REQUIREMENTS ANALYSIS

cuTPS – Carleton University Textbook Publishing System

Team Do Not Stick In Ear

Andrew MacCuaig
Graeme Jager
David Jatczak
Sina Dee

Table of Contents

Tables

Figures

1. Introduction

Given the popular rise of E-books, novels and magazines have gone the way of digital reading. With advancements in technology and the push towards paperless reading, it was a matter of time before textbooks found their way to being even more portable. The Carleton University Textbook Publishing System (cuTPS) will act as a venue to purchase course material digitally.

1.1. Purpose of System

Traditionally, course material for university students are in the form of textbooks. They are great resources with a wealth of information for specific courses. The downfall with textbooks is that they can get large, heavy and expensive - taking up space and causing back pain. With portable devices becoming more popular, the availability of electronic books has grown exponentially. Thus the service of cuTPS was born. cuTPS will allow students to purchase electronic copies of textbooks, individual chapters, or sections within, separately. The selling point of this service is to allow students to bring their course material around to view on their smartphone, or tablet along with the cost savings of not necessarily having to buy an entire textbook. By purchasing electronic textbooks, cuTPS allows publishers and universities to reduce the carbon footprint and costs that are associated with printing.

1.2. Overview of Document

With the intention to help understand the system clearly, this document includes several diagrams and description of how the system will function. Included in the document are Functional and Non-Functional Requirements, Use Cases, System Models, and a Glossary. These sections serves to clarify how cuTPS will behave and perform its tasks. All terms have been defined but it is expected of the reader to have some technical knowledge. The intended readers of the document

will be the client, analysts, and developers. The client will ensure that the system is working as to their intended vision. Analysts can use this document for several purposes such as troubleshooting and maintain efficiency. The developers can use this as a manual to help build the system.

2. Proposed System

2.1. Overview

cuTPS will run on a central server that users connect to via TCP/IP. A client program will be made available for users to interact with the system. Students will have the ability to purchase the content for the courses they are enrolled in only. Content Managers will be able to add, edit, and delete any of the content that is available on cuTPS. Administrators will be able to manage the users that can access the system as well as run reports on data in the system.

2.2. Functional Requirements

Functional requirements are actions that can be achieved by users of cuTPS. Not all users will be able to perform the same actions. Users are classified as Students, Content Managers, and Administrators. Each user will be able to perform tasks and functions that are specific to them and that other classified users will not have access to.

Table 1 - Functional Requirements

[F-01]	Users can view content. This can include:
[F-01-01]	A list of textbooks/chapters/sections
[F-01-02]	The price of a textbook/chapter/section
[F-02]	Students can view their shopping cart
[F-03]	Students can add an item to their shopping cart
[F-04]	Students can clear their shopping cart
[F-05]	Students can purchase the contents of their shopping cart
[F-06]	Content Managers and Administrators can manage courses. Content Managers and Administrators can:
[F-06-01]	Create courses
[F-06-02]	Edit courses
[F-06-03]	Delete courses
[F-07]	Content Managers can manage textbooks. Content Managers can:
[F-07-01]	Create textbooks
[F-07-02]	Edit textbooks
[F-07-03]	Delete textbooks
[F-08]	Administrators can manage user accounts. Administrators can:
[F-08-01]	Create user accounts
[F-08-02]	Edit user accounts
[F-08-03]	Delete user accounts
[F-09]	Users can manually update client data
[F-10]	Administrators can run reports. Administrators can view:
[F-10-01]	A report of revenue earned by period
[F-10-02]	A report of revenue by course for a given term
[F-10-03]	A report of frequently purchased content
[F-10-04]	A report of users that frequently purchase content
[F-10-05]	A report of revenue and quantity by content type

2.3. Non-Functional Requirements

Non-functional requirements are aspects of the system that do not relate to the execution of cuTPS.

Table 2 - Non-functional Requirements

[NF-01]	*Interface*	Textbooks once purchased are emailed to the Student
[NF-02]	*Implementation*	System must run on customer provided Ubuntu Virtual Machine
[NF-03]	*Implementation*	Source code must be in C++
[NF-04]	*Usability*	The user interface must be easily navigable
[NF-05]	*Usability*	Reports run by the Administrator must be formatted as a single line
[NF-06]	*Implementation*	Users must connect to server via TCP/IP
[NF-07]	*Implementation*	Each client process must execute on a different machine and support a single user
[NF-08]	*Reliability*	Client must be lightweight.
[NF-08-01]	*Reliability*	Client executable must be less than 50MB of disk space
[NF-08-02]	*Reliability*	Client executable must use less than 500MB of RAM
[NF-09]	*Performance*	The system must be able to support a minimum of four concurrent client processes
[NF-10]	*Implementation*	Client updates are sent to the central server for storage
[NF-11]	*Operations*	Content may not be duplicated
[NF-12]	*Operations*	Courses/Users may not be duplicated
[NF-13]	*Legal*	Upon completion and acceptance of Deliverable 4, system source code must be published under GNU GPLv3 (GNU General Public License version 3)
[NF-14]	*Packaging*	The final product, Deliverable 4, shall be packaged on optical media
[NF-15]	*Performance*	Client-server interactions should not take more than 500ms
[NF-16]	*Reliability*	Database must be backed up every 24 hours
[NF-17]	*Supportability*	System must support UI replacement
[NF-18]	*Supportability*	System must be able to support internationalization
[NF-19]	*Supportability*	IP address must be configurable
[NF-20]	*Usability*	Errors must be able to be understood by the user
[NF-21]	*Usability*	System must provide indication of wait time (loading icon)

2.4. System Models

A system model is a conceptual model that describes and represents a system. It helps analysts understand the functionality of the system. These models will help present how cuTPS will function from different perspectives. These perspectives will help the client understand how the system will behave, its architecture and how it's designed. It's a blueprint to ensure that the pieces of the system are implemented and evolve in a consistent manner. This allows for easier integration of additional functionality in the future.

2.4.1. Use Case Model

A Use Case Model describes proposed functionality of how a system works in terms of its functionality. A Use Case may "include" another Use Case's functionality or "extend" another Use Case with its own behavior.

Use Case Overview

The high level Use Case is brief description of the main processes of a system. It comprises of all actors who interact with the system. In this diagram, you'll see that the actors who will interact with cuTPS are Students, Content Managers, Administrators, and an external Email System.

