# EECS 2311 SecZ Winter 2024 Team 3 Project Planning - ITR3

Hashir Jamil

Mher Eric Gyulumyan

Ali Sina

Ahan Bhargava

Oscar Ye

# Change Log & Justification

*There have been no changes to the plan since the conclusion of ITR2. - Mon, Mar 11, 2024*

Recap of Big User Stories For Full Project:

| **User Login & Registration** |
|---|
| LongBox users must register for and login to the system to access it and use its features. |

| **User Profile & Home Dashboard** |
|---|
| LongBox users can view their personal profile along with a home dashboard that serves as a portal for the major. |

| **Users Can Browse & Contribute to the Comic Book Catalog** |
|---|
| LongBox users can search the comic book repository of the system, browse the full collection and contribute more items to the collection. |

### Users Can Build a Personal Comic Book Collection

LongBox users can select individual comic books to put into their own personal collections and remove them if they no longer want to keep them.

### Users Can Read and Write Comments About Individual Comic Books

LongBox users can view a comments section associated with all comic books and write new comments. These comments are stamped by username and time.

### Users Receive Recommendations About Trending & Relevant Comic Books

LongBox users are given trending and recommended comic books upon entry to the system and can navigate back to this page for further recommendations

### Users Can View Star Ratings

LongBox users can view the star ratings on comic books. They can also leave their own rating and change them.

## Detailed User Stories For Iteration 3

| | |
|---|---|
| **User can advanced search other users and view their information/all comments - Ahan** | |
| **Associated Big User Story: User Profile and Users can read comments** | |
| **Description:** Using the app, a user can search for users using the app, they can also be able to filter/ search by other fields such as comics reading and finished. Upon finishing the search they can choose to view the profile in depth alongside viewing all the comments that have been made by the user searched for. | |
| Priority: High | Cost: Expected: 5 Actual: 3 |

| | |
|---|---|
| **User views another users profile from comments - Ali** | |
| **Associated Big User Story: User Profile & Home Dashboard** | |
| **Description:** Users can access another user's profile directly from comments within the comic book panel. This streamlined feature improves engagement and interaction within the comic book panel by allowing exploration of profiles of other comic book readers. | |
| Priority: High | Cost: Expected: 5 Actual: 5 |

## User can view trending comic books - Eric

### Associated Big User Story: Users Receive Recommendations About Trending & Relevant Comic Books

**Description:** User is able to navigate to the trending panel and view comics trending based off of their favorites count. When a user favorites and unfavorites a comic, it will impact the trending data

| | |
|---|---|
| Priority: High | Cost: Expected: 3 Actual: 4 |

## Users are recommended personally tailored comic books based on their usage data - Hashir

### Associated Big User Story: Users Receive Recommendations About Trending & Relevant Comic Books

**Description:** Users are provided with personalized recommendations of comic books deemed by the LongBox system to be relevant to their interests. These recommendations are deterministic and based on the preferred genres metric.

| | |
|---|---|
| Priority: High | Cost: Expected: 5 Actual: 3 |

<table>
<tr><td colspan="2"><b>Users can view comic book average star rating and leave their own rating - Oscar</b></td></tr>
<tr><td colspan="2"><b>Associated Big User Story: Users Can View Star Ratings</b></td></tr>
<tr><td colspan="2"><b>Description:</b> Users can view the star rating on comic books left by other users. Users can also add their own star ratings on comic books.</td></tr>
<tr><td>Priority: High</td><td>Cost: Expected: 5 Actual: 5</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Users can change their own rating and recalculate the average - Oscar</b></td></tr>
<tr><td colspan="2"><b>Associated Big User Story: Users Can View Star Ratings</b></td></tr>
<tr><td colspan="2"><b>Description:</b> Users change their own rating and the average star rating will be updated.</td></tr>
<tr><td>Priority: High</td><td>Cost: Expected: 5 Actual: 5</td></tr>
</table>

# Developer Stories For Iteration 3

## Big Stories/Themes

### As a Developer I Want to Conduct Rigorous Testing to Achieve Close to 80% Statement Coverage

Developers of LongBox would like to substantially improve their testing efforts of the system, so that they can verify the system is ready for release.

### As a Developer I Would Like To Refactor Bad Code Smells

There are several bad code smells that have built up in the source code of LongBox in the course of development activities. These design issues and programming pitfalls need to be continually improved for the health of the system.

### As a Developer I Want to Ensure I Have Helped Review Code

Each developer will review 1 out of 5 of all user stories/features built by other developers.

