

PRACTICAL PROJECT: MOVIE REVIEWS APPLICATION

MOHAB KHALIFA

INTRODUCTION

- WHO AM I?

I AM A QA IT CONSULTANT TRAINEE. I HAVE COMPLETED MY FIRST PROJECT WITH QA. IN ORDER TO ACHIEVE THIS, I HAVE LEARNT HOW TO USE JAVA, MySQL AND MAVEN TO A COMPETENT LEVEL. I FACED MANY DIFFICULT HURDLES DURING THE PROJECT, BUT ULTIMATELY CONDITIONED ME TO BECOME A BETTER CODER.

- HOW DID I APPROACH SPECIFICATION?

PRIMARILY, I WENT THROUGH THE SPECS OF THE PROJECT TO GET TO GRIPS WITH ALL THE TECHNOLOGIES I NEED TO USE IN ORDER TO COMPLETE THE PROJECT ON TIME. MY NEXT STEP WAS TO BUILD ERDs (ENTITY RELATIONSHIP DIAGRAM) AND UML DIAGRAMS (UNIFIED MODELLING LANGUAGE) IN ORDER TO CREATE SOME STRUCTURE WHEN BUILDING MY CLASSES. I ALSO CREATED A JIRA PROJECT IN ORDER TO FOLLOW MY PLAN THROUGH A ‘SPRINT’. I TURNED THE DOMAIN SPECS INTO USER STORIES AS MY AIM WAS TO INTEGRATE THEM INTO THE PROJECT IN ORDER TO LINK EACH BRANCH AND COMMIT TO A USER STORY. THIS WILL ALLOW ME TO KEEP RECORD OF WHAT I COMPLETED AND WHAT IS YET TO BE FINISHED.

CONCEPT

- HOW DID I INITIALLY APPROACH THE PROBLEM?

FIRSTLY, I IDENTIFIED WHAT I NEEDED TO ACHIEVE THROUGH THE USE OF USER STORIES ON MY JIRA KANBAN BOARD. I THEN BROKE DOWN THE PROJECT INTO MULTIPLE LITTLE STEPS AND ASSIGNED EACH STEP A SPRINT. FROM HERE I WAS ABLE TO HAVE A STARTING POINT OF THE BACKEND DEVELOPMENT, THEN TESTING, THEN FRONT END. COMBINED WITH THE USER STORIES, I ALSO CREATED AN ENTITY RELATIONSHIP DIAGRAM (ERD) AND A UNIFIED MODELING LANGUAGE DIAGRAM (UML) TO UNDERSTAND HOW EACH PART OF THE BACKEND WOULD INTERACT WITH ONE ANOTHER.

SPRINT PLAN

- WHAT NEEDED TO BE INCLUDED?
 - THE PROJECT NEEDS A JAVA BACKEND DEVELOPED WITH SPRING BOOT ALLOWING FOR CRUD FUNCTIONALITY AND MANIPULATING THE SQL/H2 DB.
 - THE PROJECT MUST HAVE A HTML/CSS/JS FRONT-END DEVELOPED THAT PRESENTS THE USER WITH MOVIES AND THEIR REVIEWS THAT THEY CAN INTERACT WITH. ALSO IMPLEMENTS CRUD FUNCTIONALITY
 - UNIT AND INTEGRATION USING MOCKITO AND JUNIT
- WHAT DID I HOPE TO ACHIEVE?

