QA CINEMA

Team Instinct members:
Ali Khattab
Sean Fernyhough
Hamid Mohammed
Eamon Moradi
Adam Yusuf Mussa
Hamza Shah

INTRODUCTION

- Who are we?

- What was the brief?

To create a full-stack web application for a client.

Our client is a well-known cinema chain, **QA Cinemas**

Our team has been tasked with building a new website for them, which should present information about movies, listings, upcoming releases etc.

There are various features which the client has requested, some are **essential** while others are *desirable*.

There is time for **two 5-day Sprints** before the site *goes live*.

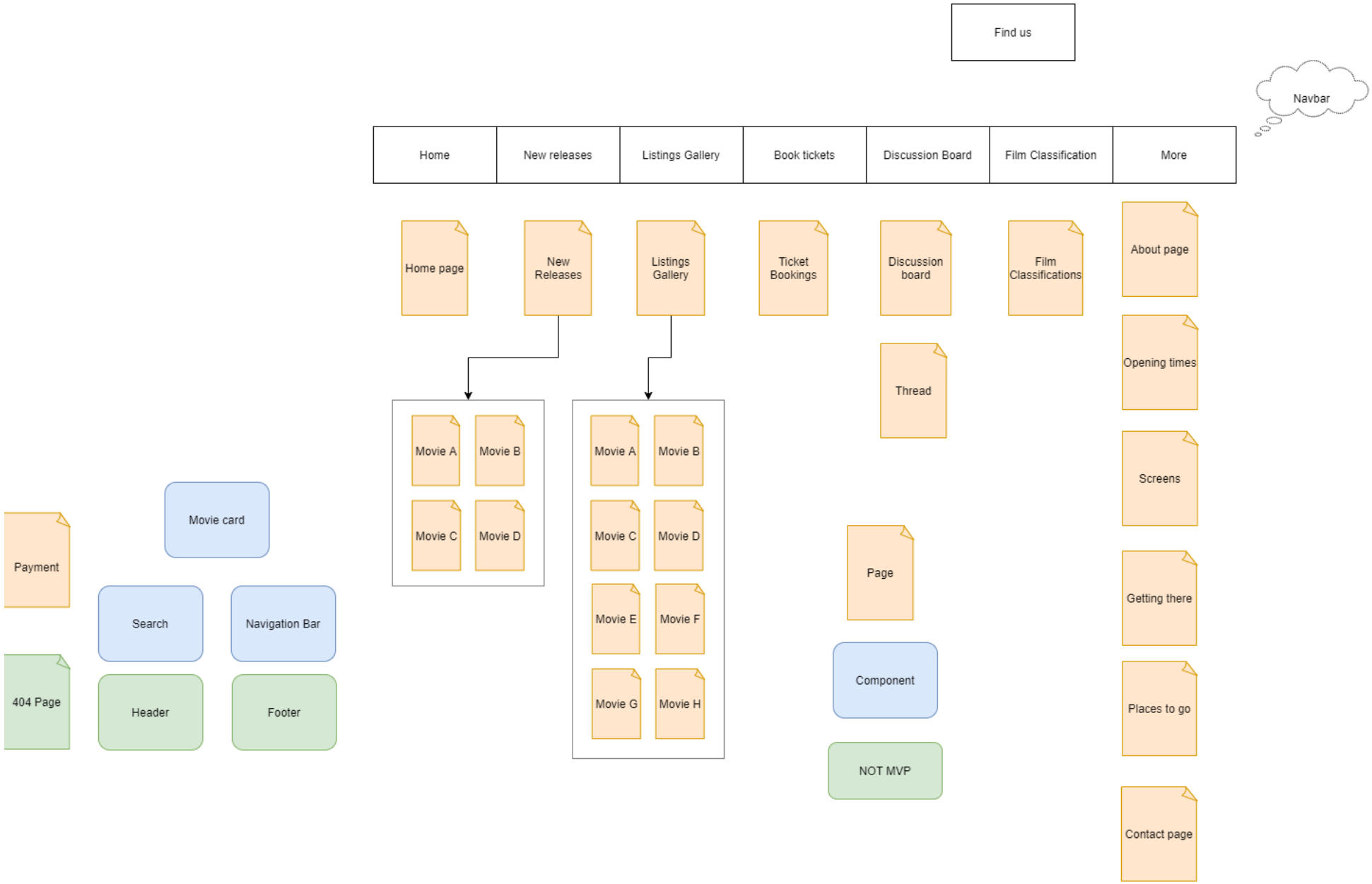
PLANNING: HOW DID WE APPROACH THE SPECIFICATION

The role of scrum master was assigned to Ali, who then set up the GitHub repository and the Jira Board.