Individual Developer Stories

## As a Developer, I Want to Generate Test Reports To Help Guide My Way to 80% Statement Coverage.

### Associated Big Story: As a Developer I Want to Conduct Rigorous Testing to Achieve at Least 80% Statement Coverage

**Description:** The goal is to create an automated testing pipeline that pairs with a code coverage reporting tool (Jacoco) and the build system (Gradle) to produce up to date statement coverage breakdown by class. The coverage reports will be used to guide the generation of further unit and integration tests. The 80% number does not apply to GUI code that is difficult to test.

| | |
|---|---|
| Priority: Medium | Cost: Expected: 1 Actual: 1 |

## As a Developer, I Want to Finish and Enhance Unit Tests Relevant to Each Feature.

### Associated Big Story: As a Developer I Want to Conduct Rigorous Testing to Achieve at Least 80% Statement Coverage

**Description:** Development of tests in the unit test folder that pertain to domain object and Business Logic classes. I want to ensure I test all branches of the code to get 80% code coverage. The purpose of this story is to go to each unit test folder and improve existing tests, ensure and refactoring related cascades are cleaned up from unit tests and to add any extra tests that were missed in earlier iterations.

| | |
|---|---|
| Priority: Low | Cost: Expected: 3 Actual: 3 |

## As a Developer,  I Want to Create Integration Tests To Test The Seams of My Architecture.

### Associated Big Story: As a Developer I Want to Conduct Rigorous Testing to Achieve at Least 80% Statement Coverage

**Description:** Carefully define and model the seams of LongBox system architecture. Use these seems to create multi layer integration tests that access the concrete postgresql database. These tests will be partitioned into the integration test folder and try to target 80% code coverage for database access layer and GUI/controller.

| | |
|---|---|
| Priority: Med | Cost: Expected: 3 Actual: 3 |

## As a Developer, I Want to Create 3 or More End-2-End Tests For LongBox User Stories

### Associated Big Story: As a Developer I Want to Conduct Rigorous Testing to Achieve at Least 80% Statement Coverage

**Description:** In the process of reviewing each other's user stories, the developers will create at least 3 E2E tests for the stories under review. These tests will be written as detailed step by step scenarios on how a potential user will interact with the system.

| | |
|---|---|
| Priority: High | Cost: Expected: 1 Actual: 1 |

## As a Developer, I Want to Help Resolve The Bad Smells in Domain Object Classes

### Associated Big Story: As a Developer I Would Like To Refactor Bad Code Smells

**Description:** The data transfer objects and entity objects have very complex conversion logic. They are used incorrectly in relation to their purpose. They are Java Beans that are the key objects for the system. They should not focus on providing methods for inter conversion. The constructors are getting confusing and inconsistent. They are hard to maintain. The solution is using separate mapper classes to convert between the two types of beans.

| | |
|---|---|
| Priority: High | Cost: Expected: 3 Actual: 3 |

## As a Developer, I Want to Abstract Data Access to a Service Layer so that the System is Less Coupled

### Associated Big Story: As a Developer I Would Like To Refactor Bad Code Smells

**Description:** There are too many instances of GUI controllers directly calling data access objects. The system should create a service layer that exposes CRUD APIs for various domain objects and their utility functions. The services should communicate with controllers and use dependency injection to further reduce coupling.

| | |
|---|---|
| Priority: High | Cost: Expected: 5 Actual: 5 |

## As a Developer, I Want to Separate Controllers From Swing UI Components

### Associated Big Story: As a Developer I Would Like To Refactor Bad Code Smells

**Description:** Right now the presentation layer has its dynamic actions programmed in anonymous classes and action performed methods in the 3 JFrame classes that exist. This ensures that the actual models (Panel and Table Model classes) are not directly performing logic, however, this can still be taken one level further. This will be done by creating a Controller package with several controller classes for user stories. This will ensure the code is less tightly coupled and the code base is easier to build up in the course of the project.

| | |
|---|---|
| Priority: High | Cost: Expected: 2 Actual: 2 |

## As a Developer, I Want to Help Resolve Code Smells Reported As Issues On The Project GitHub so that the Code Base is Continuously Refactored

### Associated Big Story: As a Developer I Would Like To Refactor Bad Code Smells

**Description:** This involves slowly documenting minor smells and creating github issues tagged as wells as well as a problem report form. This is done in the process of reviewing and by a reviewer. The assignee of the problem is encouraged to quickly fix/close the issue. The nature of these issues will be variable name changes, adding extra test cases, improving code readability, adding comments, etc.

| | |
|---|---|
| Priority: Med | Cost: Expected: 3 Actual: 3 |

## As a Developer, I Want to Help Review At Least ⅕ of All User Stories Implemented

### Associated Big Story: As a Developer I Want to Ensure I Have Helped Review Code

**Description:** Every developer will review a proportion of approximately one fifth of the user stories implemented by other users. This review process will help aid refactoring, bug finding and testing efforts. During the reviewing, developers are encouraged to write problem reports, create e2e tests and detect code smells. As well they can suggest fixes and refactorings.

| | |
|---|---|
| Priority: High | Cost: Expected: 1 Actual: 1 |

## As a Developer, I Want to Help Report Bugs and Issues Using Problem Report Forms and GitHub Issues

### Associated Big Story: As a Developer I Want to Ensure I Have Helped Review Code

**Description:** Every developer will report the bugs and issues they have encountered using problem report forms. They will ensure these problems are tagged with a unique identifier to be used in problem report forms, github issues and commit messages. As well these will be tagged appropriately as either a bug or a smell. It is important to monitor the closed/open status of these reports.

| | |
|---|---|
| Priority: High | Cost: Expected: 1 Actual: 1 |

## As a Developer, I Want to Help Document All Code Smells So That My Review Can Be Actionable

### Associated Big Story: As a Developer I Want to Ensure I Have Helped Review Code

**Description:** In the course of every developer’s review of user story code, they will be responsible for pointing out the nature and location of bad smells. As well they will justify the classification of a smell. Finally they are encouraged to respond with appropriate refactoring techniques to be applied.

