

Problem statement

Software Requirement Specifications

Expense Manager

A company has decided to create an online **Expense Manager** that will keep track of purchases made through a credit card by a user. The user interface used will be command prompt and back-end implementation will be in Python.

You have been asked to **design** & **implement** the backend application only.

**Functional Specification:**

**1.** Through the frontend, the users can **register** only one credit card each, with the following information:

1. User name
2. Credit card number [assume the number of 4 digits for convenience]
3. Credit limit

*The frontend will feed the* ***registration information*** *into the* ***backend*** *where the registration information is stored.*

**2.** Once the registration is done, all credit card transactions done by users will be **automatically captured** when the card is swiped. The swiping system is implemented **elsewhere**. The captured information will be:

* Amount
* Date
* Category
* Subcategory
* Credit Card number

If the credit card number does not exist in the backend, the transaction is **not** stored in the backend application.

***The card swiping system will feed the information of the transaction to the backend. You should implement the swiping system as a function/class that randomly swipes a card for testing.***

A sample transaction could be:

|  |  |
| --- | --- |
| Amount | : 500 |
| Date | : 20/7/2010 |
| Category | : Vehicle |
| Subcategory | : Fuel |
| Credit Card | : Card number |

**3.** From the frontend, users will be able to **view** reports of their usage in **different** formats. The reports will be **generated** by the **backend**.

Functional Requirements:

Your backend application should support the following features:

1. Registration & Deregistration of credit cards
2. Swiping system
3. Report generation of credit card usage by a user. Reports can be

* All transactions done between two dates
* All transactions that fall within a specified amount range
* All transactions are done in a category
* Total amount spent on a category
* Transactions sorted on amount

More reports may be added in the future

Expectations:

* Create a few users & demonstrate the usage of the system by entering a few transactions for the users. A transaction of a credit card that is not registered in the backend should be rejected with a suitable message
* Display all the aforementioned reports
* The application should be extensible enough to allow the addition of more reports, etc. without much impact on the existing code

General Instructions:

* This Project is aimed at testing the hands-on skills and programming attitude to a practical problem.
* The program will be evaluated on the following parameters.
  1. Data structure design
  2. Modularity: The code shall be properly modularized and divided into different files as required.
  3. Implementation
     + Functionality implemented
     + Quality of code (Good Programming Practices)
     + The delivered code should compile without any build errors right-away. The Evaluator will use your Project to build the code and if the build fails, the submission will be rejected outright.
  4. Readability: The code shall be easily readable and follow proper alignments and structure
  5. Portability: The code shall be completely portable and independent of the platform you are developing on.
  6. Optimization: The code shall be optimized to the extent possible. Take care of obvious cases like reducing the number of files writes, avoiding loss of data on unexpected program crashes.
* Consider non-functional requirements within the scope of the Problem Statement. But the primary focus would need to be on Engineered Code.
* Provide a Design document.
* Provide Test Data that you have used to test your program.
* Please report bugs in your program by yourselves in an Excel sheet.
* Packaging and delivery of the implementation are as important as the correct implementation.