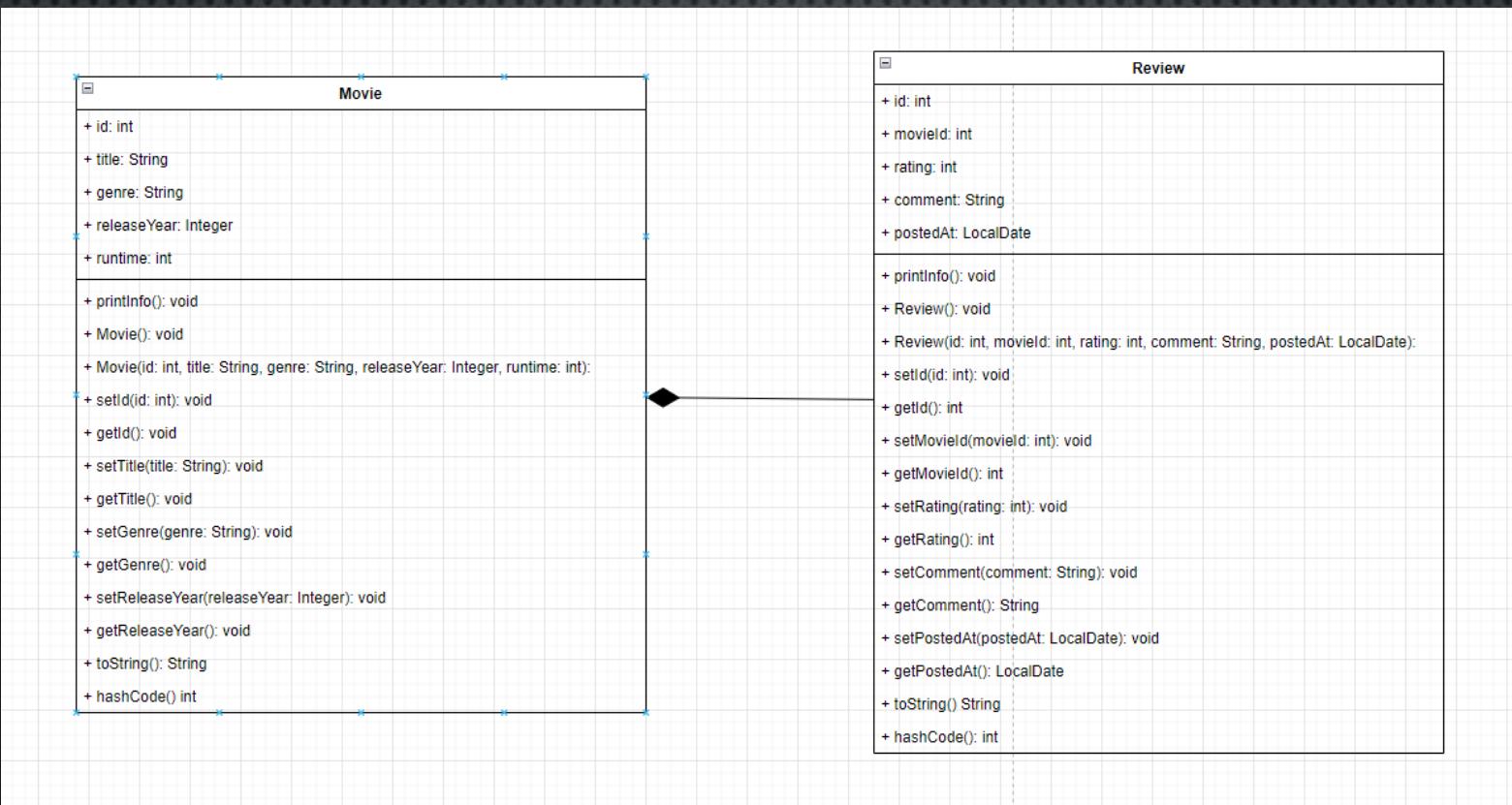
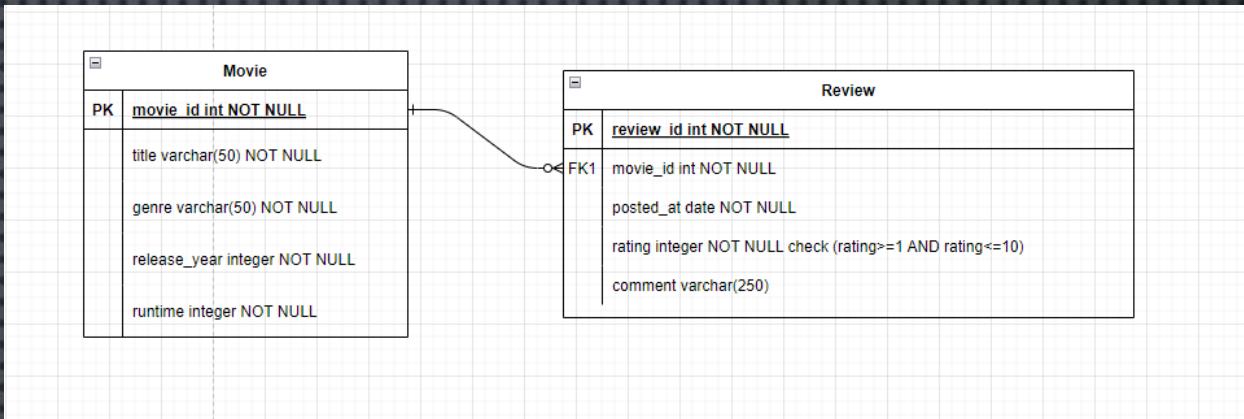
MY HOPES WERE TO HAVE A WELL DEVELOPED FRONT END PAGE UTILIZING CSS TO MAKE THE USER INTERFACE A NICE EXPERIENCE FOR THE USER.

MY PROJECT ROADMAP

- HERE IS THE SERIES OF USER STORIES I CREATED TO INSPIRE MY PROGRESSION THROUGHOUT THE PROJECT. I MANAGED TO GET MY CRUD FUNCTIONALITY FOR MOVIES WORKING, HOWEVER I STRUGGLED WITH REVIEWS. IT IS STILL CURRENTLY A WORK IN PROGRESS.

