Github: <https://github.com/amandaw800/BookClub>

Amanda:

**Use Case UC1: Allow users to sign up with email, Google, or social media accounts**

* 1. Primary Actor: User
  2. Stakeholders and Interest: User, Google, other social media, Book Club
  3. Preconditions: The user is registered to either email, Google, or social media
  4. Success guarantee: The user can log into their social media/email. The user goes to the sign-up page and signs up using their social media/email. The system validates the login and signs them up.
  5. Main success scenario: The user is on the Book Club page. The user clicks to sign up using email/social media. A separate window opens with a login page, where the user inputs their username and password for the social media/email. The user presses the sign-in button and the system validates the username and password. The user successfully signs up.
  6. Extensions:
     1. Invalid password or username:
     2. The system signals incorrect credentials.
        1. The user inputs the credentials correctly and logins.
        2. The user forgot the password and used the “Forgot Password” button. The user inputs their email and an email is sent to them. The user puts a new password in and resets it. The user logs in using the new credentials.
        3. The user forgot their username and used the “Forgot Username” button. The user inputs their email and an email is sent to them. The user is shown their username. The user logs in with their credentials.
  7. Special requirements:
     1. The login system responds within 20 seconds (Speed, Consistency)
     2. The usage of MFA to protect security (Security)
  8. Technology and variation list:
     1. Use of email must include @ symbol
     2. Information is to be inputted using a keyboard

**Use Case UC2: Password and username to validate registered accounts**

* 1. Primary Actor: User
  2. Stakeholders and Interest: User, Book Club
  3. Preconditions: The user already has a Book Club account
  4. Success guarantee: The password and username are correctly inputted. The user is logged into their Book Club account.
  5. Main success scenario: The user types in their username and password. The system validates the credentials and the user logs into their account.
  6. Extensions:
     1. Invalid password or username:
     2. The system signals incorrect credentials.
        1. The user inputs the credentials correctly and logins.
        2. The user forgot the password and used the “Forgot Password” button. The user inputs their email and an email is sent to them. The user puts a new password in and resets it. The user logs in using the new credentials.
     3. The user forgot their username and used the “Forgot Username” button. The user inputs their email and an email is sent to them. The user is shown their username. The user logs in with their credentials.
  7. Special requirements:
     1. The login system responds within 20 seconds (Speed, Consistency)
     2. The usage of MFA to protect security (Security)
  8. Technology and variation list:
     1. The usage of the @ signal when email is prompted
     2. The password must include at least 7 characters with at least one special character, number, and uppercase letter.
     3. The keyboard will be used to input information

**Use Case UC3: Create profile (profile pic/bio)**

* 1. Primary Actor: User
  2. Stakeholders and Interest: User, Book Club
  3. Preconditions: The user is registered on the Book Club. The user has a photo to upload.
  4. Success guarantee: The user has a photo and bio on their profile.
  5. Main success scenario:
     1. The user is on their profile and clicks the “Update Profile” button. The user clicks the “Add Bio” button and then can type. The user types in the textbox about them and then clicks “Save.” The bio is saved and updated on the profile. The user clicks “Add Photo,” and then chooses a photo on their computer. The photo is uploaded and a preview is shown. The user clicks “Save,” and the photo is shown on their profile.
  6. Extensions:
     1. The user has an invalid photo.
        1. The system communicates that the photo is invalid and shows the restrictions that are causing it to error (Ex: more than 20 MB).
           1. The user gets a valid photo.
           2. The user uploads the photo and successfully gets validated.
           3. The user’s profile is then updated.
     2. The bio length is too long.
        1. The system communicates that the bio can only have a maximum of 400 characters.
           1. The user decreases the number of characters being used.
           2. The user saves the new bio and their profile is updated.
  7. Special requirements:
     1. Photos take at most 5 seconds to upload (Speed, consistency)
  8. Technology and variation list:
     1. The photo must be less than 20 MB, 100x100 pixels maximum, and jpeg or png.
     2. Input is by keyboard
     3. The maximum bio length is 400 characters.

