**INTRODUCTION**

Mess Minder is a static software which is designed to help the owner of a mess to effectively and efficiently administrate their business. This system will automate administrative tasks such as billing, accounting, members tracking, etc. This can save the admin a lot of time and effort that would have otherwise been spent on manual tasks. This system will also help them to reduce the risk of errors that can occur with manual data entry. This means that the admin can rely on accurate data when making decisions and thus helping the admin to focus on more important tasks such as managing the quality of the food and the overall experience of the members. Mess management systems can help the admin to manage finances more effectively. The system can generate reports that show expenses, revenues, and profits. This information can help the admin to make informed decisions about pricing and menu offerings. Overall, a mess management system can help the admin to run the mess more efficiently and effectively, allowing them to focus on providing the best possible experience for the members.

This software is a solution to the problem which arises due to the existing systems. Currently mess providers are manually entering the records either on a paper or in an Excel sheet.

* **Human error**: Manual entry of member data is prone to human error. Mistakes can be made in recording member names, phone numbers, room numbers, and meal preferences. This can lead to confusion and errors in billing and food service.
* **Time-consuming**: Manual entry of member data is time-consuming and requires a significant amount of effort. Admins and staff members may spend a considerable amount of time maintaining and updating records, taking away time that could be spent on more important tasks.
* **Difficulty in tracking information**: Manual entry of member data can make it difficult to track information. For example, it can be challenging to keep track of dietary restrictions or to monitor meal attendance accurately. This can lead to mistakes in billing and food service.
* **Lack of security**: Manual entry of member data can compromise the security of the information. Paper records can be misplaced or stolen, and data stored in spreadsheets or other electronic documents can be accessed by unauthorized individuals.

**1. INITIATION PHASE**

The project initiation phase is the first phase in the project management life cycle, as it involves starting up a new project. In this phase we define the objectives, scope, purpose and deliverables to be produced. We also hire your project team, setup the project office and review the project, to gain approval to begin the next phase.

**1.1 Problem Definition**

The current method of administrating a mess involves manual data entry and management depends upon pen and paper, in a long run this is not an efficient way to manage a mess. The issue is that billing becomes hard for people who only wants to take mess for a specific number of times e.g., 19 days. Sometimes records are not maintained properly or one can change the records as the log can be accessed by anyone.

**1.2 Scope of the Project**

* The main purpose of the project is to automate the process of administrating a mess and replace the current method of manual data entry.
* To make the calculation of monthly or day wise bill easy.
* To main a proper record of sales and thus help in decision making in the process of creating a new menu.
* To keep a record of successful transactions and also of the dues.
* To keep a record of the members, their product choice and payment status.

**1.3 Modules**

**Member module**: Member module will display the details of the people who are taking mess facility from the mess. It will contain their name , contact information , mail id and an auto generated id will be provided to them.

**Product module**: The product is of three types, breakfast, lunch and dinner. Their will different menu for each of them and the details of their price and an id will be provided.

**Payment module**: This module will contain the details of the member their choice of product, the amount which they have to pay and the status and method of payment.

**Report module**: This module will contain the records of the members and also of the mess’s expenses, sales and profits.

**1.4 Software Platform**

* Operating System – Windows 11
* Programming Language – Visual Basic.net
* Database – My SQL
* Tools – Microsoft project, Microsoft Visual Studio 2022, Ms Word.

**1.5 Hardware Platform**

* Processor – Intel i5 10th generation
* RAM – 16GB
* System type – 64-bit Operating System.
* Solid State Drive – 256 GB.

**2. SYSTEM STUDY**

**2.1 Study of existing system**

The existing system is the use of paper or an excel sheet. The records are maintained by manual data entry. This system is neither effective nor efficient. Tracking previous records is not easy in such systems, overall, the system is time consuming. Some of the mess use excel sheet for the same but it is not very efficient and also requires training to use it effectively as people working in a mess are now aware of these technologies.

**How existing system works**

* The existing system maintains record of member’s name, date of payment and the amount which is to be paid.

**Disadvantages of existing system**

* Had to keep a track of members who has not made the payment.
* Can be accessed by anyone and thus can be payment details can be changed.
* Keeping track of previous record is time consuming and the data can be lost if the papers are lost.

**2.2 Feasibility Study**

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and schedulingfactors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it. A feasibility study tests the viability of an idea, a project or even a new business. The goal of a feasibility study is to place emphasis on potential problems that could occur if a project is pursued and determine if, after all significant factors are considered, the project should be pursued. Feasibility studies also allow a business to address where and how it will operate, potential obstacles, competition and the funding needed to get the business up and running.

**Components of Feasibility study**:

• Technical Feasibility

• Economic Feasibility

• Operational Feasibility

• Schedule Feasibility

Feasibility studies are almost always conducted where large sums are at stake. Also called as feasibility analysis. Every project is feasible for given unlimited resources and infinitive time. Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. Feasibility and risk analysis are related in many ways. The feasibility analysis in this project has been discussed below based on the above-mentioned components of feasibility.

