

CMP3035M: Cross-Platform Development

'VEGAN FINDS' APP

GABRIELLA DI GREGORIO | 15624188

EXTENSION AUTHORISATION CODE | 5LDCH5LFGLTHDQ83

Contents

App QR Code	1
Background & Concept	2
Overview	2
Competitors	2
Vegan Maps	2
TripAdvisor	3
Requirements.....	5
Persona	5
Usage Scenario	5
User Stories	5
Core Features	6
Enhancement Features	6
Design & Prototyping	6
Development.....	11
Reflection	17
References	19

App QR Code



https://expo.io/@gabby2805/cpd_dig15624188

Background & Concept

Overview

Vegan Finds is an app specifically tailored for vegans to help you find delicious vegan options in any restaurant or food-place. Eating out with friends is a common difficulty that vegans face so the aim of this app is to allow a community of vegans to help each other find their best options in any kind of restaurant. There are apps out there that help to find vegan eateries, but as vegan food has become more popular, many 'regular' restaurants and pubs are providing great vegan options. This means that vegans no longer have to be so restricted when dining out, but it can still be difficult to find out what options are available. With this app, vegan-friendly dishes that fellow vegans have discovered can be collated in a single, handy place to help you choose where next to dine with your friends and family. No longer do you need to be restricted to vegan-only places or be reliant upon unreliable search filters on popular apps like TripAdvisor that may allow a place to advertise as 'vegan-friendly' when that actually means they offer a side salad. Not only will this app help you find tasty new options to try, but you can also share your delicious discoveries with others to help the ever-expanding vegan community grow and flourish. This app is a unique and novel idea that can revolutionise the way vegans can enjoy dining out.

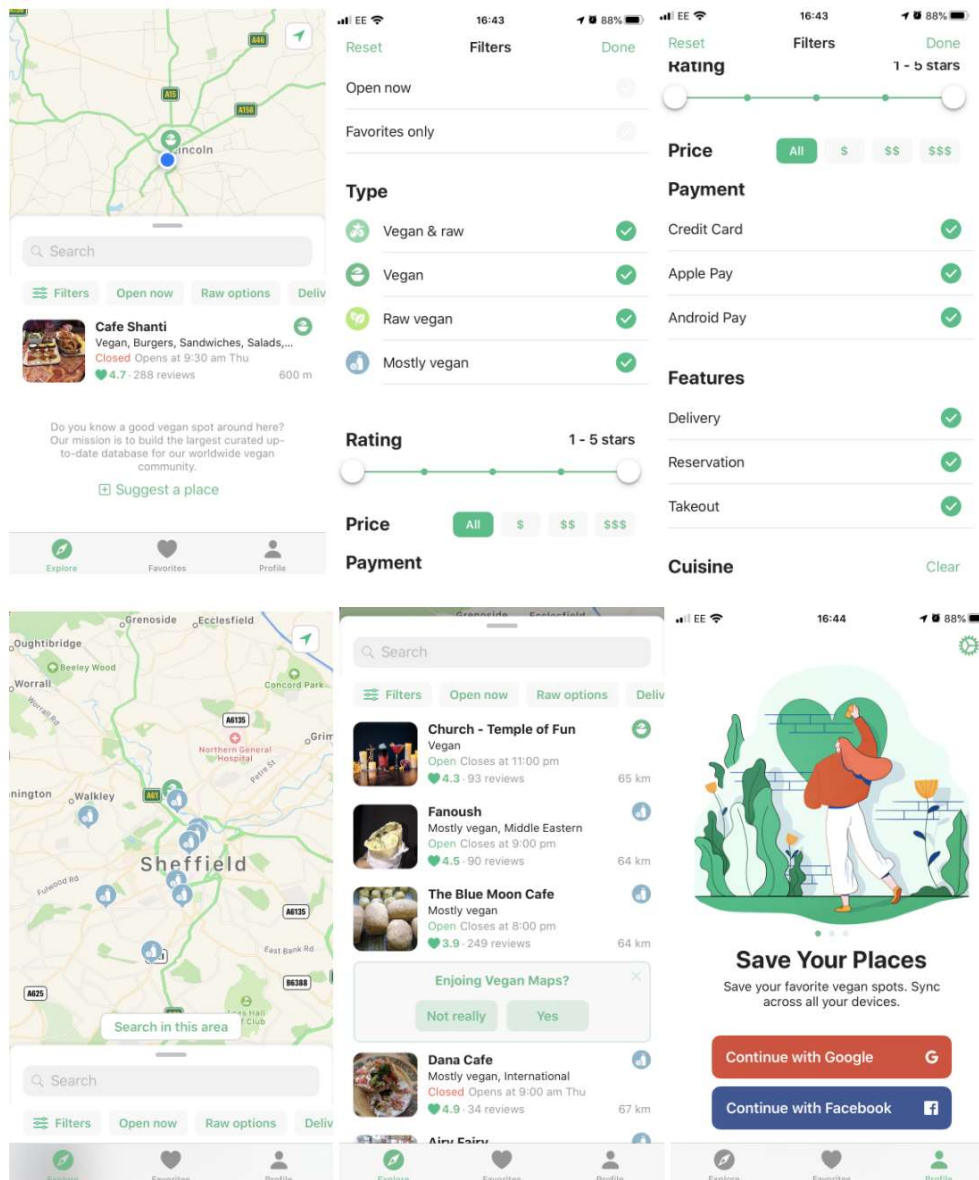
Competitors

Finding potential competitors was somewhat of a difficult task. The idea of this app is a fairly novel concept so there are few similar existing apps, many of which are locked behind paywalls. Furthermore, when searching for 'vegan' on the app store, the majority of results relate to recipes and barcode scanners which highlights the very need for this app, as vegans often find themselves having to eat at home.

Vegan Maps

Vegan Maps is a free app available on the Apple App Store that utilises location services to help find nearby vegan restaurants and other eateries. Whilst the purpose of this is clear, as vegan places can often be difficult to find due to being small, independent and unpopular amongst the masses, it does not help vegans to find where they can enjoy with their non-vegan friends. However, it does offer some flexibility by allowing you to filter by strictly vegan or 'mostly vegan'. The screenshots taken whilst testing and assessing this competitor can be seen below. From testing, the location services seem accurate and successfully display nearby vegan restaurants and cafes. However, it was tested in two city centre locations; Lincoln and Sheffield, and whilst there were a variety of places found in Sheffield, only one was found in Lincoln. This suggests that the app may not be useful in many locations as there may often be no nearby vegan eateries. The map feature is useful to help you see the exact location of the results however it had some usability issues such as the gesture to close the menu being too close to the top of the screen, resulting in the iOS phone menu being pulled down instead, which was frustrating. Furthermore, a glitch was frequently encountered where the keyboard would get stuck over the top of the map, making it difficult to see and use. Another feature that this app offers is the ability to favourite places so you can easily find them later. This is a useful feature that can help you build a list of the places you like or want to try. However, in order to use this feature, you have to log in with a Google or Facebook account and create a profile, which may deter some users. When a place is clicked on, more details are shown such as their opening hours and address. As well as this, reviews are linked from TripAdvisor. Since TripAdvisor is likely the most used source of reviews, this likely means that as many reviews can be provided as possible. However, it also means that the reviews found may not be specifically by vegans or to vegans, and therefore may be somewhat unhelpful. Lastly, a big success of this app and the key inspiration taken to move forward with this project, is the design. The UI is very clear, uncluttered, and professional-looking. It matches