**Use Case UC4: Ability to create book club group and join**

* 1. Primary Actor: Users
  2. Stakeholders and Interest: User, Book Club
  3. Preconditions: Users are registered on the site and are verified by email/social media.
  4. Success guarantee: The user creates a book club group.
  5. Main success scenario: The user goes to the Book Club page and clicks “Create new Book Club.” The system gives basic information to fill out. The user fills out basic information including Name, Bio, Photo, contact and general genre information. The user clicks “Create” and the Book Club is successfully created.
  6. Extensions:
     1. The user incorrectly fills in the basic information.
        1. The photo was not within 1200 x 800 pixels or a png/jpg or there was none.
           1. The user finds a photo that fits the requirements and uploads it. The Book Club information is saved.
           2. The system puts in a default photo and saves the photo.
        2. The bio was too long and the system showed an error or that there was none.
           1. The user deletes some characters. Hits save and updates the bio.
           2. The bio is blank.
        3. There was no given to the club
           1. The system indicates that there is no name to the club, but there must be in order to create it.

The user inputs a name for the club and creates the club

* + - 1. Information not fully given
         1. The information remains blank, but a reminder from the system indicates that having information is beneficial for people finding the club.
  1. Special requirements:
     1. The creation of the book club responds within 20 seconds
  2. Technology and variation list:
     1. 1200 x 800 pixel maximum and png/jpg image uploads only for the photo
     2. Bio can be up to 1000 characters
     3. There must be a name for the book club

Sierra:

**Use Case UC5: Share pictures with the group**

Primary Actor:

Book club member

Stakeholders and interests:

Club members: Want to share pictures of their lives

Book club groups: Wants to create a more engaging community especially for virtual clubs

Preconditions:

User must be logged into their account

Group must have set permissions for pictures to be enabled

The user is on the group page

Success guarantee:

The user shares a picture within the group.

Main success scenario:

1. The user clicks the “Share Photo” button
2. The user is prompted to upload a picture from their gallery to post.
3. The photo is uploaded and a preview of the post is shown.
4. The user types a caption for their post.
5. The user clicks “Share” and the picture with the caption is posted to the group.

Extensions:

4a) The user doesn’t type a caption

The user clicks “Share” and the picture is posted to the group without a caption

5a) The picture fails to share to the group

The user is prompted “Try again” to reupload or “Cancel upload” to remove the post

1. The user selects “Try again”
   1. The picture is shared with the group
2. The user selects “Cancel upload”
   1. The upload process is canceled and the post is discarded

Special requirements: The posts successfully upload within 1 minute of the user clicking “Share”.

Technology and data variation list:

* 1. The photos are uploaded from the user’s gallery on their computer/phone
  2. The caption takes input from a keyboard

Frequency of Occurrence: Occasionally, as users may want to share events with their friends in the group.

**Use Case UC 6: Ability to block/unblock users from joining the group**

Primary Actor:

Book Club Admins

Stakeholders and interests:

Book Club Admins: want to moderate the group, and add/remove users at their own discretion.

Preconditions:

Book club admin is logged into their account and has access to full admin privileges.

Admin is on the group’s settings page.

Success guarantee:

The admin blocks a user from joining the group.

Main success scenario:

1) The admin is prompted to enter a username to block.

2) The admin enters the username.

3) The admin is prompted with a confirmation, and clicks “Yes”

4) The username is added to the block list.

Extensions:

3a) The admin is prompted with a confirmation, and clicks “No”

The username does not get added to the block list.

1a) The admin accesses the block list.

The admin selects a user from the list and selects “Unblock”

The admin is prompted with a confirmation

i) Clicks “Yes”

Username is removed from the block list

ii) Clicks “No”

Username remains on the block list

Special requirements: This action takes no more than 15 seconds.

Technology and data variation list:

* 1. A block list middleware (like Laravel Blocking) is used to check a user's connections.
  2. Takes keyboard input

Frequency of Occurrence: On an as needed basis, more frequently for larger groups, but less frequently for smaller groups.

