App Design

Johnny Dove

Student number:1911794

Module: (CM3131)

Github:

Purpose of App

My app called Show Aid it is an app designed to aid avid television watchers with the ability to search for and favorite shows displaying enough information to help the user make a decision on what to watch or to be able to manage a favorites list after watching a show they like that they could show to others.

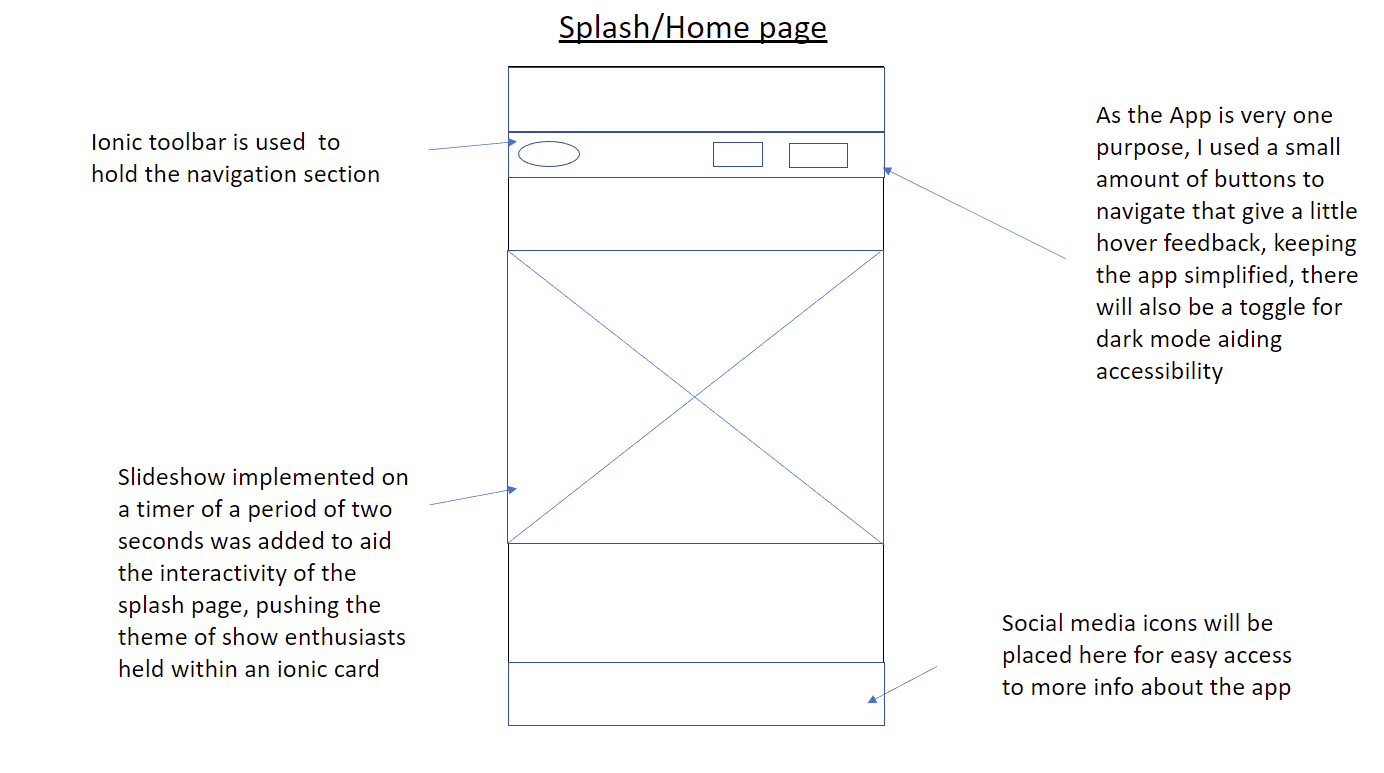
* The ability to search for a show
* Images displayed to improve memory of a show
* Information displayed about the show
* The ability to add a show to your favorites list
* The functionality to remove the show from your list
* Extra functions to aid the users comfort level when navigating
* Keep all data persistent
* Keep a clean UI

