SENG 272-SENG 384 Software Project (for Developers in Project Groups)

Project Name: Task and Wish Management Application for Children

1. INTRODUCTION

1.1 Purpose

This project aims to develop a console-based application that allows children to track their daily/weekly tasks, earn points, and manage their wishes (products/activities) with family approval.

1.2 Scope

The application will be developed in Java as a console-based program. The scope of this project includes the following:

- Task Management
- Wish Management
- Points/Level Tracking
- Data Persistence via Files: Reading and writing tasks and wish data from/to a file.

There are three input files: `Commands.txt`, `Wishes.txt`, and `Tasks.txt`. When the program starts, it will execute the commands in the command file in order, with no additional menu provided in the program. At the beginning, the Task and Wish lists are read as input from their respective files. If new tasks or wishes are added, these files will be updated accordingly. The outputs for other commands will be displayed on the console.

2. FUNCTIONAL REQUIREMENTS

2.1 Users

- **Child**
- **Parent**
- **Teacher**

2.2 Task (Assignment) Management

2.2.1 Adding a Task

- A parent can add general responsibilities and tasks for a child.
- A teacher can add school assignments for a child.
- Each task should include a title, description, due date (deadline), and point value.
- Tasks may include specific “activity time”. Some tasks only have a deadline (TASK1), while others need to be completed within a specified time range (TASK2).

2.2.2 Listing / Viewing Tasks

- The child should be able to view their list of tasks.
- They should be able to filter tasks on a daily or weekly basis.
- Only tasks that include an “activity time” should appear on the child’s calendar.

2.2.3 Completing and Approving Tasks

- When the child finishes a task, they can mark it as “Completed.”
- The task is not considered truly completed until it is approved by the person who assigned it (parent or teacher). Once approved, the child’s points should increase accordingly.
- When a task is completed, the parent or teacher can give a rating (1 to 5 stars).
- The child’s overall level is calculated based on the average of these ratings. Certain wishes become available only at or above certain levels.

2.3 Wish Management

2.3.1 Adding a Wish (Child)

- The child can record a wish for a product as WISH1 (toy, book, etc.) or an activity as WISH2 (going to the cinema, a sports course, etc.).
- It should include date and time details if it is an activity wish. Approved activities can then be added to the child’s calendar.
- Once a wish is entered, its status goes to “pending approval” (or similar).

2.3.2 Approving a Wish (Parent)

- The parent can approve or reject any pending wish.
- If a wish is not approved, it should be removed from the child’s Wish List.
- Beyond merely approving the wish, the parent also specifies which level is required for that wish to be active. (For example, if the parent sets “level 3,” the child can only access that wish if their level is at least 3.)

2.4 Points and Level System

2.4.1 Points Tracking

- The child’s current points are tracked by the system. When tasks are approved, these points increase.

2.4.2 Defining Levels

- Levels are determined based on specific criteria, for example:
    - 0–40 average points: Level 1
    - 40–60 average points: Level 2
    - 60–80 average points: Level 3
    - 80+ average points: Level 4
- The parent or teacher can add extra points to the child’s total if needed.

3. COMMANDS AND USAGE

Below are the proposed commands to be used via input/output files in the console-based system, along with example usages and parameter details.

3.1 ADD_TASK

Purpose: Adding a task by a parent or teacher.

Example Usage:

```
ADD_TASK1 T 101 "Math Homework" "Solve pages 10 to 20" 2025-03-01 15:00 10
```

- Creates a task with an ID (`101`), assigned by a teacher (`T`), titled *Math Homework*, with a deadline of *March 1, 2025, at 15:00* and worth *10 points*.

```
ADD_TASK2 T 101 "School Picnic Preparation" "Göksu Park" 2025-03-05 10:00 2025-03-05 12:00 10
```

- Similar, but includes a start and end time for the activity. This indicates a specific time slot in which the task should be completed.

