

Sprint Planning & Execution

Contents

Sprint 1	2
The targeted increment:	2
Requirements Elaboration:	5
Design	5
Planning and Estimation:	5
Development	6
Testing	6
Integration	7
Documentation	8
Approval	9
Review & Retrospective	9
Requirements Activities & Models	10
Functional requirements	10
Non-functional requirements	11
Use-Cases	11
Use case diagrams	15
User interviews	15
Customer interviews	16
Surveys	16
Design Activities & Models	17
Class Diagram	17
Deployment Diagram	17
Test Suites	18

Sprint 1

The targeted increment:

- The target increment for this sprint is to complete Level1 of the game. Level 1 demonstrates how to conduct a merge sort using an animated tutorial with 10 randomly generated numbers. To break down and distribute the creation of level 1 we created several stories and child issues on Jira, as shown below.

Sprint 1 Jira Stories:

Release 1 MVP / SG1-24

Create Level 1

AttachAdd a child issueLink issue

Description

As a user, I can play Level 1 so I can learn how the sorting algorithm works.

Child issues

	Order by	...	+
SG1-37 Write Merge Sort Instructions	EL	TO DO	
SG1-39 Random Number Generation	AN	TO DO	
SG1-40 Merge Sort Algorithm	WIG	TO DO	
SG1-41 Sort Algorithm Variables for Animation	MT	TO DO	
SG1-50 Algorithm Breakpoints to Pass to Frontend	MH	TO DO	

MH

Add a comment...

Pro tip: press **M** to comment

In Progress

Details

Assignee

Unassigned

Labels

None

Sprint

SG1 Sprint 1

Story point estimate

None

Development

Create Branch

Reporter

MH Maisie Harrington

Created 8 days ago

Updated 23 hours ago

Configure

Release 1 MVP / SG1-15

Animations

AttachAdd a child issueLink issue

Description

As a user, I can go through the game experience with a visually pleasing design.

As a student, I want the algorithm to be introduced to me through animation so that it is easy to follow along and understand.

Activity

Show: AllCommentsHistory

MH

Add a comment...

Pro tip: press **M** to comment

In Progress

Details

Assignee

CM Christopher Mohan

Labels

None

Sprint

SG1 Sprint 1

Story point estimate

None

Development

Create Branch

Reporter

MH Maisie Harrington

Created 8 days ago

Updated 17 minutes ago

Configure

Sprint 1 Jira Child Issues:

SG1-24 / SG1-37

1 1 1 1 1

Write Merge Sort Instructions

Attach Link issue

Description

As a user, I can see the instructions for merge sort as I play

Activity

Show: All Comments History Newest first

MH

Add a comment...

Pro tip: press M to comment

To Do

Details

Assignee: EL Elliot Lam

Labels: None

Sprint: SG1 Sprint 1

Story point estimate: None

Development: Create Branch

Reporter: CM Christopher Mohan

Created 8 days ago Updated 8 days ago

Configure

SG1-24 / SG1-39

1 1 1 1 1

Random Number Generation

Attach Link issue

Description

As a user, I can repeatedly play the same level using different numbers.

As an admin, I want the numbers to be randomly generated so that students actually learn how to do the algorithm rather than memorizing the process for that set of numbers.

Activity

Show: All Comments History Newest first

MH

Add a comment...

Pro tip: press M to comment

To Do

Details

Assignee: AN Alex Nelson

Labels: None

Sprint: SG1 Sprint 1

Story point estimate: None

Development: Create Branch

Reporter: CM Christopher Mohan

Created 8 days ago Updated 8 days ago

Configure

SG1-24 / SG1-40

1 1 1 3 1

Merge Sort Algorithm

Attach Link issue 3

Description

As a student, I want play the game using a merge sort so I can add to my learning of algorithms.

Activity

Show: All Comments History Newest first 17


MH

Add a comment...

Pro tip: press **M** to comment

To Do

Details


Assignee Megan Gingham

LabelsNone

SprintSG1 Sprint 1

Story point estimateNone

Development[Create Branch](#)

Reporter Christopher Mohan

Created 8 days ago Updated 8 days ago

Configure

SG1-24 / SG1-41

1 1 1 3 1

Sort Algorithm Variables for Animation

Attach Link issue 3

Description

As a student, I want the algorithm to be introduced to me through animation so that it is easy to follow along and understand.

