# CIS17A Small Business Application Documentation

## Introduction

### Purpose

The program is meant to generate a price for a user’s order and store the order for in a text file.

### Intended Audience and Reading Suggestions

The target audience are professional bakers and customers of a Bakery.

### Product Scope

Our Bakery application is intended to replace collecting orders with pen and paper with a digital system that generates a price for a user based on the cake type, filling type, and frosting type, prints out a receipt and stores the order in a text file for the bakery’s records.

### Reference

#### Source Documents for the Application Requirements Specification

Reference Project Part 1 requirements or business requirements.

**Example**

1. **Input/Output With Files**
   1. [**http://www.cplusplus.com/doc/tutorial/files/**](http://www.cplusplus.com/doc/tutorial/files/)

 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

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 purpose of the application functionality?

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 application is used to order custom cakes. Flavors, icings, and topping.

The application will be used for the P.O.S. that will allow the employee to put in the order of the customer.

There will also be a web based version that will allow the customer to order from home and pick up in store.

* 1. **Product Functions**

#### The overall description of functionality:

Highlight the program functionality and benefits for the business.

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?

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. The system is simple and clean. Which makes it easier for employees to put in orders in store.

2. A clean UI also makes it easy for customers to order form home.

3. Our program records the enters and calculates the price automatically. So there’s no confusing the type of cake with another. Also no risk of overcharging or undercharging due to human error.

* 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.

Bakery

The head baker would decide what cakes would be offered.

The employee would use the in house P.O.S. system to make orders.

Customer

The online application would take the order.

Once the order is complete it will print out the total for the customer

### 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.

The program is made to be used with windows systems.

### 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.

The main issue is for the online application is the user needing internet access. Also with a P.O.S system if it malfunctions the orders will be slowed down.

### 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 the, 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.

### Employees will have to be trained on the P.O.S. system.

### The site might require constant maintenance to make sure the it can handle large amounts of traffic.

The main issue is for the online application is the user needing internet access. Also with a P.O.S system if it malfunctions the orders will be slowed down.

***3*. External Interface Requirements**

3.1 **User Interfaces**

The user will interface with the program using prompt selections from a menu. The user can input data with the help of the keyboard to select options from a multi choice menu to select the cake flavor, frosting and filling options.

3.2 **Hardware Interfaces**

This program will be accessible on a sales terminal , available on both a Windows and Linux operating system.

3.3 **Software Interfaces**

This program will be accessible using any operating system that reads a cpp file format or C++.

3.4 Communications Interface

This application does not require web, Internet or network connectivity.

## 4. Detailed Description of Functional requirements

(Part 4 by Jonathan Bergquist)

### 4.1     Type of Requirement (summarize from Section 2.2)

Guides a user through menu of options, guarantees they select one of the valid options, and creates a cake order, offering an option to continue making more orders.

Purpose: Documents the order (single or multiple) per customer to print receipt and save information onto text file.

Inputs: Keyboard and mouse clicks on point of service machine.

Processing:  Checks the predefined prices of cake and topping combinations based on user choice, adds total of customer order, writes to file.

Outputs: Successive menu of options after customer selects each option, verifies that correct options are being selected with messages for incorrect input, confirmation of additional orders, final receipt of total order including: price, toppings, name of order.

Data: Stores order to local database to bakery. Will eventually use a server to accept online orders.

**SORT**

**SEARCH**

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

#### The application should be safe to protect customer payment information, efficient with memory as it will be running constantly on a point of sale machine, and efficient to scale with future online orders in case of sudden increase of volume.

### 4.3 Work Flow or Flow Chart

**\*SEE NEXT PAGE \***