| | |
|---|---|
| Priority: Med | Cost: Expected: 1 Actual: 1 |

## As a Developer, I Want the Dto & Entity classes to comply with the JavaBeans standard.

### Associated Big Story: As a Developer I Would Like To Refactor Bad Code Smells

**Description:** While code reviewing it was found that there were too many complex constructors in the DTO/Entity classes. Three code smells are present: long classes, long method and long parameter lists. This is to be improved by removing the constructors from the Dto/Entity classes and using mapper classes to inter convert them.

| | |
|---|---|
| Priority: Med | Cost: Expected: 2 Actual: 2 |

# General Development Tasks:

Eric:

- ☑ Add ComicRepositoryPanel, SearchResultsFrame, and TrendingPanel functionality to their appropriate controller classes
- ☑ Add test cases for trending and continent utils
- ☑ Resolve addressed bugs (improper table loading, double tables, faulty refreshing)
- ☑ Add more comics
- ☑ Address issues of improper navigation when table column are moved around
- ☐ Address github smells

Hashir:

- ☑ Abstraction of database calls to service package classes for cleaner layering.
- ☑ Create mapper classes for all domain objects.
- ☑ Consistent dto <-> entity mapping such that dto and entity are modeled as beans.
  - ☑ Comic Book DTO and Entity classes.
- ☑ Convert all relevant database calls in client code to service layer code.
- ☑ Abstraction of GUI dynamic behavior to controller classes for MVC paradigm.
  - ☑ Home controller
- ☑ Fixed document listener bug for adding the comic book page.
- ☑ Make metrics for recommendations.
- ☑ Create a front end component for viewing recommendations on profile pages.
- ☑ Generate code reports automated through the Gradle build system.
- ☑ Add more users to the user table.
- ☑ Address GitHub Issues.
  - ☑ [BR5] Adding a comic book with number of issues or year published empty #5
  - ☑ [BR6] Adding a comic book with number of issues or year published as text #6
  - ☑ [BR7] Adding a comic book and leaving all fields empty #7
  - ☑ Too big method saveAddComicBookFromInput() #15
  - ☑ reloadTable() method in ProfilePanel contains multiple smells #16
  - ☑ Improper abstraction of database form client code #18

Ahan:

- ☑ Move conversion of String to array and vice versa from ComicBookDto to Utils package.
- ☑ Users can search and filter other users and view their information/all comments.
- ☑ Move controllers to separate packages for user and comments.
- ☑ Update user SQL database and update data in it.
- ☑ Add comments panel to another user frame.
- ☑ Move registration and validate login to another class and make them as static classes.
- ☑ Remove UserDto, User, CommentDto and Comment constructors and use JavaBeans

- ☑ Add a new frame to display ‘Another User’

Oscar:

- ☑ Add star rating system in comic book pages
- ☑ Display average rating score by other users
- ☑ Allow users to add their own rating
- ☑ Allow users to change their ratings
- ☑ Create SQL database for star rating data for each comic book
- ☑ Add test cases for star rating system
- ☑ Refactor “about me” section in profile panel to use Profile Controller
- ☑ Address GitHub Issues
    - ☑ [BR4] The text is cut off in the about me text box if a big paragraph is added.

Ali:

- ☐ Add mouse listener to Jlabel within Comments
- ☐ Add panel and GUI of another user

# Table of Features

| Feature | Related Big User Story | Iteration of Completion |
|---|---|---|
| Login and Registration | User Login and Registration | 1 |
| View all comics in the repository | User Profile & Home Dashboard | 1 |
| View personal profile | User Profile & Home Dashboard | 1 |
| Add a comic to collection | Users Can Browse & Contribute to the Comic Book Catalog | 1 |
| Basic comic book search | Users Can Browse & Contribute to the Comic Book Catalog | 1 |
| Searching / Advanced Searching, Viewing Comic Information | Users Can Browse & Contribute to the Comic Book Catalog | 2 |
| View Comments | Users can read and write comments about comic books | 2 |
| Add Comments | Users can read and write comments about comic books | 2 |
| Favorite a comic | Users Can Build a Personal Comic Book Collection | 2 |
| Unfavorite a comic | Users Can Build a Personal Comic Book Collection | 2 |
| Mark a comic as reading view its list | User profile & home dashboard | 2 |
| Mark a comic as finished reading and its list | User profile & home dashboard | 2 |
| About Me section | User Profile & Home Dashboard | 2 |
| Trending Comics | Users Receive Recommendations About Trending & Relevant Comic Books | 3 |

| | | |
|---|---|---|
| Recommended Comics | Users Receive Recommendations About Trending & Relevant Comic Books | 3 |
| Users can search for other users using the app | User Profile and Users can read comments | 3 |
| Add a star rating and view average star rating | Users Can Browse & Contribute to the Comic Book Catalog | 3 |
| Editing star rating | Users Can Browse & Contribute to the Comic Book Catalog | 3 |
| User views another users profile from comments | Users can read and write comments about comic books | 3 |

# Revised Stories: Pushed to ITR3

As mentioned in the ITR2 document these stories were pushed to ITR3.