Activity

Show: All Comments History Newest first 17


MH

Add a comment...

Pro tip: press **M** to comment

To Do

Details


Assignee Michael Taran

LabelsNone

SprintSG1 Sprint 1

Story point estimateNone

Development[Create Branch](#)

Reporter Christopher Mohan

Created 8 days ago Updated 8 days ago

Configure

SG1-24 / SG1-50

1 1 1 3 1

Algorithm Breakpoints to Pass to Frontend

Attach Link issue 3

Description

As a student, I want the instructions to be presented to me one step at a time so that I can easily learn how the algorithm functions before playing.

Activity

Show: All Comments History Newest first 17


MH

Add a comment...

Pro tip: press **M** to comment

To Do

Details


Assignee Maisie Harrington

LabelsNone

SprintSG1 Sprint 1

Story point estimateNone

Development[Create Branch](#)

Reporter Alex Nelson

Created 8 days ago Updated 8 days ago

Configure

Requirements Elaboration:

Here we define the main requirements for level 1 of the project:

- Merge-Sort Algorithm
- Website layout
- random number generation
- Animations / Bringing information onto page
- Define algorithm instructions

Updates For Sprint 1 (See Requirements Activities and Models section below):

- Added functional and non-functional requirements
- Created use cases
- Implemented use cases in use case diagram
- Conducted user interviews

Design

During this sprint, we created two high-level design activities to help us better understand the implementation. Primarily, it was important to create a class diagram before splitting up and assigning programming work, because the class diagram helped us determine how the different components would work together. Our diagram features two main components - the frontend and the sort algorithm, and has a third component called an InstructionQueue that acts as a data structure for the two main components to communicate with. After identifying these key components, it became clear that the two main components should be separated across different deployment objects. We then created a deployment diagram that demonstrated how the frontend and backend would be linked. This helped us clarify that we needed to create an ExpressServer that would be hosted on a cloud server, which would communicate with a front-end object deployed on client web browsers. Both of these diagrams can be seen in the "Design Activities" section of this document.

Updates For Sprint 1 (See Design section below):

- Created class diagram
- Created deployment diagram
- Listed out system design goals

Planning and Estimation:

Projects / se3350-group 18

Backlog

MH

CM

Epic ▾

Insights

SG1 Sprint 1 24 Jan – 4 Feb (2 issues)

0 0 0 Complete sprint ▾

Complete Level 1

SG1-24 Create Level 1 RELEASE 1 MVP

DONE ▾

SG1-15 Animations RELEASE 1 MVP

DONE ▾

CM

+ Create issue

SG1 Sprint 2 7 Feb – 18 Feb (2 issues)

0 0 0 Start sprint ▾

SG1-25 Level 2 RELEASE 1 MVP

TO DO ▾

SG1-26 Level 3 RELEASE 1 MVP

TO DO ▾

+ Create issue

Projects / se3350-group 18

SG1 Sprint 1

Complete Level 1



MH

CM

Epic ▾

TO DO 2 ISSUES

Create Level 1

RELEASE 1 MVP

SG1-24

Animations

RELEASE 1 MVP

SG1-15

CM

IN PROGRESS

DONE ✓

+

Development

During sprint 1, we met multiple times to code synchronously using VScode LiveShare. This ensured that everyone was definitely contributing, and helped us all stay on the same page. We found this to be extremely beneficial during the early stages of this project, because during this sprint, we created the backbone for all major system structures and interactions. In the future, we will still use LiveShare for certain tasks that are important to have all-hands-on-deck for, but now that we all have a thorough understanding of the system, it should be easier to split up tasks and code certain components asynchronously.

Testing

During this sprint, we outlined test cases that needed to be satisfied for level 1 of the project. These were useful to test each component that we worked on individually, and ensured that our final product met all of the requirements we outlined in the “Requirements Elaboration” section. Our most updated test cases are outlined in the “Test Suite” section of this document.

