

# Acceptance Test

Factual Inter Leanings that Examine Throughput  
(F.I.L.E.T.)

**Requested By:**

Dr. Eric Breimer  
Associate Professor of Computer Science  
Siena College

**Delivered By:**

FSH Tech

**Prepared By:**

Jacquelyn Boylan  
Matthew Brancato  
Matthew Kemmer  
Serena Moore  
Nydia Negrón  
Mike Tanski

April 25, 2012

# F.I.L.E.T.

## Acceptance Test

### Table of Contents

## 1 | Introduction

1.1   Product Overview.....	4
-----------------------------	---

## 2 | Requirements Inventory

2.1   Use Case Narratives.....	5
2.1.1   Administrator.....	6
2.1.2   Participant.....	6
2.2   Functional Requirements Inventory.....	7
2.2.1   Administrator.....	7
2.2.2   Participant.....	7
2.3   Non-Functional Requirements Inventory.....	8
2.4   Future Enhancements.....	9

## 3 | External Design Specifications

3.1   User Displays.....	9
3.1.1   Login.....	9
3.1.2   Home Screen.....	10
3.1.3   View an IAT's Data .....	11
3.1.4   View Survey Data.....	12
3.1.5   View Participants Test Data .....	13
3.1.6   View Participants Survey Data.....	14
3.2   CSV Exports.....	15
3.2.1   Export Basic Participant List.....	15
3.2.2   Export All Survey Data.....	16
3.2.3   Export All Test Data.....	17
3.2.4   Export Detailed Participant List.....	18
3.3   Logical Data Dictionary.....	19
3.4   Entity Relationship Diagram.....	21

# F.I.L.E.T.

## Acceptance Test

### Table of Contents

#### Continued

## 4 | Architectural Design Specifications

4.1   Development, Operation, and Maintenance Environment.....	22
4.2   Deliverables.....	23

## Appendices

Appendix A: Test Plan and Results.....	Separate Document
Appendix B: Data Flow Diagrams.....	Separate Document
Appendix C: Source Code.....	Separate Document
Appendix D: Glossary of Terms.....	25
Appendix E: Timelines.....	26

# 1 | Introduction

## 1.1 | Product Overview

Dr. Breimer wants a web-based piece of software that will allow him to create Implicit Association Tests, or IATs. IATs are psychological tests that measure biases that test takers may have towards one group or another. The test is taken by first taking a demographic survey, then by associating stimuli objects (whether text or images) into categories and based on how much time is taken on each stimuli, a bias may be determined. Dr. Breimer wants to simply create a survey then choose these categories and stimuli and have the system create the IAT around them. Once his IATs are created and put online by the IAT system he will be able to distribute the URL to participants whom he wishes to take the test. At any point after the IAT is online he may view and export both the survey and test data in order to analyze it. F.I.L.E.T. will allow our client, Dr. Breimer, to view and analyze the results of IATs that have been previously deployed by enigma elucidation's software, S.A.S.

## 1.2 | Testing

Our project was tested in detail to make sure that all requirements listed were completed. Please see Appendix A (Test Plan and Results) for further information.

## **2 | Requirements**

### **2.1 | User Case Narratives**

In order for our client, Dr. Breimer, to gather information for research, he would like to use IATs (Implicit Association Tests). IATs are psychological tests used to determine biases or prejudices by placing stimuli objects into one of four categories. With every IAT Dr. Breimer creates he also needs information to be gathered about the person taking it, meaning a survey would also have to be created for every test.

#### **2.1.1 | Administrator**

When the Administrator logs in to the system, the Administrator will be able to create new IATs as well as view the results of pre-existing IATs. The Administrator will be able to view an organized collection of the resulting data which will be stored in the database. The database will contain all of the Participants' survey data as well as the results of every IAT that has been created and published, which the Administrator can explore individually. The Administrator may export the results of both the survey and IAT to a spreadsheet for further analysis.

## 2.1.2 | Participants

Anybody who is provided the Administrators link to one of the published IATs may choose to participate. The Participant will first be taken to the survey in their web browser before proceeding to the IAT. The survey will contain information pertinent to the IAT and this information will be important when viewing the end results. Once all the mandatory questions have been answered on the survey, the IAT page will be loaded. The Participant will be shown an introductory screen explaining the test, and introducing the four categories and stimuli objects that will follow. The Participant will need to press the E and I keys on their keyboard to categorize stimuli objects left or right to the correct categories. Once the Participant presses the space bar, the test will begin with the first of seven blocks, or sections of the test. Each block is completed by the same method of categorization; however the categories will be paired differently in each block. A screen similar to the introductory screen will be shown between each block, and the Participant will start the next block by pressing the space bar once again. The Participant will also be able to view the instruction screen during the test if need be. Once all seven blocks are completed, the Participant will be shown their results, if they are available for the test they have taken. The part of the system that the Participants will interact with will be handled by enigma elucidation's system, S.A.S.

## 2.2 | Functional Requirements Inventory

F.I.L.E.T. will be tested and used on Internet browsers such as Internet Explorer, Mozilla Firefox, Safari, and Google Chrome. F.I.L.E.T. will only be dealing with one user, the Administrator. The functional requirements that F.I.L.E.T. will be dealing with are listed below

**Bolded sections are the units included in F.I.L.E.T, un-bolded will be included in S.A.S.**

## 2.2.1 | Administrator

- **Will be able to authenticate**
  - **Will be able to enter username and password**
- Will be able to create an IAT
  - Will be able to create survey questions
  - Will be able to insert category titles
    - Will be able to pair the category titles
    - Will be able to distinguish which category we are primarily testing
  - Will be able to put image or text stimuli into categories.
  - Will be able to select whether results are shown or not
- **Will be able to view IAT results**
  - **Will be able to view survey results**
  - **Will be able to view test data**
- **Will be able to export IAT results to a spreadsheet**
  - **Will be able to export only survey data**
  - **Will be able to export all data**
  - **Will be able to export only statistical association results**

## 2.2.2 | Participant

- Will be able to follow a link to begin taking an IAT
- Will be able to take the IAT
  - Will be able to take a survey
  - Will have to categorize stimuli objects
- Will be able to see the results
  - This is an option that the Administrator can allow

## 2.3 | Non-Functional Requirements Inventory

The non-functional requirement inventory is a list of non-functional system requirements. This list is composed of requirements that specify what F.I.L.E.T. is expected to be able to do. Anything in this list can be changed as more information is gathered about the project.

- The system will be maintainable.
- The system will be stable.
- The system will be operational on several different browsers.
- The system will be easy to use.
- The system will be visually appealing.

## 2.4 | Future Enhancements

While IAT's are an excellent way to determine whether a Participant has an inherent association towards a particular group, it is possible for a Participant to "fool" the IAT system into thinking that they have a certain bias, when in fact they do not. This is sometimes the case when people have taken IAT's before, and know the way they work. One way we can stop these malicious Participants from corrupting our client's data is to add a program that can tell if the user is providing untrue data. While we do have some basic measures in our system now to take out responses that have taken too much time or too little time, in the future our software can learn of new ways the data can be corrupted, in order to ensure the most accurate results.

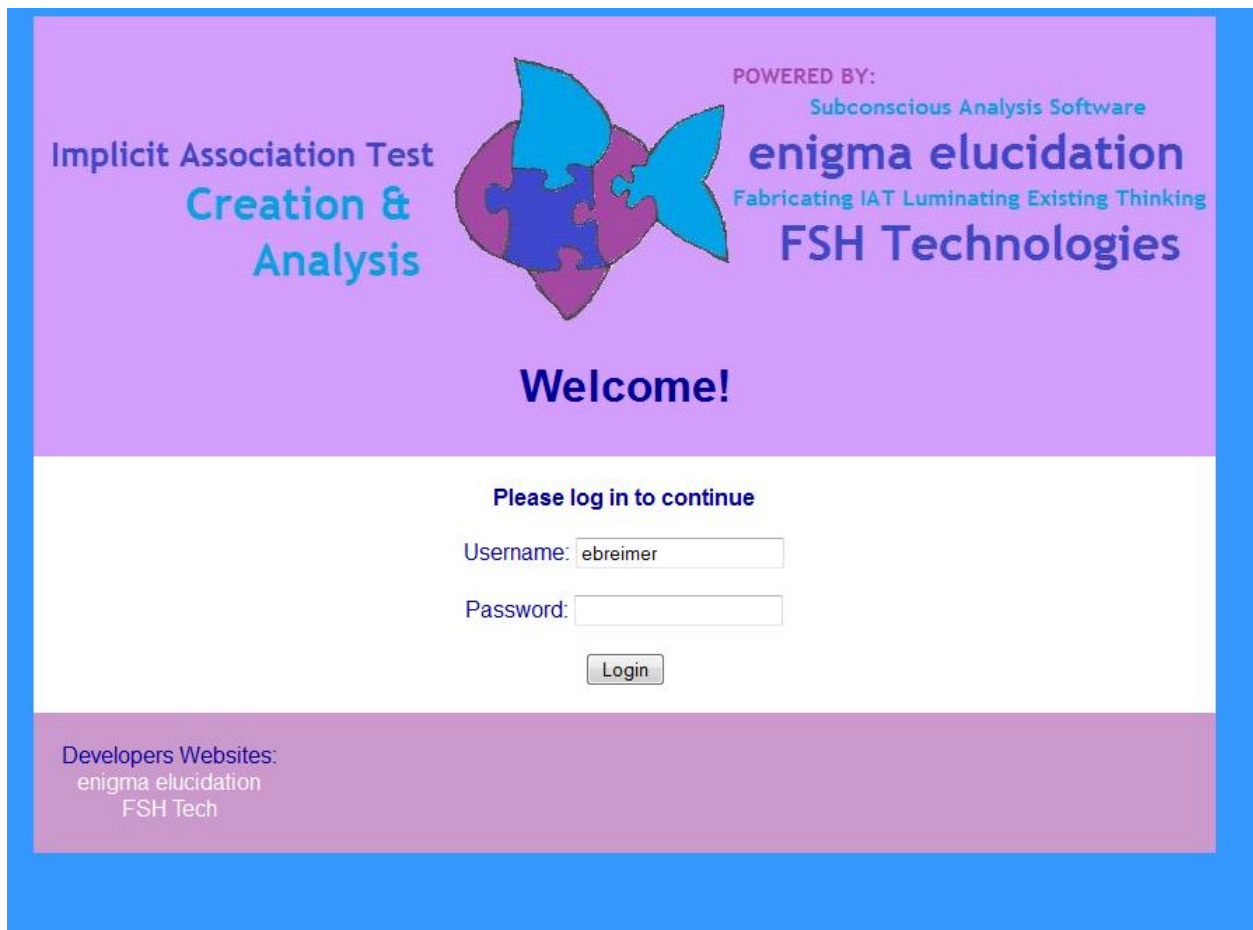


# 3 | External Design Specifications

## 3.1 | User Displays

### 3.1.1 | Login

When the Administrator first accesses the system, he will be prompted to log in.



### 3.1.2 | Home Screen

Once the Administrator is logged in, he will be redirected to the Home screen, which lists the different IAT's he has created, with the option to view the data of each of them. This will direct him to the View IAT Data screen. The administrator will also be able to click the View Survey Data button to view the survey data for any particular IAT. The Administrator will also have the option to create a New IAT on the Home page with the "Create New IAT" button.

Implicit Association Test  
Creation &  
Analysis

POWERED BY:  
Subconscious Analysis Software  
**enigma elucidation**  
Fabricating IAT Luminating Existing Thinking  
FSH Technologies

**Welcome!**

Create New IAT

**Deployed IATs**

Average Score	IAT Title		
-0.05265	Women vs. Politics	View Test Data	View Survey Data

Developers Websites:  
enigma elucidation  
FSH Tech

Logout

### 3.1.3 | View an IAT's Data

After Participants have taken the IAT, the Administrator will be able to view the results of the participants. There will be a list of the different Participant's ID's, their association score, with an option to view the test data and the survey data for the Participant in that particular row. There will also be buttons at the top of the screen to export the Participant's ID and association score to a spreadsheet. The Administrator will also be able to export an individual participants' survey data, or export all of results from the particular IAT.

The screenshot displays the 'Implicit Association Test Creation & Analysis' interface. At the top, it features a logo with puzzle pieces and the text 'POWERED BY: Subconscious Analysis Software enigma elucidation Fabricating IAT Luminating Existing Thinking FSH Technologies'. Below the header are two buttons: 'Create New IAT' and 'Return Home'. The main heading reads 'Currently Viewing: Women vs. Politics'. Below this are three export buttons: 'Export This List', 'Export All Survey Data', and 'Export All Test Data'. A table lists five participants with their IDs and association scores, each with buttons to view their test and survey data.

Participant ID	Association Score		
1	-0.00585	View Participants Test Data	View Participants Survey Data
2	-0.00585	View Participants Test Data	View Participants Survey Data
3	-0.00585	View Participants Test Data	View Participants Survey Data
4	-0.00585	View Participants Test Data	View Participants Survey Data
5	-0.00585	View Participants Test Data	View Participants Survey Data

### 3.1.4 | View Survey Data

The Administrator will have the option to view all responses to a particular IAT from the home screen. The survey will be viewed on LimeSurvey.

The screenshot displays the LimeSurvey interface for viewing survey data. At the top, it says "Browse responses: (Cats vs. Dogs)". Below this is a "Data view control" section with navigation buttons and filters. The main area is a table with the following columns: "Completed", "Response ID", "Start language", "What is your age?", and "How many pet". The table contains six rows of data. The "Actions" column for each row includes icons for search, edit, and delete. The footer of the interface shows "LimeSurvey Version 1.91 Build 10060".

Completed	Response ID	Start language	What is your age?	How many pet
Y	1	en	A3	
Y	2	en	A1	
Y	3	en	A3	
Y	4	en	A2	
Y	5	en	A4	
Y	6	en	A2	

### 3.1.5 | View Participants Test Data

If desired, the Administrator can choose to view how a specific Participant responded to the questions presented to them for their IAT. The Administrator will be able to see the question number, the stimuli they were presented with, their response time, and whether or not they responded correctly. This format is given for all 180 questions of an IAT. The Administrator can also view the Participant's ID number, as well as their calculated bias score. The Administrator will also be able to export the results of the Participant they're observing to a CSV file.

The screenshot displays the 'View Participants Test Data' interface. At the top, there is a purple header with the text 'Implicit Association Test Creation & Analysis' on the left, a logo of a butterfly made of puzzle pieces in the center, and 'POWERED BY: Subconscious Analysis Software enigma elucidation' on the right. Below the header, there are two buttons: 'View All Participants' and 'Return Home'. The main content area shows 'You are currently viewing IAT Women vs. Politics'. Below this, there are two boxes: 'Participant: 3' and 'Association: -0.00585'. A button labeled 'Export This Data' is positioned below these boxes. At the bottom, there is a table with four columns: 'Question Number', 'Stimuli', 'Reaction Time (milliseconds)', and 'Correct'. The table contains three rows of data.

Question Number	Stimuli	Reaction Time (milliseconds)	Correct
1	Red	4133	YES
2	x.png	2059	NO
3	Orange	1851	NO

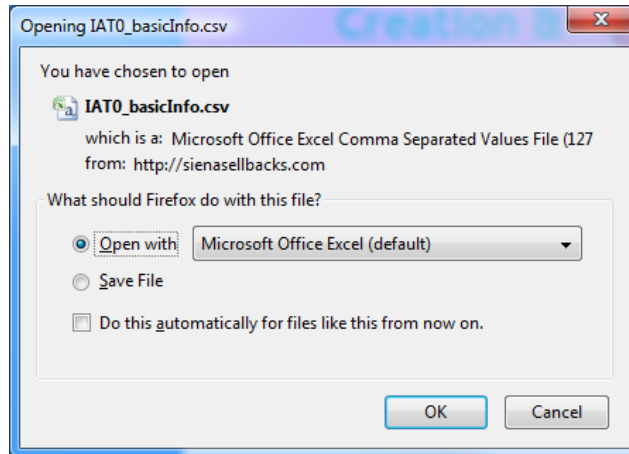
### 3.1.6 | View Participants Survey Data

If desired, the Administrator can view the specific answers a Participant gave to the Survey that goes along with an IAT. Since F.I.L.E.T. and S.A.S. are hosting the IAT surveys on LimeSurvey, which is external to our system, the answers to the survey by the specific Participant will be viewed on LimeSurvey.