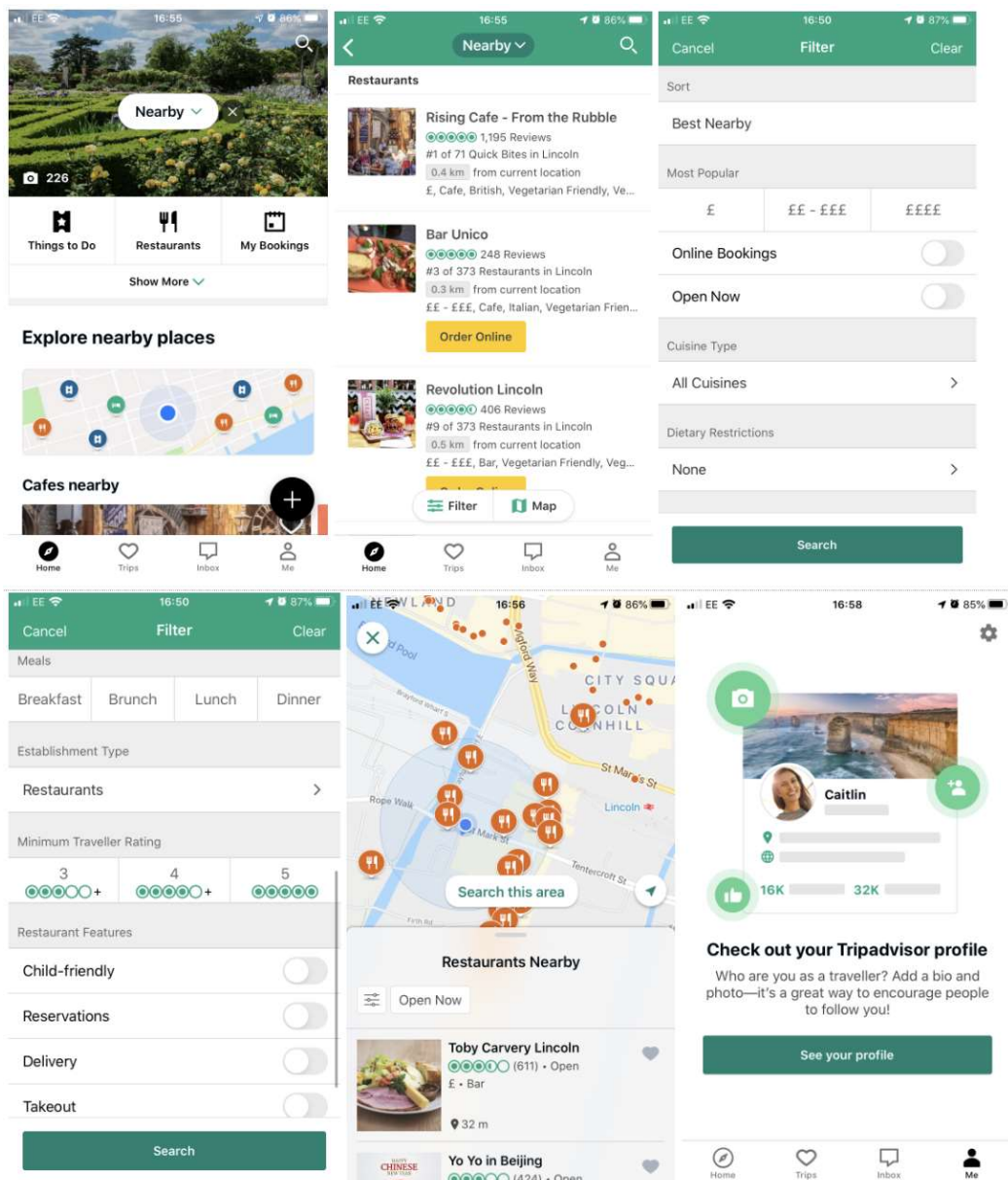
the design of iOS well by using similar font types and icons, so it does not look out of place on an iOS device. Furthermore, it uses subtle hints of colour and these are green which often springs to mind when the word 'vegan' is heard, due to many plant-based foods being green and veganism being environmentally friendly, also known as 'green'. These design concepts will be carefully considered when designing and developing the app for this project.



TripAdvisor

Whilst the extremely popular app TripAdvisor is not very similar in concept to the proposed app idea, this giant is a competitor to any other resource with the aim to help people find places to eat. For this reason, it was worth exploring to ensure that the proposed app could offer a unique edge, not already provided by TripAdvisor. TripAdvisor is a free service in which users can read and write reviews about various locations, including hotels and restaurants. As the services relating to eateries are the only ones relevant to this project, only these have been tested and analysed and the results of doing so can be seen in the screenshots below. TripAdvisor is clearly extremely successful overall, as can be seen by a 4.5 rating from 48,000 reviews on the app store, and even small independent restaurants often having hundreds of reviews inside the app. However, the aim of this research is to analyse how useful it is to vegans specifically. Whilst the app does offer the ability to filter by dietary requirements, one

of which being vegan, experience has shown that this filter is often unhelpful and inaccurate. It is evident that for a restaurant to be listed under 'vegan-friendly' or 'vegan options available', they do not actually need to offer a variety of genuinely vegan options. In many cases, these places may have a single, restricting vegan option or none other than a side salad. Furthermore, as only the entire place can be reviewed and not specific dishes, it is not helpful in discovering vegan options. As well as this, the ability to search for the word 'vegan' in written reviews, often presents you with the unenlightened opinions of others that feel that they can say that there are indeed vegan options as they have seen salad on the menu. These issues make it clear that an app specifically tailored for a community of vegans that can successfully provide accurate vegan recommendations to one another would be far more helpful for vegans. Nevertheless, it was interesting to explore TripAdvisor in order to gain an understanding of successful design. Similarly to Vegan Maps, TripAdvisor provides a clean, neat and professional interface with hints of green. As well as this, both apps used a bottom navigation bar, suggesting that this may be the most popular and most user friendly navigation design and will therefore likely be implemented in the app-to-be.



Requirements

Persona

Juniper is a 22-year-old student who has been a vegan for the past year. She is a socialite and lives in University housing and has made lots of friends through her flatmates and course mates who she enjoys various outings with. She's not particularly tech savvy but she always has her phone on her so she can talk to her friends, and check and upload to social media. Although she feels very passionately about veganism, her family worry for her health and well-being and her friends struggle to find places to go out to eat with her. She loves to try new things and she will be travelling during a gap-year starting in the summer and is eager to try new cuisines. Although she loves the food she cooks for herself, she can sometimes be a picky eater and easily get bored of eating the same foods.

Usage Scenario

Juniper has just had a fun day shopping with her friends, and they've worked up quite an appetite so are looking for somewhere to have a bite to eat. Her friends don't want to go to the vegan café, but they aren't sure where else Juniper will be able to find something suitable on the menu. After walking up the High Street browsing window menus and seeing nothing but salad options, June pulls out her phone to find somewhere nearby to save them from aimlessly walking around any longer. She then comes across a picture of a delicious looking pasta dish, rated five stars by another vegan, that she could eat at a restaurant close by. While her friends tuck into burgers and Bolognese, Juniper enjoys the meal she found the recommendation for. As they chat over dinner, her friends realise that being vegan isn't so tough and that, after stealing a bite, vegan dishes can be really tasty and interesting, and aren't only found in 'hippie' cafes. Since they all enjoyed their meals so much, they decide to treat themselves to dessert. June tries the vegan cheesecake from the menu and is delighted, it's the best cheesecake she's ever tasted! Before taking another bite, she pulls out her phone to upload her pretty dessert on social media. She then realises that she can share this find with other vegans, so they can come to try it too. Now that finding vegan dishes has never felt easier, she looks forward to finding new foods on her travels in the Summer.

User Stories

As a relatively new Vegan, I want to discover new dishes so that I can have a balanced and interesting diet.

As a socialite, I want to easily be able to dine out with my friends so that we can enjoy doing things together without feeling restricted.

