Individual QA Cinema Project portfolio
Dan Harrison

Introduction

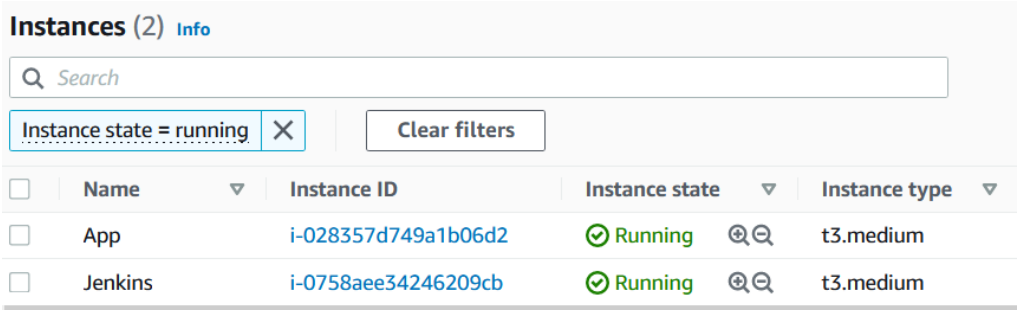
Throughout this portfolio I will document and discuss any of my individual additions to our QA Cinema Group Project, completed along-side Andrew Pimlott, Shai Lynch, and Onur Hassan. The project given required us to work as an agile team, to build a full stack application for a new upcoming cinema. This included creating a SpringBoot application, AWS database and hosting services, and an attractive and functional front-end.

Day One

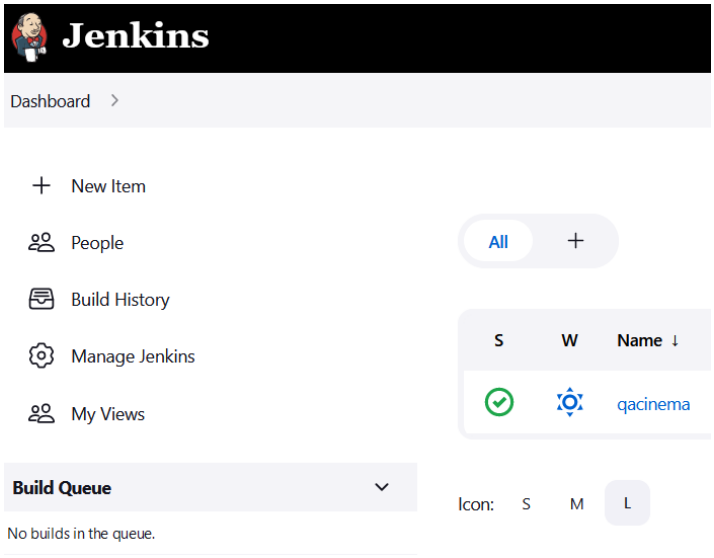
On day one of the project, it was decided that I would take the role of Scrum Master. This meant I would be conducting Sprint and Scrum meetings with daily stand-ups and ensuring the project was flowing well and we stayed on track with our deployment date.

After this, we moved on to setting up our project management software, I tried to make sure that the project was broken down into tasks as small as possible, so nobody ever felt overwhelmed with their given task and allowed me to keep the group organised throughout the entire project.

I then shifted my time to setting up and configuring the Jenkins pipeline, this including creating VPC's EC2's and an RDS instance, the EC2's needed configured, with things like Java, Maven, and Git installed, I then had to set up SSH keys on both EC2's to enable the functionality I required. I had planned to have one EC2 run the Jenkins pipeline and the other would be the machine hosting the product.



Configuring this took a few hours. I tested it using a past spring boot project so I could see that the jenkinsfile was set up correctly and to my knowledge would work just as well when we connected it to our new project repository. This took me to the end of the first day.



Day Two

Day two was mainly spent completing documentation for the project. We worked collaboratively for this, I oversaw the creating the UML documentation with the groups input, making sure work diligently and to not miss anything important for the initial imagined design of our entities.