<table>
<tr><th colspan="2">~~Upon login users are recommended comic books to browse - Hashir~~</th></tr>
<tr><td colspan="2">~~**Associated Big User Story: Users Receive Recommendations About Trending & Relevant Comic Books**~~</td></tr>
<tr><td colspan="2">~~**Description:** As a user begins to favorite comic books and comment on comic books, the system will use this information to recommend comic books to users. The recommendations are shown on the landing page that is redirected to upon login. The predictions are a best effort and are not guaranteed to be fully relevant to every user.~~</td></tr>
<tr><td>~~Priority: High~~</td><td>~~Cost: Expected: 5~~</td></tr>
</table>

<table>
<tr><th colspan="2">~~Users can view a list of trending comic books - Eric~~</th></tr>
<tr><td colspan="2">~~**Associated Big User Story: Users Receive Recommendations About Trending & Relevant Comic Books**~~</td></tr>
<tr><td colspan="2">~~**Description:** The users of LongBox have an option to view a page of trending comics that have had active comment threads (>10 comments in the last 14 days), a high number of users favoriting them (>10 users) and many users reading/finishing them (>10 users having read/finished them).~~</td></tr>
<tr><td>~~Priority: Low~~</td><td>~~Cost: Expected: 5~~</td></tr>
</table>

# Development Tasks Assigned In Iteration 1 and 2

## Iteration 1 User Stories

<table>
<tr><th colspan="2">Users Can Register For LongBox - Ahan</th></tr>
<tr><td colspan="2"><b>Associated Big User Story: Users Login &amp; Registration</b></td></tr>
<tr><td colspan="2"><b>Description:</b> Before using LongBox users must provide basic personal information such as date of birth, full name, email address and country of residence. Furthermore, users must provide a unique username that has not been used by any other member of the system before and must also provide a strong password. If the criteria are not met the user is not permitted to register</td></tr>
<tr><td>Priority: High</td><td>Cost:  Expected: 3 Actual: 5</td></tr>
</table>

<table>
<tr><th colspan="2">Users With A Registered Account Can Login To LongBox - Ahan</th></tr>
<tr><td colspan="2"><b>Associated Big User Story: Users Login &amp; Registration</b></td></tr>
<tr><td colspan="2"><b>Description:</b> After successfully creating an account or already having an account, a user will be required to enter a username and password to enter into the application with a user session if the login information matches with an existing account.</td></tr>
<tr><td>Priority: High</td><td>Cost: Expected: 3 Actual: 5</td></tr>
</table>

## Once Logged In, Users Remain Signed into the Application and Maintain This State - Ahan

**Associated Big User Story: Users Login & Registration**

**Description:** After a successful login, a new user session will be created for said user. This session will remain active for the duration of the users stay and tailor the profile, comic collection, and recommended pages to the users requirements.

| | |
|---|---|
| Priority: Medium | Cost: Expected: 1 Actual: 1 |

## Once Logged In, Users Can Choose To Logout At Any Time - Ahan

**Associated Big User Story: User Profile & Home Dashboard**

**Description:** Once logged in, the user will be able to logout via the ‘logout’ button on the top right of the menu bar. This will terminate the user session (log them out) and redirect the user back to the login page.

| | |
|---|---|
| Priority: High | Cost: Expected: 1 Actual: 1 |

## Users Have Access To A Menu Of Links To All Aspects of the Home Dashboard/Profile - Hashir

**Associated Big User Story: User Profile & Home Dashboard**

**Description:** After a successful login, the menu bar will become visible with all the available tabs (Comic Collection, Search, Add Comic, Profile, and Log Out). The user can select any of these and navigate to them, while the menu bar still remains present.

| | |
|---|---|
| Priority: Low | Cost: Expected: 2 Actual: 2 |

## Users Can View Their Personal Profile Information - Hashir

### Associated Big User Story: User Profile & Home Dashboard

**Description:** After a successful login, the users can view their profile details in text form with one click to get to the user profile screen. This screen is read only and must be synchronized at all times with the user that is logged in and no others. The only field not visible is the password.

| | |
|---|---|
| Priority: Medium | Cost: Expected: 2 Actual: 2 |

## Users Can Browse The Full Catalog of Comic Books In The System’s Repository - Eric & Hashir

### Associated Big User Story: Users Can Browse & Contribute to the Comic Book Catalog

**Description:** A user should have a one click option to view the full list of comic books in the system’s catalog in tabular form by the individual fields that a comic book is described by. The User can choose to sort these items by columns.

| | |
|---|---|
| Priority: Medium | Cost: Expected: 3 Actual: 7 |

## Users Can Add A Comic Book To The System Repository - Ali & Hashir

**Associated Big User Story: Users Can Browse & Contribute to the Comic Book Catalog**

**Description:** Users can fill out a form to add a comic book to the system repository by giving all information they know about it. The form is not strict and is lenient on empty fields as users may not have all information available to them and the purpose is to slowly build the catalog based on users best effort and knowledge about the comic books.

