

Software Requirements Specification

Sensory Evaluation Testing System

Frank Senseney

CS 480 Advanced Software Engineering

Professor Szilárd Vajda

November 16, 2017

Table of Contents

Section	Page
1. Introduction.....	3
1.1. Purpose.....	3
1.2. Scope.....	3
1.3. Definitions, Acronyms, and Abbreviations.....	3
1.4. References.....	4
2. Overall description.....	4
2.1. Product Perspective.....	4
2.2. Product Functions.....	4
2.3. User Characteristics.....	5
2.4. Constraints.....	5
3. Specific requirements.....	5
3.1 External Interfaces.....	6
3.1.1 User Interfaces.....	6
3.1.2 Hardware Interfaces.....	7
3.2 Functional Requirements.....	8
3.2.1 User class: Administrators and Test Technicians.....	8
3.2.2 User class: Sensory Panel Judge.....	10
3.3 Software System attributes.....	10
3.5.1 Reliability and Availability.....	10
3.5.2 Security.....	11
3.5.3 Maintainability.....	11
3.4 Non-functional Requirements.....	11
3.5 Requirements Prioritization.....	11
3.6 Feasibility Analysis.....	12
3.7 Validity and Consistency Check.....	12
Annex.....	13

1. Introduction

1.1. Purpose

The purpose of this document is to provide a detailed description of the Sensory Evaluation Testing System. It will explain the purpose and features of the system, the system's interfaces, and the constraints of the environment the system must operate within. This document's principle function is to act as a proposal to the client for approval and as a reference for the development of the first version of the software by the development team.

1.2. Scope

The Sensory Evaluation Testing System is intended to replace the current sensory panel procedure the food science class currently uses. At present all data is recorded using printed forms, which must be created by hand for each test variation (see figure 1.2 in the annex for an example). Test technicians must then manually transcribe the data, which must be randomly ordered for testing purposes, into spreadsheets so it can be used for statistical analysis. The goal of this system is to automate this process to make the job of creating test scenarios, running tests, and collecting data as easy as possible. The system would provide a straightforward user interface for test technician to create and manage tests and allow for panel judges to input their answers. The data from each test would then be collected and formatted so that it can be downloaded as an excel document file.

1.3. Definitions

Term	Definition
Administrator/Admin	A person who has been granted permission to manage and control the system.
User	A person who interacts with the software.
Test Technician	The person overseeing the sensory evaluation panel.
Judge	A person taking part in a sensory evaluation experiment.
Database	A collection of the information gathered by the system.
Sensory evaluation	Measurements determined by using the senses of sight, smell, and taste.
Client	The requesting program or user in a client/server relationship.
Server	The computer program that provides services to other computer programs (and their users) in the same or other computers.
Sensory Panel	A group of people asked to evaluate a specific attribute of a food product using a given set of criteria.
HTTP	Hyper Text Transfer Protocol. An application protocol for distributed, collaborative, and hypermedia information systems. The foundation of data communications on the internet.
TCP/IP stack	Transmission Control Protocol/ Internet Protocol stack. A conceptual model and set of communications protocols used on the Internet.

1.4. References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Overall description

2.1. Product perspective

The Sensory Evaluation Testing System is a self-contained product that doesn't not require additional specialized hardware or software to function. It provides the functionality stated in the product function section of this document.

The system will consist of two main components: a back-end interface for creating and managing sensory evaluation tests, and a front-end interface for judges to answer evaluation questions.

2.1.1. User Interfaces

The user interface should be easy to understand with no additional training. It should be compatible with any modern web browser such as Mozilla Firefox, Google Chrome, or Microsoft Edge.

2.1.2 Software interfaces

The system will use the MySQL database management system to store user account information, tests, and testing data on the server.

2.1.2. Communication interfaces

The system will use the Hypertext Transfer Protocol (or HTTP), part of the TCP/IP network model, to send and receive data from the server.

2.2. Product functions

This section defines the software's functionality. As explained in the user characteristics section, functionality differs depending on the user's permissions and role.

