CIS11 Course Project Part 1: Documenting the Project

Fill in the following areas (purple).

Introduction

1.1 Purpose

This document details the functionality and requirements.

1.2 Intended Audience and Users

The primary audience/user will be students, teachers, and parents who are interested in viewing easily detectable data based on various test scores.

1.3 Product Scope

What is the intention of this program?
The Test Score Calculator aims to give students and parents important figures and statistics about the student's test scores to better visualize their progress in the course. It provides the user with the most relevant information that may push them to improve their study habits or maintain them.

Example: The Curricular Information System is intended initially to replace existing Clipper and Web proposal systems with a new, integrated, web-based system having the features detailed in this document. It provides a foundation that facilitates orderly growth of future enhancements.

1.4 Reference

Source Documents for the Program Requirements and Specification

Reference Project requirements and LC-3 specifications.

LC-3 Simulator Download (Windows)

Browser based LC-3 Simulator (For incompatible Operating Systems)

How to use LC-3 Simulator

Example

1) Curricular Information System Vision and Scope
   http://web.mit.edu/ssit/cis/CISVisionScope.html
2) HTML/Print Generator and Document Management Sub-Project Vision and Scope
3) CIS Project Participants
   http://web.mit.edu/ssit/cis/CISpeople.html

4) Catalog Production Schedule Timeline
5) CSB and Registration Timeline
6) Old Catalog Data description (Clipper system, web proposal system, MITSIS)
7) Scheduling system Class Schedule Book (CSB) File layout and description
8) Old Catalog Functions (spreadsheet)
9) CSB and Registration Program documentation
10) Formal Department and Academic Services Interview Documents (detailed list available)
11) Final Exam Recommendation Draft Document

Companion Application Requirements Documents (If applicable)

What other documents should be reviewed with this document?

Example:

1) CISData.doc
2) CISMigrationRefresh.doc
3) CISWorkflow.doc
4) CIS Process Model Diagrams (These require the SilverRun BPM viewer)
5) Scheduling system Class Schedule Book (CSB) File layout and description

2. Overall Description

2.1 Product Perspective

Primary program objectives

This CIS provides:
an LC-3 program that displays the *minimum, maximum and average grade* of 5 test scores and displays the letter grade associated with the test scores.
- Help students determine their grades in a class.
- Help teachers understand class averages.
- Add-on for a class website/grading platform.

Example:
The CIS provides:
A department coordinator 'portal' for Academic Services including:
- Complete proposal development functionality (similar to IAP proposals, but more extensive)
- A home for broadcast and individual communications between Academic Services and departments
- Links to related Academic Services web applications.

Centralized Academic Services administrative control functions
Centralized Programmer control functions

2.2 Product Functions

The overall description of functionality:

Highlight the program functionality: Identify tasks and subtasks of the program in summary.

1. Find and display the highest test score.
2. Find and display the lowest test score.
3. Calculate and display the average of 5 given test scores
4. Find letter grade equivalent for outputs
5. Provide useful data to help improve or maintain study habits
6. Identify weak areas and strengths with regards to test taking which can be useful before harder exams in the future
7. Allow parents and teachers to monitor student progress

Example:

1. Provide facilities to enhance the exchange of information among faculty and staff during curriculum development. Do so by enabling distribution of official information with ancillary discussion among authorized faculty members, staff, and faculty committees during all phases of subject proposal development and review, including prior to proposal submission to the COC/CGSP.
2. Preserve a record of these decisions and their context.
3. Support versioning and workflow management of the information that it maintains.
4. Replace the current catalog production system, in which departments submit subject listing changes both electronically and on paper and curricular changes on paper, with a fully electronic system. (However, printed listings will still be obtainable upon request.)
5. Enable updating of catalog data throughout the year. Do so for more than one term/year simultaneously.
6. Provide up-to-date information about subjects, schedules, and instructors to the MIT community (faculty, academic staff, students, alumni, and prospective students).
7. Provide easy-to-use, on-demand print and on-line publishing. Non-subject data now printed in the MIT bulletin will be integrated via the web with subject data for integrated publishing.

Technical functionality

A configurable toolkit of functions including:

What are the technical functions of the program? Subroutines and operations.

1. Subroutines
2. Finding the highest, lowest, and average from user inputted test scores

Example:

- Ability to define new fields to capture for certain types of data (extensibility)
- Ability to configure fields, their sequencing, and formatting (i.e. style tags) for downloads so that any type of publication (print or web) can be downloaded without specialized programming.
- Flexible form generation including user-configurable field layout, text descriptors.

Reusable components for most functionality.

Use of Java and the new SSIT Internet platform, and when appropriate, XML

2.3 User Classes and Characteristics

Who is involved in this development process? Include business and technical personnel and their tasks.

David Gonzalez

- Testing

-Pseudocode

-Creating the Program

-Writing Documentation

-Debugging

Christopher Diaz

-Testing

-Creating the program

-Pseudocode

-Writing Documentation

- Debugging

Marlon Jimenez

- Flow Chart

-Planning

-Pseudocode

-Editing Documentation

Example:

Academic Services personnel

Responsible for the overall tracking and publishing of the MIT catalog.
Support the development of new and changed subject proposals.
Support COC and CGSP review of new and changed proposals.
Pre-register and register students. Manage Add and Drop requests.
Schedule classrooms, students, and finals.
Manage and report on pre-requisites, co-requisites.
Audit student degrees (GIR)

Department Coordinators

Responsible for helping faculty develop MIT catalog and related information for their department.
Monitor departmental roadmaps
Help develop room schedules for subjects and exams
Audit department degrees

COC and CGSP

Review subject proposals

Other Administrative Offices

The HASS Office, PSB, Communications Office review and support the development of the MIT catalog and supplemental bulletins.
Run student lotteries.
Submit grades.

Faculty

Plan and teach curricula
Use many reports provided by Academic Services: class lists, etc.

Students

Use catalog and related information to plan course work.
Use the on-line planning worksheet, lottery submittal, and pre-registration functions.

2.4 Operating Environment

What type of system will the application be operated on? Operating system? System types? Development platform?

LC-3 Simulator
LC-3 Editor
LC-3 Assembly Language

Example:
CIS is developed for use on the Unix system: student
Under the Netscape web server: enterprise using SSL and personal certificates
Accessing the Oracle database (currently on system sisjajp, but soon to be on a new database server)
New code will be developed in Java.
The existing General Table Maintenance (GTM) facility may be used for certain features.

2.5 Design and Implementation Constraints

Note any constraints or limitations to the application.

User Constraints:

1. Specifically making the program for only 5 scores.

2. Needs to install LC-3 simulator.

3. Not very user friendly when compared to a program created with a higher level programming language.

Developer Constraints:

1. Creating In LC-3 is much longer.
2. More tedious/complicated to create in LC-3.

Example: Access to the web is required. As for the developer constraints, the alumni information was not available for security reasons. Many assumptions about the data had to be made. There is a high learning curve.

2.6 Assumptions and Dependencies

Note any dependencies

1. Dependent on the user having a specific simulator installed.
2. Assumed that the user knows how to run an .asm file.
3. Assumed that the user knows to open the console.

Example:

It is assumed that alumni data will be made available for the project in some phase of its completion. Until test data will be used for providing the demo for the presentations. It is assumed that the user is familiar with an internet browser and also familiar with handling the keyboard and mouse.

Since the application is a web based application there is a need for the internet browser. It will be assumed that the users will possess decent internet connectivity.

3. External Interface Requirements

3.1 User Interfaces

How will the user interface with your program? Menus? Access prompt? Links? Icons?

1. Users will interact with the LC-3 console and Simulator.
2. Open the .obj file in the simulator so the program/console runs.

Example: The member has to register using a form provided on the website. The user can input data with the help of the keyboard or click with the mouse wherever necessary. The package provides pull down menus from which the user can select links and icons to navigate among the web pages.