**Technical feasibility**

Technical feasibility canters around the technology used. It means the computerized system is technically feasible i.e., it doesn’t have any technical fault and work properly in the given environment. Our system is technically feasible it is providing us the required output.

**Economic feasibility**

Economic analysis is the most frequently used method for evaluating the effectiveness of the computerized system. We analyse the computerized system is feasible as than the manual system because it saves the money, time and manpower. It is also feasible according to cost benefit analysis.

**Operational feasibility**

• Human effort or Manual Labour can be reduced drastically.

• Major operations that are done manually can be done within a matter seconds.

• Data storing is easier and backup is available.

## Schedule feasibility

Time evaluation is the most important consideration in the development of project. The time schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development of other systems.

## Legal feasibility

Determines whether the proposed system conflicts with legal requirements, e.g., a data processing system must comply with the local data protection regulations and if the proposed venture is acceptable in accordance to the laws of the land.

**Operational feasibility**

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters such as reliability, maintainability, supportability, usability, disposability, sustainability, affordability and others.

These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realized. A system design and development require appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

* 1. **PROPOSED SYSTEM**

The disadvantages of the existing system have been resolved by developing this software which is much more efficient and less time consuming. It also has multiple useful features which makes the overall user experience more efficient and effective.

* It helps to manage the entire record of the members, products, payment details and also the sales record, expenses and profit details.
* The records are generated based on the payment and dues.
* The system is maintainable and updateable.
* The interface is user friendly and no prior training is required to use it.
* The system is thus, simple to use and easy to learn.
* The proposed system is supposed to help the admin to facilitate the ease of tracking the payments, product and members details.
* The system also helps to maintain a record of the product choices and allergies if any.
* Saves much more time and the user can manage the services at their convenience and availability of time.
* The proposed system is fast and is also secured and reliable as it can only be accessed only by the admin credentials.

**3. SYSTEM DESIGN**

In the design phase the architecture is established. This phase starts with the requirement document delivered by the requirement phase and maps the requirements into an architecture. The architecture defines the components, their interfaces and behaviors. The deliverable design document is the architecture. The design document describes a plan to implement the requirements. This phase represents the ``how'' phase

Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established.

The design may include the usage of existing components. Analyzing the trade-offs of necessary complexity allows for many things to remain simple which, in turn, will eventually lead to a higher quality product. The architecture team also converts the typical scenarios into a test plan. In our approach, the team, given a complete requirement document, must also indicate critical priorities for the implementation team.

A critical implementation priority leads to a task that has to be done right. If it fails, the product fails. If it succeeds, the product might succeed. At the very least, the confidence level of the team producing a successful product will increase. This will keep the implementation team focused. Exactly how this information is conveyed is a skill based on experience more than a science based on fundamental foundations. System design is the process of defining the architecture components, modules, interfaces, and data for a system to satisfy specified requirements.

Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

Until the 1990s, systems design had a crucial and respected role in the data processing industry. In the 1990s, standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software enginee[ring.](https://en.wikipedia.org/wiki/Software_engineering)

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modelling software systems and is increasingly used for high designing non software systems and organizations.

### 

### **ARCHITECTURAL DESIGN:**

The architectural design of a system emphasizes the design of the system architecture that describes the structure, behavior and more views that system and analysis.

**LOGICAL DESIGN:**

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams).

### **PHYSICAL DESIGN:**

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

* Input requirements
* Output requirements
* Storage requirements
* Processing requirements
* System control and backup or recovery

Put another way, the physical portion of system design can generally be broken down into three sub- tasks,

* User Interface Design
* Data Design
* Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system.

Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system.

At the end of the system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase. Physical design, in this context, does not refer to the tangible physical design of an information system.

To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc.

It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

# **E-R DIAGRAM:**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique. Because this ER tutorial focuses on beginners.

Below are some tips that will help you build effective ER diagrams:

* Identify all the relevant entities in a given system and determine the relationships among these entities.
* An entity should appear only once in a particular diagram.
* Provide a precise and appropriate name for each entity, attribute, and relationship in the diagram. Terms that are simple and familiar always beats vague, technical-sounding words. In naming entities, remember to use singular nouns. However, adjectives may be used to distinguish entities belonging to the same class (part-time employee and full time employee, for example). Meanwhile attribute names must be meaningful, unique, system-independent, and easily understandable.
* Remove vague, redundant or unnecessary relationships between entities.
* Never connect a relationship to another relationship.
* Make effective use of colours. You can use colours to classify similar entities or to highlight key areas in your diagrams.

**The History of Entity Relationship Diagrams**

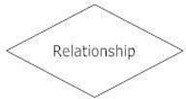
Peter Chen developed ERDs in 1976. Since then, Charles Bachman and James Martin have added some slight refinements to the basic ERD principles.