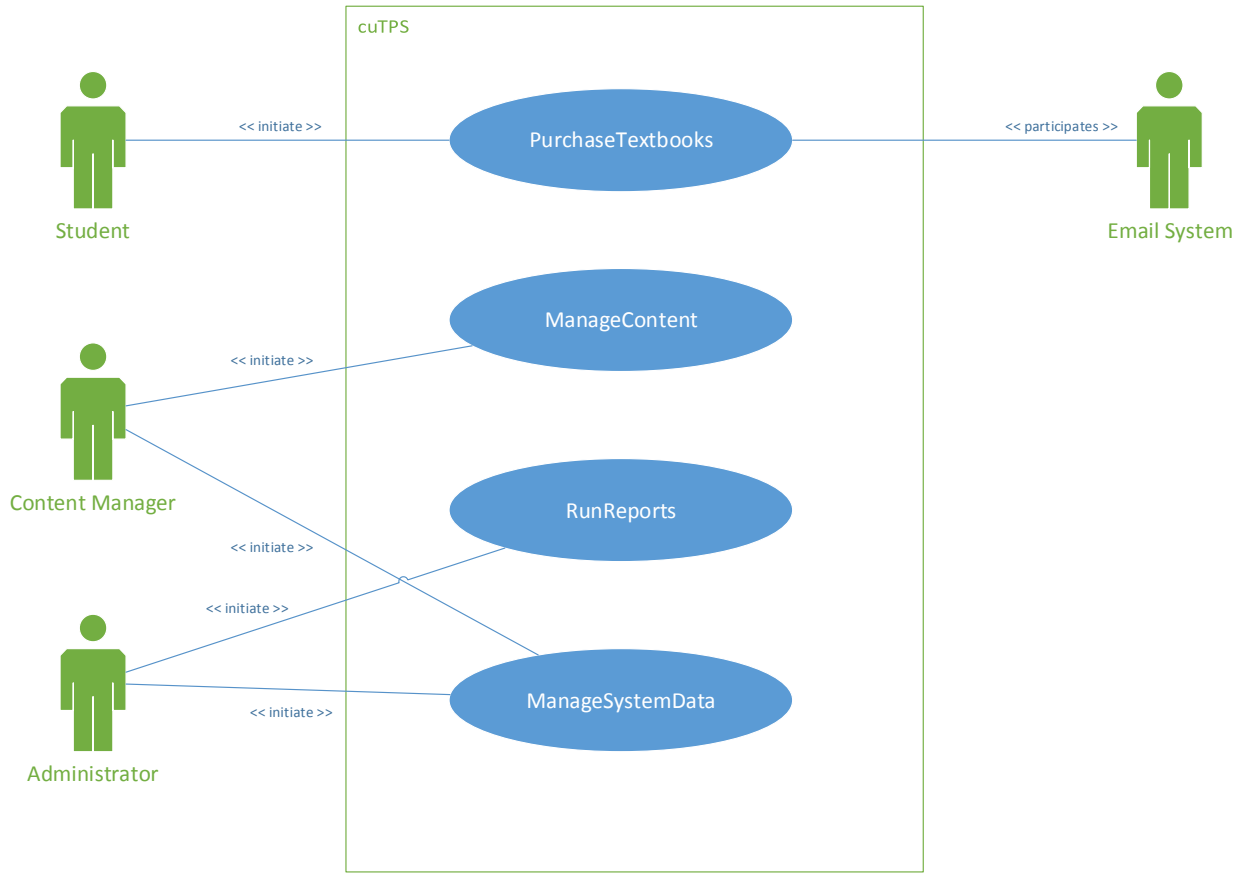

Figure 1 - High-level Use Case Diagram

Table 3 - High-level Use Case Descriptions

[UC-01]	PurchaseTextbooks	The Student browses and purchases available material
[UC-02]	ManageContent	The Content Manager maintains what is available for purchase in the system
[UC-03]	RunReports	The Administrator runs reports
[UC-04]	ManageSystemData	The Administrator and Content Manager maintain system information

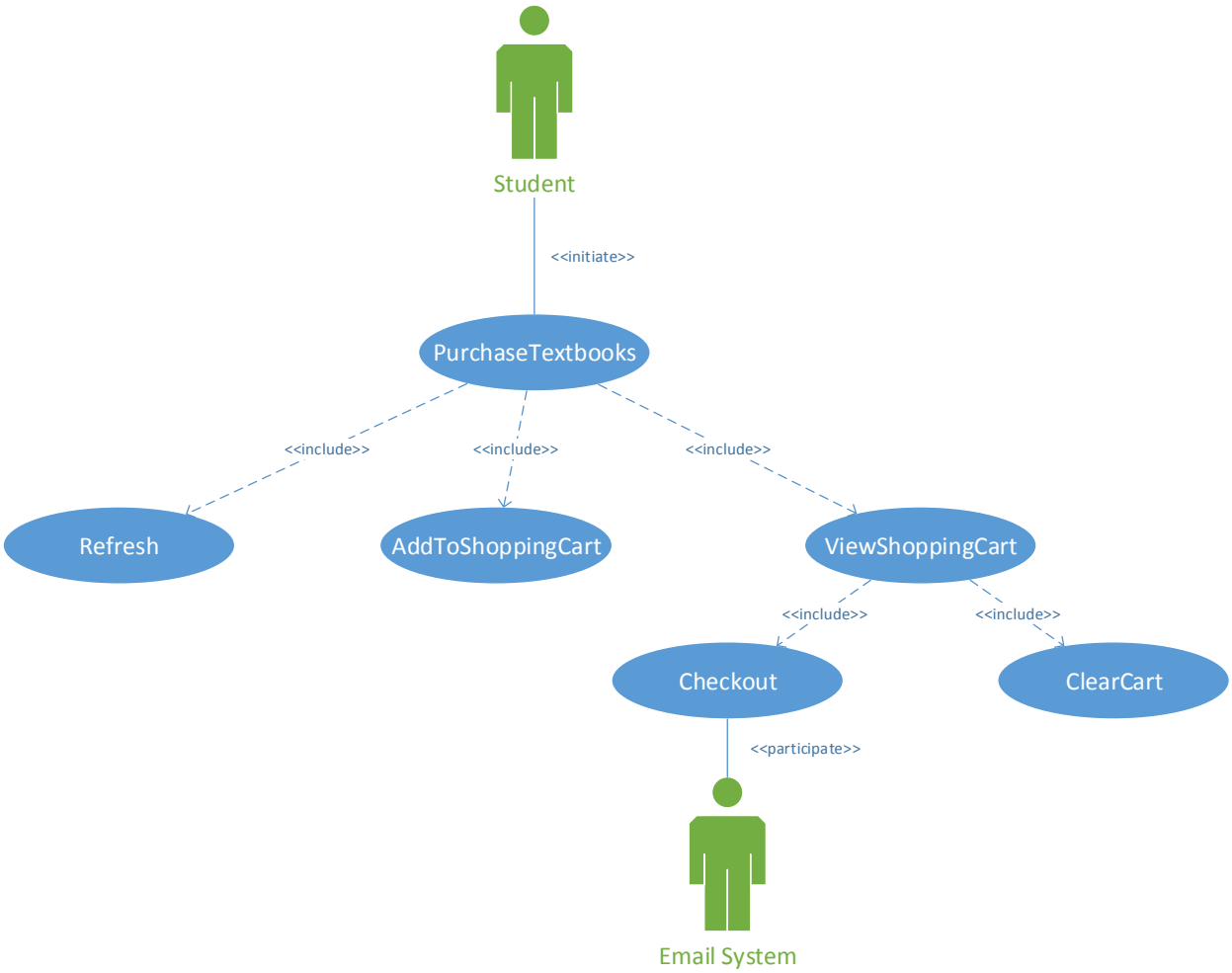

Figure 2 - PurchaseTextbooks Use Case

Table 4 - PurchaseTextbooks Use Case Descriptions

[UC-05]	Refresh	The User refreshes the list of available content
[UC-06]	AddToShoppingCart	The Student adds content to the shopping cart
[UC-07]	ViewShoppingCart	The Student views what is in their shopping cart
[UC-08]	Checkout	The Student chooses to purchase what is in their shopping cart and enters their billing information
[UC-09]	ClearCart	The Student empties their cart

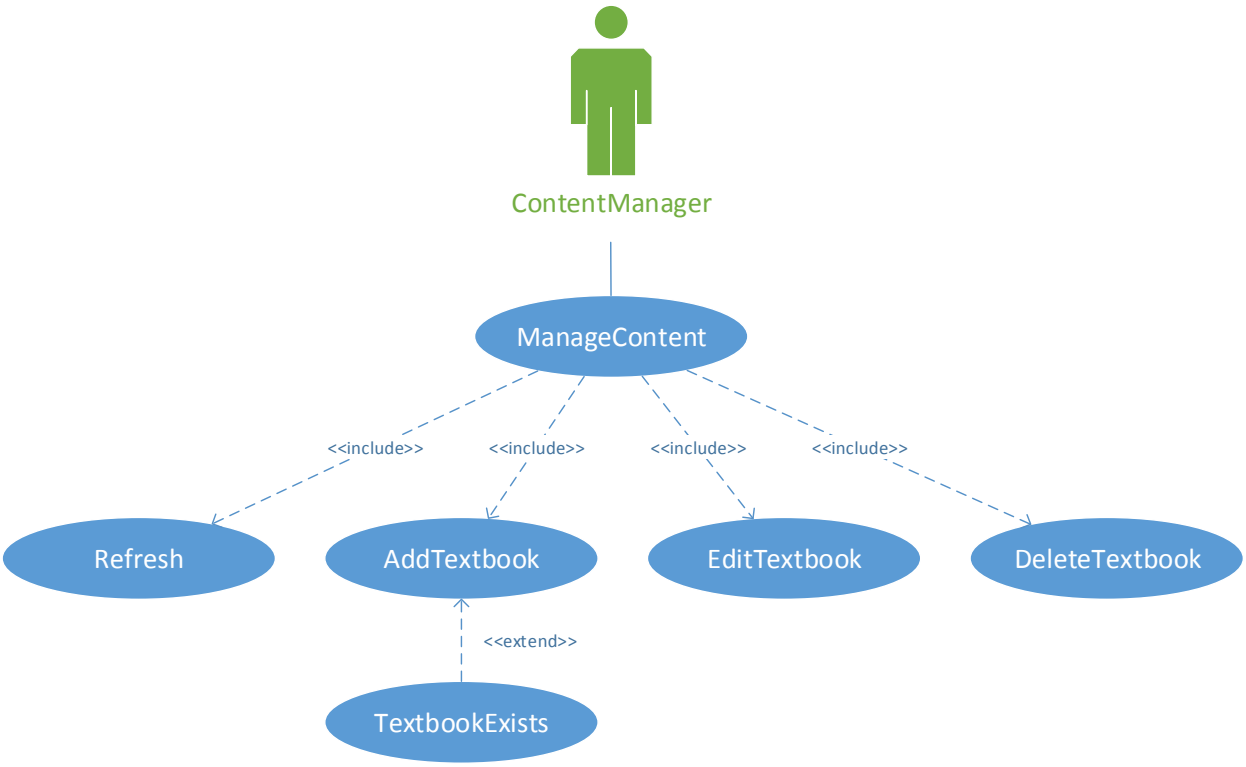

Figure 3 - ManageContent Use Case

Table 5 - ManageContent Use Case Descriptions

[UC-05]	Refresh	The Content Manager refreshes the list of textbooks
[UC-10]	AddTextbook	The Content Manager adds a textbook to the system
[UC-11]	TextbookExists	The system already contains the textbook
[UC-12]	EditTextbook	The Content Manager edits a textbook's information (what is available for purchase)
[UC-13]	DeleteTextbook	The Content Manager removes a textbook from the system