As someone who always has my phone on me, I want to be able to instantly upload the pictures I take using my phone camera.

As someone who is not a confident user of technology, I want a simple and easy to use interface, with only the core functionality I need, so that I don't feel confused and can quickly and easily achieve the task I want to do.

As a vegan with concerned friends and family, I want to prove that I can eat anywhere I like and enjoy tasty meals just like them so that they are more willing to try vegan foods too.

As a vegan with no vegan friends, I want to feel part of a wider community so that I can feel supported and help to encourage others in this good, sustainable cause.

As someone who will be travelling, I need to be able to rely on my phone so that I don't need to buy a laptop to bring with me.

As a traveller eager to find new cuisines, I need to be able to find vegan recommendations unrestricted by locations so that I can see what other vegans have recommended in other countries, supporting the worldwide vegan community.

As a relatively picky eater, I want to be able to see what a dish looks like before I try it and be able to browse from various options so that I can avoid eating the same boring foods or things I don't like.

As an enthusiastic cook, I want to discover new dishes so that I can try to recreate the tastiest ones at home.

Core Features

Based on the Persona, Usage Scenario, and User Stories above, users of the app must be able to...

- Upload pictures directly from their phone
- Write reviews of specific vegan dishes that they can share with other vegans
- Browse from a list of vegan finds and view information such as the star rating and the restaurant name
- Browse unrestricted by location or eatery type, i.e. not just vegan-specific places
- Interact with a user-friendly, simplistic interface without requiring advanced technical knowledge
- Understand that the app is clearly aimed at vegans

Enhancement Features

Based on the Persona, Usage Scenario, and User Stories above, it would be beneficial if users of the app could also...

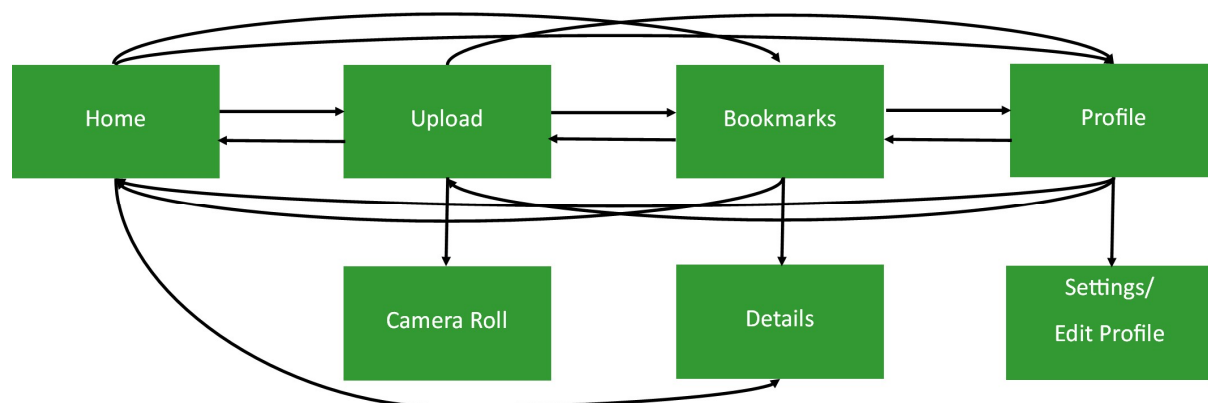
- Bookmark/favourite recommendations they see so that they can easily find them later
- Filter the recommendations based on various conditions such as location, date and rating
- Create a profile so that they can better interact with the community and potentially get to know other vegans
- Use geolocation to automatically search for the closest recommendations
- Be able to see more details when a recommendation is clicked on, such as the exact address or even a map

Design & Prototyping

With the requirements engineering phase complete, the next stage of the Software Development Life Cycle (SDLC), the design stage, is carried out. The purpose of this phase is to decide upon the structural and aesthetic properties of the app. In order to develop this app in a user-centred fashion, this stage was carried out by considering the five planes of user experience (UX). This is the concept that there are five distinct layers of UX design, ranging from abstract to concrete. These five planes help to break down the design process into steps that reflect the different levels of abstraction involved in the design. It starts with a basic concept/idea and builds up to the visual design aspect. The first two planes have been largely covered already, during background research and requirements engineering however it is crucial to reflect upon these in order to create carefully considered designs that match the defined specification. The first plane is the strategy plane which considers the reason for the system, why it is being created, who it is being built for, why people are willing to use it, and why they need it (Elgabry, 2016). As defined in the app overview, the app will be a resource specifically tailored to vegans to help them discover great vegan options available in eateries of any kind and also recommend dishes they have found to help the vegan community. Throughout the design phase, it is important to keep this in mind at all times to ensure that the correct product is being designed and

therefore the correct product will be built. The second plane is Scope which is the stage where the requirements are defined. This is important because the requirements can serve as a checklist to ensure that every feature has been incorporated into the design. As well as this, it can help to avoid 'scope creep' which could result in new ideas being implemented taking time and attention away from the core, original requirements. The requirements have already been defined in the previous section and will serve as a basis to create designs upon.

The third plane is the structure plane which considers how users get to each page and where they could go from it. This stage of the design phase is useful for deciding which pages must be designed and created, and the necessary navigation features to navigate between them. The structure helps to define how users interact with the product, how the system behaves when a user interacts with it, and how it is organised and prioritised (Elgabry, 2016). During this stage, the pages/screens the app, the content to be displayed on each, and navigation features were decided upon. The best way to visualise this was to create a diagram which shows the different screens the app will have, how users can get to each of them, and where they can go from them. The diagram below shows the general structure of the app being designed. The four screens at the top; home, upload, bookmarks, and profile can all be accessed from each other, it is clear that these should be implemented into a static navigation feature to appear on all pages, so they can be easily accessed from anywhere. This also helps to decide the features to be implement on each screen, such as camera roll access being required on the upload page. However, this design lacks detail and is not easy to visualise what the app will look like and therefore it is clear that more detailed designs are now necessary to build on this fairly high-level concept.



The next stage is the skeleton plane which adds more clarity by determining the visuals, presentation, and arrangement of elements. In other words, this is where the user interface begins to take shape and interactions are more carefully considered. The skeleton plane considers three key design elements; interface design, navigation design, and information design. Interface design refers to how elements will be arranged to enable users to interact with the system, navigation design refers to where navigation features will be and how users can get from one part of the site to another, and information design refers to the presentation of information. Therefore, all three of these design concepts will be considered and incorporated into designs created at this stage. The skeleton plane helps to establish what the site will look like, how content will be arranged, how interactions will be presented, how can users navigate, and how information will be presented clearly. A popular design tool during this stage are wireframes. A wireframe is a static sketch or mock-up that represent the visual format of the product and the layout of the content. Wireframes were chosen as the design method for stage of the app design because they allow potential problems or oversights to be caught and managed early, and they are much easier to change than code. Furthermore, they can help in

prioritising content since a elements of core functionality and related content features should be prioritised before potential screens deeper in navigation and less crucial to the functionality of the system, considered enhancement features. The image below shows the wireframes created. The initial design of the home-screen shows the potential of allowing users to do many things; navigate a map, use a search bar, and browse a list of restaurants. The initial idea was to display restaurant information and then extract reviews from an external source such as TripAdvisor. However, further planning came to the realisation that generic restaurant descriptions and reviews would not be specific and helpful enough to vegans, therefore identifying an error in design and the need for further designing. The next page shows how these posts are submitted, which is where users can create an upload. This shows that users can enter where they ate, upload a picture of what they ate, and give a review of their experience in the form of a star rating and a comment. The concept of being able to upload a picture of the meal sparked the idea of the app being based around specific dishes rather than restaurants. This way, it can help to make it clearer to vegans what exactly they will be able to enjoy as being recommended a restaurant may night be helpful if their vegan options are very limited. The next page presents the idea of being able to favourite/bookmark posts to easily find them later. This can be useful to build a list of places/dishes to try rather than having to try to remember them or search for them again. However, it is evident that the way of favouriting posts has not been made clear in these designs, as a button to do so would need to be incorporated. The last wireframe is of a users' profile which would show information about them and posts that they have submitted. However, further consideration revealed that a profile system may not be necessary as requiring users to sign up to access core features of the app can be a deterrent. Furthermore, asking for people's information and allowing profile pictures to be uploaded can cause privacy, confidentiality, and security concerns. Whilst these designs were helpful in developing ideas, it is clear that a more revised design should be created before development takes place.



