**Testing Specification**

for

**DSS Database Suite**

Version 1.0

Prepared by Iain Smith and Austyn Krutsinger

May 4, 2009

Table of Contents

1. Overview ………………………………………………………………………………………….. 3
2. Test Planning ………………………………………………………………………………….... 3
3. Test Specifications ……………………………………………………………………………. 4

3.1 Phase 1 Specifications ………………………………………………………………… 4

3.2 Phase 2 Specifications ………………………………………………………………... 5

3.3 Phase 3 Specifications …………………………………………………………………. 6

1. Test Results ………………………………………………………………………………………. 9
2. **Overview**

The DSS Database Suite is composed of two distinct projects. The first, and primary project, is called DatabaseSuite, and it encompasses all files pertaining to the structure and implementation of the DSS database, including personal information, equipment loan information, and student case notes. The second, peripheral project, is called A2MConvert, and it attempts to convert all existing records in DSS’s pre-existing Microsoft Access database over to the new database structure implemented in the project DatabaseSuite, which uses MySQL databases.

1. **Test Planning**

Testing of the DSS software occurred in distinct phases. Phase 1, encompassing the first seven weeks of the semester, from mid-January to early-March 2009, included basic testing of data field input to ensure data was transferring and saving to the MySQL database, as well as multi-user and remote user testing. Phase 2, encompassing the second five weeks of the semester, from Early-March to Early-April 2009, is when conversion testing occurred, ensuring that the database conversion program did its job and converted all data fields appropriately. Phase 3, encompassing the last four weeks of the semester, from Early-April to the beginning of May, is when conclusionary testing was done. This wrapped up testing of the equipment table and notes testing on the conversion program, as well as structural testing on the database program.

The term structural testing in the previous paragraph is used to mean testing on the structure of the code to ensure abnormal result do not occur when do not occur for taking such actions as cancelling out of a query results page.

1. **Test Specification**
	1. **Phase 1 specifications**

For this table, success = expected result, and failure = unexpected result.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Condition** | **Description of Condition** | **Test Steps** | **Expected Results** | **Results.** |
| Login – 1 | Enter correct information.  | User = user; Password = password; Correct db info. | Connect to the database. | Success. |
| Login – 2 | Enter incorrect db info. | User = user; Password = password; table = aTable. | Fail to connect to database. | Success. |
| Login – 3 | Enter incorrect login info. | User = password; Password = user; Correct db info. | Fail to connect to database. | Success. |
| Field Input – 1 thru Field Input – 45 | Insert information into text fields and add record. | DawgTag = 111111111; SSN = 111111111; Rest of the fields = anything. | Successfully add record. | Success. |
| Duplicate – 1 | Add record with the same DawgTag. | DawgTag = 111111111; SSN = 222222222; Rest of the fields = anything. | Fail to add record. Error. | Success. |
| Duplicate – 2 | Add record with the same SSN. | DawgTag = 222222222; SSN = 111111111; Rest of the fields = anything. | Fail to add record. Error. | Success. |
| Query – 1 thru Query – 100  | Query field combinations. | DawgTag, SSN, MiddleInitial, DawgTag + SSN, FirstName + PrimaryDisability, etc. | Query results matching the query. | Success. |
| Load Record – 1  | Load existing query. | Query Database, Select Record, Select OK to load. | Correct, existing record loads. | Success. |
| Update Record – 1 thru Update Record – 40 | Update a data field in an existing record. | Load record, change a field, update record. | Record updates. | Success. |
| Update Record – 41  | Update SSN of existing record. | Load record, change SSN to 333333333, update record. | Record updates. | Success. |
| Update Record – 42  | Update DawgTag of existing record. | Load record, change DawgTag, update record. | Record updates. | Failure. (1) |
| Multi-user connect – 1  | Connect a second user. | User 1 connects to database. User 2 attempts to connect to database. | User 2 connects to database. | Success. |
| Remote Connect – 1  | User 2 connects to user 1’s database. | User 2 connects, server = (user 1’s IP address); rest of the fields the same as Login – 1. | User 2 connects. | Success. |
| Logout – 1  | User disconnects from database. | User connects to database. Use clicks disconnect. | User disconnects from database. | Success. |

1. Record fails to update because no record exists in the database with the key of the new DawgTag number. Solution: Remember old DawgTag number so it can be appropriately replaced.
	1. **Phase 2 specifications**

Phase 2 testing was a little different than Phases 1 and 3. During Phase 2 testing, testing was taking place on the conversion software meant to convert the pre-existing database to the new MySQL database being developed using Visual C#. As such, testing was done solely by running the code on samples of the pre-existing database, since what really matters is that it works one time to convert the entire database over.