| | |
|---|---|
| Priority: High | Cost: Expected: 3, Actual: 1 |

## Users Can Search For Individual Comic Books By Series Title - Hashir

**Associated Big User Story: Users Can Browse & Contribute to the Comic Book Catalog**

**Description:** In the search tab of the menu bar, the user will be able to enter the title of a comic series and the system shall return the result, if it exists. If the result does not exist the user will be prompted that there is no such item. If the result does exist then the user will be guided to a page showing the basic details of the comic book.

| | |
|---|---|
| Priority: Medium | Cost: Expected: 2 Actual: 2 |

Iteration 2 User Stories

<table>
<tr><td colspan="2"><b>Users Can Write Comments On Individual Comic Books - Ahan</b></td></tr>
<tr><td colspan="2"><b>Associated Big User Story: Users Can Read and Write Comments About Individual Comic Books</b></td></tr>
<tr><td colspan="2"><b>Description:</b> Users can share their thoughts by writing comments about a comic. Once the user adds a comment they should be able to see the comment in the comment list.</td></tr>
<tr><td>Priority: High</td><td>Cost: Expected: 3 Actual: 2</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Users can View All Comments On Individual Comic Books - Ahan</b></td></tr>
<tr><td colspan="2"><b>Associated Big User Story: Users Can Read and Write Comments About Individual Comic Books</b></td></tr>
<tr><td colspan="2"><b>Description:</b> Users can view all the comments on a comic book. The comments that are viewed are in order from newest to oldest. The user should also be able to view the name of the user who commented on the feature along with when the comment was added.</td></tr>
<tr><td>Priority: Medium</td><td>Cost: Expected: 3 Actual: 3</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Users Must Use a Unique Email to Make an Account that Does Not Associate With Any Other User - Ahan</b></td></tr>
<tr><td colspan="2"><b>Associated Big User Story: Users Login &amp; Registration</b></td></tr>
<tr><td colspan="2"><b>Description:</b> Each user should have a unique email along with username when registering for the system.</td></tr>
<tr><td>Priority: Low</td><td>Cost: Expected: 1 Actual: 1</td></tr>
</table>

<table>
<tr><td colspan="2"><b>User can Search for Comic by Title, Genre, Publisher, and Year - Eric</b></td></tr>
<tr><td colspan="2"><b>Associated Big User Story: Users Can Browse &amp; Contribute to the Comic Book Catalog</b></td></tr>
<tr><td colspan="2"><b>Description:</b> User can search and filter results based on title, genre, publisher, and year. User will be presented with a new small frame displaying all the results according to the desired search entry</td></tr>
<tr><td>Priority: High</td><td>Cost: Expected: 3 Actual: 5</td></tr>
</table>

### User Can Navigate on a Comic Book Page by Clicking its Title if it Exists in a Results Table and Comic Collection - Eric

**Associated Big User Story: Users Can Browse & Contribute to the Comic Book Catalog**

**Description:** In the comic collection and results window. A User can navigate to a comic page by clicking on the title of said comic. If it is from the search results page, then it will dispose of the results window after opening the comic page

| | |
|---|---|
| Priority: Medium | Cost: Expected: 2 Actual: 2 |

### Users Can Sort A Comic Table Based on Every Attribute that a Comic Has - Eric

**Associated Big User Story: Users Can Browse & Contribute to the Comic Book Catalog**

**Description:** In the comic collection tab, a user can sort the entire table based on title, author, artist, publisher, issue count, and year. These will all be based off of an alphabetical ordering

| | |
|---|---|
| Priority: Low | Cost: Expected: 2 Actual: 4 |

## Users Can Add/Remove Comic Books To/From A List Of Finished Comic Books: Hashir

**Associated Big User Story: Users Can Build a Personal Comic Book Collection**

**Description:** In the user profile view the logged in user should be able to view a list of comic books they have finished reading. This list is built by them and they indicate what items they have finished individually. The additions to this list are done when viewing a comic book. The data built and stored here will serve as a source for data that the user recommendations are based off of. Items can also be removed from this list

| | |
|---|---|
| Priority: High | Cost: Expected: 4 Actual: 4 |

## Users Can Add/Remove Comic Books To/From A List Of Reading In Progress Comic Books: Hashir

**Associated Big User Story: Users Can Build a Personal Comic Book Collection**

**Description:** Similar to the previous story, a user can add comic books to a list they build themselves of comic books they are currently reading. This list can be edited by the user at any time and is viewable in the profile page. This will also serve as data for user recommendations.

| | |
|---|---|
| Priority: Medium | Cost: Expected: 4 Actual: 4 |

<table>
<tr><td colspan="2"><b>Users Can Login Using Email Address and Password - Eric</b></td></tr>
<tr><td colspan="2"><b>Associated Big User Story: User Login and Registration</b></td></tr>
<tr><td colspan="2"><b>Description:</b> Users at the login page can log in both with a matching email address and password or username and password; usernames at registration will no longer access the char ‘@’</td></tr>
<tr><td>Priority: Low</td><td>Cost: Expected: 2 Actual: 1</td></tr>
</table>

<table>
<tr><td colspan="2"><b>User can edit and save “about me” text box in profile page - Oscar</b></td></tr>
<tr><td colspan="2"><b>Associated Big User Story: User Profile &amp; Home Dashboard</b></td></tr>
<tr><td colspan="2"><b>Description:</b> In the profile page, users can type and edit the text in the “about me” text box to describe themselves. Will have an edit, save, and cancel button that the user can click in order to be able to change the text, save it, or cancel any changes.</td></tr>
<tr><td>Priority: Medium</td><td>Cost: Expected: 2 Actual: 2</td></tr>
</table>

