The A-Team

UMBC Textbook Marketplace

Tuc Nguyen - Amra Mo - Jun Kwon - Hao Wu

Amir Beshir - Sophia Shin

IS 436 - Structured Systems Analysis and Design

Deliverable II - Requirements Definition Document and Use Cases (D2)

March 28, 2019

**Table of Contents**

Requirements Definition…………………………………...……………………..….3

Introduction (Business Requirements)

User Requirements

Functional Requirements

Nonfunctional Requirements

Cultural and Political

System Description/Requirements

Questionnaire…………………………………………………...………………....…6

Use Case Documentation…………………………………………………………….14

Use Case Scenarios…….…………………………………………...…………..........16

**Requirements Definition**

**Introduction (Business Requirements):**

**Purpose:** Our mission is to create an application for UMBC that would serve as a marketplace for students to sell or purchase their textbooks amongst each other as a listing. Each listing would have the student’s contact information, the condition of the book, subject and class the book belongs in, catalog #, book title, author, offer price they are willing to sell/buy it for. We want to provide a safe service where UMBC-only students and alumni can reach out to one another on campus to get rid of their textbooks easier, as well as make money without going through the trouble of selling / purchasing their books by having them shipped off online.

**User Requirements:**

* The system will keep track of all data required by the user to maintain a database by generating user listings.
* Should be understandable and minimal for non-technical people to use.
* Written for Customers

**Functional Requirements:**

Process-Oriented:

1. User must be able to sell and buy.
2. User must be able to create and delete new accounts on our page.
3. User must be able to browse textbook listings.
4. User must be able to add, remove, and change any information.
5. User may make and reject offers made by other users.
6. User must be able to receive and pay for items
7. User may select new or used items

Information-Oriented:

1. Add new users
   * New users must register their username, password, and email to the database.
2. Update user info
   * Any changes made to any users username, password, email, address, payment method, and listings will be updated in our database.
3. Add new or user book/items
   * New entries must like book title, description, availability, price, seller, quantity, quality, edition, price, and author to the database.
4. Update new book/items
   * Any changes to the quantity, availability, price, condition, and seller must be updated in the database when changed.
5. Payment
   * Any major changes like payment or password changes must be verified by an email confirmation.
6. Authentication
   * User must authenticate before accessing system
7. Search
   * User can search for any books, items, and sellers.
8. Request books/items
   * User can request for any items or books

**Non Functional Requirements:**

**Operational:** To start building the application to run on most web browsers, we need to choose a web framework that is tested to run successfully on Chrome, Firefox, Safari, Edge, and other important browsers. Use popular tools such as Jquery and other frameworks that are proven to work.

**Performance:** To maximize the performance of the application we need to choose a web server that is fast and can handle many user connections. Nginx and the Apache load balancers are good choices to use to distribute user traffic to different application servers. The system should be available for user 24 hours per day, 365 days per year. It should also be able to support simultaneous users at all other times.

**Security:** System administrators will be allowed permissions to maintain the system. Developers should be authorized to access the system source code to provide new features and fix bugs associated with the system. Student users will only have permissions to view and edit their own book listings and system account information. Interpersonal politics affects the performance of the entire project. Buyer and seller protection and information security should be a priority for any ecommerce site.

**Cultural and political:** Our primary tools will be from academic partners in the industry such as Dell, Cisco, IBM, etc. We could choose to use cloud services such as Amazon Web Services, Microsoft Azure, or Google Compute Cloud instead of buying our own hardware for the servers. Find partnerships in order to reduce costs in the future. In order to reduce costs now, make minimal mistakes in the programming, have effective project management practices, and use minimum viable product releases. System should distinguish between US currency and currency from other nations. Student’s personal information is protected in compliance with the Data Protection Act

**System Description/Requirements:**

Subsystems could include a database system, a load balancing system, a network logging and metrics system, and a dedicated security system.

Running the system on a Linux based platform such as Ubuntu LTS or CentOS would allow us to have a solid and free operating system that has security updates and supports our enterprise requirements.

We need to build the system around the requirements. Use encryption, user rights restrictions, and connection blocking, and separation of duties for security. Use load balancers and commonly used frameworks in order to maximize usability across phone OSs and browsers. Cloud computing and open source software can reduce cost across the application.

**Questionnaire**

Below is a collection of questions interviewed by people of various backgrounds as a result of this requirements document.

Interviewee: Adrian Vu

Position: System Architect/ Programmer

Date/Time: 3/22/2019 at 7:00 PM

Interviewee: Tuc Nguyen

|  |  |
| --- | --- |
| **Question** | **Outcome** |
| As a programmer, what is your approach to build our application? | We first need to get the customer requirements. We organize the list of resources that the customer and the server administrators need to get access to. The web application would need to have a web service that accesses the books and a service that gets user details. Next we think about developing the user graphical interface and connecting it to the web service and the back-end database to store user and book information. Afterwards we will implement the book, the user, and other applicable Java object models based on the requirements. |
| Based on your experience,   1. how would you approach security of our application? 2. How would we be able to maximize the performance of application to minimize lag? | a.The security of the application will rely on user roles. We will define the system administrator role and the student user role. Each role will only have the least privilege. System administrators will be allowed permissions to maintain the system. Student users will only have permissions to view and edit their own book listings and system account information. The application will only allow SSL or TLS connections through the HTTPS protocol. Data will be encrypted during transit and at rest in the database. User passwords will be salted and only the SHA256 hash of the salted password is stored. The database can only be accessed by system administrators.  b.To maximize the performance of the application we need to choose a web server that is fast and can handle many user connections. Nginx and the Apache load balancers are good choices to use to distribute user traffic to different application servers. The application could support multiple user connections by using an optimized threadpool. |
| How would you suggest we start off building the application to run on most web browsers? | To start building the application to run on most web browsers, we need to choose a web framework that is tested to run successfully on Chrome, Firefox, Safari, Edge, and other important browsers. Use popular tools such as Jquery, NodeJS, ReactJS, and other frameworks that are proven to work. |
| On moving further with the project, how would you suggest we build our mobile application? | The mobile application can be built using Titanium Appcelerator, which makes the application work with Apple IOS and Android OS. Or the mobile application can be built using the Apple Objective C and the Android SDK platforms. |
| What OS do you suggest we run our system on? | Running the system on a Linux based platform such as Ubuntu LTS or CentOS would allow us to have a solid and free operating system that has security updates and supports our enterprise requirements. |
| Who should be authorized to access the system and under which circumstances | System administrators should be authorized to access the system to troubleshoot and maintain the application. Developers should be authorized to access the system source code to provide new features and fix bugs associated with the system. |
| Where should we be buying our tools (ie, computers, routers, etc.) from? | We can buy our tools from academic partners in the industry such as Dell, Cisco, IBM, etc. We could also choose to use the cloud such as Amazon Web Services, Microsoft Azure, or Google Compute Cloud instead of buying our own hardware for the servers. |
| What are certain subsystems we should include? | Subsystems could include a database system, a load balancing system, a network logging and metrics system, and a dedicated security system. |
| Summary | Build the system around the requirements. Use encryption, user rights restrictions, and connection blocking, and separation of duties for security. Use load balancers and commonly used frameworks in order to maximize usability across phone OS’s and browsers. Cloud computing and open source software can reduce cost across the application. |

Interviewee: Chris Nowak

Position: System Architect

Date / Time: 3/22/19 at 5:30 PM.

Interviewer: Jun Kwon

|  |  |
| --- | --- |
| **Question** | **Outcome** |
| 1. Even though you’re not a programmer currently, I understand that you have some experience as a programmer. What kind of languages are you proficient in, and how long has it been since you worked with website programming? 2. Do you have any experience in creating an online marketplace? What kind of work is my group up against? 3. What do you think is essential for an online marketplace? Both in the front-end, and in the back-end. 4. What kind of features should an online marketplace have, not necessarily essential, but stuff that it *should* have? 5. Are there any traps that novice programmers should look out for before trying on a project of this size? | Chris introduced himself as a systems analyst, and I explained what my group is trying to do. He has not been a programmer in a while, but he still remembers everything, he says. And while he has little experience in website programming, much less an e-commerce site, he is confident in his abilities to answer my questions.  He explained that with any programming job, it really is just hard work and dedication. The internet is a vast resource to study upon but sometimes no one has the answer to the specific problem that you’re facing, so debugging skills are crucial. And if you’re trying to work on a project, a common pitfall is the lack of comments. Comments in code save lives.  As for functional and nonfunctional requirements, Chris explained that e-commerce sites have to have, aside from what my group already listed (Listing creation and management, Profiles, Listing Browsers), they have to have the ability for accountability. A sale gone wrong needs an accountable party, whether it be seller, buyer, or website. By the very least it should have personal information available in these situations so that the police can be involved. What an e-commerce site *should* have would be security for the personal information on the database, such as firewalls, access restrictions, and access auditing. One overlooked feature of all programming projects is the team behind it, specifically how well people get along, and whether that affects their performance. |
| Summary | Communication, both passive and active, is important to any team; interpersonal politics affects the performance of the entire project. Buyer and seller protection and information security should be a priority for any ecommerce site. |

