Acceptance Test

Subconscious Analysis Software (SAS)
Appendix A: Test Plan

Requested by:

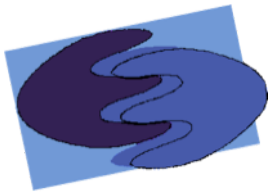
Dr. Eric Breimer

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

April 26, 2012

Version 1.0

Table of Contents

Introduction and Identifier

1.1 Introduction

In order for enigma elucidation to test the functional and non-functional requirements of our system, SAS, we have divided our system into logical units which represent each of the major functions that are required for SAS. The functional requirements these units test can be found in section 2.2 of the Acceptance Test document. Each unit is further divided into test cases. These are individual actions that when used together complete a unit. We will first test all of the test cases separately to make sure they have been implemented correctly and functioning as expected. After testing each case we will be able to test them as a unit, called unit tests. We will test the unit cases that other unit cases depend on first. This way, if an error is discovered in these first to be tested cases, we won't have to restart testing from the beginning upon fixing the error. In the event that we are testing one unit that is both depended on and depends on other units, we will retest the associated units after any errors are found and corrected in that particular unit. This process is known as integration testing. This process of performing tests that take into account how the performance is of units change due to changes in dependencies is called integration testing. Finally we will test will test our system as a whole to ensure that all functional requirements have been met.

If there are major changes to the implementation of any function during the testing process a regression test will be performed to ensure that all of the pieces of SAS are still functioning correctly. This regression test will consist of an integration test performed on the unit that was changed.

Once all of the functional requirements are met, we will test the non-functional requirements to the best of our ability. We will perform the "Stupid Roommate Test" by showing our product to our peers, and making sure that they find SAS aesthetically pleasing, can figure out how it works, and cannot break the code. We will also test the product on multiple platforms to make sure there are no compatibility issues. SAS is being developed run independent of any localized server. To test this, we will run our program from multiple servers.

Finally, after all testing is performed and it is confirmed that SAS is functioning as expected and to the standards of enigma elucidation we will present our system to our client, Dr. Eric Breimer, who will perform the final test, the acceptance test. The acceptance test will involve both the functional and non-functional requirements. Dr. Breimer will either accept or reject our implementation based on whether or not his functional or nonfunctional requirements are met.

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 checklist to ensure that the requirements of our client, Dr. Breimer, are met. Below is a checklist 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 log into SAS via FILET
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 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 spacebar to move onto the next block

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 system crashes, due to loss of internet or power during the creation or execution of an IAT. To ensure SAS does not leave the database with incomplete rows, all data collected during IAT creation or execution will only be stored in the database at the conclusion of creation or execution. 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 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 access 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.

2.5 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

2.5.1 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.

2.5.2 Directory of Unit Test

Pass/Fail Status		Unit Number	Unit Test Name	Date Last Tested	Comments or brief description	Integrated with these units	Number of Tests Passed (Passed\Failed)
				4/25/2012			
P	100%	1	Login	4/25/2012	Mostly FSH's side	2	7\0
F	71%	2	Querying the database	4/25/2012	Functional except for delete		5\2
F	89%	3	Create IAT	4/25/2012	Functional except for delete	2	8\1
P	100%	4	Take IAT	4/25/2012		2	11\0
F	75%	5	Add Stimuli	4/25/2012	Functional except for delete	2	6\2
P	100%	6	View Home Screen	4/25/2012	FSH's side		6\0
F	89%	7	View IAT Data	4/25/2012	FSH's side	2	8\1
P	100%	8	View IAT Test Data	4/25/2012	FSH's side	2	6\0

2.5.3 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 Exectuted	State Before Test	Expected Results	Observed Results	Comments	Tested By	Test Date
P	1.001	Force user to log in to FILET before linking into SAS	Link to enigma without logging on	Link to enigma without logging on	Not logged on	You are denied access and given a link to FSH's login			Nathan Levine	4/25/2012
P			100% Passing			1 passed		Date of last test =	4/25/201	
	1 test					0 failed				

2.5.4 Querying the Database

enigma elucidation

Querying the Database Unit Test

This unit test will explore all the variations of queries to the database.

Test Cases

Pass/ Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Exectuted	State Before Test	Expected Results	Observed Results	Comments	Tested By	Test Date
P	2.001	Checks Database Connection	Query database	Execute a query from the PHP code	Database exists	Connects to Database with no errors	Successfully Connected		Nathan Levine	4/25/2012
P	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			Nathan Levine	4/25/2012
F	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			Nathan Levine	4/25/2012
P	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 IAT	The specified data is displayed			Nathan Levine	4/25/2012
P	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"			Nathan Levine	4/25/2012
F	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"		Allows you to insert any kind of file, including shortcuts and word docs	Nathan Levine	4/25/2012
P	2.007	Check that tables properly connect to eachother	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.				Nathan Levine	4/25/2012

