Software Requirements Specification

for

<Texchange>

Version 1.0

Prepared by <Christopher Pieri>

<Gordon Gellert>

<December 9, 2017>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience 1

1.3 Product Scope 1

1.3 Product Goals 1

1.4 References 1

2. Overall Description 2

2.1 System Enviroment 2

2.2 Functional Requirements Specifcation 2

3. Requirements Specifications 8

3.1 External Interfaces 8

3.2 Functional Requirements 10

3.3 Detailed Non-Functional Requirements 12

4. Engineering and Realistic Constraints 13

3.1 Engineering Standards 13

3.1 Realistic Constraints 13

5. Project Design 14

5.1 Overview System Components 14

5.2 Structure and Relationship 15

5.3 User Interface 15

5.4 Reuse and Relationship to Other Products 20

5.5 Design Decisions and Tradeoffs 20

5.6 Resource List 21

6. Project Plan 21

6.1 List of Tasks 21

6.2 Project Schedule 24

6.3 Risk Plan 24

6.4 Esitmated Financial Budget 24

6.5 Teamwork Plan 25

**Citations 25**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

*The purpose of this SRS is to present a detailed description of the Texchange web application, including a comprehensive look at the system features, functions, interfaces, constraints, and how the system will react to user inputs.*

## Intended Audience

*This document is intended for the following readers, developers, project managers, marketing staff, users, and tester.*

## Project Scope

*Texchange is a web application for students at Hofstra to buy and sell textbooks amongst each other. Existing solutions include buying and selling textbooks at the bookstore. By eliminating the middle man and having students interact with each other directly, the application provides students with an opportunity to maximize their return on their textbooks. While students purchasing textbooks can find them for less than they are offered for at the bookstore.*

*The website will allow sellers to create a post by entering information relating to their textbook. Likewise, buyers can search for a textbook by entering information relating to a textbook. If a match is found than the seller can contact the buyer and establish a location to make the exchange.*

## Project Goals

*This section lists three objectives to be met as a result of this project.*

*1. Create a campus based online marketplace where Hofstra students can buy and sell textbooks.*

*2. Produce an intuitive system that performs reliably, and meets industry standards as well as user expectations.*

*3. Obtain a clever and memorable domain name that becomes familiar to all Hofstra students.*

## References

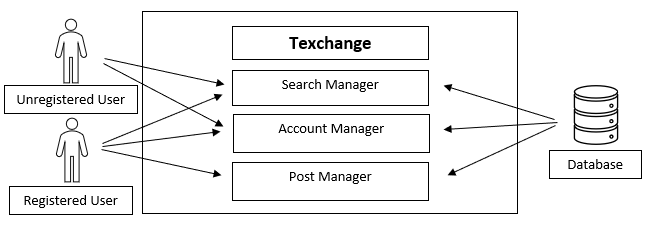
*[1] IEEE Standard 1016: Software Design Specification.*

*[2] IEEE Standard 830-1998 (Revision of IEEE Std 830-1993) IEEE Recommended Practice for Software Requirements Specifications.*

*[3] Stellman, A. and Greene, Jennifer (2006). Applied Software Project Management, O’Reilly.*

# Overall Description

## System Environment



*Figure 2.1 – System Environment*

*The Texchange has 2 Actors i.e. the Registered and Unregistered User, and one Database. Registered Users have access to the Search Manager where they can search for a textbook, access to the Account Manager where they can update their credentials, and access to the Post Manger where they can create a post or, oversee the status of their post. Unregistered Users have access to the Search Manager and Account Manager to create an account.*

## Functional Requirements Specification

*This section gives detailed descriptions of all the User Cases including a table that corresponds to each.*

### Create an Account

*User Case 1 deals with creating an account and the user/system interactions that occur. A user needs an account to create a post. The interactions will take place on the Create Account Page. They system will validate the inputs given and then save the data to the database.*

|  |  |
| --- | --- |
| **Name** | User Case 1. Create an Account |
| **Description** | New user creates account |
| **Rationale** | User needs to set up an account to create a post |
| **Users** | All types of users |
| **Preconditions** | User has access to internet |
| **Basic Course of Events** | -> User clicks on create account  <- System redirects user to create account page  -> User enters hof email address, cell phone number,      Username and password  <- User clicks create account  <- System checks user credentials are unique and accurate  <- System redirects user to confirmation page for user to check information  -> User checks information is valid  -> User clicks “Submit” button  <- System creates account and logs user in |
| **Alternative Paths** | 1) User enters invalid information  <- System displays Invalid Information message  2) User cancels on confirmation page  <- System redirects user to create account page |
| **Post Conditions** | A new account is created and stored in the system |

