**Question**:

**Dinal made a soda despesing machine how would you test it?**

https://www.careercup.com/question?id=213691

**1) Functional test pass**

What type of coins does the vending machine accept?

Verify what type power input does vending machine handle?(Voltage)

Verify what is the capacity of the vending machine

Verify what type of food does it contains?

**2) Stress Testing:**

Verify that at what level vending machine stops accepting orders?

**3) Load testing:**

verify what is the threshold of the machine

**4) Usability testing:**

verify who will use this machine i.e what is the age group

**5) Security testing :**

Verify what type of security concerns you want to capture?

**6) Performance testing :**

How many transactions does this machine handles in a day?

**7) Acceptance testing**

how would user get money back if there is a problem in machine

**Functional testing**

**Instability testing**

**Usability testing**

**Performance**

**Stress**

**Compatibility**

**Capability testing**

**Equivalence portioning /boundary value.**

**Functional testing**

check proper change

check improper change

check proper itrem and improper item

no item is there .

more than available items stuffed in the tray

machine does not have change.?

**Instability testing**

check if vending machine can be inztalled properly with proper connection

check if it is movable.

**Usability testing**

check if vending machine has proper buttons

check if it has 0-9 numbers to select

check if it has coin return

dispenser/provision for hands to fit in to take the product

buttons are not too rough

check for the height of the operationabilty.

**Performance**

how quickly it drops

test on differnt plugs and power 110 v 220 v 440v

test on incremental loads of coins and keep selecting

keep inserting coins continously for 1 hr then select

**Stress**

shake the machine

switch on/switch off machine alternatingly and select it.

in severe cold/hot atmosphere does it work.

keep inserting coins continously for 1 hr then select

insert nothing and keep pressing

**Compatibility**

different coins/dollars

takes a 1 dollr 5 dollar 100 dollar etc....

can vending machine work properly outside

**Capability testing**

what can vending machine do other than vending solid items?

keep liquid and solid items

can it be used just for vending coins?

(this is a useful feature for

taking quarters!!)

**Equivalence partitioning /boundary value.**

**What is a bug?** is an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

http://istqbexamcertification.com/what-is-the-difference-between-severity-and-priority/

**What is the Difference**

**Severity and priority?**

**Severity**

Critical, Major, Moderate, Minor, Cosmetic

Critical (Crash, Data Corruption, failed major functionality),

Major (like Critical but an alternative solution exist),

Moderate ( incorrect, incomplete or inconsistent results ),

Minor (no termination and does not damage the usability of the, working around the defects),

Cosmetic (Enhancement of the system, )

**Priority**

Low (irritant),

Medium (defect should be resolved),

High (defect must be resolved as soon as possible)

Usually the **severity** is defined in terms of financial loss,

damage to environment, company’s reputation and loss of life.

**Priority** of a defect is related to how quickly a bug should be fixed and deployed to live servers.

It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

For example: If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of  application crashing is severe. So the severity is high but priority is low.

<https://sqa.stackexchange.com/questions/1920/best-guidelines-for-bug-reporting>

**What must appear in a bug report?**

Good **Title**, steps to reproduce and fix problems

Bug reporter 🡸=communications==🡺 Bug receiver

**Sent** to the person in charge

**Agil Way**

**As a developer**, this is the information I need to solve a problem:

1. Steps to reproduce.
2. Expected result.
3. Actual result.

Additional Support:

**Screenshot**

**Video**

**Notes**

**Choose a Good Summary/Title**

**Describe the problem concisely and effectively**

**Include the Results you Expected**

**Include the Results you Actually Observed**

**Include Enough Details for Searching**

**Explain the Effects on the Customer**

**Attach Anything Else that Could Help**

**Avoid speculation**

**Be careful of the tone of your report**

**Avoid duplication - search first**

Couple of guidelines in addition to above listed items

1. Test Steps
2. Snapshots for each steps if possible (In case you are working in remote teams, to avoid to-and-fro email communications)
3. Expected result vs Actual Result
4. Environment Details - OS, Hardware, Software, Build version
5. Log file entries / values
6. Nice to have - Preliminary investigation / analysis with supporting queries / assumptions to provide couple of leads for developer to check further
7. Provide access to the test environment - URLs, Machines for the developer to check in case if needed
8. Reference to BRD, FS, Design Document where implementation conflicts design / requirements
9. Nice to Have - Triage Meeting / Issue Review meeting to run through the bugs once with the development team to provide a quick overview of issues before they start looking into it. F2F conversations are better than email / chat coversations sometimes

Be Descriptive do not use Abbrevations, No implicit assumptions. Callout your understanding of functionality and how it conflicts with implementation

**what are some of the things that must appear in a test plan?**

1.**Introduction**:

Vending machine is a unit which provides food/drink in exchange of coins.

