ICS 32 final Project: G.P.K

Overview

For the Final project, I created an Objective Management Tool.

The technologies (Packages) I would be using includes but is not limited to : tkinter (For GUI); smtplib; matplotlib; pandas;etc...

This Program which I named GPK, is designed to provide a S.M.A.R.T.E.R objective and Key Results System (a.k.a: OKR). GPK is specialized in the **E** and **R** criteria: Evaluation and Reward. In addition to allowing the user to add, tag, delete, complete tasks, GPK will allow the user to keep track of their progress working toward their higher objective. More importantly, the user will be rewarded with virtual tokens that can be consumed upon completing tasks, which serve as an rewarding system to motivate the users to strive after their goals while maintaining the balance between their work and entertainment.

Functionality Demo: (Since to be able to demonstrate the full functionality of GPK, data are required, so I recommend the grader watching this part before testing out the program, which is very unlikely to work since there is no existing account)
https://www.youtube.com/watch?v=C9s1OD9Ljz4

Supported Features:

Phase1 (Basic)

1. Syntax Rules Design
    a. Design a Task-Code Protocol that is unique for every task and provides information about task type (Special or Recursive) and Category (Academic or Professional).
2. A User Management System
    a. Allows users to create accounts and set passwords
    b. Prompt Login Interface where the identity is verified
3. Basic Task Management (The to-do list)
    a. Manually [add] a task to the 'Task-Today' to-do list
    b. Manually [delete] a task to the 'Task-Today' to-do list
    c. Display the tasks in the to-do list
    d. Manually [Complete] a task to the 'Task-Today' to-do list
    e. Allow the user to earn credits
4. Recursive Tasks (Tasks that occurs recursively)

    a. Manually [Create] a Recursive Task Template
    b. Manually [Delete] a Recursive Task Template
    c. Manually [Configure] a Recursive Task Template (On which days, etc)
5. Token Store:
    a. Design a formula to determine the reward after completing a task based on the tasks' time consumption, difficulty and tardiness.
    b. Design a formula that calculates the 'appropriate' probability for different levels of PRIZEs given the expected frequency of receiving.
    c. Allows the user to customize their Prizes' content and expected frequency of receiving.
    d. Allow The user to spend their tokens and draw lotteries that contain the prespecified Prizes with probability based on part c.
6. Basic Cascino:
    a. Allow the user to play game 'Big Dice' as an alternative way to earn token
7. Inventory:
    a. A Place where the Prize is stored after been drawn
    b. Allow the User to [USE] an item
    c. Allow the User to Trade an item for discounted token
    d. Keep a record of the Inventory whenever an item is Added/USEd/Traded
8. Archive
    a. Record a task's information when completed.
    b. Allow the user to browse the completed task records by date
9. Basic Daily Update:
    a. Everyday, load tasks from Predefined Recursive Tasks.
    b. Update the dates in Archive's record

Stage 2 (Advanced)

1. Loading
    a. Allow the user to import Weekly Plans which is a doc file following a specific template. (Protocol)
    b. Allow the User to Manually add an objective toward a week's plan
2. MeisterTasks Extension
    a. Allow users to Sync with MeisterTask when a task is created.
    b. Allow users to Sync with MeisterTask when a task is completed from MeisterTask
    c. Allow user to Plan a whole week's Plan using MeisterTasks' email features
3. Feedback System
    a. Provide a Progress Bar of an Objective when its corresponding task is completed
    b. Provide a Progress Overview of all Objectives of the week
    c. Devise a Grading Scheme where a score reflecting the total completion of a week's objective
    d. Provide a Visual Representation of the Productivity
4. Statistical Inference:
    a. Allow the user to visualize the productivity of all time by day,week,Month,Season

    b. Allow the user to visualize the time used in different categories
5. Generate a doc file report containing the following information
    a. Weekly Objective Completion progress (Part 2-3)
    b. Archive Logs of task completion (Part 1-8)
    c. OKR Stats (Part 2-4)
    d. Inventory History (Part 1-7)
    e. Weekly Evaluation (Part 2-3)
6. Advanced Task Management (The to-do list)
    a. Be able to complete tasks by SYNCing with MeisterTask
7. Advanced Cascino:
    a. Allow the user to play game 'Black Jack' as an alternative way to earn token
    b. Allow the user to play game 'Slot Machine' as an alternative way to earn token
8. Advanced Daily Update:
    a. Load tasks from Predefined Weekly Plan into the Todo List
    b. Sync With MeisterTask of today's Tasks
    c. Update the Weights of all Tasks to meet the Design (Part 2-3)
    d. Update Probability based on Average Productivity (Part 1-5)
9. DashBoard
    a. Show Weekly Obejctive's Progress
    b. Show Time Distribution for the past 14 days
    c. Show Evaluation Projection to keep track of Productivity
    d. Show To-Do List
    e. Show Inventory

Stage 3 (GUI)

1. Register and Create a new account
2. Login and Password Verification
3. Add a task
4. Delete a task
5. Complete a task
6. View Tasks and Modify Existing Tasks

Team Work:

Hua MinLiang:

- **Built-in Terris Game under [Menu>>Store]**

Leonidas Liao :

- **GPK GUI Structure**
- **GPK GUI Todo List**

- **GPK Basic Functionalities of Stage1-2**