Interviewer: Amir Beshir

Interviewee: Nathaniel Lulseged

Position: BTA Major student

Date: 3/23/19

|  |  |
| --- | --- |
| **Question** | **Outcome** |
| As a Business Technology Administrator (BTA) major student at UMBC, how would you describe the business plan we have proposed on our textbook marketplace? | The business plan provides a realistic assessment of the company's strengths as well as the market environment in which it operates. It also portrays some of the challenges that the business faces on the SWOT analysis making it a believable plan. I haven't seen any exaggerations and descriptions that seem too good to be true. |
| As a senior at UMBC, do you believe our business plan is realistic enough to compete with our competitors? Also, do you think our possible partnerships can help us grow as a company? | I believe it's very realistic and it will definitely be able to compete with the other company's. UMBC bookstore is a good start for partnership, since they have many resources they can help attract the customers and also it is a great way to gain competitive advantage. |
| Do you recommend any other cost cutting methods in our proposal since our group members are all the stakeholders? | Usually when developing an app, quality assurance and subsequent bug fixing can shoot up the cost estimates so make sure it doesn’t have any bugs. Also using agile project management is useful since it breaks down the tasks and makes it more manageable and cheaper. Finally, I would recommend MVP[Minimum Viable Product] releases. It is cheaper to release MVP and it also helps to identify improvements and isolate bugs well ahead of the full-scale implementation process, which reduces further efforts. |
| If you were to run this textbook marketplace, how would you change our organizational feasibility? | I would keep it how it is because the only stakeholders are your group members and the business, in my opinion, is profitable. |
| What possible expansion or partnership methods would you recommend? | After i get enough customers, I'll start advertising for my free app and start an advertisement free subscription with other benefits, like free monthly book discounts. I would start Ebook rentals and start covering more locations to attract other new customers. |
| Summary | Be realistic in your goals so that they’re attainable. Find partnerships and plan realistic expansions in order to reduce costs in the future. But in order to reduce costs now, make minimal mistakes in the programming, have effective project management practices, and use minimum viable product releases. |

Name of interviewee: Jay Shin

Position of interviewee: Business Analyst: Business Intelligence & Analytics

Data and time of interview: 03/25/2019, 7pm

Name of interviewer: Sophia Shin

|  |  |
| --- | --- |
| **Question** | **Outcome** |
| 1. What steps are required to develop a product from an idea? 2. How do you handle the changes to requirements? 3. What are the documents that are used by a BA in a project? 4. What kind of requirements do you perform for the projects as a BA? 5. How well do you think you understand what the purpose of the application is? | According to my interviewee, Jay Shin, to develop a product, he mostly works on use cases, SDLC, user analysis, market analysis. He said that our team should know who our customer is because it is important to find what do they want to do on this website and what will make them satisfy from this website. Either, you can communicate with the users or you can be the users of the application. The various documents that a business analyst use to handle are use case diagrams, requirement documents, System Requirement Document (SRD), Functional Specification Document (FSD), etc. As a Business Analyst, he said he doesn’t do any coding or programming. The output of a BA’s work is utilized by IT people for developing the product and by the non-IT people to see the model of their application product. He said that a BA should get signed a document by the user that the changes to the requirements will not be accepted. He recommended me to use the process called SMART rule for helping me to make sure the requirements are good to go. So, if the requirements are specific, measurable, Attainable, relevant and timely, then they’re ready. I explained about our project and he said it is reliable and useful for many people, especially, UMBC students because there is a marketplace such as facebook marketplace but UMBC marketplace would be easier to find the way to buy/sell the books for people who goes to UMBC. He suggested making sure to follow the requirement documents. |
| Summary | Finding out what the customers want and satisfying them is the number one priority; their needs and wants are our requirements. |