**Use Case UC 7: Set group moderators**

Primary Actor: Book Club Owner

Stakeholders and interests:

Book Club Owners: want their groups rules enforced without

having to do it themselves, and active coverage within the group.

Book Club Members: want an active community with little to no problems.

Preconditions:

There is more than one member in the group.

The owner is on the Members page

Success guarantee: The owner sets a member to be a group moderator.

Main success scenario:

1. The owner clicks “Make member a moderator”
2. The owner is prompted with a confirmation message.
3. The owner clicks “Yes”
4. The owner can choose admin privileges for the member.

Extensions:

3a) The owner clicks “No”

The member is not made a moderator.

Special requirements: This action takes no more than 15 seconds.

Technology and data variation list: There is a custom configuration lists that the owner can choose from to select admin privileges for moderators (like an access permissions api).

Frequency of Occurrence: On an as needed basis, more frequently for larger groups, less frequently for smaller groups.

**Use Case UC 8: Borrow or buy books**

Primary Actor: Book Club Members:

Stakeholders and interests:

Book Club Members: want to be able to borrow or buy the book that the current group is reading or interested in reading.

Book Club (Dropshipping): wants users to be able to easily buy the book that their group is reading without having to leave the website

Online stores: want users redirected here to purchase books

Libraries: want users to access the book without having to purchase it

Preconditions:

The user is on a books page from the bookshelf.

Success guarantee: The user purchases the book.

Main success scenario:

1. The user is clicks “Find a Copy”
2. The user is then given the options “Add to Cart”, “Libraries”, “More Options”.
3. The user clicks “Add to Cart”
4. The book is added to the virtual shopping cart
5. The user goes to their cart
6. The user clicks “Checkout”
7. The user enters in their personal information (card and address)
8. The user receives confirmation that their order has been received with a confirmation number.

Extensions:

3a) The user clicks “Libraries”

i) The user is redirected to a separate website (like WorldCat) to find a library near them with the book in stock.

3b) The user clicks “More Options”

i) The user is given a list of options such as “Amazon”, “Barnes and Nobles”,....etc.

ii) The user clicks on “Amazon”

iii) The user is redirected to Amazon.com and is able to purchase the book from there.

Repeat ii and iii for all possible store options.

Special requirements:

Technology and data variation list:

There is a database of the available books that are in stock.

Personal information for each user is saved securely through a payment processing system (like PayPal).

Integration with ecommerce platforms and library systems

A shopping cart system

Frequency of Occurrence: Frequently, as users are buying books for their specific groups.

Benjamin:

**Use Case UC9: Search for books**

Primary Actor: User

Stakeholders and interests:

* Users: want to find books to read with book club
* Authors and Retailers: want their books promoted on the app

Preconditions:

* The user must be logged into their account

Success guarantee:

* The user is given a list of books that match their query (title, author, ISBN)

Main success scenario:

1. The user selects a search books button

2. The user enters their query (title, author, ISBN)

3. The user selects the magnifying glass beside the text box where they entered their query

4. The app displays a list of books matching the query

Extensions:

4a. User input is invalid

4b. No matches found for the query

Special requirements:

* UI is user friendly

Technology and data variation list:

* Use of external API such as Google Books API

Frequency of Occurrence:

* Will occur frequently

**Use Case UC10: Set a group name for the book club**

Primary Actor: Club admin

Stakeholders and interests:

* Users want representative names of their club
* Club admins want appropriate names for their clubs

Preconditions:

* The user must be creating a book club or already have created the club
* To set the name, the user has to be an admin

Success guarantee:

* The book club is correctly named based on the user’s input

Main success scenario:

1. User clicks “Create a new club” or “Edit book club settings”
2. The user enters the new club name into the text box “Enter club name” or “Edit club name”
3. The user selects the “Create” or “Update” button respectively
4. The system validates the name and write this to the database
5. The user is given confirmation that their club is named what they entered in step 2

Extensions:

4a. The group name already exists

4b. The name contains invalid characters or is inappropriate

Special requirements:

* The UI should be user friendly

Technology and data variation list: None

Frequency of Occurrence:

* This use case is not expected to occur frequently

**Use Case UC11: Set the current book**

Primary Actor: Club admin

Stakeholders and interests:

* Users: Want a book selected via a poll

Preconditions:

* Book club is created
* User creating the poll must be an admin
* Admin of the club is logged in

Success guarantee:

* The app is updated to reflect the club’s new book

Main success scenario:

1. The user goes to the “Current book” section of the club
2. The user creates a poll to decide what book to choose
3. Other members of the club vote from a selection of books created by previous suggestions from the club’s members
4. The poll concludes and the book with the highest number of votes is set as the club’s current book

Extensions:

4a. There is a tie in the poll

Special requirements: The UI is user friendly

Technology and data variation list: Google books API to validate books for the poll

Frequency of Occurrence: Everytime a book is completed a different book is set for the group

**Use Case UC12: Set group type (by genre)**

Primary Actor: Club admin

Stakeholders and interests:

* Users: Want to have their group type represent the group’s most read genre

Preconditions:

* The user must be creating a book club or already have created the club
* The user must be an admin

Success guarantee:

* The app is updated to reflect the group's type (genre) inputted by a user

Main success scenario:

1. User clicks “Create a new club” or “Edit book club settings”
2. User selects “Set club type”
3. The user selects the new group type from a drop down menu
4. The user selects the “Create” or “Update” button respectively
5. The system writes this to the database
6. The user is given confirmation that their club type has been updated or the group has been created

Extensions:

4a. A group type is not selected

Special requirements: The UI is user friendly

Technology and data variation list: None

Frequency of Occurrence:

* This may occur frequently, as it may occur every time a group changes their book

Ty:

**Use Case UC13: Book recommendations**

Primary Actor:

* Book club member

Stakeholders and interests:

* Club member: wants personalized, accurate book recommendations based on interests or book club’s preferences
* Book club: wants to engage members with suitable recommendations

Preconditions:

* User must be logged into the app
* The app must have enough relevant information about user/club to give accurate recommendations

Success Guarantee:

* User receives relevant book recommendations

Main success scenario:

1. Book club member logs into the app
2. Member navigates to “Book Recommendation” section
3. The app analyzes member’s reading history, preferences, and/or club’s reading list
4. The app generates a list of recommended books
5. The member views the recommendations and selects a book to view more details or add it to their library

Extensions:

* 3a. If there is insufficient data for recommendations, the app prompts user to enter preferences manually
* 5a. The book is out of stock or unavailable, and the app suggests similar alternatives

Special Requirements: None

Technology and Data Variation List:

* App uses an external book database (ex. Google Books API) for recommendations

Frequency of Occurrence:

* Frequent;y, whenever members want new recommendations

**Use Case UC14: Add books to reading club list**

Primary Actor:

* Book club admin

Stakeholders and Interests:

* Book club admin: wants to add books to club’s reading list efficiently and ensure list is accessible to all members
* Book club members: want to view and track books in the club’s current reading list

Preconditions:

* Book club admin is logged into app and has full admin privileges

Main success scenario:

1. Admin logs into the app and navigates to reading club list section
2. Admin selects “Add book” and searches for the book in catalog or enters details manually
3. Admin confirms the selection and saves changes
4. The book appears in the reading list for all members to see

Extensions:

* 2a. If the book is not found, the admin is prompted to manually enter the book details (ex. title, author, ISBN)
* 3a. If admin selects a duplicate book, app notifies them and suggests alternatives

Special requirements:

* Reading list should be fully accessible to all members’ devices no matter the device or browser

Technology and Data Variation List:

* App may allow adding books via ISBN or through integration with online book database

Frequency of Occurrence:

* Regularly, when adding new books for club reading

**Use case UC15: Create polls**

Primary actor:

* Book club admin

Stakeholders and Interests:

* Book club admin: wants to engage members with polls, typically to select next book or meeting time
* Book club member: want to participate in decisions by voting in polls

Preconditions:

* The admin is logged into the app and has admin privileges

Success Guarantee:

* The poll is created, and members can vote

Main Success Scenario:

1. Admin logs into the app and navigates to the ‘Polls’ section
2. Admin selects ‘Create new poll’ and enters the question and options
3. Admin sets poll duration and voting restrictions if applicable
4. Admin confirms and publishes the poll
5. Members receive notification about new poll

Extensions:

* 3a. If admin enters invalid data (overlapping dates, etc.), the app prompts corrections before publishing
* 4a. If there are not enough responses or tie, admin may extend poll duration

Special Requirements: Poll results should update in real-time

Technology and Data Variation List:

* Polls may be conducted via mobile and web versions of app

Frequency of Occurrence:

* Occasionally, when club needs member input

**Use Case UC16: Remove/add books from club reading list**

Primary actor:

* Book club admin

Stakeholders and Interests:

* Book club admin: wants to manage the club’s reading list by adding/removing books as necessary
* Book club members: want an updated, accurate list of books currently being read or discussed by the club

Preconditions:

* Admin is logged into the app and has admin privileges

Success Guarantee:

* Books are added or removed successfully, and list is updated

Main success scenario:

1. Admin logs into the app and navigates to the club reading list
2. Admin selects a book to remove or adds a new book to the list
3. Admin confirms changes and saves them
4. The reading list updates for all members

Extensions:

* 2a. If the book to be removed is currently being discussed, the app warns admin before removal
* 2b. If a duplicate is attempted to be added, the app suggests alternative actions (ex. edit current entry)

Special Requirements: none

Technology and data variation list:

* Admin can add/remove books via mobile or web version of app

Frequency of Occurrence:

* Frequently, as reading list evolves over time

Sky:

**Use Case UC17: Set challenges/read by dates (progress)**

* Primary Actor: Users
* Stakeholders and interests:
  + Users: want to read a certain amount of pages by a set date to reach their reading goals (this could be to keep up with their book club reading deadlines or if they want to participate in weekly or monthly reading challenges)
  + Users: want a tangible reminder that motives and encourages them to hit their goals
* Preconditions:
  + The user must have an account and be logged in
* Success guarantee:
  + A reader is able to see a new challenge/reading goal added to their active challenges list
* Main success scenario:
  + 1. A user logs into their account
  + 2. The user navigates to the challenges tab and presses the “create a challenge” button
  + 3. The user sets their challenge or reading goal by indicating how many pages they want to read, and the frequency and length of this challenge (how many pages in a day, week, or month until the date they enter)
  + 4. The user submits this challenge, and it will appear at the top of their reading goals page
  + 5. The user will see daily reminders reminding them about their reading goal, or congratulating them if they hit their reading goal
  + 6. When a user hits their goal:
    - A. If the user is reading their book online on the platform, then the online book reader will keep track of pages read and alert readers when their goal is met
    - B. If the user is reading their book offline the platform, then the user can keep track of their own reading progress and can manually set the upcoming goal as completed
* Extensions:
  + If a user sets daily or weekly goals and then tries to mark a goal for a future date as complete, an alert will block the user from doing this and suggest marking the day as incomplete instead.
  + If a user does not have notifications turned on, they will not receive pop-up notification reminders
  + Challenges can be set for book clubs by the club creator or members who were granted access by the club creator
* Special requirements:
  + None
* Technology and data variation list
  + Goals and reminders will work on the app and website platforms
* Frequency of occurrence:
  + Occasionally (weekly or monthly)

**Use Case UC18: Read books online (with ability to bookmark, make notes, highlight, set chapter length goals, personalize font, size, colors, etc)**

