated, program testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested and on he functional externals, that is conducting tests to uncover errors and ensure that defined input will product actual results that with required results.

Testing

Design

Coding

Analysis

**Fig:** The Linear Sequential Model

Requirement & Analysis Specification

Design

Coding

Integration

Delivery

**Figure:** Waterfall Model of Software a Life cycle

***System Design***

The system design process is not a step adherence of clear procedure and guidelines. Though, certain clear procedures and guidelines have emerged in recent days, but still much of design work depends on knowledge’s and experience of the designer.

When designer starts working on system design, he will face different type of problems. Many of these will be due to constraints imposed by the user of lamination f the hardware and software available in the market. Sometimes, it is difficult to enumerate complexity of the problems and solutions therefore since the variety of likely problems is so great and no solutions are exactly similar. However, following considerations should be kept in mind during the system designing phase.

***Design Objectives***

The primary objective of the design, of course, is to deliver the requirement as specified is the feasibility report. In general, the following design objectives should be kept in mind.

1. **Practicality:** The system must be stable and can be operated by people with average intelligence.
2. **Efficiency:** This involves accuracy, timelines and comprehensiveness of the system output.
3. **Cost:** It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all the requirements.
4. **Flexibility:** The system should be modifiable depending of the changing needs of the user. Such modifications should not entail extensive reconstructing or recreation of software. It should also be portable to different computer system.
5. **Security:** This is very important aspect of the design and should cover areas of hardware reliability, fall back procedures, physical security of data and provisions for detection of fraud and abuse. In actually System Design involves first logical design and then physical construction of the system. The logical design describes the structure and characteristics of features, like the outputs, inputs, files, databases and procedures. The physical constructions, which follows the logical design, produces actual program software, files and working system.

***Design Constraints***

The designer normally will work under following constraints:

1. **Hardware:** The existing hardware will work obviously affect the system design.
2. **Software:** The available software (Operating System, Utilities, Languages etc.) in the market will constrain the design?
3. **Budget:** The budget allocated for the project will affect the scope and depth of design.
4. **Time-scale:** The new system may be required by a particular time (e.g. the start of a financial year). This may put constraints on the designer to find the best design.
5. **Interface with other Systems:** The new system may require some data from another computerized system or may provide data to another system in which case the files must be compatible in the format and the system must operate with a certain processing cycle.

In nutshell we can say that system design develops the architectural details required to build a system of product the system design contains the following activities.

* Partition the analysis model into subsystems.
* Identify concurrency that is dictated by the problem.
* Allocate subsystems to processors and tasks.
* Develop a design for the user interface.
* Choose basic strategy for implementing data management.
* Identify global resources and the control mechanisms required to access them.
* Design an appropriate control mechanism for the system, including task management.
* Consider how boundary conditions should be handled.
* Review and consider trade-offs.

In actually the computer system design process is an exercise of specifying “How” I the system will work. It is an interactive process, which is based on “what” the system will do. Mainly, following five parts have been included in the system design process:

1. **Output Design: -**

The starting point of the design process is the proper knowledge of system requirements, which will normally be converted in terms of output.

1. **Input Design: -**

**Once** the output requirements have been finalized, the next step is to find out available to the system to produce the desired outputs. the basic documents in which these data are available need to be identified. If necessary, these documents may have to be revised or new documents may have to be introduced.

1. **File Design:-**

Once the input data is captured in the system, these may have to be preserved either for a short or long period. These data will generally be stored in files in a logical manner. The designer will have to devise the techniques of storing and retrieving data from these files.

1. **Procedure Design:-**

This step involves specification of how processing will be performed in this , there are two aspects:

* Computer procedure.
* Non-computer procedure.
* The computer procedure will specify what Functions will be carried out on computer, what will be different programs and in what sequence the programs will be run. The non-computer procedure will specify the manual procedures for feeding input data, receiving output etc.

1. **Control Design: -**

Control design indicates necessary procedures which will ensure correctness of processing, accuracy of data; timely output etc.this will ensure that the system is functioning as per plan.

1. ***Input Design***

The design of input play very significant role in getting the correct output. It covers all phases of input from creation of initial data to actual entering the data to the system for processing. The input design is the link that ties the information system into the world of its users. Some feature of design may vary depending on whether the system is batch oriented or online.

The various objectives of input design focus on:

* Controlling amount of input.
* Avoiding delay.
* Avoiding errors in data.
* Avoiding extra steps.
* Keeping the process simple.

**Major Concerns Regarding Input:**

We must consider on some important points regarding input, they are:-

* What input is needed?
* How and where is the input created?
* How should the source documents be designed?
* What format should be used for the input records?
* What medium should be used for recording the input?

**Output Design**

Presenting the data processed by a computer-based information system in an attractive and usable form has become very essential. Success and acceptance of a system to some extent depend on good presentation. Therefore, system analysts must know fully how to design output report in an market attractive way. Many new output devices are being introduced in the market because of recent development in computer technology.

1. For communication to the persons concerned.
2. For re-input to the computer for being connected with other data and further processing.
3. For permanent storage.

**Output Design consideration:**

An output to be produced usually depends upon the following consideration:-

1. **Type of User and Purpose:**

Generally different levels of users will have different requirements from the system. Some want exception reports (e.g. when transaction fall below a certain level), some want summary reports (e.g. transaction quantity and value for each region) while some want details.

Again statutory reports will normally be as per requirement specified under the law and the designer will not have much flexibility to change to the format.

**(b). Contents:**

The data that are needed to be included in the Output. These will be related to the purpose of the input.

**(C). Format:**

This refers to the arrangement of data on the report, size of the paper , titles, headlines, color of the paper etc.

**(d). Frequency and Timing:**

At what frequency (monthly, annually etc.) and when (after annual closing of transaction, after the end of the financial year, before the last day of every month etc.)

**(e). Volume:**

Often sheer volume of the output deters one from using the outputs. The sheer bulk of the report may also create problems for handling, filling or printing time.

**(f). Sequence:** The usefulness of an output very often depends on the sequence of data printed. A proper sequence will also help distribution of outputs to different users (e.g. pay-slips printed department wise facilities easier payment).

**(g). quality:**

This relates to the content, appearance and accuracy of the output generated for external users should be given special attention in respect of its get-up, quality of paper etc.

**(h).** **Type of Stationary:** Reports can be generated on ordinary blank stationary or on specially printed stationary that is useful when most of the contents of the input are constant. This type of stationary has the following advantages:-

***Form Design***

As we know that data provide the basis for information systems. Without data there is no system, but data must be fed in correct way so that the information produced must be in a format acceptable to the user. In either case, it is still data- the basic element of a printed form.

A form is tool with a message. It is the physical carrier of data of information. It also can constitute authority for action. It is hard to imagine a business operating without using forms. They are the vehicles for most communications and the blure print for many activities. It is as important as a printed form is, however, the majority of forms are designed by poorly trained people who are puzzled by confusing forms, they ask for directions on how to read them and how to fill them out. When a form is poorly designed, it is a poor (and costly) administrative tool.

**Forms:**

A printed form is generally classified by what it does in the system. There are three primary classifications: Entry.

**Entry *Forms:***

The user had designed entry Forms for Data Entry Work.

**Factors to be considered in Form Design:**

Form design plays an important role in data processing. Form must have the appearance of a well-conceived and attractive design. Some of the important, factors, which should be taken care of given below.

1. Size and shape of the form should be such that it is convenient for handling, filling, sorting etc.
2. Arrange the material in logical orders so that it becomes easy to fill it up.
3. The form title must clearly identify its purpose.
4. Precise contents should be recorded. Adequate and compact space should be provided for items to be recorded. Pre-printed entries should be taken care of.
5. Special features like security and control are to be considered.
6. Introduce emphasis by shading columns, heavy lines, etc. if the form is to be used for specific clerical operation, for example copying or checking. See that the detail is arranged and spaced to provide maximum help to the operation.
7. The form designer should design the form in such a way so as to cover the specific needs of the purpose for which it is designed.

***File Design***

After designing the input and Output, the designer begins to pay his attention on the work of rile designing or how data should be organized around user requirements. How data are organized depends on the data and response requirements that determine hardware configurations. System analyst is responsible for designing the files and selecting their contents, selecting from options available for organizing the data. File organization may be sequential, index sequential, inverted list or random. Each method has its own uses and abuses.

An integral approach to file designs the database. The general theme is to handle information as an integrated whole, with a minimum of redundancy and improved performance. Various software techniques are applied to manipulated, describe and manage data. Irrespective of type data structure used, the main objectives of database are accurate and integrity, successful recovery from failure, privacy and security of data.

Files are the heart of a computer application. The basic terms used to describe file are Data item, record, File and Database.

*Data Item:*

A basic or individual element of data is called data item. Each Date item is assigned a value. Data Item is sometimes referred to as a field.

*Record:*

The collection of related data items is called a record. The analyst also determines the length and type of each field.

***File****:*

File is a collection of related records. Each record in a file is included because it pertains to the same entity.

**Database:**

The highest level in the hierarchy is the database. It is set of inter-related files for real time processing. It contains the necessary data for problem solving and can be used for several users who are accessing data concurrently.

***Code Efficiency***

Every step has been considered in order to achieve the code efficiency. “**Java”** has been used in order to achieve the high degree or optimality of code. The event driven and oops concept reduce the code. SQL meets only the target (what has to be done) to be specified and it does no need the steps (how) to be followed to achieve target. The data retrieved is done by SQL engine of the RDBMS, which is very efficient, and it increases the efficiency of the application as well. As far as user interface design of the application is concerned, object oriented paradigm has applied which reduces the lengthy codification.

***Code Optimization***

The application has been designed using two-tier. Client/server technology in which the functions of the applications segregated between the two tiers i.e. the Back End and the Front End. Every type of data security, constraints and validation rules has been applied.