IN PROGRESS 4 ISSUES	DONE 5 ISSUES ✓
As a user, I would like to be able to change my review, incase I ever change my mind about a movie  MRP-5	As a user, I would like to view all movies, so I can see what options there are  MRP-1 ✓
As a user, I would like to delete a review, incase I no longer want to keep the review from showing  MRP-6	As a user, I would like to create a movie, so that I can later review it  MRP-2 ✓
As a user, I would like to view all reviews, so that I can see what people's opinions are on all movies  MRP-7	As a user, I would like to update a movie, so that I can correct any information that is wrong  MRP-3 ✓
As a user, I would like to view reviews for a specific movie, so that I can see people's opinions on that movie  MRP-8	As a user I would like to delete a movie, so that I can remove any unwanted movies  MRP-9 ✓
	As a user, I want to be able to create a review, so that I can rate the movie

ERD & UML



CONSULTANT JOURNEY

WHAT TECHNOLOGIES HAVE YOU LEARNED FOR THE PROJECT?

- JAVA
- HTML, CSS, JS
- JIRA
- SPRING BOOT
- MAVEN
- JUNIT
- MOCKITO
- GIT
- VS CODE
- POSTMAN

CONTINUOUS INTEGRATION

- HOW DID I APPROACH VERSION CONTROL?

GIT/GITHUB – THROUGHOUT THE DEVELOPMENT OF THIS PROJECT, I TRIED TO MAINTAIN CONTINUOUS COMMITS TO UPDATE MY CHANGES CONSISTENTLY. I MADE SURE THAT THE NEW UPDATES WERE ON FEATURE BRANCHES SO THAT IT DOES NOT MAIN THE SOURCE CODE OF THE WHOLE PROGRAMME. ORIGINALLY, WHEN I FORKED THE REPO, I INSTANTLY BRANCHED OFF THE MAIN BRANCH, TO CREATE A DEV BRANCH. I THEN BRANCHED OFF THE DEV AND INCORPORATED THE FEATURE BRANCHES WHICH I THEN LATE MERGED TO THE DEV. FINALLY ONCE DEV HAS WORKING SOURCE CODE THAT IS NOT BROKEN, I MERGED IT INTO THE MAIN BRANCH.

A screenshot of a GitHub repository interface. At the top, there's a dropdown menu set to 'dev', a branch count indicator ('20 branches'), and a tag count indicator ('0 tags'). Below this is a button for 'Go to file', a 'Add file' button, and a green 'Code' button. A message states 'This branch is 119 commits ahead of main.' To the right is a 'Contribute' button. Below this, a pull request summary is shown: 'Mohab-Khalifa Merge pull request #20 from Mohab-Khalifa/feature/testing ...' by '385dd15' 2 hours ago, with '123 commits'. The pull request details show two commits: 'initial commit' at '.mvn/wrapper' 4 days ago and 'added CrossOrigin annotation to controller classes' at 'src' 2 hours ago.

```
MINGW64:/c/Users/Mohab/Web-Dev/movie-reviews-frontend-project - □ X
[feature/fixes 4467d13] corrected script
1 file changed, 1 insertion(+), 1 deletion(-)

Mohab@DESKTOP-FSD9KM3 MINGW64 ~/Web-Dev/movie-reviews-frontend-project (feature/fixes)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 355 bytes | 355.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/Mohab-Khalifa/movie-reviews-frontend-project.git
 e5a67f7..4467d13 feature/fixes -> feature/fixes

Mohab@DESKTOP-FSD9KM3 MINGW64 ~/Web-Dev/movie-reviews-frontend-project (feature/fixes)
$ git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 627 bytes | 125.00 KiB/s, done.
From https://github.com/Mohab-Khalifa/movie-reviews-frontend-project
   6be70ec..8b9e0a4 dev      -> origin/dev
Already up to date.

Mohab@DESKTOP-FSD9KM3 MINGW64 ~/Web-Dev/movie-reviews-frontend-project (feature/fixes)
$ git checkout dev
Switched to branch 'dev'
Your branch is behind 'origin/dev' by 15 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)
```

TESTING

- WHAT WAS TESTED?

I TESTED ALL MY JAVA SOURCE CODE WHICH RESIDES IN THE BACKEND.

