

# Proposed System: cuTPS

*by Christine Laurendeau*

## Problem Statement

With consumers progressively expecting more media content to be available on demand and on their own terms (music and e-books, to name a few), textbook publishing companies have been slow in adopting this new business model. Yet fewer of their consumers, namely students, are willing to hand over hundreds of dollars for a single textbook, the bulk of which they will not use. Course instructors in many disciplines have responded to this student behaviour by not requiring the use of textbooks, often to the detriment of the course delivery and student learning. Currently, few mechanisms exist to allow students to purchase individual chapters of e-textbooks, or even specific sections of a given chapter. The *Carleton University Textbook Publishing System (cuTPS)* proposes to address this shortfall by providing a tool for students to purchase selected portions of the e-textbooks required for their courses.

The main features of *cuTPS* include:

- the ability to purchase e-textbooks or individual portions thereof;
- the ability to add e-textbook content available for sale;
- the ability to run reports on e-textbook sales;
- the ability to manage essential system data, such as user accounts.

The *cuTPS* system supports three categories of users: students, content managers, and administrators. Students can view the required e-textbooks for the courses in which they are registered, and they can access details about each of these e-textbooks, including a list of the chapters and chapter sections available for sale. Using the shopping cart and checkout model common to most online vendors, students can place an order for as many e-textbooks, chapters and/or sections as they want. Content managers manage the material that is available for sale. They can add, delete or modify what e-textbooks, chapters and chapter sections can be purchased, along with their prices. Administrators manage user accounts, and they can run reports on sales data.

## Features

Student users can view a list of required e-textbooks for each course in which they are registered for a given term. For each e-textbook, they can see what individual chapters and chapter sections are available for purchase. Not every e-textbook will have chapters and sections for sale! Some will only be available as the entire e-textbook, some will have both the entire e-textbook and individual chapters for sale (but not necessarily every chapter), and others will have the e-textbook, chapters and chapter sections available. Student users can see the price of each item available for sale, and they can choose to add any item to their shopping cart. For this version of *cuTPS*, users cannot modify or delete items from their shopping cart. When they are done shopping, student users can choose to place their order and buy everything that is currently in their shopping cart, or cancel the order. Once an order is placed, delivery of the content occurs by email and user-initiated downloads that are outside the scope of *cuTPS*.

Content manager users are responsible for posting the e-textbook content that is for sale to student users. They can create, modify and delete course and e-textbook information, as well as what chapters and chapter sections are available, along with their prices. Content manager users cannot purchase the content themselves.

Administrator users manage the basic information required in the *cuTPS* system, and they can run reports. They can add, modify and delete user accounts, including billing and delivery information. They can also run reports on all sales data. These reports include viewing total revenue by term, by year, or by course. Other reports summarize data on frequent buyers and on the most frequently purchased content. Administrator users cannot purchase content, nor can they modify what content is available for sale.

## Technical Specifications

The Linux Ubuntu platform, as provided in the official course virtual machine (VM), will be used as the test bed for evaluating the *cuTPS* system, so all processes must work on this platform. All source code must be written in C++.

### GUI

The *cuTPS* user interface (UI) must be graphical in nature, and users must be able to view up-to-date information instantly.

In general, the different user features should be easily navigable, either as menu items and/or popup menus. The look-and-feel of the *cuTPS* system should be professional and consistent with commercially available UIs, including the use of popup menus where appropriate.

Reports must be concise, and they must consist of summarized data, formatted as a single line per record.

### Data Storage

Every user of *cuTPS* must be running a separate client process, which provides the *cuTPS* UI. However, all data must be stored centrally on a single host. A server process must execute on the central host, and it must manage updates and retrievals of the data.

The *cuTPS* system must be built to run on a lightweight client, possibly a mobile device in a future phase. As a result, the client must be designed to use very little memory and no persistent storage. Queries to the server must return only the minimum amount of data necessary. Almost no data should be stored on the client when the user moves from one UI screen to the next.

Data storage on the server must be organized for ease of retrieval and efficient use of storage space. There should be no duplication of information anywhere. Data may be stored in flat files, or any other mechanism available in the official COMP 3004 virtual machine provided.

### Inter-Process Communications

Each client process must execute on a different machine and support a single user. The client processes communicate with the central server process using TCP/IP sockets. Every time a user requests data, the *cuTPS* client process queries the central server, accessible at a configurable IP address, in order to populate the client's UI. Every time a user updates the data on his/her UI, the new information is sent to the server process, which updates its central storage. The *cuTPS* system must be able to support a minimum of four concurrent client processes, each executing on a different host. No client processes will run on the central server host.

# Team Project

## Deliverable #1: Phase #1 Model

The work submitted for this deliverable will consist of:

- a hard copy of the Requirements Analysis Document

The content of this deliverable will be discussed in class. A hard copy of the document must be submitted on or before Thursday, October 16 at 10:00 AM sharp, in the boxes in 3115 HP or in the classroom. The submitted copy must be typed and legible, and it must look as professional as if it was being submitted to real client. The document must be bound in a [Duo-Tang](#) folder with metal fasteners (no binders, no pocket folders, and no staples). Documents that do not conform to these specifications will not be marked.

## Deliverable #2: Phase #1 Prototype

The work submitted for this deliverable will consist of:

- the prototype source code, delivered on CD or DVD
- an in-class presentation on persistent storage design
- a soft copy of the PowerPoint or PDF slides for the presentation

The content of this deliverable will be discussed in class. The CD or DVD must be submitted on or before Thursday, November 6 at 10:00 AM sharp, in the boxes in 3115 HP or in the classroom. The presentation slides must be uploaded in [cuLearn](#) before the same date and time. An official design review will be held during class on November 11 and November 13. Each team will be given *five minutes* to present their design.

## Deliverable #3: Phase #2 Model

The work submitted for this deliverable will consist of:

- a hard copy of the System Design Document

The content of this deliverable will be discussed in class. A hard copy of the document must be submitted on or before Tuesday, November 25 at 10:00 AM sharp, in the boxes in 3115 HP or in the classroom. The submitted copy must be typed and legible, and it must look as professional as if it was being submitted to real client. The document must be bound in a [Duo-Tang](#) folder with metal fasteners (no binders, no pocket folders, and no staples). Documents that do not conform to these specifications will not be marked.

## Deliverable #4: Phase #2 Prototype

The work submitted for this deliverable will consist of:

- the prototype source code, delivered on CD or DVD
- a revised System Design Document
- a prototype demo to take place by appointment on Demo Day, Tuesday, December 9

The content of this deliverable will be discussed in class. The CD or DVD must be submitted on or before Monday, December 8 at 23:59, in the boxes in 3115 HP. The revised System Design Document must be uploaded in [cuLearn](#) before the same date and time.