Project Design & Logic: Life Manager for Students

*This document outlines the full structure, design plan, and logic behind the merged console application that combines a Task Manager and an Expense Manager for university students. The goal is to offer a simple, console-based life management tool that is easy to understand, build, and extend.*

**Project Overview**

*The Life Manager is a unified console application designed for university students. It allows users to register/login, manage tasks (like homework, assignments, deadlines) and track expenses (like food, books, utilities). The program stores everything using simple text files and avoids object-oriented complexity, sticking with procedural programming.*

**Key Features**

**1. Account System**

* *Login and registration using a single 'accounts.txt' file.*
* *Each account is protected by a username and password.*
* *User-specific files: '<username>\_tasks.txt' and '<username>\_expenses.txt'.*

**2. Task Manager Module**

* *Add a new task with title and due date.*
* *View a list of all tasks (with status).*
* *Mark tasks as done.*
* *Remove tasks by index.*
* *Tasks stored in format: Title|DueDate|[Status]*

**3. Expense Manager Module**

* *Add an expense with name, category, amount, and date.*
* *View all expenses.*
* *Search/filter expenses by keyword or category.*
* *Calculate total expenses.*
* *Expenses stored in format: Name|Category|Amount|Date*

**4. Unified Application Flow**

* *User logs in once, then sees a main menu:*
* *1. Task Manager*
* *2. Expense Manager*
* *3. Exit*
* *Each module loads, displays, and saves user-specific data to text files.*

**Logic Design (Simple & Professional)**

*The application starts by asking the user to log in or register. Once authenticated, the system creates/uses two files for the user: one for tasks and another for expenses. From the main menu, users choose to manage either tasks or expenses.*

1. *Start program, display login/registration screen.*
2. *Authenticate user, retrieve username.*
3. *Create filenames '<username>\_tasks.txt' and '<username>\_expenses.txt' if not present.*
4. *Load both task and expense data into memory (vectors).*
5. *Show main menu: Task Manager / Expense Manager / Exit.*
6. *Inside each manager, show sub-menus (e.g., Add Task, Show Expenses, etc.).*
7. *On every operation, write changes back to the respective text file.*
8. *Loop continues until user exits.*

**Final Notes**

*This merged project is structured to be beginner-friendly while still having useful features. It's modular, text-based, and can be expanded later with advanced capabilities like GUIs or export tools. It's an ideal first real-world project for students.*