# CIS11 Course Project Part 1: Documenting the Project

Fill in the following areas (purple).

**Introduction**

* 1. **Purpose**

This document details the functionality and requirements.

* 1. **Intended Audience and Users**

The primary audience/user

* 1. **Product Scope**

What is the intention of this program?

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. **Reference**

**Source Documents for the Program Requirements and Specification**

Reference Project requirements and LC-3 specifications.

**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 (spread sheet)

 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

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

**The Program will:**

* Take in Input from the user for 5 test scores
* Display the minimum, maximum, and average grades from the inputs
* Provide the respective letter grades for each of the scores
  1. **Product Functions**

**The overall description of functionality:**

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

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.
8. The user will input 5 test scores to be processed
9. Each test score will be stored and processed through a subroutine that will determine the letter grade, which will be stored.
10. When all the test scores are inputted and stored, they will then be displayed
11. The display will show the minimum, maximum, and average along with the grades from the program showing the grades in order from lowest to highest.

**Technical functionality**

A configurable toolkit of functions including:

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

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

* 1. **User Classes and Characteristics**

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

**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.

**Programmers**

Model, write, build, and test the code.

-**Brandon**: Half of the documentation, Flow chart and pseudocode. Half the code.

-**Seanna**: Half the documentation, half the code.

**Users**

Use code to determine their minimum, maximum, and average grades as well as the letter grades.

* 1. **Operating Environment**

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

Example:

CIS is developed for use on the Unix system: student

Under the Netscape web server: entprise 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.

Used with any hardware that can use the LC-3 simulator

Needs LC-3 simulator for use

Developed on the LC-3 editor

* 1. **Design and Implementation Constraints**

Note any constraints or limitation to the application.

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 high learning curve.

Need to use the LC-3 simulator

User needs to enter the scores from lowest to highest to work

* 1. **Assumptions and Dependencies**

Note any dependencies

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.

It is assumed that the user knows how to use the LC-3 simulator and has test scores available.

***3*. External Interface Requirements**

* 1. **User Interfaces**

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

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 and links and icons to navigate among the web pages.

* 1. **Hardware Interfaces**

Specify hardware interface – computer types? Terminal types?

* 1. **Software Interfaces**

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

* 1. Communications Interface

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

**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)?**

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’s 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

**Purpose**: Find min, max, avg of five test scores and give letter grades.

**Inputs**: Five test scores (keyboard/ numbers)

**Output**: Letter grades, min max avg scores

**Processing**: Converting the scores into the letter grades and determining the min, max, and avg scores.

**Data**: user database

**4.2 Performance requirements  
 What is the expected performance level of the program?**

**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 large number of users in future.

**4.2.4** Error handling should be implemented and the application should be able to handle all run time errors.

* + 1. The application should be able to be used on LC-3 simulators
    2. The input should be entered in the simulator menu that shows the output
    3. The app should hold and use five inputs

**4.3 Flow Chart and Pseudocode.**

|  |
| --- |
|  |
|  | http://web.mit.edu/ssit/cis/CISRequirements_files/image001.gif |

|  |
| --- |
|  |
|  | http://web.mit.edu/ssit/cis/CISRequirements_files/image002.gif |

Program starts

Input Subroutine

-Ask user for input

-Checks for input validation by subtracting last input

- if good, stores input

-add/ subtract counter and loop back to top

- keeps going until counter is done, returns back

Grade subroutine

-Only goes through odd inputs

-checks and assigns each grade with letter

-each letter is stored

-Returns back once all odds are checked

Display Subroutine

-Displays appropriate title(ie Min, avg, Max)

-Displays appropriate score (odds only)

-Displays appropriate Letter

End Program

Test example of grade letter

Input: 73

R1, R1, input

R7, R7, x60

NOT R7

R6, R1, R7

BRn ISF

AND R7, R7, x0

R7, R7, x70

NOT R7

R6, R1, R7

BRn ISD

AND R7, R7, x0

R7, R7, x80

NOT R7

R6, R1, R7

BRn ISC

ISC Give the ASCII equivalent of C, then store C

If less than 5

If smaller

Checks if counter has hit to 5

Checks if current input is larger than last input

Converts score into letter grade, then stores score and letter

Asks User to input score/User inputs score

Display Minimum, Average, and Maximum scores with letter grades