Team Project CEN 4010

120 points

You will be creating an API Service to support an online web application bookstore which targets a particular niche in technology. The application, named **Geek Text** (think of Thinkgeek meets Barnes and noble) will need to support the following features:

Feature ID	**Feature**	**Benefit**
1	Book Browsing and Sorting	Users will have a simple and enjoyable way to discover new books and Authors and sort results. API Actions: □ Retrieve List of Books by Genre □ Retrieve List of Top Sellers (Top 10 books that have sold the most copied) □ Retrieve List of Books for a particular rating and higher □ Retrieve List of X Books at a time where X is an integer
2	Profile Management	Users can create and maintain their profiles rather than enter in their information each time they order API Actions: □ Create a User with username(email), password and optional fields (name, email address, home address) □ Retrieve a User Object and its fields by their username □ Update the user and any of their fields except for mail □ Create Credit Card that belongs to a User and Retrieve a list of cards for that user
3	Shopping Cart	Users can manage items in a shopping cart for immediate or future Purchase API Actions: □ Create a shopping cart instance for a user. Shopping cart must belong to a user. □ Update the shopping cart with a book. □ Retrieve the list of book(s) in the shopping cart. □ Delete a book from the shopping cart instance for that user.
4	Book Details	Users can see informative and enticing details about a book API Actions: □ An administrator must be able to create a book with the book ISBN, book name, book description, price, author, genre, publisher , year published and copies sold. □ Must be able retrieve a book's details by the ISBN □ An administrator must be able to create an author with first name, last name, biography and publisher

		□ Must be able to retrieve a list of books associate with an author
5	Book Rating and Commenting	Users can rate AND comment on books they've purchased to help others in their selection API Actions: □ Must be able to create a rating for a book by a user on a 5 star scale with a datestamp □ Must be able to create a comment for a book by a user with a datestamp □ Must be able to retrieve a list of ratings and comments sorted by highest rating □ Must be able to retrieve the average rating for a book
6	Wish List Management	Users can create and have 3 different wish lists which can have books moved to from the primary list. API Actions: □ Must be able to create a wishlist of books that belongs to user and has a unique name □ Must be able to add a book to a user's wishlist □ Must be able to remove a book from a user's wishlist into the user's shopping cart □ Must be able to list the book's in a user's wishlist

The grading will be split into team components and group components. Each feature will be owned by a team member. Each member will be graded for the completion and implementation of the feature that they own. This means that although you will earn an overall team grade from the team components, each person in the team will be responsible for a feature being done. If the feature is no completed, the individual who owns the feature will not receive the implementation portion of the grade.

How will the group develop the APIs?

The team will use scrum, which is an agile methodology used to develop software. You will need to review Lecture 1 Scrum to become familiar, but here is a summary:

The team will develop the application in timeboxed intervals known as Sprints. Each **Sprint** has a start date and an end date. At the beginning of the sprint, the team conducts a **sprint planning** meeting, in which they discuss and estimate what is the functionality that they will implement during that Sprint. The functionality will be implemented by breaking down the features into **user stories**. A user story is a description what the software must do, the benefit and the acceptance criteria. During the sprint, the team will meet through a meeting known as the **standup or daily scrum**, in which each team member mentions what they are working on, what they have accomplished and any impediments. It is expected that the team does at least 2 standup meetings each week. Finally, after the sprint has finished, the team hosts a **Sprint Review**, a meeting in which they demo (show) the work that was completed during the sprint. After the Review, the team performs a **Sprint Retrospective (Retro)** in which they discuss what has gone well and what can be improved. It's a chance for the team to incorporate improvements for the next Sprint. After completing the Sprint Retrospective, they will then schedule the sprint planning meeting for the next sprint and the process starts again.

Roles:

The teams will consist of 6 people which will be the following during each sprint:

Up to 6 Developers: The developers will be responsible for creating their feature and writing their own users stories. Everyone will be developing and testing during all sprints.

1 Product Owner: This will be an alternating role each sprint. This person will review the backlog of all the user stories written by developers and make sure they meet the requirements of the features. They will be responsible for asking any questions for that sprint regarding product functionality to the instructor. This will be a shared role as it is expected that they also pull in some stories for development.

Each sprint there must be one volunteer to be the product owner.

1 Scrum Master: Another alternating role each sprint. The scrum master will be the lead person who oversees removing any impediments and bringing up any issues to the instructor during the sprint. This will be a shared role as it is expected that they also pull in some stories for development.

