WEB SITE Management tools

Web application commonly experience upkeep at a quicker rate than any other system. Web applications have an enormous client populace, in this way propose a high demand to the server's execution. The purposes of these tools are site version control tools, server management, website management, combined utilities or tools. It also includes development, publishing tools that have vast capabilities of testing. These tools are used in development, testing and maintenance phase. These tools are used to develop, deploy and run the finest content-driven applications in an easy way.

Inputs and outputs:

|  |  |
| --- | --- |
| Web site | Html ,css validation and JavaScript Analysis |
| Web site | 404 errors, Asp.net errors , php errors |
| Web site | Runtime error that user can face. |
| Web site | Load time testing, broken link errors |
| Web site | Network and Server errors |
| Web site | Server performance logs |

For validating Html, they use fast & local copy of official W3C HTML validator. For validating Cascade style sheet, they use fast & local copy of official W3C CSS validator. For Javascript they use defecto standard. They use code quality analyzer and JSUnit syntax checker.

Tools in this category is desktop based and also as web service. They provide full site automated html validation. They help architects to build best content driven web application efficiently. If we have developed a website for example application containing products. Then tool could be used to check pages that are removed or broken links. A tool in this category have capability to take screenshot of webpage associated with an error. And to send this screenshot to its hosted tool via API. One tool of this category makes statistics, report and also suggestions for the improvement of website performance and load time.

Merits of using these tools are. It provides quick feedback that either the develop code is error free and we could go next or there are some glitches/ errors that we have to solve first. Bug detection is early. These tools provide a quick review about web application. We can check multiple conditions very quickly. These tools save a lot of time for testers to perform manual testing and to evaluate system. Regression errors could be find out by this process. Could done performance, load and stress testing. Tools provide help in detecting different behavior in different platforms. Tools help to authentic critical functionality of web site. We can easily capture result and could make comparison with actual output. We have information about Missing page and broken links could be find out. We could test design as well as behavior of the system. We could detect different type of behavior of same web application on different browsers. We could judge the system with maximum repetition of same behavior. The tools are reusable i.e. used for many websites without changes. Tool reduce our cost on human. Tools provide fix and reliable outputs. We will get more accurate results if our data is complete and we have well understood system. The script that we have to write for automation should be concise and clear. We could get more benefits from testing if we testing tool follow these principles. It should be simple and concise. It has ability to check its own results and there is no need of human to interpret results. It should be sufficient that it could check all requirements.

Automated checks continue flopping because of issues other than actual bugs, they can raise false cautions. automated checks can break in light of the fact that a minor UI change was executed or there is any network error which is not associated with application but it will affect automated testing. Testing tools are better for unit testing. They are not considered useful for integration testing. One of disadvantage or we can say demerit is that these tools are not freely available they have high cost to pay for them. And there is another issue, training of staff. Maximum bugs appear to be found by "mishap" or when performing exploratory testing. This is most likely because of the way that amid each exploratory testing session we could test the application in various ways along these lines finding new loopholes through the application. We agree with the statement that testing through tools save a lot of time but some of the automation testing could take more time then manual testing for example GUI testing take more time then manual testing. There is drawback that these tools are unable to find out complex bugs / errors.

**Some view by software community:**

Contrasted and manual testing, the cost of AS is higher, particularly in the start of the procedure. Be that as it may, automation testing can be more profitable after a time. 42% persons rate this 5 out of 5. 60% rated as 4/5. 8% rated as 3/5.

Automation testing requires additional exertion for designing and keeping up test contents. 32% rated as 5, 56% rated as 4, 6% rated as 3, 5% rated as 2, and 1% rated as 1 out of scale 5.

In comparison with manual testing, tools require a high venture. To purchase apparatuses (testing tools or services) and prepare staff to better utilize these facilities. The reviews of the people were. And Its percentage with respect to scale is. 28% testing community experts rated this as 5, 49% as 4, 10% as 3, 8% as 2 and 3% as 1.

Tools requires less exertion from developer side, however can't discover complex bugs as we can get through manual testing. 17 % people rated as 5, 43% as 4, 17% as 3, 21% as 2 and 3% as 1.