*Table* 2.2 – Create an Account

### Login

*User Case 2 allows a user to sign into the account they created in User Case 1. The interaction will take place on the Login Page. The system will check the database for an account that corresponds to the input given. Then the user will have access to their account and the ability to create a post.*

|  |  |
| --- | --- |
| **Name** | User Case 2. Login |
| **Description/Rationale** | User logs into account |
| **Rationale** | Access account information |
| **Users** | Users who already created an account |
| **Preconditions** | User has already created an account |
| **Basic Course of Events** | -> User clicks “Login in” button  <- System redirects user to login page  -> User enters login information  -> User clicks “Log In” button  <- System checks to make sure log in info matches info in database  <- System redirects user to site homepage |
| **Alternative Paths** | 1) User enters incorrect information  <- System Outputs Invalid Information message |
| **Post Conditions** | User logged in the system |

*Table 2.3 – Login*

### Search for Book

*User Case 3 allows a user to search for a book. The interaction will take place on the Search for Textbook Page. The user enters information related to the desired book, then the system will validate the inputs given and check the database for associated posts.*

|  |  |
| --- | --- |
| **Name** | User Case 3. Search for Book |
| **Description** | User searches for a book they want to buy |
| **Rationale** | User needs a way to find book |
| **Users** | All types of users |
| **Preconditions** | None |
| **Basic Course of Events** | -> User clicks on “Search” button  <- System redirects user to search page  -> User enters information of book (Author, ISBN, Title)  -> User clicks on the “Enter” button  <- System checks database for matching info; displays      matching books  <- System redirects user to related search page  -> User reviews posts and clicks on preferred provider to get info on seller |
| **Alternative Paths** | 1) No matching information  <- System displays messages stating no matches were found |
| **Postconditions** | None |

*Table 2.4 Search for Book*

### Create a Post

*User Case 4 allows a user to create a post. The user is required to a have an account and to be logged in to create a post. The interaction will occur on the Create a Post Page. The user will enter information related to the book they want to sell, then the user will validate the inputs given and save the data to the database.*

|  |  |
| --- | --- |
| **Name** | User Case 4. Create A Post |
| **Description** | User creates a post offering to sell their textbook |
| **Rationale** | User can display a book they want to sell |
| **Users** | All types of users |
| **Preconditions** | User is already logged in |
| **Basic Course of Events** | -> User clicks on “Sell” button  <- System redirects user to the create a post page  -> User enters in book title, author, ISBN, condition, price, and picture of cover.  -> User clicks “Create” button  <- System checks to make sure required fields are completed and sends info to database  <- System redirects user to a success page |
| **Alternative Paths** | 1) User enters invalid information  <- System will display Invalid Information message |
| **Postconditions** | A post is created that other users can see |

*Table* 2.5 Create a Post

### Delete a Post

*User Case 5 allows a user to delete a post they created. This requires a user to have an account, be logged in and have existing posts. The interaction will occur on the Post Manager Page. The user will choose the post they want to delete and the system will request the related data in the database be removed.*

|  |  |
| --- | --- |
| **Name** | User Case 5. Delete Post |
| **Description** | User deletes a post that they have created |
| **Rationale** | In case user needs to remove add before a buyer tries to purchase book |
| **Users** | All types of users |
| **Preconditions** | User is already logged in |
| **Basic Course of Events** | -> User clicks on “My Account”  <- System redirects user to My Account page  -> User clicks on “My Posts” button  <- System redirects user to My Posts page  <- User clicks “Delete” on the desired post  -> System removes the related post from the database and refreshes page displays remaining posts |
| **Alternative Paths** | 1) User has no posts  <- System displays a message stating no posts have been created |
| **Postconditions** | Desired Post is deleted from database |

*Table* 2.6 – Delete a Post

# Requirements Specifications

## External Interface

*This section includes a detailed description of all inputs and outputs from the system. Accompanying each description is a table for further analysis.*

### Account Information

*External Interface 1 describes all inputs relating to the Create an Account User Case. The inputs include Username, Email, Password and Major.*

|  |  |
| --- | --- |
| **Name of Input** | Input 1. Account Information |
| **Source of Input** | Account Registration Page |
| **Purpose** | Get account credentials |
| **Processing** | Database nonquery |
| **Tolerance** | Allows crash 1 out 100 runs |
| **Timing** | <= 1 second |
| **Relationship to other Inputs** | User Case 1. Create an Account |

*Table 3.1 – Account Information*

### Book Search Information Input

*External Interface 2 describes all inputs that relate to the Search for Book User Case. The inputs include Name, ISBN Number, Author, Course and Department.*

|  |  |
| --- | --- |
| **Name of Input** | Input 2. Book Search Information Input |
| **Source of Input** | Search for Textbook Page |
| **Purpose** | Search for textbook |
| **Processing** | Database nonquery |
| **Tolerance** | Allows crash 1 out 100 runs |
| **Timing** | <= 1 second |
| **Relationship to other Cases** | Output -1. Book Search Information Output  User Case 1. Create an Account |

