TITLE:

TASK MANAGEMENT SYSTEM (TODOLIST)

TEAMMATES:

2033038 –VIDHYA VARSHANY J S

2033009 – DHARSSINI K

OVERVIEW:

- This application mainly focuses on the efficient improvement on the end users productivity and scheduling his/her daily routine by creating new tasks, assign them a title ,progress of the task, due dates and choosing a project for that task to belong to
- It has a text-based user interface in Command Line Interface
- Once the user started using the application, he/she should be able to also edit, mark the task as done or remove tasks.

FUNCTIONAL REQUIREMENTS:

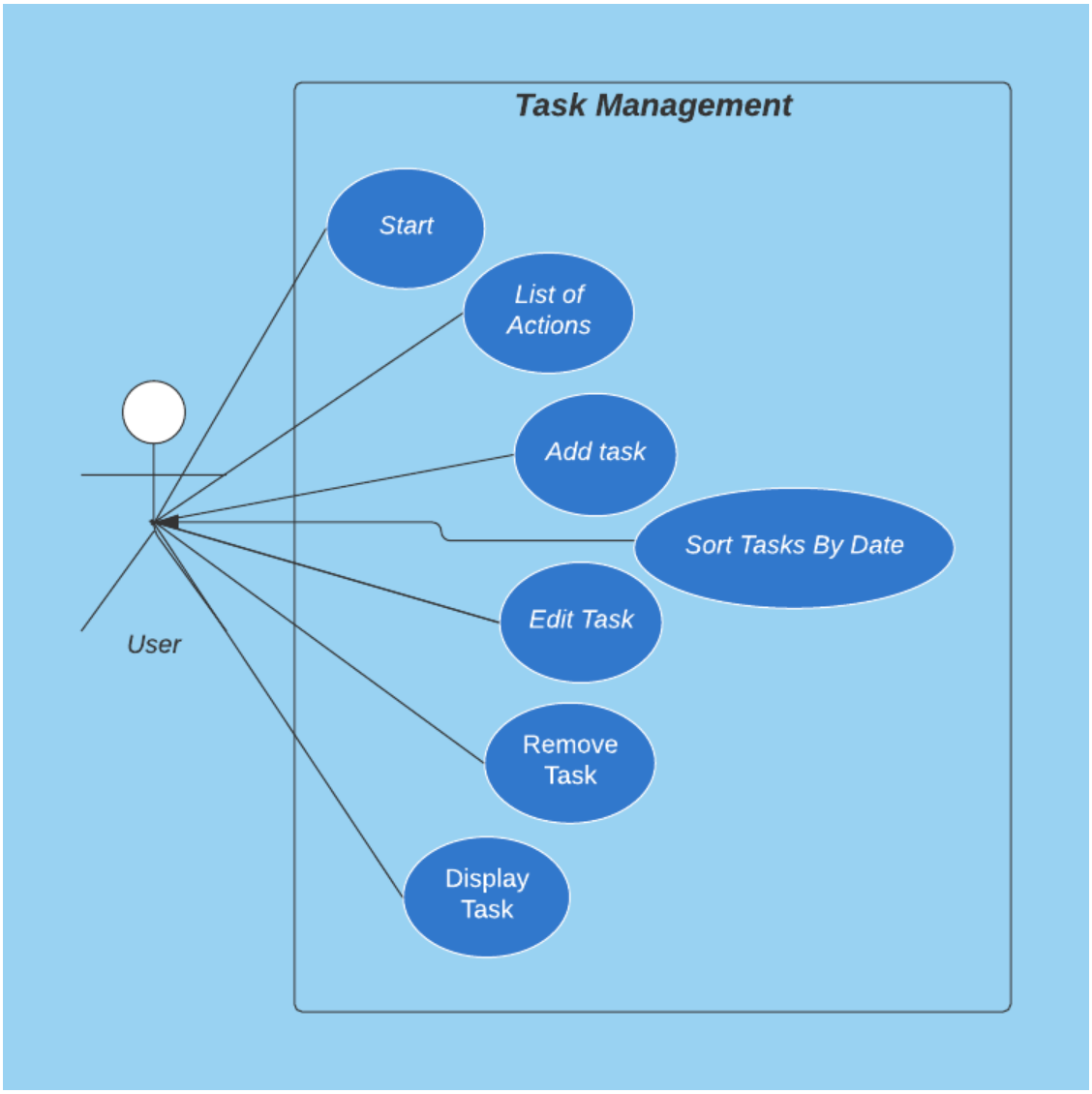
- Initialize the task with a task title, due date , status and project
- Display a collection of tasks that can be sorted by date.
- Enabling the actions like add, sort by date, edit ,mark as done and remove tasks
- Displaying the output in the Human readable format
- Additionally Load and save task list to file for future references

