Building Web Applications using ReactJS

Hackathon Guide

CONTENTS

ReactJS Hackathon

Overview

In this Hackathon you will be presented with the requirements to create a website for QA Cinemas. The skills and knowledge needed have been covered in the course so far.

Objectives

To use the skills and knowledge gained so far to create a ReactJS application that displays information using both static and stateful components. Data for the application will come from an external source of data (i.e. a RESTful service).

Application Overview

QA Cinemas has approached us to create a modern looking web application promoting their cinemas. In the first iteration, they would like to be able to provide their customers with general information about their cinemas, schedule and opening hours and allow them to subscribe to receive some promotional materials.

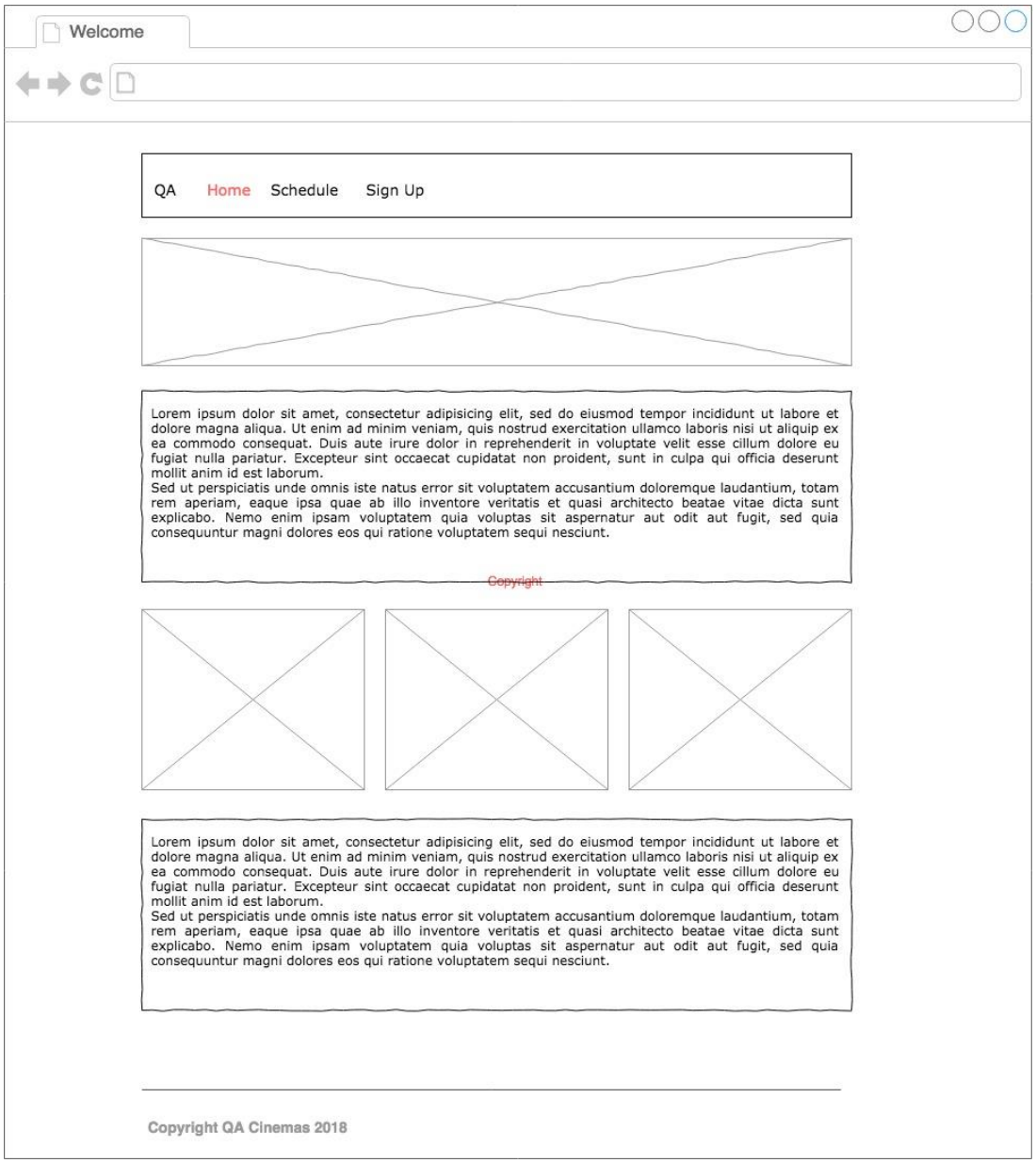
Mock Ups

The business representatives have worked with our UI/UX team and they have created the following low fidelity mock-ups. They understand that their customers view their pages on different devices and want us to accommodate to this. The site will be split into three routes –