### 3.2.1 | Export Basic Participant List

This exports the list of the participants who took a particular IAT and their association scores.

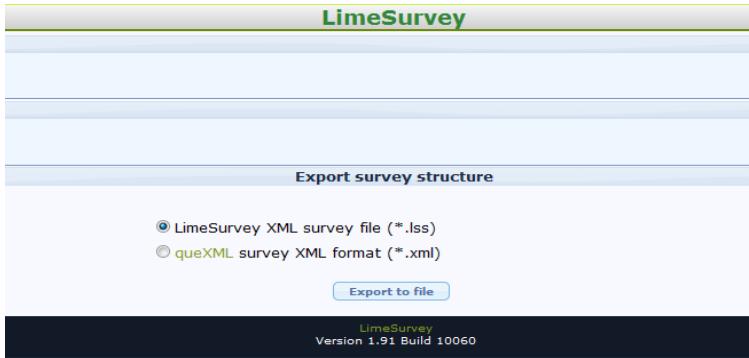


The Excel window shows the following data:

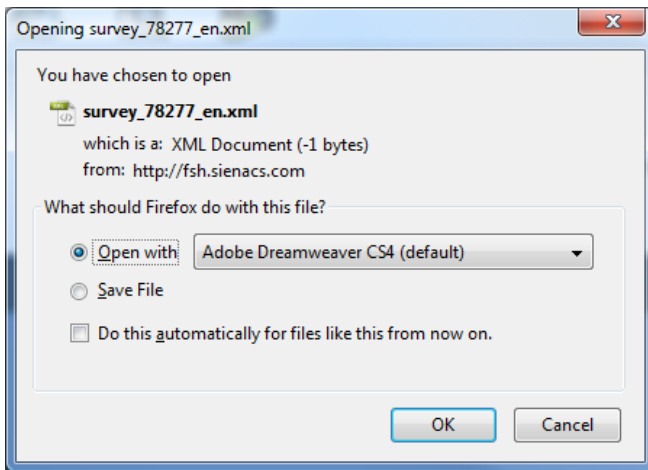
PID	Association
1	-0.00585
2	-0.00585
3	-0.00585
4	-0.00585
5	-0.00585
6	-0.00585
7	-0.00585
8	-0.00585
9	-0.00585
10	-0.00585

### 3.2.2 | Export All Survey Data

This function goes directly to LimeSurvey and takes advantage of its function to export the survey. This button will export the survey for the IAT the Administrator is currently viewing.



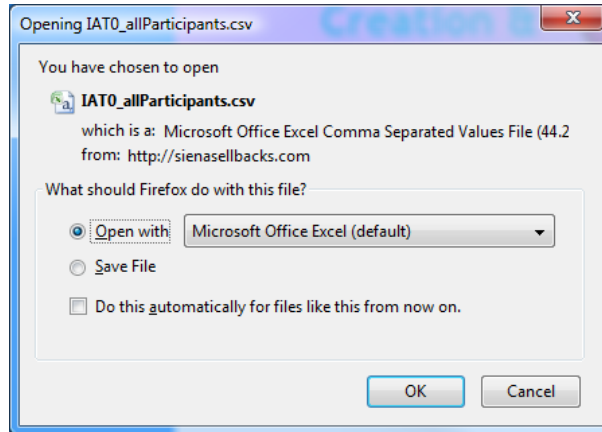
```
<?xml version="1.0" encoding="UTF-8"?>
<questionnaire id="78277">
  <title>Women vs. Politics</title>
  <investigator/>
  <dataCollector/>
  <section id="3">
    <sectionInfo>
      <position>title</position>
      <text>General</text>
      <administration>self</administration>
    </sectionInfo>
    <question>
      <text>What is your gender?</text>
      <response varName="1">
        <fixed>
          <category>
            <label>Male</label>
            <value>A1</value>
          </category>
          <category>
            <label>Female</label>
            <value>A2</value>
          </category>
        </fixed>
      </response>
    </question>
    <question>
      <text>What is your age?</text>
      <response varName="2">
        <fixed>
          <category>
            <label>&lt; 18</label>
            <value>A1</value>
          </category>
          <category>
            <label>18-25</label>
            <value>A2</value>
          </category>
          <category>
            <label>26-40</label>
            <value>A3</value>
          </category>
          <category>
            <label>41-60</label>
            <value>A4</value>
          </category>
        </fixed>
      </response>
    </question>
  </section>
</questionnaire>
```





### 3.2.3 | Export All Test Data

When “Export All Test Data” is clicked when viewing all the results for a particular IAT, all of the responses from all Participants for the particular IAT are exported to a CSV file that can be downloaded by the Administrator.

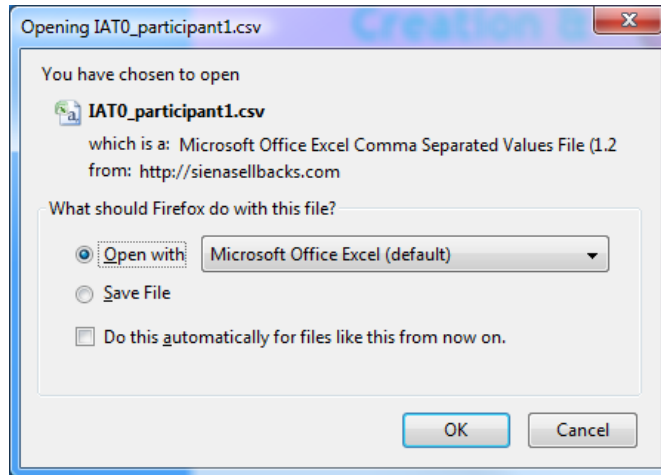


The image shows a screenshot of Microsoft Excel with the file "IAT0\_allParticipants-1 [Read-Only] - Microsoft Excel" open. The spreadsheet contains the following data:

PID	Question	StimuliID	Time(ms)	Correct	Association	
1	1	1	4133	YES	-0.00585	
2	1	2	2059	NO	-0.00585	
3	1	3	1851	NO	-0.00585	
4	1	4	6132	YES	-0.00585	
5	1	5	2763	YES	-0.00585	
6	1	6	6486	YES	-0.00585	
7	1	7	5996	YES	-0.00585	
8	1	8	6250	YES	-0.00585	
9	1	9	5191	YES	-0.00585	
10	1	10	1	5703	YES	-0.00585
11	1	11	2	2382	YES	-0.00585
12	1	12	3	2482	YES	-0.00585
13	1	13	4	2994	YES	-0.00585
14	1	14	5	4819	YES	-0.00585
15	1	15	6	5752	YES	-0.00585
16	1	16	7	4111	NO	-0.00585
17	1	17	8	1202	NO	-0.00585
18	1	18	9	5662	YES	-0.00585
19	1	19	1	2310	YES	-0.00585
20	1	20	2	6921	YES	-0.00585
21	1	21	3	4141	NO	-0.00585
22	1	22	4	3050	YES	-0.00585
23	1	23	5	3738	NO	-0.00585
24	1	24	6	4679	NO	-0.00585

### 3.2.4 | Export Detailed Participant list

When “Export This Data” is clicked when viewing a particular Participant’s results, all of the responses from the particular Participants are exported to a CSV file that can be downloaded by the Administrator.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Question	Stimuli	Time (ms)	Correct	Association									
2	1	1	4133	YES	-0.00585									
3	2	2	2059	NO	-0.00585									
4	3	3	1851	NO	-0.00585									
5	4	4	6132	YES	-0.00585									
6	5	5	2763	YES	-0.00585									
7	6	6	6486	YES	-0.00585									
8	7	7	5996	YES	-0.00585									
9	8	8	6250	YES	-0.00585									
10	9	9	5191	YES	-0.00585									
11	10	1	5703	YES	-0.00585									
12	11	2	2382	YES	-0.00585									
13	12	3	2482	YES	-0.00585									
14	13	4	2994	YES	-0.00585									
15	14	5	4819	YES	-0.00585									
16	15	6	5752	YES	-0.00585									
17	16	7	4111	NO	-0.00585									
18	17	8	1202	NO	-0.00585									
19	18	9	5662	YES	-0.00585									
20	19	1	2310	YES	-0.00585									
21	20	2	6921	YES	-0.00585									
22	21	3	4141	NO	-0.00585									
23	22	4	3050	YES	-0.00585									
24	23	5	3738	NO	-0.00585									
25	24	6	4679	NO	-0.00585									

### 3.3 | Logical Data Dictionary

Data Dictionary: a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format." - IBM Dictionary of Computing.

Table: The name(s) of the database table that the data variable name will be stored in

Data Name: The name of the data entity being stored, whether it be in a database or not.

Applied Processes: The screens that this data entity will be used for.

MySQL Data Type: The type of data for a data entity

Data Size: The size of the data in terms of it's data type.

Description: A description of what data this entity is storing.

Acceptable Input Characters: The specific range of input that is expected by the system.

Acceptable Range of Input: The range of characters that can be accepted by the system.

Incorrect Example of Input: An example of data input that would not be accepted by the system

Notes: Additional information regarding the data.

## FSH Tech

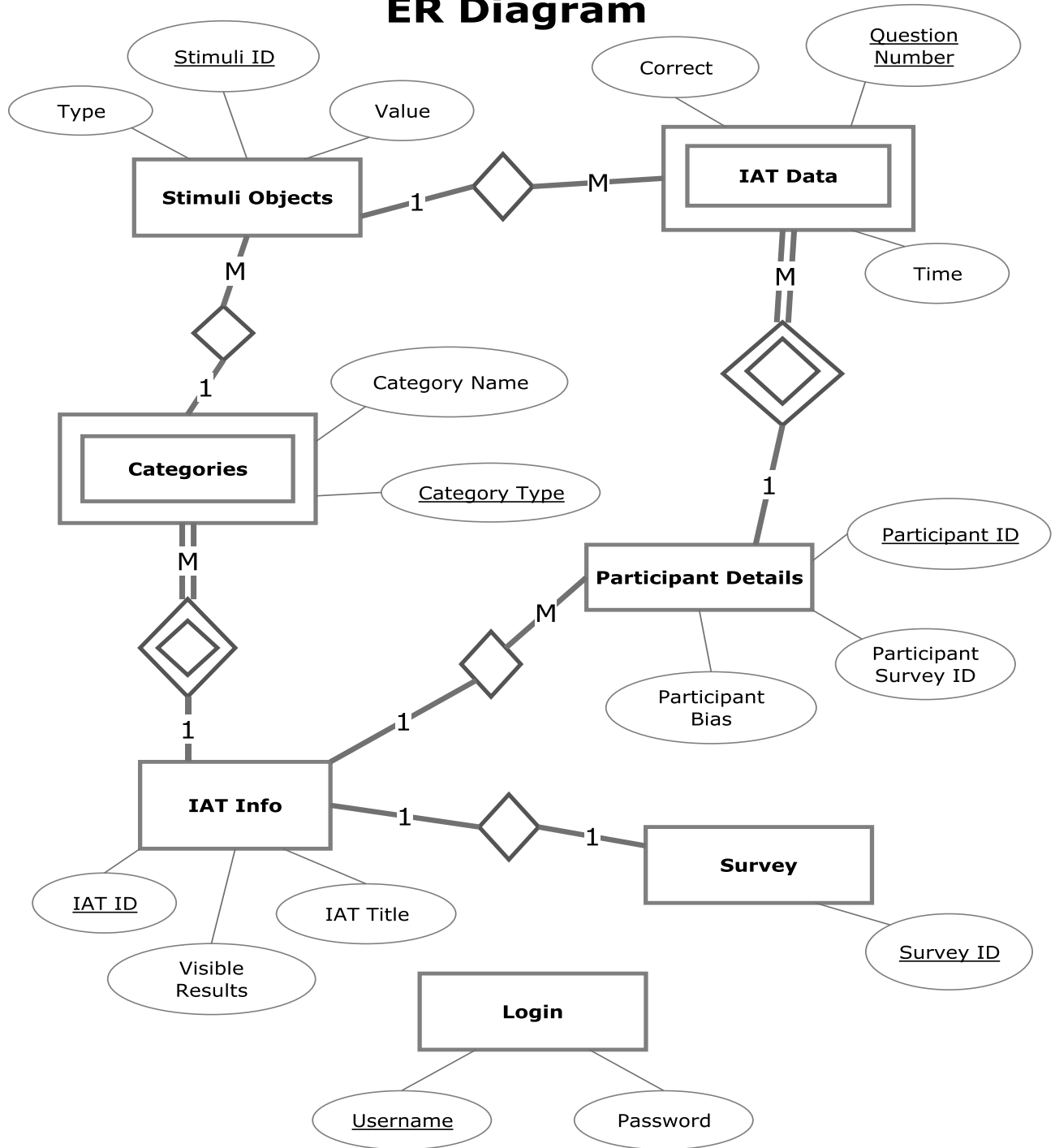
### Data Dictionary

L = Login; C = Create IAT; V = View Results; T = Take IAT; E = Export Results

Table	Data Name	Applied Process(es)	MySQL Data Type	Data Size	Description	Acceptable Input Characters	Acceptable Range of Input	Unacceptable Input	Notes
Categories	Category Name	C, V, T, E	varchar	1-50 Characters	Name given to the different categories		A-Z, a-z, 0-9	""	
Categories, Stimuli Objects	Category Type	C, V, T, E	varchar	1-50 Characters	Distinguishes the category type that the stimuli falls under	P1, O1, P2, O2	A-Z, a-z, 0-9	null	
IAT Data	QuestionNumber	V, T, E	int	1-3 Digits	Question number in IAT	numbers 1-200	0-9	first	
IAT Data	Time	V, T, E	time	1-10 Digits	Time (ms) it took participant to answer questions		0-9	flag	
IAT Data	Correct	V, T, E	int	1 Digit	if the participant got the question right or wrong		0-1	""	
IAT Data, Stimuli Objects	Stimuli ID	C, V, T, E	int	1-2 Digits	Unique ID for Stimuli Object		0-9	computer	
IAT Info	IAT Title	C, V, T, E	varchar	1-50 Characters	Title of IAT		A-Z, a-z, 0-9	IAT	
IAT Info	Visible Results	C, V, T	int	1 Digit	States whether or not participant can view their IAT result		0-1	yes	
IAT Info, Participant Details, Categories, Stimuli Objects	IAT ID	C, V, T, E	int	1-3 Digits	Unique ID for IAT		0-9	THE IAT	
IAT Info, Survey	Survey URL/ID	C, V, T, E	varchar	1-200 Characters	Reference to the survey being used for the IAT		A-Z, a-z, 0-9, /, ., :	...	
Login	username	L	varchar	1-20 Characters	Username		A-Z, a-z, 0-9	(?!)*@!	only necessary for administrator
Login	password	L	varchar	1-20 Characters	Password		A-Z, a-z, 0-9, !, @, #, \$, %, ^, &, *, (, )	null	only necessary for administrator
Participant Details	Participant Bias	V, T, E	decimal	1-15 Digits	calculated bias score		0-9	7	
Participant Details, IAT Data	Participant ID	V, T, E	int	1-9 Digits	ID of participant		0-9	q	
Participant Details, Survey	Participant Survey ID	V, T, E	int	1-9 Digits	Survey ID of participant		0-9	""	
Stimuli Objects	Type	C, V, T, E	varchar	1-20 Characters	States what type of stimuli is stored: a text or a file path to an image		A-Z, a-z	2563	
Stimuli Objects	Value	C, V, T, E	varchar	1-100 Characters	Contains either the word stimuli or the file path to the image		A-Z, a-z, 0-9, /	null	

### 3.4 | Entity Relationship Diagram

## F.I.L.E.T / S.A.S ER Diagram



## 4 | Architectural Design Specifications

### 4.1 | Development, Operation, and Maintenance Environment

FSH Tech will be developing F.I.L.E.T. using the resources available in our software engineering lab. The list below will contain the equipment used to create our software and our team website, as well as keep them maintained.