After this I moved onto writing test cases for the entity models we were using, half of the group members were not confident in writing tests so I decided that we would use peer programming for this as a way to be more productive, and also assist in the learning of the less confident team members. I ended up working with Onur, and as I wrote tests, I would walk through my thought process and explain what was happening at each stage. Having another team member there with me did assist whenever I had made a syntax error or missed something which they could see. An example of the tests can be seen here:

```
public class BookingTest {

	static Booking booking;

	@BeforeAll
	public static void createBooking() {
		System.out.println("Creating Booking");
		booking = new Booking(1L, 1L, "31-08-22", 1L);
	}

	@Test
	public void testEquals() {
		EqualsVerifier.simple().forClass(Booking.class)
				.withPrefabValues(Booking.class, new Booking(1L, "31-08-22", 1L), new Booking(2L, "30-08-22", 2L))
				.verify();
	}

	@Test
	public void testToString() {
		String expecting = "Appointment [Id=" + booking.getId() + ", viewingID=" + booking.getViewingID()
				+ ", bookingDate=" + booking.getBookingDate() + ", customerID=" + booking.getCustomerID() + "]";
		assertEquals(expecting, booking.toString());
	}
```

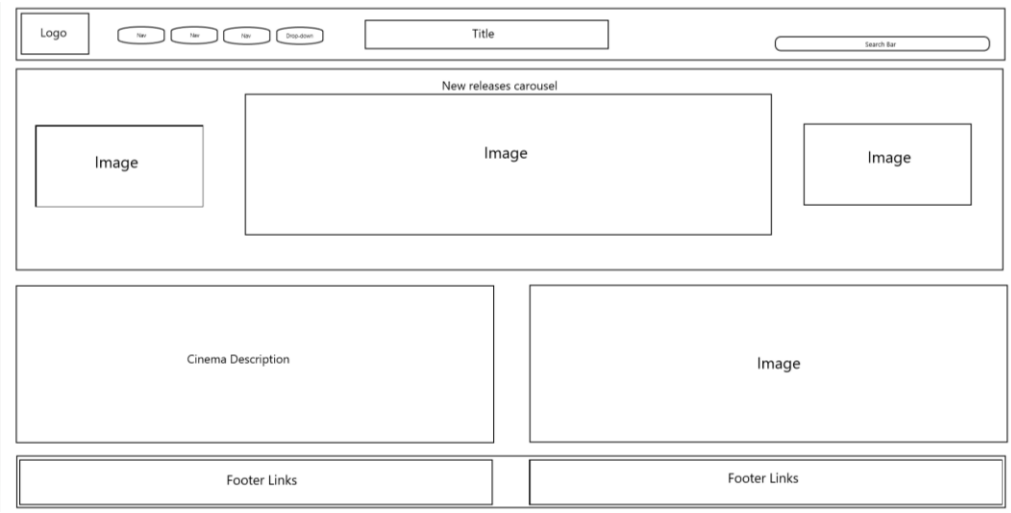
Day Three

Day three consisted of more testing, however as entities were done, I moved onto controller testing. At the beginning of the day in the stand-up meeting I put forward the idea of having me and Andrew work on testing, while Onur and Shai moved on to designing the front-end and looking at a colour scheme/theme for the overall site to keep it uniform across all pages. The Team was happy with my plan, so we cracked on with that.

```java
@SpringBootTest
public class CommentsControllerUnitTest {

	@Autowired
	private CommentsController controller;

	@MockBean
	private CommentsService service;

	@MockBean
	private CommentsRepo repo;

	@Test
	public void createNewCommentTest() {
		Comments validComment = new Comments(1L, 1L, "content", "time", "author");
		Mockito.when(this.service.addComment(validComment)).thenReturn(validComment);
		ResponseEntity<Comments> response = new ResponseEntity<Comments>(validComment, HttpStatus.CREATED);
		assertThat(response).isEqualTo(controller.newComment(validComment));
		Mockito.verify(this.service, Mockito.times(1)).addComment(validComment);
	}
```

Just before we headed to lunch, I pulled down all of Andrew's tests from the GitHub repo and ran the entire project as Junit Coverage and we ended up sitting very comfortable at close to 98% coverage for the testing after the initial three days.

