
Test Document

for

surakshIIT

Version 0.1

Prepared by

Group Name: **GITHUB SPAMMERS**

Course: **CS253**

Mentor TA: **Mr. Aman Aryan**

Date: **03-04-2022**

Group Details

Name	Roll No.	Email ID
Aarchie	200004	aarchi20@iitk.ac.in
Arpit Kumar	200189	arpitk20@iitk.ac.in
Harsh Jain	200412	harshj20@iitk.ac.in
Kajal Deep	200483	kajald20@iitk.ac.in
Kuldeep Singh Chouhan	200530	kuldeepsc20@iitk.ac.in
Mannem Rishwan Reddy	200562	mannemr20@iitk.ac.in
Rohit	200813	rohitg20@iitk.ac.in
Sandeep Kumar Bijarnia	200856	sandeepb20@iitk.ac.in
Udit Prasad	201055	uditp20@iitk.ac.in
Yash Raj Mittal	201148	yashrajm20@iitk.ac.in



CONTENTS	II
REVISIONS	II
1 INTRODUCTION	3
2 UNIT TESTING	4
3 INTEGRATION TESTING	25
4 SYSTEM TESTING	31
5 CONCLUSION	35
APPENDIX A - GROUP LOG	36

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Udit Prasad Arpit Kumar Aarchie Rishwan Reddy Yash Raj Mittal	First Draft	03-04-2022

1 Introduction

Test Strategy:

We are using Manual testing to test our software.

Testing period:

We were doing unit testing as soon as we make that unit
Integration and System testing was done once all the units were completed.

Testers:

Developers were itself the testers

Coverage Criteria:

We mainly used Branch coverage to test this software which is a type of functional coverage.

Tool used:

We have used Postman to test our software because of various reasons like:

- It has a user friendly interface from where we can easily send any HTTP request whether it be POST, GET, etc.
- Stores information in a well designed manner

2 Unit Testing

1. Register New User/Security

Unit Details: [Register, Post]

Test Owner: [Arpit Kumar]

Test Date: [03/20/2022] - [03/21/2022]

Test Results:

Case 1: isRegistered() was returning a user

Expected behaviour - Error with HTTP_400_BAD_REQUEST

Result - HTTP_400_BAD_REQUEST

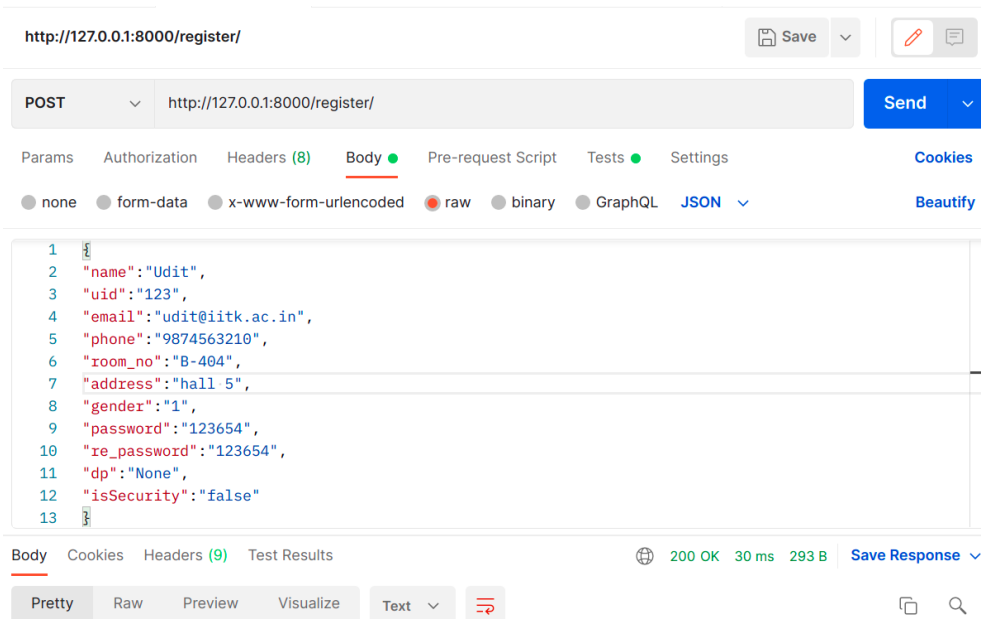
Case 2: isRegistered() was set to None

Expected behaviour - User will be registered with Response: HTTP_200_OK

Result - Same as Expected

Structural Coverage: Branch Coverage is used

Additional - Postman test result of case 2



2. Login

Unit Details: Login, post

Test Owner: [Udit]

Test Date: [03/21/2022] - [03/21/2022]

Test Results:

Case 1: If user is Registered and correct credentials are given

Expected Behaviour: Login will be successful and session will be initiated with status HTTP_200_OK

Result: Same as Expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_unauthorized

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional :

Postman test result of case 1

http://127.0.0.1:8000/login/ Save

POST http://127.0.0.1:8000/login/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2
3   "uid": "123",
4   "password": "123654"
5 }
6
```

Body Cookies (1) Headers (10) Test Results 200 OK 72 ms 437 B Save Response

2. View Lost Item

Unit Details: [LostItemView, get]

Test Owner: Arpit Kumar

Test Date: [03/20/2022] - [03/21/2022]

Test Results:

Case 1: If isLoggedIn was set to True:

Expected Behaviour: Get a list of Lost items with Response: HTTP_200_OK

Result: Same as Expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_UNAUTHORIZED

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

The screenshot shows a Postman test result for a GET request to the URL `http://127.0.0.1:8000/security/lost_item/`. The request method is GET, and the response status is 200 OK. The response body is a JSON object with the following structure:

```
1 {
2   "status": "success",
3   "data": [
4     {
5       "id": "1f49baa8-c1a7-4ba9-a4f2-a1f675ee0660",
6       "owner": "Udit",
7       "name": "cycle",
8       "item_color": "black",
9       "details": "it was of hero company with mark on left handle",
10      "last_seen": "hall 5",
11      "image": "/media/Screenshot_from_2022-02-24_23-50-28.png",
12      "lost_time": "2022-04-03T17:11:34.298000+05:30",
13      "if_found": false
14    }
15  ]
16 }
```

The response is displayed in the 'Pretty' format, showing a JSON array with one object. The object contains fields for id, owner, name, item_color, details, last_seen, image, lost_time, and if_found.

3. Add Lost Item

Unit Details: [AddLostItemView, Post]

Test Owner: Arpit Kumar

Test Date: [03/21/2022] - [03/21/2022]

Test Results:

If isLoggedIn was set to True and all the required data was sent as request:

Expected Behaviour: Add a lost item in database with Response:

HTTP_201_CREATED

Result: Same as Expected

Case 2: Else if all data was not compatible

Expected Behaviour: Error with Response: HTTP_400_BadRequest

Result: Same as expected

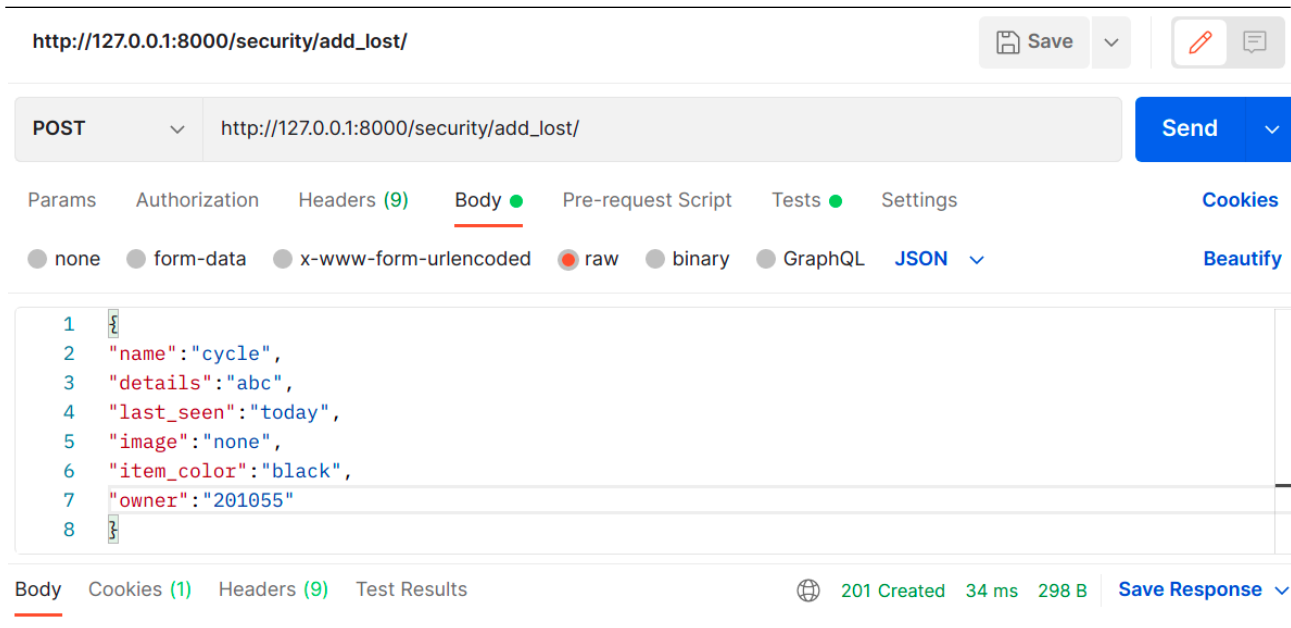
Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_unauthorized

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results



4. Delete Lost Item

[Include basic information about what was tested and what happened.]

Unit Details: [DeleteLostItemView, delete]

Test Owner: [Arpit Kumar]

Test Date: [03/21/2022] - [03/21/2022]

Test Results:

Case1: If Security is logged in:

 If required item in database

 Expected: delete lost object from database

 Response: HTTP_200_OK

 Else

 Expected: error with HTTP_400_BADREQUEST

Case 2: If security is not logged in

 Expected: Response: HTTP_401_BAD_REQUEST

Structural Coverage: Branch Coverage was used

Additional: Some of postman test results

http://127.0.0.1:8000/security/delete_lost/ Save

DELETE ▼ http://127.0.0.1:8000/security/delete_lost/ Send ▼

Params Authorization Headers (9) **Body** ● Pre-request Script Tests ● Settings Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	e8170b94-0bab-414d-84eb-3d1b9f94...			
	Key	Value	Description		

Body Cookies (1) Headers (9) Test Results 200 OK 35 ms 295 B Save Response ▼

5. View Found Item

Unit Details: [FoundItemView, get]

Test Owner: [Arpit Kumar]

Test Date: [03/21/2022] - [03/21/2022]

Test Results:

CASE1: If Security is set to logged in:

Expected Behaviour: Get a list of Lost items with Response: HTTP_200_OK

Result: Same as Expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_UNAUTHORIZED

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

http://127.0.0.1:8000/security/found_item/

GET http://127.0.0.1:8000/security/found_item/ Send

Params Authorization Headers (9) Body ● Pre-request Script Tests ● Settings Cookies

Body Cookies (1) Headers (10) Test Results 200 OK 42 ms 591 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": "success",
3   "data": [
4     {
5       "id": "512ada42-2f6e-4ea8-b2b7-0531f48fb877",
6       "who_found": "Udit",
7       "name": "cycle",
8       "item_color": "black",
9       "details": "abc",
10      "where_found": "hall 2",
11      "image": "/media/none",
12      "found_time": "2022-04-03T17:24:40.335300+05:30",
13      "if_returned": false
14    }
15  ]
16 }

```

6. Add Found Item

Unit Details: [AddFoundItemView, post]

Test Owner: [Arpit Kumar]

Test Date: [03/21/2022] - [03/21/2022]

Test Results:

If isLoggedIn was set to True and all the required data was sent as request:

Expected Behaviour: Add a lost item in database with Response:

HTTP_201_CREATED

Result: Same as Expected

Case 2: Else if all data was not compatible

Expected Behaviour: Error with Response: HTTP_400_BadRequest

Result: Same as expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_unauthorized

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

http://127.0.0.1:8000/security/add_found/ Save

POST ▼ http://127.0.0.1:8000/security/add_found/ Send ▼

Params Authorization Headers (9) **Body** ● Pre-request Script Tests ● Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● GraphQL **JSON** ▼ Beautify

```

1 {
2   "name": "cycle",
3   "details": "abc",
4   "where_found": "hall 2",
5   "image": "none",
6   "item_color": "black",
7   "who_found": "201055"
8 }
```

Body Cookies (1) Headers (9) Test Results 201 Created 19 ms 298 B Save Response ▼

7. Delete Found Item

Unit Details: [DeleteFoundItemView, delete]

Test Owner: Arpit Kumar

Test Date: 03/21/2022 - 03/21/2022

Test Results:

Case1: If Security is logged in:

 If required item in database

 Expected: delete lost object from database

 Response: HTTP_200_OK

 Else

 Expected: error with HTTP_400_BADREQUEST

Case 2: If security is not logged in

 Expected: Response: HTTP_401_BAD_REQUEST

Structural Coverage: Branch Coverage was used

Some of postman test results

http://127.0.0.1:8000/security/delete_found/ Save

DELETE ⌵ http://127.0.0.1:8000/security/delete_found/ Send ⌵

Params Authorization Headers (9) **Body** ● Pre-request Script Tests ● Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	512ada42-2f6e-4ea8-b2b7-0531f48fb...			

Body Cookies (1) Headers (9) Test Results 200 OK 32 ms 295 B Save Response ⌵

8. Add Hall Entry

Unit Details: [AddHallEntryView, post]

Test Owner: Arpit Kumar

Test Date: [03/21/2022] - [03/21/2022]

Test Results:

If isLoggedIn was set to True and all the required data was sent as request:

Expected Behaviour: Add a lost item in database with Response:

HTTP_201_CREATED

Result: Same as Expected

Case 2: Else if all data was not compatible

Expected Behaviour: Error with Response: HTTP_400_BadRequest

Result: Same as expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_unauthorized

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

http://127.0.0.1:8000/security/add_hallentry/

POST http://127.0.0.1:8000/security/add_hallentry/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "person1": "200004",
3   "person2": "200412",
4   "hall": 5,
5   "destination": "mess"
6 }
```

Body Cookies (1) Headers (9) Test Results 201 Created 31 ms 298 B Save Response

9. View Hall Entries(only those who have not yet left the hall)

Unit Details: [GetHallEntriesView, get]

Test Owner: Rishwan Reddy

Test Date: [3/23/2022] - [3/23/2022]

Test Results:

CASE1: If Security is logged in:

Expected: Get list of all Hall Entries who have not yet exited with response:

HTTP_200_OK

Result: Same as Expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_UNAUTHORIZED

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional: Some of postman test results

The screenshot shows a Postman interface with a GET request to `http://127.0.0.1:8000/security/hall_entries/`. The response is a JSON object with a status of "success" and a list of two hall entries. The response status is 200 OK, with a response time of 30 ms and a body size of 765 B.

```

{
  "status": "success",
  "data": [
    {
      "id": "0850fd2e-9929-4ca9-ae1d-026aeeade8fc",
      "person1": "Aarchie",
      "person2": "kajal",
      "destination": "room",
      "hall": "6",
      "entry_time": "2022-04-03T01:24:30.738993+05:30",
      "exit_time": null,
      "if_exited": false
    },
    {
      "id": "2276038a-bf9f-41c7-b049-a4d75016c489",
      "person1": "Harsh Jain",
      "person2": "kajal",

```

10. View All Hall Entries

Unit Details: [GetAllHallEntriesView, get]

Test Owner: Rishwan Reddy

Test Date: [3/23/2022] - [3/23/2022]

Test Results:

CASE1: If Security is logged in:

Expected: Get list of all Hall Entries with response: HTTP_200_OK

Result: Same as Expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_UNAUTHORIZED

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

http://127.0.0.1:8000/security/all_hall_entries/

GET http://127.0.0.1:8000/security/all_hall_entries/ Send

Params Authorization Headers (9) Body ● Pre-request Script Tests ● Settings Cookies

Body Cookies (1) Headers (10) Test Results 200 OK 24 ms 990 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": "success",
3   "data": [
4     {
5       "id": "1334430e-2bc4-47b7-9c0d-b30685091a5d",
6       "person1": "Udit",
7       "person2": "kajal",
8       "destination": "mess",
9       "hall": "6",
10      "entry_time": "2022-04-01T23:30:47.073823+05:30",
11      "exit_time": "2022-04-03T02:52:11.056375+05:30",
12      "if_exited": true
13    },
14    {
15      "id": "0850fd2e-9929-4ca9-ae1d-026aeeade8fc",
16      "person1": "Aarchie",
17      "person2": "kajal",

```

11. Add Hall Exit

Unit Details: [AddHallExitView, put]

Test Owner: Rishwan Reddy

Test Date: [3/23/2022] - [3/23/2022]

Test Results:

CASE1: If Security is not logged in:

Expected: Error with Response:HTTP_401_UNAUTHORIZED

Result: Same as Expected

Case 3: If security is logged in but data in request is not compatible or that entry is missing from database

Expected: HTTP_400_BAD_REQUEST

Result: Same as Expected

CASE: 2 Else:

Expected: That entry will be updated with exit time.

Response:HTTP_200_OK

Result: Same as Expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

http://127.0.0.1:8000/security/add_hallexit/ Save

PUT ▼ http://127.0.0.1:8000/security/add_hallexit/ Send ▼

Params Authorization Headers (9) Body ● Pre-request Script Tests ● Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	3c2041c3-5f95-4e35-a389-69f734a7...			

Body Cookies (1) Headers (9) Test Results 200 OK 65 ms 292 B Save Response ▼

12. Add Campus Exit**Unit Details:** [AddCampusExitView, post]**Test Owner:** Rishwan Reddy**Test Date:** [3/23/2022] - [3/23/2022]**Test Results:**

CASE1: If Security is not logged in:

Expected: Error with Response:HTTP_401_UNAUTHORIZED

Result: Same as Expected

CASE: 2If security is logged in and person1 is in database person2 is in database:

Expected: a hall entry will be created and saved to database with

Response:HTTP_201_CREATED

Result: Same as Expected

Case 3: If security is logged in but data in request is not compatible or user is missing from database

Expected: HTTP_400_BAD_REQUEST

Result: Same as Expected

Structural Coverage: Branch Coverage was used**Additional : Some of postman test results**

http://127.0.0.1:8000/security/add_campusexit/ Save

POST http://127.0.0.1:8000/security/add_campusexit/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
2 "person": "201055",
3 "destination": "kalayanpur"
4
```

Body Cookies (1) Headers (9) Test Results 201 Created 26 ms 298 B Save Response

13. Add Campus Entry

Unit Details: [AddCampusEntryView, put]

Test Owner: [Rishwan Reddy]

Test Date: [3/23/2022] - [3/23/2022]

Test Results:

CASE1: If Security is not logged in:

Expected: Error with Response:HTTP_401_UNAUTHORIZED

Result: Same as Expected

Case 2: If security is logged in but data in request is not compatible or that exitId is not in database

Expected: HTTP_400_BAD_REQUEST

Result: Same as Expected

CASE: 3 Else

Expected: Exit data will be updated with entry time.

Response:HTTP_200_OK

Result: Same as Expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

http://127.0.0.1:8000/security/add_campusentry/

PUT http://127.0.0.1:8000/security/add_campusentry/ Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	d12dd04b-38fa-4a9a-a277-a8dc2588...			

Body Cookies (1) Headers (9) Test Results 200 OK 24 ms 292 B Save Response

14. View Campus Exits(only those who have not returned yet)

Unit Details: [GetCampusExitsView, get]

Test Owner: Rishwan Reddy

Test Date: [3/23/2022] - [3/23/2022]

Test Results:

CASE1: If Security is logged in:

Expected: Get list of all Campus exits who have not yet returned with response:

HTTP_200_OK

Result: Same as Expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_UNAUTHORIZED

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

http://127.0.0.1:8000/security/campus_exits/ Save

GET http://127.0.0.1:8000/security/campus_exits/ Send

Params Authorization Headers (9) Body ● Pre-request Script Tests ● Settings Cookies

Body Cookies (1) Headers (10) Test Results 200 OK 17 ms 705 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "status": "success",
3    "data": [
4      {
5        "person": "Udit",
6        "destination": "z square",
7        "if_entered": false,
8        "entry_time": null,
9        "exit_time": "2022-04-01T23:27:32.470070+05:30",
10       "id": "5b7e1246-9c04-4ff1-b944-075741857e8a"
11     },
12     {
13       "person": "Udit",
14       "destination": "kalayanpur",
15       "if_entered": false,
16       "entry_time": null,
17       "exit_time": "2022-04-03T17:38:20.383829+05:30",

```

15. View All Campus Exits

Unit Details: [GetAllCampusExitsView, get]

Test Owner: Rishwan Reddy

Test Date: [3/23/2022] - [3/23/2022]

Test Results:

CASE1: If Security is logged in:

Expected: Get list of all Campus exits response: HTTP_200_OK

Result: Same as Expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_UNAUTHORIZED

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

http://127.0.0.1:8000/security/all_campus_exits/

GET http://127.0.0.1:8000/security/all_campus_exits/ Send

Params Authorization Headers (9) Body ● Pre-request Script Tests ● Settings Cookies

body Cookies (1) Headers (10) Test Results 200 OK 12 ms 1.09 KB Save Response

Pretty Raw Preview Visualize JSON

```

2      status: success,
3      "data": [
4        {
5          "person": "Udit",
6          "destination": "z square",
7          "if_entered": false,
8          "entry_time": null,
9          "exit_time": "2022-04-01T23:27:32.470070+05:30",
10         "id": "5b7e1246-9c04-4ff1-b944-075741857e8a"
11       },
12       {
13         "person": "Aarchie",
14         "destination": "Theme Park",
15         "if_entered": true