Parameters:

1. Who is adding the task? (`T` for Teacher, `F` for Parent — assuming the original notation "F" can stand for "Family/Parent")
2. Task_ID (e.g., `101`)
3. Title (String)
4. Description (String)
5. Deadline Date (Format: `yyyy-MM-dd`; e.g., `2025-03-01`)
6. Deadline Time (Format: `HH:mm`; e.g., `15:00`)
7. Task Points / Budget_Coin (Integer)

Note: For tasks that require a specific activity time (start/end), include additional parameters:

- `[start_date, end_date, start_time, end_time]`

3.2 LIST_ALL_TASKS

Purpose: Listing all tasks in the system. The child can see a filtered or full list depending on usage.

Example Usage:

```
LIST_ALL_TASKS
```

- Lists all tasks without any filter.

```
LIST_ALL_TASKS D
LIST_ALL_TASKS W
```

- Additional parameters for filtering daily (`D`) or weekly (`W`).

3.3 LIST_ALL_WISHES

Purpose: Listing all wishes in the system (approved, pending, or rejected).

Example Usage:

```
LIST_ALL_WISHES
```

3.4 TASK_DONE

Purpose: The child indicates that a specific task is completed.

Example Usage:

```
TASK_DONE 101
```

- The child marks task `101` as done.

After the child marks the task as completed, the parent/teacher must still approve it (using `TASK_CHECKED`) for the child to receive points.

3.5 TASK_CHECKED

Purpose: The parent or teacher approves (checks) a task that has been marked `TASK_DONE` and assigns a rating (1–5 stars).

Example Usage:

nginx
KopyalaDüzenle
```
TASK_CHECKED 101 5
```

- Approves task `101` and gives it a 5-star rating, triggering the points to be added to the child's total and affecting their average rating/level.

Parameters:

1. Task_ID
2. Rating (Integer, between 1 and 5)

3.6 ADD_WISH

Purpose: The child submits a new wish (product or activity).

```
ADD_WISH1 W102 "Lego Set" "Price:150TL, Brand:LEGO"
ADD_WISH2 W103 "Go to the Cinema" "Price:100TL" 2025-03-07 14:00
2025-03-07 16:00
```

- `ADD_WISH1` might denote a product wish (e.g., “Lego Set”).
- `ADD_WISH2` might denote an activity wish (includes date and time range, e.g., “Go to the Cinema”).

Parameters:

- Wish_ID (e.g., `W102`)
- Wish Title (String)
- Description / Extra details (String)
- date/time if it’s an activity.

3.7 ADD_BUDGET_COIN

Purpose: The parent or teacher adds extra points/coins to the child’s account.

Example Usage:

```
ADD_BUDGET_COIN 50
```

- Adds 50 coins/points to the child’s current budget.

Parameters (suggested):

- Child_ID (if needed to specify which child)
- Amount of coins/points to be added (Integer)

3.8 WISH_CHECKED

Purpose: The parent approves or rejects the child’s wishes.

Example Usage:

```
WISH_CHECKED W102 APPROVED 3
WISH_CHECKED W103 REJECTED
```

- `WISH_CHECKED W102 APPROVED 3`: Approves wish `W102` and sets it to become active at level 3 (i.e., the child must be at least level 3 to fulfill this wish).
- `WISH_CHECKED W103 REJECTED`: Rejects wish `W103`, removing it from the child’s wish list.

Parameters:

1. Wish_ID
2. Approval Status (e.g., `APPROVED` / `REJECTED`)
3. (Optional) Level at which the wish becomes available.

3.9 PRINT_BUDGET

Purpose: Prints the current total points/coins of a specified child.

Example Usage:

```
PRINT_BUDGET
```

3.10 PRINT_STATUS

Purpose: Prints the child's current level or overall performance status.

Example Usage:

```
PRINT_STATUS
```

4. ADDITIONAL NOTES

- **Parameter Format**:
  - Text fields should be enclosed in quotes (`"..."`) in the input file.
  - Date/Time Format: Use `yyyy-MM-dd` for dates and `HH:mm` for times.
- **Data Persistence**:
  - Task and wish data should be read from and written to a file so that the application maintains state across sessions.