*Table 3.2 – Book Search Information Input*

### Book Search Information Output

*External Interface 3 describes all outputs that relate to the Search for Book User Case. The outputs include Name, ISBN Number, Author, Course, Department, Price, Condition and seller information such as, Name and Email.*

|  |  |
| --- | --- |
| **Name of Output** | Output 1. Book Search Information Output |
| **Source of Input** | Result of Search Page |
| **Purpose** | Find related posts |
| **Processing** | Database query |
| **Tolerance** | Allows crash 1 out 100 runs |
| **Timing** | <= 1 second |
| **Relationship to other Inputs** | Input 2. Book Search Information Input  User Case 1. Create an Account |

*Table 3.3 – Book Search Information Output*

### Post Information

*External Interface 4 describes all inputs that relate to the Create a Post User Case. The inputs include Name, ISBN, Author, Department, Course Number, Condition and an Image.*

|  |  |
| --- | --- |
| **Name of Input** | Input 3. Post Information Input |
| **Source of Input** | Create Post Page |
| **Purpose** | Give details related to post |
| **Processing** | Database nonquery |
| **Tolerance** | Allows crash 1 out 100 runs |
| **Timing** | <= 1 second |
| **Relationship to other Inputs** | User Case 1. Create an Account |

*Table 3.4 – Post Information*

## Functional Requirements

*This section summarizes the major functions of the system, and of its components when accepting and processing inputs, or processing and generating the outputs.*

### Verify New Account

*Functional Requirement 2 handles the verification of user credentials and submission to database during the Create an Account User Case.*

|  |  |
| --- | --- |
| **Name** | FR-1. Verify New Account |
| **Summary** | System checks the new account info |
| **Rationale** | Confirm new account is unique and fills all requirements |
| **System behavior** | If Information is correct, nonquery the data  Else display message describing mistakes |
| **References** | User Case 1. Create an Account |

*Table 3.5 – Verify New Account*

### Verify User Account

*Functional Requirement 3 handles the verification of user credentials and retrieval from database during the Login User Case.*

|  |  |
| --- | --- |
| **Name** | FR-2. Verify User Account |
| **Summary** | System checks the login info |
| **Rationale** | Confirm user account matches already existing account |
| **System behavior** | If Information is correct, redirect user to home page  Else display message describing mistakes |
| **References** | User Case 2. Login |

*Table 3.6 – Verify User Account*

### Verify Book Search

*Functional Requirement 3 handles the verifications of inputs, and retrieval from the database during the Search for Book User Case.*

|  |  |
| --- | --- |
| **Name** | FR-3. Verify Book Search |
| **Summary** | System checks book search info |
| **Rationale** | Confirm all required fields are filled and correct |
| **System behavior** | If information is correct, query database for book and display results  Else display message describing mistakes |
| **References** | User Case 3. Search for Book |

*Table 3.7 – Verify Book Search*

### Verify Create Post

*Functional Requirement 4 handles the verification of inputs and submission to database during the Create a Post User Case*

|  |  |
| --- | --- |
| **Name** | FR-4. Verify Create Post |
| **Summary** | System checks Post information |
| **Rationale** | Confirm all required fields are filled and correct |
| **System behavior** | If information is correct, nonquery database the post  Else display message describing mistakes |
| **References** | User Case 3. Search for Book |

*Table 3.8 – Create a Post*

### Delete Post

*Functional Requirement 5 handles the verification of inputs and submission to database during the Delete Post User Case.*

|  |  |
| --- | --- |
| **Name** | FR-5. Delete Post |
| **Summary** | System checks removes post from database |
| **Rationale** | Free space in database and remove from available posts |
| **System behavior** | Nonquery the database and remove information related to post |
| **References** | User Case 6. Delete Post |

*Table 3.9 – Delete Post*

## Detailed Non-Functional Requirements

This section includes a detailed description of the quality attributes of the system, including performance, logical database design, security, robustness and reliability.

### Performance Requirements

The system will allow for a maximum of 5 simultaneous users. All requests will be handled in a maximum of 1 second and the system will allow for a maximum of 5 simultaneous transactions.

### Logical Database Requirements

*The system will use a SQL server provided by Hofstra Big Data Lab. The logical structure of the database is as follows, Sellers sell books, buyers buy books and sellers and buyers visit each other. Sellers and buyers have unique user names and each book has data fields ISBN, name.*

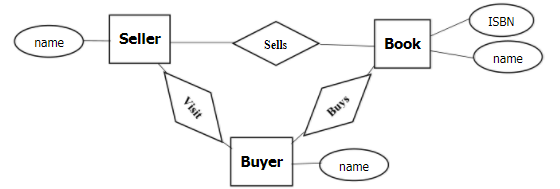


Figure 3.10 – Logical Structure of Database

### Security Requirements

All inputs will be validated, and validation failures will result in input rejection. The accepted inputs will be encrypted before entering database. Access to database will be restricted to only authorized users

### Reliability Requirements

*The system allows crash 1 out of 100 runs.*

### Robustness

*The system should handle error conditions gracefully, without failure*

# Engineering and Realistic Constraints

*This section includes constraints imposed by hardware, software, communication standards, hardware limitations.*

