

**Test Plan**

for

**Exam Preparation Software (XPS)**

**PrepT Software**

version 1.0

**Team Members**

Mason Ellis

Joshua Regino

Raul Escatel

Johnny Tran

Scott Schumacher

**Date: 04/11/2017**

**Table of Content**

**1. Purpose of this document….………………………………………………………………………………………………………….3**

**2. Strategy…….………..…………………………………………………………………………………………………………………………3**

**3. Responsibilities…….……..………………………………………………………………………………………………………………..4**

**4. Milestones………….…………………………………………………………………………………………………………………………4**

**5. Procedures…………………………………………………………………………………………………………………………………….5**

**5.1 Database Manager Unit Testing………………………………………………………………………………………..5**

5.1.1

**5.2 Integration Testing, Forms………………………………………………………………………………………………10**

**Appendix A: Glossary…………………………………………………………………………………………………………………….….16**

1. **Purpose:** The purpose of this document is to describe the testing strategy, procedures, milestones, and responsibilities for testing the Exam Preparation Software (XPS) before its release to the customer.
2. **Strategy:** Our testing strategy stands on three principles, which are: testing based on use cases; the two-person rule in developing / testing; and accelerated integration.
   1. **Testing based on use cases:** Initial testing will focus on functionality. Since there are relatively few well-defined use cases for XPS, the team’s efforts will be focused on making those use cases happen. The team will test all the corners and unusual combinations that could cause those use cases to fail, with the intent of discovering as many of the System’s weaknesses as possible.
   2. **Two-person rule in developing / testing:** No element of the software will be written and tested by the same person. The philosophy here is to get as many eyes on the product as possible, in the hope that the greatest number of flaws and potential problems will be identified as early in the process as possible. Team members are empowered to “task” one another with testing duties for the modules they write, and all testing will be reviewed by the team leader, meaning that at least three members will have had a hand in developing / testing any given part of the system.
   3. **Accelerated Integration:** Unit testing, which tests modules in isolation from other elements of the system, frequently requires writing special testing software in addition the product. Especially considering the limited time, team experience, and access to in-house testing software, Unit Testing will be minimized as much as possible; the development team will move as quickly as possible to Integration and System testing. The key component of XPS is the Database Manager, which mediates all transactions between the system components and the database. All of the rest of the elements of XPS interact with and rely on the Database Manager. A special Test Client has been developed for the Database Manager, and the Database Manager has been extensively unit tested. Most of the other modules of the XPS interact directly with the Database Manager, and will have their functions verified with and through the Database Manager.
3. **Responsibilities:** At PrepT, quality is everyones’s responsibility; specific duties for XPS testing are listed below:

**Mason Ellis:** Team Leader, Tester

**Scott Schumacher** Test Plan writer, Tester

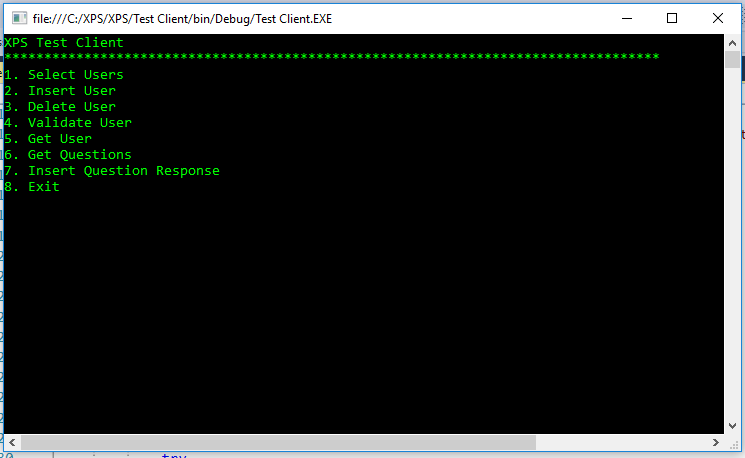
**Josh Regino** Tester

**Johnny Tran** Tester

1. **Milestones:** The milestones for the testing process and the schedule are detailed below.