An entity relationship diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

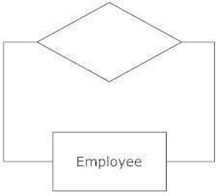
**Entity**: An entity is an object or concept about which you want to store information. A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

**Actions**: which are represented by diamond shapes, show how two entities share information in the database. In some cases, entities can be self-linked. For example, employees can supervise other employees.

A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

Actions, which are represented by diamond shapes, show how two entities share information in the database.

In some cases, entities can be self-linked. For example, employees can supervise other employees.



**Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



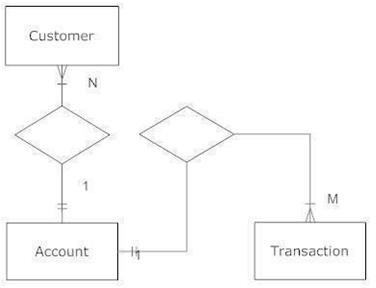
A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.

A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

**Connecting lines**, solid lines that connect attributes to show the relationships of entities in the ER Diagram.

**Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.

There are many notation styles that express cardinality .



## DATA FLOW DIAGRAM (LEVEL 0 AND LEVEL 1):

The Data Flow Diagrams (DFDs) are used for structure analysis and design. DFDs show the flow of data from external entities into the system. DFDs also show how the data moves and is transformed from one process to another, as well as its logical storage. The following symbols are used within DFDs. For clarity, a key has been provided at the bottom of this page.

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

### **HISTORY:**

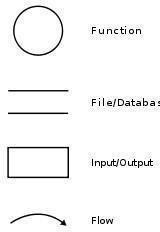
[Larry Constantine,](https://en.wikipedia.org/wiki/Larry_Constantine) the original developer of structured design, based on Martin and Estrin's "Data Flow Graph" model of computation.

Starting in the 1970s, data flow diagrams (DFD) became a popular way to visualize the major steps and data involved in software system processes. DFDs were usually used to show data flow in a computer system, although they could in theory be applied to [business process modelling. D](https://en.wikipedia.org/wiki/Business_process_modeling)FD were useful to document the major data flows or to explore a new high-level design in terms of data flow.

## THEORY:



**DATA FLOW DIAGRAM EXAMPLE:**



The Data Flow Diagrams (DFDs) are used for structure analysis and design. DFDs show the flow of data from external entities into the system. DFDs also show how the data moves and is transformed from one process to another, as well as its logical storage. The following symbols are used within DFDs. For clarity, a key has been provided at the bottom of this page.

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

Data flow diagrams are one of the three essential perspectives of the structured-systems analysis and design method. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented.

The old system's dataflow diagrams can be drawn up and compared with the new system's data flow diagrams to draw comparisons to implement a more efficient system. Data flow diagrams can beused to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram model.

In the course of developing a set of levelled data flow diagrams the analyst/designer is forced to address how the system may be decomposed into component sub-systems, and to ide[ntify the transaction data](https://en.wikipedia.org/wiki/Transaction_data) [in t](https://en.wikipedia.org/wiki/Transaction_data)he [data model.](https://en.wikipedia.org/wiki/Data_model)

Data flow diagrams can be used in both Analysis and Design phase of the [SDLC.](https://en.wikipedia.org/wiki/Systems_development_life_cycle)

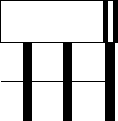
There are different notations to draw data flow diagrams (Yourdon & Coad and [Gane](https://en.wikipedia.org/wiki/Chris_Gane_%28computer_scientist%29) [&](https://en.wikipedia.org/wiki/Chris_Gane_%28computer_scientist%29) [Sarson),](https://en.wikipedia.org/wiki/Trish_Sarson) defining different visual representations for processes, data stores, data flow, and external entities.

## PHYSICAL VS LOGICAL DFD:

A logical DFD captures the data flows that are necessary for a system to operate. It describes the processes that are undertaken, the data required and produced by each process, and the stores needed to hold the data. On the other hand, a physical DFD shows how the system is actually implemented, either at the moment (Current Physical DFD), or how the designer intends it to be in the future (Required Physical DFD).

Thus, a Physical DFD may be used to describe the set of data items that appear on each piece of paper that move around an office, and the fact that a particular set of pieces of paper are stored together in a filing cabinet. It is quite possible that a Physical DFD will include references to data that are duplicated, or redundant, and that the data stores, if implemented as a set of [database tables, w](https://en.wikipedia.org/wiki/Database_table)ould constitute an un-normalized (or denormalized) relational database. In contrast, a Logical DFD attempts to capture the data flow aspects of a system in a form that has neither redundancy nor duplication.

## DATA FLOW SYMBOLS AND THEIR MEANINGS: -



An entity. A source of data or a destination for data.

**Source/Sink:** Represented by rectangles in the diagram. Sources and Sinks are external entities which are sources or destinations of data, respectively.



**Process:** Represented by circles in the diagram. Processes are responsible for manipulating the data. They take data as input and output an altered version of the data.



**Data Store:** Represented by a segmented rectangle with an open end on the right. Data Stores are both electronic and physical locations of data. Examples include databases, directories, files, and even filing cabinets and stacks of paper.



In our course, we need to understand and be able to draw 2 types of Data Flow Diagrams, they are Level-0 and Level 1 DFD’s. In this blog, I will hopefully make it easier to understand the differences between the two types of DFD’s and help understand how to draw a DFD for the exam.

Firstly, we will look at level-0 DFD’s and give an example. Then we will look at Level 1 DFD’s and

give an example.

A level-0 DFD is the most basic form of DFD. It aims to show how the entire system works at a glance. There is only one process in the system and all the data flows either into or out of this process. Level- 0 DFD’s demonstrates the interactions between the process and external entities. They do not contain Data Stores.

When drawing Level-0 DFD’s, we must first identify the process, all the external entities and all the data flows. We must also state any assumptions we make about the system. It is advised that we draw the process in the middle of the page. We then draw our external entities in the corners and finally connect our entities to our process with the data flows.

Below is an example of a Level-0 DFD.

# **3.3 GANTT CHART:**

A Gantt chart is a type of bar chart, devised by Henry Gantt in the 1910s, that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project.

Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities. Gantt charts can be used to show current schedule status using percent-complete shadings and a vertical "TODAY" line as shown here.

Although now regarded as a common charting technique, Gantt charts were considered revolutionary when first introduced. This chart is also used in information technology to represent data that has been collected.

## HISTORICAL DEVELOPMENT:

The first known tool of this type was developed in 1896 by Karol Adamiecki, who called it a harmonogram , Adamiecki did not publish his chart until 1931, however, and only in Polish, which limited both its adoption and recognition of his authorship.

The chart is named after Henry Gantt (1861– 1919), who designed his chart around the years 1910– 1915.

One of the first major applications of Gantt charts was by the United States during World War I, at the instigation of General William Crozier

In the 1980s, personal computers allowed widespread creation of complex and elaborate Gantt charts. The first desktop applications were intended mainly for project managers and project schedulers.

With the advent of the Internet and increased collaboration over networks at the end of the 1990s, Gantt charts became a common feature of web-based applications, including collaborative groupware.

## GANTT CHART BENEFITS:

**CLARITY:** One of the biggest benefits of a [Gantt chart is](http://www.brighthubpm.com/templates-forms/3418-using-excel-to-create-a-gantt-chart/) the tool's ability to boil down multiple tasks and timelines into a single document. Stakeholders throughout an organization can easily understand where teams are in a process while grasping the ways in which independent elements come together toward project completion.

**COMMUNICATION:** Teams can use Gantt charts to replace meetings and enhance other status updates. Simply clarifying chart positions offers an easy, visual method to help team members understand task progress.

**MOTIVATION:** Some teams or team members become more effective when faced with a form of external motivation. Gantt charts offer teams the ability to focus work at the front of a task timeline, or at the tail end of a chart segment. Both types of team members can find Gantt charts meaningful as they plug their own work habits into the overall project schedule.

**CO ORDINATION:** For project managers and resource schedulers, the benefits of a Gantt chart include the ability to sequence events and reduce the potential for overburdening team members. Some project managers even use combinations of charts to break down projects into more manageable sets of tasks.

**CREATIVITY:** Sometimes, a lack of time or resources forces project managers and teams to find creative solutions. Seeing how individual tasks intertwine on Gantt charts often encourages new partnerships and collaborations that might not have evolved under traditional task assignment systems.

**TIME MANAGEMENT**: Most managers regard scheduling as one of the major benefits of Gantt charts in a creative environment. Helping teams understand the overall impact of project delays can foster stronger collaboration while encouraging better task organization.

**FLEXIBILITY:** Whether you [use Excel to generate Gantt charts or](http://www.brighthubpm.com/templates-forms/31029-a-microsoft-excel-project-planning-form/) you load tasks into a more precise chart generator, the ability to issue new charts as your project evolves lets you react to unexpected changes in [project scope or](http://www.brighthubpm.com/project-planning/10057-defining-the-project-scope-tips-amp-free-template/) timeline. While revising your project schedule too frequently can eliminate some of the other benefits of Gantt charts, offering a realistic view of a project can help team members recover from setbacks or adjust to other changes.

**MANAGEABILITY:** For project managers handling complex assignments, like software publishing or event planning, the benefits of Gantt charts include externalizing assignments. By visualizing all

of the pieces of a project puzzle, managers can make more focused, effective decisions about resources and timetables.

**EFFICIENCY:** Another one of the benefits of Gantt charts is the ability for teams members to leverage each other’s deadlines for maximum efficiency. For instance, while one team member waits on the outcome of three other tasks before starting a crucial piece of the assignment, he or she can perform other project tasks. Visualizing resource usage during projects allows managers to make better use of people, places, and things.

