

QUESTION 1

- a. Explain in about SDLC, STLC and their differences

Answer :

- **SDLC**

SDLC is abbreviation from *Systems Development Life Cycle*. It is the stages of how a software is developed. In another meaning, SDLC is purposes to produce a high quality software according to users wishes or create the software according to it's purposes

- **STLC**

STLC is abbreviation from *System Testing Life Cycle*. It is the sequence of activities carried out to perform software testing. In this phase, the testing team should have to learn about the requirements from a testing POV (*Point of View*) to identify the requirements to be tested. Requirements could be functional or non-functional

The Differences about SDLC and STLC is SDLC is the steps of how a software is developed. While STLC is a sequence of tests carried out by the Testing team in order to identify requirements from various points of view

- b. Explain the steps for Bugs cycle!

Answer :

1. **New :**

When a bug is first discovered and recorded by the tester to be given to the developer

2. **Assigned :**

Assign bugs to team for fix after the bug is deemed valid, the project leader or test lead gives the bug to the team concerned to fix it

3. **Open :**

Team do analysis on bugs and fix bugs In this status, bugs will be checked. If the team feels that the bug is valid, it can be fixed. However, if the bug is invalid, it can be categorized as *rejected*, *duplicated*, *deferred*, or *not a bug*.

- *Rejected :*

If the bug is considered not a bug from developers, it would given rejected status.

- *Duplicated :*

Bug that have the same description with another bugs

- *Deferred :*

Deffered status will be given to the bug if:

- Bug finds are minor priority or found at the end of the release
- Bugs are not *scope* in current development
- Bugs will be fixed on the next release
- Requirement changes from the client

- *Not a bug :*

Bugs is not affected with the apps functionality

4. **Fixed :**

The bug has been resolved by the team. The development team made necessary changes to the coding rules to fix bugs. After being fixed and confirmed, the team can give the bug status to Fixed.

5. **Retest :**

Testing on fixed bugs. The tester will test whether the bug has been fixed and is in accordance with the needs.

6. Reopen :

The bug is still found and the results are still not as needed. When testing is still found bugs and does not meet the needs, the tester will give the status Reopen. Then the bug will be fixed again by the developer team.

7. Verified :

Testers verify bug fixes. The tester verifies that if the bug that is fixed is no longer found, it can be given a Verified status.

8. Closed :

When the bugs don't appear anymore the bug no longer appears in the application. This state is the end of the bug life cycle.

c. Give us the Real Case about Severity and Priority!

Answer :

- Severity

| Defect Severity Levels | | Real Case |
|------------------------|----------|--|
| S1 | Critical | When the application sent an OTP code to users mobile number for login purposes, but the code does not reach the user even their mobile signal is in good condition, the user cannot use the application. |
| S2 | High | User can login by using email/username and password with ignoring the uppercase/lowercase of the password letter |
| S3 | Moderate | In email service providers such as Gmail, there is an option called "Terms and Conditions" and in that option there will be many links regarding the terms and conditions of the website when one of the many links is not working properly, it is called Severity Minor because it only affects the functionality small application and do not have a big impact on the usability of the application. |
| S4 | Low | Spelling errors or the size of the letters contained in the application do not cause any effect on the application |

- Priority

| Defect Priority Levels | | Real Case |
|------------------------|-----------|--|
| S1 | Immediate | when all functionality is blocked and no testing can proceed as a result |
| S2 | High | when a feature cannot be used properly due to a program bug or that new code must be written |
| S3 | Medium | When application |
| S4 | Low | Spelling errors or the size of the letters contained in the application do not cause any effect on the application |

| | | |
|--|--|--|
| | | application but it has to be fixed before client testing for the application |
|--|--|--|