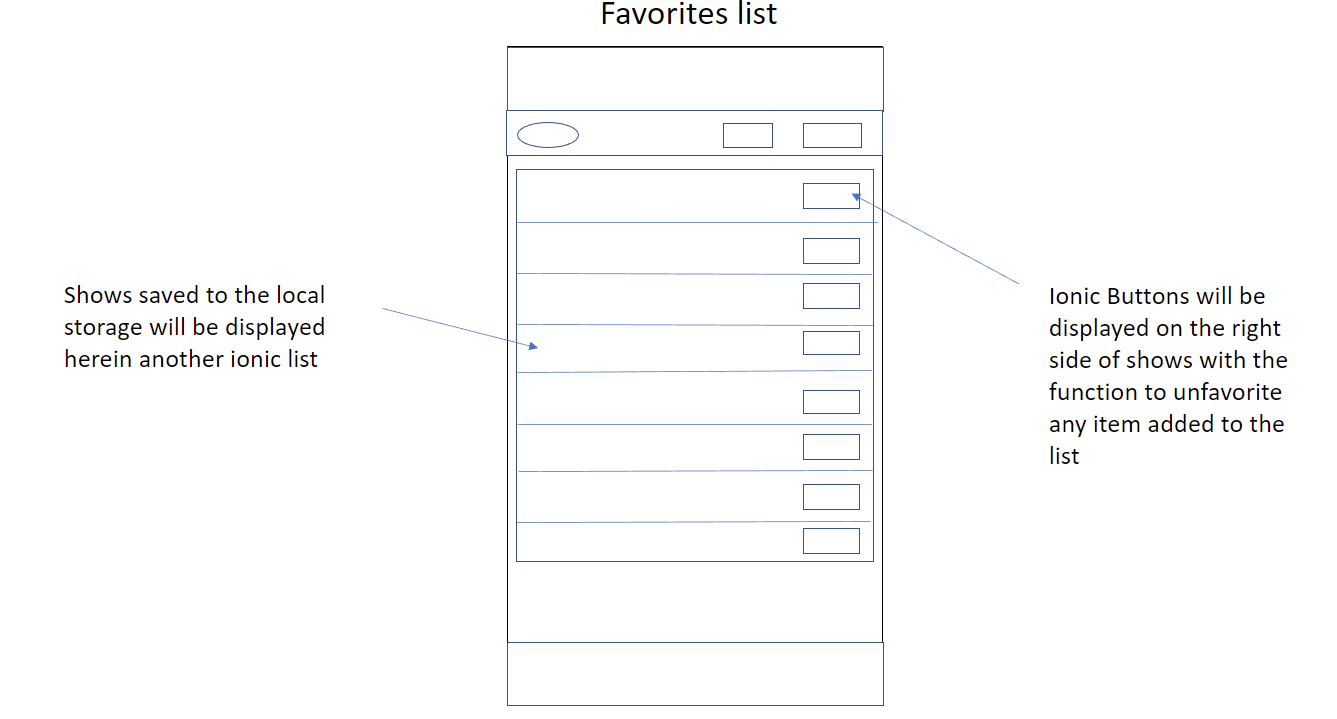
All basic main functionalities were fully achieved, being able to fetch the API and pull its data to display a large amount of shows, the API provides a level of “Fuzziness” aiding the user in finding a show by guessing what show your looking for based on the characters used in your search. Images were also successfully fetched using the fast library of posters the database holds; there are some sizing’s issues as the images are barely visible on smaller resolutions but maintain an okay level of visibility on most phone scales but all information such as (description, rating, premier, Genre) are displayed successfully with a decent level of readability. Once a user has found what they would like to add to their favorites they can add it to their favorites by clicking the favorite button displayed to the right of every single listing. Once the favorite button is clicked there is visual feedback displaying the button now as unfavourite.

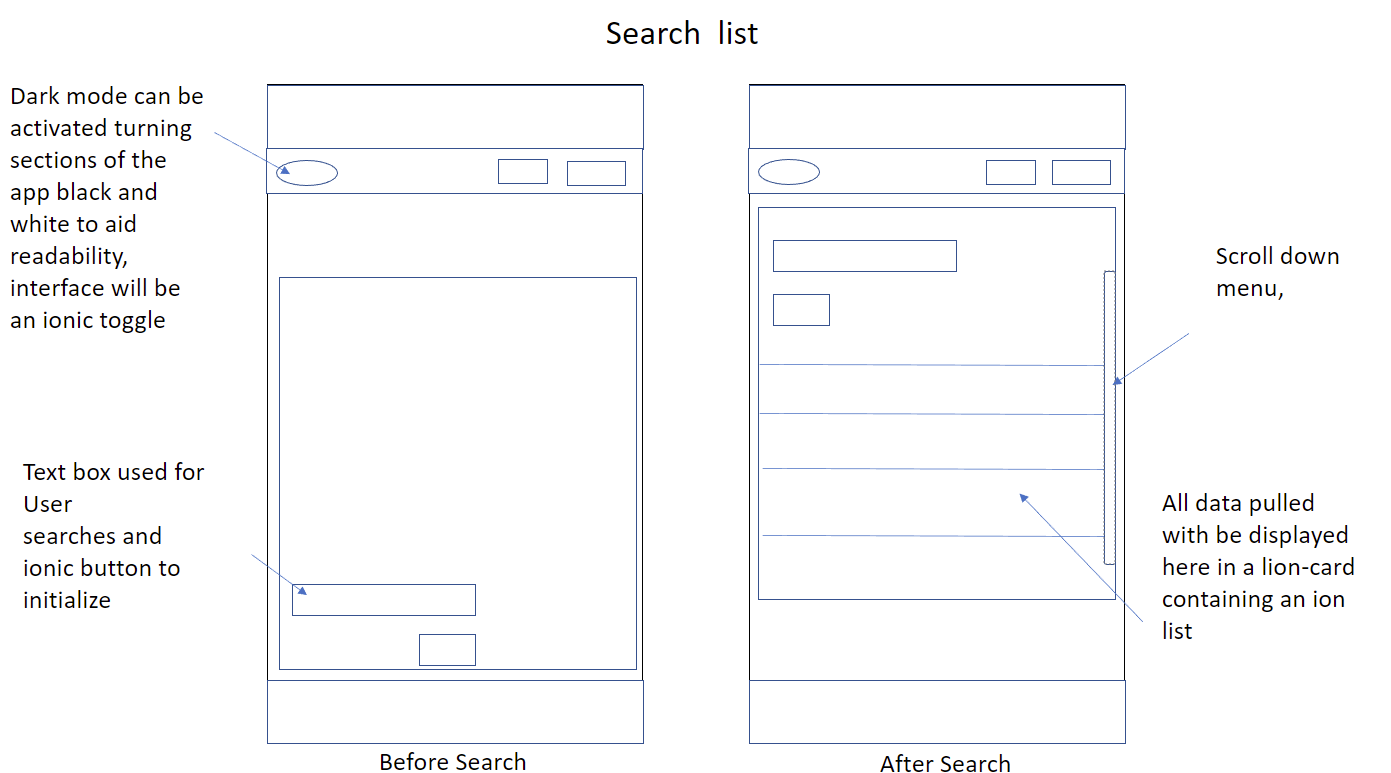
Navigating now to the favorite’s page the user will be greeted to their newly favorited show displayed in the list, again all having the option to unfavourite. All data has been successfully kept persistent throughout the app such as in testing returning to a same search a user last tried the previous item will be still favorited and refreshing the page has no impact either to the stored data. And finally a small comfort function was added of dark mode that can be applied to pages with an ionic toggle however this was a half successful as some ionic components do not react to the color change. Social media icons were also added to the bottom of each page to give easy access to more information about the company; these were done with icons that have links embedded in them I thought this would be fine in comparison to the ionic icons as they bring a little less color to the page.