## Engineering Standards

*This section lists all protocols used in conjunction with this project.*

1. *Communication between web browser and server will be done using the http standard [W3C Server-Sent Events]*
2. *Web pages will be created following the HTML 5 standard [W3C HTML 5.1 2nd Edition]*
3. *Software Design Descriptions follow the protocol [IEEE 1016-1998]*
4. *Database insertion and retrieval will follow the standard [W3C Data on the Web Best Practices]*

## Realistic Constraints

*This section includes constraints that limit the development and implementation of the system.*

### Safety Constraint

### *Buyer and seller meet ups could lead to potentially dangerous encounters. Establish all meetings in large public areas. Travel with a friend when possible.*

### Ethical Constraint

### *All private account information will be hidden from other users*

### Social Constraint

### *User must be a Hofstra student to create an account*

# Project Design

## Overview of System Components

*This section includes the system diagram, a short description of each component and of its interactions with other components*

### User’s Browser

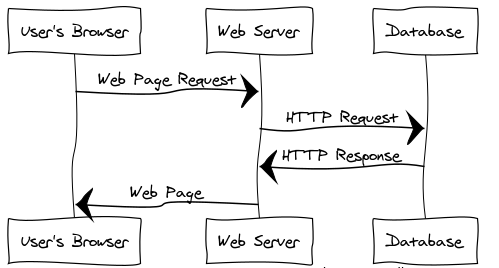
*The User’s Browser provides the interface of the system. Here the user has access to all the components of the application. As the user interacts with the application requests will be sent to the web server and subsequently the database.*

### Web Server

*The web server will host all the components and modules that the user’s browser will interact with. In addition, the web server provides connection to the database.*

### Database

*The database will store all data required by the System. Communication between the Web Server and Database is handled via HTTP protocols. [4]*



*Figure 4.1 – Sequence Diagram*

## Structure and Relationship

*This section describes in more detail what each component will do and how it will interact with or depend on other components.*

### User’s Browser

### *The users browser provides the interface of the entire application. The user would load the main web page on their browser. The user would be able to create an account, login, and create or search book postings. The web page would be communicating with the web server, querying requests such as verify new account, verify user account, verify book search, delete post. The web server would then return the requested pages to the user’s browser in order to complete the tasks.*

### Web Server

*The web server will receive requests from the web page loaded onto the user’s browser. The web server hosts all the components and modules that the user’s browser will interact with. Create an account, login, search for a book, enter payment information, create a post, and delete a post will be sent to the user’s browser by the web server. The web server will send information entered by the user to the database/database server. The web server will also receive information requested from the database/database server and send it to the user’s browser.*

### Database/Database Server

*The database will store all user’s account information and all book posts. The database will make receive all account information and book posts from the web server. The database server will respond to queries from the web server. Verify user account, new account, book search, delete post and create post will be done by the database*.

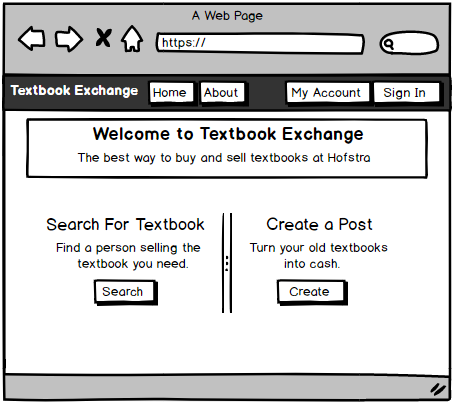
## User Interface

*This section describes the design of the system’s user interface including drawings to display the interface design, screens, buttons etc. [5]*

### Default Page

*This page is displayed when a user visits the site. Here the user has access to all*

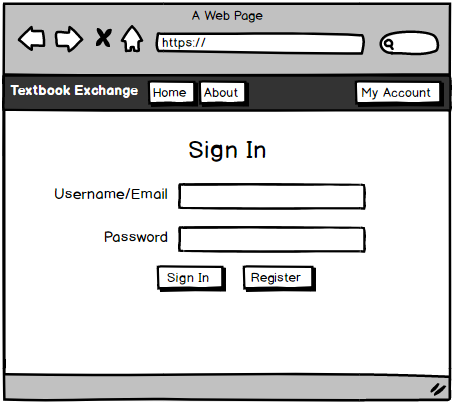
*other pages*



*Figure 4.2 – Default Page*

### Login Page

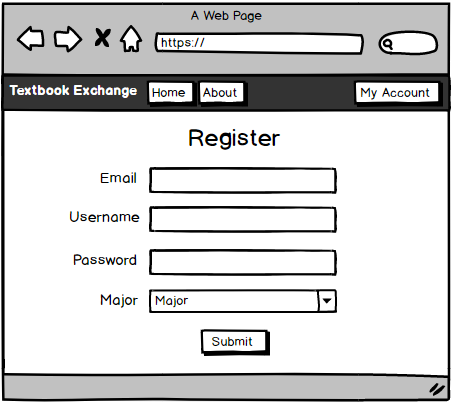
*This page allows a user to log in. The user enters their credentials and then a http request is sent to the database to search for the correct user. This page also has a button called register for new users to create an account, which will redirect them to the Create Account Page. Related to User Case 1*



