**INTRODUCTION**

In this project we are creating an expense tracker application. This application will help users track, categorize and get insights from their expenses.

**FEATURES**

CATEGORY MANAGEMENT:

[C] User should be able to define categories for expenses

[C] Users should be able to use predefined categories like groceries, utilities, entertainment

[R] User should be able to view categories available

EXPENSE MANAGEMENT:

[C] User should be able to add an expense by specifying details such as the amount, date, category, and description

[U] Users should be able to modify an expense

[D] User should be able to delete an expense

[R] User should be able to view expenses as a list of all recorded expenses in a clear and organized format

[R] User should be able to view expenses for specific categories

FILE HANDLING:

Implement file handling for saving expenses to a CSV file and loading them upon starting the application

Ensure CSV file persists between sessions

ERROR HANDLING – FILE OPERATION HANDLING:

Handle errors related to file reading and writing operations, ensuring that invalid file paths or formats don’t crash the application

ERROR HANDLING - INPUT VALIDATION:

Implement robust error handling for user input errors (e.g., invalid dates, negative  
amounts and category). Ensure that invalid inputs do not crash the application and that users receive appropriate error messages

**ARCHITECTURE**

<how classes interact and how project is structured>

Core Classes:

**Class Expense:** object to hold individual expenses

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Default | Validation |
| amount | Float | 0 | Non-Negative Float |
| category | Class Category Object | Misc | <> |
| description | String | NA | String |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Arguments | Return | Logic |
| init | self, amount, category, description |  |  |
| get\_amount | self | amount as float |  |
| get\_category | self | category as string |  |
| get\_description | self | description as string |  |

**Class Category:** object to hold categories used to track objects

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Default | Validation |
| name | String | Misc |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Arguments | Return | Action |
| init | self, name, description |  |  |
| get\_name | self | name as string |  |

**Class ExpenseTracker:** main class that handles overall expense tracking functionality

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Default | Validation |
| expenses | List |  |  |
| categories | Set | “grocery”, “travel” |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Arguments | Return | Logic |
| add\_expense | self, expense |  |  |
| delete\_expense | self, user\_index |  |  |
| modify\_expense | self, expense,  user\_index, new\_amount, new\_category, new\_description |  |  |
| add\_category | self, category |  |  |
| view\_expenses | self |  |  |
| view\_expenses\_category | self, category |  |  |

**TESTING**

**CHALLENGES**

**FUTURE ENHANCEMENTS**

* EXPENSE CREATION: Add option to specific date when creating expense
* REPORT GENERATION: User should be able to generate reports showing total expenses for specific periods (daily, weekly, monthly) and per category