Date 3/26/19

Interviewee: Rainbow Yim

Position: Data Analyst

Interviewer: Hao Wu

|  |  |
| --- | --- |
| **Question** | **Outcome** |
| How would you go about gathering information and data on what people want in an application? | “I would go to social media and ask friends to answer a quick survey to see the demographics.” |
| How would you store the customer's data in regards to their passwords and any personal data? | “Depending on where you store the data if you are storing the data locally I would have a encryption key to protect the data and basically encrypt all private and sensitive data.” |
| What are some issues that we should be aware of when making databases? | “Commenting all your functions and making sure everything works is crucial because you are not the only one that will be looking at it and if there is something wrong or if you need to figure out what something does you know where to look.” |
| Based on your experience with working on databases what is the hardest part of maintaining and handling with data on a day to day basis? | “I believe that the most difficult thing or tedious thing is the amount of data that I have to check since there is a lot.” |
| If we were to scale up our project how would you recommend us doing that? | “Take it to step by step and make sure you have a good foundation because the bigger you get the more errors and problems you will face and by scaling smaller you have a better chance in catching the errors and fixing it before it becomes unfixable” |
| What types of software did you use on a day to day bases? | “I use a lot of data mining software like Tactical Arbitrage to mine the data that I need to know about the sales rank and cost of the items that I should purchase.” |
| Summary | Communication, security, and slow and pious dedication is critical to any database. Use third-party software to ease the load. |

**Use Case Documentation**

**UMBC Textbook Marketplace System**

Application for UMBC that would serve as a marketplace for students to sell or purchase their textbooks amongst each other as a listing.

**Risk Factors**

· Before someone can register for the app, their student ID must match and be valid in the student database maintained by the registrar’s office.

· Must also include a picture of the student

· Must only be current UMBC students and alumni

· Student’s information could be compromised by a hacker.

· Feedback from beta testers to improve the product.

· Address security issues because student’s UMBC information is linked to their account with the app.

· The system must be operational prior in advance of the academic term starting.

· People designing the software may be inexperienced

· Preventing system failure

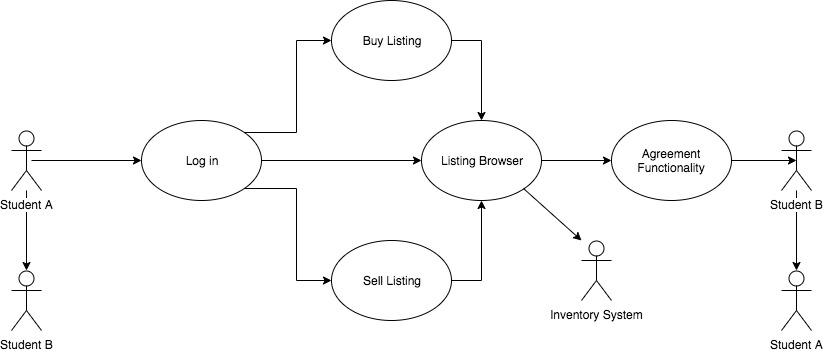
· Can we be successful if we don’t support a web interface / mobile application?

· Easy for non-technical people to use

· How to handle simultaneous users

· How to handle database crashing.

**System Level Use Case Diagram**

****

**Architecture Diagram**

In our current stage, we will be required to have certain Subsystems. The following below would be potentially included.

**Subsystem Descriptions**

· System Access Subsystem

o This subsystem handles log in and log out

o Handles student’s id

o Checks if still valid against student database maintained by registrar’s office

· Display Main Screen

o Handles a window that shows option for the user to pick amongst

o Buy Listing

o Sell Listing

o Listing Browser

· Book Info Subsystem

o Provides information about books on record,

o Inventory System

· Manage Student Info Subsystem

o Selling Book

o Buying Book

· Database Subsystem

o Contains the data we need to store for application

o Retrieve, update, delete functions for data stored

o Information about books (Title, Catalog #, Author, Year, Edition, etc)

**Use Cases Scenarios**

**Log in**

This Use Case describes the process which students log in to the system.

**Actors**

· Student A

· Student B

**Priority**

This part of the Use Case is essential to the rest. If student ID is not valid, or the user username or password is incorrect, the user will not be able to list, sell, or browse the book catalog.

**Pre-conditions**

**·**  If student ID valid and other credentials are correct, then user can enter the system.

· If student ID is not valid and other credentials are incorrect, then user can not enter the system.