* Deliverables and Artifacts Due
  + Draft Test Plan: April 11, 2017
  + Final Test Plan: April 20, 2017
* Test Schedule
  + Unit Testing: March 25 – April 25, 2017
  + Integration Testing: March 30 – April 25, 2017
  + System Testing: April 26 2017 – May 2, 2017
  + Security and Stress Testing: April 26- May 2, 2017

1. **Procedures:** Database Manager is unit tested; for all other components, testing is driven by use cases. For each use case, a well-defined set of steps by the user and responses by XPS completes a use case. Use cases will be carefully studied to identify combinations of test data in an effort ***break the system.*** Early identification of flaws will yield a higher quality, more robust product.
   1. **Database Manager unit testing**: The following Test Client software is constructed solely for the unit testing of the database manager and simulates input from the various modules of the XPS. Figure 1 illustrates what the user interface of the XPS Test Client, and Figures 2 links the Test Client’s functions with the Database Manager functions they test.



**Figure 1: XPS Test Client user interface**

|  |  |
| --- | --- |
| **UNIT TEST:** DATABASE MANAGER | |
| **DATABASE MANAGER METHOD TESTED** | **TESTED BY Test Client function:** |
| public User ValidateUser(String userName, String passWord) | 4. Validate User |
| public User GetUser(int userID) | 5. Get User |
| public List<Question> GetQuestions(int n, int[] categories) | 6. Get Questions |
| public bool InsertUser(User user) | 2. Insert User |
| public bool InsertQuestionResponse(QuestionResponse qr) | 7. Insert Question Response |
| public bool DeleteUser(int userID) | 3. Delete User |

**Figure 2: Database Manager functions and correspoding Test Client functions.**

In addition to the unit testing of the Database Manager, the Database Manager is continuously tested during the integration testing of other parts of XPS.

* + 1. Test cases used in unit testing of the Database Manager: Information in the database must be complete and correct; data fields can not be absent or of an incorrect format. Unit testing of the database manager is based on these criteria. The following figures detail the procedures used to test each use case listed in Figure 2. Figure 3 lists the test conditions, the expected results, and the Pass / Fail status for the Validate User method when tested with the Test Client.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database Manager Method:** public User ValidateUser(String userName, String passWord) | | | | | |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1. | Test for missing user name | User name: “” password: Test123 | Message displayed:  Invalid user name or password | Message displayed:  Invalid user name or password | PASS |
| 2 | Test for missing password | User name: sschumacher  password: “” | Message displayed:  Invalid user name or password | Message displayed:  Invalid user name or password | PASS |
| 3 | Test for unauthorized user | User name: swhiplash  password: Test123 | Message displayed:  Invalid user name or password | Message displayed:  Invalid user name or password | PASS |
| 4. | Test for authentic user | User name: sschumacher password: Test123 | User validated: full name displayed | User validated: full name displayed | PASS |
| 5. | Test for password case | User name: sschumacher password: test123 | Message Box displayed:  Incorrect user name or password | User validated: full name displayed | FAIL |

**Figure 3: Database Manager Tests: Validate User**

* + 1. Figure 4, below, lists the test conditions, the expected results, and the Pass / Fail status for the Get User method when tested with the Test Client.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database Manager Method:** public User GetUser(int userID) | | | | | |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1. | Test for correct user ID | Mustang ID: 5 | User validated: Raul Escatel | User validated: Raul Escatel | PASS |
| 2. | Test for invalid user ID | Mustang ID: 47 | Message displayed: Invalid user ID | Message displayed: Invalid user ID | PASS |
| 3 | Test for missing user ID | Mustang ID: “” | Message: input string was not in correct format | Message: input string was not in correct format | PASS |

**Figure 4: Database Manager Tests: Validate User**

* + 1. Figure 5, below, lists the test conditions, the expected results, and the Pass / Fail status for the Get Questions method when tested with the Test Client.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Database Manager Method:** public List<Question> GetQuestions(int n, int[] categories) | | | | | | |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Passing Criteria | Test Result |
| 1. |  |  |  |  |  | |
| 2. |  |  |  |  |  | |
| 3 |  |  |  |  |  | |

