1. **Introduction.**

A StepCounter is a device, usually portable and electronic or electromechanical, that counts each step a person takes by detecting the motion of the person's hands or hips. Because the distance of each person's step varies, an informal calibration, performed by the user, is required if presentation of the distance covered in a unit of length (such as in kilometers or miles) is desired, though there are now pedometers that use electronics and software to automatically determine how a person's step varies.

A Step Tracker/Counter generates an event each time a step is taken by the user.

**About StepCounter.**

This app is very simple to use and interact as its UI is as easy as it looks, and does exactly what you need. Stats are clear and easily accessible from the menu on the right, and it will automatically set you a daily goal and monitor your progress. In addition to your counted steps, the information will display the distance in km.

Also, you will able to

**How does it work?**

An individual will enter his Height, Weight and Age. Then he/she will select to calculate his/her BMR as per the option provided to choose from (i.e. Male/Female)

If you want to re-enter the values you can re-enter by just hitting RESET.It will generate result and then proceed further.

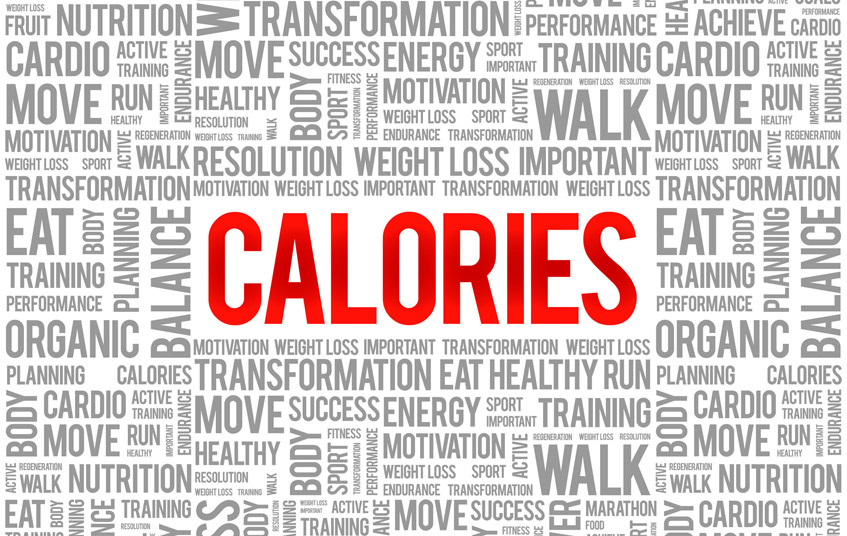
Now calculating your TDEE based on your activity levels which will take the value of your BMR and analyze further for your TDEE value.There will be 4 Activity levels to choose from which will be based on personal level.Finally the output will be displayed in a form of result.And with the final result user will get calories for how much he/she needs to lose weight or gain weight.

Benefits: Once you know your BMR, you can use it to calculate the calories you actually burn in a day. From there, you can determine how many calories you need to eat to gain muscle, lose fat, or maintain your weight.

The overall number of calories your body uses on a daily basis is referred to as your "total daily energy expenditure" (TDEE). It's determined based on your BMR as well as your activity level throughout the day.

This varies significantly based on your activity level, age, and gender.

Generally, men have a higher TDEE than women because they have more muscle mass, and both TDEE and BMR tends to fall regardless of gender as you age. Your TDEE is how many calories you expend every day.



1. **Requirement Specifications.**

**Tools and Technology Used.**

**Hardware:** Windows 7/8/10; RAM: 4+ GB; Storage Space: up to 100 GB.

**Software: NetBeans 8.0/1/2**

**NetBeans IDE:** It is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called modules.

It is an open source for all java application types (Java SE, Java ME, web, EJB and mobile applications) out of the box.

**Modularity:** All the functions of the IDE are provided by modules. Each module provides a well-defined function such as creating an interface with the help of various GUI design tool (Swing) and its components just by dragging and positioning it.

**Language used:** Java Web Application and Swing components.

You must have stable version of NetBeans IDE 8.2 version with required JDK files.

Also for using Java language, our framework must have Java Swing with proper design and source tab.

Install Java SE, Java Swing components, Java AWT, Java API.

You should also have background knowledge of designing the components in GUI by Java API- Java language and Java Swing based knowledge.

1. **System Design Details.**
   1. **Methodology.**

**Waterfall model** is a sequential flow which is used in developing particular software.

This helps us to guide through the making of software/application.

It consists of **six phases** which are as follows:

1. Requirement Analysis.
2. Designing of software.
3. Implementation of software.
4. Testing/Working of the software.
5. Deployment of the software.
6. Maintaining of a software.

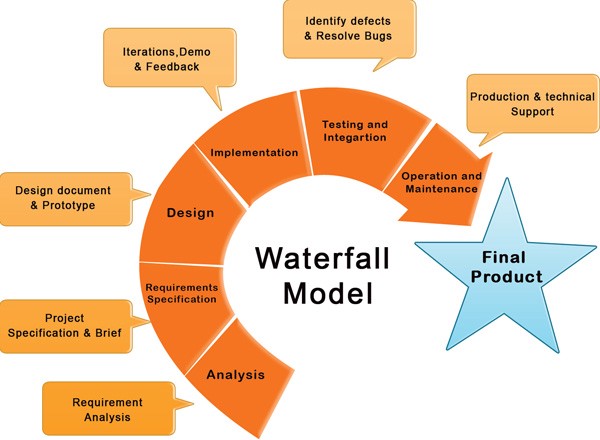
**Benefits:**

* The main purpose of this model is that it is one time process.
* When you proceed to next phase you can't go back to the previous phase so once you have completed one phase checking thoroughly and moving to next phase is a smart move.
* The use of this model allows to early design of the software in the planning phase then in modeling phase which it is only implemented.
* Tasks of this model to various group of people is easy to manage and co-ordinate.
* Once you have all the requirements needed it is easy to make the software/application.
* As the model is sequential and linear the working of well working software can be divided into group as per the phases.
* The final process or results or output are well log maintained which means it is well documented.

**Why I choose this waterfall model for my software/application?**

* As my software/application has all the requirements analyze at the start of making the software it is easy for this model to proceed.
* The root of making this software is clear and stable from beginning which means no sudden changes will be done.
* Customer/Client using the final software will not be involved in making of the software.