The exists vast variety of the testing tools accessible in the market are insufficient and partial and don't give what we need or fits in our criteria. 10 % rated as 5, 35% as 4, 26% as 3, 21% as 2 and 9% as 1.

Last but not the least statement that automation testing will replace manual testing techniques. 1% experts said 5, 5% rated as 4 , 14% people reviewed as 3, 43% said 2, 37% rated as 1 out of scale 5.

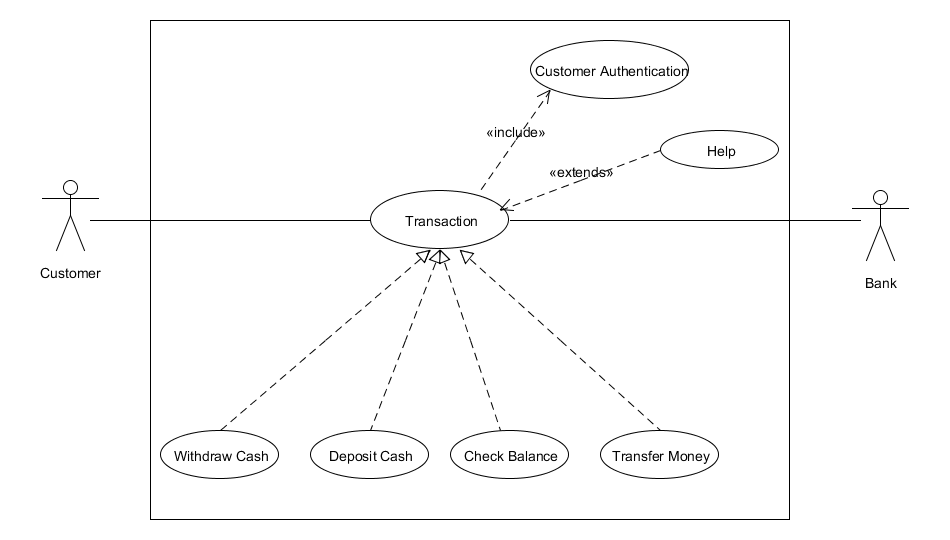
The people from testing community reviewed that could save their time while html validation, css validation, Javascript behavior checking and also many tasks like stress testing , performance testing and many more.

Tools Examples:

|  |  |
| --- | --- |
| Simply Testable | Web-based service  One click perform all web task  Price:  Personal £9  Agency £19  Business £59  People evaluation: People say that they could perform large website testing easily |
| Nuxeo | Open source platform  Easily used with less programming experience.  People said that this tool provides integration with many API. But have only available documentation in Java. If the amount of data is less then nuxeo is best option. People like its dynamic features of datatypes. |
| Radient CMS | open source content management system  It is designed for small teams. Its main features are Elegant UI. Extension system. Simple user management and etc.  It is developed in ruby and server used is Apache. |
| Frog CMS | open source content management system  It is simple templating code and uses PHP, a database and a webserver  It is free to use. |
| Auto Test Bot | Free online web service  Used for web site monitoring and testing,  It give waring and errors about broken pages. deleted page that have link with the web site. |
| BugDigger | Web bug reporting tool  A simple tool that adds button in browsers and automatically collects data that is useful for developers. It could also attach screenshot with the report and forward that image to hosted service or API.  Prancing Plans:  Team  $19 per month for 10 Active Users  10 Projects For 60 Days  Agency  $49 per month 25 Active Users  25 Projects for 90 Days  Corporate  $99 per month for 100 Active Users  100 Projects for 120 Days  Infinity  $199 per month for 500 Active Users  Unlimited Projects for 120 Days Data Retention |
| BigEasy CMS | Content management system  Features included: creation and reusability of HTML templates, template management etc. |
| A1Website Analyzer | Website analyzer and link checker developed by Microsys.  Used for search engine optimization, response time, html and css validation.  It is free for personal use. And for professional price is only 69$. |
| Errorlytics | Site management service/plugin  Monitors errors that site guests go over. Can perceive what mistakes have come up, and after that define 'rules' that on which page user should redirected. |
| WSOP | This is a tool used for load time testing and optimization.  It is developed by SoftLogica LLC  it provide stasitical analysis of test. |
| IBM tealeaf | Web application Customer Experience Management tool  Developed by IBM  Price is 124.8 $  one of the user said that he like the level of granularity Tealeaf gives you while breaking down client encounter on the web. Where apparatuses like Omniture or Google Analytics just give you the 10,000-foot view, Tealeaf can do that in addition to enable you to see the reason for any inconsistencies. |
| Broad vision | Content and publishing management tools.  It is developed by Broadvision inc.  It allows users from different area who are non-technical experts to manage every single step. |

**Task 2**

**Use case:**



**Use case Scenario:**

**Name:** Withdraw Cash

**Brief Description:** The customer withdraw cash from ATM.

**Actors:** Customer, Bank

**Preconditions:**

Network connection is established.

ATM has available cash.

Customer has ATM card.

**Basic Flow:**

Customer insert ATM card. The system read it and prompts for PIN.

The customer enters PIN. The system validates PIN.

ATM displays actions available on ATM unit. Customer selects Withdraw Cash.

ATM request account type.

Customer select account type.

ATM request amount.

Customer enter amount.

Information sent to Bank to validate the availability of requested amount

System asks the customer to withdraw the card. Customer withdraws the card.

ATM dispenses the requested amount of cash and prints receipt.

**Alternate Flow:**

Customer is not validated

ATM displays error message

Customer selects invalid amount.

ATM prompts user to re-enter valid amount.

Customer has insufficient balance

ATM shows error message.

ATM shows available withdrawal limit.

ATM has insufficient cash.

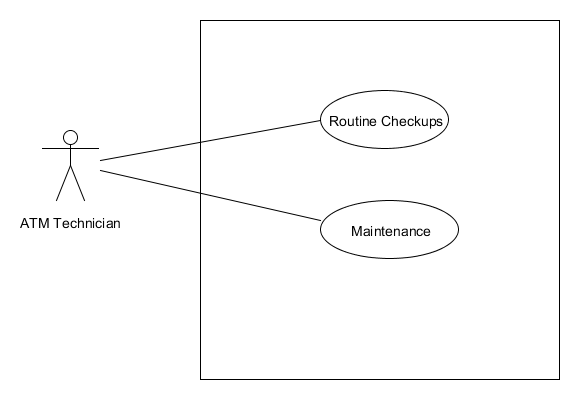
ATM Technician is alerted.

ATM displays error message and phone number to call.

Cash gets stuck in dispensing.

ATM displays error message.

**Use case:**



**Use case Scenario:**

**Name:** Withdraw Cash

**Brief Description:** The Technician will do ATM maintenance.

**Actors:** ATM Technician

**Preconditions:**

No Customer is using ATM.

**Basic Flow:**

Technician will turn the ATM switch to ‘off’ position.

Connection to bank will be shutdown.

Technician will perform maintenance task.

Technician will turn the ATM switch to ‘on’ position.

Connection to bank will be established.

**Test case: 1**

**Title:** Customer Authentication

**Description:**

The authorized user will be successfully login to their account.

**Test steps:**

Insert ATM card.

Enter PIN code.

Press Enter.

**Expected Result**

ATM will show available actions on ATM unit.

**Test case: 2**

**Title:** Insufficient ATM cash

**Description:**

When customer request amount for withdraw and ATM has not enough cash available.

Precondition: Customer is already login.

**Test steps:**

Select Withdraw cash option.

Select account type.

Enter amount

Press Enter.

**Expected Result**

ATM will show error with message of ‘Insufficient amount’

**Test case: 3**

**Title:** Insufficient Customer cash

**Description:**

When customer request more amount to withdraw then available balance in account.

Precondition: Customer is already login.

**Test steps:**

Select Withdraw cash option.

Select account type.

Enter amount

Press Enter.

**Expected Result**

ATM will show error with message of “Your account balance is insufficient”

ATM will show available amount and request to again enter amount.

**Test case: 4**

**Title:** Balance status

**Description:**

When customer want to check his account balance.

Precondition: Customer is already login.