#### **Server:**

Web Server: Apache version 2.2.21  
PHP version 5.2.17  
MySQL 5.1.58-community log

#### **Windows Machine:**

Operating System: Microsoft Windows Vista Enterprise (32 bit) Service Pack 2  
Hardware:  
Processor: Intel Core 2 Duo, 2.93 GHz  
Memory: 4.00 GB RAM  
Software:  
Microsoft Office 2007  
Macromedia Dreamweaver, Fireworks  
Internet Explorer 9, Mozilla Firefox 7.0.1, Google Chrome 15.0.874.102

#### **Macintosh Machine:**

Operating System: Apple Mac OS X Version 10.6.4  
Model: iMac 5.1  
Processor: Intel Core 2 Duo 2 GHz  
Memory (RAM): 1.00 GB

## 4.2 | Deliverables

F.I.L.E.T. and all of its related files will be accessible on all major Internet browsers on [fsh.sienacs.com](http://fsh.sienacs.com).

A DVD will be delivered containing:

- Acceptance Test Documents
- Results
- The Acceptance test Power Point Presentation
- A full copy of team files from the team directory on 'Oraserv'
  - All folders, files, images
  - The team website source code
- The F.I.L.E.T. project source files
- A README.txt file that explains what files are located in different areas
- The lyrics to the FSH Tech's team song

# Appendices

Appendix A: Test Plan and Results.....	Separate Document
Appendix B: Data Flow Diagrams.....	Separate Document
Appendix C: Source Code.....	Separate Document
Appendix D: Glossary of Terms.....	25
Appendix E: Timelines.....	26



## Appendix D: Glossary of Terms

**C++:** (Pronounced “C plus plus”) is a compiled general-purpose programming language regarded as a middle-level language.

**Classic Waterfall Model:** A basic software development strategy that clearly labels each phase of the software engineering process. The strategy follows consecutively the following steps: Requirements Specification, Design, Construction, Verification, and Maintenance

**CSS:** (Cascading Style Sheets) are used to describe a specific style for a web page or set of pages.

**Gantt Chart:** A type of bar chart that illustrates a project schedule. This chart illustrates the start and finish dates of the terminal elements and summary elements of a project.

**HTML:** (HyperText Markup Language) is a scripting language used to design the structural layout of a website.

**IAT:** (Implicit Association Test) measures within a social psychology design to detect the strength of a person’s automatic association between mental representations of objects (concepts) in memory.

**JavaScript:** An object-oriented scripting language used to enable programmatic access to objects within both the client application and other applications.

**MySQL:** An open source relational database management system.

**PHP:** (PHP Hypertext Preprocessor) a widely-used, general-purpose scripting language that is especially well-suited for web development and can be embedded into HTML.

**XHTML:** (Extensible Hypertext Markup Language) is a reformation of HTML as an XML application.

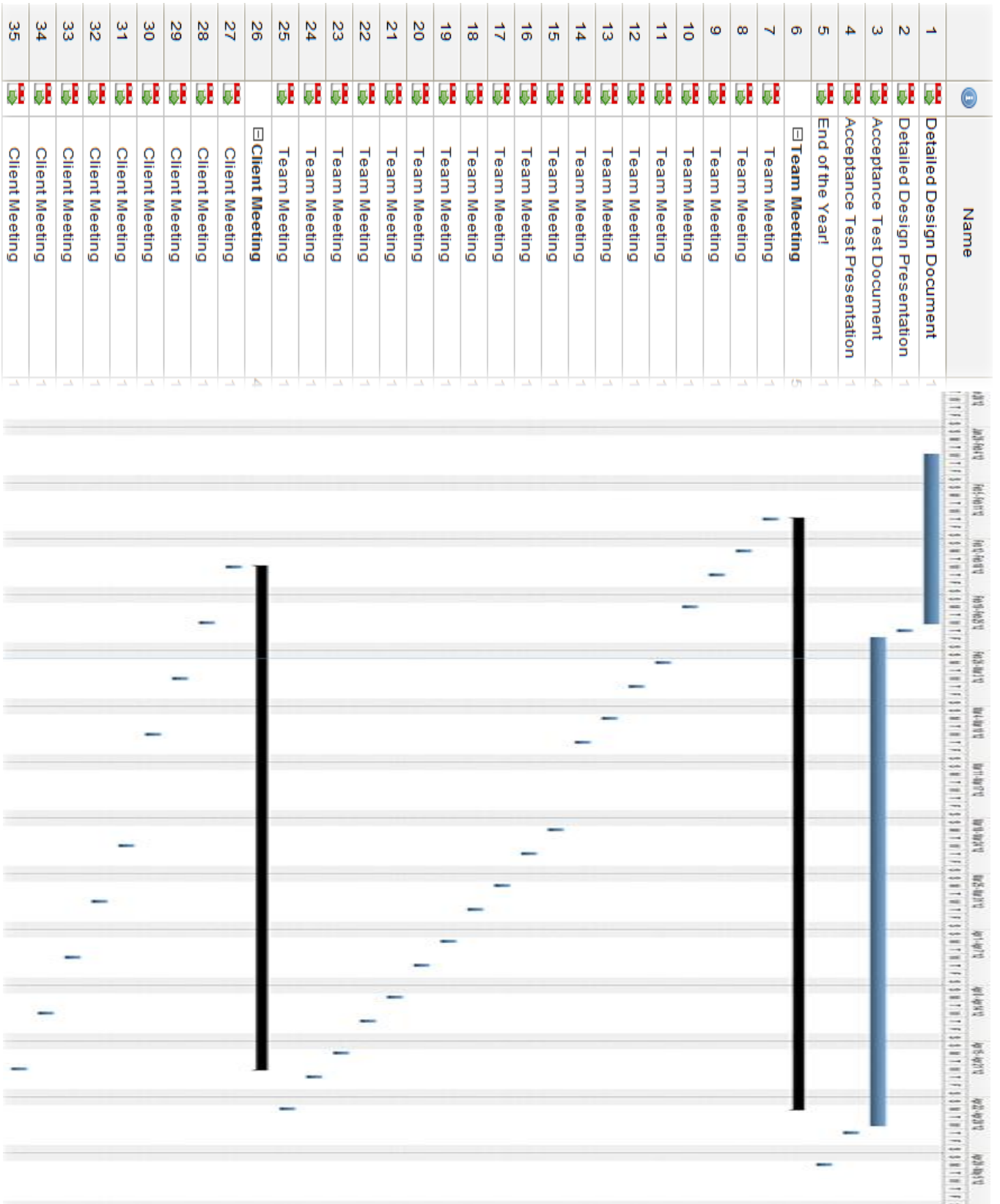
**XML:** (Extensible Markup Language) A set of rules for encoding documents electronically.

## Appendix E: Timelines

# Gantt Chart: Fall 2011



# Gantt Chart: Spring 2012



# Acceptance Test

## Appendix A: Test Plan

Factual Inter Leanings that Examine Throughput  
(F.I.L.E.T.)

**Requested By:**

Dr. Eric Breimer  
Associate Professor of Computer Science  
Siena College

**Delivered By:**

FSH Tech

**Prepared By:**

Jacquelyn Boylan  
Matthew Brancato  
Matthew Kemmer  
Serena Moore  
Nydia Negrón  
Mike Tanski

April 25, 2012

# F.I.L.E.T.

## Detailed Design

### Appendix A: Test Plan

## Table of Contents

## 1 | Introduction

1.1   Introduction.....	3
1.2   Testing Plan Identifier .....	4

## 2 | Testing Criteria

2.1   Testing Approach.....	5
2.2   Acceptance Test – Acceptance Criteria .....	6
2.3   Functional Requirements Inventory .....	7
2.4   Non-Functional Requirements Inventory.....	9
2.5   Exception Handling to Test.....	10
2.6   Unit Tests.....	11
2.6.1   Directory.....	11
2.6.2   Login.....	12
2.6.3   View Home Screen.....	13
2.6.4   View IAT Data.....	14
2.6.5   View IAT Test Data.....	15
2.7   Integration Testing .....	16

# 1 | Introduction

## 1.1 | Introduction

The Test Plan will preview and test the functionality of the inner workings of F.I.L.E.T. There are several types of tests that FSH will be using to check the functionality of F.I.L.E.T. These test types include:

Test Case – The simplest form of testing, that concerns a particular action and has two forms: pass or fail.

Unit Test – concerned with the knowledge about testing a program unit, typically developed by a single individual, to determine that it is free of data, logic, or standards errors. The unit test is made up of pertinent test cases. This unit includes knowledge of dynamic analysis (equivalent partitioning, boundary value analysis, cause-effect graphing, logic based testing, random testing, and syntax testing) and static analysis (complete path testing, decision testing, condition testing, and data-flow testing).

\*Extracted from Software Engineering Body of Knowledge Version 1.0

Integration Test – concerned with knowledge about validating that software components, which have been unit tested separately, interact correctly when they are put together to perform a higher order function. It involves checking the current unit and all other units that integrate with the unit. This unit also includes knowledge about dependency checking for class, data, and processes, and about interface checking in terms of range, type compatibility, representation, number and order of parameters, and method of transfer.

\*Extracted from Software Engineering Body of Knowledge Version 1.0

System Test – concerned with the knowledge about validating the specified functional requirements of a system. The system test includes knowledge about techniques to design and enact an independent testing process of all of the system's functions described in the software requirements specification.

\*Extracted from Software Engineering Body of Knowledge Version 1.0

## 1.1 | Introduction (Continued)

Acceptance Test – concerned with the knowledge about validating the functional and non – functional requirements of a purchased or acquired system. The acceptance test includes knowledge about techniques for using the contract, the statement of work, the software requirements specification, and the request for proposal to ensure that the delivered system meets all of the requirements (as perceived by the purchasing or acquiring organization)

## 1.2 | Testing Plan Identifier

The test plan is a document that ensures the testing requirements for the system are fulfilled. It is a dynamic document as tests are fulfilled and some requirements are removed or altered to fit the software. This document is a detailed checklist that all functional and non-functional requirements are met by the completion of the project. This document will provide all the changes that were made to the system to ensure that it fully meets requirements, as well as to any changes in the testing to ensure that the Acceptance Test phase fully tests the system.

## 2 | Testing Criteria

### 2.1 | Testing Approach

The method our team will use to test F.I.L.E.T (Factual Inter Leanings that Examine Throughput) will be to follow the functional and non-functional requirements as defined in this document and others. The first test will be the Unit Test as defined below. The Unit Test will test the lowest level functions of the application across each type of screen a user of the IAT may encounter. Each test will have detailed descriptions of each test. The tester(s) will go through each page in the system checking each test off as passing or failing. The tester(s) will repeat this test on the various web browsers that have been stated below in the acceptance test. An Integration/Regression Test (defined later in this document) will then be implemented to ensure that while each unit in the Unit Test is passed, it does not interfere with other units in the Unit Test. This will help to ensure that no errors are overlooked and can be fixed. When errors occur or Unit Tests are failed, the team will coordinate to come up with a solution to the problem and then the Unit Test will be performed again, completely, to ensure that nothing else was changed in fixing the previous problem. The Integration/Regression Test will be implemented once we are satisfied that the Unit Test has been passed acceptably, with all failed units either fixed or reported on. If the Regression Test determines that there are issues of one feature breaking another feature, the team will again coordinate to fix the problem. The Unit Test will then be re-implemented to ensure that while fixing the problem another problem was not caused and then re-implementing the Regression Test. In order to test the Non-Functional Requirements the tester(s) will have to use the system keeping in mind each of the specific Non-Functional Requirement and weighing whether F.I.L.E.T. passes each one.



## 2.2 | Acceptance Test – Acceptance Criteria

Acceptance Testing is the way in which a system is validated for meeting all functional and non-functional requirements. It is an organized and detailed way of showing how the system can meet all the requirements which makes it easier for outside viewers to understand what was done and why. The test plan of the project describes the objective, scope, and focus to make certain that the software meets all the specifications through testing.

The Acceptance Criteria will be determined by the functional and non-functional requirements specified in sections 2.3 and 2.4. The functional requirements are requirements from the client that are testable and can be measured to pass or fail a test. On the other hand, non-functional requirements are requirements of how the program will be, as in look and usability, which are not easily testable. At the completion of the software development, FSH Tech will use these testing criteria to determine which requirements were met and which have not been met. These requirements are subject to change during development and testing based on the client's request as more information about the development is gathered.

F.I.L.E.T. will be tested on both Windows and Mac operating systems and with the four major browsers Internet Explorer, Safari, Google Chrome, and Mozilla Firefox. The system will be tested based on testing requirements determined by FSH Tech. Testing will be done in three incremental stages starting at the basic unit testing level and evolving to a full system test. The unit tests are individual tests that determine whether a small minute portion of the program meets requirements specifications or not. Each unit test gets tested on its own accord separately from the other portions without knowing its effects on the other parts of the system that it will interact with. Then the modules that are comprised of certain unit tests will be tested to make certain that the system is working properly on a modular level. Once the unit testing is completed, as well as the module testing, the system will go through integration tests to see how integrated parts of the system are running to see if they are meeting requirements specifications as well. Lastly the Systems test will be performed to ensure that the system as a whole meets all the system requirements set forth by the client and FSH Tech. The full results will be documented in the Acceptance Test, which FSH Tech and the client will use to determine if the system has met all functional and non-functional requirements or not.

## 2.3 | Functional Requirements Inventory

This section is used to check that the functional requirements of F.I.L.E.T. are met. The functional requirements can be physically tested since they are either classified as met or unmet depending on the data the unit tests provide. It will serve as a high-level checklist during our detailed testing, ensuring Dr. Breimer's requirements are met.

The following is a list of functional requirements for the two types of users, the Administrator and the Participant.

### Administrator

- |                                  |    |   |
|----------------------------------|----|---|
| <input checked="" type="radio"/> | NO | Will be able to authenticate                                |
| <input checked="" type="radio"/> | NO | Will be able to enter username and password                 |
| <input checked="" type="radio"/> | NO | Will be able to view IAT results                            |
| <input checked="" type="radio"/> | NO | Will be able to view survey results                         |
| <input checked="" type="radio"/> | NO | Will be able to view test data                              |
| <input checked="" type="radio"/> | NO | Will be able to export IAT results to a spreadsheet         |
| <input checked="" type="radio"/> | NO | Will be able to export only survey data                     |
| <input checked="" type="radio"/> | NO | Will be able to export all data                             |
| <input checked="" type="radio"/> | NO | Will be able to export only statistical association results |

## 2.4 | Non-Functional Requirements Inventory

The non-functional requirement inventory is a list of non-functional system requirements. This list is composed of requirements that specify what F.I.L.E.T does.

- YES NO  
The system will be maintainable. Changes can be made easily and should not be time consuming.
- YES NO  
The system will be stable, being able to perform under a high level of activity, while maintaining a high level of performance.
- YES NO  
The system will be operational on several different browsers, without errors loading.
- YES NO  
The system will be easy to use, with logical and easy to use navigations.
- YES NO  
They system will be visually appealing, and will make sense stylistically.

## 2.5 | Exception Handling To Test

There are certain areas in F.I.L.E.T where errors can occur due to actions made outside the systems control. The system must be able to handle these types of exceptions.

- If either the connection is broken during the test, or the Participant closes the window, the results will not be recorded
- If the Participant fails to fill out necessary survey questions, they will not be able to proceed to the test and be notified of the blank questions.
- If the Participant hits foreign keys during the test, they will be notified so they may find the correct keys
- If the Participant's response times are either too slow or too fast, their results will not be recorded
- If the Administrator uses an incorrect username and password combination, he will be notified, and not allowed to proceed
- If the Administrator fails to name an IAT, he will not be able to proceed to selecting category titles, and will be notified
- If the Administrator creates an IAT with the same name as a previously created one, he will be able to overwrite the old IAT, but will be prompted to first
- If the Administrator fails to input four, unique category titles, he will not be able to proceed to inputting stimuli
- If the Administrator attempts to upload an unexpected file type instead of an image, he will be notified of the error and the file will not be uploaded
- If the Administrator attempts to create an IAT with no stimuli objects, the IAT will not be hosted and he will notified
- If there is a loss of connection while attempting to create the IAT, the Administrator will be notified
- If the Administrator attempts to view the results of a test that has not yet been taken, he will be notified that there is no data
- If the Administrator attempts to export the results of a test that has not yet been taken, he will be notified that there is no data, and a spreadsheet will not be generated.