**“Used” Use Cases**

None

**Flow of Events**

**·** The Use Case begins with the user beginning the application

· The system displays a login screen

· Student enters username, password, and studentID

· System validates information

· Sets access permissions

· Displays main screen that will show a variety of options for the user

· User selects function

· If user:

- Selects Buy Listing

Use Buy Listing option

- If wanting to use Buy Listing option:

Use Agreement Functionality option

o Else,

o Selects Sell Listing

Use Sell Listing option

· If wanting to use Sell Listing option:

Use Agreement Functionality option

o Else,

o Selects Listing Browser

Use Listing Browser option

o End if

o Student will select a function

o End loop

o Use Case ends

**User Interface**

**·** Prompts student to give username, password, and student ID

· Student gives user name, password, and student ID

· Checks to see if user name, password, and student ID is valid

**·**  Shows main screen that shows if student wants to buy, sell, or list

· Selects option

· Can use agreement functionality to if they want to buy or sell from Student to Student

**Scenarios**

**·**  Bad user name, password, or student ID

· User not in system

· No selection of a valid function

**Subordinate Use Cases**

· Display Main Screen

**Listing Browser**

This Use Case creates an entry for a book the student is selling, buying to view and use search feature

**Actors**

· Student A

· Student B

**Priority**

This part of the Use Case is not really essential unless student is wanting to use browse feature.

**Pre-conditions**

· If student ID valid and other credentials are correct, then user can enter the system.

· If student ID is not valid and other credentials are incorrect, then user can not enter the system.

· Student wants to view Listing Browser

**“Used” Use Cases**

Log in

**Flow of Events**

· Displays main screen that will show a variety of options for the user

· User picks Listing Browser

· Can use search filter to view for options: buying, selling, books on system, etc.

**User Interface**

· Ability to filter by categories such as buying or selling, price, author, and title.

· Search function, with advanced search, able to search whether a listing is buying or selling, price ranges, catalog numbers, titles, authors, and editions.

**Scenarios**

· Book is not available in the system

**Sell Listing**

This Use Case creates an entry for a book the student is wanting to sell

**Actors**

· Student A

· Student B

**Priority**

This part of the Use Case is not really essential unless student is wanting to create a sell listing.

**Pre-conditions**

· If student ID valid and other credentials are correct, then user can enter the system.

· If student ID is not valid and other credentials are incorrect, then user can not enter the system.

· Student wants to create sell listing

**“Used” Use Cases**

Log in

**Flow of Events**

· The Use Case begins with the user initiates if they want to create a sell listing

· Input information into fields

· Option to edit and delete

**User Interface**

· User will input valid information into fields such as contact information, book information, and pricing.

· Ability to edit and delete

**Scenarios**

· User wants to sell

· Invalid field information

**Buy Listing**

This Use Case creates an entry for a book the student is wanting to buy.

**Actors**

· Student A

· Student B

**Priority**

This part of the Use Case is not really essential unless student is wanting to create a buy listing.

**Pre-conditions**

· If student ID valid and other credentials are correct, then user can enter the system.

· If student ID is not valid and other credentials are incorrect, then user can not enter the system.

· Student is wants to create buy listing

**“Used” Use Cases**

Log in

**Flow of Events**

· The Use Case begins with the user initiates if they want to create a sell listing

· Input information into fields

· Option to edit and delete

**User Interface**

· User will input valid information into fields such as contact information, book information, and pricing.

· Ability to edit and delete

**Scenarios**

· User wants to sell

· Invalid field information

**Agreement Functionality**

This Use Case creates an entry if Student A wants to contact Student B to Buy/Sell from listing.

**Actors**

· Student A

· Student B

**Priority**

This part of the Use Case is not really essential unless student wants to buy / sell, and will only allow the student to do so if all credentials and listing information is correct.

**Pre-conditions**

· If student ID valid and other credentials are correct, then user can enter the system.

· If student ID is not valid and other credentials are incorrect, then user can not enter the system.

· Student wants to Buy / Sell

**“Used” Use Cases**

Log in

Buy Listing

Sell Listing

**Flow of Events**

· User selects function

· If user:

o Selects Listing Browser

Use Listing Browser option

· When seller/buyer is interested:

Send contact information to listing’s creator. Creator can then decide to message the interested party or not

o End if

o Student will select a function

* Use Case ends

**User Interface**

-User selects function and can input information into field for contact

**Scenarios**

· Information not valid

· Fields not valid