# <Project Name> Release Summary (6 pages MAX)

## Team members

|  |  |  |
| --- | --- | --- |
| Name and Student ID | GitHub ID | The number of story points (or ideal hours for tasks) that a member was an **author**. |
| **Aaron Le** | **adl409** | 140 Story Points |
| Wen Chen | wchen2654 | 150 Story Points |
| Nathan Lechner | Na7hanL | 130 Story Points |
| Scott Hoerchler | Rien98 | 130 Story Points |

Each group member is responsible for counting their own story points. It is the group leader’s duty and responsibility to make sure they are accurate. Please keep in mind that we will check your GitHub stats (go to: “Graphs” on your GitHub project page, for [example](https://github.com/marouen-lamiri/Second-Soul/graphs/contributors)). Note, if your email and GitHub id are not linked properly you will not be counted properly.

You will lose 1 mark if links below are not clickable.

## Project summary (max one paragraph)

An e-commerce store that allows users to buy and sell books via the web application. The web application implements Node.js as its framework language and MySQL as its database.

## Velocity and a list of user stories (for [example](https://github.com/solid/user-stories)) and non-story tasks for each iteration

(make sure the iteration is a clickable link to the milestone/sprint on GitHub)

24 stories over 25 weeks 24 \* 25 = 600 story points

Total: 24 stories, 600 points over 25 weeks

Iteration 1 (3 stories, 100 points) 🡪 No User Stories created

Iteration 2 (7 stories, 120 points) 🡪 Began creating User Stories

Iteration 3(11 stories, 200 points)

Iteration 4, Release(2 stories, 180 points)

## Overall Arch and Design

Show us the overall architecture (block diagram) in your system with an architecture diagram.

Show the UML class diagram for your system. If you have multiple packages, show the diagram for at least one package that has more than 10 classes. You can also include these diagrams in your stories on GitHub (by providing URLs).

Activity and State Diagram: <https://github.com/adl409/SE-project-Group-5/tree/main/activityDiagramXMLS>

Class Diagram: <https://github.com/adl409/SE-project-Group-5/tree/main/classDiagramXML>

Component Diagram: <https://github.com/adl409/SE-project-Group-5/tree/main/componentDiagramXMLS>

Sequence Diagram: <https://github.com/adl409/SE-project-Group-5/tree/main/sequenceDiagramXMLS>

Use case Diagram: <https://github.com/adl409/SE-project-Group-5/tree/main/useCaseDiagramXMLS>

Web Architecture Diagram: <https://github.com/adl409/SE-project-Group-5/tree/main/webArchitectureDiagramXMLS>

## Infrastructure

Framework: Node.js

Database: MySQL

**Name and link**

**Node.js:** [**https://nodejs.org/en**](https://nodejs.org/en)

**MySQL:** [**https://www.mysql.com**](https://www.mysql.com)

Max 1 paragraph description of why you are using this framework.

We had decided on Node.js because none of the members had really used it before and thought it would be a good thing to learn and get experience using it. Node.js is a great framework for building a web application and when paired with jest for testing can become straight forward as to how to get the web application running.

Max 1 paragraph description of other alternatives and why you didn’t choose them.

We had thought about python and its capabilities of building web applications. Although members have experience in building applications with python, we decided not to go with a python framework due to the desire to go with Node.js.

## Name Conventions

List your naming conventions or just provide a link to the standard ones used online.

For example: [Java naming conventions](http://www.oracle.com/technetwork/java/codeconventions-135099.html)

Naming conventions used:[Java naming conventions](http://www.oracle.com/technetwork/java/codeconventions-135099.html)

Naming conventions used: [Node.js naming conventions](https://github.com/felixge/node-style-guide)

## Code

Key files: top **5** most important files (full path). We will also be randomly checking the code quality of files. Please let us know if there are parts of the system that are stubs or are a prototype so we grade these accordingly.

|  |  |
| --- | --- |
| File path with a clickable GitHub link | Purpose (1 line description) |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/index.js> | This is the index for the application which is the backbone of all js code we have. |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Classes/admin.js> | This is the admin class which controls the admin portion of the application. |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Classes/buyer.js> | This is the buyer class which controls the buyer portion of the application. |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Classes/owner.js> | This is the owner class which controls the owner portion of the application. |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Classes/seller.js> | This is the seller class which controls the seller portion of the application. |

## Testing and Continuous Integration

Each story needs a test before it is complete. If some class/methods are missing unit tests, please describe why and how you are checking their quality. Please describe any unusual/unique aspects of your testing approach.

List the **5** most important unit test with links below.

|  |  |
| --- | --- |
| Test File path with clickable GitHub link | What is it testing (1 line description) |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Unit_Test/createaccount.test.js> | Create account functionality |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Unit_Test/buyer.test.js> | Buyer functionality |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Unit_Test/seller.test.js> | Seller functionality |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Unit_Test/setadmin.test.js> | Set admin functionality |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Unit_Test/login.test.js> | Login functionality |

List the **5** most important acceptance tests with links below.

|  |  |
| --- | --- |
| Test File path (if you automated the test) or as comments in Github issues (if it’s done manually) with clickable GitHub link | Which user story is it testing (1 line description) |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Unit_Test/createaccount.test.js> | Buyer user story 7 |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Unit_Test/login.test.js> | Buyer 4/ Seller 3/ Admin 2/ Owner 5 |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Unit_Test/setadmin.test.js> | Admin 3 |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Unit_Test/seller.test.js> | Seller 4 |
| <https://github.com/adl409/SE-project-Group-5/blob/main/Code/Unit_Test/block.test.js> | Owner 3 |

Describe your continuous integration environment. Include a link to your CI.

We used Github actions for CI testing for ease of use purposes when it comes to having all of our tests and information in one place.

Link: <https://github.com/adl409/SE-project-Group-5/actions>

Describe the choice of the static analysis tool and how do you run it. The static analysis tool should analyze the language that is used in the majority of your source code.

Attach a report as an appendix (not counted for the 6 pages) from static analysis tools by running the static analysis tool on your source code. Randomly select 10 detected problems and discuss what you see.