Updates For Sprint 1:

- Conducted test cases for Level 1 - these can be seen in the ‘Test Suites’ section of the document

Integration

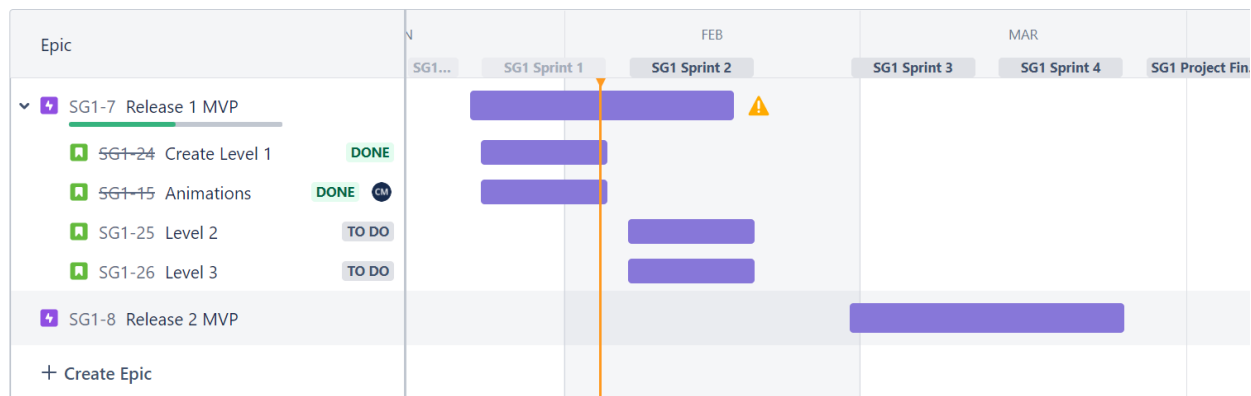
***NOTE:** We used VSCode LiveShare and all coded on the same document, so the git commits aren't representative of our collective contributions - we all coded our parts as given.



Documentation

The major documentation we developed during this sprint was the use cases and use case diagrams. These use cases were important to help us divide up tasks in Jira, and helped us ensure that our system was satisfying all necessary requirements. As mentioned above, we also created design activity diagrams (class diagram and deployment diagram) as well as documented our test suite.

Approval



- Nothing was presented to take a screenshot of when sprint was completed
- Screenshot above shows sprint 1 greyed out

Review & Retrospective

Video demo: <https://youtu.be/A3DFssA89W0>

What went well?

We worked in peer development sessions prior to the lab on Thursday so we were all able to get on the same page and get more synchronous working time together. This involved using VS code's live share function and hosting a Zoom session at the same time. Ultimately, this was very beneficial as it gave everyone equal understanding of how our project code would be structured, and it also allowed us to support one another in devising how to get our Sprint 1 functional. The team has decided to continue using peer development sessions when working on difficult portions of the project as it is beneficial when everyone's brain power is used together.

What went wrong?

As mentioned above, we initially worked in peer development sessions as we wanted to be on the same page when establishing the foundations for Sprint 1. However, we ran into an issue when trying to pull and push on Github. Some individuals had an older version of the project code from Github, while some other members were not able to push their changes successfully. Although we were able to all work on the same version in the end, we would like to avoid this in the future. Some strategies we thought of included making announcements in the group chat when a new push is being made (and to specify

which branch it is being pushed to) and using GitHub Desktop as we found it was the most reliable way to synchronize updates.

The team ran into some difficulty when downloading JSAV and using Grunt to run our project. For many members, it would indicate that Grunt was not downloaded in the local file so it could not run properly, even though it was downloaded in the VS code terminal. In the end, we resolved it by having a member who's JSAV and Grunt was running properly, upload their JSAV file to a Google Drive folder we all shared. After that, team members who ran into the issue would simply download the file off Google Drive and move it into the project's code folder manually. This solution worked for all members and we were able to continue our development without issues. Although this may be an isolated issue, we can use this strategy again if we run into a similar issue of certain extensions and tools not working for some members.

Requirements Activities & Models

Describe and model your requirements elicitation and analysis here like:

Functional requirements

- A user (either player or admin) should be able to enter their login credentials and be navigated to the appropriate page.
- A user should be able to logout of the game by either selecting the logout option or closing the browser.
- A user should be automatically logged out and returned to the home page if they are inactive for more than 5 minutes.
- When the player makes a move in the game, the game visually displays whether the move was correct or not.
- When the player makes a move in the game, the game outputs an auditory cue to indicate whether the move was correct or not.
- The game should display an active timer to show how long a player spends on each level in real-time.
- The game should record how long each player spends on a level attempt.
- The game should record what moves were made on each level attempt.
- The game should automatically generate player analytics to be displayed to admin users.
- A menu should display when a player has made 3 mistakes on a level attempt, offering the following options: restart the level, return to the level menu, attempt the same level with a different algorithm, quit the game.
- The player should be able to select which sorting algorithm they wish to practice.
- For each level, the quantity of numbers that are generated in the set should be equal to or greater than the previous level.
 - The 1st level should generate a set of 10 numbers.
 - The 2nd level should generate a set of 10 numbers.
 - The 3rd level should generate a set of 10 numbers.
 - The 4th level should generate a set of 20 numbers.
 - The 5th level should generate a set of 50 numbers.
- For each level, the number set should be randomly generated.
 - The 1st level should generate numbers within the range of 1-20 inclusive.

- The 2nd level should generate numbers within the range of 1-20 inclusive.
- The 3rd level should generate numbers within the range of 1-20 inclusive.
- The 4th level should generate numbers within the range of 1-50 inclusive.
- The 5th level should generate numbers within the range of 1-100 inclusive.
- The 1st level of the game should display an animation that walks through the correct steps for solving the number set using the appropriate algorithm.
- On all levels after the 1st, the player should be able to make moves to attempt to solve the number set in accordance with the selected algorithm.
 - The 2nd level of the game should provide text instructions for the steps that need to be made, but allow the player to move the numbers.
 - All levels after the 2nd require the player to make every decision.
- On the Custom level, the player should be able to select the size of the number set they want to attempt to solve.
- On the Custom level, the player should be able to input the numbers they want in the number set they want to attempt to solve.
- A player should be able to replay completed levels at any time.
- When a level is completed, the player should be presented with a menu that presents the following options: restart the level, return to the level menu, attempt the same level with a different algorithm, quit the game.
- At any point in a level, the player should be able to access a menu that presents the following options: restart the level, return to the level menu, attempt the same level with a different algorithm, quit the game.

Non-functional requirements

- The game should be 100% accurate in identifying if a step is correct or incorrect when doing merge sort
- The game will be implemented using Express, JavaScript and HTML
- The game should be available 24/7, excluding required maintenance periods.
- When navigating to a page, the load time should be quick (less than 2 seconds) and the transition should be smooth in appearance to the user.
- The game should inform the user if they are attempting to fill a field with the incorrect data type.
- Multiple users should be able to access the game simultaneously, with no user limit and no significant increase in loading time.
- There should be a 98% chance that the system will not experience critical failure.
- The system should support a variety of web browsers including Google Chrome, Mozilla Firefox, and Microsoft Edge.
- Data should be stored in such a way that users can only access that data that they are authorized to view (ie. professors should only be able to view the data for their assigned students).

Use-Cases

*A User may be either an Admin or a Player

Use Case Name	login
---------------	-------

Participating Actors	Initiated by the User
Flow of Events	<ol style="list-style-type: none"> 1. The User arrives at the homepage. 2. The User indicates if they are a Player or an Admin using the available toggle. 3. The User enters their credentials into the available text fields. 4. The User clicks the “Login” button. 5. If the credentials are valid, the site navigates to the appropriate page. If the credentials are invalid, this is communicated to the User through text and they remain on the homepage.
Entry Condition	The User is on the game site.
Exit Condition	<p>The User clicks the “Login” button to continue.</p> <p>The User closes the browser to exit the site completely.</p>
Quality Requirements	The page should be loaded no longer than 10 seconds after the user presses the login button.

Use Case Name	viewPlayerAnalytics
Participating Actors	Initiated by the Admin
Flow of Events	<ol style="list-style-type: none"> 1. The Admin clicks the “View Analytics” button. 2. The site navigates to the page that displays all current player analytics.
Entry Condition	The Admin is logged in with admin credentials.
Exit Condition	<p>The Admin clicks the “Logout” button to return to the homepage.</p> <p>The Admin closes the browser to exit the site completely.</p>
Quality Requirements	Analytics page should open no longer than 1 second after being redirected.

Use Case Name	selectAlgorithm
Participating Actors	Initiated by the Player
Flow of Events	<ol style="list-style-type: none"> 1. The Player selects an algorithm from the menu.

	2. The site navigates to the Level Select page.
Entry Condition	The Player is logged in with player credentials.
Exit Condition	<p>The Player selects an algorithm to continue.</p> <p>The Player clicks the “Logout” button to return to the homepage.</p> <p>The Player closes the browser to exit the site completely.</p>
Quality Requirements	Pages should load no longer than 1 second after being redirected by the user.

Use Case Name	selectLevel
Participating Actors	Initiated by the Player
Flow of Events	<ol style="list-style-type: none"> 1. The site displays all levels, indicating which levels are locked or unlocked for the Player. 2. The Player selects a level that is available to them. 3. The site navigates to the appropriate level page.
Entry Condition	The Player has completed selectAlgorithm.
Exit Condition	<p>The Player selects a level to continue.</p> <p>The Player clicks the “Logout” button to return to the homepage.</p> <p>The Player closes the browser to exit the site completely.</p>
Quality Requirements	Home Screen should be displayed no longer than 10 seconds after the user presses the logout button.

Use Case Name	playLevel1
Participating Actors	Initiated by the Player
Flow of Events	<ol style="list-style-type: none"> 1. The site generates and displays the unsorted number set. 2. The Player clicks the “Next” button to progress the animation. 3. The site displays the next appropriate action in sorting the number set according to the

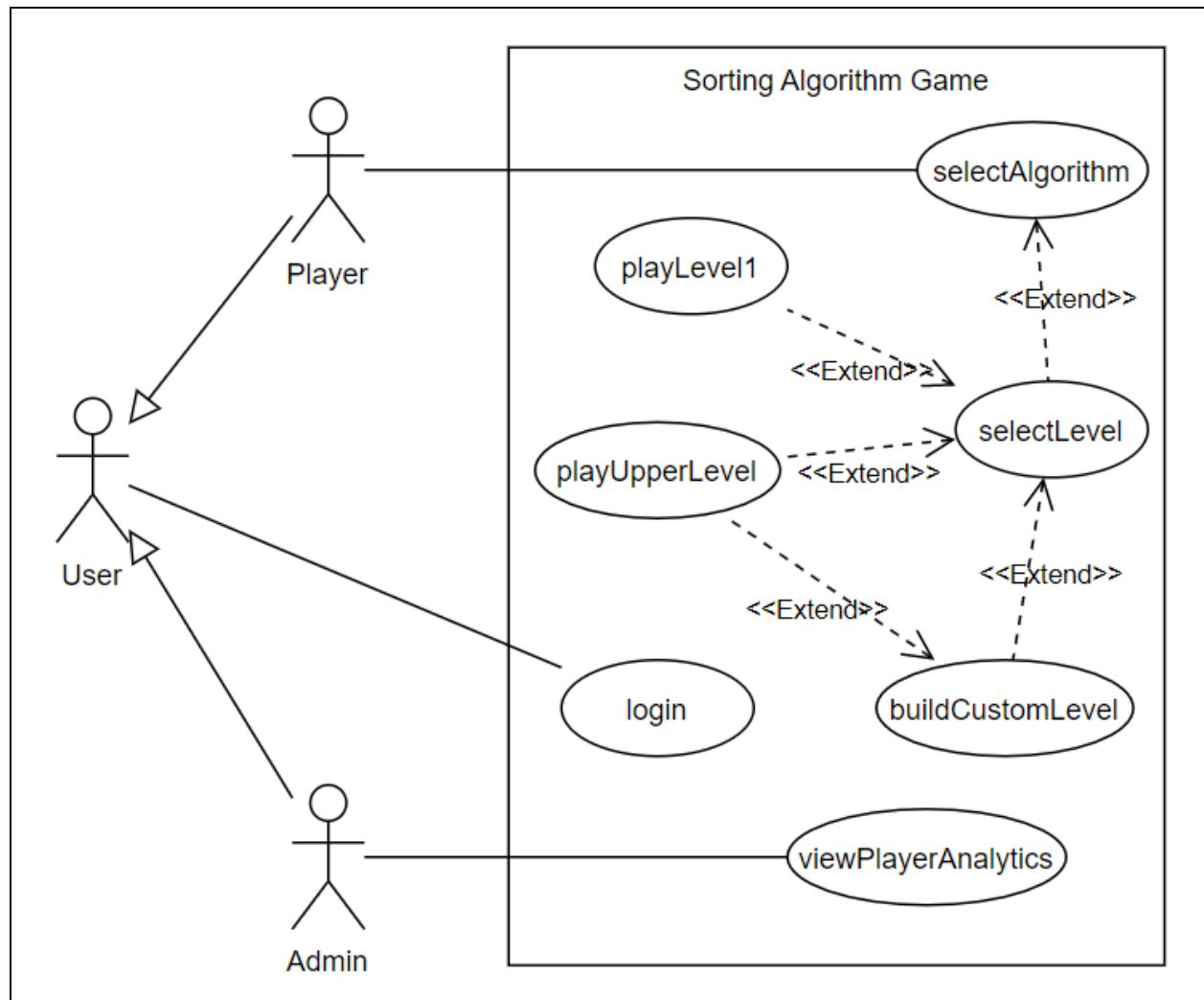
	<p>selected algorithm, with text instructions.</p> <ol style="list-style-type: none"> Repeat steps 2 and 3 until the number set is sorted. An exit menu is displayed. The Player selects an option from the menu. The site navigates to the appropriate page.
Entry Condition	The Player has completed selectLevel.
Exit Condition	<p>The Player selects an option from the exit menu and is navigated away from the completed level page.</p> <p>The Player clicks the “Logout” button to return to the homepage.</p> <p>The Player closes the browser to exit the site completely.</p>
Quality Requirements	Each step should take no longer than 5 seconds to load with the press of the next button.

Use Case Name	playUpperLevel
Participating Actors	Initiated by the Player
Flow of Events	<ol style="list-style-type: none"> The site generates and displays the unsorted number set. The Player makes a move. The site communicates whether or not the move was correct through both a visual and an auditory cue. Repeat steps 2 and 3 until the number set is sorted or 3 mistakes have been made. An exit menu is displayed. The Player selects an option from the menu. The site navigates to the appropriate page.
Entry Condition	<p>The Player has completed selectLevel.</p> <p>The Player has completed the prior level, and consequently has access to this one.</p>
Exit Condition	<p>The Player selects an option from the exit menu and is navigated away from the level page.</p> <p>The Player clicks the “Logout” button to return to the homepage.</p> <p>The Player closes the browser to exit the site completely.</p>

Quality Requirements	User feedback should take no longer than 1 second to display to the user.
----------------------	---

Use Case Name	buildCustomLevel
Participating Actors	Initiated by the Player
Flow of Events	<ol style="list-style-type: none"> 1. The Player indicates the desired size of the number set and clicks the “Randomly Generate Set” button, or enters a number set and clicks the “Continue” button. 2. The site navigates to the level page.
Entry Condition	<p>The Player has completed selectLevel.</p> <p>The Player has completed the prior level, and consequently has access to this one.</p>
Exit Condition	<p>The Player selects an option from the exit menu and is navigated away from the build page.</p> <p>The Player clicks the “Logout” button to return to the homepage.</p> <p>The Player closes the browser to exit the site completely.</p>
Quality Requirements	Custom level should take no longer than 15 seconds to load after criteria is entered by the user.

Use case diagrams



User interviews

Summary of Conclusions:

As part of testing we got three students, not involved in the creation of the program, to run through the first level. Initial feedback was positive. These interviews also provided the team with suggestions to help make the concepts of the merge sort easier for the user to follow in the tutorial. Suggestions included adding arrows to visually show the arrays splitting down and merging, adding color to highlight the current number being referenced, and adding the index numbers referred to in the instructions on the display. These recommendations will be examined and addressed by the team in the second sprint.

Sample Questions Asked:

Q: Are you familiar with the concept of sorting algorithms?

Student 1: Yes kind of, we learnt a bit about them in first year I think but it hasn't come up since.

Student 2: Yes, we took an algorithms course in second year (software engineering).

Student 3: No, I don't think so.

Q: Did the tutorial help you learn the concept of merge sort?

Student 1: Yes, seeing it breakdown physically helps get the message across.

Student 2: Yes I think it makes a lot of sense, the visuals definitely help make it easier to understand.

Student 3: Theoretically yes, the process is clear but I just don't know the purpose of it overall.

Q: Is there anything you would change about the design that would further help the user understand the concept?

Student 1: If you added some arrows to show where the individual numbers were going it would probably make it a bit more clearer. Adding some color to show each step could also help.

Student 2: I think in a written instruction you mentioned index numbers but those aren't in the picture, that may confuse some people who aren't as familiar with sorting algorithms.

Student 3: Maybe include in the instructions at the start an explanation of why someone would use a merge sort? Since I haven't been taught it in any classes so far I'm not sure how this would be useful or who would use it.

Customer interviews

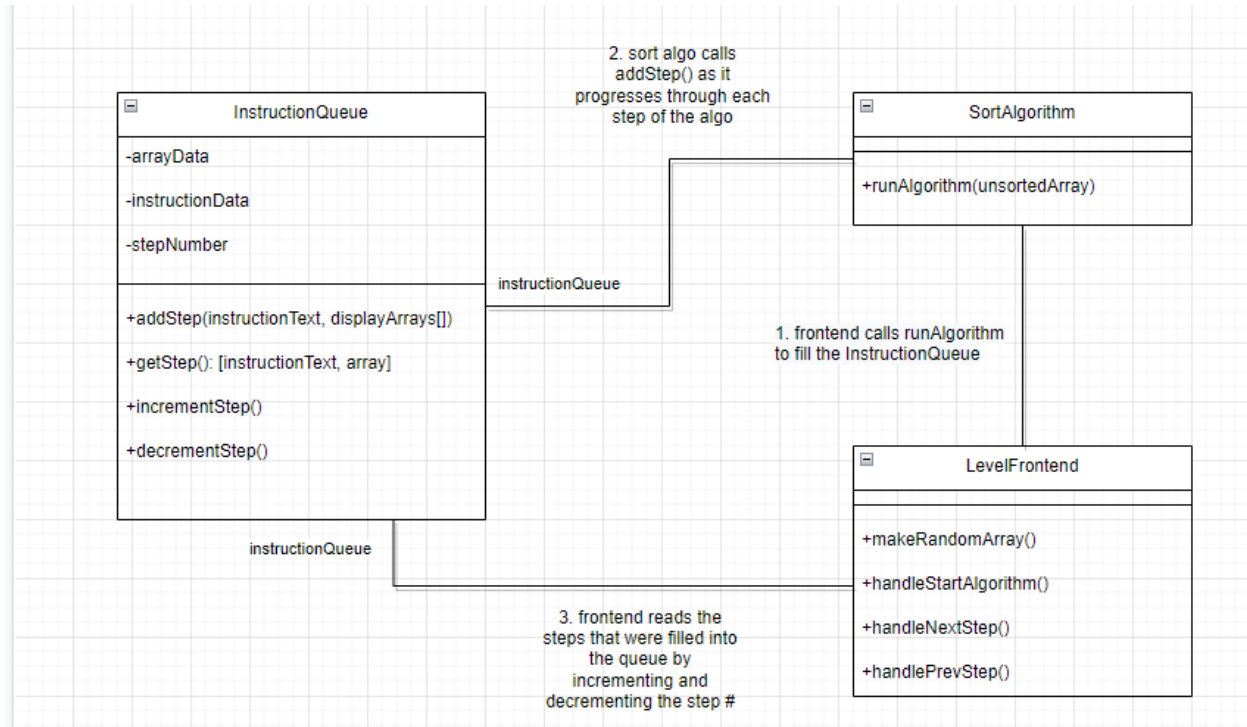
Note: an interview with the Professor will be included for Sprint 2.