CLASSES IDENTIFIED

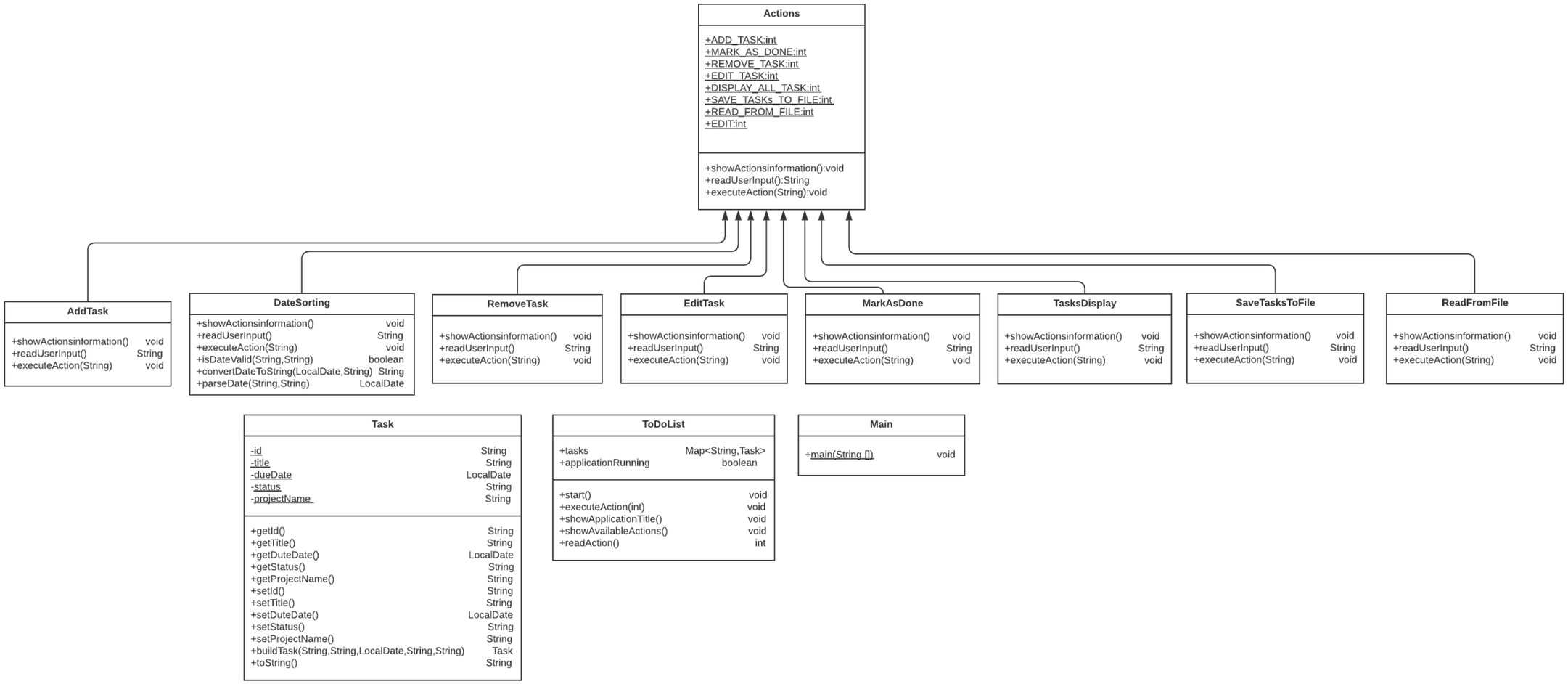
- **TodoList** This Class is the Parent class which contains all the menu for the dynamic functionality of the other sub classes are acted upon just by calling the specific feature option.

- **Actions** The Action abstract class used to perform the features like add tasks,marks as done,remove task,edit task, display all task by assigning static final variable and abstract void methods to access among the other classes.
- **AddTask** This Class is specifically to do the functionality to show Actions Information ,reading user input and execute Action
- **Sort By Date** This Class enables the user to have the ability to organize tasks by Date,to give better visualization and a better reading
- **EditTask** This Class will give the user ability to search and identify a certain task using the ID number and edit some or all parts of the task as desired and the update the list itself
- **MarkAsDone** This Class enables the user to set the task as done once they feel they are satisfied with the task.
- **RemoveTask** The Class enables the feature to delete the current file by choosing the Remove option in the menu. The File will be taken out of the list after the use had searched and identified the specific folder to be deleted
- **TasksDisplay** This Class displays all the tasks, by choosing Display All Tasks option, this will fetch all related data to tasks given in the to-do list
- **SaveTasksTo File** This Class can save any file by choosing the **Save Tasks** option, this will make sure the user doesn't miss any of his/her tasks
- **ReadFromFile** This Class displays all the tasks, by reading or fetching the tasks information form a local existing file.