The team will have a different Product Owner and Scrum Master in each sprint, so multiple will have an opportunity to work under each role.

Each sprint there must be one volunteer to be the scrum master.

Velocity:

Velocity is the measurement of how much work will be done. We will use hour units to estimate our work. So, if a person is working on a given user story that will take them 6 hours to achieve, this will be the estimated effort of their user story:

A User Story is a description of the development task that has to be done, so the teams will have multiple user stories and hour estimate to each user story.

Each team will calculate their capacity (how much work they can do) for the sprint in the following manner:

Because this is a summer B class with a short timeline, it is expected that you place as much velocity needed to complete the capabilities of your feature. We only have 6 weeks!

Timeline:

The project will be completed in 5 sprints, each lasting 2 weeks. Sprint 1 will start on the 3rd week of class.

- First Day of Class: 6/22
- Live Kickoff Meeting with Instructor **6/27**
- Sprint 1 6/22 - 7/5 Sprint Review **7/5** (Record your demo / Post documents by 10pm in your team forum)
- Sprint 2 7/6– 7/19 Sprint Review **7/19** (Record your demo / post documents by 10pm in your team forum)
- Sprint 3 7/20- 7/31 Live System Demo **8/1 and 8/2**
- UML Diagrams Due **7/27**

Note: Your demo must be recorded as a group and should be no more than 8 minutes. All video files must be uploaded in a compressed format (mp4). If you miss attending/recording **2** system demos, you will NOT be eligible for the group component (65 points).

Deliverables

There will be multiple deliverables for this project, but what I value most is a working project which will be demoed to me though online meetings. When the project is completed, you will have.

1. Source Code for Project in GitHub

2. Documents for Scrum Ceremonies
3. UML Diagrams for project
4. Recordings of Reviews (if they cannot be done live)
5. Final Demo completed (Will happen after last sprint is completed)

The source code should be checked into Github. Please make sure everyone has an account.

Scrum Documents

During each Sprint, each team will perform the following scrum ceremonies:

- **Daily Standup:** A quick 5-minute meeting in which the team members discuss what they accomplished the since they last met and what they plan to work on. Also, any impediments are brought up. You are required to have 2 standups every week. The team can determine what day and time they will meet.
- **Sprint Planning:** A meeting to determine which of the backlog items will get pulled into the upcoming sprint. This document will keep the details of the user stories that will be worked on during the sprint.
- **Sprint Review:** A meeting to showcase the completed stories to the stakeholders. This is the only meeting that the I will attend. All team members should attend this meeting. If they cannot attend, they need to record a video of the progress of their feature.
- **Sprint Retrospective:** A Meeting to determine what went right during the sprint and what can be improved.

During each ceremony, the team must keep notes and complete the scrum documents templates in each sprint. The documents will be uploaded to your group's forum during each sprint.

- ➔ Team 1 Discussion
  - ○ Sprint 1
    - ■ Standup document
    - ■ SprintPlanning document
    - ■ SprintReview document
    - ■ SprintRetro document
  - ○ Sprint 2…
- ➔ Team 2 Discussion
  - ○ Sprint 1…..

The templates for the documents will be placed under module 1 in Canvas for the teams to use.

The scrum master in each sprint must download the templates and upload the documents for every sprint.

Instructor Grading Criteria:

Feature Implementation (35 points) – Individual Grade Component

- Profile Management _____
- Book Browsing and Sorting _____
- Book Details _____
- Shopping Cart _____
- Book Rating and Commenting _____

Please review Feature Checklist for grade breakdown.

Scrum Execution (45 points) – Group Grade Component

- Sprint 1 Ceremonies in Folder _____ (6 for planning, 4 for review, 3 retro, 2 for standup)
- Sprint 2 Ceremonies in Folder _____ (6 for planning, 4 for review, 3 retro, 2 for standup)
- Sprint 3 Ceremonies in Folder _____ (6 for planning, 4 for review, 3 retro, 2 for standup)

15 points per sprint

Total: 45

UML Diagrams (19 points) – Individually Graded Component

Each Feature will have to provide:

- **Use Case**
- **Sequence Diagram**
- **Class Diagram**

1) Profile Management _____
2) Book Browsing and Sorting _____
3) Book Details _____
4) Shopping Cart _____
5) Book Rating and Commenting _____

Use Case Diagram (Worth 6 points):

- Must contain a business actor (2 pts)
- Must contain a use case/business case name with flow (2 pts)
- Must identify a system boundary (2 pts)