**ACCOUNTABILITY:** When project teams face major organizational change, documenting effort and outcomes becomes crucial to career success. Using Gantt charts during critical projects allows both project managers and participants to track team progress, highlighting both big wins and major failures. During professional review periods, team members who frequently exceed expectations can leverage this documentation into larger raises or bonuses.

## GANTT CHART IMPORTANCE:

The project's summary and terminal elements, which combine to form the project's internal structure, are shown on the Gantt chart. Many charts will also depict the precedence rankings and dependencies of various tasks within the project. The charts can illustrate the start and finish project terminal elements in project management.

It can also show summary elements and terminal dependencies. The smallest task tracked as part of the project effort is known as a terminal element. Gantt chart represents the tasks in most modern project scheduling packages. However other management applications use simpler communication tools such as message boards, to-do lists and simple scheduling etc., therefore, they do not use Gantt charts as heavily.

The way to create this chart begins by determining and listing the necessary activities. Next, sketch out how you expect the chart to look. List which items depend on others and what activities take place when. For each activity, list how many man-hours it will require, and who is responsible.

Lastly, determine the throughput time. This technique's primary advantage is its good graphical overview that is easy to understand for nearly all project participants and stakeholders. Its primary disadvantage is its limited applicability for many projects, since projects are often more complex than can be effectively communicated with this chart.

Here with is the Gantt chart for MESS MINDER.

Codesssssssssssssssssssssss

**4.SYSTEM CONFIGURATION:**

## 

## HARDWARE REQUIREMENTS

PROCESSOR : INTEL CORE SPEED : 1.50GHZ

RAM : 4GB

## SOFTWARE REQUIREMENTS:-

FRONTEN: VB.NET (VISUALSTUDIO)2022

BACKEND: MySQL SERVER DOCUMENTATON: MS WORD 2007 GANTT CHART: MS PROJECT 2013

OPERATING SYSTEM : WINDOWS 11

# **5. DETAILS OF SOFTWARE:**

# **5.1 OVERVIEW OF FRONT-END**

Microsoft Visual Studio 2022 is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio supports different programming languages and allows the code editor and debugger to support nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, Visual C++ and VB.NET. Support for other languages such as Python, Ruby, Node.js, and M among others is available via language services installed separately.

It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Java (and J#) were supported in the past**.** Microsoft provides a free version of Visual Studio called the Community edition that supports plugins and is available at no cost for all users. Support for programming languages isadded by using a specific VS Package called a Language Service.

A language service defines various interfaces which the VS Package implementation can implement to add support for various functionalities. Functionalities that can be added this way include syntax coloring, statement completion, brace matching, parameter information tooltips, member lists and error markers for background compilation.

If the interface is implemented, the functionality will be available for the language. Language services are implemented on a per-language basis. The implementations can reuse code from the parser or the compiler for the language. Language services can be implemented either in native code or managed code. For native code, either the native COM interfaces or the Babel Framework can be used. For managed code, the MPF includes wrappers for writing managed language services.

## FEATURES:

Boolean Conditions Automatic Garbage Collection Standard Library

Assembly Versioning Properties and Events

Delegates and Events Management Easy-to-use Generics

Indexers

Conditional Compilation Simple Multithreading

## ADVANTAGES:

The structure of the [Basic programming language is](http://en.wikipedia.org/wiki/BASIC) very simple, particularly as to the executable code.

VB is not only a language but primarily an integrated, interactive development environment

(“[IDE“).](http://en.wikipedia.org/wiki/Integrated_development_environment)

The VB-IDE has been highly optimized to support rapid application development(“RAD”). It is particularly easy to develop [graphical user interfaces](http://en.wikipedia.org/wiki/Graphical_user_interface) [an](http://en.wikipedia.org/wiki/Graphical_user_interface)d to connect them to handler functions provided by the application.

The graphical user interface of the VB-IDE provides intuitively appealing views for the management of the program structure in the large and the various types of entities (classes, modules, procedures, forms, …).

VB provides a comprehensive interactive and context-sensitive online help system.

When editing program texts the “[IntelliSense” t](http://en.wikipedia.org/wiki/IntelliSense)echnology informs you in a little popup window about the types of constructs that may be entered at the current cursor location.

VB is a component integration language which is attuned to Microsoft’s Component Object Model (“COM”).

COM components can be written in different languages and then integrated using VB.

Interfaces of COM components can be easily called remotely via Distributed COM (“DCOM”), which

makes it easy to construct distributed applications.

COM components can be embedded in / linked to your application’s user interface and also in/to

stored documents ([Object Linking and Embedding](http://en.wikipedia.org/wiki/Object_Linking_and_Embedding) “OLE”, “Compound Documents”). There is a wealth of readily available COM components for many different purposes. Visual Basic is built around the .NET environment used by all Microsoft

# **5.1 OVERVIEW OF BACK-END:**

MySQL is licensed under the GNU General Public License and is also available under several proprietary licenses. When Oracle bought MySQL AB in 2010, Michael "Monty" Widenius, MySQL founder, forked MySQL into a free, open source database called [MariaDB](https://www.openlogic.com/blog/mariadb-overview) -- with the intention of keeping the MariaDB project free and open source forever.

MySQL has several versions available, but there are essentially two options: a community version, which is free to use; and paid versions, which include additional functionality, extensions, and support through Oracle. Despite the branding for the paid version, the community version is still considered to be production-ready and is often used in the enterprise. MySQL is a highly scalable product and that scalability can come from several different performance tuning techniques. For starters, you can tune MySQL from the application level. Using a product like Redis which is also supported by OpenLogic, you can cache database queries in an in-memory database. This technique works well with databases containing a high read level and a low write level. An example would be queries for static content on your site.

Another technique is pre-fetching records. This is done to prevent n+1 queries which is a type of query that fetches n amount of records and then runs additional queries for each of the records. Caching columns and query results as variables are two other techniques available.

There are some other optimizations you can make at the MySQL level, such as only returning selected columns in search results, using ENUM for categorical data types, removing unused indexes, breaking up complex queries into smaller, simpler queries and using the slow query log are among other optimizations you can make when scaling. Setting up replica databases used for read operations is a way to horizontally scale your environment and MySQL comes with built-in support for replication.

### MySQL Use Cases

Potential use cases for MySQL include, but are not limited to:

* **Elastic Replication -** Where an environment requires the number OS servers to grow and shrink dynamically.
* **High Availability -** Where sharding is used for write scale-out (in which each shard maps to a replication group).
* **Source-Replica Replication Alternative -** Which allows using a single source server to make it a single point of contention.
* **Autonomic Systems -**Which allows users to deploy MySQL Group Replication for the automation that is built into replication protocol.

## ADVANTAGES: -

MySQL is becoming so popular because of many good reasons:

MySQL is released under an open-source license. So you have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. MySQL works very quickly and works well even with large data sets. MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB). MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit ther own specific environments.

## FEATURES:

Written in C and C++.

Tested with a broad range of different compilers.

Works on many different platforms. .

For portability, uses **CMake** in MySQL 5.5 and up. Previous series use GNU Automake, Autoconf, and Libtool.

Tested with Purify (a commercial memory leakage detector) as well as with Valgrind, a GPL tool.

Uses multi-layered server design with independent modules.

Designed to be fully multi-threaded using kernel threads, to easily use multiple CPUs if they are available.

Provides transactional and nontransactional storage engines.

Fixed-length and variable-length string types.

**Statements and Functions**

mysql> SELECT CONCAT(first\_name, ' ', last\_name) -> FROM citizen

-> WHERE income/dependents > 10000 AND age > 30;Full support for SQL GROUP BY and ORDER BY

clauses. Support for and[GROUP\_CONCAT()).](https://dev.mysql.com/doc/refman/5.7/en/group-by-functions.html)

Support for LEFT OUTER JOIN and RIGHT OUTER JOIN with both standard SQL and ODBC syntax. Support for aliases on tables and columns as required by standard SQL. Some were changed (affected) to, return the number of rows matched instead by setting a flag when connecting to the server. Support for MySQL-specific [SHOW st](https://dev.mysql.com/doc/refman/5.7/en/show.html)atements that retrieve information about databases, storage engines, tables, and indexes. Support for the INFORMATION\_SCHEMA database, implemented according to standard SQL.

An [EXPLAIN st](https://dev.mysql.com/doc/refman/5.7/en/explain.html)atement to show how the optimizer resolves a query.

Independence of function names from table or column names. For example, ABS is a valid column name. The only restriction is that for a function call, no spaces are permitted between the function name and the “(” that follows it.You can refer to tables from different databases in the same statement.

### **SECURITY:**

A privilege and password system that is very flexible and secure, and that enables host-based verification.

Password security by encryption of all password traffic when you connect to a server.

### **SCALABILITY AND LIMITS:**

Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.

Support for up to 64 indexes per table. Each index may consist of 1 to 16 columns or parts of columns. The maximum index width for [InnoDB ta](https://dev.mysql.com/doc/refman/5.7/en/innodb-storage-engine.html)bles is either 767 bytes or 3072 bytes. The maximum index width for [My ISAM ta](https://dev.mysql.com/doc/refman/5.7/en/myisam-storage-engine.html)bles is 1000 bytes. An index may use a prefix of a column for [CHAR,](https://dev.mysql.com/doc/refman/5.7/en/char.html) [VARCHA](https://dev.mysql.com/doc/refman/5.7/en/char.html)[R,](https://dev.mysql.com/doc/refman/5.7/en/blob.html) [BLOB, or TEXT co](https://dev.mysql.com/doc/refman/5.7/en/blob.html)lumn types.

### **CONNECTIVITY:**

Clients can connect to MySQL Server using several protocols:

Clients can connect using TCP/IP sockets on any platform.