2.2.1 Login

This function allows users to gain access to the system. The user will be required to enter a username and password. After successful authentication, administrative users and test technician will be taken to the management screen. Panel judges will be taken directly to the tests that have been assigned to them.

2.2.2 View/manage tests

This function allows users with appropriate permissions to view, edit, or remove the sensory evaluation tests that are currently in the database. Administrative users will be able to view, edit, or remove any tests. Test technicians will be able to view, edit, or remove any tests that they created. Judges cannot access this feature.

2.2.3 Manage user accounts

This function will allow administrative users to create new test technicians and judge accounts, and remove existing ones.

2.2.4 Assign/remove judges

This function allows users to add or remove specific panel judges from tests. Administrators will be able to this for any test. Test technician will be able to this for the test they have created. Panel judges will not have access to this function.

2.3. User classes and characteristics

User class	Characteristics
Administrator	Administrators would be able to access and modify all tests currently in the system, create new tests, manage accounts, and assign tests to judges.
Test technician	Test technicians would be able to create new tests, view or modify tests they have created, and assign tests to judges.
Sensory Panel Judge	Sensory panel judges would be assigned tests by administrator or test technician users. They would only able to view and answer questions on the tests assigned to them.

2.4 Constraints

- The system will most likely be hosted on CWU servers. If this hosting method is viable, the system will be required to follow any laws, standards or policies enforced by the school.
- Because the system is based on sending and receiving data from a database, an internet connection is required for all system functionality.
- The system will use the Hypertext Transfer Protocol (or HTTP), which uses the Transfer Control Protocol (or TCP) that employs a three-way handshake to establish a connection (see figure 2.4 in the annex).
- Multiple tests can take place at the same time, so the system will need to be able to process data input from multiple sources simultaneously.

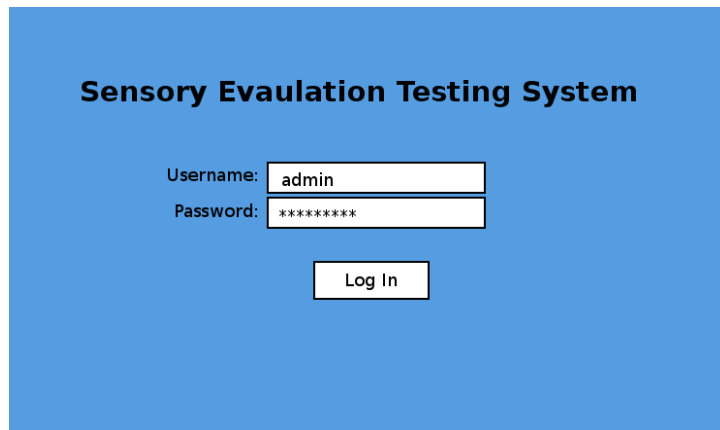
3 Specific Requirements

3.1 External Interfaces

3.1.1 User Interfaces

When a user first opens the system in their internet browser, they will see the login page (see figure 1). After successfully logging into the system, administrators and test technicians will be taken to the management page (see figure 2). Selecting the edit, copy, or 'new test' buttons will display the test creation screen (figure 3).