After lunch I pulled the team together so we could go over design ideas for the site and create some wireframes for each of the pages. Onur was given the task of drawing out the wireframes while we discussed how the pages should look, and which elements should be included on each page. Once we had a full set of wireframes for the site, I made notes all of all the pages and gave the team member a choice of which pages they wanted to start designing independently. The team all chose where they would like to begin, Andy started with the Classifications page, Shai started with the Contacts page, Onur started with the Search page, and I chose to give myself the Discussion Board. The rest of day three was spent working on these pages. Here is an example of the wireframes we created.



Day Four

To begin day four, I again held our daily morning stand-up meeting to see where each team member had gotten up to, what they would be focusing on for the day, and to tackle any issues anybody needed assistance with. After this was back to working on the Discussion board page.

The discussion board took a few complicated turns which I did not expect, I originally wanted to tackle the page using local storage before implementing a database, as I had not used local storage

manipulation before this included quite a bit of reading and understanding exactly how it worked. However, towards the end of day four it was coming together nicely, and I had a styled page with a few topic threads and a way to add comments and store them in local storage while printing them to the screen.

Although I had made good progress on the page, there was still more functionality I needed to incorporate and therefore would be continuing this on day five.

Day Five

When day five began as usual I held the daily stand-up and we discussed how our progress was going and did a quick round-up to see how far along we were on the project, I think doing this help keep the team's motivation up and they all seemed to get excited when they heard how close we were getting to the end of the project.

I was going to spend the day creating a table for comments to be stored so they could be seen by everybody using the discussion board in real time, to tackle this I had to create a whole new entity in the back-end API so most of the day was spent doing that. I created a Model, Controller, Service and Repo for the new Comments entity and wrote test cases for all of these, hitting 100% test coverage for the new entity. Here you can see the Entity attributes for the new model.

```
@Entity
public class Comments {

        @Id
        @GeneratedValue(strategy = GenerationType.IDENTITY)
        private Long commentId;

        @Column
        private Long topicId;
        @Column
        private String commentContent;
        @Column
        private String commentCreateTime;
        @Column
        private String commentAuthor;
```

From past projects I knew I wanted to ensure testing was kept at a high level throughout development so this was a personal priority for me. Now that the entity was created, I spent the last few hours of the day checking database pathing and writing some JS function to allow creation and reading of comments from the new database table.

As it was the end of the week, I stopped work a bit earlier to pull the team into a meeting so we could discuss how it was going and start figuring out a plan for the week to come, Andy and Shai were making excellent progress, each completing multiple pages and the search page was coming along nicely. I wanted to emphasise how well this was going to keep the team's motivation up.

Day Six

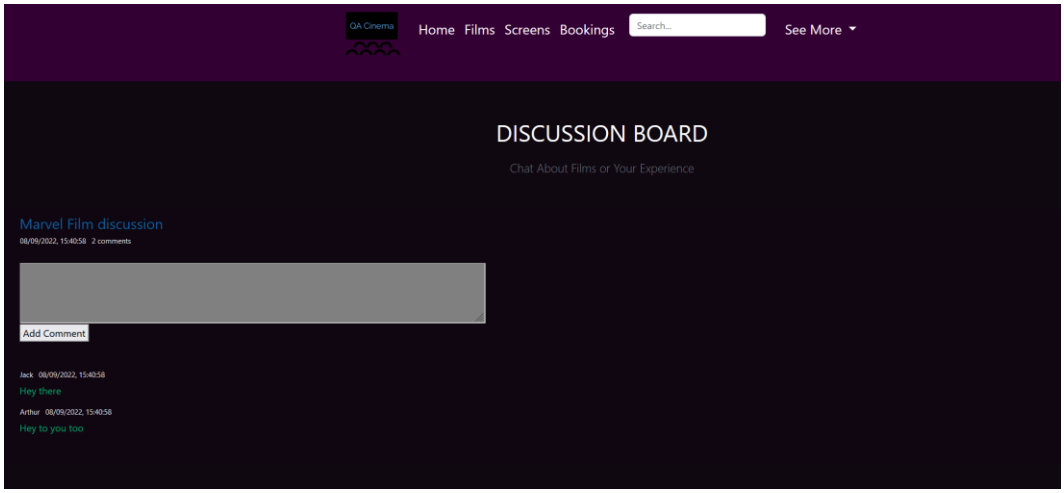
A new week, and the final push for the project had begun. As usual I help the daily morning stand-up meeting and we all discussed our working plans for the day. I went back onto the discussion board and was at a point where the comments were comfortable stored within the database, but the read

