

Detailed Design

Appendix A: Test Plan

Subconscious Analysis Software

Requested by:

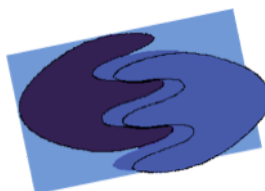
Dr. Eric Breimer

Associate Professor

Department of Computer Science

Siena College

Delivered by:



enigma elucidation

Prepared By:

Megan DeRudder

Christopher Black

Lindsay Kulzer

Amanda Kurz

Nathan Levine

Daniel West

March 1, 2012

Version 1.0



Table of Contents

Test Plan Identifier and Introduction	3
1.1 Introduction.....	3
1.2 Test Plan Identifier	3
Item Pass/Fail Criteria	4
2.1 Functional Requirements Inventory.....	4
2.1.1 Administrator	4
2.1.2 Participant.....	5
2.2 Non-Functional Requirements Inventory.....	5
2.3 Exception Handling	6
2.4 Testing Approach	7
2.5 Acceptance Test- Acceptance Criteria	8
2.6 Unit Test Directory	9
2.6.1 Unit Test Cases	9
2.6.2 Directory of Unit Test.....	10
2.6.3 Login.....	11
2.6.4 Querying the Database.....	11
2.6.5 Create IAT	13
2.6.6 Take IAT.....	14
2.6.7 Upload Stimuli.....	15



Test Plan Identifier and Introduction

1.1 Introduction

The Test Plan is used to test and confirm that project SAS meets all of the functional and nonfunctional requirements as set forth in our Requirements Specification document. The test plan consists of the following tests:

Test Case: Tests the project SAS against the individual actions taken by our client during runtime. These are pass or fail tests.

Unit Test: Focuses on testing the individual software components. These unit tests are usually performed by one person to make sure that each component is performing correctly and efficiently. These tests include logic based testing, random testing, syntax testing, etc.

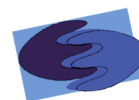
Integration Test: This test checks to make sure that all of the software components work together as a whole. Unit tests do not check for compatibility issues between components. This test not only tests compatibility, but it also checks for dependency issues, interface issues, and database, survey engine and interface interaction.

System Test: This test is concerned with making sure that Project SAS as a whole meets all of the functional requirements necessary to perform adequately as a system.

Acceptance Test: This test checks to make sure that Project SAS meets all of the functional and nonfunctional requirements as set forth by the corresponding Requirements Specification to the approval of our client.

1.2 Test Plan Identifier

The test plan will consist of a detailed checklist of how the software should perform. The details of the test plan will adjust as the functional and non functional requirements change throughout software development. The document will record all unit test and their results, pass or fail. The final version of our test plan will be provided with our Acceptance Test Documentation.



Item Pass/Fail Criteria

2.1 Functional Requirements Inventory

The functional requirements inventory is the part of the test plan that checks if the functional requirements of SAS are met. The functional requirements are components that can be tested and then classified as either met or unmet based on the data the unit tests provide. The functional requirements inventory will act as a check list to ensure that our client's, Dr. Breimer, requirements are met.

Below is a check list based on the functional requirements for the two users of SAS, the administrator and the participant.

2.1.1 Administrator

YES	NO	Will be able to securely login to SAS using the registered username and password
YES	NO	Will be able to retrieve password via email from "Forgot Password" link
YES	NO	Will be able to create IAT
YES	NO	Will be able to enter four unique categories
YES	NO	Will be able to choose stimuli objects, words or images, associated with each category
YES	NO	Will be able to delete stimuli objects before completing IAT
YES	NO	Will be able to create demographic survey
YES	NO	Will be able to choose whether or not a participant will be able to view their test results
YES	NO	Will be able to log out of SAS



2.1.2 Participant

YES	NO	Will be able to take an IAT
YES	NO	Will be able to fill out a demographic survey
YES	NO	Will be able to view directions on how to take an IAT
YES	NO	Will be able to view all of the categories and the stimuli objects correlated with them
YES	NO	Will be able to take the test by categorizing stimuli for 6 different blocks
YES	NO	Will be able to categorize stimuli by pressing the I or e keys on their keyboard
YES	NO	Will be able to press the space bar to move onto the next block
YES	NO	Will be able to view results if the administrator allows

2.2 Non-Functional Requirements Inventory

The following is a list of non-functional requirements of SAS. Below are requirements that specify how the system should be; that is, what qualities the system should have as opposed to what the system should do (functional requirements).

- The system must be aesthetically pleasing
- The system must be easy to use
- The system must be independent of any localized server
- The system must be platform independent



2.3 Exception Handling

The system must be able to handle errors caused by the environmental factors and actions made outside of the system or system's control. SAS will be built to handle these exceptions.