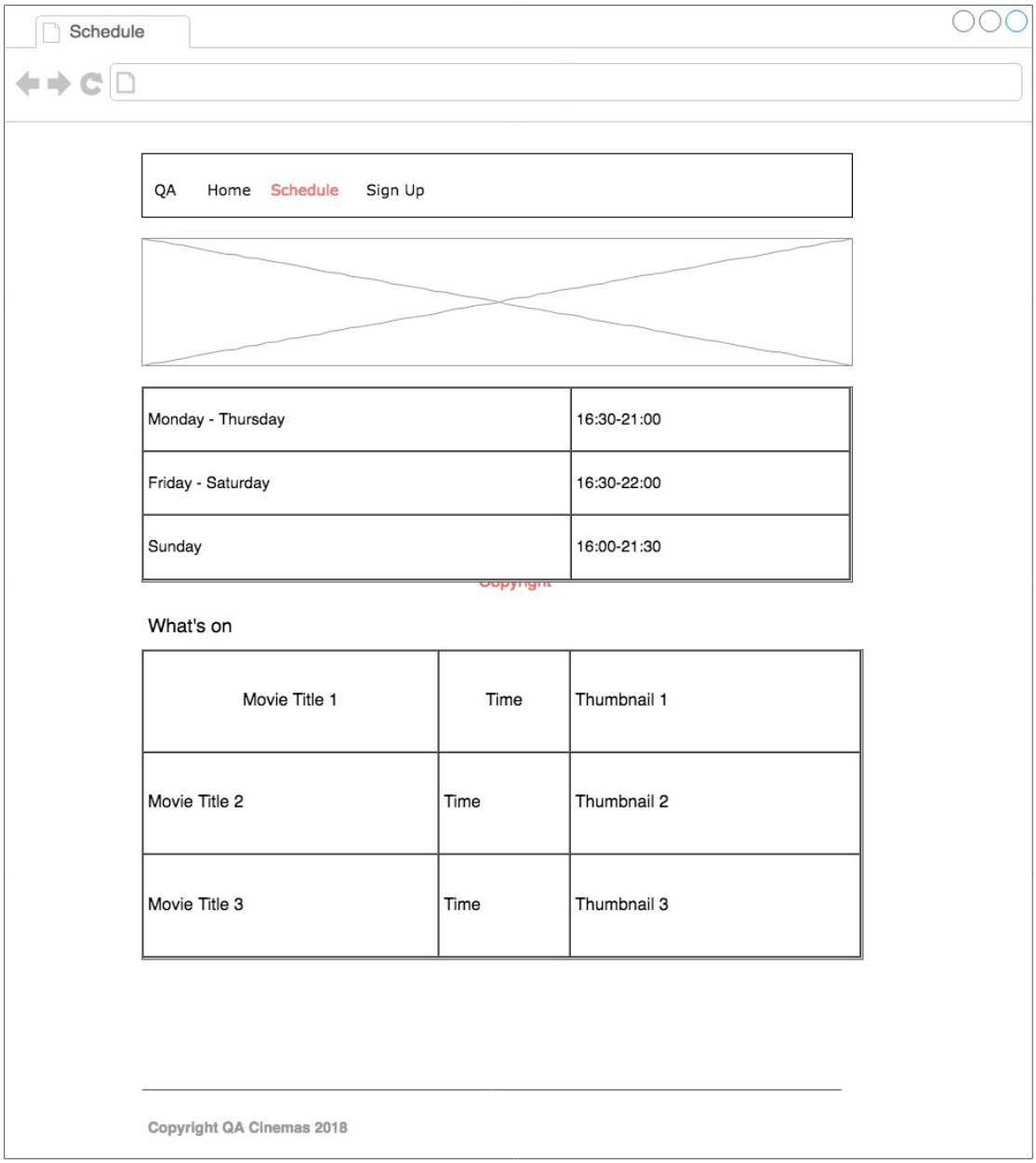
- home route with information about the Cinemas;
- a 'schedule' route displaying opening hours and movies showing this week;
- a 'signup' route to gather some information about the user.

The navigation between the pages should be easy and straightforward, with a navigation bar on top of each page.