At the Back End Database itself, which is as per the code’s rules, due to which less codification is needed at the Front End user interface level? Due to this approach, it is possible to optimize the codes so that it can makes less memory to execute and makes execution fast.

***MODULE DESIGN***

The above work processing may be divided into following modules –

1. **Modules for master file of distributer details**
2. **Modules for distributor state details files**
3. **Modules for distributor phone no**
4. **Modules for distributor fax**
5. **Modules for distributor status**
6. **Modules for distributor e-mail**

The data follow diagram for the whole project, which includes is given below-

**Distributer phone NO.**

**Distributor State**

**Distributor fax no**

**Distributor status**

**Distributor e-mail**

**DFD 1: Level 0: Overall flow of Data**

**Module 1: Master file of Retailer Details-**

This module is used for storing the details of the Retailer of Anuj Dairy pvt ltd.The details of new employee are at first entered into the master file named REGISTRATION.

The DFD for this module is shown below-

**Date**

**Retailer\_id**

View & Print

**Add/ update/**

**R\_name**

**R\_Address**

Modify/Delete The Record

**DFD 2: Level 1 Processing of data of Retailers**

**Process logic:**

1. This module is used for entering and storing the records of new retailer.
2. It gives also the report of new admitted retailer.
3. It also modifies (edits), delete the records of new admitted retailer.
4. It can display the report of new admitted retailer on monitor and print using printer. (Report may be of daily, monthly, &year
5. **Module 2: Purchase Details**

This module is used to store dues (installments) and installment. The data

**Net Purchase**

**Amount**

**Purchase data**

**Discount on purchase**

**Purchase ID**

**VAT on Purchase**

***DFD 3: Level 1: processing of Monthly Purchasing Details***

**Process logic:**

1. This module retrieves the details from “master wise and a detail” tables.
2. This module stores the data of purchase raw materials.
3. This module also updates or modifies the data.
4. This module shows the reports of purchase items on the VDU and prints the report through printer.
5. It also prints the all purchased raw materials.

**Module 3 : Module of Sale Details**

Sale Quantity

Sale Product

add/update

Enters data

Retrives

Demand report

Gives report

Retrives

D

Total Prices

Unit price

**DFE 4: Level 1: Processing of sale details**

**Process Logic:**

1. This modules test that entered Product Id and then according to product id we can sell the products.
2. It also gives the reports of all sold products. It also gives reports daily, weekly and monthly sales records.

**Module 4: Add Party Details**

**Party e-mail**

**Party address**

**Party phone no-**

**Party state**

**Party status**

**Party city**

**DFD 5: Level 1: Process for Modifying record(s) in all table**

**Process Logic:**

1. This module modifies the wrong stored data of Employee details from Registration if given Registration no exists.
2. This module modifies or edits the wrong stored employee from employee table, after checking the given record no. exists.
3. It modifies or updates the wrong stored modules details of a employee into Employee table, if give Registration no, of a employee exist into this table.

**Module 5: Module for Stock Details**

This module is used to create a new Details of Anuj Dairy Pvt. Ltd. Stock and data of this table stored into a master file stock member wants to issue or to issue or to deposit a Product from or to a Library, their data are stored into a file named Stock details. Information of new old book are stored into a file Stock details.

**Demands report Add/save**

**Sends**

**the**

**Report**

**Product id**

**Stock date**

**Product quantity**

**Stock master**

**Stock time**

**Total stock amount**



**Coding**

#include<dos.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<process.h>

#include<conio.h>

#include<graphics.h>

#include<iostream.h>

////STRUCTURE DEFINITION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

struct party

{

char pid[10],pname[25],paddr[25],pgen[25],email[25],mob[25],pcity[30],pstate[25],pdob[25];

float esal;

};

struct distributor

{

char did[10],dname[25],dadd[25],dcity[25],ddistt[25],dpin[25],dmob[25],gen[25],email[30];

};

void textflash1()

{

int c,i=0;

do

{

i++;

window(1,1,40,55);

clrscr(); textmode(C40);

for(c=2;c<40;c++)

{

textcolor(random(15)+BLINK);

gotoxy(c,1);

putch('\*');delay(100);

}

for(c=1;c<25;c++)

{

textcolor(random(15)+BLINK);

gotoxy(40,c);

putch('\*');delay(100);

}

for(c=40;c>1;c--)

{

textcolor(random(15)+BLINK);

gotoxy(c,24);

putch('\*');delay(100);

}

for(c=24;c>1;c--)

{

textcolor(random(15)+BLINK);

delay(100);

gotoxy(1,c);

putch('\*');

}

textbackground(random(7));

textcolor(random(15)+BLINK);

gotoxy(random(4),12);

cputs(" BCA DEPARTMENT \* ");

textcolor(random(15)+BLINK);

gotoxy(random(30),26);

cputs(" \* NALANDA COLLEGE ");

sleep(4);

}while(i!=1);

textmode(3);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//2ND FLASS SCREEN:-GRAPHICS MODE

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void graphflash1()

{

int x=1,y=1,gd,gm;

gd=DETECT;

initgraph(&gd,&gm,"..//BGI");

int c=240;

do

{

setcolor(random(x));

//circle(getmaxx()/2,c++,x);

circle(x,x,x);delay(10);

circle(640-x,x,x);delay(10);

circle(x,480-x,x);delay(10);

circle(640-x,480-x,x);delay(10);

//circle(getmaxx()/2,getmaxy()/2,y);

settextstyle(7,0,4);

setcolor(random(y));

outtextxy(getmaxx()/2-85,getmaxy()/2-50,"PROJECT OF");

outtextxy(getmaxx()/2-30,getmaxy()/2-20,"DAIRY");

outtextxy(getmaxx()/2-60,getmaxy()/2+10,"BILLING");

settextstyle(10,0,1);

outtextxy(getmaxx()/4+8,getmaxy()/2+200,"Name:- SIMRAN SINGH");

delay(60);

x++;y++;

if(x==97)

{

x=1;c=240;cleardevice();

}

if(y==130)y=1;

}while(!kbhit());

}

void login();

void date()

{

struct dosdate\_t d;

\_dos\_getdate(&d);

gotoxy(50,1);

printf("%d-%d-%d",d.day,d.month,d.year);

//cout<<d.day<<"-"<<d.month<<"-"<<d.year;

}

void time()

{

struct dostime\_t t;

\_dos\_gettime(&t);

gotoxy(68,1);

printf("%d:%d:%d",t.hour,t.minute,t.second);

}

void menu();

void main1()

{

int gd=DETECT,gm;

initgraph(&gd,&gm,"..//bgi");

setbkcolor(RED);

setcolor(YELLOW);

date();

time();

settextstyle(1,HORIZ\_DIR,5);

outtextxy(100,20,"DAIRY BILLING SYSTEM");

settextstyle(1,HORIZ\_DIR,1);

outtextxy(160,80,"DEVELOPED BY SIMRAN SINGH");

outtextxy(230,110,"STUDENT OF B.C.A.");

outtextxy(260,140,"N.C. M.U.");

//outtextxy(170,400,"PRESS ANY KEY TO CONTINUE ...");

sleep(8);

getch();

login();

//menu1();

closegraph();

}

void login()

{

int gd=DETECT,gm,i,c=0;

char pass[20],ch,uname[20];

login:

i=0;

clrscr();

initgraph(&gd,&gm,"..//bgi");

date();

time();

setbkcolor(RED);

setcolor(YELLOW);

settextstyle(1,HORIZ\_DIR,2);

if(c>2)

{

gotoxy(25,20);

cout<<"YOU HAVE LOST A CHANCE";

sleep(5);

exit(0);

}

outtextxy(50,30,"WELOCME TO LOGIN IN DAIRY BILLING SYSTEM");

gotoxy(30,7);

cout<<"press ESC to skip password and any key to continue..";

if(getch()==27) goto skip;

gotoxy(30,11);

cout<<"ENTER USER NAME :";

cin.getline(uname,30);

gotoxy(30,13);

cout<<"ENTER PASSWORD :";

gotoxy(48,13);

c++;

do{

ch=getch();

if(ch==27)

{

exit(0);

}

else

{

if(ch==13)

{

pass[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(48+i,13);

cout<<" \b";

}

}

else

{

gotoxy(48+i,13);

pass[i]=ch;

cout<<"\*";

i++;

if(i>10)

{

gotoxy(25,20);

cout<<"PASSWORD NOT GREATER THAN 10 CHARARACTERS LONG";

getch();

goto login;

}

}

}

}

}while(1);

if(i<5)

{

gotoxy(25,20);

cout<<"PASSWORD LENGTH MINIMUM 5 CHARACTERS LONG";

getch();

goto login;

}

if(strcmp(uname,"bihar")==0)

{

if(strcmp(pass,"nalanda")==0)

{

gotoxy(25,20);

cout<<"VALID PASSWORD";

getch();

//menu();

}

else

{

gotoxy(25,20);

cout<<"INVALID PASSWORD";

getch();

goto login;

//getch();

//exit(0);

}

}

else

{

gotoxy(25,20);

cout<<"INVALID USER NAME";

getch();

goto login;

}

skip:

}

struct sale

{

char ino[10],pid[25],pname[25];

int qn;

float tp;

};

struct product

{

char pid[5],pname[5];

float p;

};

struct passbook{

char pid[9],pname[10],p[10],acno[10];

};

struct retailer

{

char rno[10],rname[25],radd[25],rcity[25],mob[10];

int bno;

};

void pw()

{

cnt:

clrscr();