FIGURE 1 - LOGIN PAGE MOCKUP

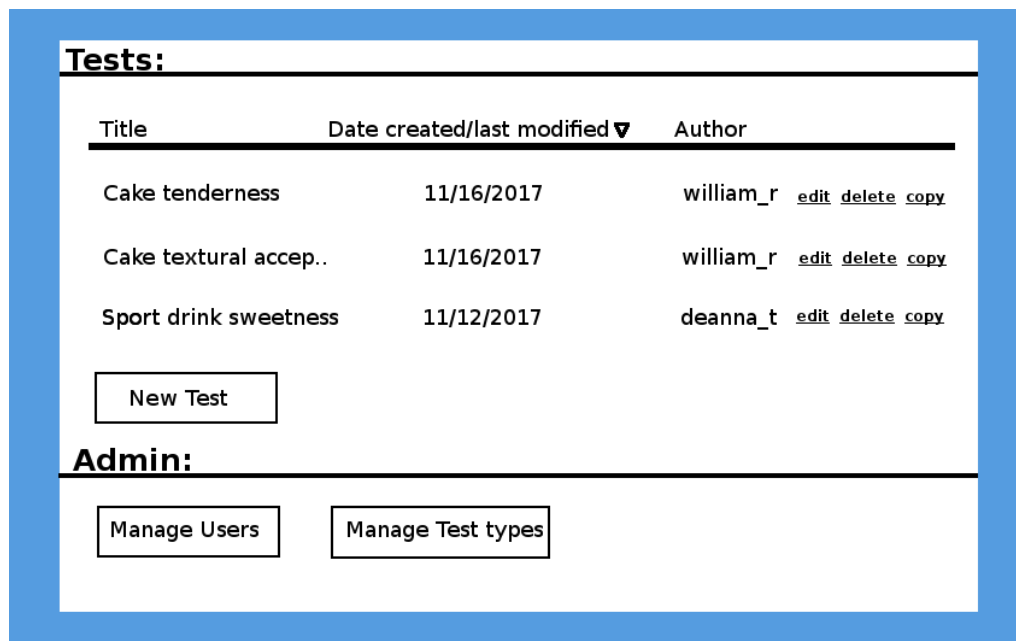


Sensory Evaluation Testing System

Username:

Password:

FIGURE 2 - MANAGEMENT PAGE MOCKUP



Tests:

Title	Date created/last modified ▼	Author
Cake tenderness	11/16/2017	william_r edit delete copy
Cake textural accep..	11/16/2017	william_r edit delete copy
Sport drink sweetness	11/12/2017	deanna_t edit delete copy

Admin:

FIGURE 3 - TEST CREATION PAGE MOCKUP

The mockup shows a test creation interface with the following components:

- Title:** A text input field containing "Cake tenderness".
- Description:** A text area containing the instruction: "Evaluate the following cake samples for tenderness using the the following unstructured scale by pressing your finger on the bars to mark your tenderness rating."
- Number of samples:** A text input field containing "6".
- Attributes:**
 - A [remove](#) link next to the attribute name.
 - Name:** A text input field containing "Tenderness".
 - Type of scale:** A dropdown menu with options: "9 point", "6 point", "Unstructured" (highlighted), and "Custom".
 - Start:** A text input field containing "Firm".
 - End:** A text input field containing "Very soft".
 - A [Add new attribute](#) link.
- Judges:**
 - Two boxes for judge names. The left box contains "Judge 1", "Judge 2", and "Judge 4". The right box contains "Judge 3".
 - Between the boxes are two arrow buttons: a right-pointing arrow (→) and a left-pointing arrow (←).
- Buttons:** "Save" and "Cancel" buttons at the bottom left.

3.1.2 Hardware Interfaces

Because the system is web-based, it doesn't have any designated hardware. The connection to the sever is controlled by the underlying operating system of the device the client is connected to the server on.

3.2 Functional Requirements

3.2.1 User class: Administrators and Test Technicians

3.2.1.1 Functional requirement 1.1

Feature: Administrative and Test Technician login

To access the system as an administrator or a test technician, the user must login via the main page.

Scenario: Successful login as administrator

If an administrator wants to login and they enter the correct information, they should be authenticated as an administrator.

Scenario: Successful login as test technician

If a test technician wants to login and they enter the correct information, they should be authenticated as a test technician.

Scenario: Failed login

If an administrative or test technician user wants to login, but enters the wrong credentials, they should be alerted that the information they entered is not correct.

3.2.1.2 Functional requirement 1.2

Feature: Manage tests

An administrator or a test technician should be able edit, copy, or remove an existing test. Administrators can manage all tests currently stored in the system. Test Technicians can only manage tests they have created. When creating a test, a title and description, test type, the number of samples, attributes can be specified. By default, there would be three test types available: Triangle test, duo/trio tests, and intensity tests. For each attribute a name, scale type, and scale description can be added. The options for attribute scale are: 9 points, 6 points, unstructured, and custom.