## When adding a comic book to the repository a user can directly favorite it at the same time - Ali

**Associated Big User Story: Users Can Build a Personal Comic Book Collection**

**Description:** In the 'Add Comic' page, users can mark a comic as favorite by selecting the 'Is Favorite' checkbox. This action adds the comic to a merged table in the database, linking user and comic entities, and displays it in the 'Favorites' tab of the GUI panel.

| | |
|---|---|
| Priority: Medium | Cost: Expected: 3 Actual: 3 |

## Users can remove a comic book from favorites - Ali

**Associated Big User Story: Users Can Build a Personal Comic Book Collection**

**Description:** In the 'Favorites' tab, users can deselect a comic as a favorite by selecting it and clicking the 'Unfavorite' button. This removes the comic from the merged database table linking user and comic entities.

| | |
|---|---|
| Priority: Medium | Cost: Expected: 3 Actual: 3 |

# Customer Interviews

The customer is Rocco Ali, an early 20s male who is interested in learning more about comic books. He wants to use an application that streamlines his introduction to the comic book world. Two rounds of interviews were done with him for this project. A first round to propose the project and build a system vision and a second round to show what has been delivered. As well in the second round acceptance of the system is determined as well as any miscommunications about what was to be made.

## Part 1: Project Proposal

This interview took place on January 24th, 2024. A full video can be found at: customer video 1.

The notes for the interview are also provided below.

Q1: What got you interested in comics?
I liked watching superhero movies as a kid and when I started going to the library I used to rent graphic novels based on those movies. Then when I was in high school and started to have disposable income I would buy comic books at both indigo and comic books stores.

Q2: What comics interest you the most? Are there any specific publishers you look towards?
The comics that interest me the most are superhero comics. I like the big two publishers Marvel and DC as well as smaller publishers such as Image, Dark Horse, and Eclipse.

Q3: How do you get your information about comics? Do you find you learn more about titles through going directly to a comic book store or through the internet, such as YouTube?
Mostly through reviews on Youtube as well as several other publishers.

Q4: How many comics do you own and how frequently do you purchase them?
I have around 80 comic books and graphic novels. I purchase them when I find a title that I enjoy.

Q5: When you do purchase comics, do you find it easier by going to a physical comic book store, or through the internet.
I usually like to do both purchasing through stores and through the internet.

Q6: What are some factors about the comics industry that prevent you from purchasing more comics than you already do? Would price, distribution, and lack of reprints be a factor?
Newer prints do not interest me as much. I would say I'm a picky consumer and would only purchase a product after I see a positive review about it. Price does not have an impact on me, Neither does distribution. Lack of reprint does have an effect since it just means that sometimes they would be lost to the media.

Q7: What could be done about any of these factors to improve your interest in comics?
Better visibility of user reviews as well as a bestseller section.
Q8: Do you find that older comics, say from the 60s - 80s, interest you more than modern age comics?
Older comics do interest me more than modern comics.

Q9: For these older comics, you may discover that no reprints of them exist, some may have been out of print since the 90s! When you want to access one of these titles, do you find it easier to visit a physical comic shop in hopes that they may have it in a longbox or resorting to internet piracy.
Yes I would support trying to find them first at a physical store, and then attempt in the second hand market before taking to the high seas. When there is usually an ease of access to comics In physical print, there are less reasons for piracy to exist.

Q10: Despite many older comics not having legal means to access them, it is still considered unethical and outright illegal to do so. Do you believe that a legal and digital archival library of these titles is an appropriate means to access them?
Yes, it would be good to preserve comics since they can be considered an art form.

Q11: What would you want from this hypothetical application? Elaborate and go into detail.
Some things that I would like from this application is

- Website that is easy and simple to navigate
- A website with a clear front page for best sellers
- A visible search bar to search for titles directly
- A website with filters to filter through which comics I want based on:
  - Years printed (eg 1975-1979 or 70s)
  - Publisher name
  - Title of book
  - Edition (for example one with a specific variant comic, reprints, special editions,
  - etc)
- A clear page for each comic title
- A star rating based on user reviews (rating it from 1-5 stars)
- A clear way to see user reviews

Q12: What do you believe can be done to rejuvenate public interest into the comics medium? Do you believe that this application can contribute to that cause?
There are several different things that can be done to rejuvenate interest in the comic book industry. One of those things would be easy accessibility to cannon events so that new readers can get into comics without feeling lost. This platform can be a useful way in solving this problem since this platform can make it easy for users to find each of the different titles by just searching for them.

# Part 2: Post-Mortem Analysis

This interview took place on March 28th, 2024. A full video can be found at: Customer Video 2.

