EECS 3311 Section M Project - Report

Ammad Qureshi	- 217996927	- ammadq87@my.yorku.ca
Kyle Chandrasena	- 218104265	- kylec107@my.yorku.ca
Nargis Ghiasi	- 218504886	- nargisg@my.yorku.ca
Nicholas Lachhman	- 218091231	- nickla@my.yorku.ca
Zohair Ahmed	- 215867633	- zohair99@my.yorku.ca
Edward Nwogwugwu	- 215163520	- eddie13@my.yorku.ca

GitHub Repo Link - https://github.com/Zohair-Ahmed/EECS3311M-Project

Team Map

Our Clients: Team 2

Name	Email
Haocheng Sun	2461987583@qq.com
Tao Chen	18108082141@163.com
Zhige Xia	379956192@qq.com
Wenxu Ye	yewenxu@my.yorku.ca

Will QA Us: Team 3

Name	Email
Mehrsa Khoshpasand Foumani	mkhoshpa@my.yorku.ca
Richard MacLean	richmac@my.yorku.ca
Preet Patel	preet152@my.yorku.ca
Apurva Singh	apurvasi@my.yorku.ca

To QA For: Team 17

Name	Email
Gregory Retinskiy	drofel@my.yorku.ca
Ayomide Salami	ensigna@my.yorku.ca
Sami Sami Ullah	samiull5566@gmail.com
Muskaan Sharma	muskaan3@my.yorku.ca

Developer Team: Team 18

Name	Email
Daniel Sheinis	sheinisd@my.yorku.ca
Viduren Sivakumar	viduren.sivakumar@gmail.com
Carlos Sosa Velasco	csosa@my.yorku.ca
Shon Vivier	shonviv@my.yorku.ca

Iteration 0

Same Page Vision Statement

Same Page (SP) will be used to seek and maintain a community-based library of books that are personalized for each member. Like a digital library, readers can search for, be recommended and learn more about their favorite books as well as view the latest releases. Extra features that allow the reader's data to be stored will be available for those readers registered as members with Same Page, including joining the same book clubs as other interested readers and tracking reading progress.

The system is primarily designed for those who have a fondness and curiosity of reading. SP will permit readers to view book information, including author, publication date, reviews, as well as add to a favorites list. SP also adds the ability to network with fellow book readers through the creation of a profile (including personal information, a feature to track statistics and set goals), joining book clubs and following each other's progress. As a result of the social and organizational aspects of this application, readers will be inclined to sustain positive reading habits.

One of the main functions of the systems is to provide information regarding books, users, and the relationship between both entities. The information will be maintained through the implemented API.

In terms of security, users can confidently expect that there will be no unnecessary personal data stored or leaked. The data the readers provide as login and preference on books will be secured to protect readers. Security in place for the book database, reviews, and related information is crucial for establishing integrity for SP and so that users can be assured the results and details SP provides are credible and unchanged. There will be no need for admins since the ability to upload/delete/update books is not an intended feature. However, since there are aspects of networking between readers, there would be a substantial focus and preventative measures paid to personal information not being leaked (such as passwords).

SP uniquely stands out as it helps motivate readers to connect in a digital book club space and be able to visually track and achieve reading goals. This application can further help connect readers with efficiently choosing books to read beforehand. Ultimately, with a created profile, readers can be inspired by other readers' progress and engage in meaningful discussions through their book club.

For the foreseeable future, the system is envisioned as a desktop application, however, it will be extensible so that future releases can have a web and mobile interface.

The system will be considered successful if readers prefer to use this resource to complement their current reading experiences, for example, using Same Page to search for books and book events when unable to do so at the library. Another criteria for success is if at least 50% of users are somehow involved with the social media aspects of the project (joining book clubs and making friends).

Big User Stories

Iteration 1

Goal - Create MVP of project, which includes landing page, search functionality and registration/login system

Landing Page

As a guest or member, I want to be able to open Same Page and view a number of books on the homepage.

Priority: High

Cost: 6 days

Search for Book

As a guest or member, I want to be able to search for a specific book.

Priority: High

Cost: 6 days

Register/Login

As a guest or member, I want to have the ability to register a new account with SamePage or login to an existing account.

Priority: Medium

Cost: 5 days

Iteration 2

Goal - The goal of this iteration is to create more “interactivity” between the books themselves and the guest/members, for example, allow both guests and members to know more about the book, be able to favorite books and leave reviews

User Reviews

As a member, I want to upload/leave a review of the book I have read.