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" PASSWORD VARIFICATION";

gotoxy(18,3);

cout<<"===========================================" ;

char olb[20]={" "};

char passw1[20],passw2[20],ch;

int i=0;

textcolor(WHITE);

textbackground(GREEN);

gotoxy(15,6);cprintf("%s",olb);

gotoxy(45,6); cprintf("%s",olb);

gotoxy(15,8);cprintf("%s",olb);

gotoxy(45,8); cprintf("%s",olb);

gotoxy(15,10);cprintf("%s",olb);

gotoxy(45,10); cprintf("%s",olb);

gotoxy(15,6);cprintf(" ENTER PASSWORD : ");

gotoxy(15,8);cprintf(" CONFIRM PASSWORD : ");

gotoxy(15,10);cprintf(" PASSWORD STATUS : ");

i=0;

gotoxy(45,6);

do{

ch=getch();

if(ch==27)

{

exit(0);

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,6);

passw1[i]=ch;

cprintf("\*");

i++;

}

}

}

}while(1);

///

i=0;

gotoxy(45,8);

do{

ch=getch();

if(ch==27)

{

exit(0);

}

else

{

if(ch==13 || i==19)

{

passw2[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,8);

passw2[i]=ch;

cprintf("\*");

i++;

}

}

}

}while(1);

gotoxy(45,10);

if(strcmp(passw1,passw2)==0)

{

if(strcmp(passw1,"god")==0)

{

cprintf(" RIGHT PASSWORD ");

}

else

{

cprintf(" WRONG PASSWORD ");

getch();

exit(0);

}

}

else

{

cprintf(" WRONG CONFIRM ");

getch();

goto cnt;

}

getch();

}

void printlb()

{

char olb[20]={" "};

textcolor(WHITE);

textbackground(GREEN);

gotoxy(15,6);cprintf("%s",olb);

gotoxy(45,6); cprintf("%s",olb);

gotoxy(15,8);cprintf("%s",olb);

gotoxy(45,8); cprintf("%s",olb);

gotoxy(15,10);cprintf("%s",olb);

gotoxy(45,10); cprintf("%s",olb);

gotoxy(15,12);cprintf("%s",olb);

gotoxy(45,12); cprintf("%s",olb);

gotoxy(15,14);cprintf("%s",olb);

gotoxy(45,14); cprintf("%s",olb);

gotoxy(15,6);cprintf(" INVOICE NO : ");

gotoxy(15,8);cprintf(" PRODUCT ID : ");

gotoxy(15,10);cprintf(" PRODUCT NAME : ");

gotoxy(15,12);cprintf(" QUANTITY : ");

gotoxy(15,14);cprintf(" TOTAL PRICE : ");

gotoxy(45,8);

}

void csearch()

{

cse:

clrscr();

sale r1;

char s1[20],ch,ch1;

int i=0,f=0;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN SEARCH ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb();

gotoxy(45,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

FILE \*fp;

fp=fopen("sale.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(strcmpi(s1,r1.ino)==0)

{

f=1;

gotoxy(45,6); cprintf("%s",r1.ino);

gotoxy(45,8); cprintf("%s",r1.pid);

gotoxy(45,10); cprintf("%s",r1.pname);

gotoxy(45,12); cprintf("%d",r1.qn);

gotoxy(45,14); cprintf("%.2f",r1.tp);

}

}

}

fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(17,18);

cprintf("Record not found, Pls enter valid course code");

getch();

goto cse;

}

else

{

gotoxy(20,18);

cprintf("Do u want to search another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto cse;

}

else

{

return;

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

///////////////////////////////////////DELETION OF RECORD

//////////////////////////////////////DELETION OF RECORD

///////////////////////////////////// DELETION OF RECORD

void cdel()

{

cde:

clrscr();

sale r1;

char s1[20],ch,ch1;

int i=0,f=0;

textcolor(YELLOW);

textbackground(BLACK);

char ts1[10],ts2[10];

int tn;

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN DELETE OPERATION ";

gotoxy(18,3);

cout<<"===========================================" ;

printlb();

gotoxy(45,6);

do{

ch=getch();

if(ch==27)

return;

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

FILE \*fp,\*tp;

fp=fopen("sale.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

tp=fopen("mytemp.dat","ab");

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(!feof(fp))

{

if(strcmpi(s1,r1.ino)!=0)

{

fwrite(&r1,sizeof(r1),1,tp);

}

else

{

f=1;

}

}

}

fclose(fp);

fclose(tp);

// fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(17,18);

cprintf("Record not found, Pls enter valid course code");

remove("mytemp.dat");

getch();

goto cde;

}

else

{

remove("party.dat");

rename("mytemp.dat","party.dat");

gotoxy(20,18);

cprintf("Do u want to Delete another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto cde;

}

else

{

return;

}

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

///////////////////////////////////// DELETION OF RECORD

/////////////////////////////////// DELETION OF RECORD

/////////////////////////////////// DELETION OF RECORD

void cupdate()

{

cue:

clrscr();

sale r1;

char s1[20],ch,ch1;

int i=0,f=0,count;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN UPDATE ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb();

gotoxy(45,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

count=0;

FILE \*fp;

fp=fopen("sale.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

while(!feof(fp))

{

count++;

fread(&r1,sizeof(r1),1,fp);

if(strcmpi(s1,r1.ino)==0)

{

//strcpy(r1.ccode,s1);

f=1;

i=0;

fclose(fp);

gotoxy(45,8);

// cprintf("%s",r1.cname);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,8);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pid,s1);

i=0;

gotoxy(45,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,10);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pname,s1);

//gotoxy(45,12);

i=0;

gotoxy(45,12);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,12);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,12);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

r1.qn=atof(s1);

i=0;

gotoxy(45,14);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,14);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,14);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

r1.tp=atoi(s1);

fp=fopen("sale.dat","rb+");

// fseek(fp,0,SEEK\_END);

count--;

fseek(fp,sizeof(r1)\*count,SEEK\_SET);

fwrite(&r1,sizeof(r1),1,fp);

}

}

}

fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(17,18);

cprintf("Record not found, Pls enter valid course code");

getch();

goto cue;

}

else

{

gotoxy(20,18);

cprintf("Do u want to update another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto cue;

}

else

{

return;

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

void centry()

{

cel:

clrscr();

sale r1;

int cc=0,i,f;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN CUSTOMER ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb();

char s1[20]={"C00"},ts[20],passw1[30],ch,cnt;

FILE \*fp;

fp=fopen("sale.dat","rb");

if(fp==NULL)

{

cc=1;

}

else

{

f=0;

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

f++;

}

fclose(fp);

fp=fopen("sale.dat","rb");

f--;

fseek(fp,sizeof(r1)\*f,SEEK\_SET);

fread(&r1,sizeof(r1),1,fp);

i=0;

while(r1.ino[i+3]!='\0')

{

ts[i]= r1.ino[i+3];

i++;

}

ts[i]='\0';

cc= atoi(ts);

cc++;

}

///////current working area????????????????????????/

fclose(fp);

textcolor(YELLOW);

itoa(cc,ts,10);

strcat(s1,ts);

strcpy(r1.ino,s1);

gotoxy(45,6);

cprintf("%s",r1.ino);

////////////////////????//

i=0;

gotoxy(45,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,8);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pid,passw1);

//////////////

i=0;

gotoxy(45,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,10);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pname,passw1);

//gotoxy(45,12);

i=0;

gotoxy(45,12);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,12);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,12);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

r1.qn=atof(s1);

i=0;

gotoxy(45,14);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(45+i,14);

cprintf(" \b");

}

}

else

{

gotoxy(45+i,14);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

r1.tp=atof(s1);

fp=fopen("sale.dat","ab");

fseek(fp,0,SEEK\_END);

fwrite(&r1,sizeof(r1),1,fp);

fclose(fp);

gotoxy(12,18);cprintf("ONE RECORD SAVED, DO U WANT TO ENTER ANOTHER RECORD Y/N: ");

cnt=getch();

if(cnt=='y'||cnt=='Y')goto cel;

//gotoxy(45,14);

}

void cdisp()

{

clrscr();

char sv1;

FILE \* fp;

sale r1;

fp=fopen("sale.dat","rb");

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" DISPLAY COURSE INFORMATION";

gotoxy(18,3);

cout<<"===========================================" ;

printlb();

/\* struct course

{

char ccode[10],cname[30],ctd[20],pmode[20];

float cfee;

}; \*/

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(!feof(fp))

{

gotoxy(45,6);

cprintf("%s",r1.ino);

gotoxy(45,8);

cprintf("%s",r1.pid);

gotoxy(45,10);

cprintf("%s",r1.pname);

gotoxy(45,12);

cprintf("%d",r1.qn);

gotoxy(45,14);

cprintf("%.2f",r1.tp);

sv1=getch();

if(sv1==27)

{

fclose(fp);

break;

}

printlb();

}

}

fclose(fp);

}

///END OF STRUCTURE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void printlb1();

void hentry();

void hdisp();

void hsearch();

void hupdate();

void hdel();

void printlb2();

void eentry();

void edisp();

void esearch();

void eupdate();

void edel();

void printlb3();

void lentry();

void ldisp();

void lsearch();

void printlb4();

void gentry();

void gdisp();

void gdel();

void gsearch();

void pentry();

void pdisp();

void pdel();

void psearch();

void pupdate();

void main()

{

textflash1();

graphflash1();

main1();

// pw();

char menu[5][19]={" PARTY","DISTRIBUTOR ","SALE","PURCHASE ","PRODUCT"};

char csm[5][15]={"Entry ","Display ","Search ","Update ","Delete "};

char tsm[5][15]={"Entry ","Display ","Search ","Update ","Delete "};

char lsm[5][15]={"Entry ","Display ","Search ","Update ","Delete "};

char ssm[6][15]={"Entry ","Display ","Search ","Update ","Delete "};

char psm[5][15]={"Entry ","Display ","Search ","Update ","Delete "};

char clb[80]={" "};

int x,y,i,ch1,ch2;

static int f=-1;