**Figure 5: Database Manager Tests: Get Questions**

* + 1. Figure 6, below, lists the test conditions, the expected results, and the Pass / Fail status for the Insert method when tested with the Test Client.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database Manager Method:** public bool InsertUser(User user) | | | | | |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1. | Valid new user | User ID: 31322001  First name: Dave  Last name: Bowman  Username: dbowman  Password: Test 2001  User type: S | User entered. | User entered. | PASS |
| 2. | Duplicate user | User ID: 5  First name: Raul  Last Name: Escatel  User name: rescatel  Password: Test123  User type: A | Message: Duplicate entry ‘5’ for primary key | Message: Duplicate entry ‘5’ for primary key | PASS |
| 3 | Blank field: user ID | User ID: “”  First name: Dave  Last name: Bowman  Username: dbowman  Password: Test 2001  User type: S | Message: Input string was not in correct format. | Message: Input string was not in correct format. | PASS |
| 4 | Blank field:  First Name | User ID: 555  First name: “”  Last name: Bowman  Username: dbowman  Password: Test 2001  User type: S | Message: Input string was not in correct format. | Message: Index was outside the bounds of the array. | FAIL |
| 5 | Blank field:  last Name | User ID: 555  First name: Tommy  Last Name: “”  User name: ttutone  Password: 8675309  User type: S | Message: Input string was not in correct format. | Message: Index was outside the bounds of the array. | FAIL |
| 6 | Blank field:  User name | User ID: 22  First name: Tommy  Last Name:Tutone  User name: “”  Password: 8675309  User type: S | Message: Input string was not in correct format. | Message: Index was outside the bounds of the array. | FAIL |

**Figure 6: Database Manager Tests: Insert User (Continued next page)**

**Continued from previous page:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | Blank field:  Password | User ID: 22  First name: Tommy  Last Name: Tutone  User name: ttutone  Password: “”  User type: S | Message: Input string was not in correct format. | Message: Index was outside the bounds of the array. | FAIL |
| 8 | Blank field:  User type | User ID: 22  First name: Tommy  Last Name: Tutone  User name: ttutone  Password: 8675309  User type: “” | Message: Input string was not in correct format. | Message: Index was outside the bounds of the array. | FAIL |

**Figure 6: Database Manager Tests: Insert User (Continued from previous page)**

* + 1. Figure 7, below, lists the test conditions, the expected results, and the Pass / Fail status for the Insert Question Response method when tested with the Test Client.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1. |  |  |  |  |  |
| 2. |  |  |  |  |  |
| 3 |  |  |  |  |  |

**Figure 7: Database Manager Tests: Insert Question Response**

* + 1. Figure 8, below, lists the test conditions, the expected results, and the Pass / Fail status for the Delete User method when tested with the Test Client.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1. | Delete existing user | User ID: 31322001 | Are you sure that you would like to delete user 31322001 (Y/N). | Are you sure that you would like to delete user 31322001 (Y/N). | PASS |
| 3. | Answer “N” to above prompt | “N” | Return to main menu | Return to main menu | PASS |
| 2. | Answer “Y” to above prompt | “Y” | User deleted | User deleted | PASS |
| 4. | Delete nonexistent user | 2247636 | Message: invalid user ID | Are you sure that you would like to delete user 2247636 (Y/N). | FAIL |

**Figure 8: Database Manager Tests: Delete User**

* + 1. Summary of Database Manager with the Test Client: For most of the cases and data sets listed, the Database Manager performed as expected. Of the failures, particularly of the Insert User method where one or more fields was blank and the error message returned addressed “index outside of array bounds,” the discrepancy may be a feature of the Test Client rather than of the unit under test, the Database Manager. Integration testing, specifically of the XPS Add User and Delete User forms, has demonstrated this to be the case. Section 5.2 details procedures, criteria, and results for integration testing of the various forms listed therein.
  1. **Log in use cases**: The Administrator and Student logins are considered separate use cases, despite their similarity. The important difference is that logging in with Administrative credentials gives a user access to all normal functions, such as taking a test, but in addition gives access to the Admin Form, which has functions for adding and deleting users and viewing reports.
     1. **Item to be tested:** Figure 9, below, describes the student Log on use case. Problems anticipated include: misssing information- one or more fields left blank; Invalid information- one or more user fields not in the database; and (If time permits) common security bypasses, particularly SQL injection.