Priority: High **Cost**: 5 days

Book Information

As a guest or member, when I click on the book, I should be able to view the main information of the book.

Priority: High **Cost**: 5 days

Favorites

As a member, I want to favorite books that I like or want to read later in the future.

Priority: High **Cost**: 5 days

Iteration 3

Goal - To create more interaction between members, by adding more social level aspects to the app. This includes looking for other users, joining book clubs and add more enhancements to a user profile (such as goals)

Find Friends

As a member, I want to be able to search for other Same Page users by their username and add them as a friend.

Priority: High **Cost**: 8 days

Book Clubs moved to Release 2 - Note, priorities changed and this story relies on the full implementation of the Friends and Social aspect of the app to be completed before Book Clubs.

~~Book Club~~

~~As a member, I want to be able to create or join book clubs.~~

~~**Priority**: Low **Cost**: 8 days~~

User Goals

As a member, I want to be able to create personalized goals that I can track my progress on.

Priority: High **Cost**: 5 days

Client Feature – Request to Add New Book

If a book is not listed, the user can submit a request/suggest the app to add the new book to a queue.

Priority: High **Big User Story**: Add New Book **Cost**: 3 days

Detailed User Stories

Iteration 1

Display Login/Register Buttons

Display the login and register buttons and their corresponding pop-up input boxes on the landing page.

Priority: High **Big User Story**: Landing **Cost**: 3 days

Display Latest Releases

Display the most recent book releases of any genre on the landing page.

Question: What external source can provide this information?

Priority: Medium **Big User Story**: Landing **Cost**: 3 days

Search Bar

Display a search bar where text can be entered or filters can be applied to search for a book.

Priority: High **Big User Story**: Search **Cost**: 4 days

Display Search Results

Display the results after searching for a book in the search bar.

Priority: High **Big User Story**: Search **Cost**: 2 days

Register New User

Add functionality to register an account with an email, username, and password.

Priority: Medium **Big User Story**: Login/Register **Cost**: 2 days

Login Existing User

Add functionality to securely log into Same Page with an existing email and password.

Priority: Medium **Big User Story**: Login/Register **Cost**: 3 days

Login Page

Create a separate login page with email and password input fields.

Priority: High **Big User Story**: Landing **Cost**: 1 days

Register Page

Create a separate register page with name, email, password, and confirm password input fields.

Priority: High **Big User Story**: Landing **Cost**: 1 days

Display Navbar

Create navbar component with login, register, clubs, and a profile icon.

Priority: High **Big User Story**: Landing **Cost**: 1 days

Iteration 2:

View Reviews

Display the book reviews upon clicking on a specific book.

Priority: High **Big User Story**: User Review **Cost**: 3 days

Add Reviews

Users logged in, and add reviews to the selected book upon clicking on a specific book.

Priority: High **Big User Story**: User Review **Cost**: 3 days

Display Favorite Books

Display the favourited user books in user information.

Priority: High **Big User Story**: Favorites **Cost**: 3 days

Add Favorite Books

Add a book to the user's favourites upon the user selecting a specific book from the search.

Priority: High **Big User Story**: Favorites **Cost**: 3 days

Remove Favorite Books

Remove a book from user favourites upon the user selecting a specific book from their user information list.

Priority: High **Big User Story**: Favorites **Cost**: 3 days

Display Book Information

Display book information such as title, author, publication date, summary, reviews, and ISBN number.

Priority: High **Big User Story**: Book Information **Cost**: 3 days

Client Feature – Request to Add New Book

If a book is not listed, the user can submit a request/suggest the app to add the new book to a queue.

Priority: High **Big User Story**: Add New Book **Cost**: 3 days

Iteration 3 User Stories:

Add User Goals

As a user, I want to be able to create a customizable reading book list and set goals of how many books to read on my profile page.

Priority: High **Big User Story**: User Goals **Cost**: 5 days

Update Goals

As a user, I want to be able to update any goals I created with my progress.

Priority: High **Big User Story**: User Goals **Cost**: 5 days

Search for Friends

As a user, I want to be able to find a friend by looking up their username and click their profile from the results.

Priority: High **Big User Story**: Find Friends **Cost**: 3 days

Add Friend

As a user, after I find my friend I want to be able to add them as my friend.