LB:

textbackground(BLACK);

clrscr();

textcolor(RED);

textbackground(YELLOW);

gotoxy(1,1);

cprintf("%s",clb);

gotoxy(1,2);

cprintf("%s",clb);

gotoxy(1,3);

cprintf("%s",clb);

x=1;

textcolor(7);

textbackground(BLACK);

gotoxy(1,24);

textbackground(GREEN);

textcolor(4);

cprintf(" F1 ->CUSTOMER., F2->GODOWN., F3->EMPLOYEE., F4->HAWKER., F5->PASSBOOK. ");

for(i=0;i<5;i++)

{

if(i==0)

{

gotoxy(1,2);

textbackground(YELLOW);

cprintf(" ");

textbackground(BLACK);

x=3;

}

gotoxy(x,2);

cprintf("%s",menu[i]);

x=x+14;

cprintf(" ");

}

textcolor(RED);

textbackground(YELLOW);

for(x=1;x<80;x++)

{

gotoxy(x,1);

cprintf("\*");

gotoxy(x,25);

cprintf("\*");

}

for(y=1;y<25;y++)

{

gotoxy(1,y);

cprintf("||");

gotoxy(79,y);

cprintf("||");

}

gotoxy(79,1);cprintf("\* ");

gotoxy(1,1);cprintf(" \*");

char ch;

/\* do

{

ch=getch();

if(ch==0)

{

ch=getch();

gotoxy(12,15);

cprintf("%d EXT",ch);

}

else

{

cprintf("%d ASCII",ch);

}

}while(ch!=27);

\*/

do{

ch=getch();

if(ch==0)

{

ch=getch();

if(ch==59||ch==60||ch==61||ch==62||ch==63)

{

textcolor(RED);

textbackground(YELLOW);

gotoxy(1,3);

cprintf("%s",clb);

}

if(ch==59)

{

//???COURSE LOGIC\*\*\*\*\*\*

int v1,v2;

textbackground(BLACK);

for(v1=3;v1<78;v1++)

{

for(v2=4;v2<23;v2++)

{

gotoxy(v1,v2);

cprintf(" ");

}

}

textcolor(BLUE);

textbackground(RED);

for(i=0;i<5;i++)

{

gotoxy(3,i+3);

cprintf("%s",csm[i]);

}

///ARROW MOVEMENT IN COURSE MENU

do

{

textcolor(BLUE);

textbackground(RED);

ch1=getch();

if(ch1==13)

{

if(f==0)

{ f=-1;

centry();

goto LB;

}

else if(f==1)

{

f=-1;

cdisp();

goto LB;

}

else if(f==2)

{

f=-1;

csearch();

goto LB;

}

else if(f==3)

{

f=-1;

cupdate();

goto LB;

}

else if(f==4)

{

f=-1;

cdel();

goto LB;

}

}

if(ch1==0)

{

ch1=getch();

if(ch1==80)

{

f++;

}

else if(ch1==72)

{

if(f==-1)f=4;

else f--;

}

for(i=0;i<5;i++)

{

if(f==i)

{

textcolor(GREEN);

textbackground(BLUE);

gotoxy(3,i+3);

cprintf("%s",csm[i]);

textcolor(BLUE);

textbackground(RED);

}

else

{

gotoxy(3,i+3);

cprintf("%s",csm[i]);

}

if(f==5)f=-1;

}

}

/\* ch2=getch();

if(ch2==13)centry(); \*/

if(ch1==27){f=-1; goto LB;}

}while(1);

}

///////////godown logic ///

if(ch==61)

{

//???godown LOGIC\*\*\*\*\*\*

int v1,v2;

textbackground(BLACK);

for(v1=3;v1<78;v1++)

{

for(v2=4;v2<23;v2++)

{

gotoxy(v1,v2);

cprintf(" ");

}

}

textcolor(BLUE);

textbackground(RED);

for(i=0;i<5;i++)

{

gotoxy(31,i+3);

cprintf("%s",csm[i]);

}

///godown MENU ARROW --> MOVEMENT LOGIC\*\*\*

do

{

textcolor(BLUE);

textbackground(RED);

ch1=getch();

if(ch1==13)

{

if(f==0)

{ f=-1;

gentry();

goto LB;

}

else if(f==1)

{

f=-1;

gdisp();

goto LB;

}

else if(f==2)

{

f=-1;

gsearch();

goto LB;

}

else if(f==3)

{

f=-1;

// aupdate();

goto LB;

}

else if(f==4)

{

f=-1;

gdel();

goto LB;

}

}

if(ch1==0)

{

ch1=getch();

if(ch1==80)

{

f++;

}

else if(ch1==72)

{

if(f==-1)f=4;

else f--;

}

for(i=0;i<5;i++)

{

if(f==i)

{

textcolor(GREEN);

textbackground(BLUE);

gotoxy(31,i+3);

cprintf("%s",csm[i]);

textcolor(BLUE);

textbackground(RED);

}

else

{

gotoxy(31,i+3);

cprintf("%s",csm[i]);

}

if(f==5)f=-1;

}

}

if(ch1==27){f=-1;goto LB;}

}while(1);

}

///////employee logic//////

if(ch==60)

{

//???employee LOGIC\*\*\*\*\*\*

int v1,v2;

textbackground(BLACK);

for(v1=3;v1<78;v1++)

{

for(v2=4;v2<23;v2++)

{

gotoxy(v1,v2);

cprintf(" ");

}

}

textcolor(BLUE);

textbackground(RED);

for(i=0;i<5;i++)

{

gotoxy(17,i+3);

cprintf("%s",tsm[i]);

}

do

{

textcolor(BLUE);

textbackground(RED);

ch1=getch();

if(ch1==13)

{

if(f==0)

{ f=-1;

eentry();

goto LB;

}

else if(f==1)

{

f=-1;

edisp();

goto LB;

}

else if(f==2)

{

f=-1;

esearch();

goto LB;

}

else if(f==3)

{

f=-1;

eupdate();

goto LB;

}

else if(f==4)

{

f=-1;

edel();

goto LB;

}

}

if(ch1==0)

{

ch1=getch();

if(ch1==80)

{

f++;

}

else if(ch1==72)

{

if(f==-1)f=4;

else f--;

}

for(i=0;i<5;i++)

{

if(f==i)

{

textcolor(GREEN);

textbackground(BLUE);

gotoxy(17,i+3);

cprintf("%s",tsm[i]);

textcolor(BLUE);

textbackground(RED);

}

else

{

gotoxy(17,i+3);

cprintf("%s",tsm[i]);

}

if(f==5)f=-1;

}

}

if(ch1==27){f=-1;goto LB;}

}while(1);

}

/////////////////Library logic

if(ch==62)

{

//???COURSE LOGIC\*\*\*\*\*\*

int v1,v2;

textbackground(BLACK);

for(v1=3;v1<78;v1++)

{

for(v2=4;v2<23;v2++)

{

gotoxy(v1,v2);

cprintf(" ");

}

}

textcolor(BLUE);

textbackground(RED);

for(i=0;i<8;i++)

{

gotoxy(45,i+3);

cprintf("%s",lsm[i]);

}

do

{

textcolor(BLUE);

textbackground(RED);

ch1=getch();

if(ch1==13)

{

if(f==0)

{ f=-1;

hentry();

goto LB;

}

else if(f==1)

{

f=-1;

hdisp();

goto LB;

}

else if(f==2)

{

f=-1;

hsearch();

goto LB;

}

else if(f==3)

{

f=-1;

hupdate();

goto LB;

}

else if(f==4)

{

f=-1;

hdel();

goto LB;

}

}

if(ch1==0)

{

ch1=getch();

if(ch1==80)

{

f++;

}

else if(ch1==72)

{

if(f==-1)f=7;

else f--;

}

for(i=0;i<8;i++)

{

if(f==i)

{

textcolor(GREEN);

textbackground(BLUE);

gotoxy(45,i+3);

cprintf("%s",lsm[i]);

textcolor(BLUE);

textbackground(RED);

}

else

{

gotoxy(45,i+3);

cprintf("%s",lsm[i]);

}

if(f==8)f=-1;

}

}

if(ch1==27){f=-1;goto LB;}

}while(1);

}

////////////sport logic

if(ch==63)

{

//???COURSE LOGIC\*\*\*\*\*\*

int v1,v2;

textbackground(BLACK);

for(v1=3;v1<78;v1++)

{

for(v2=4;v2<23;v2++)

{

gotoxy(v1,v2);

cprintf(" ");

}

}

textcolor(BLUE);

textbackground(RED);

for(i=0;i<5;i++)

{

gotoxy(59,i+3);

cprintf("%s",psm[i]);

}

///ARROW MOVEMENT IN COURSE MENU

do

{

textcolor(BLUE);

textbackground(RED);

ch1=getch();

if(ch1==13)

{

if(f==0)

{ f=-1;

pentry();

goto LB;

}

else if(f==1)

{

f=-1;

pdisp();

goto LB;

}

else if(f==2)

{

f=-1;

psearch();

goto LB;

}

else if(f==3)

{

f=-1;

pupdate();

goto LB;

}

else if(f==4)

{

f=-1;

pdel();

goto LB;

}

}

if(ch1==0)

{

ch1=getch();

if(ch1==80)

{

f++;

}

else if(ch1==72)

{

if(f==-1)f=4;

else f--;

}

for(i=0;i<5;i++)

{

if(f==i)

{

textcolor(GREEN);

textbackground(BLUE);

gotoxy(59,i+3);

cprintf("%s",psm[i]);

textcolor(BLUE);

textbackground(RED);

}

else

{

gotoxy(59,i+3);

cprintf("%s",psm[i]);

}

if(f==5)f=-1;

}

}

/\* ch2=getch();

if(ch2==13)centry(); \*/

if(ch1==27){f=-1; goto LB;}

}while(1);

}

}

}while(ch!=27);

}

void printlb2()

{

char dlb[20]={" "};

char ilb[20]={" "};

textcolor(WHITE);

textbackground(GREEN);

gotoxy(3,6);cprintf("%s",ilb);

gotoxy(20,6); cprintf("%s",dlb);

gotoxy(3,8);cprintf("%s",ilb);

gotoxy(20,8); cprintf("%s",dlb);

gotoxy(3,10);cprintf("%s",ilb);

gotoxy(20,10); cprintf("%s",dlb);

gotoxy(3,12);cprintf("%s",ilb);

gotoxy(20,12); cprintf("%s",dlb);

gotoxy(3,14);cprintf("%s",ilb);

gotoxy(20,14); cprintf("%s",dlb);

gotoxy(42,6);cprintf("%s",ilb);

gotoxy(59,6); cprintf("%s",dlb);

gotoxy(42,8);cprintf("%s",ilb);

gotoxy(59,8); cprintf("%s",dlb);

gotoxy(42,10);cprintf("%s",ilb);

gotoxy(59,10); cprintf("%s",dlb);

gotoxy(42,12);cprintf("%s",ilb);

gotoxy(59,12); cprintf("%s",dlb);

gotoxy(3,6);cprintf(" PARTY ID :");

gotoxy(3,8);cprintf(" PARTY NAME :");

gotoxy(3,10);cprintf(" PARTY ADDRESS :");

gotoxy(3,12);cprintf(" CITY :");

gotoxy(3,14);cprintf(" STATE :");

gotoxy(42,6);cprintf(" EMAIL :");

gotoxy(42,8);cprintf(" MOB NO :");

gotoxy(42,10);cprintf(" D.O.B. :");

gotoxy(42,12);cprintf(" GENDER :");

gotoxy(20,8);

}

void eentry()

{

tel:

clrscr();

party r1;

int cc=0,i,f;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN party DETAIL ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb2();

char s1[20]={"T00"},ts[20],passw1[30],ch,cnt;

FILE \*fp;

fp=fopen("party.dat","rb");

if(fp==NULL)

{

cc=1;

}

else

{

f=0;

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

f++;

}

fclose(fp);

fp=fopen("party.dat","rb");

f--;

fseek(fp,sizeof(r1)\*f,SEEK\_SET);

fread(&r1,sizeof(r1),1,fp);

i=0;

while(r1.pid[i+3]!='\0')

{

ts[i]= r1.pid[i+3];

i++;

}

ts[i]='\0';

cc= atoi(ts);

cc++;

}

///////current working area????????????????????????/

fclose(fp);

textcolor(YELLOW);

itoa(cc,ts,10);

strcat(s1,ts);

fclose(fp);

strcpy(r1.pid,s1);

gotoxy(20,6);

cprintf("%s",r1.pid);

////////////////////????//

i=0;

gotoxy(20,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,8);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pname,passw1);

//////////////

i=0;

gotoxy(20,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,10);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.paddr,passw1);

//gotoxy(45,12);

i=0;

gotoxy(20,12);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,12);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,12);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pcity,passw1);

// r1.cfee=atof(passw1);

i=0;

gotoxy(20,14);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,14);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,14);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pstate,passw1);

i=0;

gotoxy(59,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,6);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.email,passw1);

// r1.cfee=atof(passw1);

i=0;

gotoxy(59,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,8);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.mob,passw1);

i=0;

gotoxy(59,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,10);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pdob,passw1);

// r1.cfee=atof(passw1);

i=0;

gotoxy(59,12);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,12);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,12);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pgen,passw1);

//r1.esal=atof(passw1);

//i=0;

fp=fopen("party.dat","ab");

fseek(fp,0,SEEK\_END);

fwrite(&r1,sizeof(r1),1,fp);

fclose(fp);

gotoxy(12,22);cprintf("ONE RECORD SAVED, DO U WANT TO ENTER ANOTHER RECORD Y/N: ");

cnt=getch();

if(cnt=='y'||cnt=='Y')goto tel;

}

void edisp()

{

clrscr();

char sv1;

FILE \* fp;

party r1;

fp=fopen("party.dat","rb");

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" DISPLAY PARTY INFORMATION";

gotoxy(18,3);

cout<<"===========================================" ;

printlb2();

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(!feof(fp))

{

gotoxy(20,6);

cprintf("%s",r1.pid);

gotoxy(20,8);

cprintf("%s",r1.pname);

gotoxy(20,10);

cprintf("%s",r1.paddr);

gotoxy(20,12);

cprintf("%s",r1.pcity);

gotoxy(20,14);

cprintf("%s",r1.pstate);

gotoxy(59,6);

cprintf("%s",r1.email);

gotoxy(59,8);

cprintf("%s",r1.mob);

gotoxy(59,10);

cprintf("%s",r1.pdob);

gotoxy(59,12);

cprintf("%s",r1.pgen);

sv1=getch();

if(sv1==27)

{

fclose(fp);

break;

}

printlb2();

}

}

fclose(fp);

}

void esearch()

{

tse:

clrscr();

party r1;

char s1[20],ch,ch1;

int i=0,f=0;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN SEARCH ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb2();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

FILE \*fp;

fp=fopen("party.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(strcmpi(s1,r1.pid)==0)

{

f=1;

gotoxy(20,6); cprintf("%s",r1.pid);

gotoxy(20,8); cprintf("%s",r1.pname);

gotoxy(20,10); cprintf("%s",r1.paddr);

gotoxy(20,12); cprintf("%s",r1.pcity);

gotoxy(20,14); cprintf("%s",r1.pstate);

gotoxy(59,6); cprintf("%s",r1.email);

gotoxy(59,8); cprintf("%s",r1.mob);

gotoxy(59,10); cprintf("%s",r1.pdob);

gotoxy(59,12); cprintf("%.2f",r1.pgen);

}

}

}

fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid Reg. No.");

getch();

goto tse;

}

else

{

gotoxy(20,18);

cprintf("Do u want to search another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto tse;

}

else

{

return;

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

void eupdate()

{

tue:

clrscr();

party r1;

// course r2;

char s1[20],ch,ch1;

// FILE \*fp1;

int i=0,f=0,count;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN UPDATE ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb2();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

count=0;

FILE \*fp;

fp=fopen("party.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

while(!feof(fp))

{

count++;

fread(&r1,sizeof(r1),1,fp);

if(strcmpi(s1,r1.pid)==0)

{

f=1;

i=0;

fclose(fp);

gotoxy(20,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,8);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pname,s1);

//////////////

i=0;

gotoxy(20,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,10);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.paddr,s1);

i=0;

gotoxy(20,12);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,12);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,12);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pcity,s1);

i=0;

gotoxy(20,14);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,14);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,14);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pstate,s1);

i=0;

gotoxy(59,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.email,s1);

i=0;

gotoxy(59,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,8);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.mob,s1);

i=0;

gotoxy(59,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,10);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pdob,s1);

i=0;

gotoxy(59,12);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,12);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,12);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

r1.esal=atof(s1);

strcpy(r1.pgen,s1);

fp=fopen("party.dat","rb+");

count--;

fseek(fp,sizeof(r1)\*count,SEEK\_SET);

fwrite(&r1,sizeof(r1),1,fp);

}

}

}

fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid Teach. No.");

getch();

goto tue;

}

else

{

gotoxy(20,18);

cprintf("Do u want to update another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto tue;

}

else

{

return;

}

}

}

Else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

void edel()

{

tde:

clrscr();

party r1;

char s1[20],ch,ch1;

int i=0,f=0,count;

textcolor(YELLOW);

textbackground(BLACK);

int tn;

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN DELETE OPERATION ";

gotoxy(18,3);

cout<<"===========================================" ;

printlb2();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

return;

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

count=0;

FILE \*fp,\*tp;

fp=fopen("party.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

tp=fopen("mytemp.dat","ab");

while(!feof(fp))

{

count++;

fread(&r1,sizeof(r1),1,fp);

if(!feof(fp))

{

if(strcmpi(s1,r1.pid)!=0)

{

fwrite(&r1,sizeof(r1),1,tp);

}

else

{

f=1;

}

}

}

fclose(fp);

fclose(tp);

// fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid Teach. No.");

remove("mytemp.dat");

getch();

goto tde;

}

else

{

remove("party.dat");

rename("mytemp.dat","party.dat");

gotoxy(20,18);

cprintf("Do u want to Delete another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto tde;

}

else

{

return;

}

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

void printlb4()

{

char dlb[20]={" "};

char ilb[20]={" "};

textcolor(WHITE);

textbackground(GREEN);

gotoxy(3,6);cprintf("%s",ilb);

gotoxy(20,6); cprintf("%s",dlb);

gotoxy(3,8);cprintf("%s",ilb);

gotoxy(20,8); cprintf("%s",dlb);

gotoxy(3,10);cprintf("%s",ilb);

gotoxy(20,10); cprintf("%s",dlb);

gotoxy(42,6);cprintf("%s",ilb);

gotoxy(59,6); cprintf("%s",dlb);

gotoxy(42,8);cprintf("%s",ilb);

gotoxy(59,8); cprintf("%s",dlb);

gotoxy(42,10);cprintf("%s",ilb);

gotoxy(59,10); cprintf("%s",dlb);

gotoxy(3,6);cprintf(" RETAILER ID. :");

gotoxy(3,8);cprintf(" RETAILER NAME :");

gotoxy(3,10);cprintf("RETAILER ADDRESS :");

gotoxy(42,6);cprintf("REATILER CITY :");

gotoxy(42,8);cprintf(" MOB NO. :");

gotoxy(42,10);cprintf(" BOOTH NO :");

gotoxy(20,8);