```

16. Entry for Non-campus Resident

Unit Details: [GetNonResidentsCampusEntriesView, get]

Test Owner: Yash Raj Mittal

Test Date: [03/23/2022] - [03/23/2022]

Test Results:

CASE1: If Security is set to logged in:

Expected Behaviour: Get a list of Non-Residents who have entered but not excited yet with Response: HTTP_200_OK

Result: Same as Expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_404_NotFound

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional Comments: [Place screenshot here](#)

17. Non Residents Campus Exit

Unit Details: [AddNonResidentCampusExitView, put]

Test Owner: Aarchie

Test Date: [3/23/2022] - [3/23/2022]

Test Results:

CASE1: If Security is not logged in:

Expected: Error with Response:HTTP_401_UNAUTHORIZED

Result: Same as Expected

Case 2: If security is logged in but data in request is not compatible or that entryId is not in database

Expected: HTTP_400_BAD_REQUEST

Result: Same as Expected

CASE: 3 Else

Expected: Entry data will be updated with exit time.

Response:HTTP_200_OK

Result: Same as Expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

http://127.0.0.1:8000/security/add_nonresident_campusexit/ Save

PUT ▼ http://127.0.0.1:8000/security/add_nonresident_campusexit/ Send ▼

Params Authorization Headers (9) **Body** ● Pre-request Script Tests ● Settings Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	079bf89c-7693-4684-89b2-fe4028ae...			

Body Cookies (1) Headers (9) Test Results 200 OK 28 ms 292 B Save Response ▼

18. See Non Residents Campus Entry list

Unit Details: [GetNonResidentsCampusEntryView, get]

Test Owner: Aarchie

Test Date: [3/23/2022] - [3/23/2022]

Test Results:

CASE1: If Security is logged in:

Expected: Get list of all Non Residents entries who have not yet exited with response: HTTP_200_OK

Result: Same as Expected

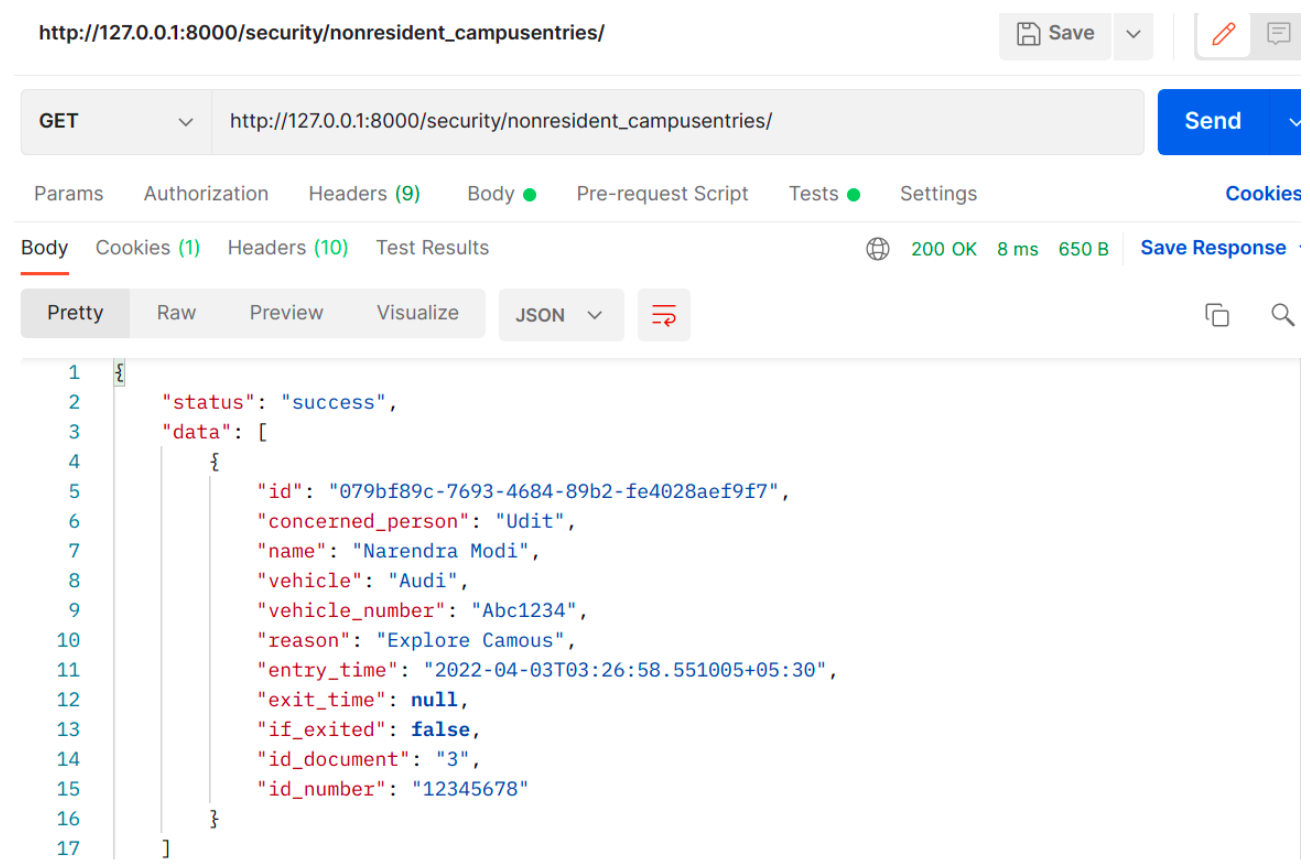
Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_UNAUTHORIZED