* Primary Actor:
  + User
* Stakeholders and interests:
  + User: Wants to be able to read books online within the same platform as their other book interests
  + User: Wants to be able to keep track of pages and reading progress with bookmarks and chapter length goals, and write down observations within the online book
  + User: Wants to be able to customize the online book view to their reading preferences
* Preconditions:
  + The user is logged into the app and has purchased the online book from our platform
* Success guarantee:
  + The book is able to be loaded on the app, and is scrollable for the user to read through
* Main success scenario:
  + 1. The user logs into the app
  + 2. The user navigates to their purchased/downloaded book list and picks the book they want to read
  + 3. The book is loaded onto the online reader to the page the reader last closed the online book reader at
* Extensions:
  + Readers can read offline if they have downloaded the book while online, otherwise users will need a connection to read a non-downloaded book
* Special requirements:
  + None
* Technology and data variation list
  + Has functionality on web and app versions
* Frequency of occurrence:
  + Frequently (daily use)

**Use Case UC19: Find local book clubs nearby (register your book club online/on the app)**

* Primary Actor: User
* Stakeholders and interests:
  + User: wants to be able to find nearby book clubs
  + User: wants to find more people to join their already existing club, or find more users to start a club with
* Preconditions:
  + The user must have an account and be logged in
  + The user must have already created and named a book club (if they want to register a book club online)
* Success guarantee:
  + For finding book clubs nearby: users are able to see a listing of nearby book clubs
  + For registering book club: users book club appears as a new entry to the nearby book club listing
* Main success scenario:
  + To find a nearby book club:
    - 1. A user logs into their account
    - 2. A user navigates to the find a book club tab
    - 3. A user enters their location or shares it through their settings
    - 4. The user can look through the listings of nearby clubs and select the one they choose (can be one or more clubs)
  + To register their book club online:
    - 1. A user logs into their account
    - 2. A user navigates to the find a book club tab
    - 3. A user chooses the Register Book Club option and enters in their book club information
    - 4. The user submits the information, which will add their club to the list
* Extensions:
  + If a user tries to register their book club without creating one first, they will be prompted to create and name a new book club
* Special requirements:
  + The radius of book clubs showed to the user will be around 25 miles
* Technology and data variation list
  + Users will be able to find book clubs and register events from both the app and website
* Frequency of occurrence:
  + Occasionally

**Use Case UC20: Submit club thoughts, opinions, reviews about the books as public notes/posts**

* Primary Actor:
  + User
* Stakeholders and interests:
  + User: wants to communicate their thoughts, ideas, reviews, and feelings about the book to a public audience
* Preconditions:
  + The user must have an account and be logged in
* Success guarantee:
  + A post is published to all users on the post tab
* Main success scenario:
  + 1. A user logs into their account
  + 2. The user navigates to the post tab and presses the create a new post button
  + 3. The user writes their thoughts into the text space of the draft post
  + 4. The user presses the submit button, which then releases their post to the public post page
* Extensions:
  + If there a spoiler mentioned in the post, the user will have the ability to be flagged as spoiler
    - If another user finds a post to be a spoiler without the tag being present, the other user can suggest a spoiler flair for the post
  + Other user may like and comment on posts
  + The user can save and format posts as drafts
  + The user can delete a post at any time
* Special requirements:
  + All posts with harsh inappropriate language will be blocked from being posted
* Technology and data variation list
  + Posts should be able to be submitted from every device, and from app/website versions
* Frequency of occurrence:
  + Frequently

Jaden:

**Use Case UC21: Have private chat channels for book clubs to discuss books**

Primary Actor: User

Stakeholders and interests:

* User: Wants to be able to talk about their favorite books
* Company: Wants users to use the website and facilitate interactions between different users.

Preconditions:

* An admin has created a private chat room for other users to join from a shared book club.

Success guarantee:

* There is now a chatroom for users to speak freely about a book.

Main success scenario:

* An admin of a book club has navigated to “Create Chat” in the options of the book club settings. The admin creates the chat room and invites the other users from the book club to join the chat.

Extensions:

* Any kind of “not safe for work” content will be unallowed, this includes hate speech of any kind and users may report anyone if this has happened.
* Slurs will be automatically detected and be deleted upon being sent
* Admins have the power to make the chat “kid friendly” meaning curse words will not be allowed.
* Admins will also have the power to mute certain members.