Before refining the design, user feedback was gathered based on the wireframe designs in the form of a system usability scale (SUS). The SUS was originally created by John Brooke in 1986 and provides a “quick and dirty”, reliable tool for measuring the usability. It consists of a 10-item questionnaire with five response options for respondents; from strongly agree to strongly disagree. SUS has become an industry standard, with references in over 1300 articles and publications. The noted benefits of using SUS include that it: is a very easy scale to administer to participants, can be used on small sample sizes with reliable results, and is valid – it can effectively differentiate between usable and unusable systems

(usability.gov, n.d.). Three participants were presented with the wireframe designs and then asked to complete an online questionnaire which consisted of a predefined set of questions required to calculate the SUS score, along with an optional section for participants to explain their answers and give additional feedback in order to gather both quantitative and qualitative data. In order to calculate the SUS score, 1 is subtracted from the responses of each odd numbered question (user response – 1) and the responses to each even numbered question is subtracted away from 5 (5 – user response). Then, these are all added together which gives a total score ranging from 0 to 40, with 0 being the lowest usability and 40 being the highest. To get a score out of 100, this result is simply multiplied by 2.5. Whilst only three responses were gathered, the SUS method is effective and reliable even with small sample sizes and also meant that any written feedback provided could be more carefully analysed and considered. The first participant gave the app design an SUS score of 82.5, the second resulted in a score of 85 and the third participant gave a rating of 95. These scores resulted in a mean SUS score of 87.5 out of 100, indicating that the designed app seemed highly usable. The table shown below in Figure 1 shows that any SUS score above 80.3 is equivalent to ‘A Grade’ or ‘excellent’ usability, therefore suggesting that app had been successfully designed with user experience in mind.

SUS Score	Grade	Adjective Rating
> 80.3	A	Excellent
68 – 80.3	B	Good
68	C	Okay
51 – 68	D	Poor
< 51	F	Awful

Figure 1: SUS Scores Interpretation (Alathas, 2018)

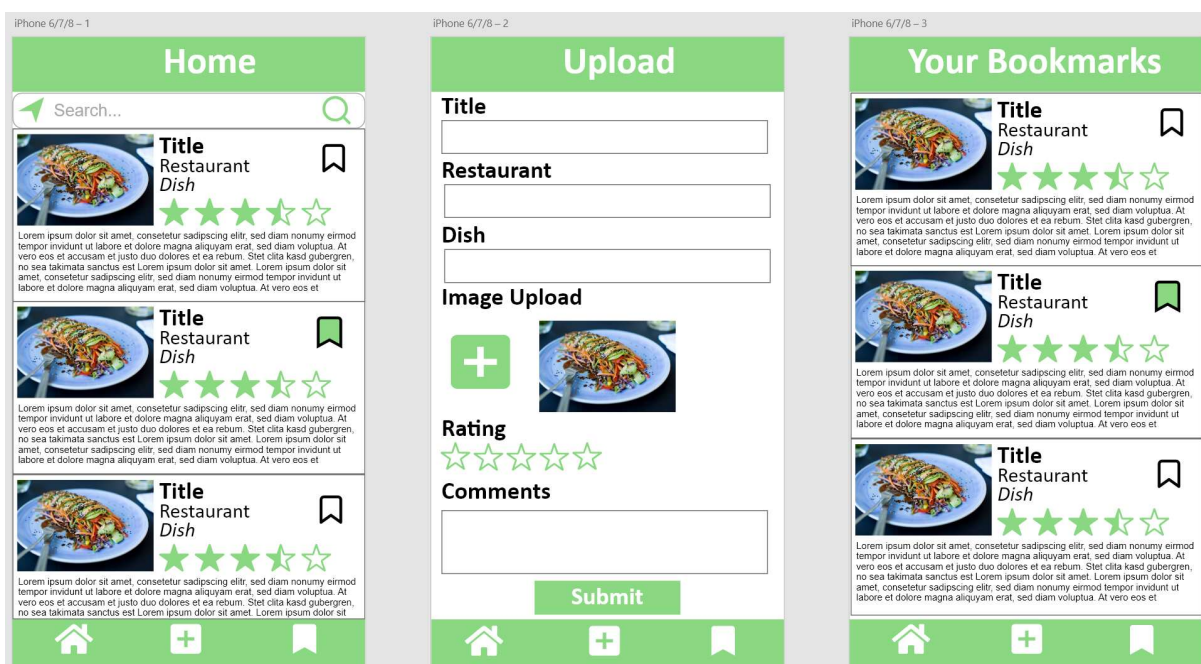
However, although the SUS score may suggest a near-perfect design, it is clear that there is potential room for improvement and the qualitative data could potentially aid in understanding this. Feedback gathered that the designed bottom navigation is an appealing design based on responses such as “the navigation is clear and not too complex”, “everything is where you'd expect it to be, navigation bar is clear and icons are self-explanatory”, and “navigation was easy”. Based on this positive feedback, it was decided that the bottom navigation bar concept would remain consistent in further designs. However, some constructive criticism such as “it can feel a little cluttered (home page)”, “I would argue the first screen may be too cluttered”, and “on a small phone screen it might be hard to actually navigate around the map or navigating through the restaurants with the small buttons” led to the realisation that the home-screen needed redesigning as it was too cluttered and incorporated too many features. Therefore, it was decided that the map feature was not necessary as it cluttered the homepage and may have resulted in a poor user experience. It is easy to find the exact location of a restaurant by using very popular third-party apps that dominate the market and therefore was not a necessary feature to tackle in this app. As well as this, the comment “there are some consistency flaws” revealed that the overall aesthetic was not clearly consistent and some buttons on some pages, such as being able to add posts from the profile page as well as the upload posts screen, were not necessary and somewhat illogical which could be confusing for less technologically experienced users. This was also backed up by another comment that “people who have little technical ability may take longer to learn how to use the app”, suggesting that usability and accessibility needed more careful consideration.

The final stage of the five planes of user experience is the surface plane. This final layer is where the visual features of the system are fully established. In order to decide upon design features such as colour schemes, fonts and imagery, a moodboard was created to spark these ideas. A moodboard is a collection of assets and materials intended to communicate the style, voice, direction, and language of a particular design or project (Gidalevitz, n.d.). An advantage of using moodboards is that they are quick and easy to create so they can easily be changed if necessary and can present various options and comparisons to be able to choose between more clearly. They act as a good foundation in order to move on to the next stage of design and prototyping. As well as this, they encourage creative inspiration as they can be an enjoyable 'warm-up' activity to get ideas flowing and spark the motivation to create more advanced designs. A colour palette can often be decided through the creation of moodboards as similarities between elements can be spotted and therefore frequently occurring or well-matched colours can be identified. As can be seen in the moodboard below and the screenshots of competitor apps, it is clear that green is a relevant and suitable colour theme in relation to veganism. This process of creating a moodboard helped to find colour codes for shades of green that could be noted and used during development, thus speeding up the development process.



