Testing

# Unit Tests(mocks)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test description | Pre-Conditions | Test Steps | Test Data | Expected Results |
| 1 | Check JSON Integrity | User in home page | 1. Open app 2.land in homepage | none | News from website should appear on the screen |
| 2 | Check current GPS location | GPS permission should be given , GPS should be on, GPS should be the location setting | 1. Registered User opens app 2. Presses ‘Scenarios’, and enters the Scenarios Activity  3. Should see the closest scenarios by proximity | 1.Location 2. Pressing the ‘scenarios’ activity | Scenarios should be sorted by proximity to the device |
| 3 | Check GPS delta (=distance from scenario in JSON) | GPS permission should be given , GPS should be on, GPS should be the location setting | 1. Registered user opens app  2. Presses ‘Scenarios’ and enters the Scenario Activity | 1.Location,  scenarios,  2. Pressing the ‘scenarios’ activity | Scenario location should be the correct distance away from the device |
| 4 | Check Scenario correctness | User is registered,  Location setting is ‘city’ | 1. User opens app  2. User selects scenarios and gets GPS from server  3. Check that the scenarios in his city match the ones in the server | 1. We should have a pre-written scenarios list 2. We pull said list from server  3. check json received from server against our own for inconsistencies | 0 inconsistencies |

# Integration Tests

1. Check if approved scenario from scenarios list is updated as approved scenario in list scenarios activity
2. Check if mocked local GPS location is near real GPS scenario



1. Check if real local GPS location is near mocked GPS scenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test description | Pre-Conditions | Test Steps | Test Data | Expected Results |
| 1 | Check if approved scenario from scenarios list is updated as approved scenario | User is registered, active scenarios | 1. Registered User opens app 2. User checks active scenarios 3. User approves scenario   4. the scenario id sent to the server, who marks the chosen scenario and returns updated list and also marks the chosen scenarios. | accept scenario from user,  active scenarios from the server | The chosen scenario will be marked as approved scenario in the scenarios list. |
| 2 | Check if mocked local GPS location is near real GPS scenario | None, connected to server(for checking scenarios) | 1. Registered user opens app. 2. User got mocked GPS location from Emulator 3. User checks for active scenarios and sees if they close to his mocked location | Mocked location for user, real location from server(scenario) | The scenarios that we get in the Scenarios list are close to user's mocked location |
| 3 | Check if real local GPS location is near mocked GPS scenario | GPS permission should be given , GPS should be on, GPS should be the location setting | 1. Registered user opens the app   1. Checks if the GPS is on.   Checks if the setting is based on GPS location(and not city)   1. Checks the active scenarios and sees if mocked GPS location of scenarios are close to the user's real GPS location | real location from server(scenario),  Mocked location for user | The scenarios that we get in the Scenarios list are close to user's real location |

# System Tests



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test description | Pre-Conditions | Test Steps | Test Data | Expected Results |
| 1 | Scenario #1 | User not registered with same mail and phone. | 1. User opens app and is not a registered user. 2. User registers for system. 3. User tries to log in but cannot do so because they were not approved by administrator. 4. User does not see the logged in tabs from the main screen. 5. User tries to register again with same e-mail/phone but cannot do so because the mail should be unique. | Register data | User cannot log in |
| 2 | Scenario #2 | User not registered with same mail and phone. | 1. User opens app and is not a registered user. 2. User registers for system. 3. Admin approves registration immediately, 4. User turns on settings and sets locations by GPS 5. User receives list of scenarios and picks one near him. 6. User files report for pictures. | registration,  click Scenario list, fills report | User has been approved by admin.  Users sets setting.  User upload a new report. |
| 3 | Scenario #3 | Registered user.  User is on GPS location mode in settings | * 1. Pre-registered user opens app, and is greeted by main screen for logged in users.   2. Changes GPS settings from ‘GPS’ mode to ‘city’ mode.   3. Opens ‘Scenarios’ screen and only sees scenarios in their city.   4. Picks one scenario and accepts it.   5. Uploads video of scenario. | GPS location settings, click scenarios list,  Chosen scenario, video | User updated his location setting to city, will get scenarios by city location mode. Upload a report with a video. |