**Diagram Representation.**

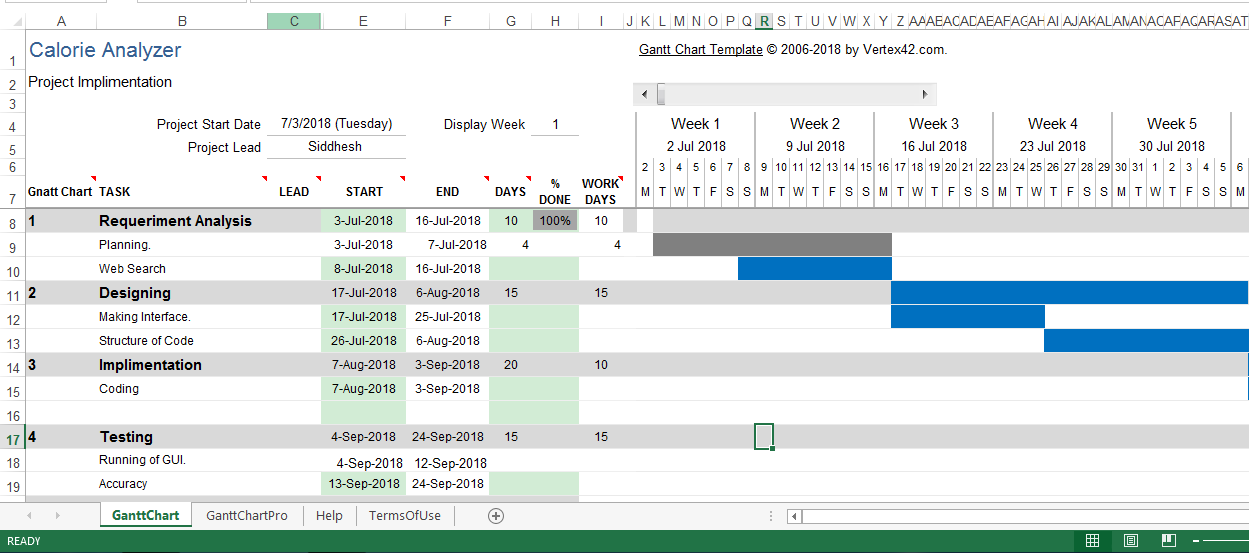


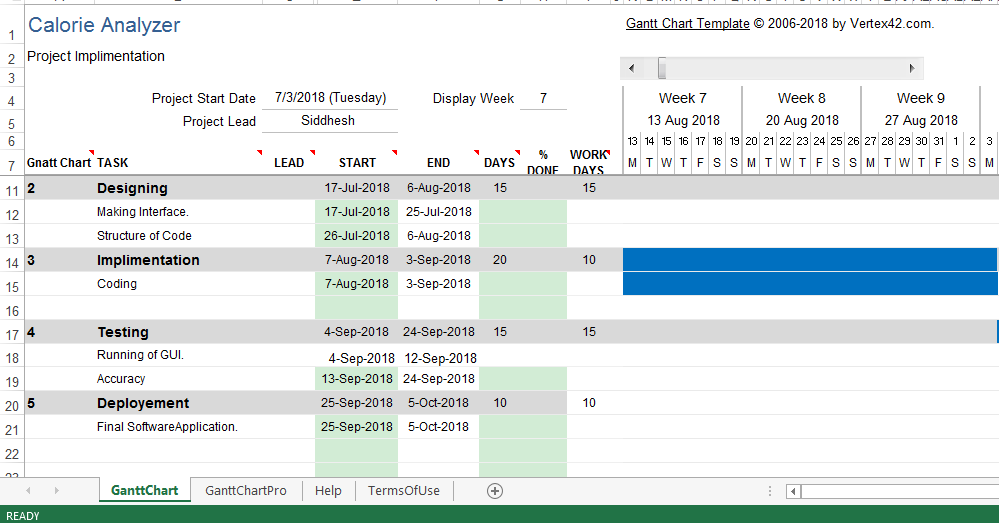
* 1. **Architecture.**

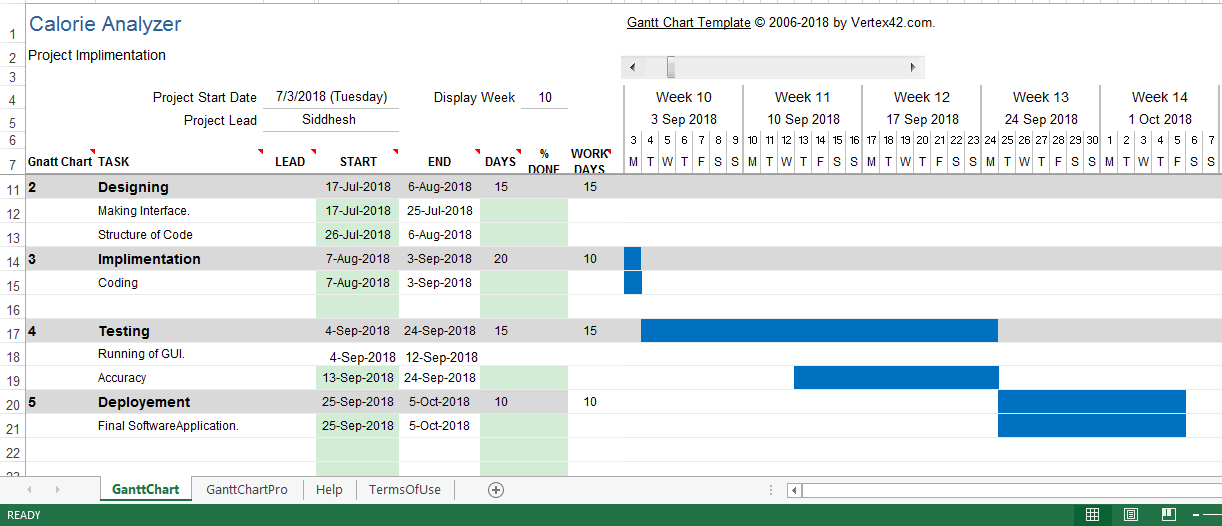
1. **Gantt chart.**

A Gantt chart is a horizontal bar chart developed as a production control tool. Frequently used in project management, a Gantt chart provides a graphical illustration of a schedule that helps to plan, coordinate, and track specific tasks in project.

Gantt chart give a clear illustration of project status, but one problem with them is that they don’t indicate task dependencies – you cannot tell how one task is falling behind schedule affects other tasks.