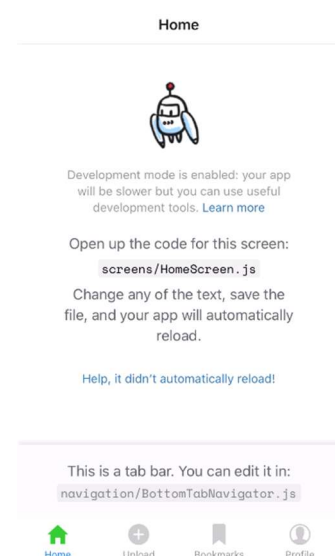
To complete this final plane, a more advanced design was created in Adobe XD which incorporated the themes and ideas gathered from the feedback of the wireframes and the styles identified from the moodboard. Rather than using placeholders to represent the general layout, more specific elements such as images, icons, buttons, and text are used here. Many features of the effects and interactivity are decided at this stage, but more advanced capabilities may be discovered and implemented during development than can be represented in a design tool such as Adobe XD. The advantage of making an interactive prototype design at this stage not only establishes the static aesthetics of the finished site but also represents some of the basic navigation and interactivity features such as how users can get from page to page and what happens when a button is clicked. The image below shows a screenshot of the interactive prototype created and from this it is clear that the design is now much more established than in the wireframes. Real images and icons have been used, and whilst only a limited selection is displayed, it makes the overall aesthetic much clearer than placeholders. Whilst the visuals have been improved in this design, the features of functionality have been simplified to ensure that the expectations are realistic and can be met in the required development timeframe. Furthermore, based on the user feedback, the design has been made much

less cluttered and easier to use. The home screen now acts as a feed in which the latest or most popular posts are displayed, and all of the relevant information about a post is displayed as no further details are necessary. Furthermore, the number of buttons on each page has been kept to a minimum as the bottom navigation bar provides adequate navigation between all parts of the app, making it simpler to learn and use for those less technologically advanced. As well as this, the profile screen has been removed in order to avoid deterring users from submitting their recommendations because they have to log in. Implementing a signup/login system would have been an inefficient use of time since the profile feature is not necessary and may be more detrimental than beneficial due to privacy and security issues. Moreover, some accessibility considerations have been made. Few and large buttons make the app easier to use for those that may suffer with motor disabilities. As well as this, large font and no big chunks of text make the app accessible to those with poor eyesight, dyslexia, and some forms of autism. The final design is now complete, and development can take place with a clear idea of what the finished app should look like.



Development

To begin development, a new Expo project was created using the navigation template in order to save development time by having a basic navigation system in place to adapt appropriately. Four blank screens were set up and whilst only two were core to the requirements, it established a navigation bar and page linking and any unused pages could be easily removed later. A static bottom navigation bar was implemented as designed as this allows for a positive user experience due to being reachable and easy to use. It was important to ensure that the home screen is accessible from all parts of the app since making it easy to get back to the homepage is one of Google's 25 mobile UX design best practices (Google Developers, 2015). Figure 2 shows how when a phone is being held in either hand, the bottom portion of the screen is reachable with the thumb whilst the top is not. For this reason, having a navigation feature at the bottom of the screen, where users can comfortably reach, was the best design choice and helps to meet the requirement of users



being able to interact with a user-friendly interface. As well as this, the hint of green was added to the navigation bar since the background research and design process highlighted that green is a colour that relates to the vegan theme. This helps to meet the requirement that it should be clear to users that the app is aimed at vegans. However, only a hint of colour was used since overly colourful interfaces can make an app look unprofessional and overwhelming to users.

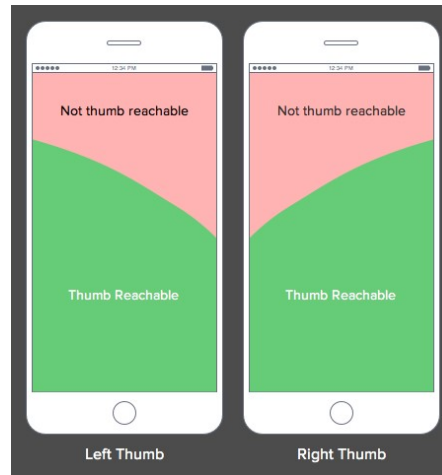
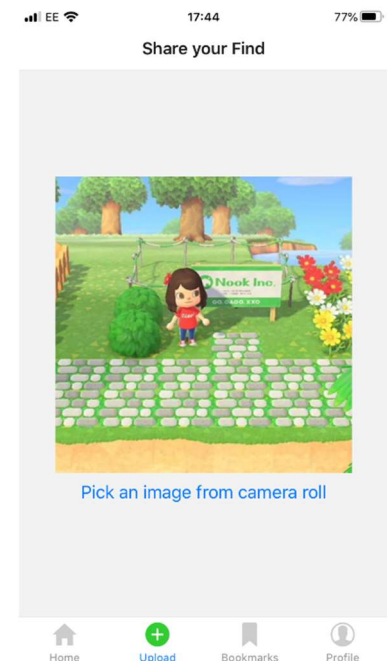


Figure 2: Reachable screen area of Phone Screen with Thumb (UX Movement, 2017)

Next, since a key feature of the app is to allow users to upload images directly from their camera rolls and this was a core requirement, this was chosen to be the first feature to be implemented. So, the expo-image-picker module was installed and used to access images stored on the device and upload it to the app. An image picker was chosen over a camera feature as this is more likely to result in higher quality images, due to utilising the full capabilities of the devices camera feature rather than being restricted by a widget. Furthermore, user may not want to take a picture for use in the app only, they may want to keep it in their photo library or upload it to social media platforms. As well as this, it means that the app is not rendered useless in a place with no internet connection. Users could take a picture of their vegan meal in a remote countryside pub, then upload their picture to the app when they reach a location with internet access. In order to implement this feature, the first thing to do was to ensure that the app could be granted permission to access the camera roll. This was done using the code snippet shown below in Figure 3. Next, the code shown in Figure 4 was used to select an image from the camera roll on the device's local storage. The media type is specified to images as videos would not be appropriate for this app, and the user can slightly edit the photos they upload by selecting the area they will be cropped to. As well as this, the images are imported at high quality and in a square shape. The results of a successfully selected image is shown on the right. The preview of the image that the user has selected is shown in order to Nielsen's Usability heuristics of 'visibility of system status' and 'recognition rather than recall' (Nielsen, 1994). This makes it clear to the user that an image has been selected and they can check if it is the right picture that they intended to upload rather than having to remember. The addition of base64 was made later in development in relation to the development of the home screen and will therefore be discussed later. This implementation met the requirement that users will be able to upload pictures directly from their phone. Of course,



the upload screen needed further development as an image alone without context would not be useful to other users.

```
getPermissionAsync = async () => {  
  if (Constants.platform.ios) {  
    const { status } = await Permissions.askAsync(Permissions.CAMERA_ROLL);  
    if (status !== 'granted') {  
      alert('Sorry, you must grant camera roll permissions in order to do this.');    }  
  }  
}
```

Figure 3

```
_pickImage = async () => {  
  try {  
    let result = await ImagePicker.launchImageLibraryAsync({  
      mediaTypes: ImagePicker.MediaTypeOptions.Images,  
      allowsEditing: true,  
      aspect: [3, 3],  
      base64: true,  
      quality: 1,  
    });  
    if (!result.cancelled) {  
      let selectedimage = `data:image/jpg;base64,${result.base64}`;  
      this.setState({ image: selectedimage });  
    }  
  
    console.log(result);  
  } catch (E) {  
    console.log(E);  
  }  
};
```