}

void gentry()

{

sel:

clrscr();

retailer r1;

int cc=0,i,f;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN STOCK ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb4();

char s1[20]={"S00"},ts[20],passw1[30],ch,cnt;

FILE \*fp;

fp=fopen("retailer.dat","rb");

if(fp==NULL)

{

cc=1;

}

else

{

f=0;

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

f++;

}

fclose(fp);

fp=fopen("retailer.dat","rb");

f--;

fseek(fp,sizeof(r1)\*f,SEEK\_SET);

fread(&r1,sizeof(r1),1,fp);

i=0;

while(r1.rno[i+3]!='\0')

{

ts[i]= r1.rno[i+3];

i++;

}

ts[i]='\0';

cc= atoi(ts);

cc++;

}

///////current working area????????????????????????/

fclose(fp);

textcolor(YELLOW);

itoa(cc,ts,10);

strcat(s1,ts);

strcpy(r1.rno,s1);

gotoxy(20,6);

cprintf("%s",r1.rno);

////////////////////????//

i=0;

gotoxy(20,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,8);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.rname,passw1);

//////////////

i=0;

gotoxy(20,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,10);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.radd,passw1);

//gotoxy(45,12);

i=0;

gotoxy(59,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,6);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.rcity,passw1);

i=0;

gotoxy(59,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,8);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.mob,passw1);

i=0;

gotoxy(59,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,10);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

r1.bno=atoi(passw1);

fp=fopen("retailer.dat","ab");

fseek(fp,0,SEEK\_END);

fwrite(&r1,sizeof(r1),1,fp);

fclose(fp);

gotoxy(12,18);cprintf("ONE RECORD SAVED, DO U WANT TO ENTER ANOTHER RECORD Y/N: ");

cnt=getch();

if(cnt=='y'||cnt=='Y')goto sel;

}

void gdisp()

{

clrscr();

char sv1;

int i=1;

FILE \* fp;

retailer r1;

fp=fopen("retailer.dat","rb");

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" DISPLAY GODOWN INFORMATION";

gotoxy(18,3);

cout<<"===========================================" ; gotoxy(2,5);cprintf("RETAILER NO ");

gotoxy(9,5);cprintf("REATILER NAME ");

gotoxy(26,5);cprintf("RETAILER ADDRESS ");

gotoxy(44,5);cprintf("RETAILER CITY ");

gotoxy(62,5);cprintf("MOBILE NO ");

gotoxy(73,5);cprintf("BOOTH NO ");

gotoxy(1,6);

while(i<=80)

{

cprintf("-");

i++;

}

i=7;

textcolor(GREEN);

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(!feof(fp))

{

gotoxy(2,i);

cprintf("%s",r1.rno);

gotoxy(9,i);

cprintf("%s",r1.rname);

gotoxy(26,i);

cprintf("%s",r1.radd);

gotoxy(44,i);

cprintf("%s",r1.rcity);

gotoxy(62,i);

cprintf("%s",r1.mob);

gotoxy(77,i);

cprintf("%d",r1.bno);

i++;

}

}

getch();

fclose(fp);

}

void gsearch()

{

/\* lse:

clrscr();

retailer r1;

char s1[20],ch,ch1;

int i=0,f=0;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN SEARCH ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb4();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

FILE \*fp;

fp=fopen("retailer.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(strcmpi(s1,r1.bno)==0)

{

f=1;

gotoxy(20,6);cprintf("%s",r1.rno);

gotoxy(20,8); cprintf("%s",r1.rname);

gotoxy(20,10); cprintf("%s",r1.radd);

gotoxy(59,6); cprintf("%s",r1.rcity);

gotoxy(59,8); cprintf("%s",r1.mob);

gotoxy(59,10); cprintf("%s",r1.bno);

}

}

}

fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid lib. ID");

getch();

goto lse;

}

else

{

gotoxy(20,18);

cprintf("Do u want to search another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto lse;

}

else

{

return;

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1); \*/

}

void gdel()

{ /\*

tde:

clrscr();

retailer r1;

char s1[20],ch,ch1;

int i=0,f=0,count;

textcolor(YELLOW);

textbackground(BLACK);

int tn;

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN DELETE OPERATION ";

gotoxy(18,3);

cout<<"===========================================" ;

printlb4();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

return;

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

count=0;

FILE \*fp,\*tp;

fp=fopen("retailer.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

tp=fopen("mytemp.dat","ab");

while(!feof(fp))

{

count++;

fread(&r1,sizeof(r1),1,fp);

if(!feof(fp))

{

if(strcmpi(s1,r1.bno)!=0)

{

fwrite(&r1,sizeof(r1),1,tp);

}

else

{

f=1;

}

}

}

fclose(fp);

fclose(tp);

// fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid Teach. No.");

remove("mytemp.dat");

getch();

goto tde;

}

else

{

remove("retailer.dat");

rename("mytemp.dat","retailer.dat");

gotoxy(20,18);

cprintf("Do u want to Delete another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto tde;

}

else

{

return;

}

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1); \*/

}

void printlb5()

{

char dlb[20]={" "};

char ilb[20]={" "};

textcolor(WHITE);

textbackground(GREEN);

gotoxy(3,6);cprintf("%s",ilb);

gotoxy(20,6); cprintf("%s",dlb);

gotoxy(3,8);cprintf("%s",ilb);

gotoxy(20,8); cprintf("%s",dlb);

gotoxy(3,10);cprintf("%s",ilb);

gotoxy(20,10); cprintf("%s",dlb);

gotoxy(3,12);cprintf("%s",ilb);

gotoxy(20,12); cprintf("%s",dlb);

gotoxy(3,14);cprintf("%s",ilb);

gotoxy(20,14); cprintf("%s",dlb);

gotoxy(42,6);cprintf("%s",ilb);

gotoxy(59,6); cprintf("%s",dlb);

gotoxy(42,8);cprintf("%s",ilb);

gotoxy(59,8); cprintf("%s",dlb);

gotoxy(42,10);cprintf("%s",ilb);

gotoxy(59,10); cprintf("%s",dlb);

gotoxy(42,12);cprintf("%s",ilb);

gotoxy(59,12); cprintf("%s",dlb);

gotoxy(3,6);cprintf(" DISTRIBUTOR ID :");

gotoxy(3,8);cprintf(" DISTRIBUTOR NAME :");

gotoxy(3,10);cprintf("DISTRIBUTOR ADD :");

gotoxy(3,12);cprintf(" CITY NAME :");

gotoxy(3,14);cprintf(" DISTRICT :");

gotoxy(42,6);cprintf(" PIN NO. :");

gotoxy(42,8);cprintf(" MOB. NO. :");

gotoxy(42,10);cprintf(" GENDER :");

gotoxy(42,12);cprintf(" EMAIL :");

gotoxy(20,8);

}

void hentry()

{

tel:

clrscr();

distributor r1;

int cc=0,i,f;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN HAWKER DETAIL ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb5();

char s1[20]={"T00"},ts[20],passw1[30],ch,cnt;

FILE \*fp;

fp=fopen("distributor.dat","rb");

if(fp==NULL)

{

cc=1;

}

else

{

f=0;

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

f++;

}

fclose(fp);

fp=fopen("distributor.dat","rb");

f--;

fseek(fp,sizeof(r1)\*f,SEEK\_SET);

fread(&r1,sizeof(r1),1,fp);

i=0;

while(r1.did[i+3]!='\0')

{

ts[i]= r1.did[i+3];

i++;

}

ts[i]='\0';

cc= atoi(ts);

cc++;

}

///////current working area????????????????????????/

fclose(fp);

textcolor(YELLOW);

itoa(cc,ts,10);

strcat(s1,ts);

fclose(fp);

strcpy(r1.did,s1);

gotoxy(20,6);

cprintf("%s",r1.did);

////////////////////????//

i=0;

gotoxy(20,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,8);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dname,passw1);

//////////////

i=0;

gotoxy(20,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,10);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dadd,passw1);

//gotoxy(45,12);

i=0;

gotoxy(20,12);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,12);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,12);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dcity,passw1);

// r1.cfee=atof(passw1);

i=0;

gotoxy(20,14);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,14);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,14);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.ddistt,passw1);

i=0;

gotoxy(59,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,6);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dpin,passw1);

// r1.cfee=atof(passw1);

i=0;

gotoxy(59,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,8);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dmob,passw1);

i=0;

gotoxy(59,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,10);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dmob,passw1);

// r1.cfee=atof(passw1);

i=0;

gotoxy(59,12);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,12);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,12);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

// strcpy(r1.gen,passw1);

// r1.hdm=atof(passw1);

//i=0;

fp=fopen("distributor","ab");

fseek(fp,0,SEEK\_END);

fwrite(&r1,sizeof(r1),1,fp);

fclose(fp);

gotoxy(12,22);cprintf("ONE RECORD SAVED, DO U WANT TO ENTER ANOTHER RECORD Y/N: ");

cnt=getch();

if(cnt=='y'||cnt=='Y')goto tel;

}

void hdisp()

{

clrscr();

char sv1;

FILE \* fp;

distributor r1;

fp=fopen("distributor.dat","rb");

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" DISPLAY DISTRIBUTOR INFORMATION";

gotoxy(18,3);

cout<<"===========================================" ;

printlb5();

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(!feof(fp))