2.**Goal** : Test the functionality of vending machine

3.**Team involved**:

4.**Bugs Information**

5.**Spec Links**

6.**Known Issues**:

7.**Test Scope** :

**What is a BRD?** Business Requirement Document (BRD)

software requirement specification;

Details the business solution for a project

including the documentation of customer’s needs and expectations.

If an initiative intends to modify existing (or introduce new) hardware/software,

a new BRD should be created.

The BRD process can be incorporated within a

Six Sigma DMAIC (Define, Measure, Analyze, Improve, Control) culture.

The most common objectives of the BRD are:

* To gain **agreement with stakeholders**
* To provide a foundation to communicate to a technology service provider what the **solution needs to do to satisfy the customer’s and business’ needs**
* To provide **input into the next phase** for this project
* To describe what not how the customer/business needs will be met by the solution

**what is an FRS?**

File Replication service (FRS) is a technology that **replicates files and folders** stored in the SYSVOL shared folder on domain controllers and Distributed File System (DFS) shared folders.

When FRS **detects** that **a change** has been made to a file or folder within a replicated shared folder, FRS replicates the updated file or folder to other servers.

FRS can **resolve** **file** and folder **conflicts** to make data consistent among servers.

**Have you ever created a test case ?**

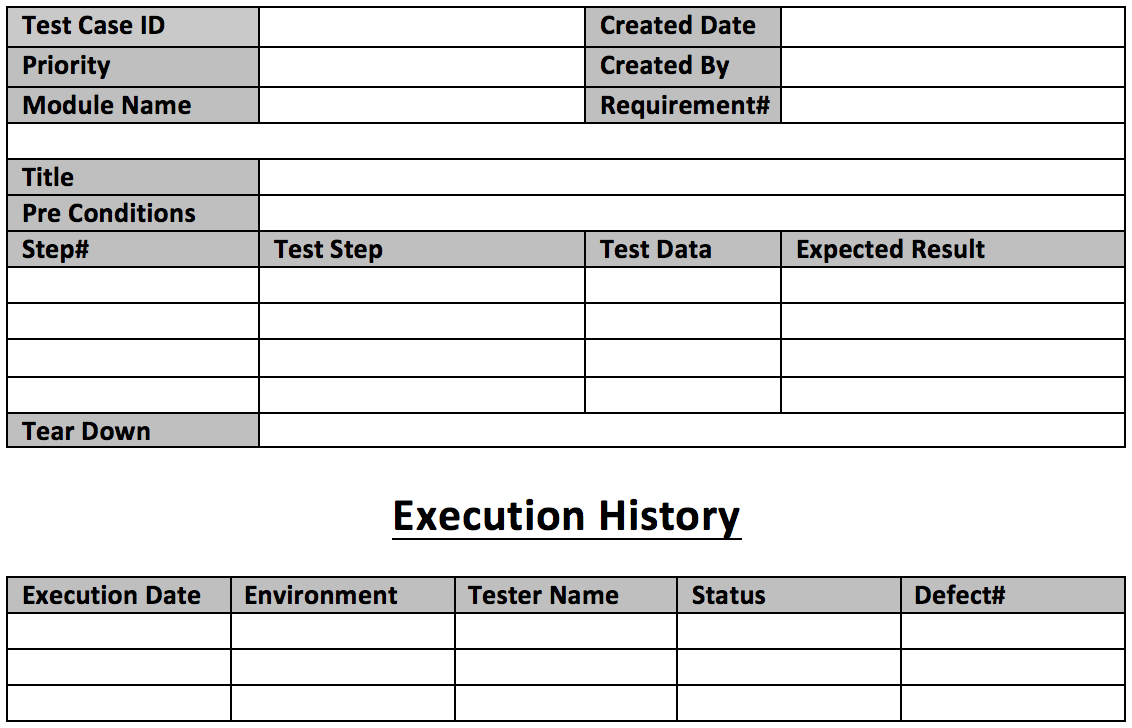
**or where test cases given to you?**

Yes both.

**and how many test cases did you execute?**

Umm 100’s if not 1000.

**what should be included in test case template?**



**what is**

**ad-hoc testing?**

Ad hoc testing is a commonly used term for **software testing** performed without planning and documentation, but can be applied to early scientific experimental studies.

**Accessibility testing?**

is a type of systems testing designed to determine whether individuals with disabilities will be able to use the system in question, which could be software, hardware, or some other type of system.

**end-to-end testing?**

This is more about the **actual flow through a system** in a more realistic end user scenario.

Can a user navigate the application as expected and does it work. You're testing the workflow.

This is more about the actual flow through a system in a more realistic **end** user scenario. Can a user navigate the application as expected and does it work. You're **testing** the workflow. For example if you were to **test** an e-commerce site the shop front...

**what interest you in most about website and application from user perspective for forturne 500 companes for a company using tablet or cell phone?**