**Test steps:**

Select Check Balance option.

Press Enter.

**Expected Result**

ATM will show account status.

**Task 3**

Solution:

|  |  |
| --- | --- |
| Test Number | Test |
| T1 | (ab, ab, 0) |
| T2 | (ab, a, 0) |
| T3 | (ab, ac, -1) |
| T4 | (ab, c, -1) |
| T5 | (a, bc, -1) |
| T6 | (abc, bc, 1) |
| T7 | (ab, b, 1) |
| T8 | (abc, ba, -1) |
| T4 | (ab, c, -1) |
| T2 | (ab, a, 0) |

1. **Give a minimal test set that satisﬁes all defs coverage. Use the test cases given.**

Since all du-paths can be visited directly, we don’t look side trips. Direct visiting is shown by ‘+’ symbol. The table will elaborate our answer.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | Du paths |
| + |  |  |  |  |  |  |  | [2, 3, 4, 5, 6, 7, 9] |
|  | + |  |  |  |  |  |  | [2, 3, 4, 5, 6, 10] |
|  |  | + |  |  |  |  |  | [2, 3, 4, 5, 6, 7, 8, 10] |
|  |  |  | + |  | + | + | + | [2, 3, 4, 10] |
|  |  |  |  | + |  |  |  | [2, 3, 11] |
|  |  |  |  |  | + |  |  | [10, 3, 4, 5, 6, 7, 9] |
|  |  |  |  |  |  | + |  | [10, 3, 4, 5, 6, 10] |
|  |  |  |  |  |  |  | + | [10, 3, 4, 5, 6, 7, 8, 10] |
|  |  |  | + |  |  |  |  | [10, 3, 4, 10] |
| + | + | + | + |  | + | + | + | [10, 3, 11] |

For all defs coveage , we have to visit one of the du-paths starting with 2 and one with 10. There are almost 7 minimal sets except T5.

1. **Give a minimal test set that satisﬁes all uses coverage.**

From the calculations of table. To tour the def-use relation from 2 to 9 we have to use t1 . t5 is required for association of 2 to 11. And t6  is required for 10 to 9. Similarly these test happened to tour 10 to 11 and 2 to 10 associations. Therefore, we have only three minimal all-uses sets that are given below.

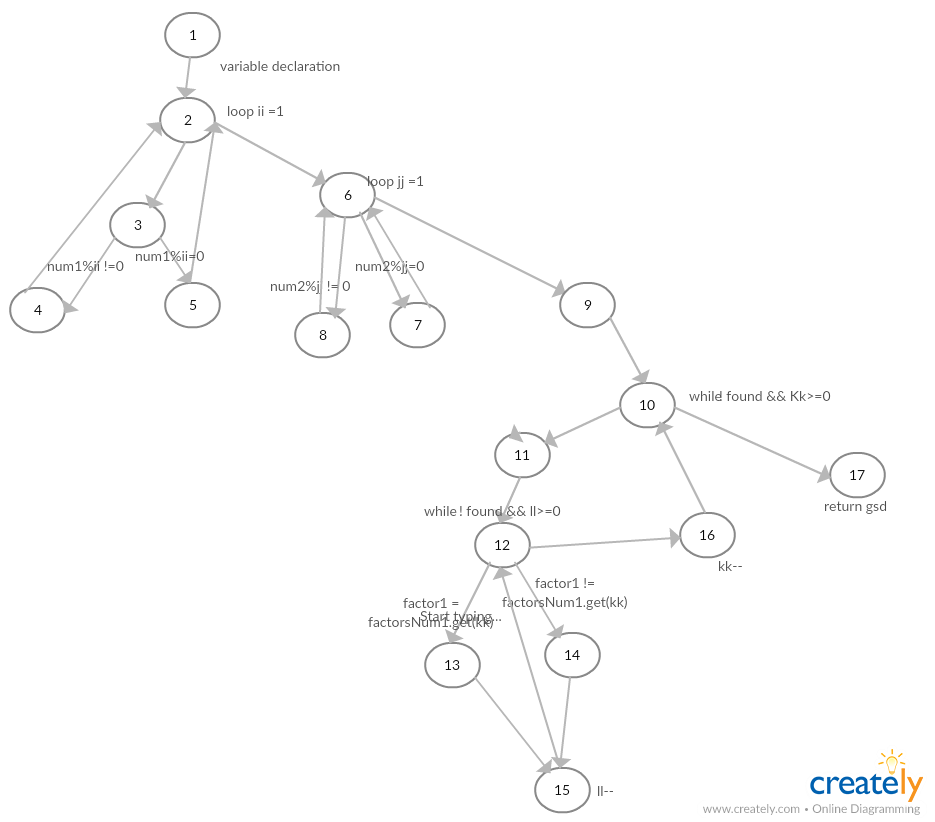
* {t1, t5, t6, t8}
* {t1, t4, t5, t6}
* {t1, t5, t6, t7}