Reference: https://www.uml-diagrams.org/use-case-diagrams.html

https://www.lucidchart.com/pages/uml-use-case-diagram

Sequence Diagram (Worth 7 points):

- Must contain actor (1 pts)
- Must contain activation box (2 pts)
- Must contain lifeline (2 pts)
- Must contain messages with flow (2pts)

Reference: https://www.uml-diagrams.org/sequence-diagrams.html

https://www.lucidchart.com/pages/uml-sequence-diagram

Class Diagram (worth 6 points):

- Must contain class name, attributes and methods (2 pts)
- Must contain associations (2 pts)
- Must contain multiplicity (2 pts)

Reference: https://www.uml-diagrams.org/class-diagrams-overview.html

https://www.lucidchart.com/pages/uml-class-diagram

Integrated Product Demos – Group Graded Component

Feature is functional on an integrated environment will all other team source code and Github clearly demonstrations contributions to master branch ____ (0 no, 20 yes)

Score Break Down

Individual Piece		Group Piece	
Feature Implementation	: 36	Scrum Execution	: 45
UML Diagrams	: 19	Integrated Product Demo	: 20
Total	55		65

Document Templates:

The document templates are under module 1, make a copy of them to use in your sprints.

TeamStandup: This document should have 1 entry per class (or twice a week) where team members discuss what they are working on until the project is completed.

TeamPlanning: This document is filled out once per sprint at the beginning and will have the stories pulled into the sprint, their estimate and who is assigned to them. The team also agrees to a Sprint Goal (what do they plan to accomplish in the sprint).

TeamRetro: This document is filled out once per sprint after the review and will have 1) what went well 2) what didn't go well and 3) what improvements can be made.

TeamReview: This document is filled out before the review occurs and WORKING software is demoed to the product owner or stakeholders. The product owner should review which stories were completed and if any, where not completed.

Working with User Stories:

The following are example stories have been provided to understand how to write the user stories:

Feature: **Profile Management**

User Story Format Example:

Title

As a book browser or purchaser, I can create a user profile So that I do not have to enter my information each time I add books to my shopping cart or purchase books

Acceptance Criteria:

Functional

User can enter a user name which will stored on a database
User can enter a first name and last name which will be stored on a database
User can enter a password which will be stored on a database
User can enter an address which will be stored on a database
User will click on a save button to persist the info on a database
User will receive a success message.

Nonfunctional (if Needed)

When the user saves the profile information, a response should occur within 3 seconds of success profile creation or an descriptive error message.

Acceptance Test Cases (A list of the test cases which were performed to validate that the user story functionality was completed)

Test Case 01: Account Creation

1.Go to the create profile screen, fill out a username and password (must adhered to security standards) and click save.

2. A confirmation screen will let you know that the account was created.

3. Verification of database will show that account was created.

…(more as needed)

Estimate

8 (hours)

I will provide a planning document template with the following info

StoryId, Title, Status (New, Implementing, Done), AssignedTo , Estimate , Acceptance Criteria.

Final Demo

During the final weeks of class, the teams will have the project ready to demo from a given integrated sandbox. I will ask you to demo the core features and any additional features.

I will provide all the Sprint Review Schedules once the teams are assigned. Please refer to the following Google Calendar:

FIU Roque CEN 4010

https://calendar.google.com/calendar/embed?src=1pthsodlk0craqsc992pmp5g1s%40group.calendar.google.com&ctz=America/New_York

Getting Started Checklist:

Before Sprint 1, you need to have the following ready:

1. Meet your team members and exchange contact/method of communication in your group discussion. Meet face to face or online using collaboration tools. You should schedule to meet regularly twice a week.
2. Review Lecture 1 to understand the scrum fundamentals.
3. Review Feature Checklist. Agree on Feature owners.
4. Agree on technology and architecture. Fill out the features and architecture document.
5. Decide for Sprint 1 who would be:
    a. Product Owner
    b. Scrum master
6. Review Document Templates.
7. Review User Story example. You will receive feedback each sprint.
8. Make sure you can access your team forum.
9. Create GitHub project for your team.

To assure that the group has a clear understanding of the items in this checklist, there will be a project kickoff meeting scheduled for each group. Please Check the calendar for the date.

Requirements for System Demo

Everyone must have the following for the class:

1. Google Account
2. Webcam and Microphone
3. PC Desktop Recording Software
4. Ability to meet for 15 to 30 minutes for demos on weekends