## 2.6 | Unit Tests

### 2.6.1 | Directory

#### 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	Comments or brief description	Integrated with these units	
F	100%	6	0	1	<a href="#">Login</a>	04/25/12	Authentication	2,6
F	100%	5	0	2	<a href="#">Querying the Database</a>	04/26/12	Access to the Database	1,3,4,5,6,7,8
F	100%	8	0	3	<a href="#">Create IAT</a>	04/26/12	Testing the view of a specific IAT's data	2,4,5,6
F	100%	11	0	4	<a href="#">Take IAT</a>	04/26/12	View Homescreen	2,7,8
F	100%	6	0	5	<a href="#">Add Stimuli</a>	04/26/12	View IAT Test Data	2,3,4
F	100%	6	0	6	<a href="#">View Homescreen</a>	04/25/12	Tests the functions of the home screen after authentication	1,2,3,7,8
F	100%	9	0	7	<a href="#">View IAT Test Data</a>	04/25/12	Testing the view of a specific participant's results	2,4,5,6,7
F	100%	6	0	8	<a href="#">View IAT Data</a>	04/26/12	Testing the view of a specific IAT's data	2,4,5,6,8

100.00% of Test Cases Passed (99.44% passes the Ivory Snow Test)



## 2.6.2 | Login

FSH Tech  
 F.I.L.E.T.  
 Dr. Eric Breimer  
 Login

Tests the Login

Test Cases										
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
P	1.001	Null username field	Leave username field blank	Click the login button	Blank username field	Output "please log in to continue"	Output "please log in to continue"		Jackie Boylan	4/25/12
P	1.002	Null password field	Leave password field blank	Click the login button	Blank password field	Output "please log in to continue"	Output "please log in to continue"		Jackie Boylan	4/25/12
P	1.003	Null UserName and Password	Leave both username and password field blank	Click the login button	Blank user name and password field	Output "please log in to continue"	Output "please log in to continue"		Jackie Boylan	4/25/12
P	1.004	Incorrect password for identified username	Insert incorrect password for given username	Click the login button	Valid username, invalid password	Output "please log in to continue"	Output "please log in to continue"		Jackie Boylan	4/25/12
P	1.005	Non existant username	Insert an invalid username	Click the login button	Invalid username	Output "please log in to continue"	Output "please log in to continue"		Jackie Boylan	4/25/12
P	1.006	Login	Insert valid username and corresponding password	Click on the login button	Filled in form	User brought to their homepage	homepage loaded		Jackie Boylan	4/25/2012
P	Unit Summary		100%	passing	6	passed	0	failed	Date of last test = 4/25/12	
	6 tests									

[Directory Page](#)

## 2.6.3 | View Home Screen

FSH Tech  
 F.I.L.E.T.  
 Dr. Eric Breimer  
 View Homescreen

Tests the functions of the home screen after authentication

Pass/Fail Status	Test Cases		Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
	Test Number	Description								
P	6.001	Opens Correctly in All Browsers	View Screen in IE, Firefox, Chrome, Safari	Load page in all web browsers	Pre-Populated Form	Homescreen Page is loaded correctly	homepage loads correctly	only works with javascript enabled	MB	4/25/12
P	6.002	Load Page	Entering page after login	Enter page	Login Screen	Deployed IAT's are listed	Deployed IAT's display	needs more iats	MB	4/25/12
P	6.003	Create IAT button works	Button click	Clicking "create IAT" button and navigating to Create IAT page	List of Deployed IAT's are shown	Create IAT page is loaded	page loads and prompts for necessities		MB	4/25/12
P	6.004	All deployed IAT's are shown	Entering page after login	Observe to see all IAT's created are listed on the page	Login Screen	Every IAT deployed is listed	list shown		MB	4/25/12
P	6.005	View Data button connects to correct IAT	Make sure that "View Data" button navigates to correct IAT Data	Selecting "View Data" next to desired IAT	List of Deployed IAT's are shown	Navigate to selected IAT's Data page	view data displays		MB	4/25/12
P	6.006	View Survey Data connects to LimeSurvey Data page	Make sure that "View Survey Data" button navigates to correct LimeSurvey survey based on the IAT	Selecting "View Survey Data" next to desired IAT	List of Deployed IAT's are shown	Limsurvey View Survey data page is loaded	links to limesurvey	must be logged in before to be taken in otherwise prompted	MB	4/25/2012
P	Unit Summary		100%	passing	6	passed	Date of last test =		4/25/12	
	6 tests				0	failed				

[Directory Page](#)

## 2.6.4 | View IAT Data



Testing the view of a specific IAT's data, containing all the participants who have taken the specified IAT

Test Cases										
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
P	7.001	Populates	Press View IAT Button	Press Button	Pre-Populated Form	View IAT Page is loaded	page loads		mb	4/25/12
P	7.002	Opens Correctly in All Browsers	View Screen in IE, Firefox, Chrome, Safari	Load page in all web browsers	Pre-Populated Form	View IAT Page is loaded correctly	loads on browsers		mb	4/25/12
P	7.003	Correct Data	Make sure that the correct IAT data is loaded	View individual participant results and make sure data matches IAT Title	Text box filled and no category radio button selected	No action	does nothing till radio button selected		mb	4/25/12
P	7.004	Check View Survey Button	Make sure when View Survey button brings user to View IAT Survey Results	Click View Survey Button	At View IAT Data page	View IAT Survey Results is populated	prompts for login then takes to survey	only prompts if prior login for limesurvey wasn't in current session	mb	4/25/12
P	7.005	Check View Test Data Button	Make sure when View Test Data button brings user to View Test Data Results	Click View Test Data button	At View IAT Data page	View Test Data Results is populated	works as expected		mb	4/25/12
P	7.006	Check IAT Score is calculated	Check that a possible score is being calculated as an participant's IAT score (between 1 and 1)	Observe Participant Test Data and calculate score, and compare to IAT score calculated	View IAT Data displays computed IAT Score	Manual calculations matches computed IAT score			MK	4/25/12
P	7.007	Check Export All Data button is working	Makes sure that every participant's data is being exported to an excel file separate from the system	Click View Test Data button	At View IAT Data page	View Test Data Results is populated	works as expected		mb	4/25/12
P	7.008	Check Export All Survey button is working	Make sure when View Test Data button brings user to View Test Data Results	Click View Test Data button	At View IAT Data page	View Test Data Results is populated	works as expected		mb	4/25/12
P	7.009	Check Export List Here button is working	Make sure when View Test Data button brings user to View Test Data Results	Click View Test Data button	At View IAT Data page	View Test Data Results is populated	works as expected		mb	4/25/12
P = Unit Summary			100% passing		9	passed	Date of last test =		4/25/12	
9 tests					0	failed				

[Directory Page](#)

## 2.6.5 | View IAT Test Data

FSH Tech  
 F.I.L.E.T.  
 Dr. Eric Breimer  
 View IAT Test Data

Testing the view of a specific participant's results

Pass/Fail Status	Test Cases		Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
	Test Number	Description								
P	8.001	Opens Correctly in All Browsers	View Screen in IE, Firefox, Chrome, Safari	Load page in all web browsers	Pre-Populated Form	View IAT Test Data Page is loaded correctly	works as expected		MB	4/25/12
P	8.002	Load Page	Press View Test Data Button	Press Button	Pre-Populated Form	Participant's IAT Page is loaded	works as expected		MB	4/25/12
P	8.003	Correct Participant's Data is shown	Make sure that the correct participant IAT data is loaded	View individual participant results and make sure data matches IAT Title	View IAT Data Page of a particular participant	Participant's IAT data matches the data in the database	works as expected		MB	4/25/12
P	8.004	Numbering system is correct	list going from 1-200	View Test Data page for the individual	View IAT Data Page of a particular participant	IAT Data is listed from 1-200	works as expected		MB	4/25/12
P	8.005	Stimuli objects are for the correct IAT	Stimuli Objects are shown and are correctly methoded to the IAT	View Test Data page for the individual	View IAT Data Page of a particular participant	The IAT name matches the stimuli shown	works as expected		MB	4/25/12
P	8.006	Participant's bias is displayed	Check participant's page to make sure calculated bias is shown	Loading View IAT Test Data page	Pre-Populated Form	Correct calculated bias	works as expected		MB	4/25/12
P	= Unit Summary 6 tests		100%	passing	6 0	passed failed		Date of last test =		4/25/12

[Directory Page](#)

## 2.7 | Integration Testing

Integration Testing is concerned with knowledge about validating that software components which have been unit tested separately. It checks that the units interact correctly when they are put together to perform a higher order function. This testing unit also includes knowledge about dependency checking for calls, data, and processes, and about interface checking in terms of range, type compatibility, representation, number and order of parameters, and method of transfer.

# Detailed Design

Factual Inner Leanings that Examine Throughput  
(F.I.L.E.T.)

**Requested By:**

Dr. Eric Breimer  
Associate Professor of Computer Science  
Siena College

**Delivered By:**

FSH Tech

**Prepared By:**

Jacquelyn Boylan  
Matthew Brancato  
Matthew Kemmer  
Serena Moore  
Nydia Negrón  
Mike Tanski

February 29, 2012

# F.I.L.E.T.

## Appendix B: Data Flow Diagrams

### Table of Contents

## 1 | Introduction

1.1   Introduction.....	3
1.2   Data Flow Legend.....	4

## 1 | Data Flow Diagrams

2.1   Context Diagram.....	5
2.2   Level 0 Diagram .....	6
2.3   Level 1: Process 1 Take IAT.....	7
2.4   Level 2: Process 1.1 Take Survey.....	8
2.5   Level 2: Process 1.2 Take Association Test.....	9
2.6   Level 1: Process 2 Manage IATs.....	10
2.7   Level 2: Process 2.1 Create IAT.....	11
2.8   Level 3: Process 2.1.1 Create Categories.....	12
2.9   Level 3: Process 2.1.2 Upload Stimuli.....	13
2.10   Level 2: Process 2.2 Manage Data.....	14
2.11   Level 3: Process 2.2.1 Select IAT .....	15
2.12   Level 1: Process 3 Authenticate.....	16

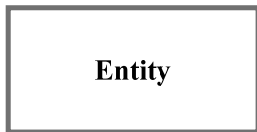
# 1 | Introduction

## 1.1 | Introduction

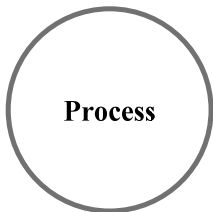
Data Flow Diagrams give a visual representation of how data moves and is transformed throughout the system, from users to processes to data stores. The diagrams start at a very high level and get more detailed at each further level. What will follow is a legend of the symbols used, and the diagrams themselves. While the full set of Data Flow Diagrams appears in this appendix, only the bolded Processes are portions of the system that are being developed by FILET, while the un-bolded Processes are portions of the system being developed by enigma elucidation and appear in their part of the system, SAS.

## 1.2 | Data Flow Legend

### F.I.L.E.T. by FSH Tech Data Flow Diagram Legend



**External or Internal Entities, which are either sinks for or sources of data**



**A Process that manipulates and passes on data in some way from or to Entities, other Processes, or a Datastore**



**Datastore which represents a place where data is received and held, such as a database**



**Data flow between Entities, Processes, and Datastores**



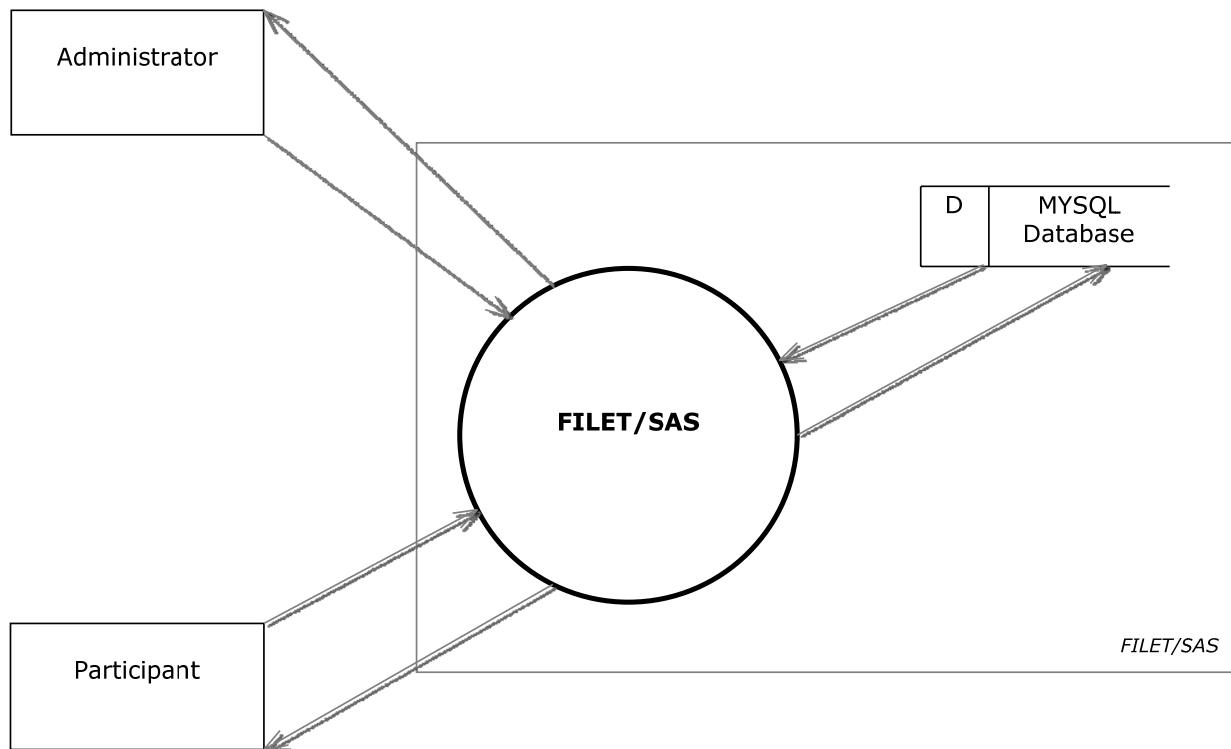
**The system boundary, between which the outside Entities interact with the system**

## 2 | Data Flow Diagrams

### 2.1 | Context Diagram

The context diagram gives a broad overview of the system and lists the primary users and external data stores.