Priority: High **Big User Story**: Find Friends **Cost**: 3 days

View Friends Profile & Goals

As a user, I want to be able to find my added friends on my profile page and view their profile page and user goals.

Priority: High **Big User Story**: Find Friends **Cost**: 3 days

Remove Friends

As a user, I want to be able to remove any friends I have made

Priority: High **Big User Story**: Find Friends **Cost**: 3 days

Iteration 3 Developer Stories:

Refactoring: Increase loadtime speed for landing page

Priority: High **Big User Story**: Refactor **Cost**: 3 days

Refactoring: Convert App to Single Page App

Priority: High **Big User Story**: Refactor **Cost**: 3 days

Refactoring: Prevent Spam Reviews

Priority: High **Big User Story**: Refactor **Cost**: 3 days

Testing: Add a User Goal

Priority: High **Big User Story**: User Goals **Cost**: 2 days

Testing: Update User’s Goal

Priority: High **Big User Story**: User Goals **Cost**: 2 days

Testing: Add Friend

Priority: High **Big User Story**: Find Friends **Cost**: 2 days

Testing: View Friend Profile

Priority: High **Big User Story**: Find Friends **Cost**: 2 days

Testing: Removing Friend

Priority: High **Big User Story**: Find Friends **Cost**: 2 days

Testing: Searching for Friend(s)

Priority: High **Big User Story**: Find Friends **Cost**: 2 days

Peer Evaluation

<table>
<tr><td colspan="3"><b>Group Number: 1</b></td></tr>
<tr><td>Member’s Name</td><td>Mark</td><td>Explanation (only if mark &lt; 100)</td></tr>
<tr><td>Ammad</td><td>100</td><td></td></tr>
<tr><td>Kyle
KC</td><td>100</td><td></td></tr>
<tr><td>Nargis
NG</td><td>100</td><td></td></tr>
<tr><td>Nicholas
n.L.</td><td>100</td><td></td></tr>
<tr><td>Zohair</td><td>100</td><td></td></tr>
<tr><td>Edward
EN</td><td>100</td><td></td></tr>
</table>

Appendix

GitHub Etiquette

Branches

Main Branches

- `main`: **Production code**. All development code is merged into `main` in sometime
- `develop`: **Pre-production code**. When the features are finished then they are merged into `develop`

Supporting Branches

- `feature/`: **used to develop new features** for the upcoming releases. May branch off from `develop` and must merge into `develop`
- `hotfix/`: used when `main` **has an undesired status**. May branch off from `main` and must merge into `main` and `develop`
- No need for `release` branch

Naming Convention

`<supporting-branch>/<issue-number>-<title-of-issue>`

Examples:

- `feature/29-add-unit-test`
- `hotfix/37-fix-variable-name`

Rules

- **All pushes should be to the `develop` branch** - we will keep `main` branch for “production” (what we want the TAs to use) code only (this will be important when it comes to deadlines and unfinished work), unless merging a `hotfix` branch
- `rebase` **often!!** The more you `rebase`, the less likely there will be merge conflicts. In the devastating circumstance you do encounter a merge conflict, it should be a small one (▶ Learn Git Rebase in 6 minutes // explained with live animations! , why don’t all git tutorials have animations?)

- Add **meaningful `commit` messages**, good practice for when you enter the workforce and maybe we'll get maintenance marks?? I learned how to write good commit messages via https://cbea.ms/git-commit/, I highly recommend it!

- (To be discussed) To merge to `develop` branch, 2 approvals are needed and to merge to `main` branch, 3 approvals are needed

Workflow

Git workflow is based off Gitflow Workflow

Ideal Scenario

1. Open **CLI** and `cd` into *EECS3311M-Project* if not already there

2. `git checkout develop`
    - Go to the `develop` branch on your laptop (aka local machine)
    - Sometimes `git` won't let you change branches because there are modifications in your local branch and it'll lose those if you change branches, this means you'll have to `git stash` (remove your changes but save them) before checking out

3. `git pull`
    - Update local `develop` branch to match what's on GitHub (aka remote)

4. `git checkout <local-branch>`
    - Go to the branch you are working on

5. `git rebase develop`
    - Update your local branch to include the changes of the remote `develop` branch
    - If you had to `git stash`, after rebasing you'll have to `git stash pop` (bring your changes back to the code)

6. Fix any merge conflicts
    - Feel free to discuss with a teammate if you don't know which option to choose

7. Start/continue the changes you are making (repeat step 2-7 every so often)

8. Once you have completed the task, rebased with `develop` and have all the test cases created and passed, `git add <your-files>` (you can do `git add .` to add ALL changed files)

9. `git status`
    - This lets you know which files are going to be added to GitHub (may seem unnecessary but doesn't hurt to double check)

10. `git commit`

    - This will let you write a message so your future self will know what changes your past self made and when
    - Press **i** to begin typing your message, then press **ESC** and type **:wq** once done
    - If your commit message does not need a body paragraph feel free to just `git commit -m “<your-message>”`, then **Enter**

11. `git push`
    - Sent your 10/10 code to the remote version of your branch 🙂

12. Request an MR to merge your changes to the `develop` branch. It’ll have to be approved by 2 of us before merging to `develop` (maybe we’ll remove this rule, idk)

13. Celebrate 🎈

Amending

- Let’s say you already pushed into the your remote branch but realized you made a small mistake (defined as nothing big enough to change your commit message), you can amend it

1. Make the necessary changes

2. `git add <your-files>`
    - Let git know that you want to update this/these files

3. `git status`
    - Just to make sure you are adding the right files

4. `git commit --amend`
    - Modify/update the most recent commit

5. `git push -f`
    - Force a git to make the changes to the remote files

GoodReads Type Project:

Itr #1: Setting up page and search functionality

- Login/Sign-up
    - Create account (email?)
- Search
    - Filters by genre, author, etc
- Inventory (Goodreads api?)
- Landing Page
    - Search feature
    - Results display
- Favourites Page
    - On results display, allow for book to be favourited?

//// V2 ITERATION 1

- **Landing Page**
    - Search bar (ability to search for books)
    - Login Link/register ()
    - Display latest release (links to clubs for future, favorites, trending, genres)
    - Priority: High
    - Cost:

- **Search**
    - Search book by name
    - Inventory (Goodreads API?)
        - Information on book (summary, author), stock
            - **Q**: should this be moved to itr2 since that is more about book information?
    - Priority: High
    - Cost:
- **Login/Sign-up**
    - Create account (email?)
        - Account:
            - Email
            - Username
            - Password
        - (technical) Need to setup db
    - Priority: High
    - Cost:

Itr #2: Book Information

- User reviews
    - Upload reviews
    - Rate reviews (dislike/like) ?
- Book information
    - Leetcode type of thing?

  - Description
  - Discussion
    - **Unclear**
  - Review is located here?
- Suggested Page
  - Suggest by authors, genre, etc
  - Filter option?
- Notifications
  - **Unclear**

// V2 ITERATION 2

- User reviews
  - Upload reviews
  - Rate reviews (dislike/like) ?
- Book information
  - Description of book, genre, author, generic book info
  - Leetcode type of thing?
  - Discussion
    - **Unclear**
  - Review is located here?
- Search Filter
  - Search books with a filter (genre, author)
- Suggested Page
  - Suggest by authors, genre, etc
  - Filter option?
- Notifications
  - **Unclear**
- Favourites Page
  - Why are we favouriting?
    - Is it b/c I like that book or want to read it later
  - On results display, allow for books to be favorited?

Itr #3: Social Aspect

- Find other users
  - Find through search?
  - Add users as friends
  - Followers/Following
  - Follow authors??
  - Spotify wrap type of thing?
    - **Unclear/Need more info**
- User Goals
  - Based on honour system
  - Track how many books you read
    - **Need more info**

//

- Buy book

        - **Need more info**
- Book club
    - Search for book clubs
    - ad
- User reviews
    - Upload reviews
    - Rate reviews (dislike/like) ?
- Book information
    - Description of book, genre, author, generic book info
    - Leetcode type of thing?
    - Discussion
        - **Unclear**
    - Review is located here?
- Search Filter
    - Search books with a filter (genre, author)
- Suggested Page
    - Suggest by authors, genre, etc
    - Filter option?
- Notifications
    - **Unclear**
- Favourites Page
    - Why are we favouriting?
        - Is it b/c I like that book or want to read it later
    - On results display, allow for books to be favorited?