Design process, the design was kept simplistic to mainly focus on the core functions of the app which was the use of the API. Overall the functionality I set out to create in this app was completed however there is definitely room for improvements to extend the current features such as searching for cast actors, also refinement in the UI is needed with more styling however the UI currently works sufficiently at a basic level.

**WireFrames**







**Javascript**

Firstly the built-in IndexedDB function onupgradeneeded creates the database and Favourite Shows object store with a given schema if the database does not already exist. If the database already exists then it is simply connected to. Once the user presses the search button their search terms are captured and formatted so they are appropriate for use in the TVMaze API. The Ajax method getJSON is used to query and fetch TV show data from the API. Once this data has been parsed it is added to an ion-item which is added to the search results list.

When a user chooses to favourite a show, the show's details are added to a Favourite Shows object store. Each time a user performs a search, the Javascript checks if each discovered show is in this object store (meaning the user has favourited it). If it isn't, the show's favourite button text is set to "Favourite." If not, it's set to "Unfavourite." When a user opens the Favourite Shows page, details of shows stored in the Favourite Shows object store are added to ion-items which are added to a displayed ion-list. When a user unfavourites a show on this page, the show (identified by its unique ID) is simply removed from the object store and the displayed list.

**Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose of Test | Actions performed during test | Expected outcome | Actual outcome |
| To find out if the database is holding data effectively and staying persistent throughout sessions | Refreshing the page and leaving and returning to the app | For the favorites list to retain all items that were previously favorited | All listed items in the favorites list were successfully held within the app |
| To see if the search function worked correctly | Did several tests running through different examples of phrases that could be linked to a show | For data to be pulled from the database and shows to be displayed on a list that are similar to the search phrase the user used | Information was successfully displayed in the correct format, pulling essential information and images for the shows as expected |
| Navigation test, this test was to see If a user could navigate through the app successfully | I began to navigate using the ionic buttons provided on the ionic toolbar moving from page to page | That the button indicating a certain page such as “home” would successfully take the user to page that was labeled on click of the button | This test was successful, however at the time of this test one button was missing for a return from the favorites list to the search bar, this was implemented shortly after the test |
| Visual feedback, the purpose of this test was to see if favouriteing a show registered properly triggering the button to give visual feedback to the user | For this test I searched up an example phrase in this case “car” and clicked on the favorite button on the first result displayed on the ionic list | I expected the button to quickly change from “favorite” to unfavourite” showing that the database has requested the request, hoping to also see some response come through on the console | This test was also successful seeing a full response from the API in the console, and the buttons text changing to “unfavourite” |
| The purpose of this test was to test the function of the dark mode ionic toggle | For this I tested the toggle on every page seeing what impact it would have on the readability of the page | What I would expected was for the page to entirely change to black and text change to white, greatly increasing the readability for certain users | Unfortunately this test failed and the dark mode would not affect the ionic toolbars |

**Security**

A consideration for the app would be to have some sort of log in user system; this would provide basic layer of security protecting the user’s data, in this case there stored favorite’s list so no sensitive data is used. No personal information being stored on the app other than a show favorite’s list really could aid users in their concerns for security of the app as they use the app there is no risk of any sensitive data loss at any point of the apps main processes.