Due to the way access was given to the Access database in question, Phase 3 testing was done in 3 phases. The first phase of testing was done on a 10 record sample obtained in late February. The second phase was done using a 50 record sample obtained in early March. The third phase of testing was done using the full 2800 records obtained in early April.

Furthermore, each phase of testing uncovered more inadequacies in the code because more anomalies in the Access database were uncovered only as larger samples of the Access database were received.

As a result of the testing procedure mentality explained above, there were no test cases in phase 2 beyond run the code and see if it succeeds. This “test case” was run many times, hundreds of times, as more and more of the code was completed and revised and added to.

Iterations of this case shrank the number of unconvertible anomalies from 200+ to 80 by the end of the semester. Solutions to code crashes included converting double quotes to single quotes in pet names and nicknames to thwart MySQL syntax errors, increasing field sizes as longer field values were discovered for such fields as Primary disability and home addresses.

The most extensive testing in this phase, however, was to the fields of birthdates and phone numbers, which experienced the largest number of variance in format.

* 1. **Phase 3 specifications**

Similar to phase 2 testing, phase three consists of several type phases of testing. This time, the phases are different types of testing altogether. Type 1 testing is a wrap up of phase 1 style testing on the newly implemented code.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Condition** | **Description of Condition** | **Test Steps** | **Expected Results** | **Results.** |
| Query Results -1 thru Query Results 45  | Add and remove each data field from query results visibility.  | Query database and get some results. Add each data field. Remove each data field. (Fields include DawgTag, SSN, HomePhone, etc.) | Data field becomes visible, then it disappears. | Success. |
| Equipment – 1 | Add Equipment record. | Right-click equipment tab. Select add. Fill out form. Click Okay. (SIUC Number = 123456; Name = wheelchair; Date checked out = 01/01/2009; Date Due = 05/04/2009) | Equipment added to equipment table. | Success. |
| Equipment – 2  | Edit equipment record. | Right-click equipment tab on an existing record. Select edit. Add note = flat tire. Click okay. | Equipment updated with note. | Success. |
| Equipment – 3 | Delete an equipment record. | Right-click and existing equipment record. Select delete. | Equipment record deleted. | Success. |
| Update Record – 1 thru Update Record – 4 |  Update new fields in an existing record. | Load Record. Change new field. Select Update. (New fields = HousingAssistance, CaseWorker, CaseNotes, Disabled.) | Record updated with new information. | Success. |

Type 2 testing is a wrap up of phase 2 style testing on the newly implemented data fields. Since some of these fields are new to the DatabaseSuite project, they are simply empty strings in the conversion program. Primarily, this testing is on the equipment table conversion of the Access database to the MySQL database. Similar to the student information database, some fields were unconvertible and had to be appended to the record’s notes.

Type 3 testing is a testing of the DatabaseSuite code to discover any locations in the code where the Dataset containing the current record gets wiped when it shouldn’t. Initial discovery of this type of bug occurred during presentations to DSS stakeholders and CS departmental faculty in mid-March.

The primary way to enact such an error is to query the database then cancel out of the query results page back to a pre-loaded record. Due to coding practices, it was discovered that the dataset was “pre-deleted” on the occurrence of a query to make space for the query dataset. Once this was corrected, the problem was alleviated until the addition of the equipment table, when it arose again. Test specifications to discover these issues, primarily after the fact of initial discovery, looked like so:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Condition** | **Description of Condition** | **Test Steps** | **Expected Results** | **Results.** |
| Update Record – 1  | Update modified record. | Load existing record. Query database. Cancel out of results. Modify record. Select update. | Record updates. | Failure. |
| Equipment – 1 | Add Equipment record. | Load existing record. Query database. Cancel out of results. Right-click equipment tab. Select add. Fill out form. Click Okay. (SIUC Number = 123456; Name = wheelchair; Date checked out = 01/01/2009; Date Due = 05/04/2009) | Equipment added to equipment table. | Failure. |
| Equipment – 2  | Edit equipment record. | Load existing record. Query database. Cancel out of results. Right-click equipment tab on an existing record. Select edit. Add note = flat tire. Click okay. | Equipment updated with note. | Failure. |
| Equipment – 3 | Delete an equipment record. | Load existing record. Query database. Cancel out of results. Right-click and existing equipment record. Select delete. | Equipment record deleted. | Failure. |

1. **Testing Results**

As a result of all these tests, over 250 on the DatabaseSuite and 100s more ensuring A2MConvert successfully converted the Access Database, a number of debilitating bugs were removed from the code, ensuring a more stable product for the stakeholders, Disability Support Services. Since their discovery, all listed failures have been corrected and no longer occur upon retest under the same conditions.