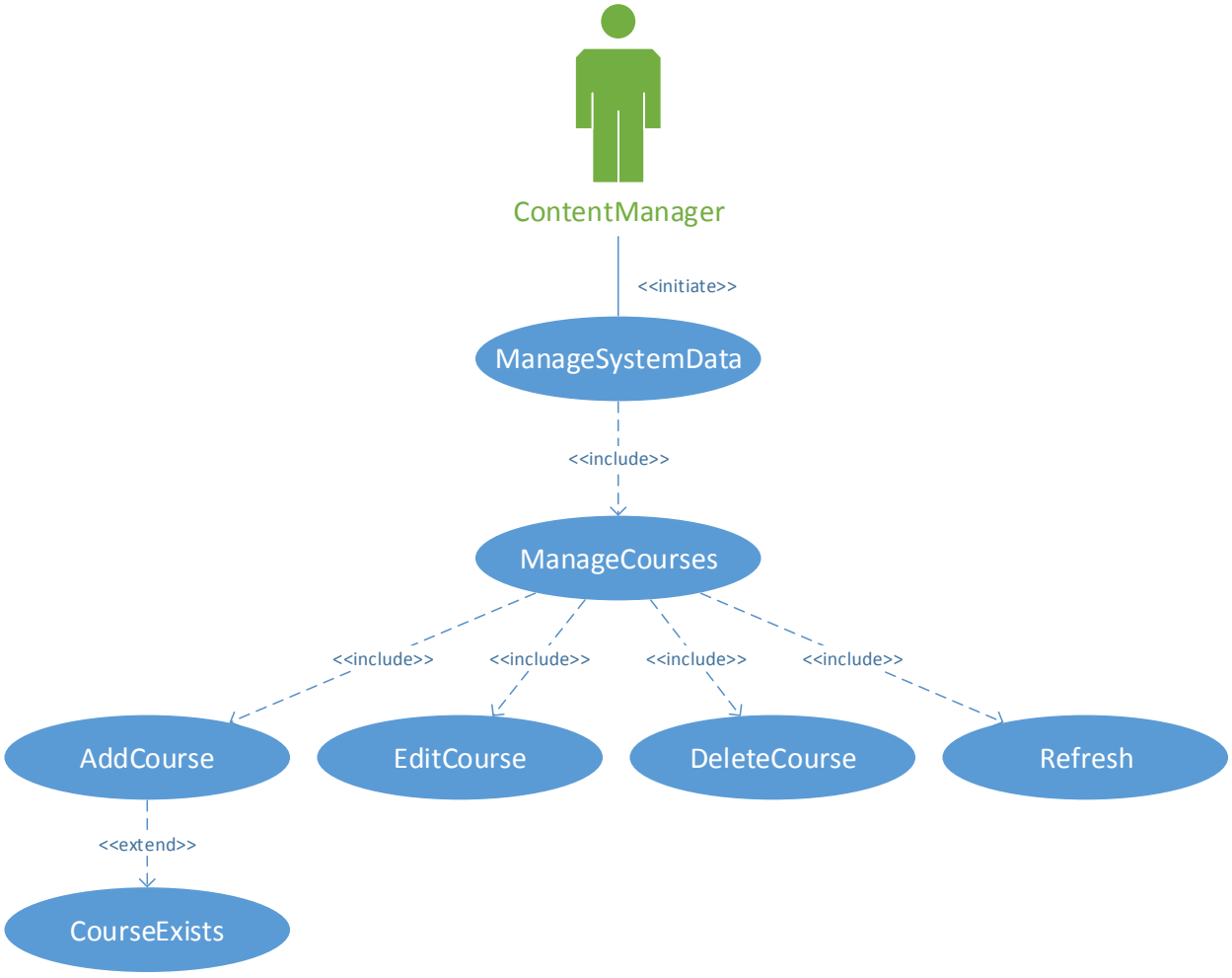

Figure 4 - ManageSystemData Use Case

Table 6 - ManageSystemData Use Case Descriptions

[UC-14]	ManageCourses	The Content Manager manages the courses that are in the system
[UC-15]	AddCourse	The Content Manager adds a course to the system
[UC-16]	CourseExists	The system already contains the course
[UC-17]	EditCourse	The Content Manager edits a course's information
[UC-18]	DeleteCourse	The Content Manager removes a course from the system
[UC-05]	Refresh	The Content Manager refreshes the list of courses

Use Case Flow of Events

Table 7 - PurchaseTextbooks Flow of Events

Use Case Identifier	[UC-01]
Name	PurchaseTextbooks
Participating actors	Initiated by Student
Flow of events	1. Student is shown available content. 2. Student can add the content to the shopping cart (include *AddToShoppingCart*) 3. Student can view shopping cart, they're shown the contents of their shopping cart (include *ViewShoppingCart*). 4. Student can refresh the available list of content (include *Refresh*).
Entry conditions	User logs in as a Student
Exit conditions	
Quality requirements	1. Retrieving the content should not take more than 500ms.
Traceability	[F-01], [F-01-01], [F-01-02], [NF-04], [NF-15]

Table 8 - Refresh Flow of Events

Use Case Identifier	[UC-05]
Name	Refresh
Participating actors	Initiated by Student Initiated by Content Manager
Flow of events	1. User enters command to refresh the user interface. 2. User interface displays most up-to-date information.
Entry conditions	The user has elected to refresh.
Exit conditions	
Quality requirements	1. While the system is processing the request, a loading icon is displayed. 2. Refreshing should not take more than 500ms.
Traceability	[F-09], [NF-15], [NF-21]

Table 9 - AddToShoppingCart Flow of Events

Use Case Identifier	[UC-06]
Name	AddToShoppingCart
Participating actors	Initiated by Student
Flow of events	1. Student enters command to add content to shopping cart. 2. System adds content to the Student's shopping cart.
Entry conditions	Student has selected content to purchase.
Exit conditions	Shopping cart reflects changes.
Quality requirements	
Traceability	[F-03]

Table 10 - ViewShoppingCart Flow of Events

Use Case Identifier	[UC-07]
Name	ViewShoppingCart
Participating actors	Initiated by Student
Flow of events	1. Student is shown the shopping cart, along with options to buy or clear what is in the cart. 2. If Student chooses to checkout, they're directed to a confirmation page to review their order and enter their billing information (include *Checkout*). 3. If Student chooses to clear their shopping cart, their shopping cart is emptied (include *ClearCart*).
Entry conditions	Student has elected to view the shopping cart.
Exit conditions	
Quality requirements	
Traceability	[F-02]

Table 11 - Checkout Flow of Events

Use Case Identifier	[UC-08]
Name	Checkout
Participating actors	Initiated by Student
Flow of events	1. Student reviews their order. 2. Student enters their billing information. 3. Student enters command to confirm their order. 4. System passes order information off to the email system 5. The shopping cart is emptied.
Entry conditions	Student has elected to checkout. The cart cannot be empty.
Exit conditions	Billing information cannot be empty.
Quality requirements	
Traceability	[F-05], [NF-01]

Table 12 - ClearCart Flow of Events

Use Case Identifier	[UC-09]
Name	ClearCart
Participating actors	Initiated by Student
Flow of events	1. The system empties the shopping cart. 2. The Student is displayed an empty shopping cart.
Entry conditions	Student has elected to empty their shopping cart.
Exit conditions	Shopping cart is empty.
Quality requirements	
Traceability	[F-04]