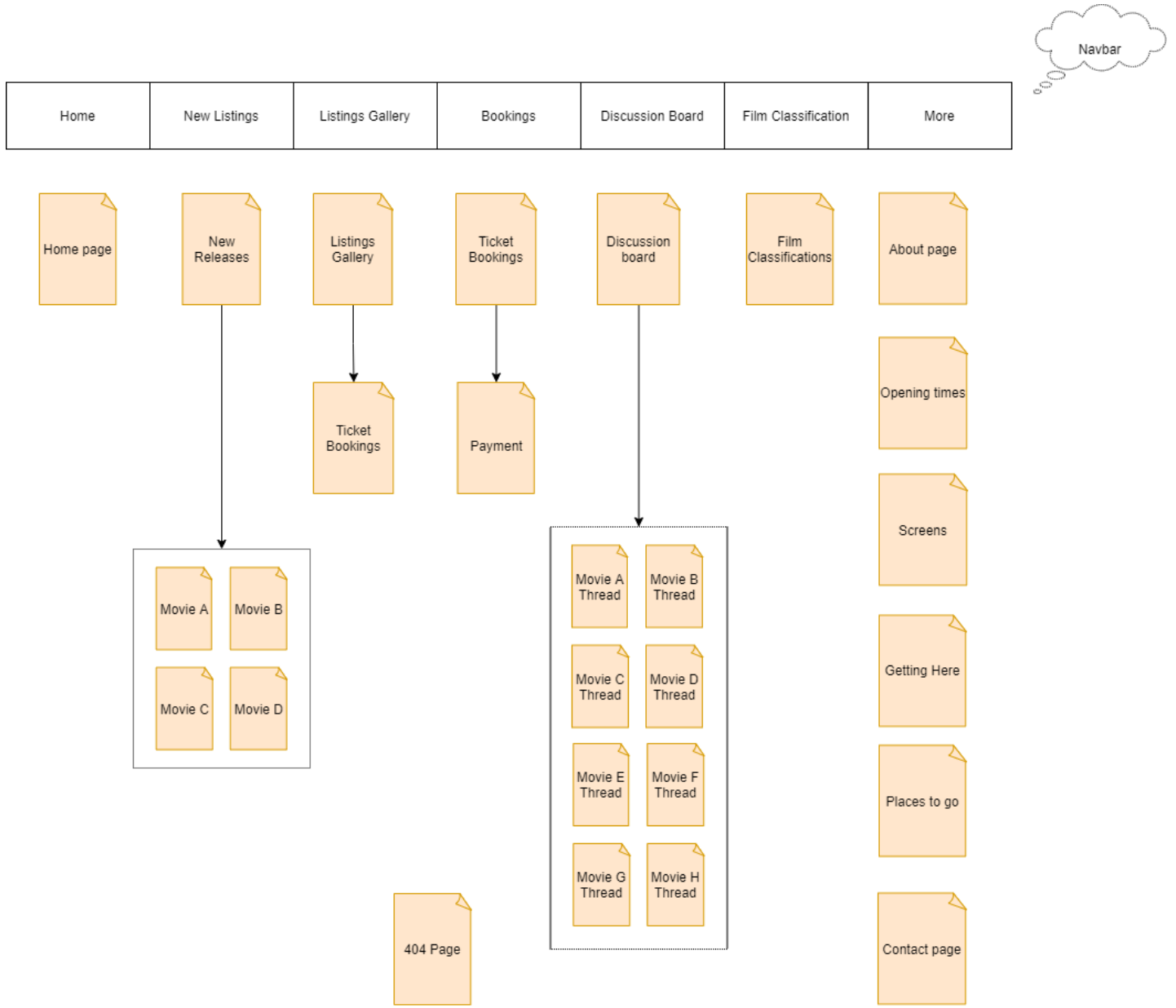
In depth discussion talking about the MVP and breaking it down into manageable user stories

Negotiations around who would take each user story based on interest, confidence as well as vision and creativity.