|  |  |
| --- | --- |
| **USE CASE:** LOG IN TO XPS (Student) | |
| **Actor**:Student | 1. XPS loaded on computer |
| 1. TUCBW user starts XPS. | 1. XPS displays login form |
| 1. User enters username and password | 1. XPS validates user information;  * If valid, displays main menu (4.1) * If invalid, displays message box (4.2) |
| 1. TUCEW the user sees the main menu. | |

**Figure 9: Student login to XPS**

* + 1. Figure 10, below, lists the test conditions, the expected results, and the Pass / Fail status for the Log on form (student credentials) when tested in conjunction with the Database Manager and database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use case tested: Log on (Student)** | | | | | |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1. | valid username and password | Username: rschumacher  Password: formula1 | XPS displays Main menu:  Ralph Schumacher, student | XPS displays Main menu:  Ralph Schumacher, student | PASS |
| 2. | Valid username;  Invalid password | Username: rschumacher  Password: soccer | XPS displays message box: username / password incorrect | XPS displays message box: username / password incorrect | PASS |
| 3. | Invalid username; valid password | Username: tschumacher  Password: formula1 | XPS displays message box: username / password incorrect | XPS displays message box: username / password incorrect | PASS |
| 4. | Valid username; no password | Username: tschumacher  Password: “” | XPS displays message box: username / password incorrect | XPS displays message box: username / password incorrect | PASS |
| 5. | No username;  Valid password | Username: “”  Password: formula1 | XPS displays message box: username / password incorrect | XPS displays message box: username / password incorrect | PASS |

**Figure 10: Student login to XPS test conditions and results**

* + 1. **Item to be tested:** Figure 11, below, describes the administrator Log on use case. Problems anticipated include: misssing information- one or more fields left blank; Invalid information- one or more user fields not in the database; and (If time permits) common security bypasses, particularly SQL injection.

|  |  |
| --- | --- |
| **USE CASE**: LOG IN TO XPS (Administrator) | |
| **Actor:**Administrator | 1. XPS loaded on computer |
| 1. TUCBW user starts XPS. | 1. XPS displays login form |
| 1. User enters administrator username and password | 1. XPS validates user information;  * If valid, displays main menu with “Admin” button (4.1) * If invalid, displays message box (4.2) |
| 1. TUCEW the admin user sees the main menu with the “Admin” button | |

**Figure 11: Student login to XPS test conditions and results**

* + 1. Figure 12, below, lists the test conditions, the expected results, and the Pass / Fail status for the Log on form (student credentials) when tested in conjunction with the Database Manager and database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use case tested: Log on (Administrator)** | | | | | |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1. | valid username and password | Username: sschumacher  Password: Test123 | XPS displays Main menu:  Scott Schumacher, Admin | XPS displays Main menu:  Scott Schumacher, Admin | PASS |
| 2. | Valid username;  Invalid password | Username: sschumacher  Password: soccer | XPS displays message box: username / password incorrect | XPS displays message box: username / password incorrect | PASS |
| 3. | Invalid username; valid password | Username: dschumacher  Password: Test123 | XPS displays message box: username / password incorrect | XPS displays message box: username / password incorrect | PASS |
| 4. | Valid username; no password | Username: sschumacher  Password: “” | XPS displays message box: username / password incorrect | XPS displays message box: username / password incorrect | PASS |
| 5. | No username;  Valid password | Username: “”  Password: Test123 | XPS displays message box: username / password incorrect | XPS displays message box: username / password incorrect | PASS |
| 6. | Student username;  Admin password | Username:  Rschumacher  Password: Test123 | XPS displays message box: username / password incorrect | XPS displays message box: username / password incorrect | PASS |