Table 13 - ManageContent Flow of Events

Use Case Identifier	[UC-02]
Name	ManageContent
Participating actors	Initiated by the Content Manager
Flow of events	1. Content Manager is shown all content. 2. Content Manager can add a textbook (include *AddTextbook*). 3. Content Manager can edit a textbook (include *EditTextbook*). 4. Content Manager can remove a textbook (include *DeleteTextbook*). 5. Content Manager can refresh the list of content (include *Refresh*).
Entry conditions	User logs in as a Content Manager
Exit conditions	
Quality requirements	
Traceability	[F-01], [F-07]

Table 14 - AddTextbook Flow of Events

Use Case Identifier	[UC-10]
Name	AddTextbook
Participating actors	Initiated by Content Manager
Flow of events	1. Content Manager enters command to add a textbook. 2. Content Manager adds textbook information. 3. Content Manager specifies the purchasable content. 4. Content Manager confirms the creation of the textbook 5. The new textbook is added to the system.
Entry conditions	Content Manager has elected to add a new textbook.
Exit conditions	A new textbook has been added to the system.
Quality requirements	
Traceability	[F-07-01]

Table 15 - TextbookExists Flow of Events

Use Case Identifier	[UC-11]
Name	TextbookExists
Participating actors	Content Manager
Flow of events	1. The Content Manager is notified that the textbook they are trying to add is already in the system.
Entry conditions	The Content Manager adds a book that already exists
Exit conditions	The add operation is cancelled.
Quality requirements	Notifications must be easily understood by the user.
Traceability	[NF-11], [NF-20]

Table 16 - EditTextbook Flow of Events

Use Case Identifier	[UC-12]
Name	EditTextbook
Participating actors	Initiated by the Content Manager
Flow of events	1. Content Manager has selected a textbook to edit. 2. Content Manager modifies the textbook information. 3. Content Manager enters a command to confirm the changes. 4. The system updates the textbook.
Entry conditions	Content Manager has elected to edit a textbook.
Exit conditions	The textbook's information has been updated.
Quality requirements	
Traceability	[F-07-02]

Table 17 - DeleteTextbook Flow of Events

Use Case Identifier	[UC-13]
Name	DeleteTextbook
Participating actors	Initiated by the Content Manager
Flow of events	1. Content Manager selects a textbook to delete. 2. Content Manager confirms the deletion of the textbook. 3. The system deletes the textbook.
Entry conditions	Content Manager has elected to delete a textbook
Exit conditions	The textbook has been deleted from the system
Quality requirements	
Traceability	[F-07-03]

Table 18 - ManageSystemData Flow of Events

Use Case Identifier	[UC-04]
Name	ManageSystemData
Participating actors	Initiated by the Content Manager
Flow of events	1. Content Manager is shown all system data types available to manage. 2. Content Manager can manage courses (include *ManageCourses*)
Entry conditions	User logs in as Content Manager.
Exit conditions	
Quality requirements	
Traceability	[F-06], [F-08]

Table 19 - ManageCourses Flow of Events

Use Case Identifier	[UC-14]
Name	ManageCourses
Participating actors	Initiated by Content Manager
Flow of events	1. Content Manager enters command to manage courses. 2. Content Manager is shown all courses. 3. Content Manager can add a course (include *AddCourse*. 4. Content Manager can edit a course (include *EditCourse*). 5. Content Manager can remove a course (include *DeleteCourse*) 6. Content Manager can refresh the list of courses (include *Refresh*).
Entry conditions	Content Manager has elected to manage courses.
Exit conditions	
Quality requirements	
Traceability	[F-06]

Table 20 - AddCourse Flow of Events

Use Case Identifier	[UC-15]
Name	AddCourse
Participating actors	Initiated by Content Manager
Flow of events	1. Content Manager enters command to add a course. 2. Content Manager adds course information. 3. Content Manager confirms the creation of the course 4. The new course is added to the system.
Entry conditions	Content Manager has elected to add a new course.
Exit conditions	
Quality requirements	
Traceability	[F-06-01]

Table 21 - CourseExists Flow of Events

Use Case Identifier	[UC-16]
Name	CourseExists
Participating actors	Content Manager
Flow of events	1. The Content Manager is notified that the textbook they are trying to add is already in the system.
Entry conditions	The Content Manager adds a book that already exists.
Exit conditions	The add operation is cancelled
Quality requirements	Notifications must be easily understood by the user.
Traceability	[NF-12], [NF-20]

Table 22 - EditCourse Flow of Events

Use Case Identifier	[UC-17]
Name	EditCourse
Participating actors	Initiated by the Content Manager
Flow of events	1. Content Manager has selected a course to edit. 2. Content Manager modifies the course information. 3. Content Manager enters a command to confirm the changes. 4. The system updates the course.
Entry conditions	Content Manager has elected to edit a course.
Exit conditions	The courses information has been updated.
Quality requirements	
Traceability	[F-06-02]

Table 23 - DeleteCourse Flow of Events

Use Case Identifier	[UC-18]
Name	DeleteCourse
Participating actors	Initiated by the Content Manager
Flow of events	1. Content Manager selects a course to delete. 2. Content Manager confirms the deletion of the course. 3. The system deletes the course.
Entry conditions	Content Manager has elected to delete a course
Exit conditions	The course has been deleted from the system
Quality requirements	
Traceability	[F-06-03]

2.4.2. Object Model

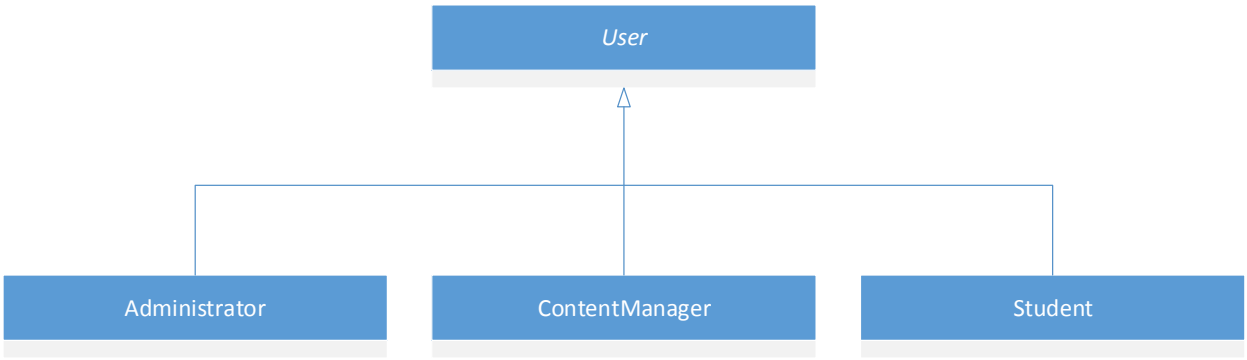

Figure 5 - User Entity Object Diagram

Table 24 - User Entity Object Data Dictionary

Entity Object	Attributes / Associations	Definition
[OB-01] User	- Name - Username - Password	A user who can access the cuTPS system. *Refs:* [UC-01], [UC-02]
[OB-02] Administrator		A user object that manages system data including user accounts and courses. Can also run reports on data in the system. *Refs:* [UC-03], [UC-04]
[OB-03] ContentManager		A user object that manages content data as well as courses. *Refs:* [UC-02]
[OB-04] Student	- Student Number - Shopping Cart - Course List	A user object that can view and purchase content for the courses they are registered in. *Refs:* [UC-01]