Spring_Movie_Project_1	68.0 %	2,182	1,025	3,207
src/main/java	43.1 %	751	990	1,741
src/test/java	97.6 %	1,431	35	1,466

MY AIM IS TO HOPEFULLY ACHIEVE 80% BY END OF THE PROJECT AND A BARE MINIMUM OF 60%. I STRUGGLED WHEN IT CAME TO TESTING AS I CAME ACROSS AN ERROR I COULD NOT SEEM TO RESOLVE. THIS HINDERS ME FROM ACHIEVING 80% BEFORE THE PRESENTATION.

```
org.hibernate.LazyInitializationException: could not initialize proxy [com.qa.intro_project.entity.Movie#1] - no Session
    at org.hibernate.proxy.AbstractLazyInitializer.initialize(AbstractLazyInitializer.java:176)
    at org.hibernate.proxy.AbstractLazyInitializer.getImplementation(AbstractLazyInitializer.java:322)
    at org.hibernate.proxy.pojo.bytebuddy.ByteBuddyInterceptor.intercept(ByteBuddyInterceptor.java:45)
    at org.hibernate.proxy.ProxyConfiguration$InterceptorDispatcher.intercept(ProxyConfiguration.java:95)
    at com.qa.intro_project.entity.Movie$HibernateProxy$lQI5x3Vf.setTitle(Unknown Source)
    at com.qa.intro_project.service.MovieService.updateMovie(MovieService.java:66)
    at com.qa.intro_project.service.MovieServiceIntegrationTest.updateMovieTest(MovieServiceIntegrationTest.java:93)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)
```

INTERACTING WITH THE API USING POSTMAN

DEMONSTRATION

INTERACTING WITH THE API USING FRONT-END

DEMONSTRATION

SPRINT REVIEW

- WHAT DID I COMPLETE?

I DID NOT MANAGE TO COMPLETE ALL THE USER STORIES I AIMED TO ACHIEVE WITHIN THE SPRINT. I MANAGED TO HAVE ALL CRUD FUNCTIONALITY WORKING FOR BOTH MOVIE AND REVIEW ENTITIES IN POSTMAN.

- WHAT GOT LEFT BEHIND?

I ONLY MANAGED TO ACHIEVE CRUD FOR MOVIE IN THE FRONT-END WEB DEVELOPMENT. I DID NOT CREATE THE CRUD FUNCTIONALITY FOR REVIEWS IN THE FRONT-END IN TIME, ALTHOUGH IT IS STILL FUNCTIONAL IN THE BACK-END. FURTHERMORE, I DID NOT ACHIEVE THE ACQUIRED 80% TEST COVERAGE, HOWEVER I STILL HAVE A BIT OF TIME...

SPRINT RETROSPECTIVE

- WHAT WENT WELL?

CREATING THE ENTITY CLASSES WAS RELATIVELY STRAIGHTFORWARD AND SIMPLE TO MAKE WHILST IDENTIFYING THEIR RELATIONSHIP (I.E. ONE TO MANY). SERVICE CLASSES WERE ALSO EASY TO MAKE. I MANAGED TO GET FULL CRUD FUNCTIONALITY IN THE BACK-END TO WORK.

- WHAT COULD BE IMPROVED?

POSSIBLY CREATE A BETTER PLAN IN THE FUTURE. I DID NOT HAVE ENOUGH TIME TO COMPLETE THE FRONT-END AS I LOST TOO MUCH TIME DEVELOPING THE BACK-END (MOST IMPORTANT), AND TESTING THE BACKEND CODE. I RAN INTO A LOT OF UNKNOWN ERRORS IN TESTING WHICH I HAD TO TRY OVERCOME BY DOING A LOT OF RESEARCH. I FELT LIKE ONE WEEK WAS NOT ENOUGH FOR ME TO ACHIEVE EVERYTHING I SET OUT TO ACHIEVE.

CONCLUSION

- REFLECTIONS ON THE PROJECT, FUTURE STEPS, ANY OTHER RELEVANT INFO

ONE OF MY MAIN REFLECTIONS IS DEFINITELY TIME MANAGEMENT FOR THE PROJECT. TO MAKE SURE I PLAN MY TIME ACCORDINGLY WITH SOME LEEWAY TO PREPARE FOR BUG FIXES THROUGHOUT MY PROJECT.

I AM PRETTY PROUD AS TO HOW I MANAGED MY VERSION CONTROL, THROUGH CREATING FEATURE BRANCHES AND LATER MERGING THEM INTO THE DEVELOP BRANCH.

ALSO TAKING BREAKS ACCORDINGLY WHEN WORKING. AT THE END OF THE WEEK, I FELT MENTALLY EXHAUSTED AS I SPENT THE WHOLE WEEK ON MY MONITOR, NOT TAKING AS MANY BREAKS AS I PROBABLY SHOULD HAVE. I NOTICED WHEN I DID TAKE THE BREAKS, IT CLEARED MY MIND A BIT MORE AND HELPED ME IDENTIFY ISSUES QUICKER AT TIMES.

FINALLY, THE MAIN TAKE FROM THIS, IS TO KEEP UPSKILLING AND TO TRY UNDERSTAND THE CODE MORE, ALLOWING ME TO CREATE CLEANER CODE AND INCREASE WORK EFFICIENCY.

ANY QUESTIONS?