On Unix systems, clients can connect using Unix domain socket files.

### **LOCALIZATION**:

The server can provide error messages to clients in many languages.

Full support for several different character sets, including latin1 (cp1252), german, big5, ujis, several Unicode character sets, and more. For example, the

Scandinavian characters “å”, “ä” and “ö” are permitted in table and column names. All data is saved in the chosen character set.

Sorting and comparisons are done according to the chosen character set and collation

(Using latin1 and Swedish collation by default). It is possible to change this when the MySQL server is started. To see an example of very advanced sorting, look at the Czech sorting code. MySQL Server supports many different character sets that can be specified at compile time and runtime.

The server time zone can be changed dynamically, and individual clients can specify their own time zone.

### **CLIENTS AND TOOLS:**

MySQL includes several client and utility programs. These include both command-line programs and graphical programs.

MySQL Server has built-in support for SQL statements to check, optimize, and repair tables.

These statements are available from the command line through the mysqlcheck client. MySQL also includes mismatch, a very fast command-line utility for performing these operations on MyISAM tables..

MySQL programs can be invoked with the --help or -? option to obtain online assistance.

## ABOUT THE PLATFORM:

Windows is a series of [Operating Systems dev](https://techterms.com/definition/operating_system)eloped by Microsoft. Each version of Windows includes a [Graphical User Interface, wit](https://techterms.com/definition/gui)h a [desktop tha](https://techterms.com/definition/desktop)t allows users to view files and folders in [Windows. For](https://techterms.com/definition/window) the past two decades, Windows has been the most widely used operating system for personal computers PCs.

Microsoft Windows is designed for both home computing and professional purposes. Past versions of Windows home editions include Windows 3.0 (1990), Windows 3.1 (1992), Windows 95 (1995),

Windows 98 (1998), Windows Me (2000), Windows XP (2001), and Windows Vista (2006),windows

7(2009),windows 8(2012),windows 8.1(2013)..

Windows 10 is a series [of operating systems](https://en.wikipedia.org/wiki/Operating_system) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) and released as part of its [Windows NT](https://en.wikipedia.org/wiki/Windows_NT) family of operating systems. It is the successor to [Windows 8.1](https://en.wikipedia.org/wiki/Windows_8.1), released nearly two years earlier, and was [released to manufacturing](https://en.wikipedia.org/wiki/Software_release_cycle#Release_to_manufacturing_(RTM)) on July 15, 2015, and broadly released for the general public on July 29, 2015 .Windows Vista and Windows 7 have followed the same release strategy. Windows is designed to run on standard [x86 har](https://techterms.com/definition/x86)dware, such as Intel and AMD [processors. Th](https://techterms.com/definition/processor)erefore, it can be installed on multiple brands of hardware, such as Dell, HP, and Sony computers, as well as home-built PCs.

During upgrades, Windows 10 licenses are not tied directly to a product key. Instead, the license status of the system's current installation of Windows is migrated, and a "Digital license" (also known as "Digital entitlement" in version 1511 or earlier) is generated during the activation process.

# **.NET FRAMEWORK:**

.NET Framework (pronounced as "dot net") is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for .NET Framework execute in a software environment (in contrast to a hardware environment) named Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling.

As such, computer code written using .NET Framework is called "managed code". FCL and CLR together constitute the .NET Framework. FCL provides [user interface,](https://en.wikipedia.org/wiki/User_interface) [data access,](https://en.wikipedia.org/wiki/User_interface) [database](https://en.wikipedia.org/wiki/Database_connection) [connectivity,](https://en.wikipedia.org/wiki/Database_connection) [cryptography,](https://en.wikipedia.org/wiki/Database_connection) [web](https://en.wikipedia.org/wiki/Web_application) [a](https://en.wikipedia.org/wiki/Web_application)pplication d[ev](https://en.wikipedia.org/wiki/Web_application)elopment, numeric algorithms, a[nd](https://en.wikipedia.org/wiki/Algorithm) [network](https://en.wikipedia.org/wiki/Computer_networking) [communications. Pro](https://en.wikipedia.org/wiki/Computer_networking)grammers produce software by combining their [source](https://en.wikipedia.org/wiki/Source_code) [code wit](https://en.wikipedia.org/wiki/Source_code)h .NET Framework and other libraries.

The framework is intended to be used by newest applications created for the Windows platform. Microsoft also produces an [integrated development environment lar](https://en.wikipedia.org/wiki/Integrated_development_environment)gely for .NET software called [Visual Studio**.**](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio) **.**NET Framework began as [proprietary software, alt](https://en.wikipedia.org/wiki/Proprietary_software)hough the firm worked to [standardize th](https://en.wikipedia.org/wiki/Software_standard)e software stack almost immediately, even before its first release.