**Figure 12: Administrator login to XPS test conditions and results**

* 1. **Administrative use cases:** Among the functions reserved to Administor- level users are adding and deleting users from the database. Particular care in this area must be given to ensuring that data is validated for format (always) and for content where applicable; in addition, Administrative users must be warned and reminded before they commit irreversible or unrecoverable changes to the database.
     1. **Item to be tested:** Figure 13, below, describes the administrative Add User use case. Problems anticipated include: missing information- one or more fields left blank; Invalid information- one or more user fields not in the database or improperly formatted. As an example, the user id field requries an integer. The XPS must not accept input for a user id that includes characters such as letters or punctuation. Also, (If time permits) common security bypasses, particularly SQL injection, should be tested.

|  |  |
| --- | --- |
| **USE CASE**: ADD USER | |
| **Actor**:Administrator | 1. XPS displays main menu with “Admin” button |
| 1. TUCBW user clicks the “Admin” button | 1. XPS displays Admin Form |
| 1. Admin clicks “Add User” button | 1. XPS displays Add User Form |
| 1. Admin enters data to create new user and clicks “Add” button | 1. XPS validates information;  * if correct and complete, creates new user and displays message box confirming creation * if incorrect / incomplete, displays message box instructing user to enter correct / complete information |
| 1. TUCEW user clicks the “Close” or “Admin Menu” button | |

**Figure 13: Administrative Add User use case**

* + 1. Figure 14, below, lists the test conditions, the expected results, and the Pass / Fail status for the Add User form (administrative credentials) when tested in conjunction with the Database Manager and database. It is important to note that the full- length M number is accepted by the database through the Database Manager; this was a failed test when tested using the Test Client.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use case tested: Add User (Administrator)** | | | | | |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1. | Add valid user; all data correct | First name: Fred  Last name: Flintstone  User name: fflintstone  Password: yabbadabba  MustangID: 222551234  Type: S | Message Box:  “Fred Flintstone added to database.” | Message Box:  “Fred Flintstone added to database.” | PASS |
| 2. | Add valid user; first name missing | First name:  Last name: Flintstone  User name: fflintstone  Password: yabbadabba  MustangID: 222551234  Type: S | Message Box:  “First name can’t be blank.” | Message Box:  “First name can’t be blank.” | PASS |
| 3. | Add valid user; last name missing | First name: Fred  Last name:  User name: fflintstone  Password: yabbadabba  MustangID: 222551234  Type: S | Message Box:  “Last name can’t be blank.” | Message Box:  “Last name can’t be blank.” | PASS |
| 4. | Add valid user; user name missing | First name: Fred  Last name: Flintstone  User name:  Password: yabbadabba  MustangID: 222551234 | Message Box:  “user name can’t be blank.” | Message Box:  “user name can’t be blank.” | PASS |
| 5. | Add valid user; password missing | First name: Fred  Last name: Flintstone  User name: fflintstone  Password:  MustangID: 222551234 | Message Box:  “password can’t be blank.” | Message Box:  “password can’t be blank.” | PASS |
| 6. | Add valid user; Mustang ID missing | First name: Fred  Last name: Flintstone  User name: fflintstone  Password: yabbadabba  MustangID: | Message Box:  “Please enter the Mustang ID without the M.” | Message Box:  “Please enter the Mustang ID without the M.” | PASS |
| 7. | Add valid user; Mustang ID misformatted | First name: Fred  Last name: Flintstone  User name: fflintstone  Password: yabbadabba  MustangID: M222551234 | Message Box:  “Please leave off the M from the M number.” | Message Box:  “Please leave off the M from the M number.” | PASS |
| 8. |  |  |  |  | PASS |

**Figure 14: Administrator login to XPS test conditions and results**

5.3.3 **Item to be tested:** Figure 15, below, describes the administrative Delete User use case. The Admin user need know only the user ID of the user to be deleted. The only anticipated problem is missing or malformed data.