Figure 4

Now that images could be successfully uploaded from users' devices, the next step was to create a form on the upload page in which users could enter more details about their images, creating useful contextual recommendations for other users to see. The early stages of this can be seen on the right, which shows fields for basic, important information such as the restaurant and dish in order to give the image being uploaded context. Whilst this provides core functionality, it is clear that styling is necessary for a more positive user experience, since white text on a light background would be very hard for users to see. Furthermore, some sort of rating system needed to be implemented so users can see whether a dish is being highly recommended or perhaps being advised to avoid it. The code snippet shown below in Figure 5 shows the code used to display the completed form. The button to upload an image from the camera roll was changed from a line of text to a '+' icon as a line of text is not usually associated with a click/press interaction whereas a button is. Furthermore, the symbol of an addition side is an affordance which suggests that users are able to add/upload something. This adheres to Nielsen's heuristics of 'match between system and the real world' and 'aesthetic and minimalistic design' as users will clearly be able to recognise the purpose of the button and avoid having to read instructions. As well as this, a star rating system was added in which users can tap empty stars to make their desired number of filled stars. The star rating system has become very popular and therefore should be easily recognisable and

Share your Find

Title

Restaurant

Dish

[Click to select an image from camera roll](#)

Add

Home Upload Bookmarks Profile

understandable to all users. This system makes it clear at a glance whether the post is recommending a dish highly to try or recommending avoiding it as it was poor. Furthermore, the colour scheme was added to this star system to ensure consistency throughout the design without overwhelming colourfulness. Lastly, a comment field was implemented so that users could give additional information. All of these fields bring up the keyboard when they are clicked, and the keyboard hides when the user presses return, so that they can easily go back to seeing the full screen. This adheres to Google’s advice of ‘streamline information entry’ as generally, the fewer taps the user must perform, the better (Google Developers, 2015).

```
<ScrollView style={styles.container}>
  <Text style={styles.title}>Title</Text>
  <TextInput style={styles.itemInput} value={this.state.Title} onChangeText={Title => this.onChangeTitle(Title)} />
  <Text style={styles.title}>Restaurant</Text>
  <TextInput style={styles.itemInput} value={this.state.Restaurant} onChangeText={Restaurant => this.onChangeRestaurant(Restaurant)} />
  <Text style={styles.title}>Dish</Text>
  <TextInput style={styles.itemInput} value={this.state.Dish} onChangeText={Dish => this.onChangeDish(Dish)} />
  <Text style={styles.title}>Image</Text>
  {image && <Image source={{ uri: image }} style={{ width: 300, height: 300, alignSelf: 'center', paddingBottom: 10 }} />}
  <Ionicons name="ios-add-circle" color= '#32CD32' size= '50' style={{ textAlign: 'center'}} onPress={this._pickImage} />
  <Text style={styles.title}>Rating</Text>
  <StarRating
    disabled={false}
    maxStars={5}
    fullStarColor={'#32CD32'}
    rating={this.state.Stars}
    selectedStar={(rating) => this.onStarRatingPress(rating)}
  />
  <Text style={styles.title}>Comments</Text>
  <TextInput style={styles.itemInput} value={this.state.Comment} onChangeText={Comment => this.onChangeComment(Comment)} />
  <TouchableHighlight
    style={styles.button}
    underlayColor="white"
    onPress={this.handleSubmit}
  >
    <Text style={styles.buttonText}>Upload</Text>
  </TouchableHighlight>
</ScrollView>
```

Figure 5

With the form complete, to input information now needed to be stored. As the posts submitted are to be visible to all users of the app, the logical choice was online storage rather than local storage. There are many advantages of online storage over local but the fact that data stored in online storage can be easily accessed on any device with an internet connection is the reason it was chosen for this app. The online storage solution chosen was a Firebase Realtime Database which is an efficient, low-latency solution for mobile apps that require synced states across clients in real-time (Firebase, 2020). Whilst the Firestore is Firebase’s newer solution, there are still advantages of using a Realtime Database. For example, data is stored as one large JSON tree so simple data is very easy to store (Firebase, 2020). Furthermore, the low-latency of the Realtime Database means that posts will be uploaded quickly, avoiding users to be frustrated or wasting large amounts of their mobile data. Database uploads were implemented by importing Firebase into the app and creating a config file to store the database connection information. Then, variable states were set based on the information input into the form fields. On the press of the submission button, these states are pushed to the database, as can be sampled in the code shown below in Figure 6. A sample of the data stored inside the database can be seen in Figure 7. With this part of the implementation complete, the requirement to be able to ‘write reviews of specific vegan dishes that they can share with other vegans’ had now been met. When the submit button is clicked and data is sent to the database, the user is presented with a popup message informing them that their post was successfully created in order to adhere to Nielsen’s heuristic of visibility of system status. It was decided that all users would be able to create posts without having to create an account and login for several reasons, including the fact that having to sign up to something can often be a deterrent and would have likely resulted in less users being

willing to share their recommendations. The more vegans submitting suggestions, the more beneficial for the community. This point is backed up by Google's tip to allow users to explore before they commit as study participants were frustrated by sites that require upfront registrations to view the site, especially when the brand was unfamiliar. Although customer information may be integral to a business, asking for it too early may result in fewer registrations (Google Developers, 2015). Furthermore, no personal information was necessary to gather for this app, therefore the creation of a sign up and profile system would have raised unnecessary concerns regarding the privacy of users and the security of stored information. As well as this, this saved time during development as the core functionality could be implemented without needing to first develop a sign-up system which may have been a lengthy and complex task.

```
handleSubmit = () => {
  db.ref('/Posts').push({Title: this.state.Title, Restaurant: this.state.Restaurant, Dish: this.state.Dish, Image: this.state.image, Stars: this.state.Stars, Comment: this.state.Comment});
  Alert.alert('Post saved successfully');
  console.log(this.state);
};
```

Figure 6

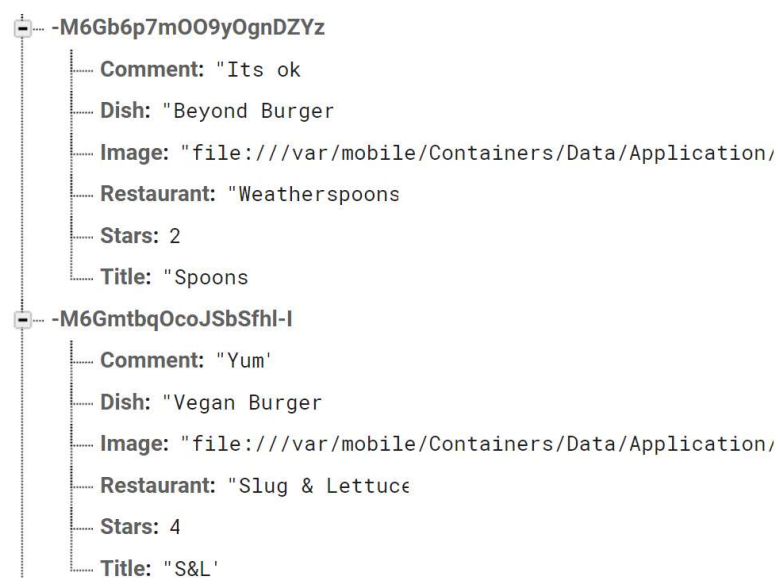
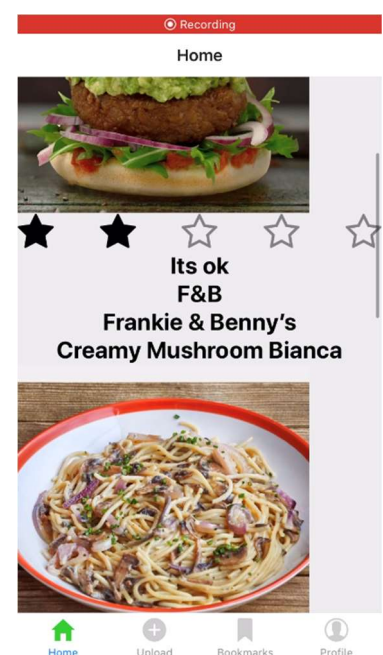