F := **Unit Summary** — **5 passed**

7 test — **71% passing** — **2 failed** — **Date of last test = 4/25/2012**

2.5.5 Create IAT

enigma elucidation

Create IAT Unit Test

This unit test will explore all the variations of input and link usage to create an IAT

	Test Cases								
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**
P	3.001	Test link to Create Survey	Click on Create Survey Button	Click on Create Survey Button	The Create Survey Page	Lime Survey Login Page			Nathan Levine	4/25/2012
P	3.002	Choose Categories	Input four unique categories	Click on Update Button	All four category fields full and unique	"Add Stimuli" option created			Nathan Levine	4/25/2012
P	3.003	Input Invalid Categories	Input repeat categories	Click on Update Button	Repeated words in category fields	Output "Please input four unique categories!"			Nathan Levine	4/25/2012
P	3.004	Blank category field	Leave 1(or more) categories blank	Click on the Update Button	Blank category field	Output "Please input four unique categories!"			Nathan Levine	4/25/2012
P	3.005	Repeat for 3.004 for all categories							Nathan Levine	4/25/2012
F	3.006	Test delete link	Click "delete" link	Click "Delete" Link	Stimuli Table with delete links	Stimuli deleted from table	Doesn't delete the right data		Nathan Levine	4/25/2012
P	3.007	Test Finish Link	Click "finish" button	Click "finish" button	Full Stimuli Tables	Confirmation Page			Nathan Levine	4/25/2012
P	3.008	Test "Go Back!" Link	Click "Go Back!" button	Click "Go Back!" button	Confirmation Page	Create IAT Page			Nathan Levine	4/25/2012
P	3.009	Test "Yes, finish the test!" Link	Click "Finish" button	Click "finish" button	Confirmation Page	New IAT			Nathan Levine	4/25/2012
F	:= Unit Summary		89% Passing			8 passed		Date of last test =	4/25/2012	
	9 test					1 failed				

2.5.6 Take IAT

enigma elucidation
Take IAT Unit Test
This unit test will explore all the input variations and links while taking an IAT

Test Cases

Pass/ Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Exectuted	State Before Test	Expected Results	Observed Results	Comments	Tested By	Test Date
P	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			Nathan Levine	LimeSurvey takes care of this
P	4.002	Survey Link	Test survey link from directions page	Press Link to Survey	Directions page	Survey Page			Nathan Levine	LimeSurvey takes care of this
P	4.003	Incomplete Survey	Do not complete survey	Leave mandatory fields of survey blank	Survey	Error message "Please complete survey before completing IAT"		LimeSurvey takes care of this	Nathan Levine	LimeSurvey takes care of this
P	4.004	Complete Survey	Complete survey	Finish survey	Survey	Directions page of IAT		LimeSurvey takes care of this	Nathan Levine	LimeSurvey takes care of this
P	4.005	Directions Page	Test continue link from directions page	Press Spacebar	Directions page	First "Block" of the IAT			Nathan Levine	LimeSurvey takes care of this
P	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			Nathan Levine	LimeSurvey takes care of this
P	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			Nathan Levine	LimeSurvey takes care of this
P	4.008	Input invalid key	Press any other key then E or I	Press any other key then E or I	Stimuli Object	Ignores Input			Nathan Levine	LimeSurvey takes care of this
P	4.009	Finish "block"	Moving to next IAT "block"	Press space bar	Message "You have completed this block, the next section will have new categories Press space bar to continue"	First Stimuli object of next block of IAT			Nathan Levine	LimeSurvey takes care of this
P	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"			Nathan Levine	LimeSurvey takes care of this
P	4.011	Sudden End of Session	Exit IAT in the middle of the test	Close IAT	Take IAT	Doesn't store data in database			Nathan Levine	LimeSurvey takes care of this
P	:= Unit Summary		100% Passing			11 passed		Date of last test =	4/25/2012	
	11 test					0 failed				

2.5.7 Upload Stimuli

Pass/ Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Exectuted	State Before Test	Expected Results	Observed Results	Comments	Tested By	Date Tested
P	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.	Matches the Expected Results Exactly		Nathan Levine	4/25/2012
P	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	Stimuli is added to the table, you can see it in the listing		Nathan Levine	4/25/2012
P	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	Stimuli is added to the table, you can see it in the listing		Nathan Levine	4/25/2012
F	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!"	You can upload word docs as pictures		Nathan Levine	4/25/2012
P	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"	Proper message prints to screen		Nathan Levine	4/25/2012
P	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	Stimuli is added to the table, you can see it in the listing.		Nathan Levine	4/25/2012
P	5.007	Check Image Size	Select Image that is too big	Click Add Stimuli	Stimuli is not added	Scales down the image to an acceptable size	Scales down the image to an acceptable size	The original plan was to print a warning, and then we changed the functionality.	Nathan Levine	4/25/2012
F	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!"	Allows you to add anything as an image		Nathan Levine	4/25/2012
P	:= Unit Summary		75% Passing			6 passed		Date of last test =	4/25/2012	
	8 test					2 failed				