CSC 4350: Software Engineering

Fall 2016 Semester

October 23, 2016

Document #5 – Rational

Team Name: Real Team 6

Application Name: Booki

Team 6 Members:

Anthony Jones

McKenzie Rochester

Daniel Campbell

Jeff Sikala

Jordan Smith

Table of Contents

Rational3

Requirements Trace Matrix (RTM)6

Software Architecture7

Work Share Document (WSD)8

Gantt Chart9

Dictionary10

**Rational**

Bookkeeper is an application that is designed to allow students within their local area to be able to sell books to another student by direct contact. Currently students have limited options to sell back their expensive books that they have purchased over school semesters. Students are only left with either selling their books to online stores such as Amazon and Chegg, or they are left selling back their books to the local bookstore for a very low price. We are providing a way for students to get back a good portion of their money by offering a way for students to communicate with each other to sell books privately. Bookkeeper does not facilitate in any sales, rather it gives users the ability to post and communicate with each other.

As an added benefit, users will be able to post advice or any other information on forums by either course or major. It is another avenue for students to be able to engage in their college communities and to be able help an encourage their fellow students.

**Use Case Rational**

**UC\_01\_AccountLogin**: Booki needs a login section in order to allow students to see their account and content. If we did not have a login, it would be impossible to allow student accounts to be personally tailored to them.

**UC\_02\_AccountRegistration**: An account creation process must be implemented in order to create new users for the system.

**UC\_03\_UpdateAccount**: Students should have the option to update their information so that they can remain up to date on their personal information.

**UC\_04\_BookPost**: Booki needs a book posting option so that students are able to post the books that they need to sell through the web application.

**UC\_05\_PrivateMessenger:** The application must have a private messaging system so that buyers and sellers can trade their items. Since we will not be facilitating monetary transactions, students need to have the feature to talk to each other in order to complete private transactions.

**UC\_06\_TransactionSystem**: Booki needs a transaction system in order to process what book posts are still available. It also helps the experience of trading books by giving seller’s a rating.

**UC\_07\_SearchMenu**: It is necessary to implement a search menu so that student will be able to find the books that they need. Also this helps sellers sell their items by making them available to all students.

**Object Rational**

**App.py** - the App.py object gathers input from the user and makes changes to the database and the display of the app.

**Form.py** - parent class for the various forms on the website

**Login\_Form** - user writes in information and it is then put into the database

**Registration\_Form** - user input information necessary for profile creation and the database is updated with the new information

**Bookpost\_form** - user input information necessary for posting a book on the website and the database is updated with new information

**Survey\_form** - ratings and comments form for the buyer’s experience with the seller

**Search\_form** - form for user input to allow searches for different books

**Model.py** - parent class for the various model’s for the app.

**User** - handles creating and storing the user model to the database.

**Book\_post** - handles creating and storing the book post to the database.

**Message** - handles creating and storing messages to the database.

**Survey** - handles creating and storing surveys to the database.

**Requirements Trace Matrix (RTM)**

|  |  |  |  |
| --- | --- | --- | --- |
| Entry Number | Paragraph Number | RTM Entry | Type |
| 1 | 2.1 | User is prompted to login using email and password | SW |
| 2 | 2.2 | If the User does not have an account, then the user will be prompted to create a new account. | SW |
| 3 | 2.3 | The web application will redirect a new user to an account registration page when they are directed to make one. | SW |
| 4 | 2.4 | The user will be taken to their timeline page where they can view book postings, transactions and messages. | SW |
| 5 | 3.1 | Users are about to create book postings when they are ready to sell their items. | SW |
| 6 | 3.2 | Book Transactions will send a notification and email to both the potential seller and buyer when a transaction is complete. | SW |
| 7 | 3.3 | The application will send a notification and email to the potential seller and buyer when a transaction is complete. | SW |
| 8 | 3.4 | The application will allow users to update their account information via an update page. | SW |
| 9 | 4.1 | The application will ask the users to update the status of the transaction one hour after completion. | SW |
| 10 | 4.2 | A survey will be initialized to ask the buyer how satisfied they were with the seller. | SW |
| 11 | 4.3 | If the survey is not complete, then the system will not allow another transaction until it has been completed. | DR, SW |
| 12 | 4.4 | A search menu will be used to find books by specific categories. The search option will be able to search by book title, ISBN, course, and major. | SW |
| 13 | 4.5 | The application will run on a Linux server. | HW |
| 14 | 4.6 | The application will utilize SQLite as its database system. | HW, SW |

