6MMCS001W.1 - Mobile User Experience

Common Group Report

Group_6_Std_1_iPhone_Portrait
Tomasz Jurek – w1576380

Enhanced navigation bar
Mainly given as the major highlight from the Feedback on design given on 08/11/18, the design did not clearly dictate if the navigation bar will or will not be sticky. Given the amount of content of the web application and strictly conservative approach to viewable space I have applied a function in format of jQuery tags that makes the navigation bar sticky giving user constant ability to maneuver the web application, however if user switches focus to an object on web page and clicks on it, the navigation bar will hide and field view for user will be expanded, giving them more reading/viewing space. The menu can be easily brought back with click onto the same object.

Built and enhanced welcome page
When user enters the website, they are greeted by animated boxes which showcase the fashion week events much before user engages with the website. The technique captures user's attention and within those animated gifs are buttons that also highlight most important features of the application, thus not only encouraging user to click and engage with the website but also give quick and direct access to core features of the web app without need of the clicking on hamburger icon to get full view of all possible accessibilities of the web app.

Implementation is pretty simple as the page is mainly static and thus it does not require any JavaScript. The avoidance of JavaScript was intentional as the index page is the very first page that user sees, and thus it must be loaded with optimal speed. I was considering of using videos for the interactive aspect of the welcome page, although decided to convert videos to high quality gifs and essentially save about 3-4 seconds of buffer time and also achieved same effect as video.

Built and enhanced Events Page
When building the events page, it was essential to achieve quick filtering and searchability of all available events. The jQuery mobile became extremely handy as it allowed me to create a "list of lists" effect. The listview initially splits the categories into different sections and user can instantly click to which section they desire to go. When opened another embedded listview with thumbnails appears. On top of that, to save the user from scrolling time, if the user knows exact name of the event for given category, the filter appears right at the top ready to search for desired event.

Updated Events Item page
The events page highlights the single event with clear heading, description, time and 3 control buttons which allows user to engage with given event. The first button covers the gamification aspect of application, where user can invite friend to given event, and if user invites 10 friends via email, a 20% discount will be applied to the next purchase. This discount is incorporated in the payment backend. The favorites button adds the given event to the favorites page and the buy tickets button allows user to enter and save their payment details and purchase the ticket, more on this feature in next paragraph.

Booking form and Backend Payment

Developed a form as modal which pops up when user attempts to purchase a ticket. This form has ability to save users card and details for next purchase thus no continuous data entry is needed. This modal form is connected to the Stripe backend vendor which allows to process the transaction. Currently the application uses a test token and runs in the test mode thus no real purchases can be applied, and test cards can be used.

However, before the transaction is committed, a script detects if user has reached the discount potential with their invites, if so, alternative function is triggered which calculates the discounted amount and applies it to the price of the ticket. This approach at gamification is most likely the easiest and user friendly, as the system takes care of the tracking and does not require special promotion codes which can be tiresome to users as they can forget or lose them.

Miscellaneous fixes, debugging and refactoring

After the team has completed all allocated tasks and we merged our work, we had to fix issues across the entire page, every group member was involved including me, for which I took lead in fixing linking to each pages and choosing the best method, from using jQuery load functions to href direct to web pages to preserve slide animations. Furthermore, I have implemented various fixes in payment mechanism where each ticket has different price ranging between 5 to 25 GBP and also implementation of 20% discount function which is applied when user has unlocked the discount through our gamification process. Other miscellaneous fixes include fixing footers in the web page to correctly display vector svg images and share the persistent theme across the web app. Also I have added a favicon (an icon displayed on tab) which adds extra level of professional feel as each aspect of the web apps outlook is considered.

Group_6_Std_2_iPhone_Landscape
Bror Nordstrom – W1552186

Front-end events page
When developing the events page, My main focus was to develop a user interface that would catch the users attention.
Following the high fidelity produced for the first part of the coursework facilitated this process.
I created similar design but strictly ensuring that it followed gestalt's principle of similarity and proximity as each event item with similar attributes are placed and perceived together.
As part of the event page I added a video which will auto resize to fit all devices, when navigating on the events page on IPad, the user can view the video and minimise it to continue navigating.

Creating events item pages
The events page which will have a gif running in the background is an important feature for a successful UX as it will set the mood of the page, having a gif running will be very eye catching feature for users and will encourage them to keep reading about the event items.
Each event item implemented has an auto sizeable map with the POI of the given event.
If the user likes the specific event, I implemented a mailto function where the user can press and it will take them to a mail with an auto generated subject title and a body which will link users to this specific event.
These features has been repeated throughout all different event items to maintain the design.