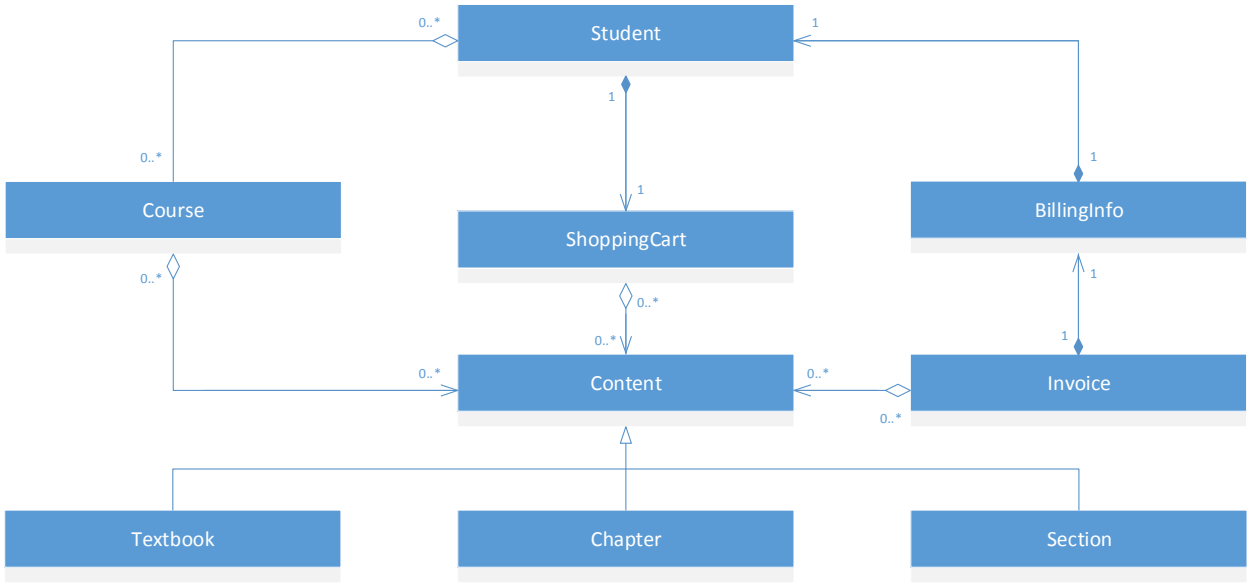

Figure 6 - Student Entity Object Diagram

Table 25 - Student Entity Object Data Dictionary

[OB-05] Course	- Course Number - Course Name	An object that can be associated with a Student that allows the Student to see content from that course. *Refs:* [UC-04], [UC-01]
[OB-06] ShoppingCart	- Content List	An object that contains a list of content that a Student wishes to purchase. *Refs:* [UC-01], [UC-06], [UC-07], [UC-08], [UC-09]
[OB-07] BillingInfo	- Credit Card Number - Credit Card Expiry Date - Credit Card Name - Credit Card Security Code - Address - Email Address	An object that contains the billing information for a Student that is required to complete a purchase. *Refs:* [UC-07], [UC-08]

[OB-08] Content		An object representing content that can be purchased by a Student. *Refs:* [UC-01], [UC-02]
[OB-09] Textbook	- Title - ISBN - Publisher - Author(s) - Year Published - Edition - Chapter List	A Content object representing a textbook that can be purchased by a Student *Refs:* [UC-02]
[OB-10] Chapter	- Chapter Number - Chapter Name - Parent Textbook - Section List	A Content object representing a chapter that can be purchased by a Student. *Refs:* [UC-02]
[OB-11] Section	- Section Number - Section Name - Parent Chapter	A Content object representing a Section that can be purchased by a Student. *Refs:* [UC-02]
[OB-12] Invoice	- Content list - Student - Billing information	Contains information about a processed order to be given to the Email System and stored in the system. *Refs:* [UC-08]

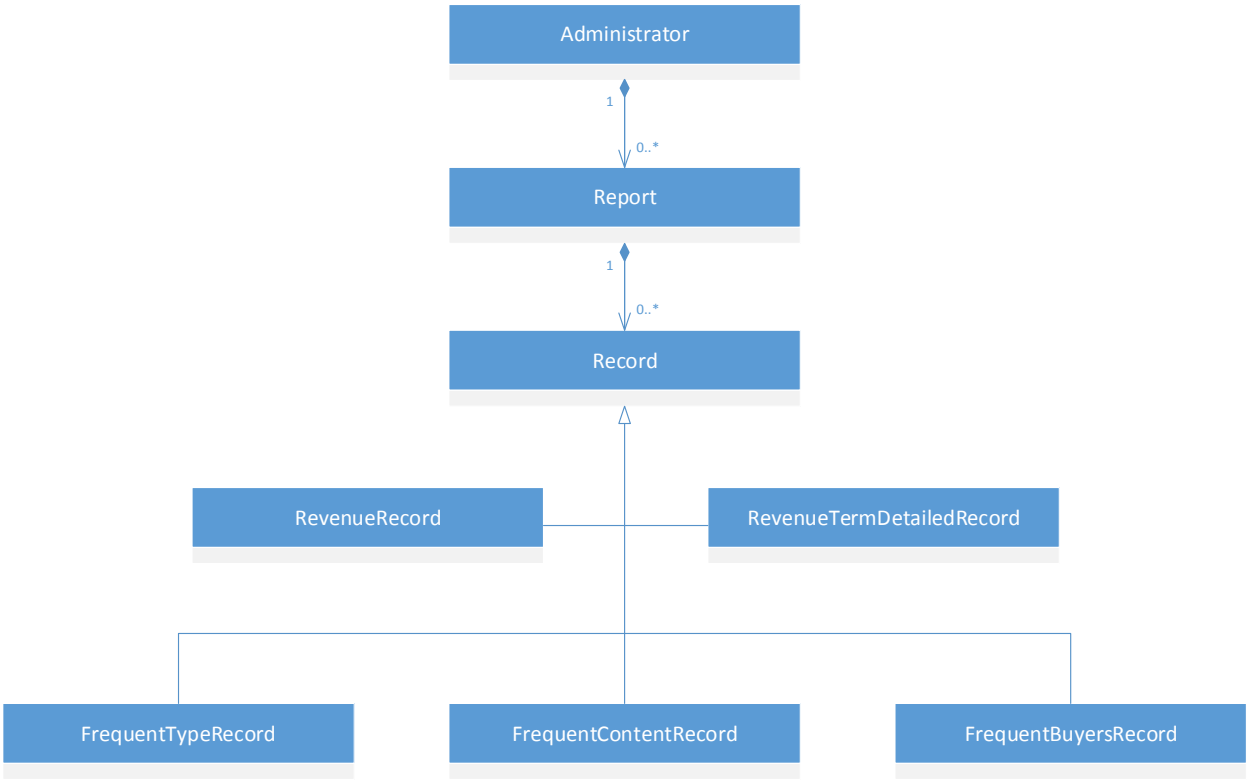

Figure 7 - Administrator Entity Object Diagram

Table 26 - Administrator Entity Object Data Dictionary

Entity Object	**Attributes / Associations**	**Definition**
[OB-13] Report	- Record	A report contains records that convey information to the Administrator. *Refs:* [F-10]
[OB-14] Record		An object that represents an entry in a report. *Refs:* [OB-13]
[OB-15] RevenueRecord	- Period - Revenue	A record that shows revenue for a requested period. *Refs:* [F-10-01]

[OB-16] RevenueTermDetailedRecord	- Term - Course - Total	A record that shows revenue for a requested period but grouped by course. *Refs:* [F-10-02]
[OB-17] FrequentContentRecord	- Content Title - Course - Number of Purchases - Revenue	A record that shows which content is the most frequently purchased. *Refs:* [F-10-03]
[OB-18] FrequentBuyersRecord	- Student Name - Student ID - Number of Purchases - Total Spent	A record that shows which students are the most frequent buyers. *Refs:* [F-10-04]
[OB-19] FrequentTypeRecord	- Type of Content - Total Sold - Revenue	A record that shows the revenue and the quantity by content type. *Refs:* [F-10-05]