|  |  |
| --- | --- |
| **USE CASE**: DELETE USER | |
| **Actor**:Administrator | 1. XPS displays main menu with “Admin” button |
| 1. TUCBW user clicks the “Admin” button | 1. XPS displays Admin Form |
| 1. Admin clicks “Delete User” button | 1. XPS displays Delete User Form |
| 1. Admin enters data to create new user and clicks “Add” button | 1. XPS validates information;  * if correct and complete, creates new user and displays message box confirming creation * if incorrect / incomplete, displays message box instructing user to enter correct / complete information |
| 1. TUCEW user clicks the “Close” or “Admin Menu” button | |

**Figure 15: Administrative Delete User use case**

5.3.4 Figure 16, below, lists the test conditions, the expected results, and the Pass / Fail status for the Delete User form (administrative credentials) when tested in conjunction with the Database Manager and database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use case tested: Delete User (Administrator)** | | | | | |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1. | Delete valid user; all data correct | 16 | Delete User box displays  First Name: Richard  Last name: Simpson  “If this is the user to delete, click the "Delete User" button. Otherwise, click the "Cancel" button.” | Delete User box displays  First Name: Richard  Last name: Simpson  “If this is the user to delete, click the "Delete User" button. Otherwise, click the "Cancel" button.” | PASS |
| 1.1 | Select Cancel at above prompt | Cancel Button clicked | All fields return blank;  Richard Simpson not deleted. | All fields return blank;  Richard Simpson not deleted. | PASS |
| 1.2 | Select Delete at above prompt | Delete Button clicked | Message Box: “Do you want to delete Richard Simpson? | Message Box: “Do you want to delete Richard Simpson? | PASS |
| 1.3 | Select No at confirmation prompt | ‘No’ button clicked | All fields return blank;  Richard Simpson not deleted. | All fields return blank;  Richard Simpson not deleted. | PASS |
| 1.4 | Select Yes at confirmation prompt | ‘Yes’ button clicked | Message Box: “Richard Simpson deleted from Users table.” | Message Box: “Richard Simpson deleted from Users table.” | PASS |
| 2. | Enter invalid user ID | M71 | Message Box: “No record found for that user ID.” | Message Box: “No record found for that user ID.” | PASS |
| 3. | Enter no user ID | “” | Message Box: “No record found for that user ID.” | Message Box: “No record found for that user ID.” | PASS |

**Figure 16: Administrator login to XPS test conditions and results**

5.3.5**Item to be tested:** Figure 17, below, describes the administrative View Reports use case. The only input required from the administrator is selecting the report desired.

|  |  |
| --- | --- |
| **USE CASE**: VIEW REPORTS | |
| **Actor**:Administrator | 1. XPS displays main menu with “Admin” button |
| 1. TUCBW user clicks the “View Reports” button | 1. XPS displays Reports Form |
| 1. Admin clicks “the Report She Wants” button | 1. XPS displays requested report |
| 1. TUCEW user clicks the “Close” or “Admin Menu” button | |

**Figure 17: Administrative View Reports use case**

5.3.6 Figure 18, below, lists the test conditions, the expected results, and the Pass / Fail status for the Delete User form (administrative credentials) when tested in conjunction with the Database Manager and database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use case tested: View Reports (Administrator)** | | | | | |
| # | Description | Input | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1. |  |  |  |  |  |
| 2. |  |  |  |  |  |
| 3. |  |  |  |  |  |

**Figure 18: Administrator View Reports test conditions and results**

**5.4 Test-taking use cases:** This is the reason for the XPS; all other functions exist to support the XPS’s ability to present and evaluate practice tests to students preparing for the Major Field Test (MFT) in Computer Science. The student uses two forms in the test-taking process: the Main menu, where he or she will specify the categories to be tested, how many questions on the test, and whether the test is to be timed or not. Once the student has set all the parameters for the test, she will then click the GENERATE TEST button, at which the time the test will be composed and presented on the Exam form. The student then completes the test. There is no text input required from the student in the test-taking process; all inputs are through radio buttons, check boxes, or drop-down menus. Malformed data, therefore, is not likely a source of errors, but missing or incomplete data, which might take the form of a student forgetting to check one of the required fields, will probably occur frequently.

* + 1. **Item to be tested:** Figure 19, below, describes the administrative Take Test use case.