- Expected User Experience
    - User opens application, sees landing page, search bar and option to register/login
    - Upon clicking register, the user will see input fields for email, username, and password
    - Upon clicking login, the user will be prompted to enter email and password
    - After logging into SP, the user can now write reviews on books, and favourite books they like
    - After logging into SP, the user can now view the book clubs section, where they can create and join book clubs
    - After logging into SP, the user can now create and track book reading progress via goals
    - Ability to write reviews
- Possible Workflow stuff
    - Same Page will be a desktop application that can be an improvement over current book related applications which do not offer a combination of social networking and book discovery. The application will provide a seamless workflow where users can discover books and connect with like minded readers all in one place. When a user opens a page for a certain book they should expect to find

reviews for the book, recommended book clubs, Same Page users with similar book interests

Meeting Logs

Itr1 Meeting 1

February 3rd, 2023 @ 11:00 am
Attendance: Zohair, Kyle, Ammad, Nick, Nargis

Meeting Topic: Itr1 requirements analysis

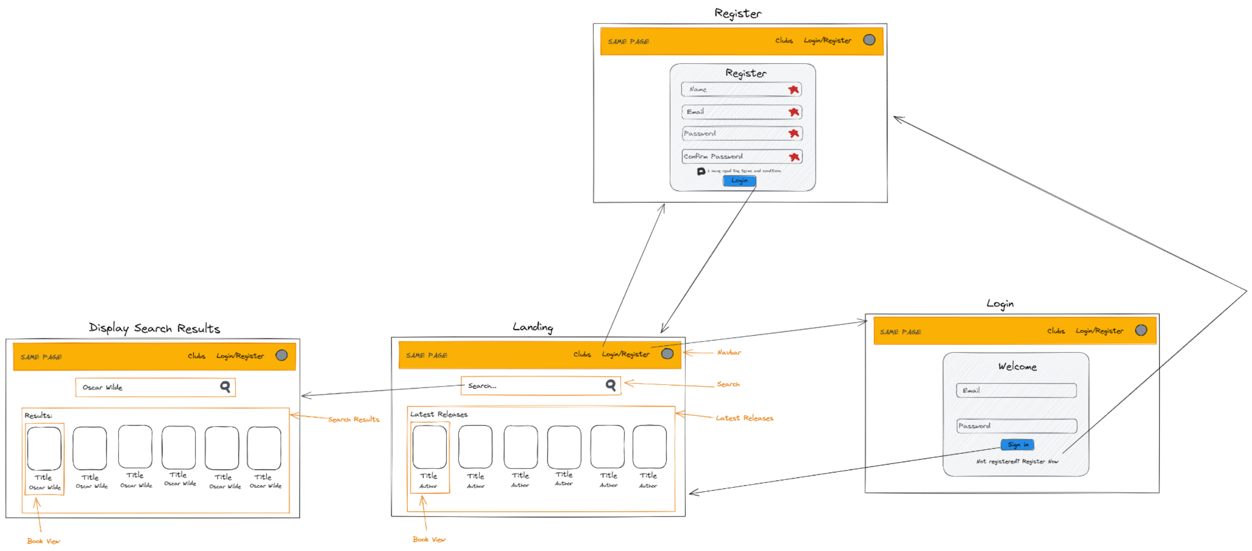
- Discover deliverable priorities
- Understand who we need to reach out to as a team (Client team and developers)
- Client has a Gym Fitness Class Application, Zohair has been in contact with them through email/whatsapp
- Zohair to follow up with Developer team

Initial Design Architecture discussion:

- Student Class, student class to DB
- Course Class, course to DB
- Admin Class, admin to DB
- Instructor Class

Tasks:

- Find out what to use for GUI and Database access
  - Look into: JavaFX, JavaSpring, Hibernate, JavaAWT


Register
SAME PAGE
Clubs
Login/Register
Register
Name
Email
Password
Confirm Password
I have read the terms and conditions
Login
Display Search Results
SAME PAGE
Clubs
Login/Register
Oscar Wilde
Results:
Title
Oscar Wilde
Title
Oscar Wilde
Title
Oscar Wilde
Title
Oscar Wilde
Title
Oscar Wilde
Title
Oscar Wilde
Search Results
Book View
Landing
SAME PAGE
Clubs
Login/Register
Navbar
Search...
Search
Latest Releases
Latest Releases
Title
Author
Title
Author
Title
Author
Title
Author
Title
Author
Title
Author
Book View
Login
SAME PAGE
Clubs
Login/Register
Welcome
Email
Password
Sign in
Not registered? Register Now