QUESTION 2

1. Positive Testing Scenario

| Features | User Story | Test Step | Expected Result | Result |
|------------------------------------|--|---|---|--------|
| Create Ticket AM (Connectivity) | User be able to create ticket AM | User clicked Create ticket menu | Systems showing the costumer list | PASSED |
| | | User Select Costumer on the list | There's a costumer list | PASSED |
| | | | Costumer list can be selected | PASSED |
| | | | User can select the costumer on the list | PASSED |
| | | | System would directing user to the service category | PASSED |
| | User Select Service Category (Connectivity) | User Select Service Category (Connectivity) | There's service category list | PASSED |
| | | | Category list can be selected | PASSED |
| | | | User can select service category | PASSED |
| | | | Systems would directing user to SID Form | PASSED |
| | | User Typing Suggest SID | There's a SID Form | PASSED |
| User Select location | User Typing Suggest SID | SID Form can be fill in | SID Form can be fill in | PASSED |
| | | | User can fill in the suggest SID with valid data | PASSED |
| | | | Systems displaying IP WAN | PASSED |
| | | | Systems Showing Location List | PASSED |
| | | User Select location | There's location dropdown | PASSED |
| | | | Location dropdown showing the locations after click on it | PASSED |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| | | | |
|--|------------------------------|---|--------|
| | | User can select location | PASSED |
| | | Systems displaying service name list | PASSED |
| | User Select Service Name | There's Service Name Dropdown | PASSED |
| | | Service Name dropdown showing all service name after click on it | PASSED |
| | | User can select service name | PASSED |
| | | Systems Displaying Symptom Category List | PASSED |
| | | Sympton Category List is appeared | PASSED |
| | User Select Symptom Category | Sympton Catgeory list can be selected by user | PASSED |
| | | User can select symptom category | PASSED |
| | | Selected category will be signed with the bold text on it | PASSED |
| | | | |
| | User Typing PIC Name | There's PIC Name Field | PASSED |
| | | Fill in with the String data | PASSED |
| | | User can see the inputed data on the field | PASSED |
| | User Typing PIC Phone Number | There's PIC Phone Number | PASSED |
| | | There's the dropdown of country code | PASSED |
| | | User can select the country code | PASSED |
| | | If the country does not appear on the list, user could fill in manually | PASSED |
| | | | |

| | | | | |
|----------------------------------|--|--|---|--------|
| | | | Fill in with the numeric data | PASSED |
| | | | User can see the inputed data on the field | PASSED |
| User Click Submit Button | | | There's submit button | PASSED |
| | | | Button can be clicked | PASSED |
| Validation Create Ticket Success | | | There's notification that ticket has been created | PASSED |
| | | | Go to the main menu | PASSED |

2. Negative Test Scenario

| Features | User Story | Test Step | Expected Result | Result |
|---------------------------------|--|--------------------------------------|--|--------|
| Create Ticket AM (Connectivity) | User be able to create ticket AM | User clicked Create ticket menu | Systems showing the costumer list | PASSED |
| | | User Didn't select any costumer list | There's a costumer list | PASSED |
| | | | System would appearing error message "Please select costumer first" | PASSED |
| | | | System would keep on the page | PASSED |
| | User didn't Select Service Category (Connectivity) | | There's service category list | PASSED |
| | | | User didn't select category | PASSED |
| | | | Systems would directing user to SID Form | PASSED |
| | User Typing Suggest SID with invalid data | | There's a SID Form | PASSED |
| | | | User filled in the suggest SID with invalid data | PASSED |
| | | | There's error notification "SID must be filled in with correct data" | PASSED |
| | | | Systems will keep the user on the page | PASSED |

| | | | | |
|--|--|--|--|--------|
| | | User didn't select location | There's location dropdown | PASSED |
| | | User didn't selected location | User didn't selected location | PASSED |
| | | | System will showing error notification "Please select location" | PASSED |
| | | User didn't Select Service Name | There's Service Name Dropdown | PASSED |
| | | | User didn't selected service name | PASSED |
| | | | Systems Displaying error notification "Service Name must be filled in" | PASSED |
| | | User didn' select Symptom Category | Sympoton Category List is appeared | PASSED |
| | | | User didn't select symptom category | PASSED |
| | | | System will showing error notification "You have to select symptom category" | PASSED |
| | | User Typing PIC Name with invalid data | There's PIC Name Field | PASSED |
| | | | Fill in with the numeric or unique character | PASSED |
| | | | System will showing error message "PIC Name must in correct characters" | PASSED |
| | | User Typing PIC Phone Number With invalid data | There's PIC Phone Number | PASSED |
| | | | Filled in with the chacarter will showing error notification "Phone number must be in numeric" | PASSED |

| | | | |
|--|--|--|--------|
| | | User didn't select the country code | PASSED |
| | | System will showing error notification "Please input country code" | PASSED |
| User Click Submit Button when data is not correct/incomplete | There's submit button | PASSED | |
| | Button can be clicked | PASSED | |
| | Showing error message "data is not correct/incomplete, please check again" | PASSED | |

QUESTION 3

You are testing a new version of software for a coffee machine. The machine can prepare different types of coffee based on four categories. i.e., coffee size, sugar, milk, and syrup. The criteria are as follows:

- Coffee Temperature should be about 78%
- Coffee size (small, medium, large)
- Sugar (none, 1 unit, 2 units, 3 units, 4 units)
- Milk (yes or no)
- Coffee flavor syrup (no syrup, caramel, hazelnut, vanilla)
- Sugar (3 and 4 Units) cannot add Coffee flavor syrup

And now we have some issues:

- You can add Coffee flavor syrup when sugar is 3 Units
- The coffee temperature is below 78 %

As a QA:

- A. Describe how a test scenario can ensure that all test coverage is covered
- B. How do you identification issue and write defect report from the following information?

Answer :

- a. Test Scenario that I would like to make is have to make sure all the machine components such as Thermator (For measure the water temperature), Bolier (for boiling the coffee) etc. can be detected by the software. Then the tester must ensure that all possibilities that will occur can be overcome by software and carried out by machines, such as if the coffee cup is not in its place, the temperature is not in accordance with the predetermined provisions, and the syrup and sugar do not match
- b. For the issues, bellow here are the possibilities :
 1. The machine put 2 units of sugar, even though the user inputs 3 units of sugar (Software reduces sugar units by 1 from the sugar inputted by the user)
 2. When the machine will not add sugar, the user reduces the amount of sugar by 1 so the machine will add sugar to the coffee

QUESTION 4

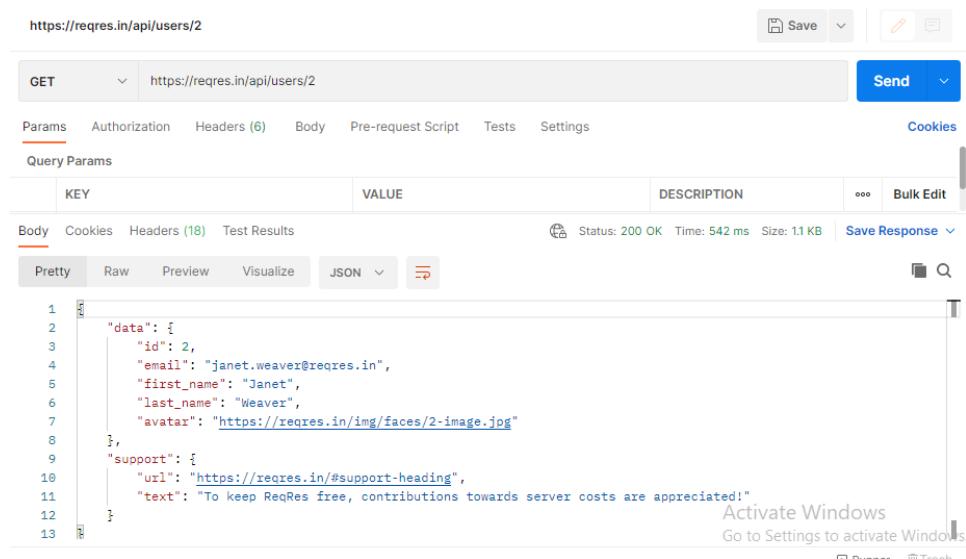
- As a QA, please compare between MockAPI and RealAPI. Is there any differences? Explain it
- If there are differences, what defects will appear?

Answer :

On the Mockup API :

```
1  {
2      "data": {
3          "id": "2",
4          "email": "janet.weaver@reqres.in",
5          "first_name": "Janet",
6          "last_name": "Weaver",
7          "avatar": "https://reqres.in/img/faces/2-image.jpg"
8      },
9      "support": {
10         "url": "https://reqres.in/#support-heading",
11         "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
12     }
13 }
```

On the Real API :



The screenshot shows the Postman application interface. A GET request is made to <https://reqres.in/api/users/2>. The response status is 200 OK, with a time of 542 ms and a size of 1.1 KB. The response body is displayed in JSON format:

```
1  {
2      "data": {
3          "id": 2,
4          "email": "janet.weaver@reqres.in",
5          "first_name": "Janet",
6          "last_name": "Weaver",
7          "avatar": "https://reqres.in/img/faces/2-image.jpg"
8      },
9      "support": {
10         "url": "https://reqres.in/#support-heading",
11         "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
12     }
13 }
```

- There are several differences between the mockup API and the real API :
 - where in the mockup, id has a string data type, while in the Real API, id has a numeric data type.
 - Naming the "Last Name" field in the mockup using camelcase while the Real API uses an underscore.
- When the Real API is implemented in the system, there will be some impact :
 - When we fill out all the forms contained in the field (except id), then the id of the next data will be increased, because id is numeric and allows for auto increment. Meanwhile, if we implement the Mock API, then no matter how much data we have created, the id will have the same value because it has string type which is not allows for auto increment

2. Naming the field "Last Field" doesn't have as much effect as in the previous case. Because both have the same data type, namely string