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results



[19. See All Non Residents Campus Entry list](#)

Unit Details: [GetAllNonResidentsCampusEntryView, get]

Test Owner: Aarchie

Test Date: [3/23/2022] - [3/23/2022]

Test Results:

CASE1: If Security is logged in:

Expected: Get list of all Non Resident Entries with response: HTTP_200_OK

Result: Same as Expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_UNAUTHORIZED

Result: Same as expected

Structural Coverage: Branch Coverage was used

Additional : Some of postman test results

The screenshot shows a Postman interface for a GET request to `http://127.0.0.1:8000/security/all_nonresident_campusentries/`. The request is saved and ready to be sent. The 'Body' tab is selected, showing a table with one entry:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	079bf89c-7693-4684-89b2-fe4028ae...	

Below the table, the 'Body' tab is selected, showing the response in JSON format:

```

1 {
2   "status": "success",
3   "data": [
4     {
5       "id": "5c357b09-69d1-44da-af65-8ab15aa12c7f",
6       "concerned_person": "Udit",
7       "name": "Arti Kumari",
8       "vehicle": "Maruti Suzuki",
9       "vehicle_number": "BR24R9855",
10      "reason": "to meet son",
11      "entry_time": "2022-04-03T03:18:03.981375+05:30",

```

20. LogOut

Unit Details: *Logout, post*

Test Owner: Archie

Test Date: [03/21/2022] - [03/21/2022]

Test Results:

Case 1: If a session was initiated with security or user being logged in

Expected Behaviour: Logout will be successful and session will be removed with status HTTP_200_OK

Result: Same as Expected

Case 2: Else

Expected Behaviour: Error with Response: HTTP_401_unauthorized

Result: Same as expected

Structural Coverage: Branch Coverage was used
Additional: Some of postman test results

http://127.0.0.1:8000/security/logout/

Save

POST

http://127.0.0.1:8000/security/logout/

Send

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests ●

Settings

Cookies

body

Cookies (1)

Headers (10)

Test Results

200 OK

31 ms

437 B

Save Response

3 Integration Testing

1. User Register and Login

Module Details: This module is integration of Register and Login units

Test Owner: Udit

Test Date: 25/03/2022

Test Results:

Case1: If Register didn't happened correctly and we proceeded for Login

Result: Login will give error with status code 401_Unauthorized

Case 2: Register happened correctly then we proceeded for Login and we give incorrect credentials in Login as compared to given during Register

Result: Login will give error with status code 401_Unauthorized

Case 3: Register happened correctly then we proceeded for Login and we give correct details in Login

Result: Login will be successful with status code 200

Additional Comments: *[Enter any summary comments]*

2. Security Register and Login

Module Details: This module is integration of Security Register and Security Login units

Test Owner: Udit

Test Date: 25/03/2022

Test Results:

Case1: If Security Register didn't happened correctly and we proceeded for Login

Result: Security Login will give error with status code 401_Unauthorized

Case 2: Register happened correctly then we proceeded for Login and we give incorrect credentials in Login as compared to given during Register

Result: Login will give error with status code 401_Unauthorized

Case 3: Register happened correctly then we proceeded for Login and we give correct details in Login

Result: Login will be successful with status code 200

3. Login and View Profile

Module Details: This module is integration of Login and UserProfile units

Test Owner: Udit

Test Date: 25/03/2022

Test Results:

Case1: If Login didn't happened correctly but user tries to fetch profile

Result: viewProfile will give error with status code 404_NOT_FOUND

Case 2: if Login happened correctly then user tries to fetch profile

Result: viewProfile will give user details with status code 200_OK

4. Lost_Item add and View

We will assume that security is logged in before carrying out these functions

Module Details: This model is integration of addLostItem and viewLostItems

Test Owner: Udit

Test Date: 25/03/2022

Test Results:

Case 1: If addLostItem didn't worked well and we proceeded for viewLostItems

Result: We will not see that item which we tried to add but the method will work fine with status code 200

Case 2: if addLostItem worked fine and we proceeded for viewLostItems

Result: We will be able to see the added item with status code 200

5. Lost_item add and Delete

We will assume that security is logged in before carrying out these functions

Module Details: This model is integration of addLostItem and viewLostItems

Test Owner: Udit

Test Date: 25/03/2022

Test Results:

Case 1: If addLostItem didn't worked well and we proceeded for Deleting that item

Result: No id will be generated so we cannot proceed to delete that item.

Case 2: if addLostitem worked fine and we proceeded for deleting that item with its ID.

Result: Selected item will be deleted from the database with status code 200

6. Found_Item add and View

We will assume that security is logged in before carrying out these functions

Module Details: This model is integration of addFoundItem and viewFoundItems

Test Owner: Udit

Test Date: 27/03/2022

Test Results:

Case 1: If addFoundItem didn't worked well and we proceeded for viewFoundItems

Result: We will not see that item which we tried to add but the method will work fine with status code 200

Case 2: if addFoundItem worked fine and we proceeded for viewFoundItems

Result: We will be able to see the added item with status code 200

7. Found_item add and Delete

We will assume that security is logged in before carrying out these functions

Module Details: This model is integration of addFoundItem and viewFoundItems

Test Owner: Udit

Test Date: 27/03/2022

Test Results:

Case 1: If addFoundItem() didn't worked well and we proceeded for Deleting that item

Result: No id will be generated so we cannot proceed to delete that item.