FILET/SAS  
Context Diagram

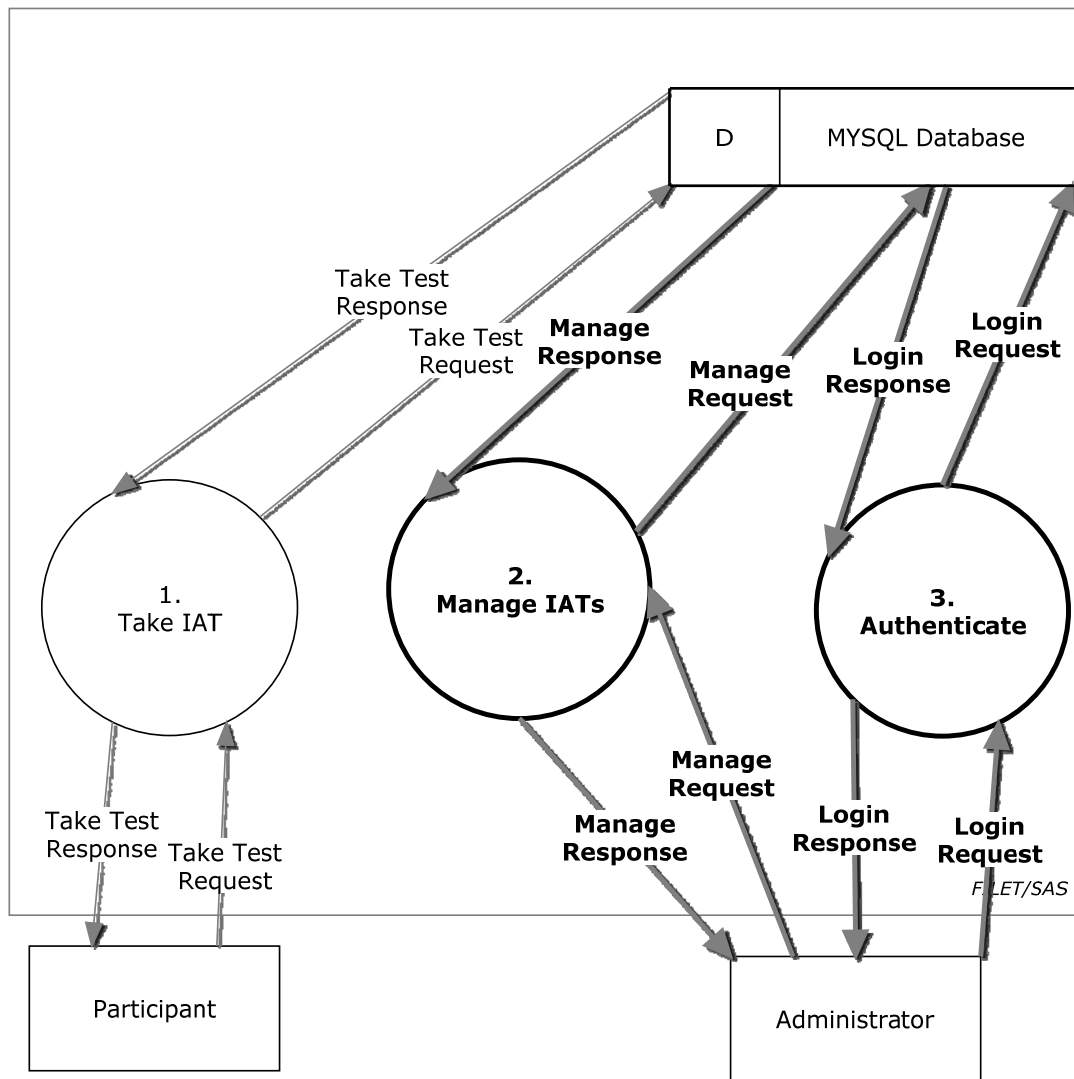




## 2.2 | Level 0 Diagram

The Level 0 Diagram lists the main processes that the system will be able to perform and the main pieces of data that they require and use. [Contains both FILET and SAS Processes]

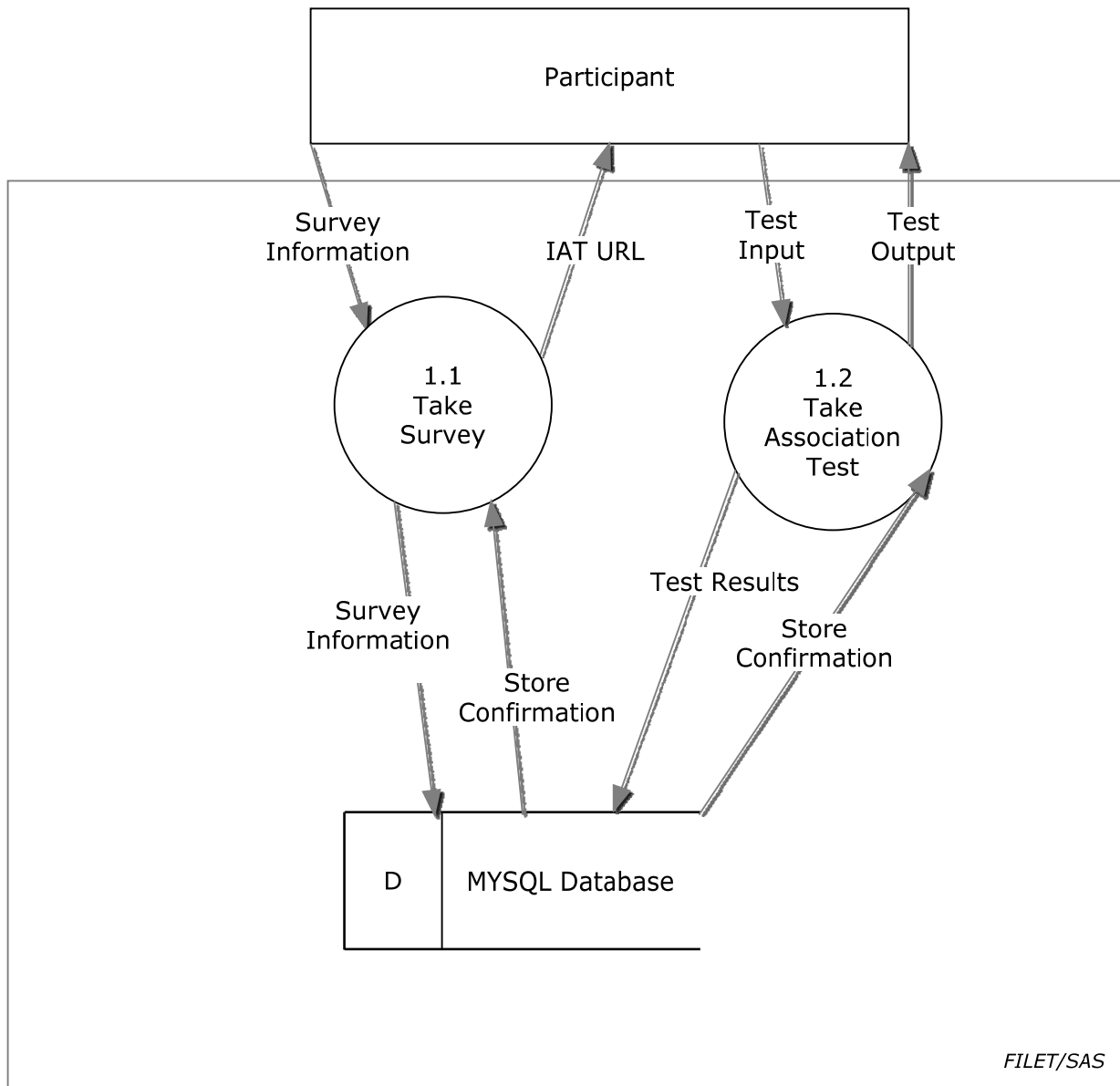
### FILET/SAS Level 0 Diagram



## 2.3 | Level 1: Process 1 Take IAT

The Participant's taking of the survey and the Association Test. [Contains only SAS Processes]

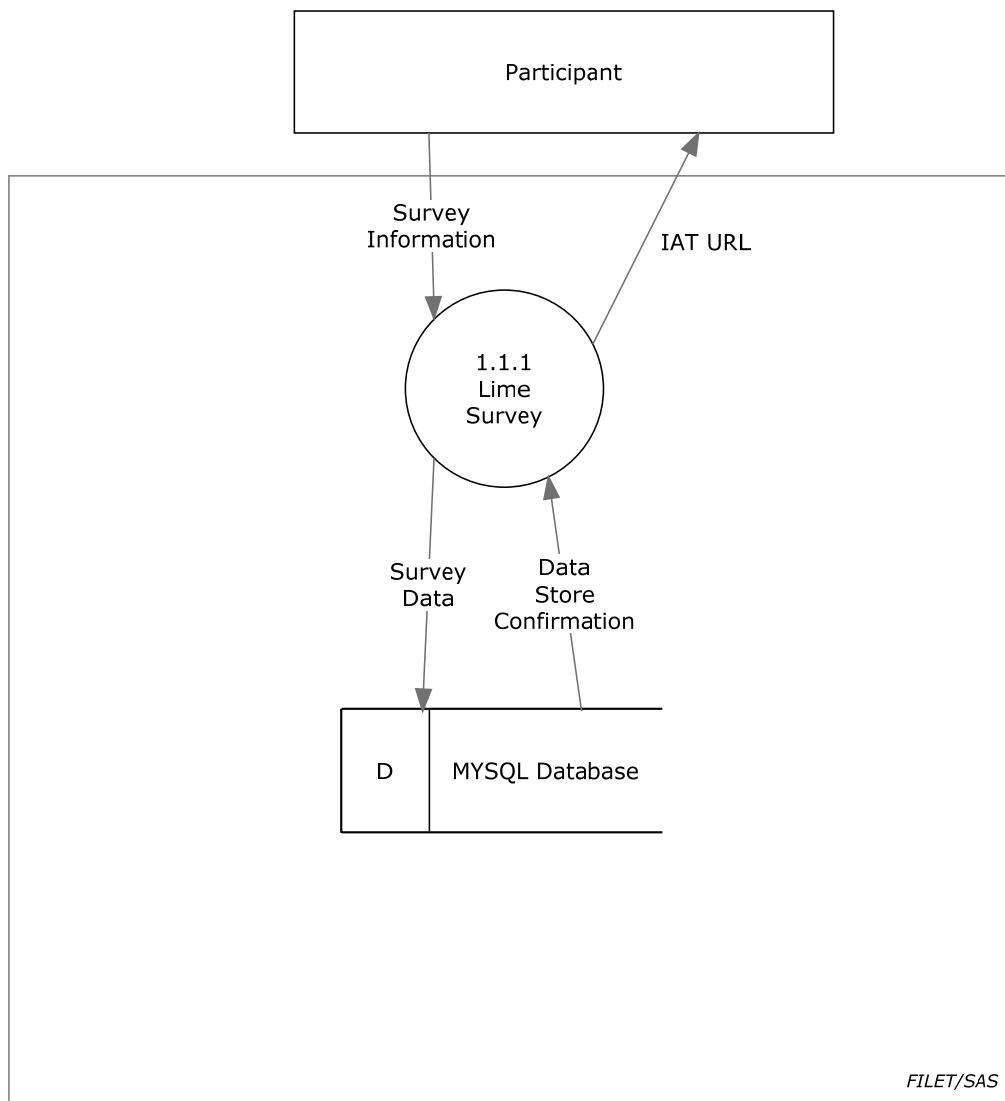
### FILET/SAS Level 1: Process 1 Take IAT



## 2.4 | Level 2: Process 1.1 Take Survey

The Participant is sent to Lime Survey to take a demographic survey. [Contains only SAS Processes]

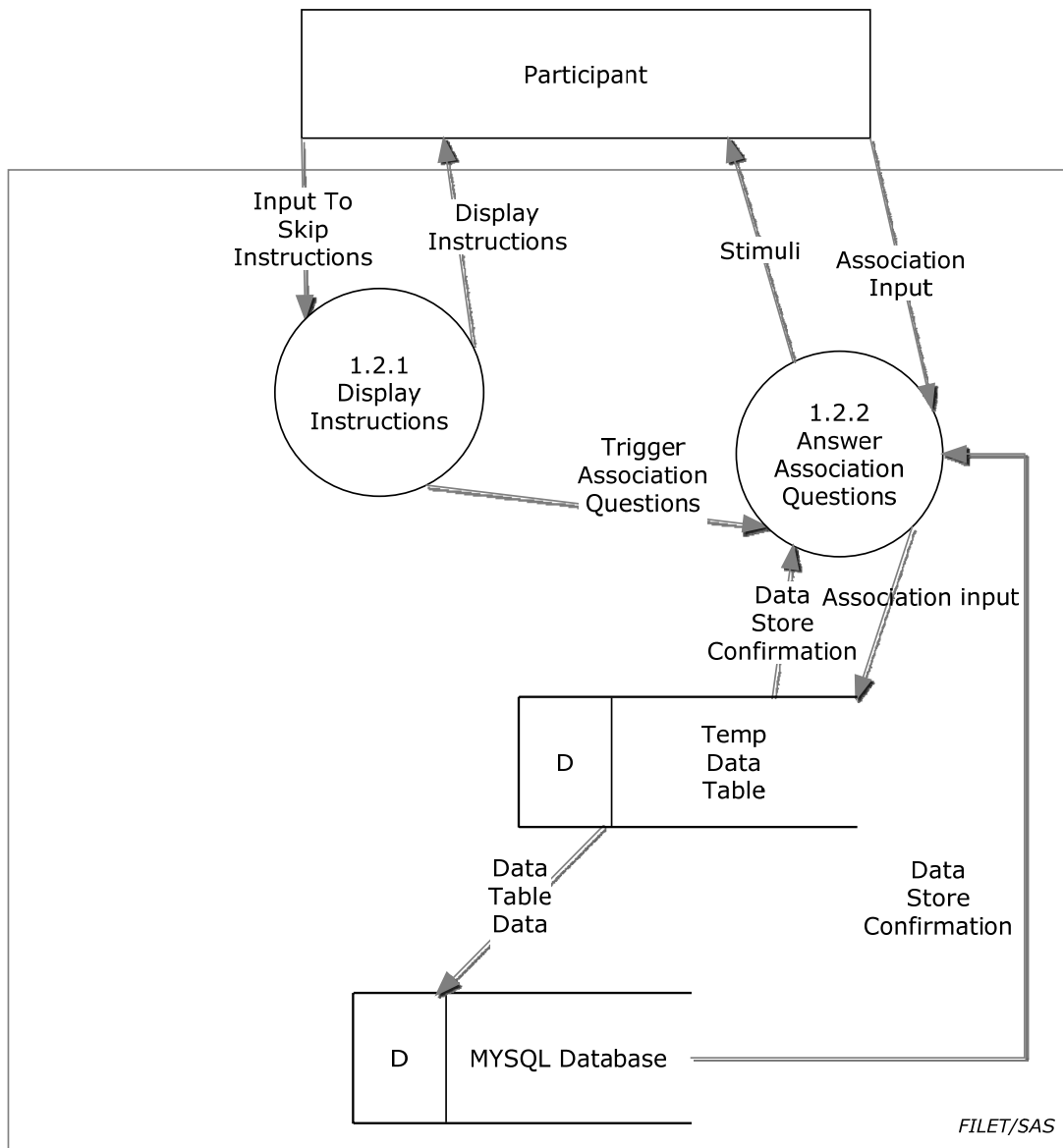
### FILET/SAS Level 2: Process 1.1 Take Survey



## 2.5 | Level 2: Process 1.2 Take Association Test

The Participant views the instructions of and takes the Association Test. [Contains only SAS Processes]

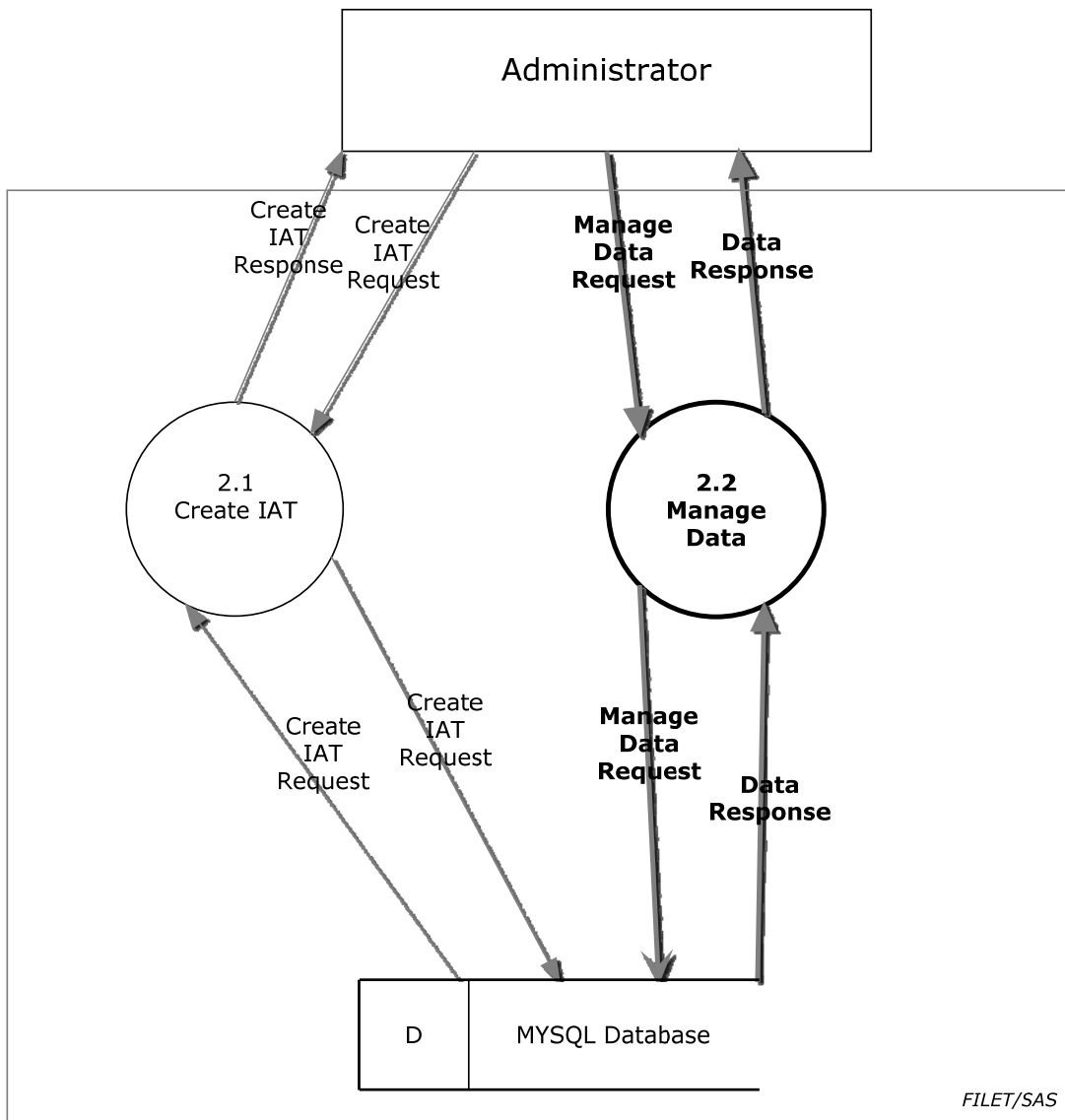
### FILET/SAS Level 2: Process 1.2 Take Association Test