2.4.3. Dynamic Model

2.4.3.1. State Machines

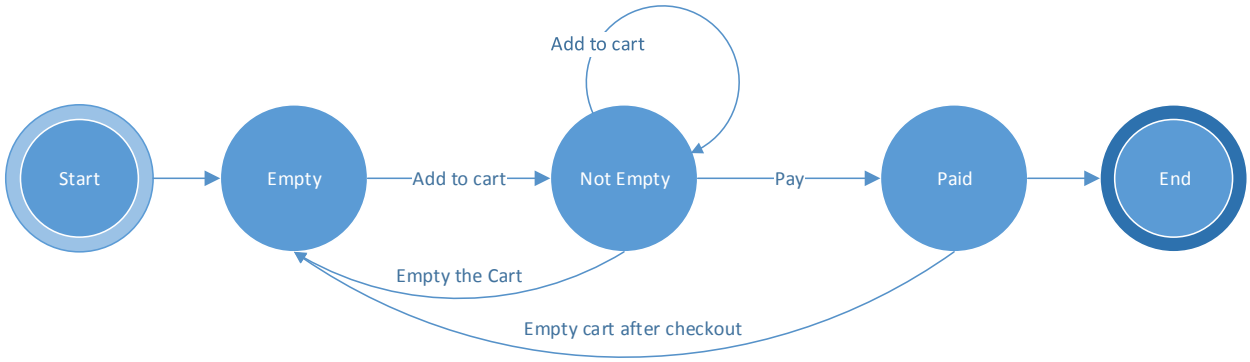

Figure 8 - ShoppingCart State Diagram

Table 27 - ShoppingCart State Machine Description

Identifier	[SD-01]
Class Reference	[OB-06] Shopping Cart

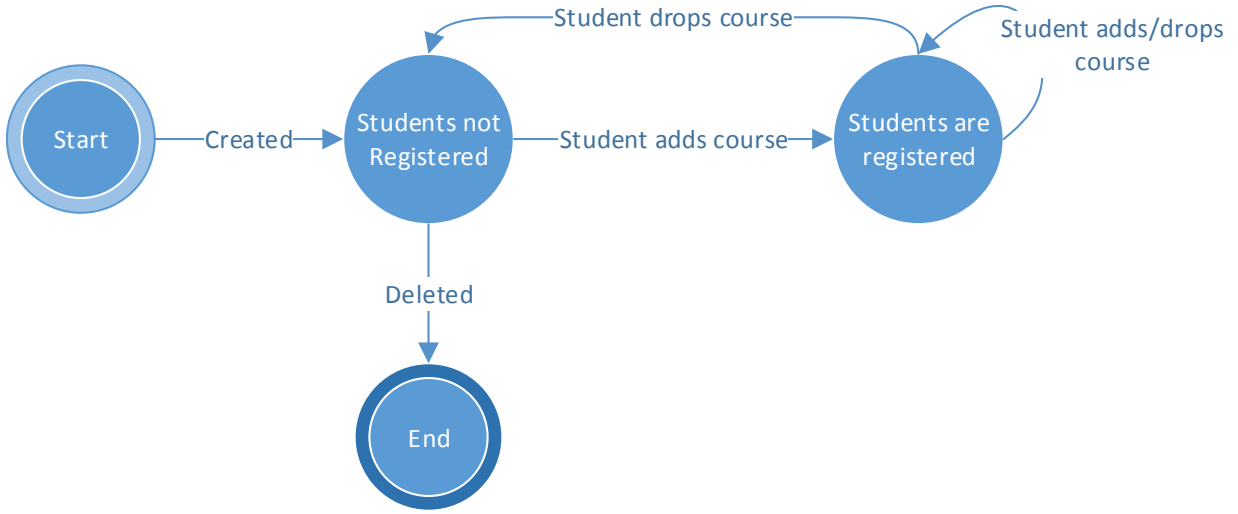

Figure 9 - Course State Diagram

Table 28 - Course State Machine Description

Identifier	[SD-02]
Class Reference	[OB-05] Course

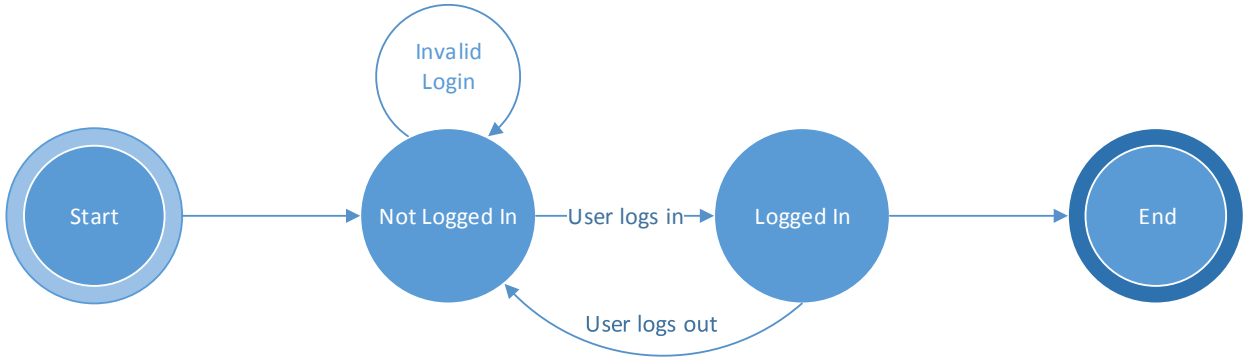

Figure 10 - User State Diagram

Table 29 - User State Machine Description

Identifier	[SD-03]
Class Reference	[OB-01] User

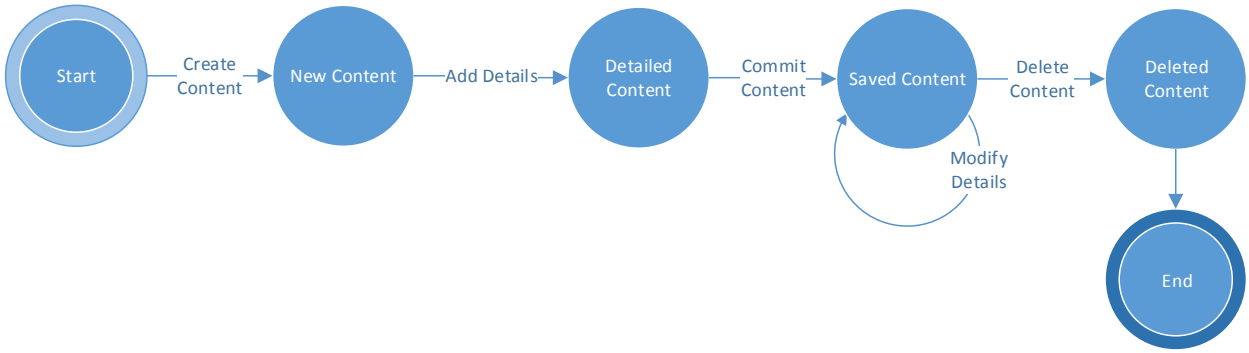

Figure 11 - Content State Diagram

Table 30 - Content State Machine Description

Identifier	[SD-04]
Class Reference	[OB-08] Content