Case 2: if addfounditem() worked fine and we proceeded for deleting that item with its ID.

Result: Selected item will be deleted from the database with status code 200

Additional Comments: [\[Enter any summary comments\]](#)

8. Add Hall Entry and view hall entry

We will assume that security is logged in before carrying out these functions

Module Details: This model is integration of addHallEntry and viewHallEntry

Test Owner: Udit

Test Date: 27/03/2022

Test Results:

Case 1: If addHallEntry didn't worked well and we proceeded for viewHallEntries

Result: We will not see that item which we tried to add but the method will work fine with status code 200

Case 2: if addHallEntry worked fine and we proceeded for viewHallEntries

Result: We will be able to see the added item with status code 200

9. Add Hall Entry and Add Hall Exit

We will assume that security is logged in before carrying out these functions

Module Details: This model is integration of addHallEntry and addHallExit

Test Owner: Udit

Test Date: 27/03/2022

Test Results:

Case 1: If addHallEntry() didn't worked well and we proceeded for addHallExit() for that very item.

Result: We will not be able to proceed as we will not have ID which will be used to update the status

Case 2: if addHallEntry() worked fine and we proceeded for addHallExit()

Result: The item will be updated with exit time with status code 200

10. Add Hall Exit and view all hall entry

We will assume that security is logged in before carrying out these functions

Module Details: This model is integration of addHallExit and viewAllHallEntry

Test Owner: Udit

Test Date: 27/03/2022

Test Results:

Case 1: If addHallExit didn't worked well and we proceeded for viewAllHallEntries

Result: We will see that item for which we tried to update exit details but without any exit status with status code 200

Case 2: if addHallExit worked fine and we proceeded for viewAllHallEntries

Result: We will be able to see the updated item with exit details too with status code 200

11. Add Campus Exit and view Campus Exits

Module Details: This model is integration of addCampusExit and viewCampusExit

Test Owner: Udit

Test Date: 27/03/2022

Test Results:

Case 1: If addCampusExit didn't worked well and we proceeded for viewCampusExit

Result: We will not see that item which we tried to add but the method will work fine with status code 200

Case 2: if addCampusExit worked fine and we proceeded for viewCampusExit

Result: We will be able to see the added item with status code 200

12. Add Campus Exit and view All Campus Exits

Module Details: This model is integration of addCampusExit and viewAllCampusExit

Test Owner: Udit

Test Date: 27/03/2022

Test Results:

Case 1: If addCampusExit didn't worked well and we proceeded for viewCampusExit

Result: We will not see that item which we tried to add but the method will work fine with status code 200

Case 2: if addCampusExit worked fine and we proceeded for viewCampusExit

Result: We will be able to see the added item with status code 200

13. Add Campus Exit and Add Campus Entry

We will assume that security is logged in before carrying out these functions

Module Details: This model is integration of addCampusExit and AddCampusEntry

Test Owner: Udit

Test Date: 27/03/2022

Test Results:

Case 1: If addCampusExit didn't worked well and we proceeded for addCampusEntry

Result: We will not be able to proceed as we will not have ID which will be used to update the status

Case 2: if viewCampusExiys worked fine and a item is in the list and we proceeded for making campusEntry of that item i.e. updating the entry status.

Result: Selected item will get updated with entry time with status code 200

14. Non-Resident Campus Entry and View

We will assume that security is logged in before carrying out these functions

Module Details: This model is integration of addNonResidentCampusEntry and viewnonResidentCampusEntry

Test Owner: *Udit*

Test Date: *27/03/2022*

Test Results:

Case 1: If addNonResidentCampusEntry didn't worked well and we proceeded for viewnonResidentCampusEntry

Result: We will not see that item which we tried to add but the method will work fine with status code 200

Case 2: if addNonResidentCampusEntry worked fine and we proceeded for viewnonResidentCampusEntry

Result: We will be able to see the added item with status code 200

15. Non-Resident Campus Entry and Exit

We will assume that security is logged in before carrying out these functions

Module Details: This model is integration of addNonResidentCampusEntry and addnonResidentCampusExit

Test Owner: Udit

Test Date: 27/03/2022

Test Results:

Case 1: If addNonResidentCampusEntry didn't worked well and we proceeded for addnonResidentCampusExit

Result: We will not be able to proceed as we will not have ID which will be used to update the status

Case 2: if viewnonResidentCampusEntry worked fine and we proceeded for making nonResidentCampusExit for that item i.e. updating the exit status.

Result: Selected item will updated with exit time with status code 200

4 System Testing

In all of the below mentioned tests, we are assuming that the functionality is being used by a loggedIn security user.