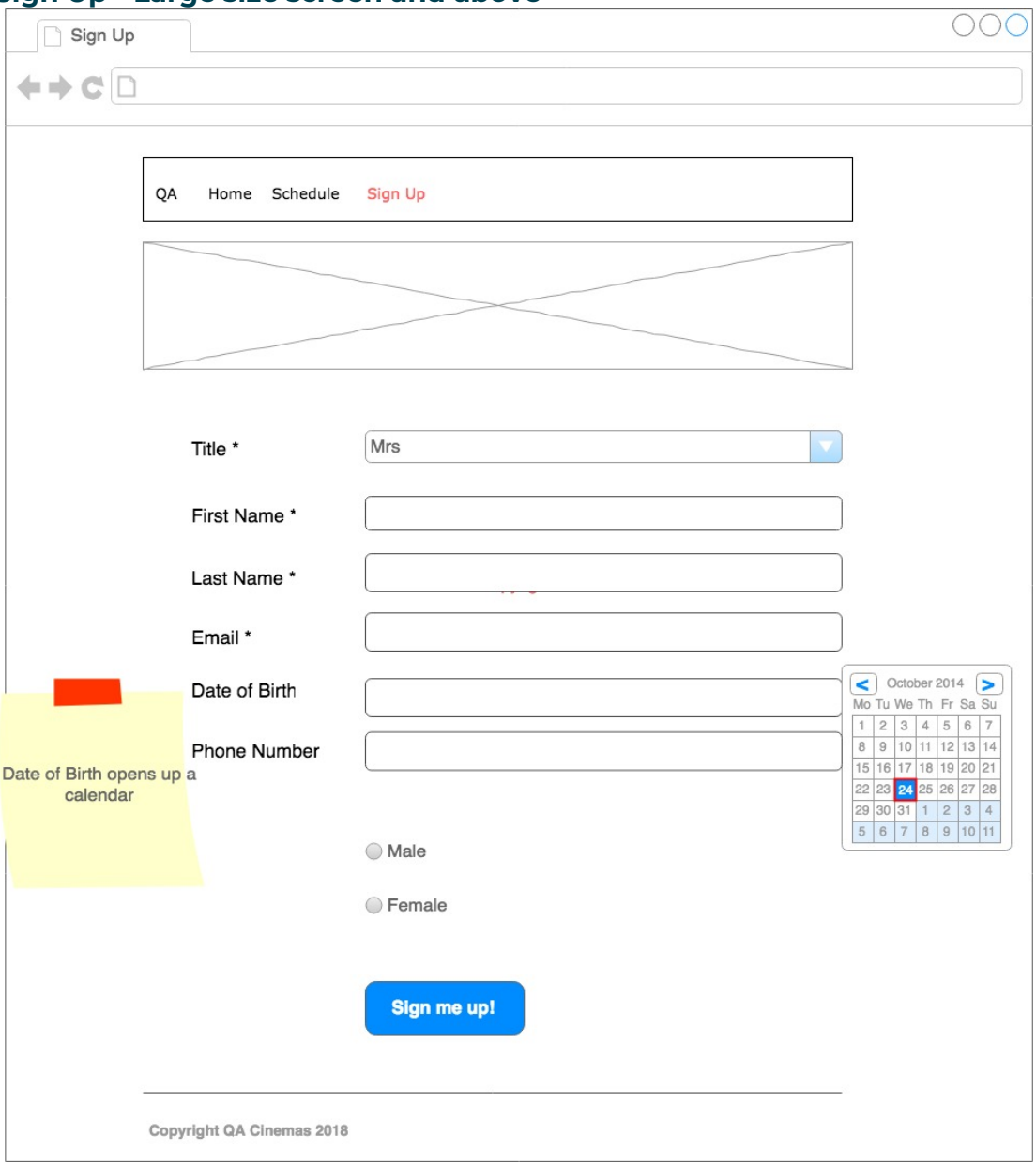
Home Page Mock Up - Large size screen and above



Schedule Mock Up - Large size screen and above



Sign Up - Large size screen and above



Mock Data:

A FILM object will be in the following format:

```
{
 "_id": "5c9e51c24c6ee53ff09d5d01",
 "title": "Test Film 1",
 "synopsis": " Test Film 1 Synopsis",
 "cast": "Test Film 1 Cast",
 "directors": "Test Film 1 Directors",
 "showingTimes": "11:45, 13:30, 17:15, 20:45",
 "releaseDate": "2019-05-27T00:00:00.000Z",
 "filmStatus": "1",
 "img": "film1Image.jpg"
}
```

Please note:

_id is stored as a string in MongoID value format.

releaseDate is an ISO Date string. It can be passed to a JavaScript Date object as a string to create a JavaScript Date object.

User Stories

The User Stories have been prioritised and ordered from 1 (most important) to 5 (least important):

1. As a User, I want to be presented with a home page, so that I can see some general information and pictures showing QA Cinemas.
2. As a User, I want to be able to see the opening hours of the cinema, so that I know when the cinema is open.
3. As a User, I want to be able to see the movie schedule for this week, so that I know what is on and at what time.
4. As a User, I want to be able to subscribe to the website using a form, so that I can receive the latest promotions and information.
5. As a User, I want to be able to access the website on any device, so that I can access the information I need on the device of my choice.

Minimum Acceptance Criteria

Discussions with the client have yielded the following requirements with regards to the architecture and development of the application:

1. It should be built using ReactJS.
2. A RESTful service should be used - this is provided complete in a Docker container. (Information about how to download, run and use this service can be found in the Appendix of this document).
3. The application should be developed using a Test or Behaviour Driven Development methodology.
4. Test Driven Development requires tests to be written BEFORE production code is written.
5. Behaviour Driven Development requires tests to be written to check that the application behaves as expected given know data.