## 2.6 | Level 1: Process 2 Manage IATs

The Administrator can create a new IAT or view existing IAT data. [Contains both FILET and SAS Processes]

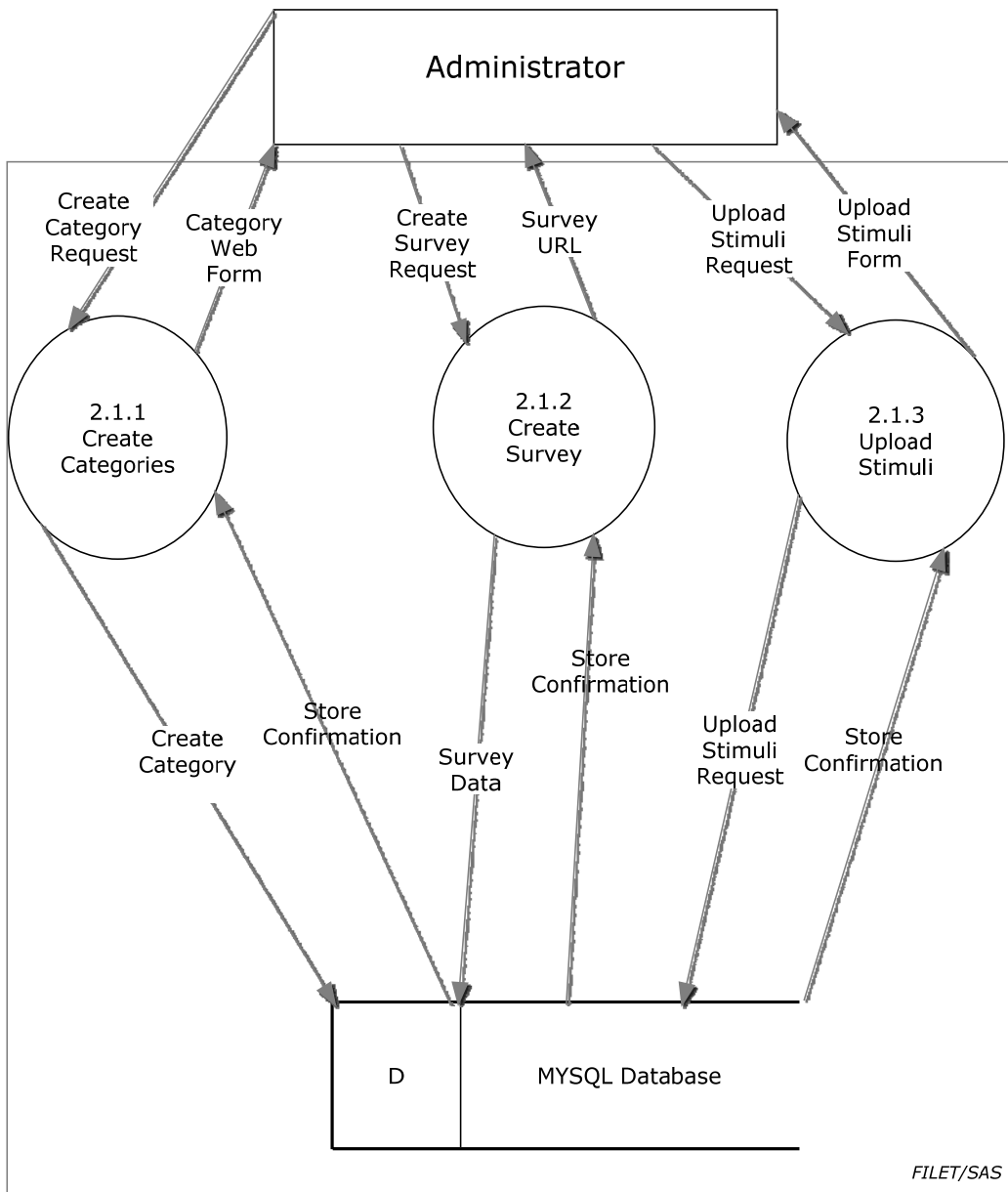
### FILET/SAS Level 1: Process 2 Manage IATs



## 2.7 | Level 2: Process 2.1 Create IAT

The Administrator creates a new IAT, made up of categories and stimuli, and a survey. [Contains only SAS Processes]

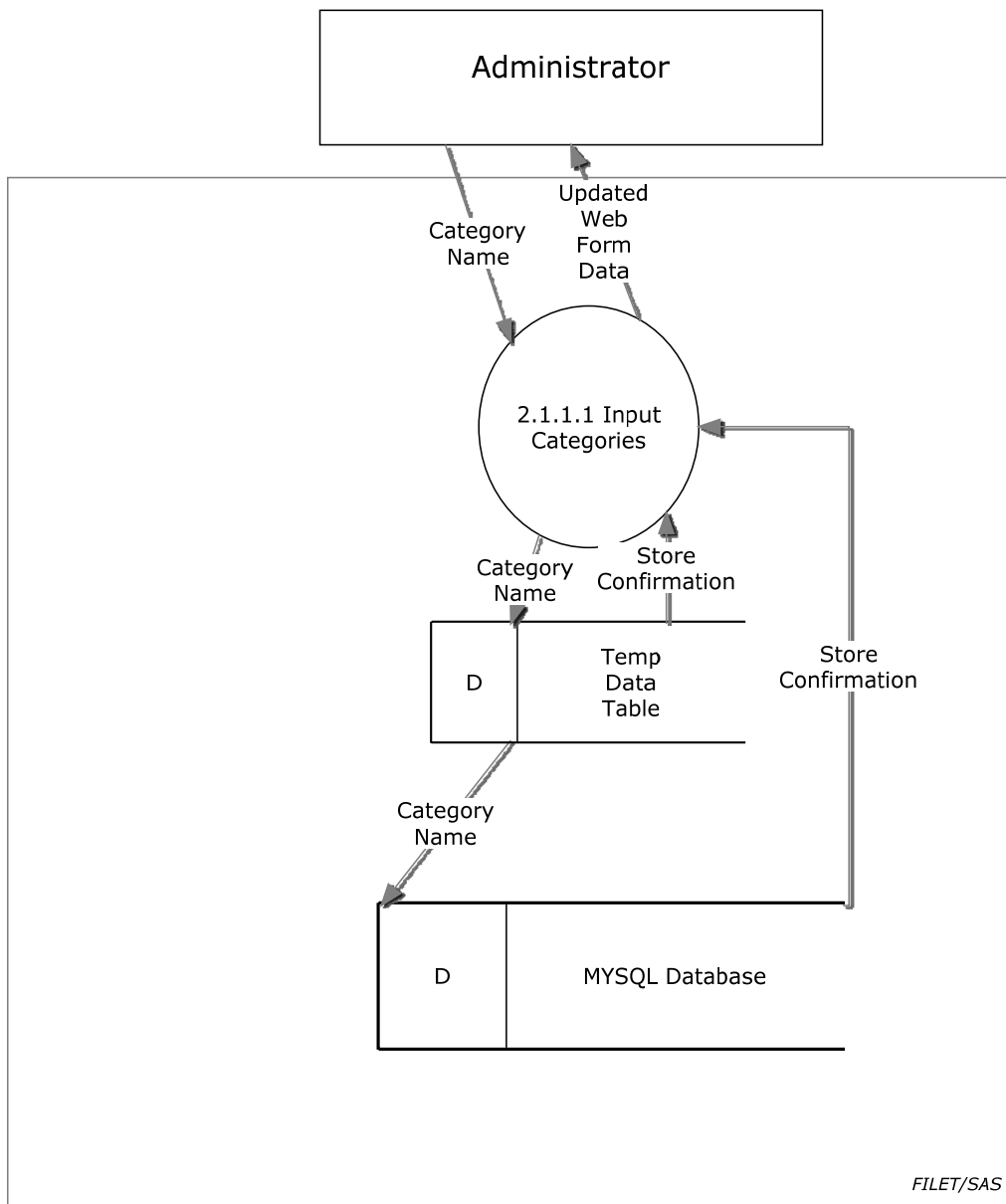
### FILET/SAS Level 2: Process 2.1 Create IAT



## 2.8 | Level 3: Process 2.1.1 Create Categories

The Administrator inputs the four categories associated with the test. [Contains only SAS Processes]

### FILET/SAS Level 3: Process 2.1.1 Create Categories

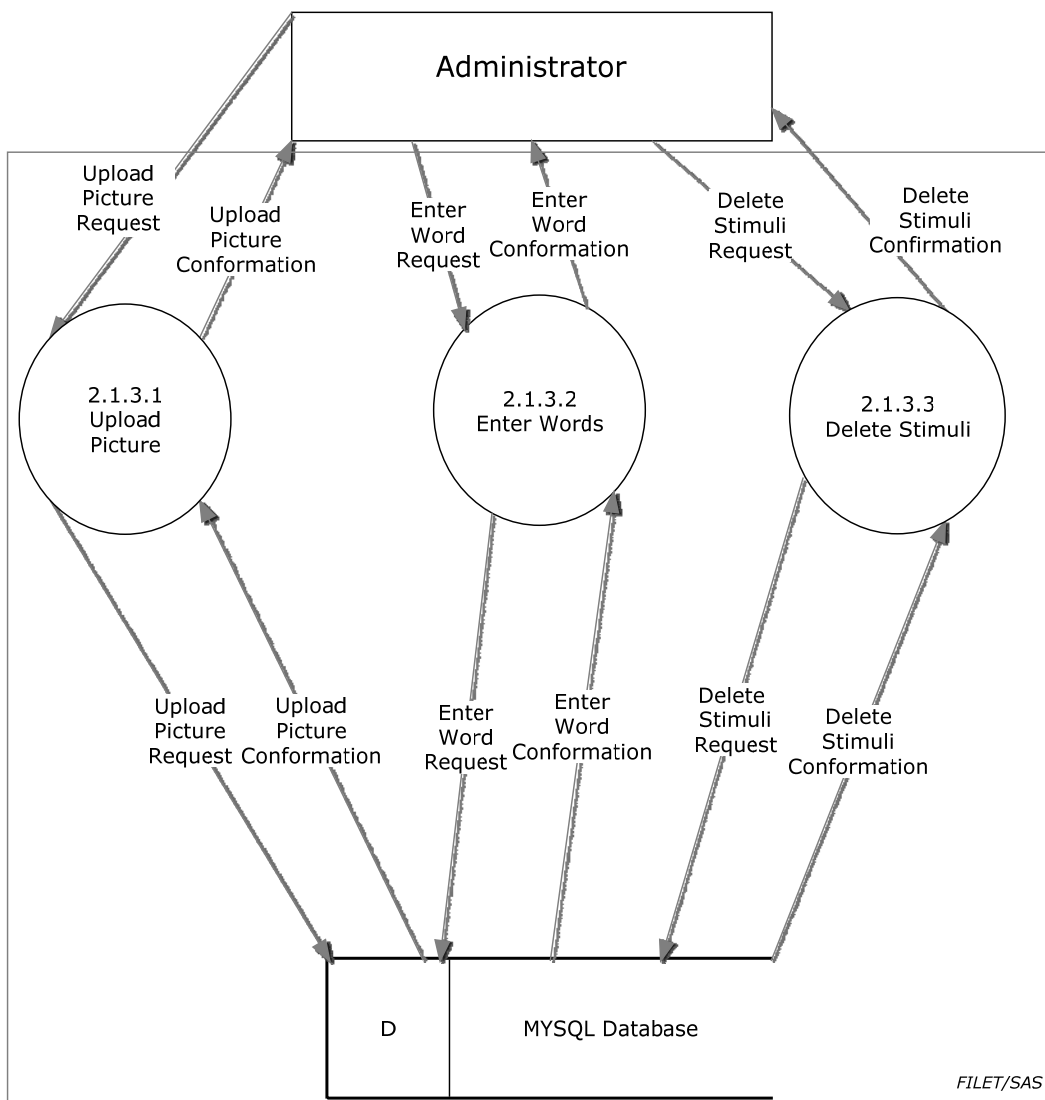


## 2.9 | Level 3: Process 2.1.1 Upload Stimuli

The Administrator can upload images or enter text as stimuli and delete existing stimuli.

[Contains only SAS Processes]