This felt like the best way to get through the tasks efficiently and creating the best product we could make



Initial Frontend Layout Diagram



Final Frontend Layout Diagram

SPRINT PLANNING

- Sprint planning was done at the start of each week
- Planning involved filling the Jira Board with relevant User Stories with priority for MVP and a few optional user stories which had a lower priority
- Stories were assigned point estimates based on difficulty and members of the team were assigned stories
- Daily Standup meetings were held to measure progress and reassign tasks if necessary
- Daily Standup meetings also allowed for people to express concerns and other team members to help out where they were needed most.

EXAMPLE OF USER STORIES:

Customers of the cinema will have the following user stories:

As a user I want to be able to navigate through the website and access various pages.

As a user I want to be able to see the new listings at the cinema.

As a user I want to be able to see what movies are currently listed at the cinema.

As user I want to be able to contact the cinema directly through the contact us page.

As a user I want to be able to make bookings for their desired film.

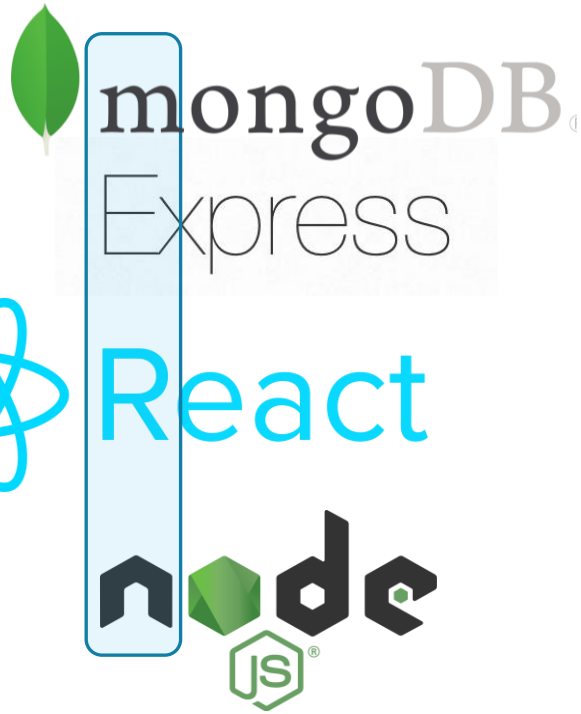
As a user I want to be able to use a moderated discussion board to discuss listed films. (age friendly)

As a user I want to be able to search for movies.

ERD DIAGRAM


showings
_id: String
screen: String
date: String
timeRaw: Number
time: String
movie: movieSchema
seats: seatChema
movies
_id: String
title: String
rating: String
runtime: Number
cast: [String]
imageUrl: String
release: String
tags: [String]
description: String
discussion
_id: String
name: String
text: String
movieId: String
thread: String
rating: String
booking
_id: String
showingId: String
bookingRef: String
showingDate: String
showingTime: String
name: String
seats: [String]
children: Number
adults: Number
movieTile: String
screen: String
total: Number
seats (schema only, not stored in DB)
seat: String
taken: boolean

CONSULTANT JOURNEY



mongoose

elegant mongodb object modeling for node.js

axios

EmailJS

https://www.embedgooglemap.net/

Google Maps

stripe

CI: HOW DID WE APPROACH VERSION CONTROL?

Git was used as Version Control System (VCS)

Feature-Branch Model (FBM) was used to reduce merge conflicts

GitHub was used as online Repository Hosting Site

Jira was used to keep track of user stories

MONGODB

- MongoDB is a non-relational database management system that stores objects as documents within collections.
  - Collections are stored in databases.

- Database stored on MongoDB Atlas
  - Requests from express to remote MongoDB Atlas URI

- Allows for persistent storage accessible from any device

- Mongoose is a node module that acts as an interface between express and MongoDB
  - Mongoose allows MongoDB schemas and queries to be written in JavaScript

EXPRESS

- Express was used as our backend framework within JS
- Along with Mongoose and Axios, Express allowed us to create an API to interact with our cloud instance of the database
- Express is installed as a Node Module which allows routing of different paths allowing for http requests (POST, PUT, PATCH etc...)