Figure 7

With information successfully being uploaded to an online database, the last step to meet all remaining core requirements was to display these posts on a home-screen in the form of a news feed to be visible to all users. This was done using two files, one for the homepage itself and another to display the posts in the database on the homepage. The code shown below in Figure 8 shows the data inside the database being imported to the home-screen. The code shown in Figure 9 is the code used in a components page which displays all fields of each post in order. Whilst this method resulted in the posts successfully being displayed on the home-screen, it seems that this is not the most efficient way of doing so as using object values and snapshot data resulted in restrictions such as making the data difficult to sort and add features to develop further. It was not until later testing that it was discovered that the images were only visible on the device from which they were uploaded which is when the image picker code was updated to use base64. With this change, two different iPhones showed the same feed. This met the requirement of being able to 'browse



from a list of vegan finds and view information such as the star rating and the restaurant name'. The feed is automatically arranged from newest to oldest posts. With frequently changing menus, it is useful that users are being presented with the most up-to-date information. However, being able to filter the list on different criteria such as location or rating would be beneficial in potential future development. Furthermore, since the user is free to enter any restaurant they like, they are not restricted by eatery type or location. However, geolocation could also be a way of potentially improving the app in the future, however it is important that users are not restricted to only the places detected by a third-party API since small, off the grid places are those that need to be shared the most.

```
import { db } from '../config';
let postsRef = db.ref('/Posts');

export default class Posts extends Component {
  state = {
    posts: []
  };

  componentDidMount() {
    postsRef.on('value', snapshot => {
      let data = snapshot.val();
      let posts = Object.values(data);
      this.setState({ posts });
    });
  }

  render() {
    return (
      <ScrollView style={styles.container}>
        {this.state.posts.length > 0 ? (
          <PostsComponent posts={this.state.posts} />
        ) : (
          <Text>No items</Text>
        )}
      </ScrollView>
    );
  }
}

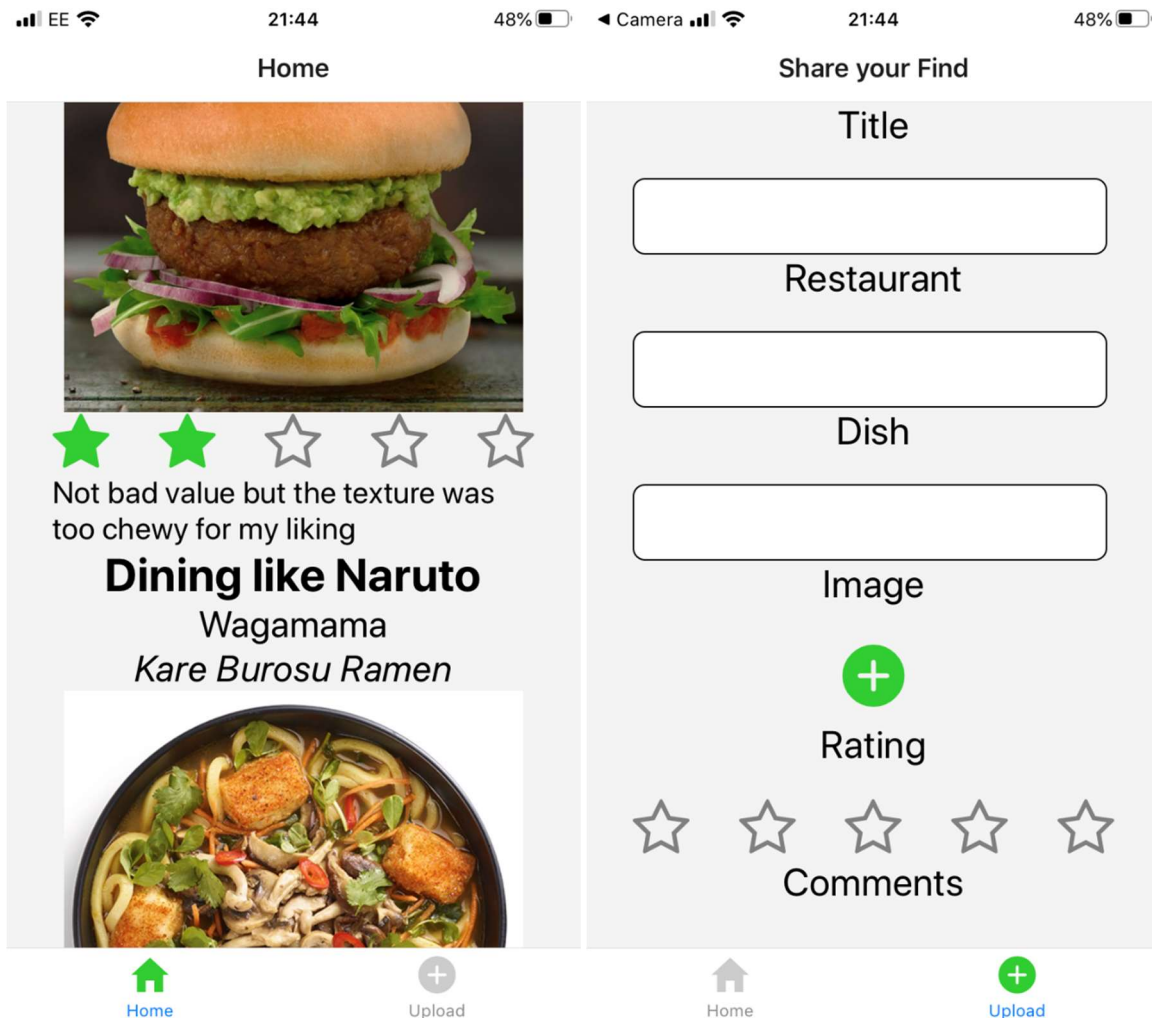
export default class PostsComponent extends Component {
  static propTypes = {
    posts: PropTypes.array.isRequired
  };

  render() {
    return (
      <View style={styles.itemsList}>
        {this.props.posts.map((post, index) => {
          return (
            <View key={index}>
              <Text style={styles.Title}>{post.Title}</Text>
              <Text style={styles.Restaurant}>{post.Restaurant}</Text>
              <Text style={styles.Dish}>{post.Dish}</Text>
              <Image style={styles.itemimg} source={{uri: post.Image}}/>
              <StarRating fullStarColor={'#32CD32'} disabled={true} rating={post.Stars}/>
              <Text style={styles.Comment}>{post.Comment}</Text>
            </View>
          );
        }).reverse())
      </View>
    );
  }
}
```

Figure 8

Figure 9

Lastly, to fully meet the requirement ensuring that users will be able to 'interact with a user-friendly, simplistic interface without requiring advanced technical knowledge', styling was needed to make the app more presentable and usable. This included things like ensuring that the text colour contrasted well with the background to ensure maximum visibility, clear and appropriate font size and styles, content being justified centrally to ensure all elements were in comfortable view and reach and adding hints of colour to ensure consistent themes. Screenshots of the resulting app can be seen below. Whilst the app is sufficiently neat and professional looking and has core functionality working correctly, further development would be beneficial to add more functional features and more advanced stylistic features. It was difficult to make the vegan purpose of the app clear within the app itself, however this could be more effectively achieved with an appropriate name such as 'Vegan Finds' and clear description in the App Store. Overall, the finished app is fairly simplistic but was designed and developed with user experience prioritised and suitably follows standards and recommendations set out by Nielsen's Usability Heuristics and Google's UX Design Principles.