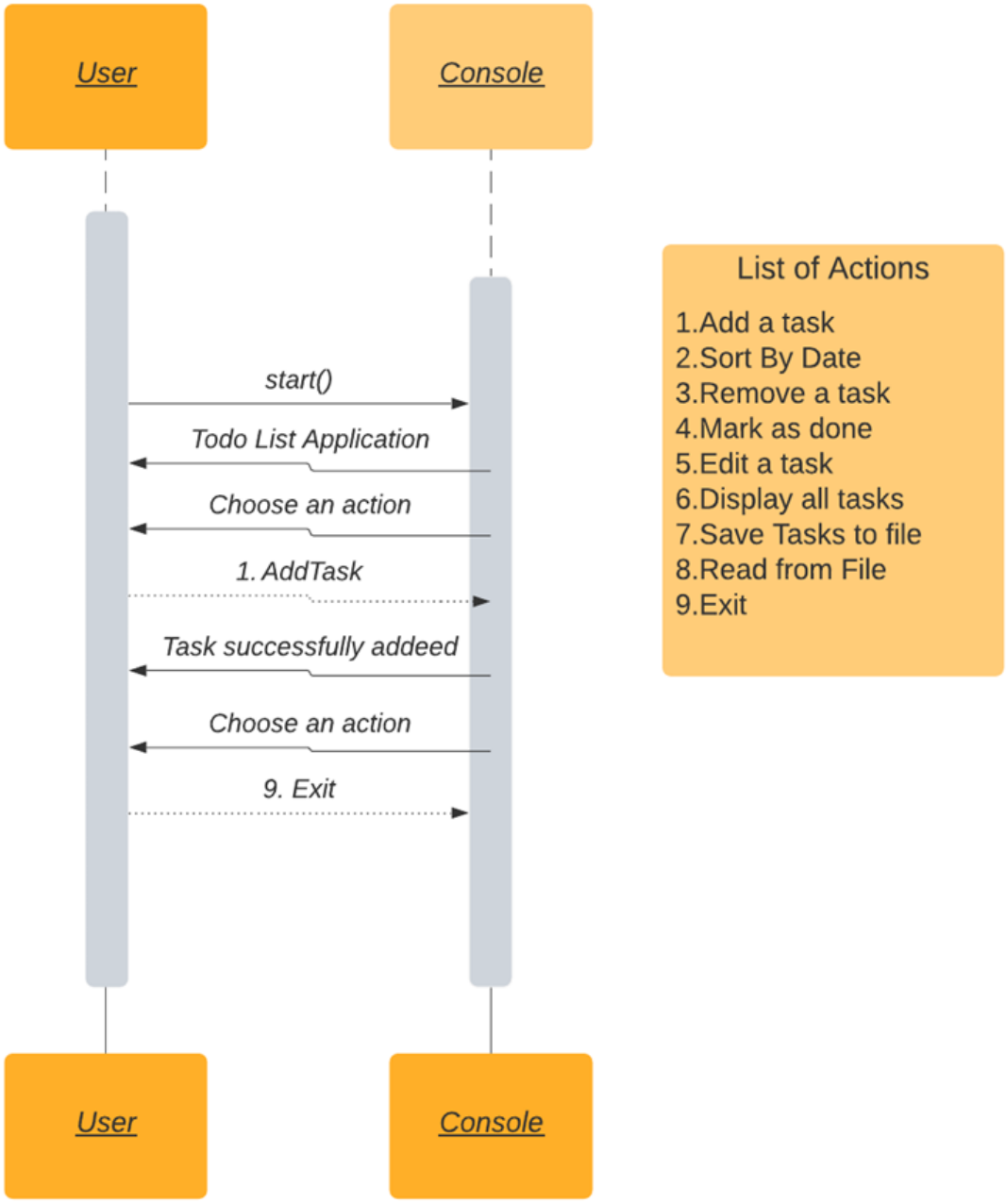
Use-Case Diagram



Class Diagram



Sequence Diagram



CODE:

```
package javaswings;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.DefaultListModel;

public class list extends javax.swing.JFrame {

   public list() throws ClassNotFoundException, SQLException {
      initComponents();
      table_update();
   }

   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">
   private void initComponents() {

      jScrollPane1 = new javax.swing.JScrollPane();
      jList1 = new javax.swing.JList<>();
      jLabel1 = new javax.swing.JLabel();

      setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

      jList1.addMouseListener(new java.awt.event.MouseAdapter() {
         public void mouseClicked(java.awt.event.MouseEvent evt) {
            jList1MouseClicked(evt);
         }
      });
      jScrollPane1.setViewportView(jList1);

      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
      getContentPane().setLayout(layout);
      layout.setHorizontalGroup(
           layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
                          .addGroup(layout.createSequentialGroup()
                               .addGap(92, 92, 92)
                               .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 241, javax.swing.GroupLayout.PREFERRED_SIZE))
                          .addGroup(layout.createSequentialGroup()
                               .addGap(181, 181, 181)
                               .addComponent(jLabel1)))
```

```
                    .addContainerGap(144, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(21, 21, 21)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
261, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(44, 44, 44)
                    .addComponent(jLabel1)
                    .addContainerGap(95, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jList1MouseClicked(java.awt.event.MouseEvent evt) {

        jLabel1.setText("Welcome " + (String) (jList1.getSelectedValue()));// TODO add your
handling code here:
    }

    /**
     * @param args the command line arguments
     */
    Connection con1;
    PreparedStatement insert;

    private void table_update() throws ClassNotFoundException, SQLException {
        int CC;
        DefaultListModel model = new DefaultListModel();
        Class.forName("com.mysql.cj.jdbc.Driver");
        con1 = DriverManager.getConnection("jdbc:mysql://localhost:3306/StudentList", "root",
"");
        insert = con1.prepareStatement("SELECT name FROM records");
        ResultSet Rs = insert.executeQuery();
        while (Rs.next()) {
            String itemCode = Rs.getString("name");
            model.addElement(itemCode);
        }
        jList1.setModel(model);
    }

    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
```

```
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(list.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(list.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(list.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(list.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                new list().setVisible(true);