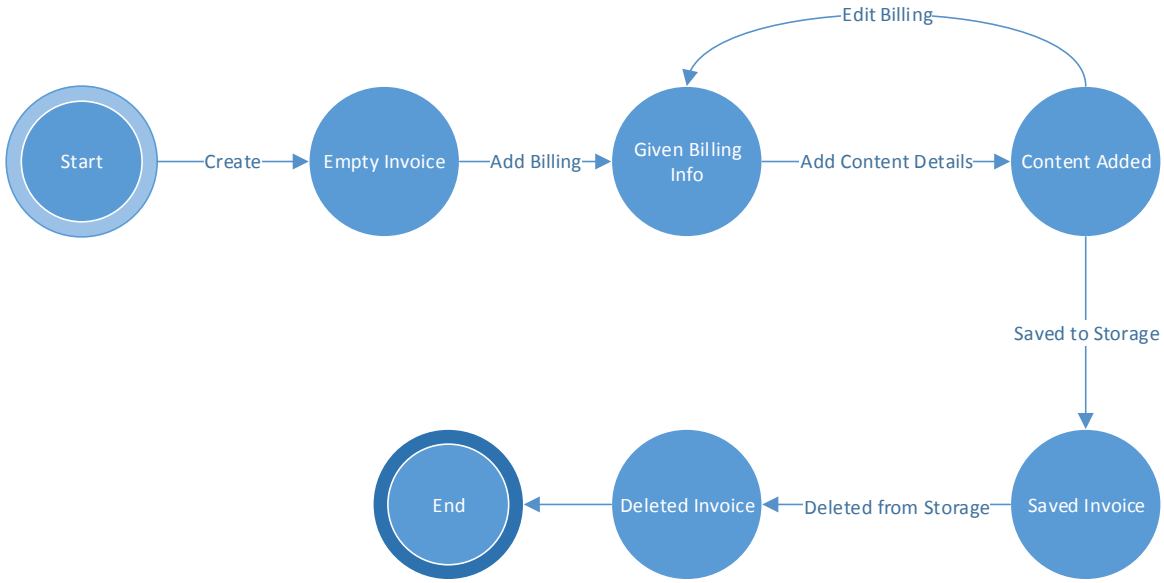

Figure 12 - Invoice State Diagram

Table 31 - Invoice State Machine Description

Identifier	[SD-05]
Class Reference	[OB-012] Invoice

2.4.3.2. Sequence Diagrams

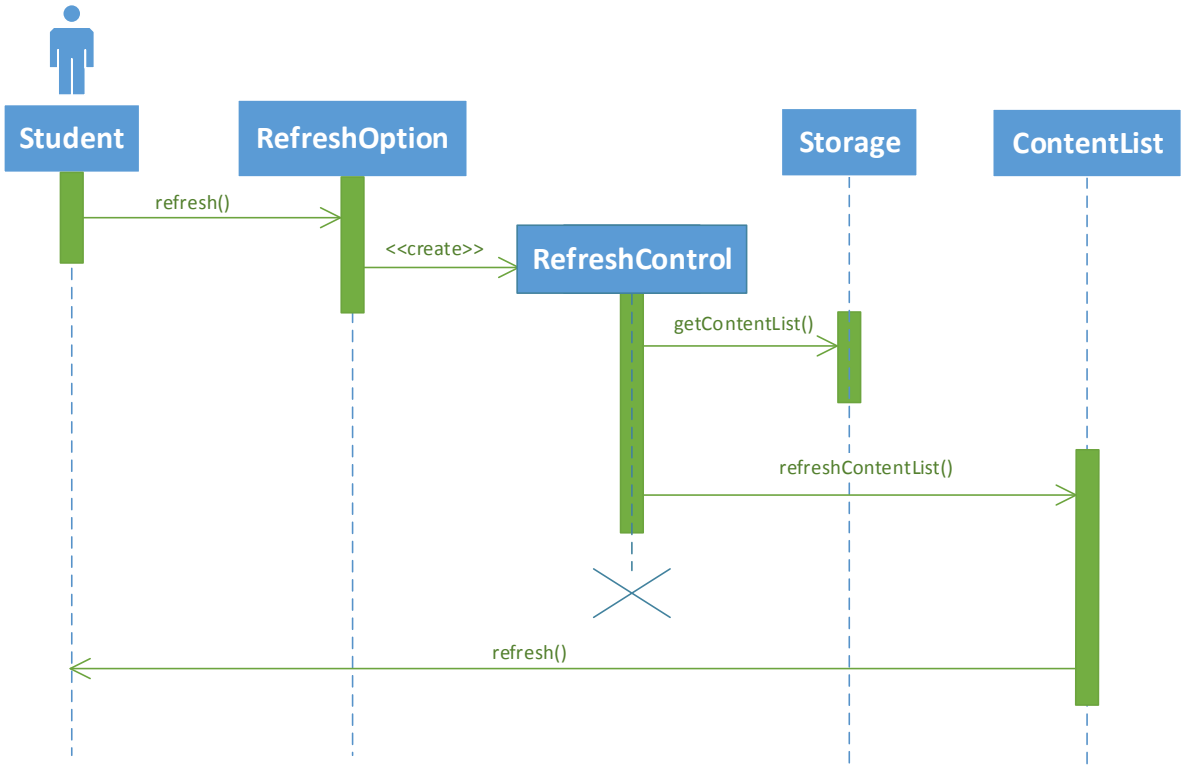

Figure 13 - Refresh Sequence Diagram

Table 32 - Refresh Sequence Description

Identifier	[SD-01]
Name	Refresh
Traceability	[UC-05]

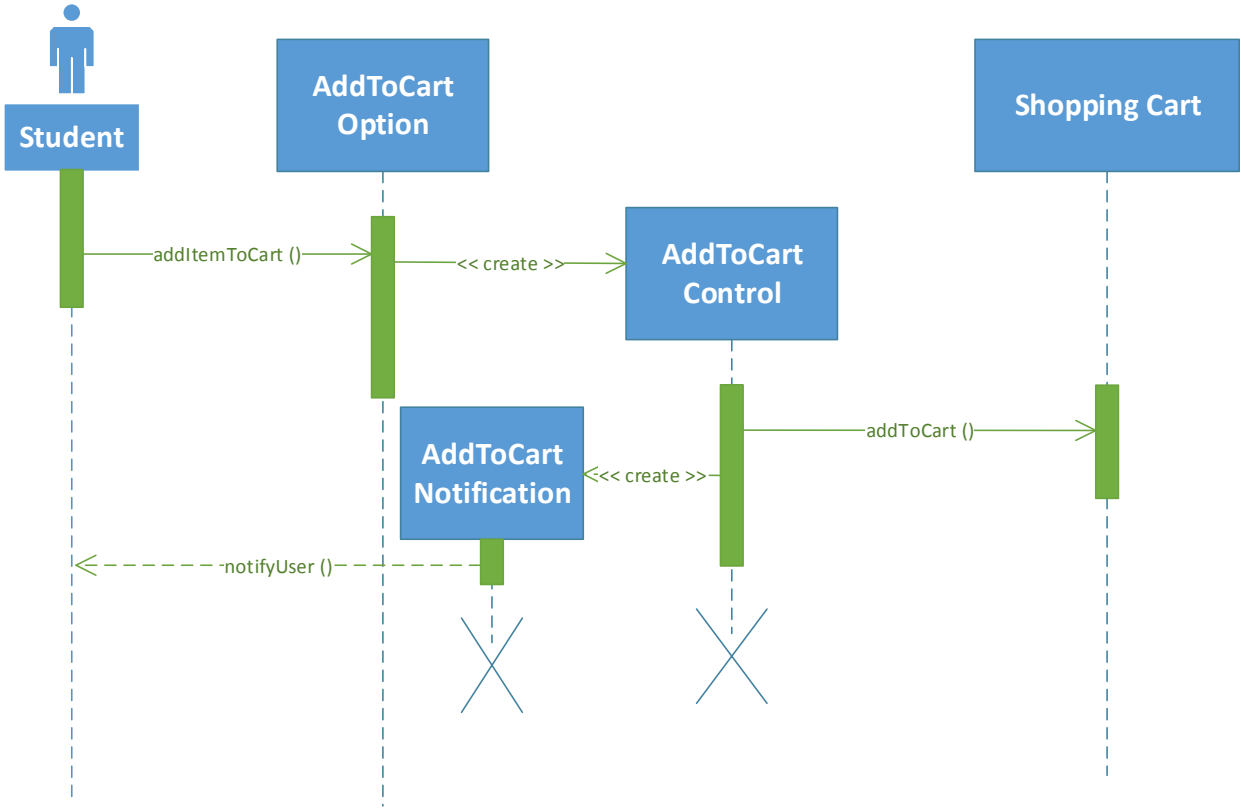

Figure 14 - AddToCart Sequence Diagram

Table 33 - AddToCart Sequence Description

Identifier	[SD-02]
Name	AddToCart
Traceability	[UC-06]

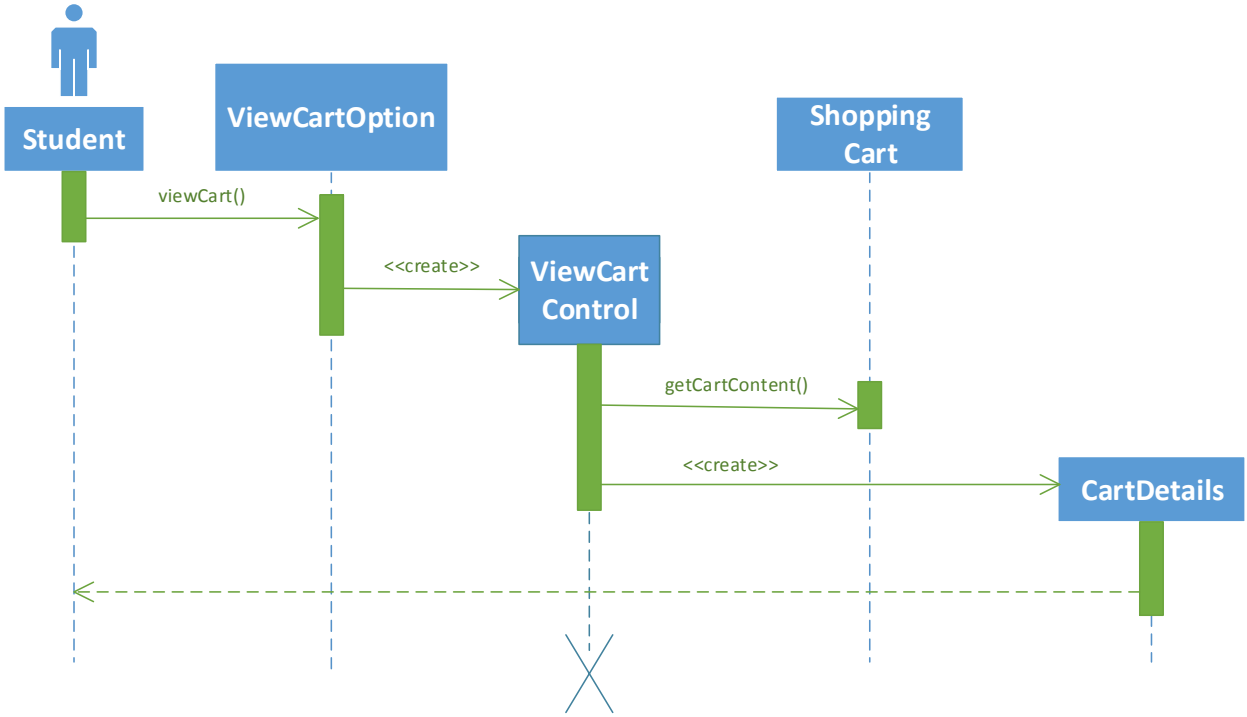

Figure 15 - ViewCart Sequence Diagram

Table 34 - ViewCart Sequence Description

Identifier	[SD-03]
Name	ViewCart
Traceability	[UC-07]