### FILET/SAS Level 3: Process 2.1.3 Upload Stimuli

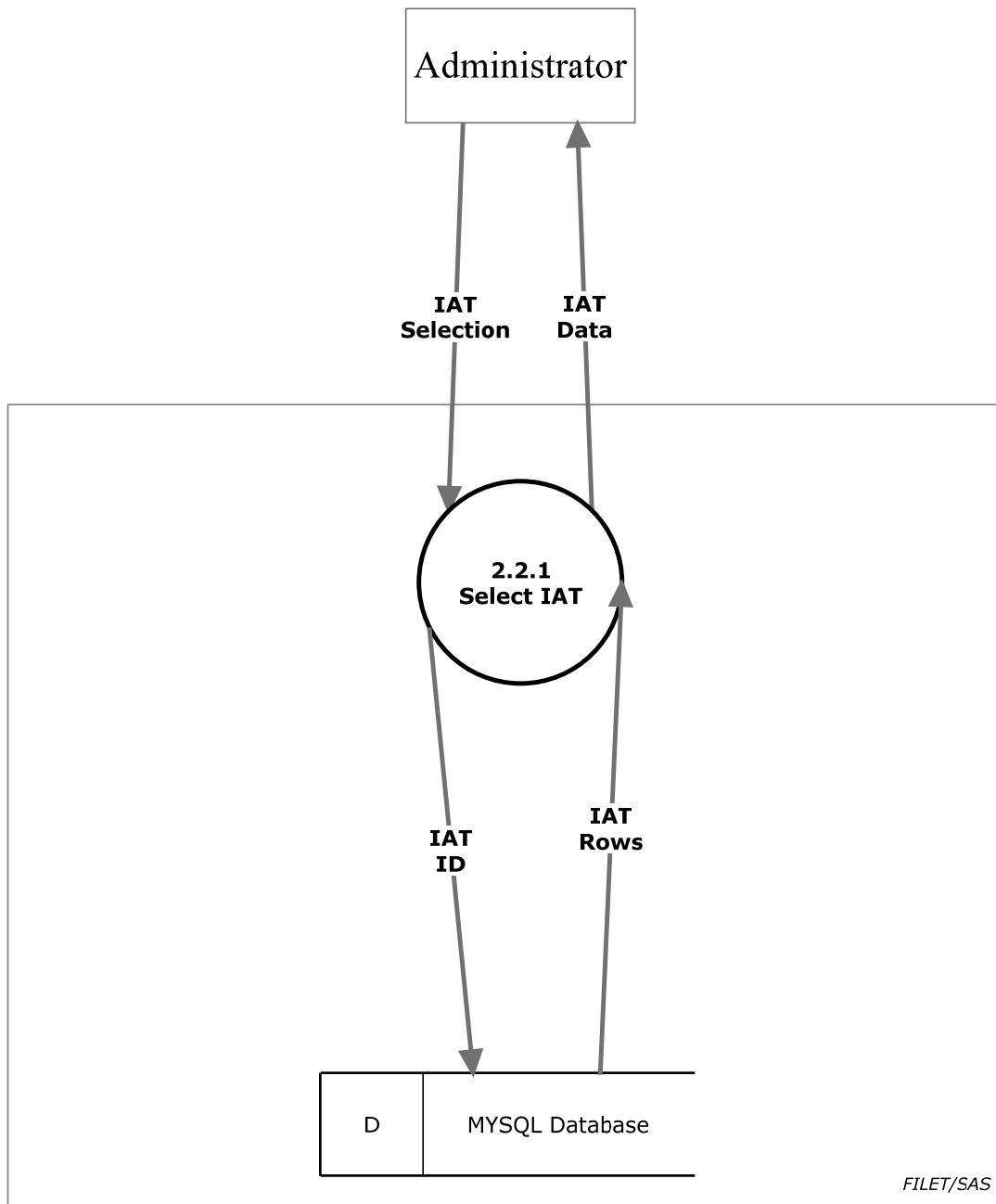




## 2.10 | Level 2: Process 2.2 Manage Data

The Administrator selects an IAT for data management

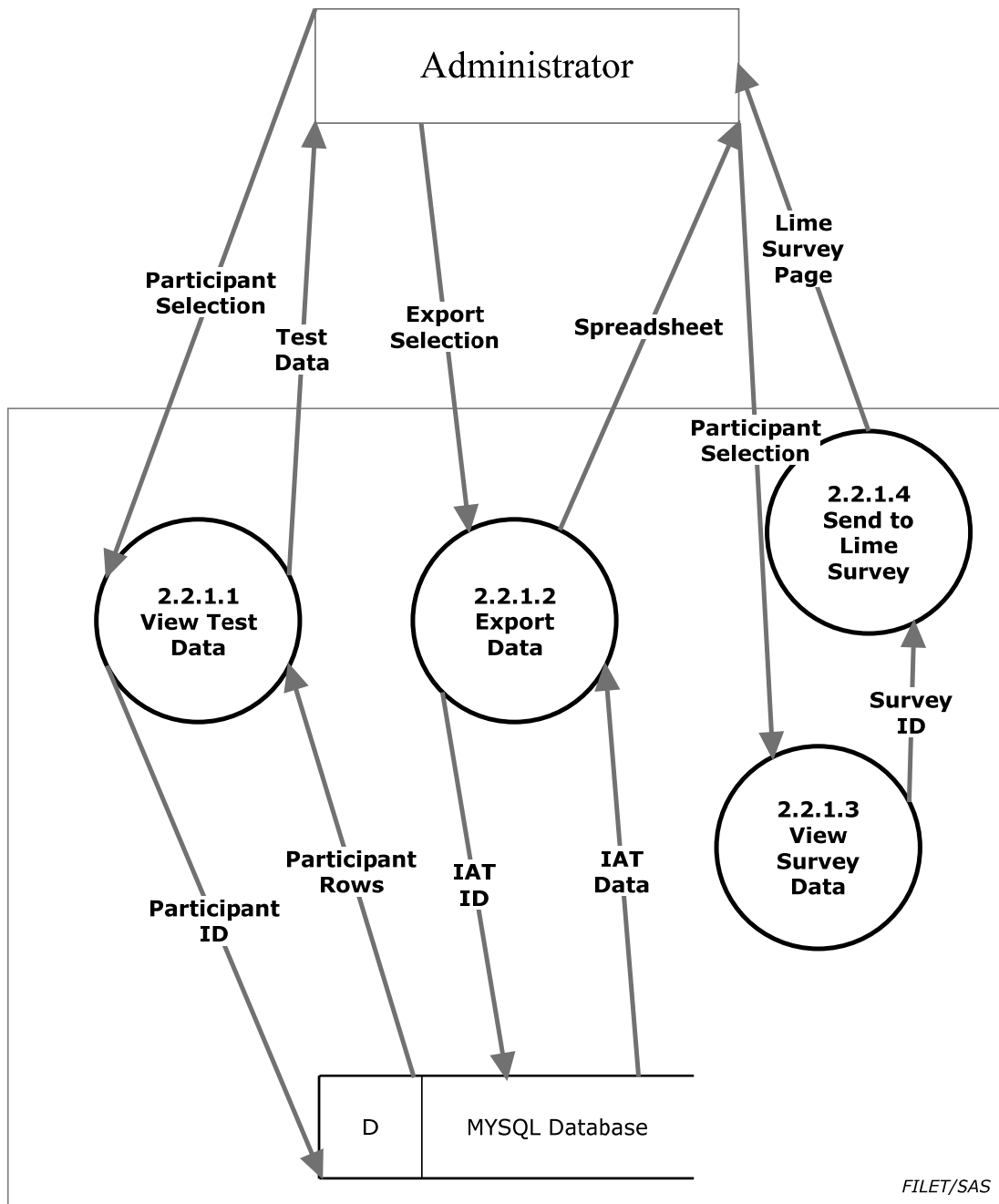
### FILET/SAS Level 2: Process 2.2 Manage Data



## 2.11 | Level 3: Process 2.2.1 Select IAT

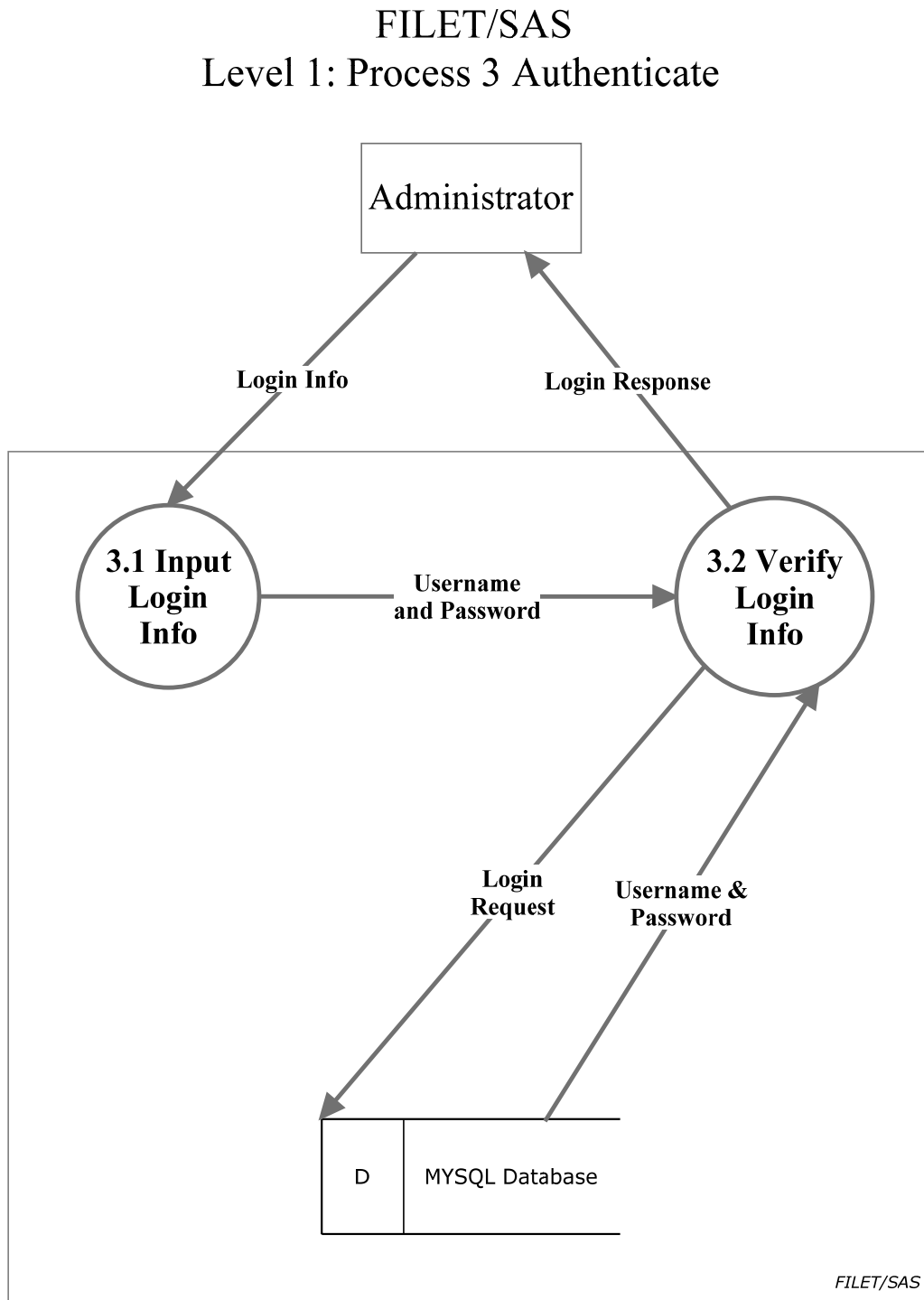
Once an IAT is selected the Administrator can view or export the test or survey data.

### FILET/SAS Level 3: Process 2.2.1 Select IAT



## 2.12 | Level 1: Process 3 Authenticate

The Administrator can log in with his unique username and password.



# Acceptance Test

## Appendix C: Source Code

### Factual Inter Leanings that Examine Throughput (F.I.L.E.T.)

#### **Requested By:**

Dr. Eric Breimer  
Associate Professor of Computer Science  
Siena College

#### **Delivered By:**

FSH Tech

#### **Prepared By:**

Jacquelyn Boylan  
Matthew Brancato  
Matthew Kemmer  
Serena Moore  
Nydia Negrón  
Mike Tanski

April 25, 2012

# F.I.L.E.T.

## Acceptance Test

### Appendix C: Source Code

### Table of Contents

## 1 | Source Code

1.1   index.php.....	3
1.2   viewiatdata.php.....	6
1.3   viewiattestdata.php.....	10
1.4   IATScript.php.....	13
1.5   scoringalgorithm.php.....	14
1.6   main.php.....	22

# 1 | Source Code

## 1.1 | index.php

```
<?
session_start();

include 'functions.php';

$username = $_POST['username'];
$password = $_POST['password'];
$action = $_POST['action'];

$message = 'Please Log In to Continue';

if($action == "Logout")
{
    $_SESSION['loggedIn'] = false;
    session_destroy();
}

else if ($action == "Login" && $username != "" && $password != "")
{
    $result = runQuery("SELECT password FROM Login WHERE username =
'$username'");
    $row = mysql_fetch_array($result);
    $actual_password = $row['password'];
    if ($password == $actual_password)
    {
        $_SESSION['loggedIn'] = true;
    }
    else
    {
        $message = "Incorrect Login Combination";
    }
}
?>

<!DOCTYPE HTML>
<html>

<head>
```

```

<meta charset="UTF-8" />
<title>FILET Home</title>
<link rel="stylesheet" href="main.css" type="text/css" />
</head>

<body>

<div id="outer_container">
<div id="content">
<br /><br /><h1>Welcome!</h1></div>
<?
$message = "Please log in to continue";
if(!$_SESSION['loggedIn'])
{
    echo '
        <div align="center"><br /><strong>!. $message .!</strong><br /><br /><form
action="index.php" method="post">
        Username: <input type="text" name="username" /><br /><br />
        Password: <input type="password" name="password" /><br /><br />
        <input type="submit" name="action" value="Login" />
        </form></div><br />';
}
else
{
    $session_id = session_id();
    echo '
        <div id = menulinks>
<span id = mlinkIAT><a
href="http://enigma.sienacs.com/createIAT.php?session_id=!. $session_id.'"> Create New
IAT </a></span></div>
        <div align="center"><strong><h2>Deployed IATs</h2></strong></p>
        </div>
        <div align="center">
        <table width="750" border="1">
        <tr>
            <td width="127" height="33"><div align="center"><strong>Average
Score</strong></div></td>
            <td width="352"><div align="center"><strong>IAT Title</strong></div></a></td>
            <td width="160"></td>
            <td width="190"></td></tr>';
    $maxresult = runQuery("SELECT MAX(IATID) FROM IATInfo");
    $maxfetch = mysql_fetch_array($maxresult);
    $max = $maxfetch['MAX(IATID)'];
    $i = 1;
}
}

```

```

while($i <= $max)
{
    $result = runQuery("SELECT * from IATInfo WHERE IATID = $i");
    $row = mysql_fetch_array($result);
    $IATTitle = $row['IATTitle'];
    $surveyID = $row['surveyID'];
    $average = 0;
    $result = runQuery("SELECT count(*) from ParticipantDetails WHERE
IATID = $i");
    $row = mysql_fetch_array($result);
    $partCount = $row['count(*)'];
    $result = runQuery("SELECT * from ParticipantDetails WHERE IATID =
$i");
    $row = mysql_fetch_array($result);
    $loop = 0;
    while($loop <= sizeof($row))
    {
        $average = $average+$row['participantAssociation'];
        $loop++;
    }
    $averageBias = $average / $partCount;
    echo '<tr><td height="50"><div align="center">' . $averageBias .
'</div></td><td><div align="center">' . $IATTitle . '</div></td><td><span id="mlink"><a
href="http://sienasellbacks.com/fsh/viewiatdata.php?IATID=' . $i . '">View Test
Data</a></span></td>';
        echo '<td><span id="mlink"><a
href="http://fsh.sienacs.com/survey/admin/admin.php?action=browse&sid=' . $surveyID .
'&subaction=all">View Survey Data</a></span></td></tr>';
    $i = $i + 1;
}

echo '
<br /></div><br /></table><br /><br />
';
}
echo '<section id="footer">
    <div id="teams">
        Developers Websites: <br/>
        <a href = "http://oraserv.cs.siena.edu/~perm_enigma/"> enigma elucidation </a> <br
/>

        <a href= "http://oraserv.cs.siena.edu/~perm_fshtech/"> FSH Tech </a> <br />
    </div>';

```



```

        if($_SESSION['loggedIn']) echo '<div id="logout"><form action="index.php" method =
"post"><input type="submit" name="action" value="Logout" /></form></div>';
        echo '</section>
</div>';
        ?>

</body>
</html>

```

## 1.2 | viewiatdata.php

```

<?
session_start();
?>

<!DOCTYPE HTML>
    <html>
    <head>
    <meta charset="utf-8" />
    <title>IAT Data</title>
    <link rel="stylesheet" href="main.css" type="text/css" />
    </head>

    <body>

    <div id="outer_container">
        <div id="content">
            <br /><br /><br />
        <?
include 'functions.php';
$IATID = $_GET['IATID'];
$message = "Please log in to continue";
if (!$_SESSION['loggedIn'])
{
    echo '</div><br /><div align="center"><strong>' . $message . '
</strong><br /><br /><form action="index.php" method="post">
Username: <input type="text" name="username" /><br /><br />
Password: <input type="password" name="password" /><br /><br />
<input type="submit" name="action" value="Login" />
</form></div>';
}

```

```

else
{
    $session_id = session_id();
    echo '

    </div><br /><br />
    <div align="center"><span id=mlinkIAT><a
href="http://enigma.sienacs.com/createIAT.php?session_id='.$session_id.'">Create New
IAT</a></span>
    <span id=mlinkIAT><a href="http://sienasellbacks.com/fsh/index.php">Return
Home</a></span>
    <br />
    <br />
    <br />
</div>
<div id="title">
    <div align="center"><br /><h2>Currently Viewing:
    ';
    $result = runQuery("SELECT * from IATInfo WHERE IATID = $IATID");
        $row = mysql_fetch_array($result);
        $IATTitle = $row['IATTitle'];
        echo $IATTitle;

    $fileName = 'IAT' . $IATID . '_basicInfo.csv';
    $header = array('PID', 'Association');
    $fp = fopen($fileName, 'w');
    fputcsv($fp, $header);
    $maxresult = runQuery("SELECT MAX(participantID) FROM ParticipantDetails
WHERE IATID = $IATID");
    $maxfetch = mysql_fetch_array($maxresult);
    $max = $maxfetch['MAX(participantID)'];
    $participants = array();
    for($i = 1; $i <= $max; $i++){
        $res = runQuery("SELECT * from ParticipantDetails P WHERE
P.participantID = $i AND P.IATID = $IATID");
        $row = mysql_fetch_array($res);
        $data = array($row['participantID'], $row['participantAssociation']);
        fputcsv($fp, $data);

        $participants[$i] = array($row['participantID'],
$row['participantAssociation']);
    }
    fclose($fp);

