# -Audit Case: Tooth Fairy

Australian Taxation Office

# Part 1: Design

## Pseudocode for the auditing system:

print “ Audit Case: Tooth Fairy”

print “ Developer name: Bikash Shrestha Student Id: 30377519”

print menu:

1. Print the statistics of given csv file

2. Export the list of children to a file who haven’t lost any teeth yet

3. Display a graph to show total claims in each state

4. Display a graph that compares two states based on average number of teeth lost in the particular states.

5. Exit the program.

user\_choice = take users’ input that points to respective menu item

if user\_choice = 1:  
 total\_childrenCalculate total number of children in the text file given:

lines = separated records of the file

total\_children = length of lines -2

print “ Total children in the file = ” + total\_children

sum\_of\_claims = 0

for each line in lines:

sum\_of\_claims += line[number \_of\_teeth\_lost]

print “ average number of tooth claims =”+ sum\_of\_claims/total\_children

no\_teeth\_lost\_children = empty list

for each child in lines

If total number of teeth lost != 0

append child to no\_teeth\_lost\_children

print( “ Number of children who never lost a teeth = ”+ len(no\_teeth\_lost\_children))

all\_teeth\_lost\_children = empty list

Filter all the children who have lost all of their baby teeth:

For each child in lines:

if child[total\_number\_of\_teeth\_lost]= 20

append child to all\_teeth\_lost\_children

print “ Number of children who lost all baby teeth = ”+count of all\_teeth\_lost\_children

total\_expenditure = 0

for each child in lines:

if a child has lost one tooth only

total\_expenditure +=1

else if child has lost more than one tooth

total\_expenditure += 0.5

else

continue

print total\_expenditure

else if user\_choice = 2:

file\_name = take filename from user

file = open file with file\_name

file.write(no\_teeth\_lost\_children)

else if user\_choice = 3:

state\_list = empty list

For each child in lines:

If state of child is not in state\_list

append state of child to state\_list

claims\_list = empty list

For each state in state\_list:

append 0 to claims\_list

For each child in lines

i=0

For each ith state in state\_list:

If state of child = state\_list[i]

claims\_list[i] += 1

i+=1

graph = create\_bar\_graph state vs claims

save graph

else if user\_choice = 4:

first\_state= take input for first state name

second\_state = take input for second state name

lost\_tooth\_sums = [0,0]

claims\_list= [0,0]

for each child in lines

if state of child = first

first\_lost\_tooth\_sum += total number of teeth lost by the child

claims\_list[0]+=1

else if state of child = second

second\_lost\_tooth\_sum += total number of teeth lost of the child

claims\_list[1]+=1

average\_list[0] = first\_lost\_tooth\_sum/claims\_list[0]

average\_list[1] = lost\_tooth\_sums[1]/claims\_list[1]

graph = create bar graph of states vs average\_list

save graph

else if choice = 5

exit

else

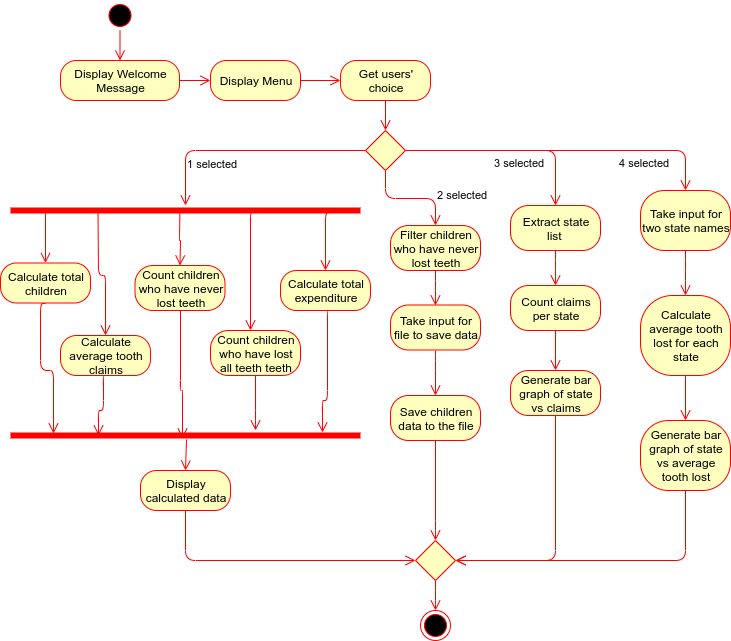
pass

## Test cases:

|  |  |  |  |
| --- | --- | --- | --- |
| s.n | Case | Data | Expected Result |
| 1 | Test if splitLinesToList() function in main.py returns a list. | Lines of Addresses.csv file. | splitted\_lines: splitted list of each lines of the  file. |
| 2 | Test if getStateList() function returns valid list of states. | List of Splitted lines containing data of each record from the csv file. | List of states which includes states of Australia: NSW, QLD, VIC, etc. |
| 3 | Test if getLostToothList() function returns a list of the total number of teeth lost. | List of Splitted lines containing data of each record from the csv file. | List of numbers that represent the number of teeth lost by children |
| 4 | Test if getTotalExpenditure() calculates the total expenditure. | List of numbers representing total number of teeth lost by each children. | Expenditure value > 0 |
| 5 | Test if exportDetails() function saves data to a file. | User entered filename and list of children who never lost a teeth. | The list of children who never lost a teeth are written in a file with supplied filename |
| 6 | Test if displayClaimsByState() generates a graph of claims vs state. | List of states and list claims per state. | Graph of Claims vs State |
| 7 | Test if compare2StatesByAverageToothLost() generates a comparative graph of two states. | User entered two states and average tooth lost in each state | Comparative graph of state vs average teeth lost. |

## 

# Part 2: Activity Flowchart



The above activity diagram shows the events and activities of the auditing system. The system first show the welcome message: name of organization, audit case, name of developer, and student id of the developer. Then the system displays a menu with choices: display statistics, export details to a file, display claims per state, and compare two states based on average number of teeth lost. When display statistics option is chosen, the system will calculate statistical information: total number of children in the given file, average tooth claims, the number of children who haven’t lost any teeth yet, the number of children who have lost all of their baby teeth, and total expenditure and then the calculated data is displayed. The system asks for a file-name when export details option is selected. The data of children who have never lost teeth, is filtered from the given file and then saved to the file with supplied file name when this option is chosen. The display claims per state option will count the number of claims per state and generate a bar-graph of state vs claims. And, compare two states option will take input for name of two states and then generate a graph of state vs average tooth lost per state. The graphs generated are saved by the system in the directory of the main program.

# Part 3

The results of test cases from above:

Test Case 1:

Action:.

Result:

## 

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case no. | Test Steps | Status | Reason |
| 1 | Run test\_splitLinesToList() of tester.py. | Success | Function returns a splitted list of each line of the csv file. |
| 2 | Run test\_getStateList() of tester.py. | Success | The list returns a test State of Australia. |
| 3 | Run test\_getLostToothList() of tester.py. | Success | The list returned by the function contains numeric values. |
| 4 | Run test\_getTotalExpenditure() of tester.py. | Success | The function returned expenditure greater than 0. |
| 5 | Run test\_exportDetails() of tester.py. | Success | The function created a file with the data of children who never lost a tooth. |
| 6 | Run test\_displayClaimsByState() of tester.py. | Success | The function created and saved a graph. |
| 7 | Run test\_compare2StatesByAverageToothLost() of tester.py | Success | The function created and saved a graph. |

## Screenshots of testing:

