For the system test cases, we will use files named input/landmarks\_sample.csv

and input/trails\_sample.csv.

input/landmarks\_sample.csv:

LANDMARK\_ID,DESCRIPTION,TYPE

L01,Park Entrance,Location

L02,Entrance Fountain,Fountain

L03,Waste Station 1,Pet Waste Station

L04,Entrance Restrooms,Restroom

L05,Overlook 1,Overlook

L06,Rock Formation 1,Rock Formation

L07,Overlook 2,Overlook

L08,Overlook Restrooms,Restroom

L09,Waste Station 2,Pet Waste Station

L10,Hidden Gardens,Gardens

L11,Campsite 1,Campsite

L12,Campsite Restrooms,Restroom

input/trails\_sample.csv:

LANDMARK\_ID,LANDMARK\_ID,DISTANCE

L01,L02,3013

L01,L03,1046

L01,L04,1179

L02,L10,3613

L03,L05,4204

L04,L09,2311

L05,L06,1039

L06,L07,2912

L07,L08,1891

L11,L12,1066

| Test ID | Description | Expected Results | Actual Results |
| --- | --- | --- | --- |
| **testInvalidInput**  **(Exceptions/unexpected input- trying to load a file that does not exist)** | **Preconditions:**   * TrailManagerUI has been loaded successfully and prompts “Please enter a landmark file and trail file separate by a space (landmark.csv trail.csv) or type 'quit' to quit”   **Steps:**   1. User specifies a file that does not exist. | Trail Manager prompts “File invalid  Please enter a landmark file and trail file separate by a space (landmark.csv trail.csv) or type 'quit' to quit” |  |
| **testLoadValidInput**  **(Equivalence class- loads valid input from file)** | **Preconditions:**   * TrailManagerUI has been loaded successfully.   **Steps:**   1. User inputs input/landmarks\_sample.csv input/trails\_sample.csv | Files load into the Trail Manager program with no exceptions. |  |
| **testInputTrails**  **(Boundary value- adding at the lower bound)** | **Preconditions:**   * testLoadValidInput passes   **Steps:**   1. 2Choose to view a report of proposed landmarks where first aid stations should be located. 2. Input 1 as the minimum number of trails. | The program reports the correct number of intersecting trails. |  |
| **testInvalidInputTrails**  **(Exceptions/unexpected input- trying to load an invalid number of trails)** | **Preconditions:**   * TrailManagerUI has been loaded successfully. * Input-landmark.txt and input-trails.txt have been loaded successfully.   **Steps:**   1. Choose to view a report of proposed landmarks where first aid stations should be located. 2. Input “0” as the minimum number of trails. | The system reports “Number of intersecting trails should be greater than zero.” |  |
| **testInvalidLandmark**  **(Exceptions/unexpected input- trying to load a landmark that does not exist)** | **Preconditions:**   * TrailManagerUI has been loaded successfully. * Input-landmark.txt and input-trails.txt have been loaded successfully.   **Steps:**   1. Choose to view a report of proposed landmarks where first aid stations should be located. 2. Input “3” as the minimum number of trails. 3. Enter “xx” as the landmark ID. | The program outputs the correct landmarks with 3 intersecting trails.  After inputting “xx” as the landmark ID, the system reports “The provided landmark ID xx is invalid for the park.” |  |