If our administrator, Dr. Breimer forgets his password, there must be a method to access his account. SAS will provide a "Forgot Password," link on the login screen. This link will send the administrator to a second screen which will ask Dr Breimer to provide his email address. If this email address matches up with the one preregistered within the system, the system will send an email to Dr Breimer's registered email account providing his password and a link back to the login screen of SAS.

While Dr. Breimer creates an IAT he must fill out forms for both, creating categories for the IAT and inputting stimuli to the IAT. An IAT cannot be created without inputting four unique categories. JavaScript will first be used to first check that there is data in each category field; the "Submit" button will not be active until this requirement is fulfilled. Second, each category must be unique. We will not accept the category fields unless all four are unique.

The second portion of the create IAT form is used to upload stimuli. The upload stimuli form will contain radio buttons to choose between, image or word. To upload a word, the user must input a word in the stimuli text field. JavaScript will be used to check if this field is left blank. If the text field is blank, the upload button will not be active. To upload an image, the user must select the radio button for image. JavaScript will be used to check which radio button is selected; the "Browse for Images" button will not be active if the image radio button is not selected. Our system will only allow certain file extensions and file size to be uploaded as stimuli objects. Another measure of handling will take place during image upload to check the selected file is valid.

SAS must also be able to handle if the system was to crash, due to loss of internet or power during the execution of an IAT. To ensure SAS does not leave the database with incomplete rows, the system will mark a participant's data as complete once the IAT is 100% completed. This information will be available for FSH Technologies to check before analyzing, exporting, or removing any IAT data for Dr. Breimer's research.



2.4 Testing Approach

Our primary goal in testing Project SAS is to make sure that our system will follow all of the functional and non-functional requirements as listed in this document and in the Requirements Specification. We will begin by extensively testing the individual low level functions and functionalities in the manner listed in our unit tests. These tests will check to make sure that the System responds appropriately for each individual action that the user makes. The tester will perform each test case for each unit test, marking each test as having either been successfully completed or failed. These tests will be repeated for each of the web browsers listed in our Requirements Specification in order to ensure that all of the listed browsers are adequately supported.

In the event a test fails, adjustments will be made to the system to correct the problem. After every adjustment, the unit test will be performed again to make sure that not only the issue has been corrected, but no other problems within the component have been created as a result of the changes.

Once all of the unit tests have been finished, an Integration test will be performed to make sure that all of the units as a whole are working properly, and that no one component is causing a problem for the other component. This test will be performed in multiple browsers to make sure that all of the necessary browsers are supported.

Once all the functional requirements have been met, the tester(s) will test the system for the non-functional requirements. This will be done by running the system through procedures similar to what will be performed by the prospective user, while making sure that the system performs in accord with the non-functional requirements.



2.5 Acceptance Test- Acceptance Criteria

A software test plan is essential to the design and development of a desired product. The test plan forces the developers to assess all functions of the product taking into account how they are to perform with both expected and unexpected input. Doing this helps to better the performance of the product later when development is complete and ready for use by its intended user(s). It addresses any problems that may have not been obvious during the planning or design phases.

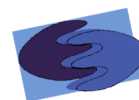
The test plan documents how each of the functional and non-functional requirements are to perform based on their objectives, scope, approach, and/or input. It also contains details for testing each of these functions and how the product should handle any input or condition, desired or undesired. The test plan should be explicit enough so that any user would be able to test the product and determine whether it meets the acceptance criteria or not.

The acceptance criteria are based on the functional and non-functional requirements of the product, which are listed in sections 2.1 and 2.2 of this document, respectively. The functional requirements describe what the system or product should be able to do and how, while the non-functional requirements describe how the system should be, for example user-friendly or aesthetically pleasing. Non-functional requirements cannot be tested and the acceptances of these requirements ultimately lie in the opinions of our client.

Upon completion of these tests, enigma elucidation will not only be able to determine if the system was implemented correctly but also have a better understanding of how the system is organized and what should be changed to make it more cohesive, if anything.

Our system, SAS, Subconscious Analytical Software, will be tested on both Windows and Mac operating systems and on major browsers, such as Internet Explorer, Google Chrome, Mozilla Firefox, and Safari. Testing conditions will be determined by enigma elucidation and will be organized in a hierarchy that will break down into more detail at each level.

The roots of the test plan are the unit tests. The unit tests will divide each of the functional requirements into categories, or units, which will contain more specific tests for each test case. Each of the cases will be tested separately at first then as a unit. Once all units are functioning as expected they will be tested together to ensure that they continue to perform correctly, this is called the integration test. The outcome of all these tests will be compiled into the Acceptance Test document. This will determine whether or not all of the requirements have been met.



2.6 Unit Test Directory

The following is a list of all units which will be tested. Once every individual unit test passes, a full systems test will be performed to check the overall correctness of the system.

List of Units

- Login
- Querying the Database
- Create IAT
- Take IAT
- Add Stimuli

2.6.1 Unit Test Cases

Each individual test case consists of an identifying test number and a description. Also included is the input to be entered by the user, the state before the test, and the expected result. After each test has been performed, they will be marked with pass/fail, and observations made of the test results will be recorded.


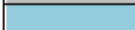


2.6.2 Directory of Unit Test

System Test - Test Results for All Unit Tests

Enigma Elucidation - FSH Tech
F.I.L.E.T./S.A.S.
Dr. Eric Breimer

Directory of Unit Tests

 = Enigma Elucidation
 = FSH Tech

Pass/Fail Status		Passed	Failed	Unit Number	Unit Test Name	Date Last Tested
F	0%	0	0	1	Login	01/00/00
F	0%	0	0	2	Querying the Database	01/00/00
F	0%	0	0	3	Create IAT	01/00/00
F	0%	0	0	4	Take IAT	01/00/00
F	0%	0	0	5	Add Stimuli	01/00/00
F	80%	4	1	6	View Homescreen	03/02/12
F	80%	4	1	7	View IAT Test Data	03/12/12
F	88%	7	1	8	View IAT Data	03/02/12

83.33% of Test Cases Passed (99.44% passes the Ivory Snow Test)



2.6.3 Login

enigma elucidation

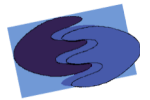
Login Unit Test

This unit test will explore all the variations of username and password combinations that may be inputted

Test Cases

Pass/ Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected Results	Observed Results
N/A	1.001	Null username field	Leave username field blank	Click the login button	Blank username field	Output "invalid username"	
N/A	1.002	Null password field	Leave password field blank	Click the login button	Blank password field	Output "please enter your password"	
N/A	1.003	Null UserName and Password	Leave both username and password field blank	Click the login button	Blank user name and password field	Output "invalid username"	
N/A	1.004	Incorrect password for identified username	Insert incorrect password for given username	Click the login button	Valid username, invalid password	Output "incorrect password"	
N/A	1.005	Non-existent username	Insert an invalid username	Click the login button	Invalid username	Output "invalid username"	
N/A	1.006	Forgot password	Click on provided link	Click the forgot password link	Blank form	Redirect to forgot password screen	
N/A			Click yes when user is asked if username and password should be emailed to the registered email	Log on to registered email and look for message		An email should be received at the registered address indicating username and password	
N/A	1.007	Forgot password Screen	Insert valid username and corresponding password	Click on the login button	Forgot Password Screen	User brought to their homepage	
N/A	1.008	Login			Filled in form		
F = Unit Summary			0%	passing	0	passed	
				6	0	failed	

[Directory Page](#)



2.6.4 Querying the Database

enigma elucidation										
Querying the Database Unit Test										
This unit test will explore all the variations of queries to the database.										
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected Results	Observed Results	Comments	Tested By	Test Date
N/A	2.001	Check if table exist	Enter table name into name field	Runs a query using that table name	Table exist	Table appeared with their data				
N/A	2.002	Insert data into table	Input data into corresponding fields	Runs a query updating the table to contain the new data	Table exist	The table's data will be updated				
N/A	2.003	Delete data from table	Enter table name and data to be deleted	Runs a query deleting the data from the specified table	Table and data exist	The specified data is deleted				
N/A	2.004	Query database	Enter table name and data to be displayed	Runs a query returning desired data from specific table	Table and data exist. Data generated by hand, not by JAT	The specified data is displayed				
N/A	2.005	Check for non-existent table	Enter table name	Runs a query looking for specific table	Table does not exist	Output "Table Does Not Exist"				
N/A	2.006	Insert invalid data into table	Enter table name and data into corresponding field	Run a query inserting new data in table	Table exist, but input data is not the right type	Output "Invalid Data"				
N/A	2.007	Delete nonexistent data from database	Enter table name and data to be deleted	Runs a query deleting the data from the specified table	Table exist but data does not	Output "Data does not exist in table"				
N/A	2.008	Query database for nonexistent data	Enter table name and desired data	Runs a query returning desired data from specific table	Table exist but data does not	Output "Nonexistent data"				
N/A	2.009	Check that tables properly connect to each other	Enter table names and data to be displayed	Runs a query that joins all of the tables together	Some data should be in each table that is not listed in one of the other tables. The rests should be referenced in all the tables.					
F	9	Unit Summary tests	0%	passing	0	passed	0	failed	Date of last test =	1/0/00
Directory Page										