{

gotoxy(20,6);

cprintf("%s",r1.did);

gotoxy(20,8);

cprintf("%s",r1.dname);

gotoxy(20,10);

cprintf("%s",r1.dadd);

gotoxy(20,12);

cprintf("%s",r1.dcity);

gotoxy(20,14);

cprintf("%s",r1.ddistt);

gotoxy(59,6);

cprintf("%s",r1.dpin);

gotoxy(59,8);

cprintf("%s",r1.dmob);

gotoxy(59,10);

cprintf("%s",r1.gen);

gotoxy(59,12);

cprintf("%.2f",r1.email);

sv1=getch();

if(sv1==27)

{

fclose(fp);

break;

}

printlb2();

}

}

fclose(fp);

}

void hsearch()

{

tse:

clrscr();

distributor r1;

char s1[20],ch,ch1;

int i=0,f=0;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN SEARCH ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb5();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

FILE \*fp;

fp=fopen("distributor.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(strcmpi(s1,r1.did)==0)

{

f=1;

gotoxy(20,6); cprintf("%s",r1.did);

gotoxy(20,8); cprintf("%s",r1.dname);

gotoxy(20,10); cprintf("%s",r1.dadd);

gotoxy(20,12); cprintf("%s",r1.dcity);

gotoxy(20,14); cprintf("%s",r1.ddistt);

gotoxy(59,6); cprintf("%s",r1.dpin);

gotoxy(59,8); cprintf("%s",r1.dmob);

gotoxy(59,10); cprintf("%s",r1.gen);

gotoxy(59,12); cprintf("%.2f",r1.email);

}

}

}

fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid Reg. No.");

getch();

goto tse;

}

else

{

gotoxy(20,18);

cprintf("Do u want to search another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto tse;

}

else

{

return;

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

void hupdate()

{

tue:

clrscr();

distributor r1;

// course r2;

char s1[20],ch,ch1;

// FILE \*fp1;

int i=0,f=0,count;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN UPDATE ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb5();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

count=0;

FILE \*fp;

fp=fopen("distributor.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

while(!feof(fp))

{

count++;

fread(&r1,sizeof(r1),1,fp);

if(strcmpi(s1,r1.did)==0)

{

f=1;

i=0;

fclose(fp);

gotoxy(20,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,8);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dname,s1);

//////////////

i=0;

gotoxy(20,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,10);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dadd,s1);

i=0;

gotoxy(20,12);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,12);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,12);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dcity,s1);

i=0;

gotoxy(20,14);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,14);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,14);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.ddistt,s1);

i=0;

gotoxy(59,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dpin,s1);

i=0;

gotoxy(59,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,8);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.dmob,s1);

i=0;

gotoxy(59,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,10);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.gen,s1);

i=0;

gotoxy(59,12);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(59+i,12);

cprintf(" \b");

}

}

else

{

gotoxy(59+i,12);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.email,s1);

fp=fopen("distributor.dat","rb+");

count--;

fseek(fp,sizeof(r1)\*count,SEEK\_SET);

fwrite(&r1,sizeof(r1),1,fp);

}

}

}

fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid Teach. No.");

getch();

goto tue;

}

else

{

gotoxy(20,18);

cprintf("Do u want to update another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto tue;

}

else

{

return;

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

void hdel()

{

tde:

clrscr();

distributor r1;

char s1[20],ch,ch1;

int i=0,f=0,count;

textcolor(YELLOW);

textbackground(BLACK);

int tn;

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN DELETE OPERATION ";

gotoxy(18,3);

cout<<"===========================================" ;

printlb5();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

return;

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

count=0;

FILE \*fp,\*tp;

fp=fopen("distributor.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

tp=fopen("mytemp.dat","ab");

while(!feof(fp))

{

count++;

fread(&r1,sizeof(r1),1,fp);

if(!feof(fp))

{

if(strcmpi(s1,r1.did)!=0)

{

fwrite(&r1,sizeof(r1),1,tp);

}

else

{

f=1;

}

}

}

fclose(fp);

fclose(tp);

// fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid Teach. No.");

remove("mytemp.dat");

getch();

goto tde;

}

else

{

remove("distributor.dat");

rename("mytemp.dat","distributor.dat");

gotoxy(20,18);

cprintf("Do u want to Delete another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto tde;

}

else

{

return;

}

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

void printlb6()

{

char dlb[20]={" "};

char ilb[20]={" "};

textcolor(WHITE);

textbackground(GREEN);

gotoxy(3,6);cprintf("%s",ilb);

gotoxy(20,6); cprintf("%s",dlb);

gotoxy(3,8);cprintf("%s",ilb);

gotoxy(20,8); cprintf("%s",dlb);

gotoxy(3,10);cprintf("%s",ilb);

gotoxy(20,10); cprintf("%s",dlb);

gotoxy(3,6);cprintf(" PRODUCT ID :");

gotoxy(3,8);cprintf(" PRODUCT NAME :");

gotoxy(3,10);cprintf("PRICE :");

gotoxy(20,8);

}

void pentry()

{

tel:

clrscr();

product r1;

int cc=0,i,f;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN product DETAIL ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb6();

char s1[20]={"P00"},ts[20],passw1[30],ch,cnt;

FILE \*fp;

fp=fopen("product.dat","rb");

if(fp==NULL)

{

cc=1;

}

else

{

f=0;

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

f++;

}

fclose(fp);

fp=fopen("product.dat","rb");

f--;

fseek(fp,sizeof(r1)\*f,SEEK\_SET);

fread(&r1,sizeof(r1),1,fp);

i=0;

while(r1.pid[i+3]!='\0')

{

ts[i]= r1.pid[i+3];

i++;

}

ts[i]='\0';

cc= atoi(ts);

cc++;

}

///////current working area????????????????????????/

fclose(fp);

textcolor(YELLOW);

itoa(cc,ts,10);

strcat(s1,ts);

fclose(fp);

strcpy(r1.pid,s1);

gotoxy(20,6);

cprintf("%s",r1.pid);

////////////////////????//

i=0;

gotoxy(20,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,8);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pname,passw1);

//////////////

i=0;

gotoxy(20,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

passw1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,10);

passw1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

r1.p=atof(passw1);

//gotoxy(45,12);

i=0;

gotoxy(20,12);

fp=fopen("product.dat","ab");

fseek(fp,0,SEEK\_END);

fwrite(&r1,sizeof(r1),1,fp);

fclose(fp);

gotoxy(12,22);cprintf("ONE RECORD SAVED, DO U WANT TO ENTER ANOTHER RECORD Y/N: ");

cnt=getch();

if(cnt=='y'||cnt=='Y')goto tel;

}

void pdisp()

{

clrscr();

char sv1;

FILE \* fp;

passbook r1;

fp=fopen("product.dat","rb");

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" DISPLAY Product INFORMATION";

gotoxy(18,3);

cout<<"===========================================" ;

printlb6();

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(!feof(fp))

{

gotoxy(20,6);

cprintf("%s",r1.pid);

gotoxy(20,8);

cprintf("%s",r1.pname);

gotoxy(20,10);

cprintf("%f",r1.p);

sv1=getch();

if(sv1==27)

{

fclose(fp);

break;

}

printlb2();

}

}

fclose(fp);

}

void psearch()

{

tse:

clrscr();

product r1;

char s1[20],ch,ch1;

int i=0,f=0;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN SEARCH ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb6();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

FILE \*fp;

fp=fopen("product.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

while(!feof(fp))

{

fread(&r1,sizeof(r1),1,fp);

if(strcmpi(s1,r1.pid)==0)

{

f=1;

gotoxy(20,6); cprintf("%s",r1.pid);

gotoxy(20,8); cprintf("%s",r1.pname);

gotoxy(20,10); cprintf("%f",r1.p);

}

}

}

fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid passbook No.");

getch();

goto tse;

}

else

{

gotoxy(20,18);

cprintf("Do u want to search another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto tse;

}

else

{

return;

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

void pupdate()

{

tue:

clrscr();

passbook r1;

// course r2;

char s1[20],ch,ch1;

// FILE \*fp1;

int i=0,f=0,count;

textcolor(YELLOW);

textbackground(BLACK);

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN UPDATE ENTRY";

gotoxy(18,3);

cout<<"===========================================" ;

printlb6();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

count=0;

FILE \*fp;

fp=fopen("product.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

while(!feof(fp))

{

count++;

fread(&r1,sizeof(r1),1,fp);

if(strcmpi(s1,r1.pid)==0)

{

f=1;

i=0;

fclose(fp);

gotoxy(20,8);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,8);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,8);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.pname,s1);

//////////////

i=0;

gotoxy(20,10);

do{

ch=getch();

if(ch==27)

{

return;

}

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

break;

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,10);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,10);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

strcpy(r1.acno,s1);

fp=fopen("product.dat","rb+");

count--;

fseek(fp,sizeof(r1)\*count,SEEK\_SET);

fwrite(&r1,sizeof(r1),1,fp);

}

}

}

fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid product ID.");

getch();

goto tue;

}

else

{

gotoxy(20,18);

cprintf("Do u want to update another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto tue;

}

else

{

return;

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

}

void pdel()

{

tde:

clrscr();

product r1;

char s1[20],ch,ch1;

int i=0,f=0,count;

textcolor(YELLOW);

textbackground(BLACK);

int tn;

clrscr();

gotoxy(20,2);

cout<<" WELCOME IN DELETE OPERATION ";

gotoxy(18,3);

cout<<"===========================================" ;

printlb6();

gotoxy(20,6);

do{

ch=getch();

if(ch==27)

return;

else

{

if(ch==13 || i==19)

{

s1[i]='\0';

count=0;

FILE \*fp,\*tp;

fp=fopen("product.dat","rb");

if(fp==NULL)

{

gotoxy(31,18);

cprintf("Data not available");

getch();

return;

}

else

{

tp=fopen("mytemp.dat","ab");

while(!feof(fp))

{

count++;

fread(&r1,sizeof(r1),1,fp);

if(!feof(fp))

{

if(strcmpi(s1,r1.pid)!=0)

{

fwrite(&r1,sizeof(r1),1,tp);

}

else

{

f=1;

}

}

}

fclose(fp);

fclose(tp);

// fclose(fp);

///////current working area????????????????????????/

if(f==0)

{

gotoxy(20,18);

cprintf("Record not found, Pls enter valid passbook No.");

remove("mytemp.dat");

getch();

goto tde;

}

else

{

remove("product.dat");

rename("mytemp.dat","product.dat");

gotoxy(20,18);

cprintf("Do u want to Delete another record Y/N: ");

ch1=getch();

if(ch1=='Y'||ch1=='y')

{

goto tde;

}

else

{

return;

}

}

}

}

else

{

if(ch=='\b')

{

if(i!=0)

{

i--;

gotoxy(20+i,6);

cprintf(" \b");

}

}

else

{

gotoxy(20+i,6);

s1[i]=ch;

cprintf("%c",ch);

i++;

}

}

}

}while(1);

**Testing Approach**

Computer s/w has become more complex. The need for specialized testing approaches has also grown. The "white box “and "black box testing “methods are applicable across all environments.