*Figure 4.3 – Login Page*

### Create Account Page

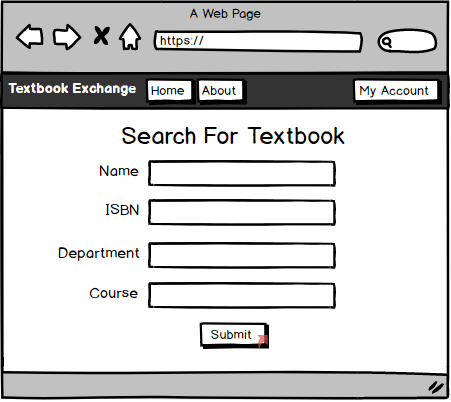
*This page allows new users to create an account. The user will fill in the required fields and, if all fields are filled correctly, then a http request is sent to the database to save the new users information. Related to User Case 2*



*Figure 4.4 – Create an Account Page*

### Search for Textbook Page

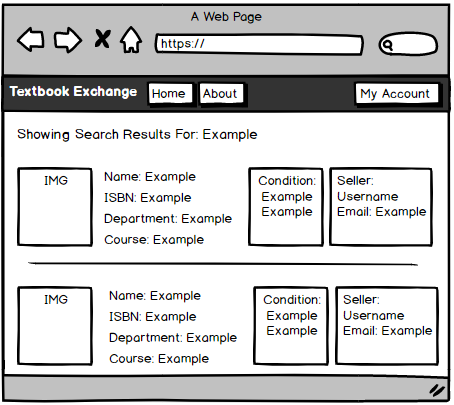
*This page is accessed through the search button on the default page. The user must fill at least one of the fields to get results. The user then clicks submit button and is redirect to the Result of Search Page while an http request is sent to the database to retrieve the related information. Related to User Case 3*



*Figure 4.5 – Search for Textbook Page*

### Result of Search Page

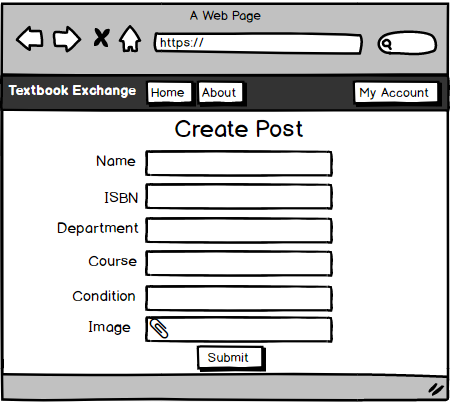
*This page displays all available posts related to results of Search for Textbook Page. The posts give details about the name, ISBN, department, course, and condition of the textbook. The seller’s information is displayed for user to contact them. Related to User Case 3*



*Figure 4.6 – Result of Search Page*

### Create Post Page

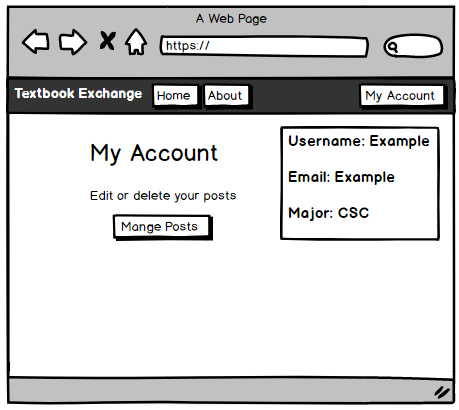
*This page is accessed through the create button on the default page. The user must fill all required fields to get results. User then clicks submit button and is redirect back to home page while an http request is sent to database to save the information. Related to User Case 4*



*Figure 4.7 – Create Post Page*

### My Account Page

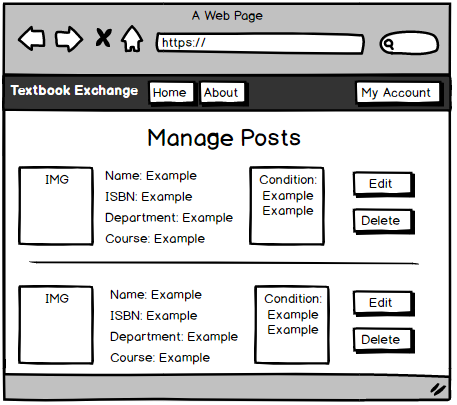
*This page is accessed through the My Account button on the default page. The user can see their user credentials and access the post manager by clicking the manage posts button. Related to User Case 5*