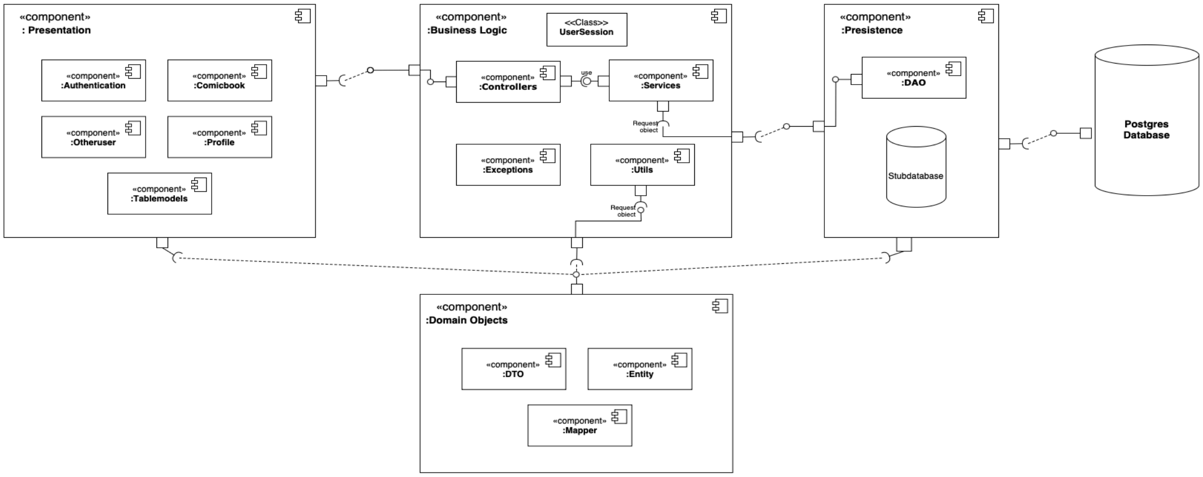
The notes for this interview are provided below as a comprehensive “post-mortem analysis” of the prototype that has been released by the delivery 2 deadline. As well there is a large focus placed not only on the product but also the project management. This is to provide a retrospective so that problems can be addressed for future projects.

## Architecture

The overall architecture was a 3 layer architecture very similar to the one provided in the course lectures. However it had many additional elements that were realized by exploring alternative technologies and consulting java best practices for projects of this nature.

The GUI was developed around the MVC framework using Swing. The Domain objects were modeled using the Dto pattern for separation between database objects and client side objects. Furthermore, the data access layer used the dao pattern for specialization of queries by grouping them according to domain objects. Other design elements included the use of an ORM (hibernate) to abstract database mapping and connectivity. As well, the user session and database session factory both worked as a singleton. There were also many uses of dependency injection.

The final architecture is shown below.



One other thing to note is that the database schema has evolved as time progressed and there are more tables now than in the early phases of the project. As well, SQL triggers are used.

## Source Code Metrics

These metrics were generated by using the IntelliJ IDEA plugin called Statistic. Overall this was what we expected the project to encompass. The data below is only from the source; the test is excluded.

| **Package/Layer** | **Number of Classes** | **Lines of Code** |
|---|---|---|
| Business logic | 28 | 1792 |
| Domain objects | 20 | 842 |
| Persistence | 22 | 1664 |
| Presentation | 21 | 2582 |
| **Total** | 133 | 10 364 |

The distribution is as expected. The Swing GUI code and its controllers in the business logic take up a lot of code. As well, the database access methods are the third highest contributor. The domain objects being the smallest chunk of code was intended. The objects here are like Java Beans. They only use no argos constructors, getters, setters, equals/hashcode and toString.

## Story Point Review

| **Stories** | **Story Points Estimated** | **Story Points Actual** | **Discrepancy (estimated - actual)** |
|---|---|---|---|
| ITR 01 | 24 | 29 | -5 |
| ITR 02 | 32 | 34 | -2 |
| ITR 03 (User + Developer) | 26 + 21 | 27 + 21 | -1 |
| User stories | 82 | 90 | -8 |
| Developer stories | 21 | 21 | 0 |
| All Stories | 103 | 111 | -8 |

The table below shows this by user but omits developer stories. It shows only user stories.

| Team Member | User Story Points Estimated | User Story Points Actual | Discrepancy (estimated - actual) |
|---|---|---|---|
| Ahan | 20 | 21 | -1 |
| Ali | 14 | 12 | -2 |
| Eric | 15 | 23 | -8 |
| Hashir | 25 | 25 | 0 |
| Oscar | 12 | 12 | 0 |

The story points went as expected. Slightly underestimated throughout but near the end the estimate was practically the same. The team understood better what they can accomplish in a given time. The feature that took the most amount of time was the star ratings system. The feature that took the least amount of time was recommendations. Finally, a major surprise was the addition of developer stories near the end. We were not expecting it to occur but we managed it and prioritized what we could do in the time and excluded anything that was too ambitious.

## Things that went right in the development process

- The team was always available to meet and discuss ideas
- Almost all work was done on time aside from a small number of isolated situations
- No deliverable or iteration went late. Everything was usually ahead of schedule or just on time
- The group members learned each others strengths and weaknesses to complement the team
- Group members were available to pair program or provide help when others got stuck
- Debugging in groups usually lead to very fast resolution of problems
- Design decisions were never drastically changed
- There was continuous refactoring

## Things that went wrong in the development process

- Some group members were falling behind at the start and not meeting the obligations to the same standards as other group members.
- Skill levels in the group were not consistent. Some members had a very easy time with the code base while others had a hard time due to the higher skill members moving forward at a faster pace.
- Ownership of tasks in iteration 1 had a slight conflict that was later resolved.