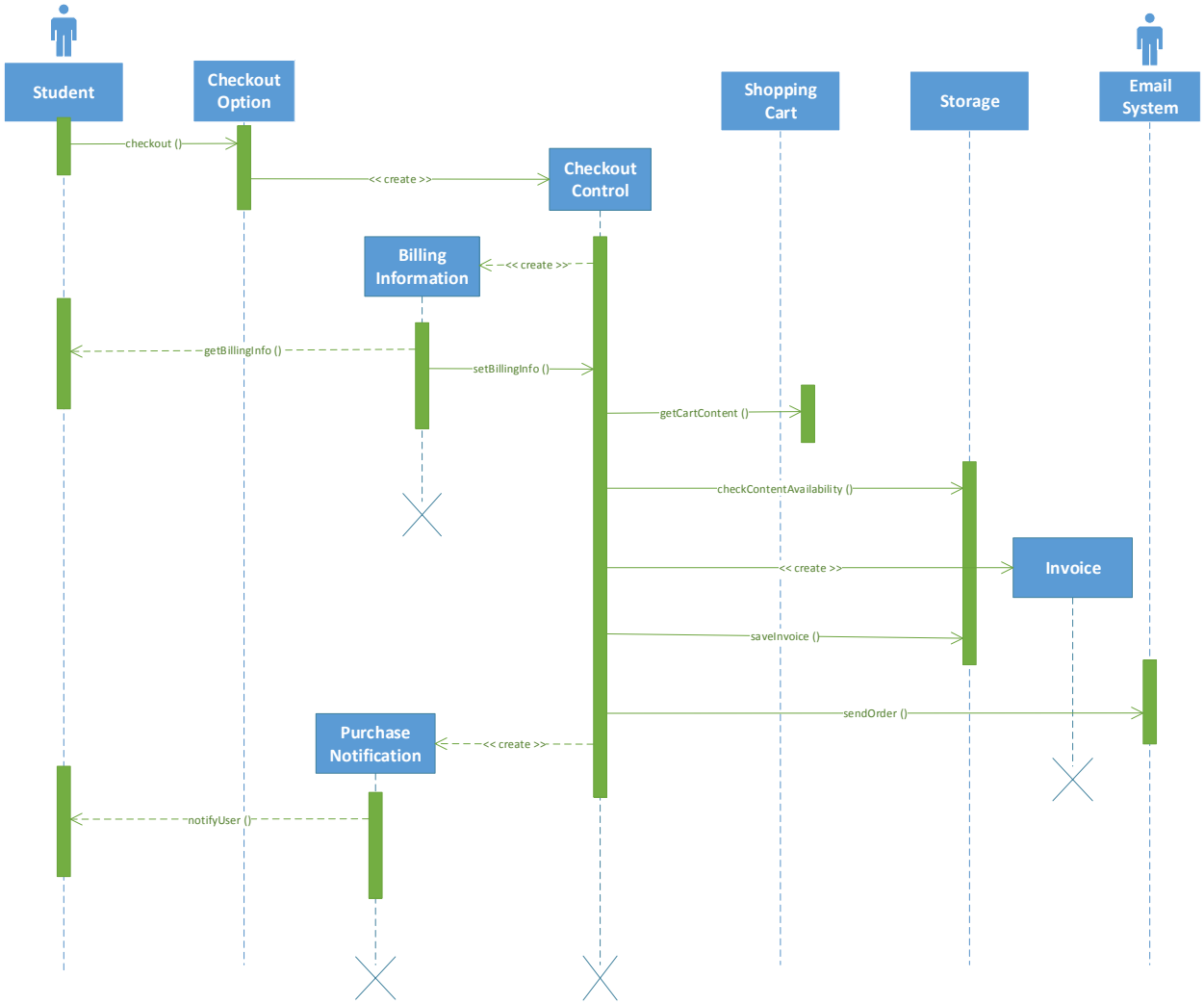

Figure 16 - Checkout Sequence Diagram

Table 35 - Checkout Sequence Description

Identifier	[SD-04]
Name	Checkout
Traceability	[UC-08]

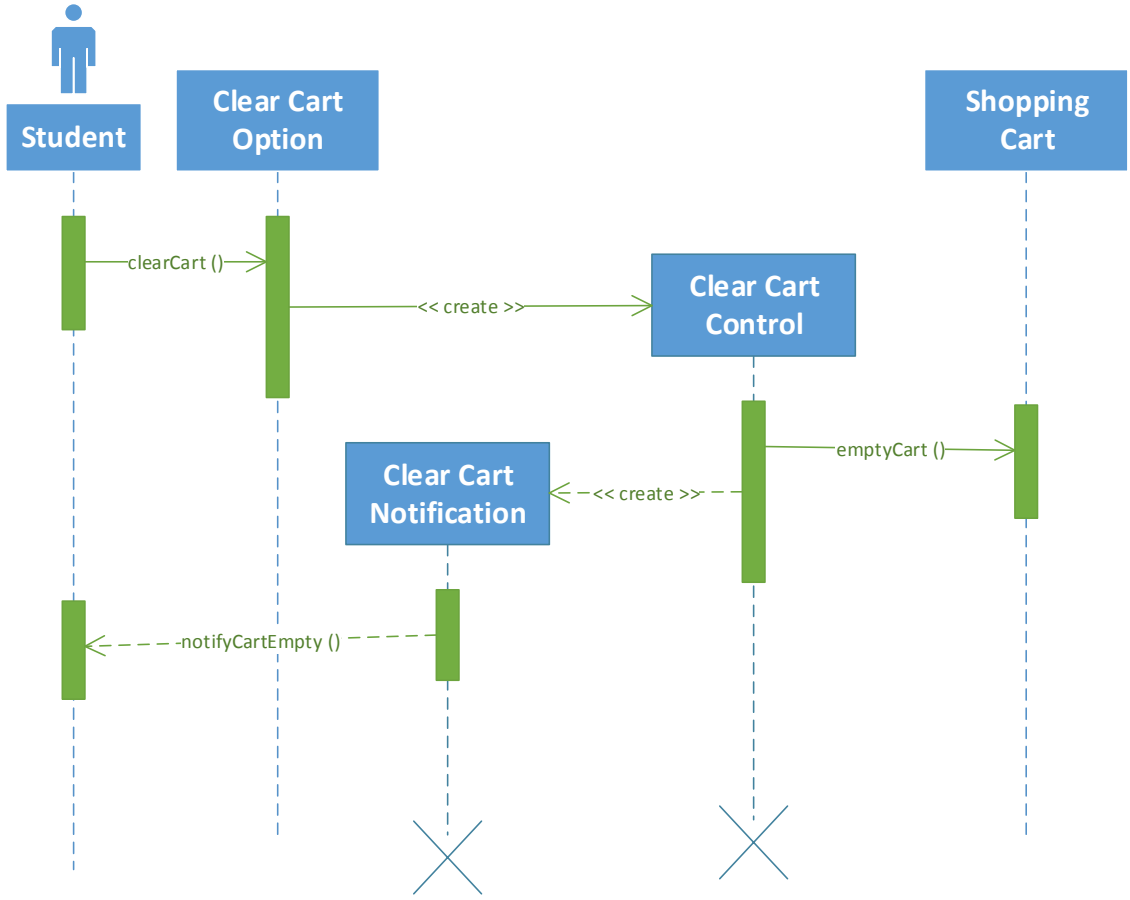

Figure 17 - ClearCart Sequence Diagram

Table 36 - ClearCart Sequence Description

Identifier	[SD-05]
Name	ClearCart
Traceability	[UC-09]

3. Glossary

Content: Content can mean a textbook, a chapter of a textbook or a section thereof unless explicitly stated as such.

System: A general term for the cuTPS system in its entirety.