*Figure 4.8 – My Account Page*

### Post Manager Page

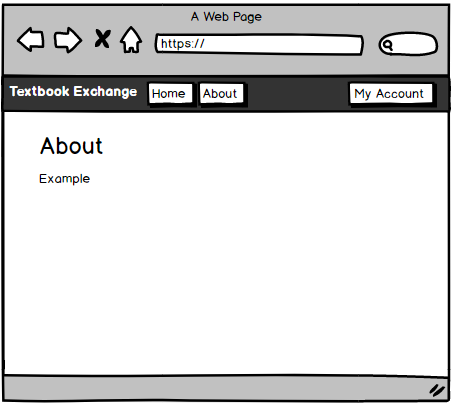
*Accessed through the My Account Page. Here the user can edit or delete all available posts. Related to User Case 5*



*Figure 4.9 – Post Manager Page*

### About Page

*Gives details about the website, how it works, the creation, etc.*



*Figure 4.10 – About Page*

## Detailed Component Description

*This section provides further description of each component of the system and includes a corresponding table for additional reference.*

### Web Server

|  |  |
| --- | --- |
| **Identification** | SC-1: Webserver |
| **Type** | Client Server |
| **Purpose** | The web server hosts all the components and modules that the user’s browser will interact with. |
| **Function** | The web server will act as the middle man between the data server and the user’s browser. The user will interact with the user’s browser which may involve the need for sending or retrieving data from the database. In this case the web server will instantiate a HTTP request for all required interactions. |
| **Dependencies** | This component will interact with the Dataserver using the HTTP Protocol and the Userserver. Each retrieval and query will take a noticeable amount of time, expect 1 second each. |
| **Interfaces** | The user will not be able to interact with the web server directly, thus there will be no interface available. All request to the data server will be through the HTTP protocol, utilizing both GET and POST techniques. |
| **Cross-reference** | All user cases |

*Table 5.1 – Web Server*

### User’s Browser

|  |  |
| --- | --- |
| **Identification** | SC-2: Userserver |
| **Type** | Server-Side Server |
| **Purpose** | This is where the user will interact with the application. All pages will be displayed through the Userserver. The Webserver will then communicate with the Dataserver and relay any information back to the Userserver if needed. |
| **Function** | What the component does, the transformation process, the specific inputs that are processed, the algorithms that are used, the outputs that are produced, where the data items are stored, and which data items are modified. |
| **Dependencies** | This component will interact with the Webserver. Each retrieval and query will take a noticeable amount of time, expect 1 second each. |
| **Interfaces** | All the web pages will be displayed through the Userserver. For a detailed description of the interface check section 5.3 User Interface. |
| **Resources** | The Userserver will be constructed using ASP.Net, part of the .NET framework, and implemented in Visual Studio. |
| **Cross-reference** | All user cases |

*Table 5.2 – User’s Server*

### Data Server

|  |  |
| --- | --- |
| **Identification** | SC-3: Dataserver |
| **Type** | Database |
| **Purpose** | The purpose of the Dataserver is to save information about user credentials and book information. Function and performance requirements implemented by the design component, including derived requirements. Derived requirements are not explicitly stated in the requirements section, but are implied or adjunct to formally stated requirements. |
| **Function** | The Dataserver is responsible for saving and retrieving data related to the user and user’s posts. The database will make use of the HTTP Protocol for receiving and sending information. |
| **Internal**  **Structure** | The Dataserver will be constructed with 2 tables one for saving the user’s account credentials and the other for saving the related information about the created posts. |
| **Dependencies** | This component will interact with the Webserver using the HTTP Protocol. Each retrieval and query will take a noticeable amount of time, expect 1 second each. |
| **Interfaces** | The user will not be able to interact with the web server directly, thus there will be no interface available. All request to the data server will be through the HTTP protocol, utilizing both GET and POST techniques. |
| **Resources** | The Dataserver will be constructed using Microsoft’s SQL Explorer, part of the .NET framework, and implemented in Visual Studio |
| **Cross-reference** | All user cases |

*Table 5.3 – Data Server*

## Reuse and Relationships with other Products

*This section describes all frameworks that are used in conjunction with the system.*

### ASP.NET 4.5

*ASP.NET is a powerful tool for creating dynamic websites developed by Microsoft.*

### Bootstrap 4

*Bootstrap is an open sourced front-end framework developed by Twitter. It will be used to assist in the design of the user interface*

## Design Decisions and Tradeoffs

*This section describes the reasoning behind our design decision and any decisions that did not follow through.*

### Design Purpose

*The user interface is an intuitive design that allows the user to access either all or most of the site on any page*

### Abandoned Designs

Originally the system would allow the user to create a post stating they’re in the market for a certain textbook. This design was removed because a simpler intuitive design is valued.