- One group member had to drop out of the course due to a family emergency.

## Things that would be done differently if starting over

Initially, we should have had an immediate separation between the GUI panels and their associated controllers. There are three main reasons as to why this would be a superior way to start off the project.

1. Many professional projects separate GUI code and logic, which makes debugging and modifications easier
2. It would have saved time as it would not involve much of a big refactoring step along the way
3. It would have made the team used to working with distinct GUI and controller files, rather than having to learn how to work with a new approach after the refactor

## Design Decisions: Good and Bad

1. We decided to separate the client code from the database via a service layer, which makes our code secure and maintainable in future.
2. We also decided to separate the controller logic from GUI logic, which also made our code clean, better readable, and maintainable.
3. Less custom exceptions being used, having more custom exceptions would make our code more readable and maintainable.

## Outstanding Bugs

There are still some outstanding problem reports. In total there were 25 GitHub issues created with associated problem report forms created in the Lab 5 code review phase. The majority of the issues have been closed. 17 issues in total were closed and there are still 8 open issues. The reasoning for the issues staying open is given within the LongBox GitHub Issues page. As well you can find the resolutions for these issues. Detailed problem reports are found in the lab 5 team document.

## Features that worked better than expected

During the interview, our customer Rocco stated a few things that impressed him.

1. Easy User Flow: it was very easy to navigate through the program, the card layout made it easy to switch between the panels and navigate to different areas of the project
2. Searching: The advanced search within the Comic Repo and Favorites make it easy to search for comics based on their attributes
3. Navigating to Comic Pages: Almost all panels/frames involving the display of comic books and users can navigate to the appropriate comic book and user

4. Social: The social section was praised for its ability to display and rank users based on the number of comics they have read and are reading, creating a sense of competition among the users

## Additional Technologies Used

1. Swing
2. Hibernate
3. JaCoCo
4. Lombok
5. Gson

## Project Scope Changes

The major changes that occurred were a shift of focus from displaying pdf's, pictures and the look of the GUI. The focus shifted towards information organization, database integrity, user interaction, recommendations/trends and leaderboards. This was done because at first we were going to have two databases. One SQL database for the basic data and a NoSQL database for the pictures and pdf's. To ensure that the project is easier to handle/build and given the lack of emphasis on the visual look/feel being the key, we decided to avoid having two databases and a pdf reader as they increase complexity substantially. As well these services are available elsewhere. The project focussed on creating a good foundation so that it can be extended to a web application later. The current GUI will essentially be phased out and a javascript GUI will be made. The "backend" was designed to allow for upgrades like this to occur at a later date as members of the group have the intention to continue the project after the course.

## Lessons Learned

We learned that a large project involving a team is no easy task. It requires a lot of coordination and conflict management. As well, we learned how to quickly catch up with team members who are not as exposed to the technologies that are being worked on. Not everyone will have the same skill level. This is okay. What matters more is that those who perceive they have a knowledge gap must make their teammates aware and work with them through peer to peer learning by working together to solve problems that one person may not be able to solve alone. As well the help needs to be in the form of active learning not by just giving each other all the answers. We need to ensure that each person struggles just the right amount so that they are learning a lot but also still accomplishing a lot.

## Conclusions

The project met the customers requirements and satisfies the desires of the comic fans. All of what was described from the first interview was met and at points exceeded the requirements. Some minor missing feature and

However, some features such as user navigating to a comic info page from the trending panel and when viewing another user's pane/pagel were unimplemented at this time. The lack of these features breaks user flow, as all the other tables can navigate to comic book info pages by double clicking them. The navigation to a comic books info page was added after the interview, and functions properly for both tables. Navigation to a comic book info page from viewing another user's info page has not happened yet, as it requires passing a user session causing more entanglement to manage with.

Moreover, another complaint was the overall look of the application, which relies heavily on the default swing graphics. The main critique was that it looked too much like an early 2000s application with the color scheme and graphics.

Indeed, with all of the main features fully implemented and functioning properly, it would be appropriate to work on a GUI overall of the project including graphics to match the comic book theme of the application, more color diversity, cleaner swing component designs, and more diverse fonts. On top of the graphical changes, adding the navigation to comic book and user info pages to all appropriate tables would be a benefit as it would increase user flow within the application.

To add, another major change to add would be having all frames within the home frame. Currently, the search results, user info, and comic book info pages appear as separate frames from the main one. This can create clutter as a user may unintentionally open multiple of these frames without closing them. Having these three frames integrated to the home frame would greatly increase user flow and reduce clutter, thus further improving the user experience.

To conclude, the LongBox project has been a great learning experience and in achieving the user requirements from both the first user interview and the planning documents. While all of the big features of the customer were met, some minor additions to info panel navigation would increase user experience as it would refrain from the user having to manually search for comics and users. Additionally, a complete overhaul of the GUI away from the default Swing look and adapting all frames into the main home frame would greatly increase user flow. As with all projects, there is always room for further improvement. A good team will always look for this ahead of time to tackle.