            } catch (ClassNotFoundException ex) {
                Logger.getLogger(list.class.getName()).log(Level.SEVERE, null, ex);
            } catch (SQLException ex) {
                Logger.getLogger(list.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    });
  }

  // Variables declaration - do not modify
  private javax.swing.JLabel jLabel1;
  private javax.swing.JList<String> jList1;
  private javax.swing.JScrollPane jScrollPane1;
  // End of variables declaration
}
```

OUTPUT

```
To DO List Application
----------------------

1. Add a task
2. Mark task as done
3. Remove task
4. Edit task
5. Display all tasks
6. Sort tasks by date
7. save tasks to file
8. read from file
9. Exit

Enter action: 1

To add a new task, please follow the instructions and press ENTER:
Task ID, Task Title, Due Date (format: dd-mm-yyyy), Status, Project Name
eg: 1, Dharssini Vidhya,01-12-2021,in-progress,Java Project
Enter 0 to RETURN

Enter Information: 1, Dharssini Vidhya,01-12-2021,in-progress,Java Project
Task successfully added!

1. Add a task
2. Mark task as done
3. Remove task
4. Edit task
5. Display all tasks
6. Sort tasks by date
7. save tasks to file
8. read from file
9. Exit

Enter action: 1

To add a new task, please follow the instructions and press ENTER:
Task ID, Task Title, Due Date (format: dd-mm-yyyy), Status, Project Name
eg: 1, Dharssini Vidhya,01-12-2021,in-progress,Java Project
Enter 0 to RETURN
```

```
Enter Information:  2, CAT 1 Project,01-01-2022,in-progress,Java Project
Task successfully added!

1. Add a task
2. Mark task as done
3. Remove task
4. Edit task
5. Display all tasks
6. Sort tasks by date
7. save tasks to file
8. read from file
9. Exit

Enter action: 5

Here are all the tasks:
ID: 1, Title:  Dharssini Vidhya, Due Date: 01-12-2021, Status: in-progress, Project: Java Project
ID:  2, Title:  CAT 1 Project, Due Date: 01-01-2022, Status: in-progress, Project: Java Project

1. Add a task
2. Mark task as done
3. Remove task
4. Edit task
5. Display all tasks
6. Sort tasks by date
7. save tasks to file
8. read from file
9. Exit

Enter action: 6
Tasks successfully Sorted!

1. Add a task
2. Mark task as done
3. Remove task
4. Edit task
5. Display all tasks
6. Sort tasks by date
7. save tasks to file
8. read from file
9. Exit
```

```
1. Add a task
2. Mark task as done
3. Remove task
4. Edit task
5. Display all tasks
6. Sort tasks by date
7. save tasks to file
8. read from file
9. Exit

Enter action: 2

To mark a task as done, enter ID and press ENTER:

Enter 0 to RETURN

Enter task id:2
There is no task with this ID, try again:

Enter task id:7
There is no task with this ID, try again:

Enter task id:8
There is no task with this ID, try again:

Enter task id:1
Status is set as Done for the task with ID: 1

1. Add a task
2. Mark task as done
3. Remove task
4. Edit task
5. Display all tasks
6. Sort tasks by date
7. save tasks to file
8. read from file
9. Exit
```

File.txt

> 1, Dharssini vidhya,01-12-2021,in-progress,Java Project

CONCLUSION:

This project aims to increase the productivity of the user by maintaining a jot down list to track his/her work with a minimal necessary features in a terminal environment