REACT

- React is a JavaScript library that we used in our web application. It is a tool used to build User Interfaces (UI). It was created by Facebook with the aim of simplifying the development of visual interfaces.
- Throughout the web application we have used numerous react components such as carousels or cards. These components allowed us to organise information across our webpages.
- Libraries such as Reactstrap allowed us to incorporate the bootstrap toolkit in React to easily modify style sheet elements

TESTING

Testing was performed on the express backend using mocha and chai

- Mocha is a JavaScript testing framework which allowed us to write tests for our backend
- Chai is an assertion library that allowed us to send HTTP requests to the backend and make assertions based on the response, this ensured express could effectively communicate with our MongoDB database
- Istanbul is a JavaScript coverage tool that showed the total code coverage of our tests, this meant we could be sure exactly how much of our code was being tested.

TESTING COVERAGE

```
  29 passing (2s)

--------------------|---------|----------|---------|---------|----------------------------
File                | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
--------------------|---------|----------|---------|---------|----------------------------
All files           |   82.15 |    51.85 |   86.95 |   82.15 |
 server             |   90.32 |       75 |      50 |   90.32 |
  server.js         |   90.32 |       75 |      50 |   90.32 | 19,45,63
 server/persistence |     100 |      100 |     100 |     100 |
  booking.js        |     100 |      100 |     100 |     100 |
  discussion.js     |     100 |      100 |     100 |     100 |
  movie.js          |     100 |      100 |     100 |     100 |
  showing.js        |     100 |      100 |     100 |     100 |
 server/routes      |    78.6 |       50 |   89.23 |    78.6 |
  bookings.js       |   80.55 |       50 |    92.3 |   80.55 | 8-9,19,29,42,49,59
  discussions.js    |   76.92 |       50 |    92.3 |   76.92 | 9-10,20-21,31-32,48,55,69
  movies.js         |   80.43 |       50 |   94.11 |   80.43 | 10-11,21,31,41,51,64,71,81
  payments.js       |   54.54 |      100 |       0 |   54.54 | 7-8,13-35
  screens.js        |       0 |        0 |       0 |       0 |
  showings.js       |      82 |       50 |   94.73 |      82 | 8,18,41,48,58,68,78,88,98
  testAPI.js        |      80 |      100 |       0 |      80 | 5
```

SPRINT REVIEW: WHAT DID YOU COMPLETE? WHAT GOT LEFT BEHIND?

Sprint 1: We completed all our user stories on Jira for our epics which you can see on our Jira board. Along the way we added new user stories to our sprint where we thought the project could be improved and spiced up by adding extra features.

No user stories were left behind during sprint 1 and we accomplished all our targets for this specific sprint.

Sprint 2 : Likewise, all user stories were completed and extra features were added to our sprint where we thought our project could go that extra mile.

Again, no user stories were left behind but we did modify pages from our previous sprint to improve functionality. Throughout the project smart commits were made to keep our work tracked and tracible.

SPRINT RETROSPECTIVE: WHAT WENT WELL? WHAT COULD BE IMPROVED?

Overall the project went well learnt many things on the way and debugging was great fun. Completed the checklist plus wish list and lived up to the specification requirements by going further to achieve further stretch goals.

We worked very well as a group in terms of planning, delegating tasks, support from each member and our overall punctuality plus the motivation to achieve our goal.

Things to add if more time:

- User account section - modify bookings, subscription accounts
- Cinema admin section (add movie, delete movie etc..)
- Trailer video
- Discussion board from other sites e.g. Facebook, IMDB etc
- Search for anything within the site
- Email customer when successfully booked with seat numbers etc
- Age validation for movie bookings and discussion boards

CONCLUSION: REFLECTIONS ON THE PROJECT, FUTURE STEPS, ANY OTHER RELEVANT INFO

- We completed the MVP and the Wish List and began implementing ideas and features that we believed would make the website not only more appealing, but intuitive to navigate making a seamless user experience.
- We had time to refine and continuously work on design to present the MVP and our skills to the best of our ability.
- We believe that a further step could be acceptance testing to see how easily users can interact with the website. And adding features to streamline ticket sales and increase traffic to the website.

DEMONSTRATION

QUESTIONS

https://github.com/aliktb/team-instinct-cinema-project

https://emaplejiralxc1.atlassian.net/jira/software/projects/CIN/boards/4/backlog