****

****

****

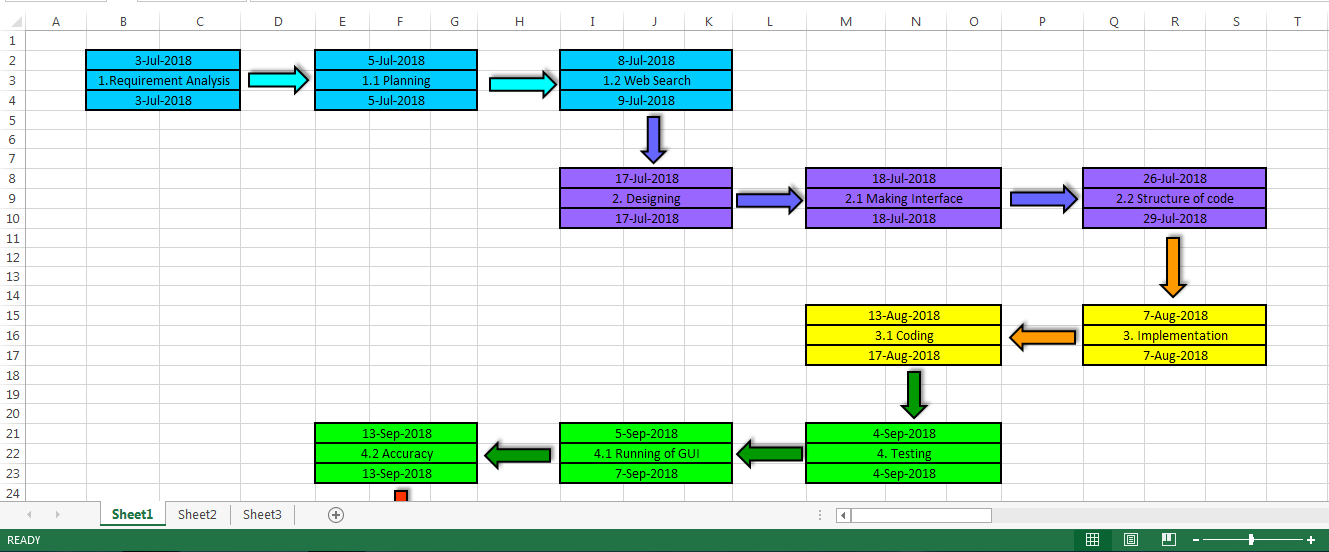
1. **Pert Chart.**

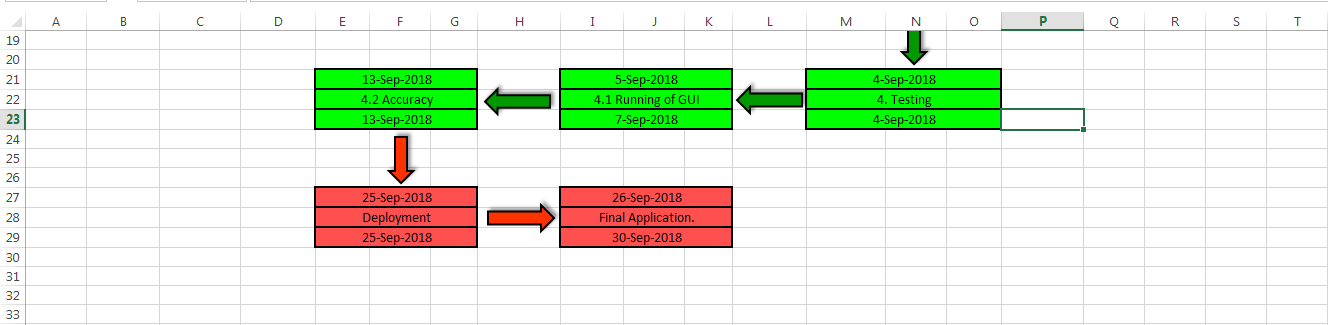
**(Program Evaluation Review Technique)**

A PERT chart is a project management tool used to schedule, organize, and coordinate tasks within a project.

Pert represents a graphical illustration of a project as a network diagram consisting of numbered nodes representing events, or milestones in the project linked by labelled vectors representing tasks in the project.

The direction of the arrows on the lines indicates the sequence of tasks.

****

****

* 1. **UML (Unified Modeling Language).**
     1. **Activity Diagram.**

Activity diagram presents a number of benefits to users. They describe what must happen in the system being modeled.

It is used for starting, ending, merging, or receiving steps in the flow.

The components of an activity diagram includes:

Action, Decision mode, Control flows, Start node, and end node.

| **Symbol** | **Name** | **Description** |
| --- | --- | --- |
| start Symbol | Start symbol | Represents the beginning of a process or workflow in an activity diagram. It can be used by itself or with a note symbol that explains the starting point. |
| activity Symbol | Activity symbol | Indicates the activities that make up a modeled process. These symbols, which include short descriptions within the shape, are the main building blocks of an activity diagram. |
| connector Symbol | Connector symbol | Shows the directional flow, or control flow, of the activity. An incoming arrow starts a step of an activity; once the step is completed, the flow continues with the outgoing arrow. |
| condition text | Condition text | Placed next to a decision marker to let you know under what condition an activity flow should split off in that direction. |
| end symbol | End symbol | Marks the end state of an activity and represents the completion of all flows of a process. |

NO

YES

End of the Process

Insert your personal details as mentioned.

Start of the Process

Get your TDEE value based on Activity Level

Get your BMR value

(Male / Female)

If details are correct.

RESET

* + 1. **Flow Chart**

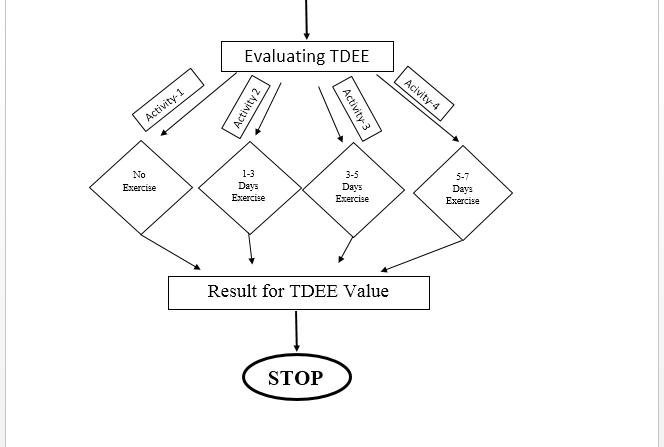
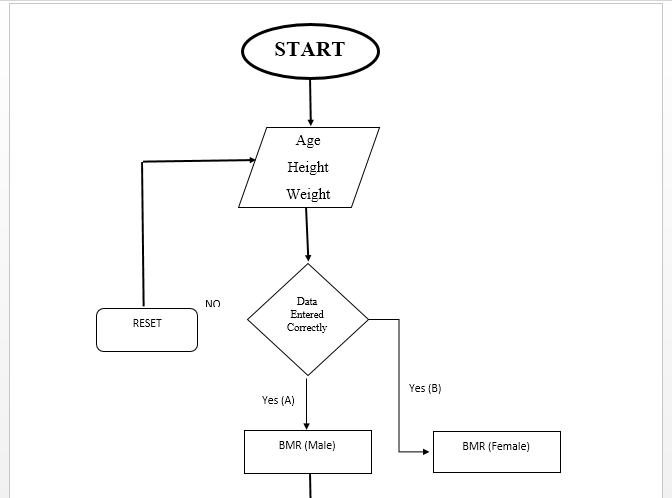
A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, and plan improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.

As a visual representation of data flow, flowcharts are useful in writing a program or algorithm and explaining it to others or collaborating with them on it. You can use a flowchart to spell out the logic behind a program before ever starting to code the automated process.

## Flowchart symbols

Here are some of the common flowchart symbols:

| **Flowchart Symbol** | **Name** | **Description** |
| --- | --- | --- |
| Process Flowchart Symbol | Process symbol | Also known as an “Action Symbol,” this shape represents a process, action, or function. It’s the most widely-used symbol in flowcharting. |
| Start/End Flowchart Symbol | Start/End symbol | Also known as the “Terminator Symbol,” this symbol represents the start points, end points, and potential outcomes of a path. Often contains “Start” or “End” within the shape. |
| Decision Flowchart Symbol | Decision symbol | Indicates a question to be answered — usually yes/no or true/false. The flowchart path may then split off into different branches depending on the answer or consequences thereafter. |
| Input/Output Flowchart Symbol | Input/output symbol | Also referred to as the “Data Symbol,” this shape represents data that is available for input or output as well as representing resources used or generated. While the paper tape symbol also represents input/output, it is outdated and no longer in common use for flowchart diagramming. |



* + 1. **Sequence Diagram.**

Sequence diagrams are a popular dynamic modeling solution in UML because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

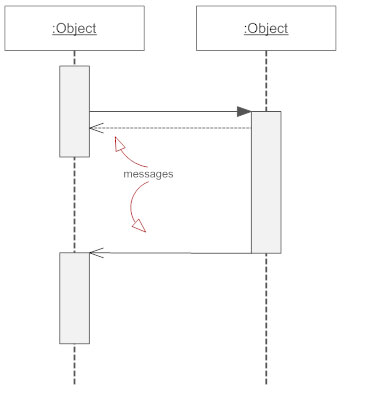
**Classes**  
Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

Object symbol - Sequence diagram

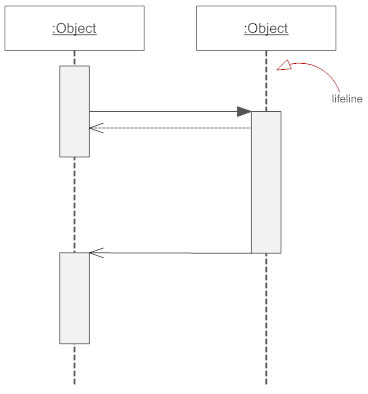
**Activation & Execution:** Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.



**Messages**  
Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks. For message types, see below.



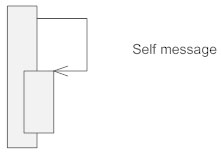
**Lifelines**  
Lifelines are vertical dashed lines that indicate the object's presence over time.



**Synchronous-Messages**  
A synchronous message requires a response before the interaction can continue. It's usually drawn using a line with a solid arrowhead pointing from one object to another.

Synchronous message - Sequence diagram

**Self-Message**  
A message an object sends to itself, usually shown as a U shaped arrow pointing back to itself.



AlgorithmA()

AlgorithmB()

Final Result

PostprocessingB()

PreprocessingA()

PreprocessingB()

Result B

Input Data B

Input Data A

Result A

/client: ClientContextA

/server: ServerA

/User

* + 1. **Use – Case Diagram.**

A use case is a written description of how users will perform tasks on your website.  It outlines, from a user’s point of view, a system’s behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.

## Benefits of Use Cases

Use cases add value because they help explain how the system should behave and in the process, they also help brainstorm what could go wrong.  They provide a list of goals and this list can be used to establish the cost and complexity of the system. Project teams can then negotiate which functions become requirements and are built.

## Elements of a Use Case

Depending on how in depth and complex you want or need to get, use cases describe a combination of the following elements:

**Actor** – anyone or anything that performs a behavior (who is using the system)

**Stakeholder** – someone or something with vested interests in the behavior of the system under discussion (SUD)

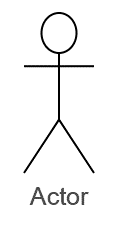
**Primary Actor** – stakeholder who initiates an interaction with the system to achieve a goal

**Preconditions** – what must be true or happen before and after the use case runs.

**Triggers** – this is the event that causes the use case to be initiated.

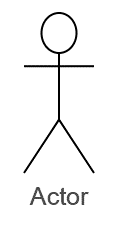
**Main success scenarios** [Basic Flow] – use case in which nothing goes wrong.

**Alternative paths [**Alternative Flow] – these paths are a variation on the main theme. These exceptions are what happen when things go wrong at the system level.



Display Result

Input



Display Result

* 1. **DFD (Data Flow Diagram).**

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.

Retrieving selected Activity

Selection of particular Activity

Selection of particular Activity

ACTIVITIES

Level 1 - 4

User Information

User Information

**USER**

Evaluate TDEE (Total Daily Energy Expenditure).

Giving User the evaluated Result

* 1. **Algorithm.**

Steps

1. Start.
2. Initialize variables (age, height, weight).
3. Initialize the variables as a data type to Double.
4. Initialize the variable to store the calculated value.
5. Take the Input from User.
6. User has to input the values in given respective text box.
7. If the given data is WRONG then go to Step 7.
8. Click RESET if data entered is incorrect.
9. Click on button created for BMR for either Male or female.
10. This will evaluate the BMR (Basic Metabolic Rate).
11. Select the data to evaluate as per your gender (Male/Female).
12. Evaluate the data given by the user and store it.
13. Display the evaluated result to the user.
14. Now for next phase, to evaluate the TDEE (Total Daily Energy Expenditure).
15. Initialize variables (TDEE, gain, loose).
16. Now give the initialized variables the data type as Double.
17. Now select the input based on choices displayed in form of button accordingly.
18. After selecting the input choice provided, evaluate the following and store in the variable TDEE.
19. Evaluated data should be displayed in the GUI.
20. You will get the value desired and also, how much you need to lose weight as well as gain weight.
21. STOP.
22. **System Implementation.**

The implementation of Calorie Analyzer is better easily presented with using the NetBeans Framework.

By using JAVA language with the NetBeans functions have great advantage on how the flow of the application / software is set and performed properly.

The Calorie Analyzer make best use of the provided GUI by JAVA swing components to make it look neat and clean and easy to use by the user.

Everyone will able to use and understand without guide on how to use it as the input are simple and in simple language.

By using Java Frame as base and dragging and placing the JTextbox, JLabel and JButton as per the requirement of application is done easily and effortless.

Also, the back end of interface also is easy to do by double click on components and write the desired code to give a specific functions

Initializing, Declarations and Comments are easy to do.

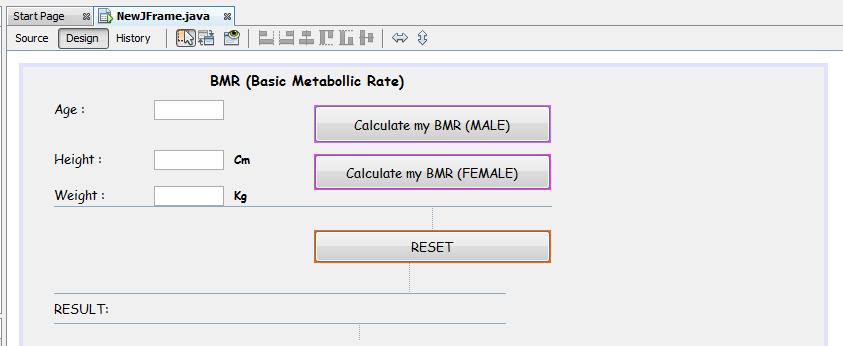
After the coding you can easily run the application / software created.

1. **Results.**
   1. **Preview/Screenshots.**

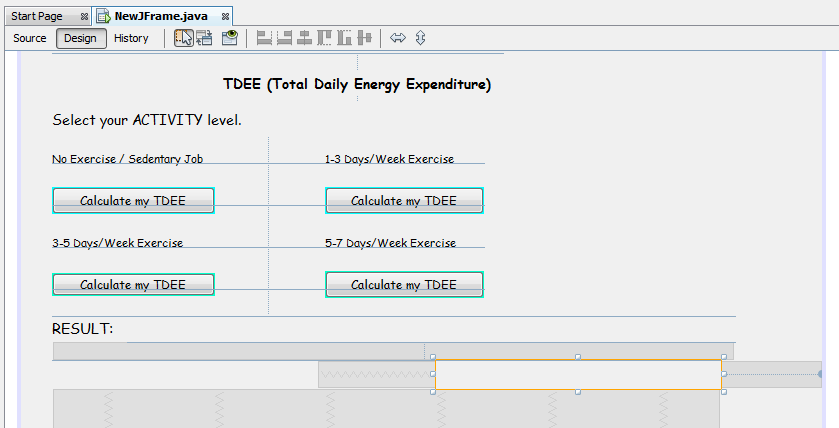
It is a blueprint of what the GUI will look like.

Here you can set up the Java Swing components, and drag and place them where you needed to be. Also you can customize the look and feel and alignment of components for best of user experience. Look at the following image to see the NetBeans design screenshots.

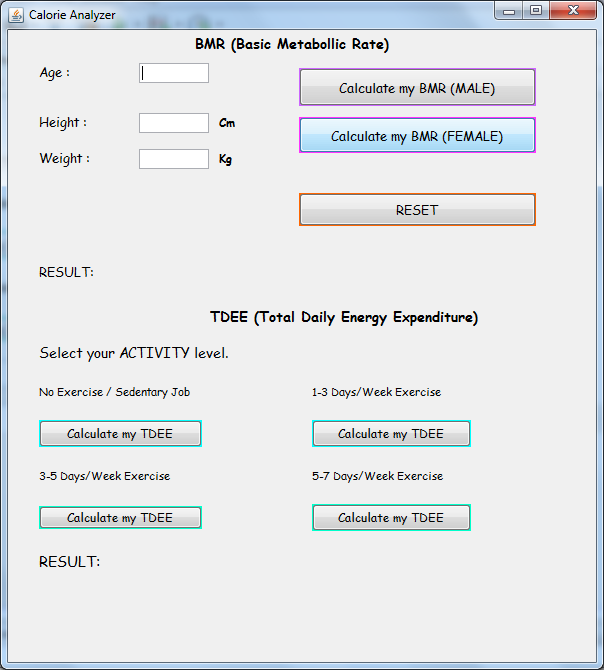
**For analyzing the BMR.**

****

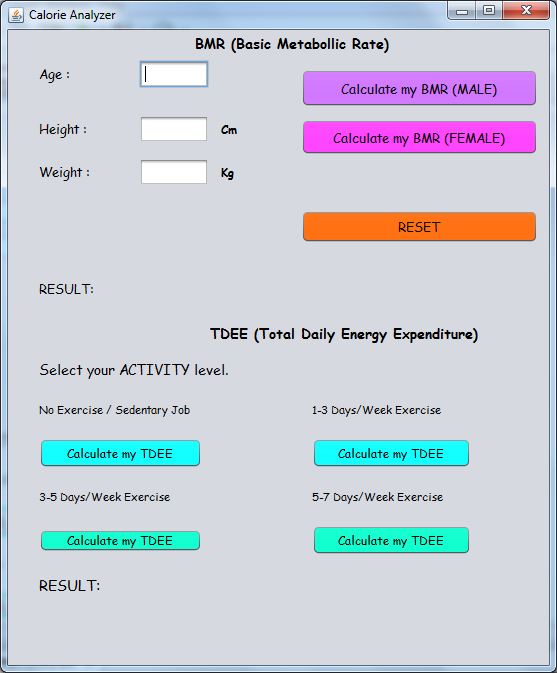
**For analyzing the TDEE.**

****

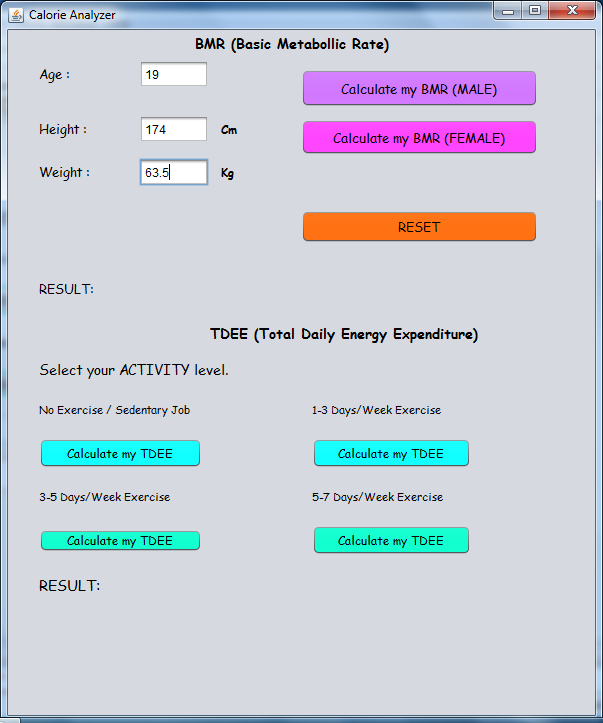
**The full basic preview of design, of how the user will see the interface.**

****

**Actual Interface that will User experience.**

****

**Input given by the user.**

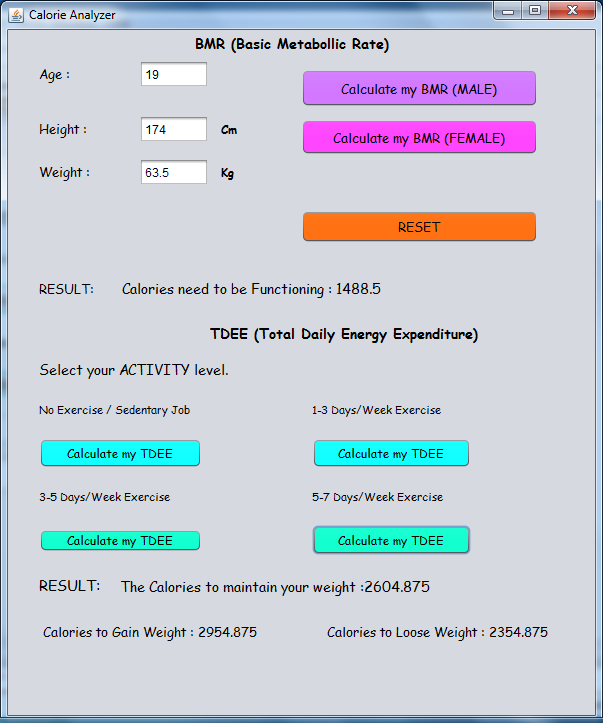
****

**Generated BMR calories.**

****

**Generated TDEE calories.**

**Also displaying calories for an individual needs to lose or gain weight.**

****

* 1. **Reports**

The objectives of the project are –

To evaluate BMR (Basic Metabolism Rate) i.e. Calories needed to the body to survive.

To evaluate TDEE (Total Daily Energy Expenditure) i.e. Calories needed to the body if you are doing physical activity.

This Calorie Analyzer can be used by both Male Female

By getting the result of your calories specifically for your body it can further help you to gain weight or lose weight.

By calculating TDEE you will get calories on based off of your specified activity level. So that you can maintain your weight as per your physical activity throughout the day.

You will also get the calories needed for gaining weight and losing weight.

Now by knowing this personalized information you can further change your body by consistently eating nutritious food, doing activity and exercising.

The calorie analyzer covers the huge section for getting fit and into shape you are going for. This is the only way to start up, if you want to change the way you look and feel in your body.

It is important to take care of your body for not only to get a good physique or to just lose weight if you are obese or if you are skinny but because it is the only place you have to live in for your rest of your life.

1. **Conclusions and future scope.**

The Calorie Analyzer will results will help you further to make up your Macro Nutrients which are branches of those calories.

With Calorie analyzer which is implemented now into desktop system environment can be also implement into android based applications.

If you are start getting into the health and fitness related niche knowing your calories is much important because you have to have your calories in-check for Muscle building, losing fat, also losing weight.

Total daily calories consumption can also be manipulated/maintain to stay same or better your life with your own personalized goals.

The Calorie Analyzer will also be used to evaluate each and every food intake as food contains calories you might have to watch out exactly how much body needed and how much you are taking IN.

Once you’ve established your daily calorie intake, I suggest initially tracking your weight on a weekly basis.

This will help determine if you need to adjust calorie intake to optimize your fat-loss goals.

* 1. **References.**

## For source code and GUI

## For coding reference java tutorials point.

## Oracle docs.

## NetBeans.org

## Information about BMR and TDEE

## Body Bodybuilding.com Forum for Description.

## Harris–Benedict Formula used for Source code.

## Understanding the fundamentals of BMR for body.

## Making documentation and Diagram presentation

## Lucid chart.com