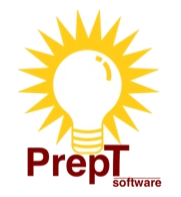
|  |  |
| --- | --- |
| **USE CASE**: TAKE TEST | |
| **Actor:**Student / Admin | 1. XPS displays Main Menu Form |
| 1. TUCBW user clicks the “Take Test” button | 1. XPS Main Menu displays test options, including categories, timed yes / no, and number of questions |
| 1. User enters desired options and clicks “Generate Test” button | 1. XPS validates information for correctness / completeness  * If complete / correct, generates test, displays Test Form * If incomplete / incorrect, displays message box with instructions |
| 1. User clicks “Start” button | 1. XPS displays conditions, illustrations, questions and possible answers; starts timer and displays time remaining |
| 1. User selects radio button representing best answer, clicks “Submit” button to continue | 1. XPS evaluates, stores response info |
| 1. TUCEW user clicks the “Close” or “Main Menu” button | |

**Figure 19: Take Test use case**

* + 1. Figure 18, below, lists the test conditions, the expected results, and the Pass / Fail status for the Delete User form (administrative credentials) when tested in conjunction with the Database Manager and database.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use case tested: Take Test (Student)** | | | | | | |
| # | Description | Input | | Expected Output (EO) | Actual Output (AO) | Test Result |
| 1 | Main menu:  Valid selection | Categories checked: | Discr. Structures  Programming  Algorithms / Complexity | Exam Form;  20 question displayed from selected categories only;  Clock running. |  |  |
| Timed? | Yes (default) |
| Number of questions | 20 |
| 2 | Main menu:  Valid selection | Categories checked: | Discr. Structures  Programming  Algorithms / Complexity | Exam Form;  20 question displayed from selected categories only;  Clock not displayed. |  |  |
| Timed? | No |
| Number of questions | 20 |
| 3 | Main menu:  Invalid selection | Categories checked: | none | Message Box:  “Please select one or more categories.” |  |  |
| Timed? | Yes (default) |
| Number of questions | 20 |
| 4 | Main menu:  Invalid selection | Categories checked: | Discr. Structures  Programming  Algorithms / Complexity | Message Box:  “Please select the number of questions for the test.” |  |  |
| Timed? | Yes (default) |
| Number of questions | Not selected |
| 5 | Main menu:  Valid selection | Categories checked: | Discr. Structures  Programming  Algorithms / Complexity |  |  |  |
| Timed? | Yes (default) |
| Number of questions | 20 |
| 6 | Main menu:  Valid selection | Categories checked: | Discr. Structures  Programming  Algorithms / Complexity |  |  |  |
| Timed? | Yes (default) |
| Number of questions | 20 |
| 1 | Main menu:  Valid selection | Categories checked: | Discr. Structures  Programming  Algorithms / Complexity |  |  |  |
| Timed? | Yes (default) |
| Number of questions | 20 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Figure 18: Administrator View Reports test conditions and results**

**6. Contact Information**

XPS will give support to its customers and clients when the need arises. Many helpful hints about XPS can be found under the Help tab found on the menu bar. For contact information please see below.

**6.1 Administration:** For students who need access, have general questions about XPS or   
 needs to be added/removed from XPS please contact the administrator below.

Dr. Ranette Halverson Bolin Science Hall 126A

Chair, Computer Science 940.397.4189

Midwestern State University [ranette.halverson@mwsu.edu](mailto:ranette.halverson@mwsu.edu)

**6.2 PrepT Software:** Administers that need updates or modifications to the existing version   
 please contact:

Mason Ellis Moffett Library 225

Project Manager 940.397.2323

PrepT Software masonellis@prept.com

**Appendix A Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Database** | Collection of data that is specially organized for rapid search by the software. |
| **MFT** | Major Field Test- A national standardized exam administered organization |
| **PDT** | Prept Development Team- Members from Prept Software Company |
| **User** | Student or Admin that use the program |
| **Windows** | Family of operating systems from Microsoft Corporation |
| **XPS** | Exit Preparation Software- Product name |