Special Requirements:

* none.

Technology and data variation list:

* Uploads cannot be any higher than 59 mb
* Links to websites like twitter will have the links be embedded

**Use Case UC22: Notifications for due dates**

Primary Actor: Group Admin

Stakeholders and interests:

* Group Admins: Admins can be sure that people are reading the book in a timely manner and for discussion to follow.

Preconditions:

* A book club as to already be made
* A book club has agreed on a book
* User has notifications turned on

Success guarantee:

* The user is notified when a due date has been published and the details of what's due

Main success scenario:

* A user will be going about their day and then get pinged by the app. The user will check their phone and see the due date and the details released by a group admin. The user will be more motivated to read and get the reading done by the due date.

Extensions:

* Notifications can be scheduled in advance by admin so they are sent out a week before the due date, then a few days before, then the day before, and then the day of.

Special Requirements:

* Don’t ping users who have the notifications of the app turned off.

Technology and data variation list:

* Work with laptops and different brands of cell phones.

**Use Case UC23: Export reading data to external file**

Primary actor: User

Stakeholders and interests:

* User: Will be able to see their data and see their growth as a reader and the way their tendencies change over time.

Preconditions:

* User has indicated which books they have already read, when they started and when they finished, and book clubs they’ve participated in

Success guarantee:

* A user will have a pdf file listing different stats about themselves like: books read, book clubs participated in, time spent reading, etc

Main success scenario:

* At the end of a year a user may be curious on how their reading has looked for the past year so they request a PDF document of their reading stats. The user will be able to see different books read, new genres that they tried out and what kinds of books they gravitate towards.

Extensions:

* Users can request for different formats of the information.
* Users can opt out of this so their data is never stored
* There can be an automatic annual display of relevant stats users may want to see.

Special Requirements:

* Data is displayed in a clean and readable format
* Data is never excessively long

Technology and data variation list:

* Users can see different formats of data and choose which one they want to keep and download.
* App will store document so users can see data points from different times

**Use Case UC24: Calendar features to see book club dates, deadlines, and community events**

Primary actor: User

Stakeholders and interests:

* User: Sees what's happening and when so they have ample time to prepare
* Group Admins: Has comfort of mind that users will know what is happening and when and requirements are met.

Preconditions:

* Book club has to be made and scheduled due dates and events must already be listed

Success guarantee:

* Users will see a calendar and relevant due dates and events.

Main success scenario:

* A user will log onto the site and navigate to the calendar tab. A user will see a calendar and different events or due dates for different book clubs in a nicely laid out format. The user will be able to prioritize what they need to do and when.

Extensions:

* Users can add their own deadlines or events to the calendar.
* Calendar can be sync to other applications like google calendar.

Special Requirements:

* none.

Technology and data variation list:

* App is able to connect to other calendar applications such as google calendar.

**Supplementary specifications**

Benjamin-**Security**- Application will support multi-factor authentication to verify user identities. All sensitive data will be encrypted to protect against unauthorized access.

Ty - **Scalability** - Application will use AWS technologies (AWS RDS, specifically) to ensure app is automatically scaled for any number of users and user information

Amanda - **Performance** - All server-side validations (books, features, logging in) should be taken within 5 seconds to ensure consistency in user experience.

Sierra - **Usability** - The design of our application will be accessible across multiple devices (scalable fonts and elements, and testing across multiple OS) and user friendly to all users (adheres to WCAG standards).

Jaden - **Availability** - With the help of linguists the application will be available in many languages. We will also add features for accessibility to people with disabilities, we will do this by having alternative text for our images on the platform.

Sky - **Reliability** - To ensure maintenance of the system, regular backups will be scheduled on an incremental basis to prevent any data loss. The backups will take place daily during low activity periods to keep track of daily changes in data, and a more robust backup will be scheduled monthly to preserve the overarching system and data structure.

Use Case Diagrams:





