

# Software Engineering- Sprint 4

## Documentation - Team 09

Arnav Singh, Petri Gjoni

April 27, 2021

---

## Contents

<b>1</b>	<b>Implemented Features/Functionality as per User/Requirements in this Sprint</b>	<b>2</b>
1.1	Implemented User Authentication . . . . .	2
(a)	Player Login/Sign-Up . . . . .	2
(b)	Instructor Login/Sign-Up . . . . .	2
1.2	Player . . . . .	4
(a)	Can Join Games/ See Joined Games . . . . .	4
(b)	Leave a Game . . . . .	6
(c)	Enter Game . . . . .	7
1.3	Instructor . . . . .	8
(a)	Can Create Games/See Created Games . . . . .	8
(b)	Instructor View Details . . . . .	10
(c)	Instructor Can delete Games . . . . .	10
(d)	Instructor Can Modify Games . . . . .	11
1.4	Improved the Database Design and Rewrote SQL Queries/Statements	11
<b>2</b>	<b>Testing- Python Unit testing</b>	<b>12</b>
2.1	Testing Python API- Unit testing . . . . .	12

# 1 Implemented Features/Functionality as per User/Requirements in this Sprint

## 1.1 Implemented User Authentication

### (a) Player Login/Sign-Up

The login and sign-up respective of instructor and player has been implemented. So a Player can login and he will see his own views. A pop-up is implemented to for UI functionality to indicate if logged in or Logged out.

### (b) Instructor Login/Sign-Up

The login/register page is the same for the instructor, it is just that if you click on instructor checkbox, it will register you as an instructor. In the database the Model is stored as a Boolean variable meaning if the button is checked box it will also insert into the Instructor table. We made use of session variables upon login to identify which instructor has logged in and to pass the id to the different pages.

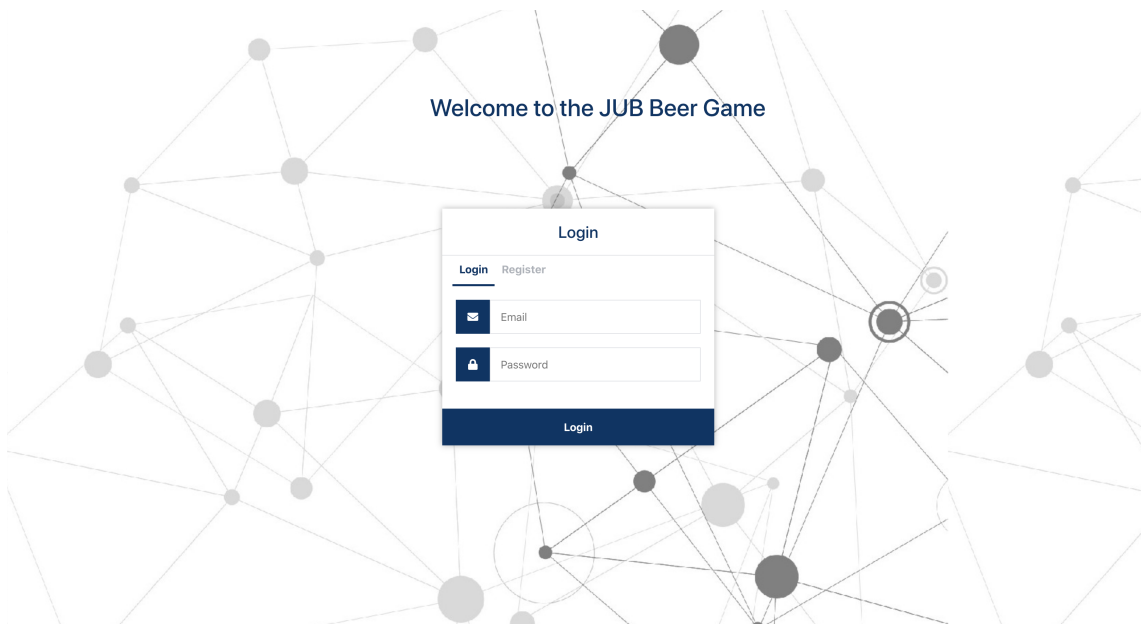


Figure 1: Home page for the Login



Figure 2: Home page for the REGISTER

## 1.2 Player

### (a) Can Join Games/ See Joined Games

The screenshot displays the 'Beer Game' Player Home Page. At the top, a dark blue navigation bar contains the text 'Beer Game' on the left and links for 'Home', 'Profile', and 'Logout' on the right. Below the navigation bar, the page title 'Player Home Page' is centered. A white box contains the message 'Welcome back, !'. Below this, there are two links: 'View Joined Games' and 'Join a Game'. A white box with a green border contains the message 'Game joined succesfully' followed by a close button 'x'. Below this, a dark blue box titled 'Join a Game' contains two input fields: 'Game ID' and 'Role'. The 'Game ID' field has the placeholder text 'Game ID' and the 'Role' field has the placeholder text 'Consumer'. A yellow 'Join' button is located at the bottom left of the 'Join a Game' box.