1. Requirement: *Reporting and Solving Lost Query*

Test Owner: *Udit*

Test Date: 30/03/2022

Test Results:

We will proceed stepwise

- Reporting Lost query
 - i) All the required fields with compatible data was sent as request
 - ii) Given UID was registered
 - ☐ Lost Query was submitted
 - ☐ Response: HTTP_201_CREATED
- Proceeded to view lost items
 - ☐ That item was fetched from the database and its ID was recorded
 - ☐ Response: HTTP_200_OK
- Proceeded to delete that item
 - i) Recorded ID was sent in the request
 - ☐ That item was successfully deleted from the database
 - ☐ Response: HTTP_200_OK
- Proceeded to view lost items
 - ☐ That item was not in the fetched items
 - ☐ Response: HTTP_200_OK

2. Requirement: *Reporting and Solving Found Query*

Test Owner: *Udit*

Test Date: 30/03/2022

Test Results:

We will proceed stepwise

- Reporting found query
 - i) All the required fields with compatible data was sent as request
 - ii) Given UID was registered before this process
 - ☐ Found Query was submitted
 - ☐ Response: HTTP_201_CREATED

-
- Proceeded to view found items
 - ☐ That item was fetched from the database and its ID was recorded
 - ☐ Response: HTTP_200_OK
 - Proceeded to delete that item
 - i) Recorded ID was sent in the request
 - ☐ That item was successfully deleted from the database
 - ☐ Response: HTTP_200_OK
 - Proceeded to view found items
 - ☐ That item was not in the fetched items
 - ☐ Response: HTTP_200_OK

3. Requirement: *Hall Movement*

Test Owner: Udit

Test Date: 31/03/2022

Test Results:

We will proceed stepwise

- Add Hall entry
 - i) All the required fields with compatible data was sent as request
 - ii) Given UID was registered before this process
 - ☐ Hall Entry was submitted
 - ☐ Response: HTTP_201_CREATED
- Proceeded to view hall entries who have not yet exited
 - ☐ That item was fetched from the database and its ID was recorded
 - ☐ Response: HTTP_200_OK
- Proceeded to view all hall entries
 - ☐ That item was fetched from the database
 - ☐ Response: HTTP_200_OK
- Proceeded to add Hall Exit for that entry
 - i) Recorded ID was sent in the request
 - ☐ That item was successfully updated with exit time
 - ☐ Response: HTTP_200_OK

-
- Proceeded to view hall entries who have not yet exited
 - ☐ That item was not fetched from the database
 - ☐ Response: HTTP_200_OK
 - Proceeded to view all hall entries
 - ☐ That item was fetched from the database with updated exit details
 - ☐ Response: HTTP_200_OK

4. Requirement: *Campus Movement*

Test Owner: Udit

Test Date: 31/03/2022

Test Results:

We will proceed stepwise

- Add Campus Exit
 - i) All the required fields with compatible data was sent as request
 - ii) Given UID was registered before this process
 - ☐ Campus Exit was submitted
 - ☐ Response: HTTP_201_CREATED
- Proceeded to view campus exits who have not yet returned
 - ☐ That item was fetched from the database and its ID was recorded
 - ☐ Response: HTTP_200_OK
- Proceeded to view all campus exits
 - ☐ That item was fetched from the database
 - ☐ Response: HTTP_200_OK
- Proceeded to add Campus Entry for that exit
 - i) Recorded ID was sent in the request
 - ☐ That item was successfully updated with entry time
 - ☐ Response: HTTP_200_OK
- Proceeded to view campus exits who have not yet returned
 - ☐ That item was not fetched from the database
 - ☐ Response: HTTP_200_OK
- Proceeded to view all campus exits
 - ☐ That item was fetched from the database with updated entry details
 - ☐ Response: HTTP_200_OK

5. Requirement: *Non Resident Campus Movement*

Test Owner: Udit

Test Date: 31/03/2022

Test Results:

We will proceed stepwise

- Add entry
 - i) All the required fields with compatible data was sent as request
 - ii) Given UID of associated person was registered before this process
 - ☐ Non Resident campus Entry was submitted
 - ☐ Response: HTTP_201_CREATED
- Proceeded to view Non Resident campus entries who have not yet exited
 - ☐ That item was fetched from the database and its ID was recorded
 - ☐ Response: HTTP_200_OK
- Proceeded to view all Non Resident campus entries
 - ☐ That item was fetched from the database
 - ☐ Response: HTTP_200_OK
- Proceeded to add Exit for that entry
 - i) Recorded ID was sent in the request
 - ☐ That item was successfully updated with exit time
 - ☐ Response: HTTP_200_OK
- Proceeded to view Non Resident campus entries who have not yet exited
 - ☐ That item was not fetched from the database
 - ☐ Response: HTTP_200_OK
- Proceeded to view all Non Resident campus entries
 - ☐ That item was fetched from the database with updated exit details
 - ☐ Response: HTTP_200_OK

5 Conclusion

How Effective and exhaustive was the testing?

Effectiveness-

All the features of software was tested and all of them was working according to expectations like when it need to send the errors, it was sending and when it needed to work fine , it was working so.

Exhaustiveness-

We tried to exhaustively test the software upto a level but due to time constraints and other constraints, we have not achieved that much exhaustive testing.

Which components have not been tested adequately?

We have tried to test all the components adequately but Non Resident Campus Movement component was not tested that adequately because it was somewhat like campus movement component, so we only tested its effectiveness but not tested it exhaustively.

What difficulties have you faced during testing?

Difficulties faced during testing:

- We did not knew that much amount tools used for testing
- automated testing was tough to implement, so we used the manual method
- We used branch coverage so many a times, we need to test many branches manually.

How could the testing process be improved?

Testing process could be improved by:

- starting testing process as early as possible
- Using automation wherever it is feasible
- planning test cases in a well designed way

Appendix A - Group Log

Date	Activity
15/03/2022- 24/03/2022	Unit testing was done along with implementation
24/03/2022- 28/03/2022	Integration Testing was done
30/03/2022-31/03/2022	System testing was done and also exhaustive testing was done
01/04/2022-02/04/2022	Created the first draft of the document
03/04/2022	Got the document reviewed by the TA
03/04/2022	Updated the document