Scenario: Admin adds new test

If an administrative user adds a new test, a new test should be created that is visible to all users.

Scenario: Technician adds new test

If an test technician adds a new test, a new test should only be visible to the that user.

Scenario: Admin or technician removes test

If an administrative user removes a test, the test should be removed for all users.

Scenario: Admin or Technician edits test

If an administrator user edits a test, the test should be updated for all users.

Scenario: Admin or Technician copies test

If an administrator or test technician copies a test, the test should be duplicated with the same parameters but no testing data.

3.2.1.3 Functional requirement 1.3

Feature: Manage users

An administrative user should be able to add, edit and remove test types

Scenario: Add new test technician

If an administrative user adds a new test technician user, the administrator should be able to define their username, full name, class section, and password.

Scenario: Add new judge user

If an administrative user adds a new judge user, the judge should be numbered sequentially in accordance to existing judges.

Scenario: Remove user

If an administrative user removes a user, that user should be removed from the system. If the user is a test technician, any tests they have created should also be removed. If the user is a judge, they should be removed from any tests they were assigned.

Scenario: Edit user

If an administrative user edits a user, they should be able to change their username, password, or class section.

3.2.1.4 Functional requirement 1.4

Feature: Manage test types

An administrative user should be able to edit, remove, or edit test types.

By default, there are three test types: Triangle test, duo/trio tests, and intensity tests.

Scenario: Add type

If an administrative user adds a test template, it should be added to the list of available templates on the test creation page.

Scenario: Edit existing type

If an administrative user edits an existing test template, the list of available types on the test creation page should be updated to reflect the change.

Scenario: Remove existing template

If an administrative user removes an existing test template, it should be removed from the list of available types on the test creation page.

3.2.2 User class: Sensory Panel Judge

3.2.2.1 Functional requirement 3.1

Feature: Judge login

To take a test, a judge user must login via the main page.

Scenario: Successful login

If a judge user wants to login and they enter the correct information, they should be authenticated as a judge.

Scenario: Failed login

If a judge user wants to login, but enters the wrong credentials, they should be alerted that the information they entered is not correct.

3.2.2.2 Functional requirement 3.2

Feature: Take a test

A judge can take any tests assigned to them.

Scenario: No tests have been assigned, or all have completed

If a judge has no tests assigned, a message will appear informing them that no tests have been assigned to them.

Scenario: One test has been assigned

If a judge has only one test assigned, they will be taken directly to the test after logging in. After completing the test, a message will appear informing them that they have completed all assigned tests.

Scenario: Multiple tests assigned

If a judge user has been assigned to multiple tests they have not completed, after logging in they will see a list of all available tests. Upon completing a test, it will be removed from the list.

3.3 Software System attributes

3.3.1 Reliability and Availability

- Not accounting for network outages, the system should be connected to the internet so clients can send and receive data from the server.
- When the system is running, it should be available for use at a minimum of 98% of the time.

3.3.2 Security

- Account login information sent between the system and the server should be encrypted so the information can't be intercepted by a third party.
- If a user tries to login with a non-existing username, they should be alerted that the information they entered is incorrect.
- If a user enters an existing username but an incorrect password to many times, they should be prompted to request a password change.
- All database inputs should be sanitized to prevent a potential loss of information (see figure 3.5 in Annex).
- Entering incorrect information should never result in a login.

3.3.3 Maintainability

- The system should be programmed and documented in such a way that future expansion and modification is possible.

3.4 Non-functional Requirements

3.4.1 Non-functional Requirement 1

Feature: Assigned random sample IDs

A random sample IDs is generated for all the possible permutations of the number of samples. For example, a test with 4 samples would have 24 possible IDs.

