Team 4 Deliverable 2

First 5 Test Cases

The Testing Process

Martus is considered a finished product, and has been released to the public for some years. Based on the status of the project, our test plan will follow a requirements-based release testing format. Our goal this semester is to, at a minimum, thoroughly test what we consider the most fundamental requirements of the system. To ease into the testing process and familiarize ourselves with Martus, we will start with some less than critical non-functional requirements.

Requirements Traceability

user name and password: user name must not exist in the account directory when creating a new account and must not be the empty string.

Passwords must be at least 8 characters long and is considered weak if the length is less than 15 with less than 2 meta symbols which will warn the user of its lack of complexity.

Password must also be different than the user name entered.

Upon creation of an account, the user must enter both, user name and password, twice.

client authentication: authorizeLog file must exist and contain no blank lines

clientStrings generated from crypto must match authorized client strings

File scrubbing:The system may call a file scrubber that will write “0x55” bytes over all existing bytes in the file.

Create bulletin: Users, once logged in, can create bulletins with summaries of abuse reports.

Tested Items

user name and password: requirements for this are only tested at account creation.

client authentication: requirements tested when client requests server access

file scrubber: org.martus.util.ScrubFile

create bulletin: org.martus.common.bulletin; mainly tests to be sure any functions related to bulletins are private, bulletin deletion is untraceable, and bulletins can be deleted.

Testing schedule

As our software is in release phase and we are not performing any development tasks, our schedule is largely based on our software engineering class schedule and deliverable due dates.

Test Recording Procedures

Devise naming convention that includes date and timestamp for log files. Return results of tests to calling test suite and write a new log file with all test results each time tests are run.

Hardware and Software Requirements

Software requirements:

- Ubuntu Desktop 12.04
- Java 1.6+
- Eclipse Indigo 3.7.2
- MecurialEclipse plugin

Hardware requirements:

- Minimal (requirements unspecified)

Constraints

The team has little permanently scheduled time together. Scheduling conflicts and non-shared workspace presents a challenge.

System Tests

user name and password: user name and password lengths must be satisfied.
User name and password complexity must be met as stated by the requirements.
User name must be unique to the account directory.

client authentication: test for authorizeLog file existence
test for authorizeLog file meeting specific requirements
test for clientString matching list of authorized clientStrings

file scrubber:
Create arrays (“outData” & “inData”) of 100 random bytes.
Write each element in outData consecutively to temp file.
Read temp file in to inData.

♠ Test that inData and outData contain the same elements.
Call ScrubFile.scrub()

♠ Test that inData and outData do not contain the same elements.
Loop through inData and ensure that each byte is equal to “0x55”.
Ensure that exceptions cause test failure (No expected exceptions).

Create bulletin: Create 6 fully capable bulletins

-no functions or attributes to bulletins can be of type Public.
-All docs are time and date stamped when altered.
-draft bulletins are allowed and checked for upon exit.
-all bulletins must have an ID.
- if bulletin's date is erased or empty, current date is added.
-deleted bulletins must be untraceable.
-all bulletin data must be encrypted individually
(that is each characteristic separately)
-ability to find all bulletins posted today.
-allow bulletins to have attachments.