

Webdesc Test Plan

Introduction

This test plan addresses the test coverage for the first public beta release of the Webdesc online personal computer system and associated web application.

Primary Testing Concerns

Testing concerns for Webdesc currently focus on stability and data integrity issues. Of primary concern is completion of functionality visible to end users. This is a paradigm not commonly used and the first impressions are crucial.

Primary Testing Focus

Initial testing will concentrate on petty testing of various user inputs and consistency of our database models. User input testing will be related to registration and general administrative tasks at the core of the web application. Further testing will require placing a high level of load on the system to determine conditions that may occur as the system acquires new customers at an accelerated rate.

Personnel

The following is a list of the current development team for Webdesc:

- Manager and Developer: Abe Gurjal
- Developer: Nick Sillik
- Developer: Jon Bringhurst

Testing Schedule

Webdesc is currently following an agile two week development cycle. Iterative processes keep the testing schedule short and repetitive. Each two week cycle is composed of the following:

- Planning and revisions to design documents as needed.
- Test planning.
- Implementation.
- Running of automatic testing suite.

- Manual testing of user input.
- Production push.

Each stage of this process may overlap. Duration of each process in the cycle varies greatly depending on the scope and intended time frame for implementation. Some stages of the cycle will not be completed for each two week period (e.g. in cases of larger sub-projects which require an extended time frame for implementation).

Primary Feature Areas

The entire Webdesc system can be generalized into two distinct areas of operation: "front-end" and "back-end". Webdesc's front-end consists of primary means of user tracking, account creation, as well as a web presence. The area categorized as back-end functionality consists of a clustered file system for storing virtual machines, the configuration of the virtual machine environment (xen), and the xmlrpc server which acts as a controller for starting, stopping, and creating virtual machines.

Operational Issues

Various activity involving the long term as well as day to day operations can be categorized into the following categories:

- Backup
- Recovery
- Monitoring
- Alert methods and issue escalation

These processes should be considered when testing the system as a whole.

Each operational issue ties into the overall user experience. Backup and recovery is currently based on a periodic rsync. Future plans call for migration to an S3 platform to counter inevitable limitations and practicality of rotating backups from a clustered file system.

Acknowledgement and escalation of real time issues is currently integrated into a web based trac system. Methods for response to critical issues may be addressed through use of KVM over IP. Most serious issues can be solved by actual on site administration following a significant travel related delay.

Scope of Test Cases

Test cases will be limited in scope to function correctness with respect to the specification. Cases in which performance issues are deemed to create an extreme burden on the end user will be considered as test case failure. However, performance will be strictly secondary to the ability for a feature to perform as desired. Unit testing will be limited to consistency checks of the database model code. Third party code, such as Glusterfs and Xen, will be tested, configured, and modified within the constraints of the project management of the respective project.

Acceptance Criteria

Criteria for completion of a public beta of the Webdesc system consists of a system with minimal functionality required to perform the critical tasks required to use the system. Also, a relatively acceptable level of stability and security is required by the limited user base. The acceptance criteria for the finished public release will differ greatly in regards to stability from these minimal acceptance criteria for our private beta.

Test Approach

Data Validation

Data validation is primarily handled by Django's form validation framework.

Modes and Run-time Options

In regards to the public beta of Webdesc, the primary mode of operation is the launch of a Linux based Ubuntu operating system. Modes planned for the future, such as a Microsoft Windows based operating system will not be tested at this stage. Run-time options at this stage are not available as each user is given a default configuration.

Client Compatibility

Client compatibility scope is limited to the Windows operating system using the Firefox web browser for testing purposes. Other platforms, although they may appear to function as expected, are not within the testing scope for the public beta.

Performance & Capacity Testing

Stress testing of the Webdesc system for purposes of the public beta will be accomplished by the use of invited end users to utilize the system. Future testing will involve scripted stress testing that should, in time, cover the portions of the system that may be vulnerable to stress (such as CPU usage and storage).

Scalability

Although scalability will not be tested in the public beta, future plans call for inclusion of support for EC2 and S3 to increase capacity for CPU and storage respectively. Scalability has been kept in mind, and will be indirectly tested when the clustering file system and controller are stressed by public beta users.

Bug Reporting

Reporting of issues encountered through testing (as well as development) are stored in a web based ticket system (known as trac). Each ticket is given a priority based on its expected impact and developers are expected to pull from the queue of expected issues as they arise.

Primary External Dependencies

The following software is used in a significant manner in the Webdesc system:

- Ubuntu Linux - <http://www.ubuntu.com/>
- Xen Hypervisor - <http://www.xen.org/>
- Django Web Framework - <http://www.djangoproject.com/>
- Gluster File System - <http://www.gluster.org/>

The requirement of testing of this software will be kept within the constraints of the respective project management.

Sample Of Testing Results

The following is a sample of the test results from the database modification code within the front-end. Other test results are similar in nature and only provide relevant output in the case of test failure.

```
$ python manage.py test
Creating test database...
Creating table django_admin_log
Creating table auth_message
```

Creating table auth_group
Creating table auth_user
Creating table auth_permission
Creating table django_content_type
Creating table django_session
Creating table django_site
Creating table profiles_node
Creating table profiles_virtualmachinesession
Creating table virtual_machine
Creating table profiles_userprofile
Creating table newsletter_newslettersignup
Creating table registration_registrationprofile
Creating table invite_invite
Creating table usertracking_usertracking
Installing index for admin.LogEntry model
Installing index for auth.Message model
Installing index for auth.Permission model
Installing index for profiles.VirtualMachineSession model
Installing index for profiles.UserProfile model
Installing index for registration.RegistrationProfile model
Installing index for usertracking.UserTracking model
.....

Ran 9 tests in 0.046s

OK
Destroying test database...