3.4.2 Non-functional Requirement 2

Feature: How to guide

The software includes a page with step-by-step directions to help new users create and manage tests.

3.4.3 Non-functional Requirement 3

Feature: Detect statistical significance

For triangle tests (where 1 of 3 samples is different), and duo/trio tests (where 1 of 3 samples is a reference) the system should track how many judges correctly select the right sample. If for example, 50% of judges select the correct sample, selecting the right sample is not random.

3.4.4 Non-functional Requirement 4

Feature: Printable list of IDs

The software can generate a printable list of the randomized IDs for a given test.

3.5 Requirements Prioritization.

1. Software can record judge scores in database. - Essential for the system to function.
2. Database to store user information, tests, and test results. - Needed to for most if not all functionality.
3. Software can generate spreadsheet from test data. - The main goal of the system, all other requirements build to this.
4. Software has login page, test management page and test creation page. – Essential for the management and creation of tests.

5. Software supports three main types of tests: Triangle tests, Duo/trio tests, and intensity tests. - Needed for most test scenarios.
6. Software supports 9 points, 6 points, continuous/unstructured, and custom scales for attributes. – Essential for almost all tests.
7. Software allows for users to add descriptive terms to as many of the values on the scales as they want.
8. Software supports touch input. - Needed so judges can input scores from tablet pcs.
9. Software can generate a printable list of randomly generated ids for tests – Would be helpful for students, but is lower priority because it's not required for the system to function.
10. Software can detect statistical significance on triangle and duo/trio tests.
11. Custom test types - Lower priority than predefined test types, can be implemented if time allows.
12. How to guide for students - We be helpful, but is not a priority, can be implemented if time allows.

3.6 Feasibility Analysis

Since the system is based around using a standard database to send and receive data from client computers, it should be very similar in implementation time and difficulty to comparable systems. The development team has experience with databases and back-end development, so there should not be significant problems implementing the system's testing and management system. No matter what hardware is chosen, there shouldn't be any issues with hardware compatibility because the system is browser based.

3.7 Validity and Consistency Check

- Requirements do not violate domain constraints.
- Requirements are stated clearly and concisely.
- Requirements are stated in quantitated terms when necessary.
- All requirements can be testable.
- All requirements are traceable to the overall product objectives.
- There are not conflicts between requirements.

Annex

FIGURE 1.2 – EXAMPLE OF DOCUMENTS CURRENTED USED FOR EVALUATION

[illegible]

FIGURE 2.4: THE TCP/IP PROTOCOL STACK

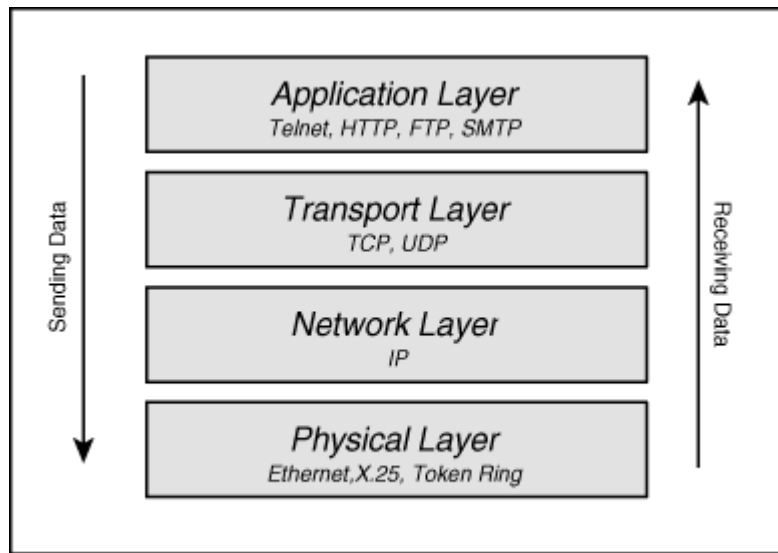


FIGURE 3.4 - THE DANGERS OF NOT SANITIZING DATABASE INPUTS