1. **Give a minimal test set that satisﬁes all du-paths coverage.**

All the tests uniquely toured the path so we can conclude the set.

* {t1, t2, t3, t4 t5, t6, t7, t8}

**Task 4**



1. **List the test requirements and a test set for:**
2. **(i). All-du-path Coverage.**

[1,2,3,4,2,6,8,6,9,10,11,12,13,15,12,16,10,17]

[1,2,3,5,2,6,8,6,9,10,11,12,13,15,12,16,10,17]

[1,2,3,4,2,6,7,6,9,10,11,12,13,15,12,16,10,17]

[1,2,3,5,2,6,7,6,9,10,11,12,13,15,12,16,10,17]

[1,2,3,4,2,6,8,6,9,10,11,12,14,15,12,16,10,17]

[1,2,3,5,2,6,8,6,9,10,11,12,14,15,12,16,10,17]

[1,2,3,4,2,6,7,6,9,10,11,12,14,15,12,16,10,17]

[1,2,3,5,2,6,7,6,9,10,11,12,14,15,12,16,10,17]

[1,2,6,9,10,17]

**(ii). Prime Path Coverage.**

[1,2,3,4,2,6,8,6,9,10,11,12,13,15,12,16,10,17]

[1,2,3,5,2,6,8,6,9,10,11,12,13,15,12,16,10,17]

[1,2,3,4,2,6,7,6,9,10,11,12,13,15,12,16,10,17]

[1,2,3,5,2,6,7,6,9,10,11,12,13,15,12,16,10,17]

[1,2,3,4,2,6,8,6,9,10,11,12,14,15,12,16,10,17]

[1,2,3,5,2,6,8,6,9,10,11,12,14,15,12,16,10,17]

[1,2,3,4,2,6,7,6,9,10,11,12,14,15,12,16,10,17]

[1,2,3,5,2,6,7,6,9,10,11,12,14,15,12,16,10,17]

**(b) Undertake logic coverage of the given program:**

**(i). List all predicates.**

* !found && KK>= 0
* !found && ll>=0

**(ii). Define all reachability conditions for the predicates defined in part (i).**

* !found && KK>= 0
  + Found must be false and kk should be greater or equal to zero.
* !found && ll>=0
  + Found must be false and ll should be greater or equal to zero

**(iii). List test cases (and expected output) for:**

1) Predicate Coverage

**Test case: 1**

**Title:** Number less than Zero

**Description:**

The first number entered is less than zero.

**Test steps:**

Enter 1st value in negative and second value correct.

**Expected Result**

Will not processes

**Test case: 2**

**Title:** valid numbers

**Description:**

Both numbers are valid.

**Test steps:**

User enter valid 1st and 2nd number

**Expected Result**

Output should be shown.

**2) Clause Coverage**

**Test case: 1**

**Title:** 1st Number Zero

**Description:**

The first number entered is zero.

**Test steps:**

Enter 1st value zero and second value correct.

**Expected Result**

Loop will not be executed.

**Test case: 2**

**Title:** 2nd number greater

**Description:**

Both numbers are accurate but 2nd one is greater.

**Test steps:**

User enter valid 1st value = 10 and 2nd value =1000

**Expected Result**

Return gsd -1;

**3) Correlated Active Clause Coverage (CACC)**

There does not exists correlated active clause coverage in this code.