Reflection

A benefit of React-Native development is that it allows the reuse of a codebase between both iOS and Android. Therefore, full cross-platform development is possible to some extent, depending on the number of native modules used in the application. Unfortunately, the finished app did not seem to be compatible with Android devices as it caused Expo to crash when the upload screen was loaded on an Android device. Originally, this was thought to be caused by the camera roll permissions however with code adaptation made specifically for Android, it continued to crash. This means that the likely culprit was an installed module however this would have meant extensive debugging of uninstalling and reinstalling packages one by one to find the problem. This would have been an inefficient use of time which shows a drawback of cross platform development as ensuring compatibility across devices can be a time-consuming task. Furthermore, without access to a variety of devices, it is difficult to check how the app looks and performs on any more devices than one during development.

Furthermore, the cross-platform nature brings limitations to design as styling that would look good on one screen size may not look good on another. For example, using pixel padding to align items is a poor choice in CPD as whilst it may result in an element in the intended place on an iPhone, it is likely to look entirely different on an iPad. For this reason, more globalised positioning methods such as justifying content centrally and flex containers were used instead. Moreover, it is also difficult to style for both iOS and Android devices, even of similar screen-sizes, as in most cases only either one or none will have a native look and feel. There is often a divide between iOS and Android users, each defending

their chosen mobile device passionately. People often have strong preferences due to the different types of UI design and user experience offered by the two competing platforms. For this reason, iOS users are likely to have a different idea of what looks aesthetically pleasing on their mobile device than Android users. As an iOS device was accessible throughout development, iOS compatibility and design was prioritised, resulting in an app native looking to iOS devices. Similar grey-tone backgrounds were utilised and icons specific to iOS were also implemented to create this native feel. So, whilst cross-platform development can be time and cost efficient due to its compatibility and reusability, a drawback can result in apps that do not look or feel native to some devices they are available on.

Lastly, another disadvantage of cross-platform development experienced throughout this process was the sheer difficulty. Whilst Node.js is a JavaScript runtime, it presents a completely new way of programming and development. Previous experience with HTML, CSS, PHP, SQL and both jQuery and classic JavaScript was much less helpful in learning this coding and development style than expected. Having such a wide range of optional module to install was overwhelming. It is often unclear which is the best package to use for a certain feature, which can result in modules being installed that are never used, resulting in large file sizes and a bloated app. Both PHP and Node.js are used in backend development and whilst they fall under the same categories, they are very different. PHP is far simpler to use than Node JS and Node JS requires more understanding and more lines of code. Learning the basics does not seem to be enough foundation upon which to build an entire app. It was a long and complex learning process, resulting in more time learning than developing, and therefore a basic app being implemented due to limited expertise. As well as this, after being used to SQL, the jump to Firebase, a NoSQL solution, was a rather big leap. The unstructured nature of the data in storage lead to more confusion in coding than simple MySQL queries.

Overall, whilst this was a very difficult task, it is clear to see the benefits of learning cross platform development. The graph shown below in Figure 10 shows the trend in mobile apps which shows how the percentage of entirely native apps fell from 20% to just 3% between 2015 and 2019. Whereas, the percentages of apps that are a mix between hybrid and native and entirely hybrid have increased. This shows that hybrid, therefore cross-platform, development is becoming the new norm as it is more time and cost effective. Furthermore, as mobile phone use continues to increase and the market continues to be dominated by the two giants, Apple and Android, it is important that all mobile users can have access to a product. For this reason, it is clear that learning cross-platform development is useful as is likely to be needed in the near future. However, the steep learning curve caused frustration and a less advanced app than what could have been achieved with a mobile site. With more time, this app could be developed further in many ways such as better compatibility between devices, and additional features such as geolocation and a favouriting system. Nevertheless, the designed app was a novel and useful idea that responds well to a mobile moment (being out as a vegan and looking for somewhere to eat) and therefore offers a purpose that could not be fulfilled by a desktop application.

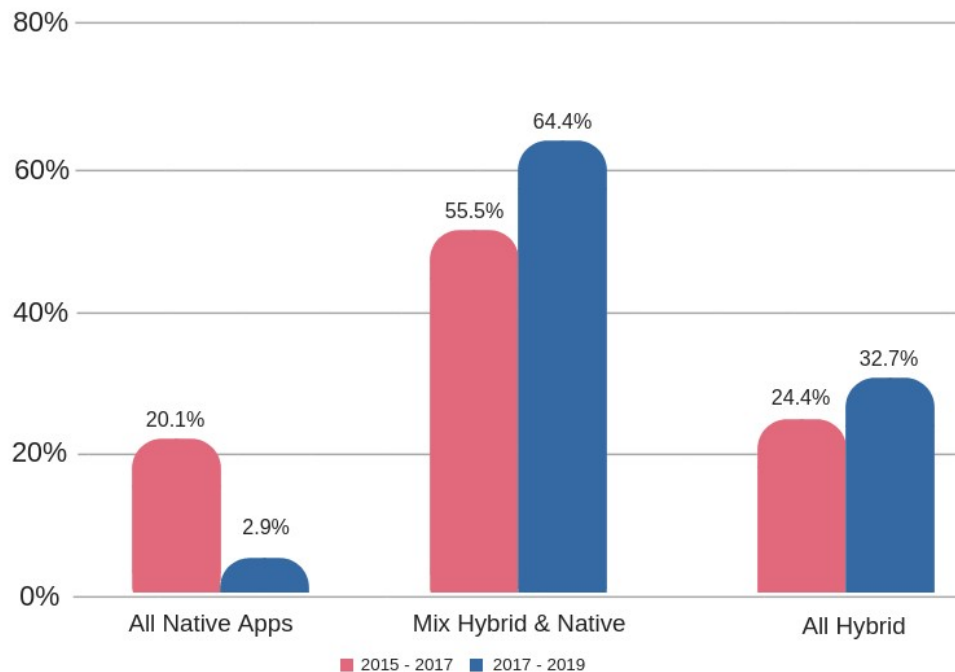


Figure 10

References

- Alathas, H., 2018. *How To Measure Product Usability With The System Usability Scale (SUS) Score*. [online] Medium. Available at: <<https://uxplanet.org/how-to-measure-product-usability-with-the-system-usability-scale-sus-score-69f3875b858f>> [Accessed 4 May 2020].
- Elgabry, O. (2016). *UX — A quick glance about The 5 Elements of User Experience (Part 2)*. [online] Medium. Available at: <<https://medium.com/omarelgabrys-blog/ux-a-quick-glance-about-the-5-elements-of-user-experience-part-2-a0da8798cd52>> [Accessed 4 Mar. 2020].
- Gidalevitz, Y., n.d. *Mood Boards In UX Design: Ignite Passion In Your Users*. [online] Usabilitygeek.com. Available at: <<https://usabilitygeek.com/mood-boards-ux-design/>> [Accessed 9 March 2020].
- Firebase. 2020. *Choose A Database: Cloud Firestore Or Realtime Database*. [online] Available at: <<https://firebase.google.com/docs/database/rtdb-vs-firestore>> [Accessed 5 May 2020].
- Google Developers. 2015. *What Makes A Good Mobile Site?*. [online] Available at: <<https://developers.google.com/web/fundamentals/design-and-ux/principles/>> [Accessed 5 May 2020].
- Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. ACM CHI'94: Conference on Human Factors in Computing Systems. Pg.152-158
- Usability.gov. n.d. *Usability Testing*. [online] Available at: <<https://www.usability.gov/how-to-andtools/methods/usability-testing.html>> [Accessed 20 April 2020].
- UX Movement. 2017. *Why Mobile Menus Belong At The Bottom Of The Screen*. [online] Available at: <<https://uxmovement.com/mobile/why-mobile-menus-belong-at-the-bottom-of-the-screen/>> [Accessed 5 May 2020].