Creating the custom_css
I created a custom_css file which followed the style guide created in the first part of the coursework and implemented the changes given on feedback, the importance of the custom_css file was to create it so all students developing their parts would find it useful and it would remain consistent and fully functional,.
The css file respected the typography, colours and style for all devices to ensure legibility on the device used by user, it impacts a lot of the objects in the application.
This was one of the first steps done when developing the application, and which has been added too and modified to satisfy everyone's need.

Merging the files and fixing bugs

Upon completing of the project, I decided to merge the files together into one big file where the applications would run smoothly.
When merging, there appeared to be a few bugs which had to be taken care of for the application to run.
Merging creates a complete picture of the applications and all pages runs the functions as intended and the href is consistent with the files within the folder.
Some specific inline styling was needed to adapt to the landscape view.

Group_6_Std_3_iPad_Portrait
Steven Siht – W1616792

Built navigation bar with navigation menu panel
When building the navigation bar and navigation menu panel I tried to keep things as simple as possible. The button that opens up the navigation panel is styled as a “hamburger icon” which is used all across the internet so it is known for every user. Clicking on the button opens up a panel that contains all the menu items. Navigation bar also includes the logo of the fashion week.

To make things happen I used prebuilt jQuery Mobile functions and elements which made it easy to implement and guarantees that everything works on all the different devices available.

Built and Enhanced News Page
When user enters the news page, they will see previews of different news items ordered from the latest entry. The user sees a picture that goes with the news, the heading and a few sentence preview of the context of each news item. To lead the user to open up the news item I made the “Read More” button green as it will attract attention and “Add favourites” button is coloured red so all the interaction points for the user are clearly visible.

Built news item pages
When user opens up a news item they will see a 100% width picture of the selected item, under that a header written in bold and uppercase letters, the date of the article and content. I have created the page with light background and dark text so that it would be easy for the user to read. I also added line-height and letter-spacing to make it easier to read.

One of the news items that talks about different events taking place also contains a Google Maps API that shows the location of all the events described. I used Google Maps API to implement the map and added all the interest points manually.

Built QR Reader
As proposed in the initial idea our web app contains a QR reader to get quick information about different interest points, news, tickets etc. The implementations of the QR reader is very simple. When the user navigates to the QR page, he/she has to allow the web page to gain access to the camera of the device that they are using and then just scan the QR code and they will be automatically taken to the target-page of the QR code.

For the implementation of the QR reader I used pre-built JavaScript library that creates a canvas element on the page that shows the view of the camera.

Group_6_Std_4_iPad_Landscape
Cyril Asomani – W1544390

Built Favourites Page
When user has logged into the site they are able to view the events, designers & news items that they have favourited from previous pages. This page was made through utilising the tabs provided by the JQuery mobile web framework in order to create the different categories of favourites that the user has thus simplifying where to look for their favourite item on the page. Each favourite item has an “Open” button which will direct the user to the original page previously visited from the user & an “Email” button which contains a mailto attribute which opens up and email terminal and allows users to share the favorited item through email.

The IPad Landscape version will display three team members one a single screen while having to scroll down to view the other team member while two users display on the screen for the portrait view for the IPad.

Implemented Dynamic Favorites Page
Javascript was used to access the local storage of a users browser. The code will save the name of the event or team member to the users browser & then will be accessed and displayed on the favourites page under each unique tab on the favourites page.

Built About Us Page
When the user clicks on the About Us page the user is able to view a picture of every team member along with their name & role. The Email button will open an email terminal with a specified subject and email address unique to each team member thus allowing users to contact us to provide feedback about the website.

This page was made possible through using customly created CSS for each team members section on the page. The mailto attribute was used to allow users to communicate with each team member within the website.

Bug Fixing throughout implementation
Throughout the implementation, I would conduct bug testing to test the application and to ensure all functions and scripts was fully functional.
Once finding a bug, I would immediately look at where the code went wrong and began to fix all bugs.
The main bugs to fix was the Javascript as it had to be fully functional for the pages to open smoothly, Another mayor bug I had to fix was the layout of the pages in Ipad landscape mode as there is a wider screen compared to the Iphone landscape.
Ipad landscape needed a different layout that would adapt and change if the user were to change the orientation of the device.
All bug was either found by my fellow team members and reported to me or I would find it and then implement the changes.