Figure 3: Join games page

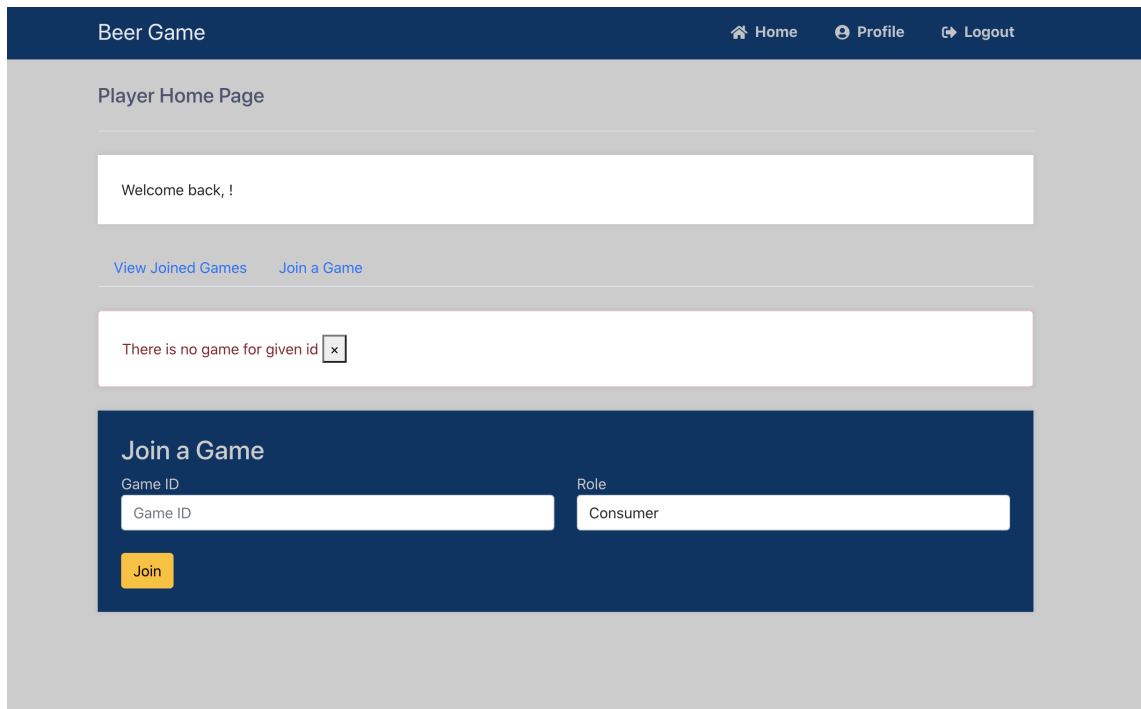


Figure 4: Feedback Error

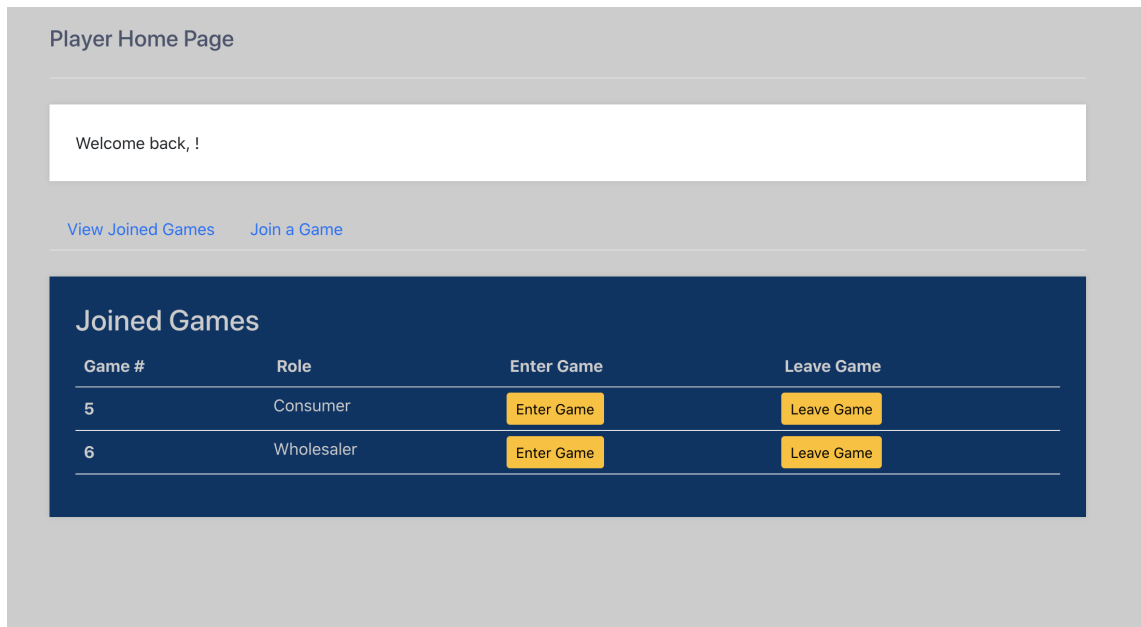


Figure 5: See the joined Games

**(b) Leave a Game**

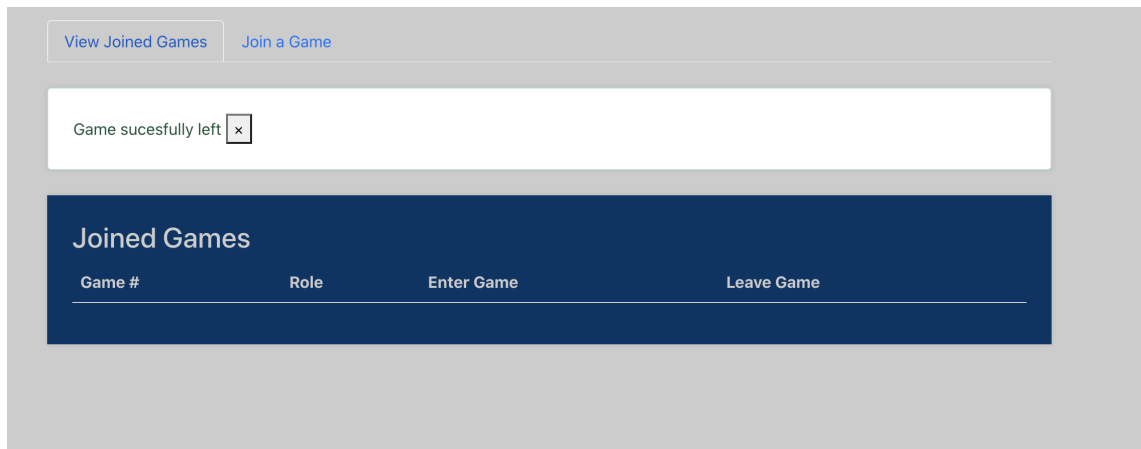


Figure 6: Leave a Game

(c) Enter Game

For Week 1

Beginning Inventory : 0

Demand from Distributor : 1

Incoming Shipment : 1

On Backorder : 4

On Total Requirement : 1

Total Available : 5

Units Shipped to Distributor this week : 4

Ending inventory 2

Enter the number of units to be purchased from Brewery :

Place Order

FACTORY INFORMATION FOR THE LAST TEN WEEKS

Week	Inv/Bk	Demand	Incom. ship	Outg. ship	Order Placed	Current cost
1	0	3	0	1	2	2

Status of other Supply Chain Channel Members Game {gameNo}

This page will update automatically after a period of time.

When all players have completed the order for the current week,  
the player will automatically receive a link to procede to next week.

Week 1

Factory: {factoryOrder ? "Has ordered" : "Has not ordered"}

Distributor: {distributorOrder ? "Has ordered" : "Has not ordered"}

Inventory and Order Status plots For Factory

Demand plot

Order plot

Inv/Backorderplot

Plot all

Supply Chain Settings for Factory:

Holding cost : {holdingCost}

Backorder cost : {backorderCost}

Downstream player : {downstream}

Upstream player : {upstream}

Shipping Dealy : {shipDelay1} wks ((current) -> {downstream}) {shipDelay2} wks ({upstream} -> {current})

Information Dealy : {infoDelay1} wks ({downstream} -> {current}) {infoDelay2} wks ({current} -> {upstream})

Figure 7: Enter a Game

## 1.3 Instructor

### (a) Can Create Games/See Created Games

Beer Game

[Home](#) [Profile](#) [Logout](#)

Instructor Home Page

Welcome back Instructor, !

[Create Game](#) [View Created Games](#) [View Details](#) [Create Demand Pattern](#)

Game sucessfully created! ✕

Create Game

Session length

Session length

Holding Cost

holdingcost

roundscompleted

Roundscompleted

Submit

Backlogcost

backlogcost

startinginventory

startinginventory

☐ Wholesaler Present

☐ Distributor Present

☐ Enable Inforsharing

Figure 8: Create Games with popup feedback



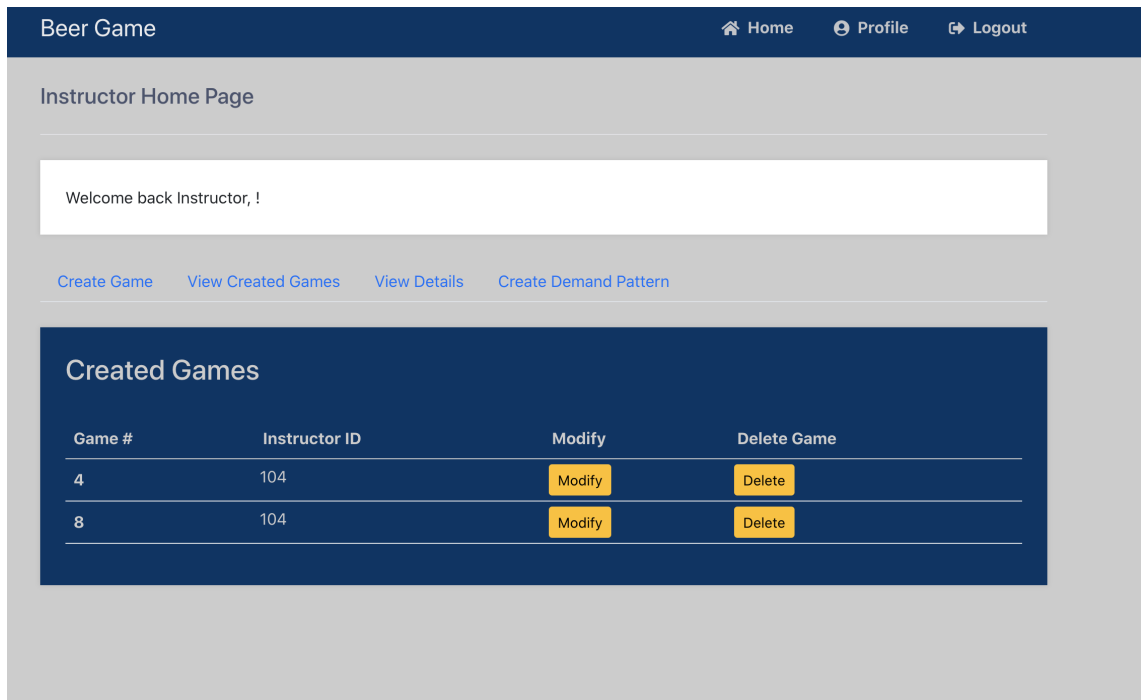


Figure 9: List of created games

## (b) Instructor View Details

The screenshot shows the 'Beer Game' Instructor Home Page. At the top, there is a dark blue header with 'Beer Game' on the left and navigation links 'Home', 'Profile', and 'Logout' on the right. Below the header, the page title 'Instructor Home Page' is displayed. A white message box says 'Welcome back Instructor, !'. Below this, there are four navigation links: 'Create Game', 'View Created Games', 'View Details', and 'Create Demand Pattern'. The main content area is titled 'View Game Details' and contains a table with the following data:

Game #	Game Length	Rounds Completed	Holding Cost	Backlog Cost	Starting inventory	Info sharing?	Distributor Present?	Wholesaler Present?
8	1	1	1.0	1.0	1	0	0	0
9	1	1	1.0	1.0	1	0	0	0

Figure 10: List of created Games detailed

## (c) Instructor Can delete Games

The screenshot shows the 'Beer Game' Instructor Home Page after a game has been deleted. At the top, there is a dark blue header with 'Beer Game' on the left and navigation links 'Home', 'Profile', and 'Logout' on the right. Below the header, the page title 'Instructor Home Page' is displayed. A white message box says 'Game sucesfully deleted' with a close button. Below this, there are four navigation links: 'Create Game', 'View Created Games', 'View Details', and 'Create Demand Pattern'. The main content area is titled 'Created Games' and contains a table with the following data:

Game #	Instructor ID	Modify	Delete Game
--------	---------------	--------	-------------

Figure 11: Delete specific game

(d) Instructor Can Modify Games

The screenshot shows a web application titled 'Beer Game' with a dark blue header. The header contains navigation links: Home, Profile, and Logout. The main content area is divided into two sections. The first section, 'Current Game Parameters', displays a table with the following data:

Game #	Game Length	Rounds Completed	Holding Cost	Backlog Cost	Starting inventory	Info sharing?	Distributor Present?	Wholesaler Present?
4	1	1	1.0	1.0	1	0	0	0

The second section, 'Modify Game Parameters', contains input fields for 'Session length', 'Holding Cost', 'Backlogcost', and 'startinginventory'. There are also checkboxes for 'Wholesaler Present', 'Distributor Present', and 'Enable Inforsharing'. A yellow 'Submit' button is located at the bottom left of the form.

Figure 12: Modify Specific Game

## 1.4 Improved the Database Design and Rewrote SQL Queries/Statements

To make sure that the ids will be unique we added SQL Auto increments into the SQLschema.sql file such that each entry is unique and is consistent with the database. We also changed the ids from a VARCHAR datatype to INT which will be auto-incremented each time a new entry has been added.

Furthermore we added a role property to the Plays\_In table to store the role of the player which was previously in the player table. This was a problem because before because the implementation limited us to only have one player play one game with one role. However now , each player can play many games with different roles.

## 2 Testing- Python Unit testing

### 2.1 Testing Python API- Unit testing

We implemented unit-testing of the new components, to test python endpoints, you can simply run. This was previously different because of the different endpoints.

```
python3 tests.py
```