2.6.5 Create IAT

Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected Results	Observed Results	Comments	Tested By	Test Date
N/A	3.001	Test link to Create Survey	Click on Create Survey Button	Click on Create Survey Button	The Create Survey Page All four category fields full and unique	Lime Survey Login Page "Add Stimuli" option created				
N/A	3.002	Choose Categories	Input four unique categories	Click on Update Button	Repeated words in category fields Blank category field	Output "Please input four unique categories!" Output "Please input four unique categories!"				
N/A	3.003	Input Invalid Categories	Input repeat categories	Click on Update Button						
N/A	3.004	Blank category field Repeat for 3.004 for all categories	Leave 1(for more) categories blank	Click on the Update Button						
N/A	3.005	Test delete link	Click "delete" link	Click "Delete" Link	Stimuli Table with delete links	Stimuli deleted from table				
N/A	3.007	Test Finish Link Test "Go Back" Link	Click "finish" button Click "Go Back" button	Click "finish" button Click "Go Back" button	Full Stimuli Tables Confirmation Page	Confirmation Page Create IAT Page				
N/A	3.009	Test "Yes, finish the test!" Link	Click "delete" button	Click "delete" button	Confirmation Page	New IAT				
F	= Unit Summary 9	tests	0%	passing	0	passed		Date of last test =		1/0/00



2.6.6 Take IAT

Pass/ Fail Status	Test Number	Description	Action to perform test (Input)	Steps to be Executed	State Before Test	Expected Results	Observed Results	Comments	Tested By	Test Date
N/A	4.001	Test Link to IAT from given URL	Click on URL or enter URL into browser	Go to URL	Email or Open Browser	Instructions pop up upon entering the IAT				
N/A	4.002	Survey Link	Test survey link from directions page	Press Link to Survey	Directions page	Survey Page				
N/A	4.003	Incomplete Survey	Do not complete survey	Leave mandatory fields of survey blank	Survey	Error message "Please complete survey before completing IAT"				
N/A	4.004	Complete Survey	Complete survey	Finish survey	Survey	Directions page of IAT				
N/A	4.005	Directions Page	Test continue link from directions page	Press Spacebar	Directions page	First "Block" of the IAT				
N/A	4.006	Correctly sort stimuli	Sort the stimuli correctly using E and I keys	Press E (to sort left) and I (to sort right)	Stimuli Object	Next stimuli Object appears				
N/A	4.007	Incorrectly sort stimuli	Sort stimuli to wrong category	Press E (to sort left) and I (to sort right)	Stimuli Object	A red "X" appears over stimuli object				
N/A	4.008	Input invalid key	Press any other key then E or I	Press any other key then E or I	Stimuli Object	Error message "Please use E and I keys for input"				
N/A	4.009	Finish "block"	Moving to next IAT "block"	Press space bar	Stimuli Object	First Stimuli object of next block of IAT				
N/A	4.01	Finish IAT	Complete all blocks of IAT	Complete all blocks of IAT	Last page of IAT	Message "Thank you for taking this IAT"				
N/A	4.011	Sudden End of Session	Exit IAT in the middle of the test	Close IAT	Take IAT	Removes Data Stored from IAT				
N/A	4.012	Check data storage	Run test 2.004 again after taking an IAT to make sure the data was properly stored	This test should be taken sometime after test 2.004 is 100% passed. Query the database to see if the new data pops up, and the correct values are in place	The data to be generated shouldn't already be in the table.	The correct data should be in the Table.				
N/A	F	= Unit Summary	0%	passing	0	0	passed			1/10/00
	12	tests								



2.6.7 Upload Stimuli

Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected Results	Observed Results	Comments	Tested By
N/A	5.001	Radio Button Stimuli Type	Toggle both Stimuli type values for Radio Button	Click one button, then click the other	Radio Button set to default (word), nothing inputted	The word button activates empty text box and deactivates image browse button. The image button activates browse button and deactivates the text box.			
N/A	5.002	Uploading stimuli image	Input an appropriate root and extension to image	Browse for image, Click Add Stimuli	Stimuli field with valid input	Stimuli added to stimuli table			
N/A	5.003	Uploading stimuli text	Input text	Type in word into stimuli field, Click Add Stimuli	Stimuli field with valid input	Stimuli added to stimuli table			
N/A	5.004	Input invalid link to image	Input an invalid root or extension to image	Browse for image, Click Add Stimuli	Stimuli field with invalid input	Output "Can not upload image, try again!"			
N/A	5.005	Do not select a category type	Do not select a radio button	Click Add Stimuli	No Selected Radio Button	Output "Please select a category"			
N/A	5.006	Add Stimuli	Input valid stimuli and chose 1 category	Click Add Stimuli	Stimuli field with valid input and category selected	Stimuli added to stimuli table			
N/A	5.007	Check Image Size	Select Image that is too big	Click Add Stimuli	Stimuli is not added	Output "Size too big!"			
N/A	5.008	Check Image Type	Select Image that is not in correct format	Click Add Stimuli	Stimuli is not added	Output "Image is not in correct format!"			
F	Unit Summary		0%	passing	0	passed		Date of last test =	1/0/00
	8	tests			0	failed			