Despite the standardization efforts, developers, mainly those in the [free](https://en.wikipedia.org/wiki/Free_and_open-source_software) [and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) [co](https://en.wikipedia.org/wiki/Free_and_open-source_software)mmunities, expressed their unease with the selected terms and the prospects of any free and open- source implementation, especially regarding [software patents. Sin](https://en.wikipedia.org/wiki/Software_patent)ce then, Microsoft has changed

.NET development to more closely follow a contemporary model of a community developed software project, including issuing an update to its patent promising to address the concerns

# **6.TESTING:**

Testing is a vital part of software development, and it is important to start it as early as possible, and to make testing a part of the process of deciding requirements. To get the most useful perspective on your development project, it is worthwhile devoting some thought to the entire lifecycle including how feedback from users will influence the future of the application.

The tools and techniques we've discussed in this book should help your team to be more responsive to changes without extra cost, despite the necessarily wide variety of different development processes. Nevertheless, new tools and process improvements should be adopted gradually, assessing the results after each step.

Testing is part of a lifecycle. The software development lifecycle is one in which you hear of a need, you write some code to fulfil it, and then you check to see whether you have pleased the stakeholders— the users, owners, and other people who have an interest in what the software does.

Hopefully they like it, but would also like some additions or changes, so you update or augment your code; and so the cycle continues. This cycle might happen every few days, as it does in Fabrikam's ice cream vending project, or every few years, as it does in Contoso's carefully specified and tested healthcare support system. Software development lifecycle Testing is a proxy for the customer.

You could conceivably do your testing by releasing it into the wild and waiting for the complaints and compliments to come back. Some companies have been accused of having such a strategy as their business model even before it became fashionable. But on the whole, the books are better balanced by trying to make sure that the software will satisfy the customer before we hand it over.

We therefore design tests based on the stakeholders' needs, and run the tests before the product reaches the users. Preferably well before then, so as not to waste our time working on something that isn't going to do the job.

In this light, two important principles become clear:

**TESTS REPRESENT REQUIREMENTS**: Whether you write user stories on sticky notes on the wall, or use cases in a big thick document, your tests should be derived from and linked to those requirements. And as we've said, devising tests is a good vehicle for discussing the requirements.

**WE ARE NOT DONE TILL THE TEST IS PASSED:** The only useful measure of completion is when tests have been performed successfully.

Those principles apply no matter how you develop your software.

## SOFTWARE TESTING TYPES:

**BLACK BOX TESTING** – Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

**WHITE BOX TESTING** – This testing is based on knowledge of the internal logic of an application’s code. Also known as Glass box Testing. Internal software and code working should be known forthis type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

**UNIT TESTING** – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.

**INCREMENTAL INTEGRATION TESTING** – Bottom up approach for testing i.e continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. done by programmers or by testers.

**INTEGRATION TESTING** – Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

**FUNCTIONAL TESTING** – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

**SYSTEM TESTING** – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

**END-TO-END TESTING** – Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate

**SANITY TESTING** – Testing to determine if a new software version is performing well enough to accept it for a major testing effort. If application is crashing for initial use then system is not stable enough for further testing and build or application is assigned to fix.

**REGRESSION TESTING** – Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

**ACCEPTANCE TESTING** -Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application.

**LOAD TESTING**– Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

**STRESS TESTING**– System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

**PERFORMANCE TESTING** – Term often used interchangeably with ‘stress’ and ‘load’ testing. To check whether system meets performance requirements. Used different performance and load tools to do this.

**USABILITY TESTING** – User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing.

**INSTALL/UNINSTALL TESTING**– Tested for full, partial, or upgrade install/uninstall processes on different operating systems under different hardware, software environment.

**RECOVERY TESTIG** – Testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

**SECURITY TESTING** – Can system be penetrated by any hacking way. Testing how well the system protects against unauthorized internal or external access.

Checked if system, database is safe from external attacks.

**COMPABILITY TESTING**– Testing how well software performs in a particular hardware/software/operating system/network environment and different combination s of above.

**COMPARISON TESTING**– Comparison of product strengths and weaknesses with previous versions or other similar products.

**ALPHA TESTING**– In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

**BETA TESTING** – Testing typically done by end-users or others. Final testing before releasing application for commercial purpose.

# **7.CONCLUSION AND FUTURE ENHANCEMENT**

## 

## CONCLUSION:

The Blood Bank Management System objective is to provide a platform which manages the activity of donors and the recievers. It can update or delete the details according to further information provided. The admin will consume less amount of time when compared to manual work through the application system. The system will take care of all the blood bank activity in a quick manner.

Some existing system do not have security to the data and are not user friendly and reliable. There is lot more to be done in order to provide a 100% secure platform to make it more flexible

## FUTURE ENHANCEMENT:

A website Application of the can be created because it is helpful in different way such as it reduces the travel expense of the receiver as well as the blood can be delivered to the receiver s location .The receiver can request the required units of blood and its type from his own location .We can do it in a better way by providing our software and the receiver can purchase the blood anywhere anytime. This also provides high level of convenience.

# 8. **BIBLIOGRAPHY**