functionality seemed very buggy, I requested fresh eyes from the team to see if it was something silly, I was missing, however after an hour or so of troubleshooting nobody could come up with anything and all signs pointed to it should be working. There was some more trouble shooting required within the team on this day, so I spent some time working with others fixing issues, some of which required back-end tweaks. These issues did not last long and in no time the day was ending. Here you can see the database saving newly added comments from the discussion board.

```
mysql> select * from comments;
+------------+-----------------+-----------------------------------+-----------
----------+----------+
| comment_id | comment_author  | comment_content                   | comment_cr
eate_time | topic_id |
+------------+-----------------+-----------------------------------+-----------
----------+----------+
|          1 | Anonymouse User | hello                             | 8-9-2022 1
6:32      |        1 |
|          2 | Anonymouse User | add comment                       | 8-9-2022 1
6:41      |        1 |
|          3 | Anonymouse User | Can I talk to your manager please? | 8-9-2022 1
8:57      |        1 |
+------------+-----------------+-----------------------------------+-----------
----------+----------+
3 rows in set (0.00 sec)
```

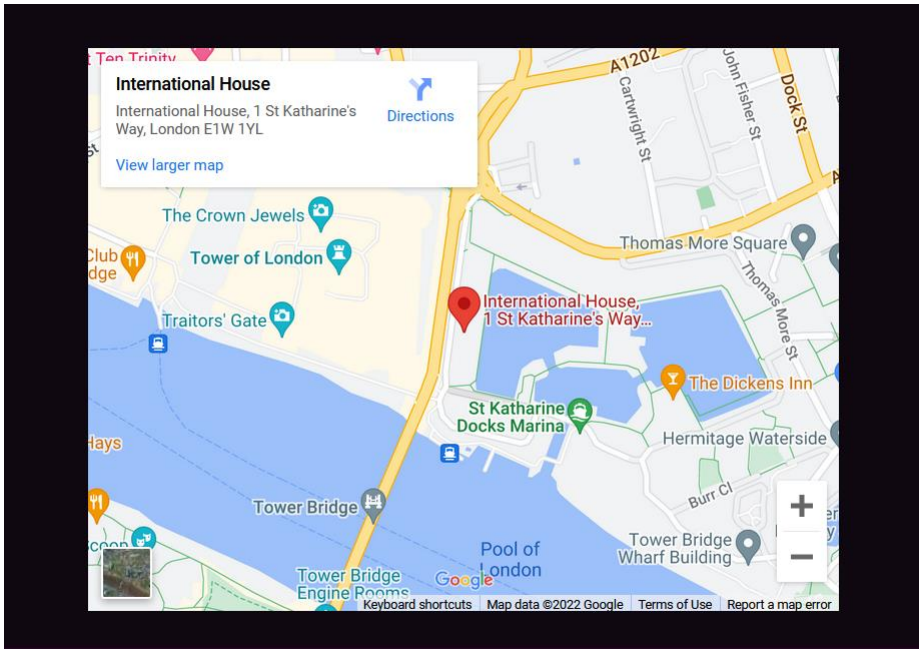
Day Seven

Day seven began as usual with a stand-up meeting and everybody discussing their tasks for the day. I did not want to spend another day troubleshooting the read functionality of the discussion board, so I requested the aid of a tutor, and unfortunately, we could not figure out why the read functionality was not working. However, it was pointed out that what I was trying to achieve was well above the MVP for the project. Here you can see the final discussion board:



This gave me peace of mind and I was able to move on from the discussion board, overall, I was very happy with it and how it can be used. From this I moved onto the Getting Here page, for this I wanted to implement an embedded google map as the added functionality of being able to scroll around the map to effectively plan a route to the cinema would be useful for many customers.

I had never done this before, but thankfully upon researching how it worked, I learned it was one simple line of HTML copied straight from the google maps interface. I wrapped it up in a container, sorted the positioning out, styled the rest of the page and that was that. Another page completed.



The rest of day seven was spend helping others and making sure we were in a good spot to just add some finishing touches on day eight before making a presentation.