```

```

$fileNameAll = 'IAT' . $IATID . '_allParticipants.csv';
$header = array('PID', 'Question', 'StimuliID', 'Time(ms)', 'Correct', 'Association');
$fp = fopen($fileNameAll,'w');
fputcsv($fp, $header);

for($i = 1; $i <= $max; $i++){
    for($j = 1; $j <= 180; $j++){
        $res = runQuery("SELECT * from IATData WHERE questionNumber = $j AND
participantID = $i");
        $row = mysql_fetch_array($res);
        if($row['correct'] == "0") { $c = "NO"; }
        else { $c = "YES"; }
        $data = array($i, $j, $row['stimuliID'], $row['time'], $c, $participants[$i][1]);
        fputcsv($fp, $data);
    }
}
fclose($fp);

echo '<br /><br /></div></h2>
</div>

<div align="center">
    <p><span id=elink><a href=' . $fileName . '>Export This List</a></span>
    <span id=elink><a
href="http://fsh.sienacs.com/survey/admin/admin.php?action=exportstructure&sid=';
        $result = runQuery("SELECT * from IATInfo WHERE IATID = $IATID");
        $row = mysql_fetch_array($result);
        $surveyID = $row['surveyID'];
        echo $surveyID;
        echo "">Export All Survey Data</a></span>
    <span id=elink><a href=' . $fileNameAll . '>Export All Test Data</a></span>
</p>
</div>
<br />
<div align="center">
    <table width="710" border="1">
    <tr>
        <td width="90" height="30"><div align="center"><strong>Participant
ID</strong></div></td>
        <td width="100"><div align="center"><strong>Association
Score</strong></div></td>
        <td width="250"></td>

```

```

        <td width="270"></td>
    </tr>

';

for($i = 1; $i <= $max; $i++){
    $pid = $participants[$i][0];
    $score = $participants[$i][1];

    echo '<tr><td height="50"><div align="center">'. $pid .'</div></td>';
    echo '<td><div align="center">'. $score .'</div></td>';

    $result = runQuery("SELECT * from ParticipantDetails WHERE participantID =
$i");
    $row = mysql_fetch_array($result);
    $partSurveyID = $row['participantSurveyID'];
    echo '<td><span id="mlink"><a
href="http://sienasellbacks.com/fsh/viewiattestdata.php?pid=';
    echo $i;
    echo "'>View Participants Test Data</a></span></td>';
    echo '<td><span id="mlink"><a
href="http://fsh.sienacs.com/survey/admin/admin.php?action=browse&sid=';
    echo $surveyID;
    echo '&subaction=id&id=';
    echo $partSurveyID;
    echo "'>View Participants Survey Data</a></span></td>';
    echo '</tr>';
}

echo '</table>';

}

echo '<br /><section id="footer"><div id="teams">
Developers Websites: <br/>
<a href = "http://oraserv.cs.siena.edu/~perm_enigma/"> enigma elucidation </a> <br
/>

<a href= "http://oraserv.cs.siena.edu/~perm_fshtech/"> FSH Tech </a> <br />
</div>';
    if($_SESSION['loggedIn']) echo '<div id="logout"><form action="index.php" method =
"post"><input type="submit" name="action" value="Logout" /></form></div>';
    echo '</section>
</div>';

```

```
    ?>
</body>
</html>
```

### 1.3 | viewiattestdata.php

```
<?
session_start();

?>

<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8" />
<title>View IAT Test Data</title>
<link rel="stylesheet" href="main.css" type="text/css" />
</head>

<body>
<div id="outer_container">
<div id="content">
<br /><br /><br />
<?

include 'functions.php';
$message = "Please log in to continue";
$pid = $_GET['pid'];
if (!$_SESSION['loggedIn'])
{
    echo '</div><br /><div align="center"><strong>' . $message . '
    </strong><br /><br /><form action="index.php" method="post">
    Username: <input type="text" name="username" /><br /><br />
    Password: <input type="password" name="password" /><br /><br />
    <input type="submit" name="action" value="Login" />
    </form></div>';
}

else
{
```

```

        $result = runQuery("SELECT IATID from ParticipantDetails WHERE participantID =
$pid");
        $IATIDarray = mysql_fetch_array($result);
        $IATID = $IATIDarray['IATID'];
echo '</div>
<div id = menulinks>
<span id =mlink><a href="http://sienasellbacks.com/fsh/viewiatdata.php?IATID=';
echo $IATID;
echo "'>View All Participants</a></span>
<span id =mlink><a href="http://sienasellbacks.com/fsh/index.php">Return
Home</a></span>';

        $titleresult = runQuery("SELECT IATTtitle from IATInfo WHERE IATID = $IATID");
        $IATTtitlearray = mysql_fetch_array($titleresult);
        $IATTtitle = $IATTtitlearray['IATTtitle'];
        $scorerresult = runQuery("SELECT participantAssociation from
ParticipantDetails WHERE participantID = $pid");
        $scorearray = mysql_fetch_array($scorerresult);
        $score = $scorearray['participantAssociation'];

        $fileName = 'IAT' . $IATID . '_participant' . $pid . '.csv';
        $header = array('Question Number', 'Stimuli', 'Time (ms)', 'Correct',
'Association');
        $fp = fopen($fileName, 'w');
        fputcsv($fp, $header);
        $pData = array();
        for($i = 1; $i <= 180; $i++){
            $res = runQuery("SELECT * from IATData WHERE questionNumber = $i
AND participantID = $pid");
            $row = mysql_fetch_array($res);
            if($row['correct'] == "0") { $c = "NO"; }
            else { $c = "YES"; }
            $data = array($i, $row['stimuliID'], $row['time'], $c, $score);
            fputcsv($fp, $data);
            $var = $row['stimuliID'];
            $test = runQuery("SELECT * from StimuliObjects WHERE stimuliID =
$var");

            $testRow = mysql_fetch_array($test);
            $pData[$i] = array($testRow['value'], $row['time'], $c);
        }
        fclose($fp);

echo '</div><h1>You are currently viewing IAT ' . $IATTtitle . '</a></h1><br />

```

```

<span id=icon1><span id=pid>Participant: ' . $pid . '</a></span></span>
<span id=icon2><span id=bias>Association: ' . $score . '</span></span>
<br /><br /><br /><br />
<div id = menulinks><span id = elink><a href=' . $fileName . '>Export This
Data</a></span></div>
<br />
<div align="center">
  <table width="600" border="1">
    <tr>
      <td width="107"><h3><div align="center"><strong>Question
Number</strong></div></h3></td>
      <td width="103"><h3><div align="center"><strong>Stimuli</strong></div></h3></td>
      <td width="210"><h3><div align="center"><strong>Reaction Time
(milliseonds)</strong></div></h3></td>
      <td width="152"><h3><div align="center"><strong>Correct</strong></div></h3></td>
    </tr>;

    for($i = 1; $i <= 180; $i++)
    {
      echo'<tr><td><div align="center">'. $i . '</div></td>';
      echo '<td><div align="center">'. $pData[$i][0] . '</div></td>';
      echo '<td><div align="center">'. $pData[$i][1] . '</div></td>';
      echo '<td><div align="center">'. $pData[$i][2] . '</div></td>';
      echo '</tr>';
    }

    echo '</table>';

  }

echo '</div>
<br /><section id="footer"><div id="teams">
  Developers Websites: <br/>
  <a href = "http://oraserv.cs.siena.edu/~perm_enigma/"> enigma elucidation </a> <br
/>

  <a href= "http://oraserv.cs.siena.edu/~perm_fshtech/"> FSH Tech </a> <br />
</div>';
  if($_SESSION['loggedIn']) echo '<div id="logout"><form action="index.php" method =
"post"><input type="submit" name="action" value="Logout" /></form></div>';
  echo '</section></div>';
?>
</body>

```

</html>

## 1.4 | IATScript.php

<?

```
include 'functions.php';
include 'main.php';
```

```
//Parse variable and insert into DB
$resultStr = $_POST['resultStr'];
$dataArray = explode(",", $resultStr);
$password = $dataArray[0];
if($password == "fishenigma12")
{
    $maxquery = runQuery("SELECT MAX(participantID) FROM ParticipantDetails");
    $maxans = mysql_fetch_array($maxquery);
    $max = $maxans['MAX(participantID)'];
    $participantID = $max + 1;

    $iatID = $dataArray[1];
    $participantSurveyID = $dataArray[2];
    $questionNumber = 1;

    $i = 3;
    for($j = 0; $j < 180; $j = $j + 1)
    {
        $i1 = $dataArray[$i];
        $i = $i + 1;
        $i2 = $dataArray[$i];
        $i = $i + 1;
        $i3 = $dataArray[$i];
        $i = $i + 1;
        $query = "INSERT INTO IATData (QuestionNumber, ParticipantID, StimuliID, Time,
Correct) VALUES ('$questionNumber', '$participantID', '$i1', '$i2', '$i3)";
        runQuery($query);
        $questionNumber = $questionNumber+1;
    }
}
```



```

    $score = main($resultStr);
    echo "Score = ".$score;

    $query = "INSERT INTO ParticipantDetails(participantID, participantSurveyID,
participantAssociation, IATID) VALUES ($participantID, $participantSurveyID, $score, $iatID)";

    runQuery($query);
}

?>

```

## 1.5 | ScoringAlgorithm.php

```

<?

function ScoringAlgorithm($block3, $block4, $block6, $block7)
{

    $NUM_QUESTIONS = 180;

    $NUM_COUNTED = 120;

    //Compute means; eliminate high/low time trials

    //first index stores mean, second index stores number of values < .300

    $mean3 = computeMean($block3);

    $mean4 = computeMean($block4);

    $mean6 = computeMean($block6);

    $mean7 = computeMean($block7);

    //stop computing if 10% of trials are less than 300ms

    $mean = $mean3[1] + $mean4[1] + $mean6[1] + $mean7[1];

```

```

    if(( $mean / $NUM_COUNTED ) >= 0.1){
return -2;
}

//compute SDs of each block for correct trials
$sD3 = standardDeviation($block3, $mean3[1]);
$sD4 = standardDeviation($block4, $mean4[1]);
$sD6 = standardDeviation($block6, $mean6[1]);
$sD7 = standardDeviation($block7, $mean7[1]);

$block36 = array();
//$block36[$block3][$block6];

$block36 = merge_array($block3,$block6);

$block47 = array();
//$block47[$block4][$block7];

$block47 = merge_array($block4,$block7);

//compute pooled SDs for the combined blocks of correct values
$temp1 = computeMean($block36[0]);
$temp2 = computeMean($block47[0]);
$pooled36 = standardDeviation($block36, $temp1);

```

```

$pooled47 = standardDeviation($block47, $temp2);

    //replace incorrect trials with replacement values
$block3 = replaceValues($block3, $mean3[0], $sD3);
$block4 = replaceValues($block4, $mean4[0], $sD4);
$block6 = replaceValues($block6, $mean6[0], $sD6);
$block7 = replaceValues($block7, $mean7[0], $sD7);

    //compute mean with replacement values
$newMean3 = computeMean($block3);
$newMean4 = computeMean($block4);
$newMean6 = computeMean($block6);
$newMean7 = computeMean($block7);

    //compute the differences of the means
$diff63 = $newMean6[0] - $newMean3[0];
$diff74 = $newMean7[0] - $newMean4[0];

    //compute the quotients of the differences and pooled SDs
$quotient63 = $diff63 / $pooled36;
$quotient74 = $diff74 / $pooled47;

```

```
        //returns the average of the quotients

        $q = $quotient63 + $quotient74;

return ($q) / 2;

}
```

```
function computeMean($block)

{

    $TOO_SHORT1 = 300;

    $TOO_SHORT2 = 400;

    $TOO_LONG = 10000;

    $NUM_QUESTIONS = 180;

    $numSmall = 0;

    $totalLatency = 0;

    $correctLatency = 1;

    printf($block);

    for($i = 0; $i < sizeof($block); $i++){

        //record number of latencies < 300ms

        if($block[$i][0] < $TOO_SHORT1){

            $numSmall++;

        }

    }

}
```

```

//eliminate trials with latencies greater than 10000ms and less than
400ms
if($block[$i][0] > $TOO_LONG || $block[$i][0] < $TOO_SHORT2){
    $block[$i][0] = -1;
}

else{
    //Record latency if correct answer
    if($block[$i][1] == 1){
        $totalLatency += $block[$i][0];
        $correctLatency++;
    }

    //Eliminate trial if incorrect answer
    if($block[$i][1] == 0){
        $block[$i][0] = -1;
    }
}

}

$mean;

if($correctLatency > 0){
    $mean = $totalLatency / $correctLatency;
}

else{

```

```

    $mean = 0;
}

    $returnArray = array();
    $returnArray[0] = $mean;
    $returnArray[1] = $numSmall;

return $returnArray;
}

function standardDeviation($vals, $mean)
{
    $stdDev = 0;

for($i = 0; $i < sizeof($vals); $i++){
    if($vals[$i][0] >= 0){
        $t = $vals[$i][0] - $mean;

        $stdDev += $t * $t;
    }
}

    $stdDev = $stdDev / sizeof($vals);

```

```

return sqrt($stdDev);
}

function combineArrays($a1, $a2)
{
    $retA = array();

    $size = sizeof($a1) + sizeof($a2);

    $retA[$size][2];

    $j = 0;

    for($i = 0; i < sizeof($a1); $i++){

        $retA[$j][0] = $a1[$i][0];

        $retA[$j][1] = $a1[$i][1];

        $j++;

    }

    for($i = 0; $i < sizeof($a2); $i++){

        $retA[$j][0] = $a2[$i][0];

        $retA[$j][1] = $a2[$i][1];

        $j++;
    }
}

```

```
}

return $retA;

}

function replaceValues($block, $mean, $stdDev)
{

for($i = 0; $i < sizeof($block); $i++){
    if($block[$i][0] == -1){
        $stdDev = $stdDev * 2;
        $block[$i][0] = $mean + $stdDev;
    }
}

return $block;

}

?>
```



## 1.6 | main.php

```
<?

include 'ScoringAlgorithm.php';

function main($args) {

    $NUM_IGNORE = 3;

    $NUM_PER_RECORD = 3;

    $BLOCK3START = $NUM_PER_RECORD + $NUM_IGNORE;

    $BLOCK3START = $BLOCK3START * 40;

    $BLOCK4START = $NUM_PER_RECORD + $NUM_IGNORE;

    $BLOCK4START = $BLOCK4START * 60;

    $BLOCK4END = $NUM_PER_RECORD + $NUM_IGNORE;

    $BLOCK4END = $BLOCK4END * 100;

    $BLOCK6START = $NUM_PER_RECORD + $NUM_IGNORE;

    $BLOCK6START = $BLOCK6START * 120;

    $BLOCK7START = $NUM_PER_RECORD + $NUM_IGNORE;

    $BLOCK7START = $BLOCK7START * 140;

    $BLOCK7END = $NUM_PER_RECORD + $NUM_IGNORE;

    $BLOCK7END = $BLOCK7END * 180;

    $data = explode(",", $args);
```

```
$block3 = array();
```

```
$block4 = array();
```

```
$block6 = array();
```

```
$block7 = array();
```

```
$j = 0;
```

```
for($i = $BLOCK3START; $i < $BLOCK4START; $i = $i + 3){
```

```
    $block3[$j][0] = $data[$i + 1];
```

```
        $j = $j + 1;
```

```
    $block3[$j][1] = $data[$i + 2];
```

```
}
```

```
$j = 0;
```

```
for($i = $BLOCK4START; $i < $BLOCK4END; $i = $i + 3){
```

```
    $block4[$j][0] = $data[$i + 1];
```

```
        $j = $j + 1;
```

```
    $block4[$j][1] = $data[$i + 2];
```

```
}
```

```
$j = 0;
```

```
for($i = $BLOCK6START; $i < $BLOCK7START; $i = $i + 3){
```

```
$block6[$j][0] = $data[$i + 1];  
    $j = $j + 1;  
$block6[$j][1] = $data[$i + 2];  
}
```

```
$j = 0;  
for($i = $BLOCK7START; $i < $BLOCK7END; $i = $i + 3){  
    $block7[$j][0] = $data[$i + 1];  
        $j = $j + 1;  
    $block7[$j][1] = $data[$i + 2];  
}
```

```
$answer = ScoringAlgorithm($block3, $block4, $block6, $block7);
```

```
return $answer;
```

```
}
```

```
?>
```