## Resource List

*This section lists all resources that were used to create this project.*

### Beginning ASP.NET 4.5 in C# by Matthew MacDonald [2]

*This book is the first step to becoming a successful ASP.NET developer*

### Pro ASP.NET 4.5 in C# by Matthew MacDonald [3]

*This book improves your skills further as an ASP.NET developer*

## Resource Course List

*This section lists all courses that assisted throughout the creation of this project*

### Software Engineering

*This course covers the development process of software*

### Web Applications

*This course covers the foundational aspects of web applications*

# Project Plan

*This section gives a detailed description of plan of the project going forward, including a list of tasks that need to be completed for the project to be finished.*

## List of Tasks

*This section gives a detailed description of all tasks that will result in the development of the system. Included are the efforts needed to implement the tasks and people who can complete them.*

1. **Name**: Default Page

**Description**: Check section 5.3.1

**Resources**: asp.net, html

**People**: Visual Studio 15, Gordon Chris

**Effort**: 4 Hours

1. **Name**: Login Page

**Description**: Check section 5.3.2

**Resources**: Visual Studio 15, asp.net, html

**People**: Gordon Chris

**Effort**: 2 Hour

1. **Name**: Create Account Page

**Description**: Check section 5.3.3

**Resources**: Visual Studio 15, asp.net, html

**People**: Gordon Chris

**Effort**: 2 Hour

1. **Name**: Search for Textbook Page

**Description**: Check section 5.3.4

**Resources**: Visual Studio 15, asp.net, html

**People**: Gordon Chris

**Effort**: 3 Hour

1. **Name**: Result of Search Page

**Description**: Check section 5.3.5

**Resources**: Visual Studio 15, asp.net, html

**People**: Gordon Chris

**Effort**: 2 Hour

1. **Name**: Create Post Page

**Description**: Check section 5.3.6

**Resources**: Visual Studio 15, asp.net, html

**People**: Gordon Chris

**Effort**: 2 Hour

1. **Name**: My Account Page

**Description**: Check section 5.3.7

**Resources**: Visual Studio 15, asp.net, html

**People**: Gordon Chris

**Effort**: 1 Hour

1. **Name**: Post Manger Page

**Description**: Check section 5.3.8

**Resources**: Visual Studio 15, asp.net, html

**People**: Gordon Chris

**Effort**: 1 Hour

1. **Name**: About Page

**Description**: Check section 5.3.9

**Resources**: Visual Studio 15, asp.net, html

**People**: Gordon Chris

**Effort**: 1 Hour

1. **Name**: Users Table

**Description**: Create a user database with relevant data fields

**Resources**: Visual Studio 15, SQL Server Object Server

**People**: Gordon Chris

**Effort**: 3 Hours

1. **Name**: Posts Table

**Description**: Create a posts table with relevant data fields

**Resources**: Visual Studio 15, SQL Server Object Server

**People**: Gordon Chris

**Effort**: 3 Hours

1. **Name**: Search for Textbook

**Description**: textbook information will be sent from the web browser to the server and checked against the data in the database server

**Resources**: web browser, database

**People**: Gordon Chris

**Effort**:5 Hours

1. **Name**: Create a Post

**Description**: A new post will be sent from the web browser to the database server and **added** to the database

**Resources**: web browser, web server, database server, programming languages

**People**: Gordon Chris

**Effort**:5 Hours

1. **Name**: Edit Post

**Description**: Instructions to edit a post are sent from web browser to web server and information is updated in database by the database server

**Resources**: web browser, web server, database server, programming languages

**People**: Gordon Chris

**Effort**:5 Hour

1. **Name**: Verify User Login

**Description**: Username and password are sent from the web browser to the web server and then checked against the data in the database server

**Resources**: web browser, web server, database server, programming languages

**People**: Gordon Chris

**Effort**:2 Hours

1. **Name**: Verify New Account

**Description**: Username and password are sent from the web browser to the web server and then checked against the data in the database server

**Resources**: web browser, web server, database server, programming languages

**People**: Gordon Chris

**Effort**:2 Hour

1. **Name**: Delete a Post

**Description**: Instructions are sent from web browser to web server and then data is **removed** and updated in database server

**Resources**: web browser, web server, database server, programming languages

**People**: Gordon Chris

**Effort**:3 Hour

1. **Name**: Obtain Domain Name [1]

**Description**: Create a domain site

**Resources**: GoDaddy.com

**People**: Gordon Chris

**Effort**:2 Hours

1. **Name**: Push Application to Domain Site [1]

**Description**: Upload the existing application to the domain site

**Resources**: GoDaddy.com

**People**: Gordon Chris

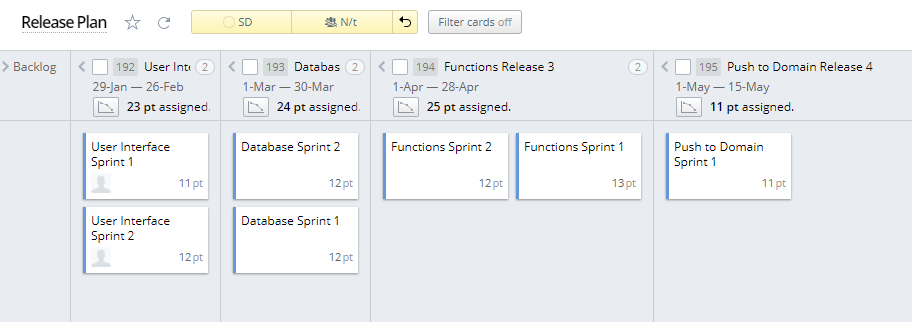
**Effort**:1 Hour

## Project Schedule

*This section includes a schedule detailing all tasks and their relative production rate within the timeline, created using TargetProcess.com.*