3.2 Hardware Interfaces

Specify hardware interface – computer types? Terminal types?

1. Any computer running Windows OS.
2. Monitor
3. Keyboard
4. Mouse

3.3 Software Interfaces

Specify additional software interface – if any. What type of software will the application require to run?

1. LC-3 Simulator

3.4 Communications Interface

Does your application require web, Internet or network connectivity? If so, which browser? What type of network connection?

1. Download LC-3 simulator from the internet.

4. Detailed Description of Functional requirements

4.1 Type of Requirement (summarize from Section 2.2)

What are the functions? Their purposes? Inputs? Outputs? Data? Where is the data stored (internal or external to the application)?

Subroutines:

Lowest Score

- **Input: Gathered from user input**
- **Output: console window**
- **Found through sorting/searching**
- **Data: stored internally**
- **Purpose: To find the lowest value from the user inputs**

Highest Score

- **Input: Gathered from user input**
- **Output: console window**
- **Found through sorting/searching**
- **Data: stored internally**
- **Purpose: To find the highest value from the user inputs**

Average Score

- **Input: Gathered from user input**
- **Calculated with 5 user inputs.**
- **Output: console window**
- **Data: stored internally**
- **Purpose: To find the average of all the scores inputted**

Score to Letter grade conversion

- **Input: Gathered from user input and ASCII converted**
- **Output: console window**
- **Data: stored internally**
- **Purpose: To convert the outputs of the previous subroutines into their letter grade equivalent**

Example: Login Requirement
Purpose: Provides member authentication

Inputs: Inputs are through the keyboard and mouse clicks.

Processing: The input is verified by checking if the member already exists in the database.

Outputs: The correct input will result in the next page i.e the analysis page being loaded. If the input is incorrect then an error message will be displayed.

Data: User database

Registration Form Requirement

Purpose: Registration of a non-member.

Inputs: Inputs are through the keyboard and mouse clicks.

Processing: The input is validated using client side as well as server side validation. The client side validation will include checks for missing information in the required fields and other text fields like email and phone numbers will be checked for validity. The server side validation will involve checking if the username entered is already used by a member in the database. The appropriate error messages are displayed if the input is not acceptable

Outputs: The member is directed to the main page on successful registration.

Data: Client database

Analysis Requirement

Purpose: The research question is selected to perform analysis like regression.

Inputs: Input will be the research question selected by the user and consequently the data that the user wants to use for the analysis.

Processing: Depending on the research question, the appropriate statistical analysis is performed with the help of the EJB which provide the middle layer in this three tier application. It can be regression analysis, correlation, hypothesis test or the chi square test.
If an invalid input is entered, there will be appropriate error messages handled by using java exceptions in the java programs (EJB and Servlets). The HTML will include java scripts to handle error checks at the client side. Thus both client-side and server-side errors and exceptions will be handled completely within the application. The SAT will also undergo rigorous testing with various inputs to check whether the analysis is being conducted correctly and that all invalid inputs are not accepted.

Outputs: The output will be a graph or table of the analysis results displayed on the web browser page.

Data: Report database

4.2 Performance requirements

What is the expected performance level of the program?

1. **Program should be able to complete all tasks.**
2. **Program should include error handling.**
3. **Program should Display test scores converted to letter grade.**

4.2.1 The application should be portable and possible to users of Netscape Navigator as well as Internet Explorer.

4.2.2 Since the application will be displaying graphs for the analysis, the response time for a particular analysis should be not be greater than 3-4 seconds for a respectable internet connection speed.