**Software Architecture**

Booki will be a web application that will utilize a relational database to store all of its information. The web application will be built using the Python language. Python is a very popular choice for back end web applications due to its many libraries for web development. We decided to use Flask as the Python back end because it is lightweight and fast to use implement an application. Flask has many of the requirements that we were looking for in order to create Booki.

**Work Share Document (WSD)**

**Daniel Campbell**

Role: Front-End Programmer

**McKenzie Rochester**

Role: Back-End Programmer

**Jeff Sikala**

Role: Database Programmer

**Jordan Smith**

Role: Technical Writer

**Anthony Jones**

Role: Team Leader/Client/Tester

**Project Timeline (Gantt Chart)**

October 23, 2016

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Task Description | Duration (Days) | Start Date | End Date |
| Introduction | | | | |
| 1 | Team Creation | 3 | 23-Aug | 25-Aug |
| 2 | Identify Team Roles | 3 | 25-Aug | 27-Aug |
| 3 | Identify Project Idea | 4 | 27-Aug | 1-Sept |
| 4 | Work Sign Document (WSD) | 5 | 28-Aug | 1-Sept |
| Requirement Elicitation | | | | |
| 5 | Problem Statement | 3 | 13-Sept | 17-Sept |
| 6 | Requirements Trace Matrix (RTM) | 2 | 17-Sept | 18-Sept |
| 7 | Gantt Chart | 2 | 17-Sept | 18-Sept |
| 8 | Dictionary | 2 | 17-Sept | 18-Sept |
| 9 | Project Rational | 2 | 17-Sept | 18-Sept |
| System Analysis | | | | |
| 10 | Updated Problem Statement | 3 | 19-Sept | 21-Sept |
| 11 | Update RTM, Gantt Chart, Dictionary | 3 | 22-Sept | 24-Sept |
| 12 | Use Cases | 9 | 24-Sept | 2-Oct |
| 13 | Interaction Diagrams | 1 | 2-Oct | 2-Oct |
| 15 | Horizontal Prototype | 1 | 2-Oct | 2-Oct |
| Object Design | | | | |
| 16 | Object Design | 2 | 12-Oct | 14-Oct |
| 17 | Function Point Cost Analysis | 1 | 15-Oct | 16-Oct |
| 18 | Update Gantt Chart | 1 | 15-Oct | 16-Oct |
| Rational | | | | |
| 19 | Rational | 2 | 17-Oct | 19-Oct |
| 20 | Category Interaction Diagrams | 5 | 19-Oct | 23-Oct |
| 21 | Update Gantt Chart | 1 | 23-Oct | 23-Oct |

**Dictionary**

* Web Application: A client server software application in which the client runs in a web browser.
* Database: A location where data is stored. It uses a language called SQL to issue commands to it. In our project we will be using SQLite as our database.
* URL: Uniform Resource Locator. Provides an address to a location on the world wide web.
* Redirected: Takes a user from one website to another website typically through a hyperlink.
* Query: A command for SQL databases.
* App: Short for application.
* Web Browser: A software application for retrieving, presenting, and traversing informational resources on the world wide web.
* Course: A class that is offered by a university or college.
* Web Browser Credentials: Information that is temporarily stored within the web browser in order to keep a user logged into the system for a period of time.
* Model: handles the database portion of the application. Also adds functionality to any objects.