GUI presents integrity challengers for software engineers. Because of reusable components provided, the creation of the user interface has become less time consuming and more process. But at the same time the completely of GUI"s has grown, leading to more difficulty is the design and execution of test crises. Finites step modeling graphs man be used to drive a series of test that address specific data and programs objects that are relevant to the GUI. Due to the large no. of permutation associated with GUI operations testing should be approached using automated fools.

**Verification and validation:-**

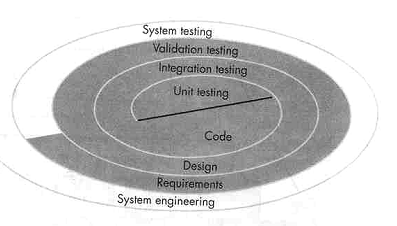
Verification refers to the set of activity that insures that software correctly implements a specific function. Validation refers to a different set of activities that insure that the software that has built that is traceable customer requirements.

In other words validation provides final assurance that software meets all functional, behavioral and performance requires.

"Boebam" states another way:-

* + - * + Verification: - "Are we building the product right".
        + Validation: - "Are we building the right product".

Verification and validation encompasses a wide array of SQA (software quality assurance) activities that include formal technical review, quality and configuration audits, performance monitoring, simulation, fusibility study, documentation review, development testing, qualification testing and installation testing. Software testing strategy:-

A strategy for software testing may also be viewed in the context of the spiral. 

Unit testing is begins at the vortex of the spiral and concentration on each units (component) of he software as implemented in source code. Testing programs by moving outward along the spiral to integrated testing, where the focus is on design and construction of the software architecture. Taking another term out word on the spiral be in countered validation testing, where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive other at system testing where the s/w , we special out along stream lines that we spiral the scope of testing with each turned.

**Unit testing:-**

Unit testing focus verification afford on the smallest unit of the software design-the software component/module. Using component level design description as a guide, important control paths are tested o uncover within the boundary of the module. The unite test is white box printed. The test that occurs as parts of unit tests is illustrated schematically.

The module interface is tested to insure that information properly follows into and out of the program unit under test. The local data structure is examined to insure that data stored temporarily maintains its integrity during all steps in an algorithms execution. Boundary condition are tested to insure that the module operates properly at boundaries established to limit or restrict processing. All independent paths through the control structure are executed at least once. And finally all error handling paths are tested.

**Integration testing:-**

Integration testing is a systemic technique for contracting the program structure while at the same time conducting tests to uncover error associated with interfacing. A number of different incremental integration strategies are:-

Components are combining to form clusters 0, 1, 2, & 3.Each of the cluster 1&2 subordinate to "Ma". Similarly driver "D3" for cluster "3" is removed prior to integration with module "Mb". Both "Ma" and "Mb" will ultimately bee integrated with component "Mc" and forth. Integration moves upward.

**Validation testing:-**

Software validation is achieved through a series of "black box" tests that demonstrates conformity with requirements. Each validation test case has been conducted, one of two possible conditions exists:-

* The functions or performance characteristics conform to specification and are accepted.
* A deviation from specification is uncovered and a deficiency list is created. Deviation or error discovered at this stage in a project can recovery be corrected prior to schedule delivery.

There are two types of validation testing are:-

* + - * + Alpha testing
        + Beta testing

A. alpha testing:-

The alpha testis conducted at the developer's site by a customer. The software is used in natural setting with the developer “looking over and shoulder" of the user and recording errors and using problems. Alpha test are conducted in a control environment.

B. Beta testing:-

Beta testing is conducted at one or more costumer site by the end user of the software. Unlike alpha testing, the developer is generally not present. Therefore the beta test is a "live" application of the software in an environment that cannot be controlled by the developer. The customer records all problems that are countered during beta testing and reports these to the developer at regular intervals. As a result of problems reported during beta tests, software engineers make modifications and then prepare for relation of the software product to the enter customer based.

System testing:-

System testing is actually a series of different tests who's primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work of verify that system elements have been property integrated and perform allocated functions.

* Recovery Testing:-

Recovery testing is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed. If recovery is automatic (perform by the system itself) reutilization, check pointing mechanism, data recovery and restart are evaluated for correctness. If repair (MTTR) is evaluated to determine whether it is within acceptation limits.

* Security Testing:-

Security testing attempts to verify that protection mechanism built into a system will infect, protect it from improper penetration. During security testing the tester plays the role of the individual who desire to penetrate the system. The role of the system designer is to make penetration cost more than the value of the information that will be obtained.

Debugging Process:-

Debugging occurs as a consequence of successful testing. That is, when a test case uncovers and error, debugging is the process that results in the removal of the error.

Integration Testing

Unit Testing

Debugging

The debugging process will always have one of two outcomes:-

1. The cause will be found and corrected.

2. The cause will not be found.

The person performing debugging may suspect a cause, esign a test case to help validate the suspicion, and work toward error correction in an iterative fashion.

**Software maintenance:-**

Software maintenance is of course for more than "fixing mistakes”. We may define maintains by describing for a activates that are under ken after a program is released for use. For different maintained activates are:-

**1. Corrective maintains:-**

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective main tense changes the software to correct defects.

**2. Adaptive Maintains:-**

Overtime, the original environment (CPU, OS, business rules, external product characteristics) for which the software’s was developed in likely to change. Adaptive maintains result in modification to the software to accommodate change to its external environment.

**3. Perfective maintains or enhancement maintence:-**

As software is used, the customer/user will recognize additional function that will provide benefit. Perfective maintaince extends the software beyond its original function requirement.

**4. Perfective maintains/Reengineering:-**

Computers software deteriorates due to change, and because of this, preventive maintaince, often called software reengineering, and must be conducted to enable the software to serves the needs of its end users. In essence, preventive maintains makes change to computer programs so that they can be more easily corrected, adapted and enhanced.

**5. Installation testing:-**

Implementation means to take into practice. A crucial phase in the system line cycle is the successful implementation of the new system design. Implementation includes all those activities that takes place to convert or system or automated system.

The software of **DAIRY BILLING SYSTEM** completely new system i.e. the existing system is manual.

**The main aspects of implementation are as follows:-**

* Training personnel
* Conversion procedure.
* Demonstration.

For the proposed system DAIRY BILLING SYSTEM will be beneficial. Under this approach, users continue to operate the old system in the usual manner but they also start using the new system. This method is safest one because it ensures that in case of any problem in using the new system, the organization can still fall back to the old system without loss of time and money.

**Modifications and Improvements:-**

The title of employee training system can make the project during different types of modification when compile the program then different types come of error and that error read and it correct debug and then compile the program. Project during compile and run to again then when correct the program so, I check the fill data and different types of validation creating during the project.

**LIMITATIONS OF THE PROJECT**

The Software is developed for only window based environment. Actually not only Windows environment is sufficient rather Windows 98 onward version is needed, because here I am going to use JAVA as front end and Oracle edition as Backend, which doesn't support earlier version of windows.

**Minimum Hardware Requirement**

1. **Computer system**: Pentium Based system with at least 2.4 GHz speed or Higher

2. **RAM**: At least 256 MB, it is recommended to have 512 MB of RAM for fast processing on data.

3. **Hard Disk Space**: At least 10 GB of free space, it is recommended to have 20 GB of free space for data storage.

**Minimum Software Requirement**

1. **Operating System**  : Windows XP or Higher.

2. **DATABASE Package** : C++ FILE HANDLING

3. **FRONT**-END : C++

**Others**

This version will not work on a **network system** as this is developed for a single system because in the industry there is only one computer available for handling those tasks. if, this software is to be implemented on a network based organisation then some modified will be necessary.

**SCOPE OF FUTURE APPLICATION**

for use in other NALANDA DAIRY FARM, BIHAR SHARIF is a place which deals each and every product made up milk. Anyone who wants to business with Raj product it is so simple. Anuj dairy play very crucial role in the field of milk business in the market. To manage Anuj dairy manually is a complex job. an automated "Dairy Billing System" is therefore needed for every industry to manage the transaction of product with party of industry.

The project can be used in any industry with some minor modification. These modifications may be of both types- either hardware or software limitation or additional requirement made by the staffs of the industry.

There is always room for further enhancement of this system in the following areas.

* + Hierarchy modification / additional capabilities are inbuilt in the system.
  + Dynamic screens according to requirement can be introduced any time.
  + System is very flexible for further modifications.
  + With very little modification, it can be deployed industry system.

**BIBLIOGRAPHY**

1. MASTERING IN C++

2. SYSTEM ANALYSIS AND DESIGN - E.M.AWARD

3. COMPLETE REFERENCE