### Release Plan

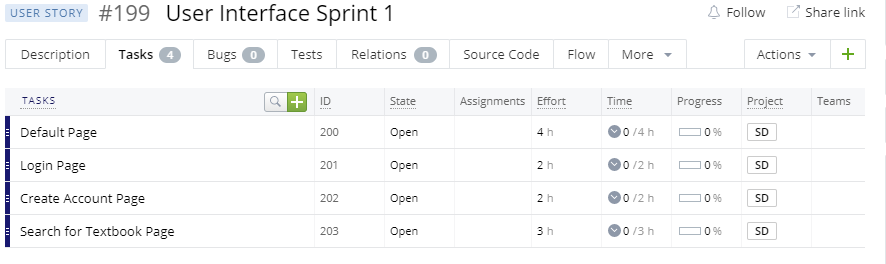
*Here you can see the release plan for the project. In total there will be 4 releases, the first 3 lasting 1 month and the last lasting 15 days. The first 3 releases have 2 sprints and the last has 1 sprint each lasting 15 days.*



*Table 6.1 – Release Plan*

### Task Plan

*Within each sprint there are tasks relating to section 6.1 List of Tasks. Each task has a set amount of effort or hours, and a developer, either Gordon or I.*



*Table 6.2 Task Plan*

## Risk Plan

*This section includes a list of the major risks taken during the project plan and a sequence of actions to alleviate its outcome.*

### Performance

*The system needs to meet industry standards as well as user expectations on performance. Consideration must be given to benchmarks and threshold testing throughout the lifespan of the project to guarantee an efficient product.*

### Communication & Organization

*Throughout the project, it is imperative that all cooperating players of are in sync. To ensure orderly collaboration weekly meetings will be instantiated and a project management software, such as TargetProcess.com, will be used.*

## Estimated Financial Budget

*This section includes a detailed description of all items needed to purchase in conjunction with the project.*

### Domain Site [1]

*Purchased from GoDaddie.com costing $14.99 per year*

### Beginning ASP.NET 4.5 in C# by Matthew MacDonald [2]

*Purchased from Amazon.com costing $39.98*

### Pro ASP.NET 4.5 in C# by Matthew MacDonald [3]

*Purchased from Amazon.com costing $59.97*

|  |  |
| --- | --- |
| ***Item*** | ***Estimated Cost*** |
| *Domain Site* | *$14.99* |
| *Beginning ASP.NET* | *$39.98* |
| *Pro ASP.NET* | *$59.97* |
| *Total cost* | *$114.94* |

*Table 6.1- Estimated Financial Budget*

## Teamwork Plan

*This section describes how the team will work together to complete the project. To ensure orderly collaboration weekly meetings will be instantiated and TargetProcess.com, a project management software tool, will be used. This software follows an AGILE model for project management. Sprints lasting a duration of 2 weeks each will be created. Each sprint will be comprised of tasks described in section 6.1. Releases lasting 1 month will contain 2 sprints. A period of reflection will thus ensue. In total an estimated 6 sprints and 3 releases will occur.*

**Citations**

*[1] GoDaddy.com, 1 Jan. 2018,* [*https://www.godaddy.com/*](https://www.godaddy.com/)

*[2] Amazon.com, 1 Jan. 2018,* [*https://www.amazon.com/Beginning-ASP-NET-Experts-Voice-Net-ebook/dp/B00FB2PZLG/ref=sr\_1\_fkmr0\_2?ie=UTF8&qid=1515647907&sr=8-2-fkmr0&keywords=asp.net+beginners+matthew*](https://www.amazon.com/Beginning-ASP-NET-Experts-Voice-Net-ebook/dp/B00FB2PZLG/ref=sr_1_fkmr0_2?ie=UTF8&qid=1515647907&sr=8-2-fkmr0&keywords=asp.net+beginners+matthew)

*[3] Amazon.com, 1 Jan. 2018,* [*https://www.amazon.com/Pro-ASP-NET-4-5-Dan-Mabbutt-ebook/dp/B00ACC5Z4M/ref=sr\_1\_fkmr0\_1?ie=UTF8&qid=1515647907&sr=8-1-fkmr0&keywords=asp.net+beginners+matthew*](https://www.amazon.com/Pro-ASP-NET-4-5-Dan-Mabbutt-ebook/dp/B00ACC5Z4M/ref=sr_1_fkmr0_1?ie=UTF8&qid=1515647907&sr=8-1-fkmr0&keywords=asp.net+beginners+matthew)

*[4] WebSequenceDiagrams.com, 23 Nov. 2017,* [*https://www.websequencediagrams.com/*](https://www.websequencediagrams.com/)

*[5] Balsamiq.com, 10 Dec. 2017,* [*https://balsamiq.com/*](https://balsamiq.com/)