Chapter One

Problem Analysis

The problems associated with this particular project includes the following

1. UNDERSTANDING DATASET: The understanding of the provided dataset is very paramount in understanding the problem and providing a suitable solution and best tools to use for solving the problem.
2. PARSING AND READING DATASET: The parsing and reading of the dataset poses another problem as the dataset would have to be loaded and read by the computer for manipulation, calculating and extracting features from it. Another problem under the parsing and reading of dataset is the choice of the data type/structure to be used to store the dataset for the computer.
3. UNDERSTANDING THE STATISTICAL FUNCTIONS: Understanding the statistical functions that have been asked to be calculated (reference to the instructions given) is also paramount, no understanding of their formula requirement can make the calculated result wrong.
4. PERFORMING STATISTICAL FUNCTIONS: Performing the different statistical functions on user transactions dataset, functions which include functions that calculates and returns the minimum and maximum amount of a particular user, functions that calculates and returns the location centroid of any user based on their transaction locations, e.t.c.
5. DESIGNING A USER INTERFACE: Designing a user interface that makes use of our both modules and provides users with a textual interactive interface to communicates with the modules imported whereas making sure to catch all exceptions and not break.

Chapter Two

Design Decisions

Design Decisions used in this project was formed primarily from reading the instructions from the Programming concepts and practice tasks PDF document. As the project is divided into three modules , we shall be taking the design decision per module of concerns.

**Module One : load\_dataset.py**

The Module is responsible for opening our datasets using the python inbuilt open keyword, extracting the data values and formatting it and returning a dictionary storage with the user\_id’s as keys and their respective user transactions details and different values stored in lists respectively.

Decisions:

1. To hold the Datasets in a storage such as each user id can have all their transaction details attributed to them we have to make use of the python dictionary data type as this gives us the feature.
2. The datasets has a multiple lines(rows) of transactions which have the following features, the user\_id for the user that made the transaction, the transaction id, the description of the transaction, the amount of the transaction, the amount of the transaction, the x and y coordinate of the transaction location and a boolean value to represents the fraud status of that the transaction. Design decision here was to put each transaction and it features in a list , then put all the individual list with a the same user\_id under a general list and attribute it as value to the user\_id key in the dictionary been returned.

Pseudocode:

def load\_data():

* Opens Datasets in Read mode , uses the with keyword to keep file buffer open while performing data extraction.
* Remove new lines “\n” from each line in dataset
* Create empty dictionary variable for RESULTS
* Loop through each line and split every item in a line by using colons as delimiter and store it
* Check IF first index in the stripped items list(i.e The user\_id) exists in RESULTS.

IF Exist it would append to the value of the key(user\_id) in the RESULTS variable which is a list that hold other lists which are containing every column feature of each line related to the particular user\_id.

* ELSE it would create the user\_id as a key in the RESULTS variable and create a list as a value that would hold another list populated with the that particular line columns features all specific to the user\_id found in their first index
* RESULTS variable which is a dictionary that would be holding every user\_id as keys with a value of a list holding different list with each transaction features in them.

**Module Two : user\_statistics.py**

This module is responsible for importing the load\_dataset module and using the returned data with the multiple functions within it for performing different user\_statistics which are.

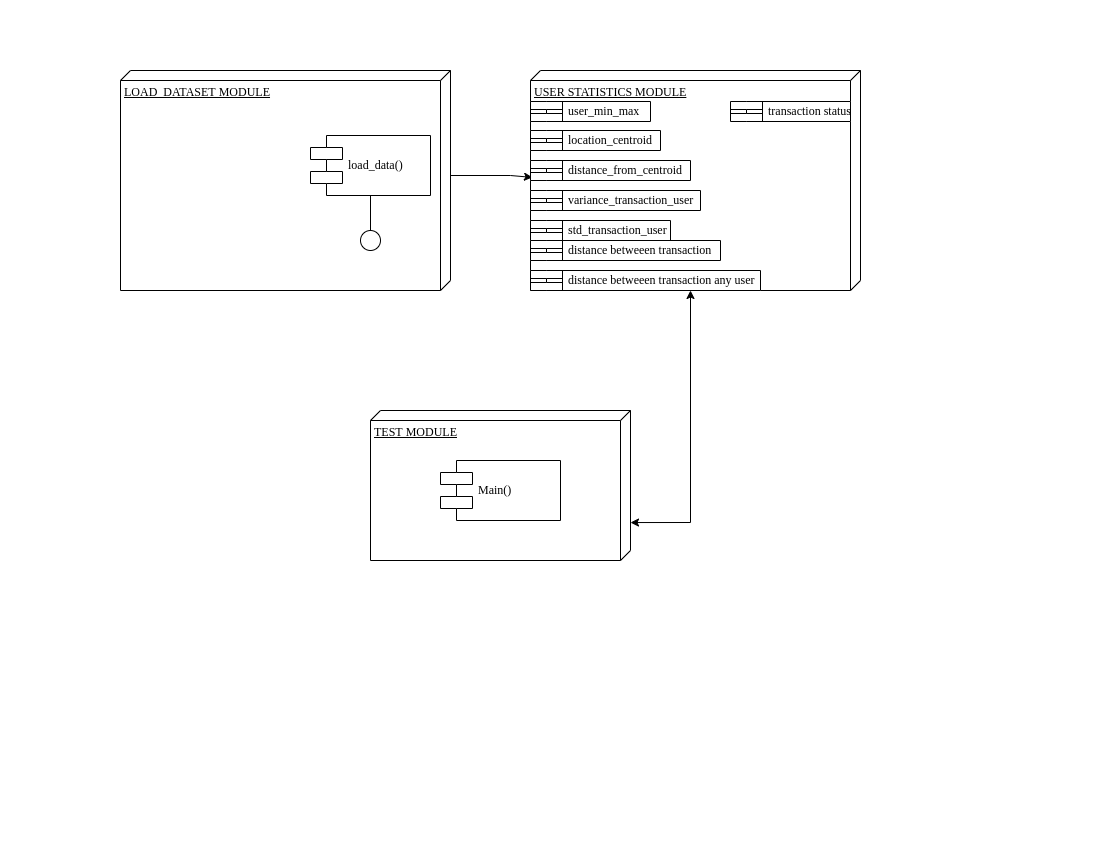
1. Function to return the minimum and maximum transaction amount of a particular user.
2. Function to return the location centroid of a particular user based off on the user transactions location
3. Function to return the difference in location distance between a particular user transaction and the user location centroid.
4. Function to return the variance of a particular user transactions amount
5. Function to return the standard deviation of a particular user transactions amount
6. Function to return the transaction fraud status of a particular user transaction.
7. Function to return distance between two transaction of a particular user
8. Function to return distance between two transaction of different users

Decisions:

Design Decision performed in this module was to make each module very much independent of each other , and to achieve this each module had to call the load\_data function from the load\_dataset module imported and make use of the returned data for it calculations instead of making use of global call to load\_data function from load\_datasets module

Chapter Four

Modules Relationship



Chapter Five

**APPLICATION EXECUTION STEPS**

Our code base projects exists of Three python file, which includes the load\_dataset.py, user\_statistics.py and the test\_module.py. The execution of this program requires you have the following requirement

* Python 3.4+

The main file to execute is our test\_module.py as this file makes the textual interface for our users to interact with the other modules and perform several task. Run the highlighted text below in your console or terminal in the directory where our project file is stored.

> **python test\_module.py**