Day Eight

Day eight began as usual with a stand-up meeting, with a more in-depth round up of where we were and which tasks we had left to complete, made sure Jira was up to date, assigned tasks and go to work. I initially spent time assisting with troubleshooting a JavaScript issue which took a little time, I then set my attention to ensuring the AWS RDS was functional and designed the way we needed and could be access correctly by Jenkins and the EC2 which was going to be used for hosting.

Once I figured out exactly how it was going to work out, a quick change within the RDS configuration was all it took to be sure that the database would be functional and work exactly as we needed it.

Once this was done, I shifted my focus back to the pipeline which I had originally configured on day one. When I plugged our repo into the pipeline, I had expected it would work as it did one day one. However, this was not the case. I ran into an issue on basically every stage of the deployment, so I spend the rest of the day dealing with this. I had limited experience with Jenkins however I was confident I could make it work.

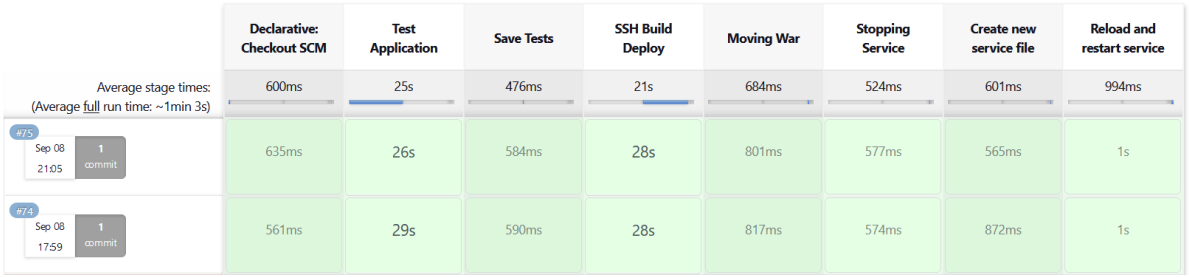
After a few hours I had done enough troubleshooting and fixing to allow the deployment to get to the second last stage, however it was getting close to the end of the day, so I again requestion a second pair of eyes on it to see if it was something simple, I was missing. It was not something we could see easily, but I knew I was close to solving it, when we logged off for the day, I had planned to get an early start to ensure I had enough time to solve the issue.

Day Nine

The final day of coding and final touches, I had started work a few minutes early to get my head back into the Jenkins problem I was having, however as 9am I held the daily stand-up, it was very exciting knowing how close we were to being finished. We split the final 4 tasks between us. However, there was still issues with the search page, so a good portion of the day would be spent on this. I made the

decision that we would get the other tasks done, and then we would collaborate on the search page issues.

As it was a new day, I had come into the problem refreshed and around 5 minutes after I ended the scrum meeting, I was able to solve the Jenkins problem and get the pipeline up and running as it was supposed to.



	Declarative: Checkout SCM	Test Application	Save Tests	SSH Build Deploy	Moving War	Stopping Service	Create new service file	Reload and restart service
Average stage times: (Average full run time: ~1min 3s)	600ms	25s	476ms	21s	684ms	524ms	601ms	994ms
#75 Sep 08 21:05 1 commit	635ms	26s	584ms	28s	801ms	577ms	565ms	1s
#74 Sep 08 17:59 1 commit	561ms	29s	590ms	28s	817ms	574ms	872ms	1s

As is tradition of the last day of a project, unexpected problems love to pop-up, and this day was no different, problems with the search page which required whole new functions to be written, JavaScript not working correctly on the films or films admin pages to name a few. The rest of the day, and then some, was spent dealing with these and overall, we got to a point with the final product that is very solid.

At the end

At the end of the final sprint, I am incredibly happy with the product we have produced, everybody put their heart into their work and really tried their hardest to create a great product and I think we succeeded in that goal. I am incredibly proud of everyone in my team and loved working with each of them.

With that I would like to say thank you to the team, Andrew Pimlott, Shai Lynch, and Onur Hassan. And I would also like to give a special shout out to all of the QA tutors we have had along the way, if you had told me 11 weeks ago I would be creating a full stack application for a cinema, with a Jenkins pipeline running on AWS EC2's connected to an AWS RDS instance, using a SpringBoot Java Back-end and a JavaScript front-end with full functionality I would have either laughed or cried. But here I am.