4.2.3 The database should be scalable; it must have the capacity to hold a large number of users in future.

4.2.4 Error handling should be implemented and the application should be able to handle all runtime errors.

4.3 Flow Chart and Pseudocode.

```
; CIS-11 GROUP PROJECT
; AUTHORS: David Gonzalez, Christopher Diaz, Marlon Jimenez
```

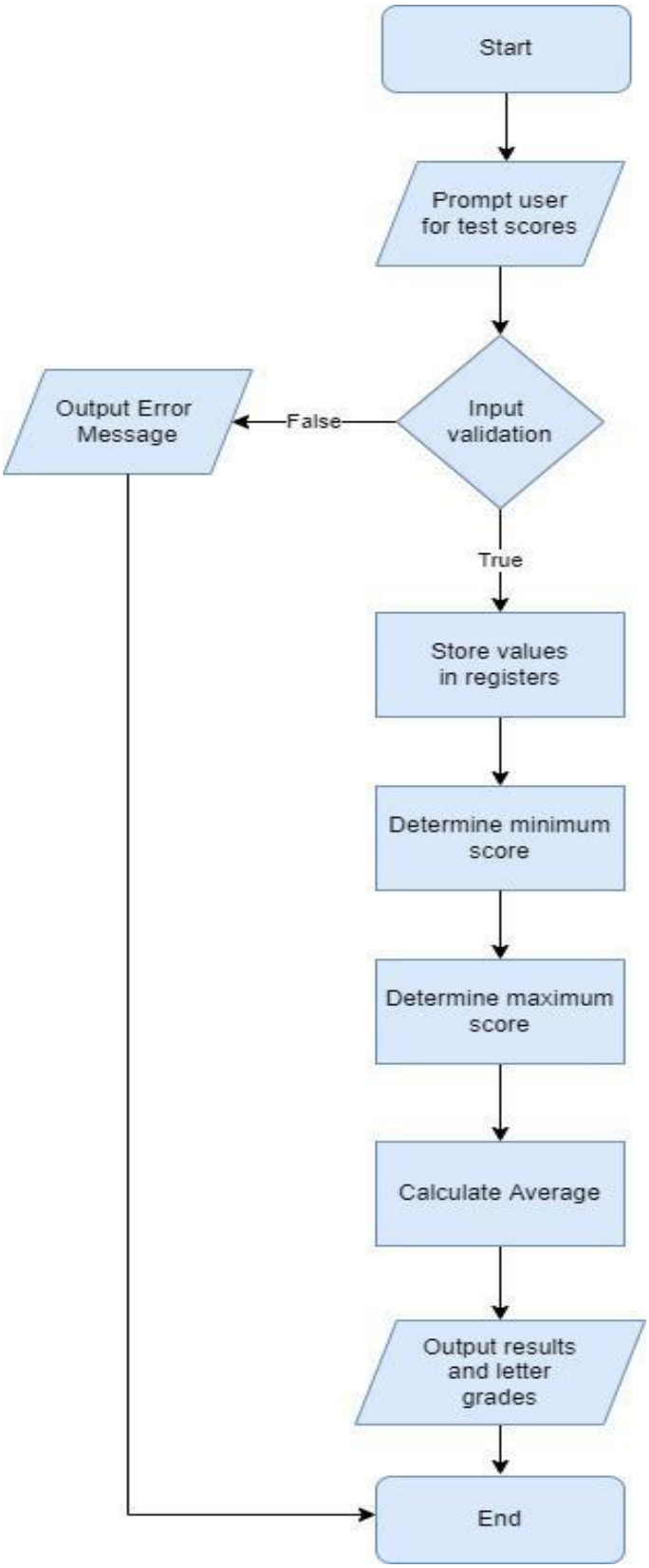
Set memory address to x3000
Puts to console “Please enter 5 test scores and press Enter”
(depends on method chosen; loop for 5 inputs or enter all 5 in 1 line)
Error catcher for inputs outside of parameters.
(characters, n >100, n < 0)
If input invalid, Error catcher output “The value you entered
is invalid, please enter number between 0 - 100”

store values in registers
If valid, push values to stack
Branch
Pop values in registers
sort to find the lowest value
store lowest value

sort to find the highest value
store highest value

load AVG formula subroutine and 5 values
store average value

convert resulting lowest, highest and average value to grade letter
output letter grades to console
end program


Start
Prompt user for test scores
Input validation
False
Output Error Message
True
Store values in registers
Determine minimum score
Determine maximum score
Calculate Average
Output results and letter grades
End