Surveys

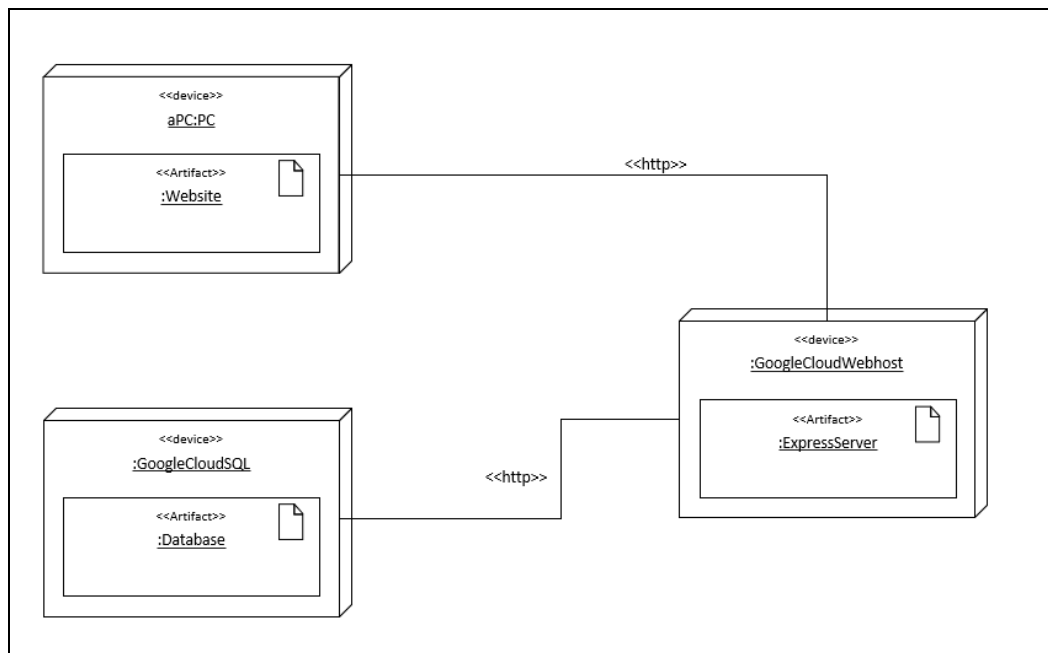
Note: surveys with the potential users will be conducted in release 2.

Design Activities & Models

Class Diagram



Deployment Diagram



Design goals

Performance

- Response time is an important factor for this project. In order to create an enjoyable experience for users, the site should react to user requests and actions using the minimum required time possible. Specific times for certain actions are specified under the Nonfunctional Requirements section of this document.

Dependability

- Robustness is a critical factor for this development, as the game centers around the trial and error method of learning. The game should be able to receive incorrect inputs from the user without failure, and inform the user that the input was incorrect.
- Reliability should be considered, as the game should function as expected for the user to ensure that what they are learning is correct and that they are learning it efficiently.
- Availability should also be considered, as this is intended to be a learning tool for students. To ensure that students are able to learn at their own pace, the game should be available at all times for them to practice.
- Security is important on the admin side of the game, as analytics are being collected from players. All recorded data should be stored securely, and only allow users to access the data that they have permission for (ex. Professors should be able to access all necessary information regarding their students).

Cost

- As this is a class project, the cost to develop the system only depends on the time and effort required by the developers, and for this reason it is not a significant consideration.
- Deployment costs will be covered by the Google Cloud Platform credits that were provided through the course.

Maintenance

- Extensibility is critical for our project, as the game needs to allow the addition of new algorithms, besides merge sort, in the future.
- The system should allow for easy modification and changes to be made across sprints to accommodate user feedback.

End User Criteria

- New users should be able to easily navigate the program and its various pages without consultation of a user manual.

Test Suites

Story	Test Case	Input	Expected Output	Actual Output	result
Start Button	TC1	Click	Initial Array	Initial Array	Successful
Next Button	TC2	Click	Next Step and Instructions	Next Step and Instructions	Successful
Merge Sort Algorithm	TC3.1	Full Array	Sub-divided array	Sub-divided array	Successful
	TC3.2	Single Numbers	Sorted Array	Sorted Array	Successful
	TC3.3	Sorted Arrays	Sorted Array	Sorted Array	Successful
Random Number Generator	TC4.1	N/A	Numbers are random	Numbers are random	Successful
	T4.2	N/A	No repeated numbers	No repeated numbers	Successful