

## Notes

### 4.1.1

TravelTracker is an app that provides its users with all the necessary information, regarding a given destination, on their fingertips. This app will be supplemented with a website, which will introduce the app and include a link to get on your phone. Additionally, the android app and the website will register users into its database which will allow more advanced features like filtered results based on a user's preferences.

#### Functional requirements:

- Store user data in a database, containing returning and new users, that is updated when new users register
- The database storing information related to the countries should include unique visits.
- Storing previously searched keywords
- The database driven website and app will MariaDB as the chosen database management system. SQL queries will retrieve necessary information from pre-existing databases. The testing of this website will be done on XAMPP as the local host/testing environment.
- Certain features will only be able to be accessed by registered users for example bookmarking a country or saving a search.
- The database of a country will store tourist attractions, tourist tips, most visited cities, a cultural fact and how long a typical trip is for a tourist.
- The database of users will be one expected to change overtime as users join or leave. This database will store, for each user, their name, country of residence, number, email and user ID and a Boolean that records if the user has signed up for updates/a newsletter. This Boolean should acknowledge how much the user would like to be informed i.e. updates relating to only xyz topic or abc and xyz, or all updates.
- Queries responsible for accessing the countries database should be fast, therefore a searching algorithm that prioritizes fast results should be preferred.
- 

#### Application functions:

- Register new users
- Include a map with countries colored in
- Each country is clickable and leads to a new page that will reference a pre-existing database to retrieve information related to that particular country
- The app will include categories so users can sift through countries that fit a criteria for example: 'This week's most searched countries'.
- Countries can be bookmarked and added to custom lists.
- The app will allow users to turn on dark mode.

#### Website functions:

- Include a landing page
- Register new users
- Include a privacy policy and terms of service on the footer
- Includes why, where and how to get it
- Includes which platforms the app is compatible with
- Includes an 'Email us!' page

- Includes a pop-up giving users (returning or new) to sign up for a newsletter/updates
- 

## Requirements bit

### 4.1.1 Product Perspective

The TravelTracker app will be supplemented with a website, which will introduce the app and include a link to get on your phone. The database driven website and app will utilize MariaDB as the chosen database management system. SQL queries will retrieve necessary information from pre-existing databases. The testing of this website will be done on XAMPP as the local host/testing environment.

### 4.1.2 Product Functions

#### 1. Application Functions

- a) The system provides two varying paths of user experiences, with a registered user having access to certain functions like 'bookmarking'.
- b) The primary function of this application is to provide a condensed pamphlet of sorts that contains information relevant to the country picked by the user.

#### 2. Website Functions

- a) The primary function of this website is to supplement the application by providing an introduction to it.

### 4.1.3 Constraints

1. This project is susceptible to time constraints
2. Changes to database are permitted to select team-members to preserve data integrity
3. Users cannot bookmark a country unless they are signed in
4. The app is not a cross-platform app and will only be compatible with android phones
5. The app needs user permissions to run: access precise location (GPS and network based), access approximate location (network based), full network access...
6. App should not exceed 30 MB
7. Website should be cross-browser platform friendly
8. Data retrieval and transferal between server and client will utilize TCP/IP protocol and therefore will not be accessible from the internet.

## 4.2 Specific Requirements

### 4.2.1 User Interfaces

1. Website and app layout should be responsive to the size of the viewport
2. Font size should be legible and text to speech friendly
3. In order to increase accessibility to users who require assistive technology, ARIA (Accessible Rich Internet Applications) can be added to the HTML elements, assigning them with roles, properties and states so users needs are accommodated
4. Sidebar menu should be implemented so users can navigate the app with ease.

5. Interface should have a hierarchical layout in how information is presented. Consider the following scenario: There are 195 countries. However the user should only be presented with information pertaining to their country of interest.
6. The user interface should be consistent and highlight the user's actions. For example, once a link embedded in the word 'Africa' has been clicked, the text should have a visible change indicating that its current state is visited. Another example would be if the user is currently on the 'Home' page of the website/app then within the menu, that menu heading should be obviously distinct from the other menu headings.

#### 4.2.2 Hardware Interfaces

1. The system must have at least 30 MB free to allow installation of the app.
2. The website/app does not store information onto the user's computer, rather it utilizes a database located on the network server.
3. By utilizing network protocols, the user's computer should be able to retrieve data from the server

#### 4.2.3 Software Interfaces

1. The website requires XAMPP to be installed and running.

#### 4.2.4 Functional Requirements

##### 1. Android Application

- a) *Register New Users / login for returning users* : Users are directed to register or login. Alternatively they may also view the app anonymously. It should be made clear that anonymous browsing means they cannot bookmark countries.  
*Input* : Radio buttons are utilized to determine if a form is necessary. In the instance of the user choosing to brows anonymously, a form would not be necessary. Otherwise, the user is presented with the relevant form (registration or logging in). Both forms require an email, a username and a password, however the registration form also requests the user fill in their number, country of residence and age. At the bottom of the registration form the user should be requested to agree to the terms of service. Additionally, an option to opt int to updates/a newsletter should be included.
- b) *Colored Map* : The homepage of the application includes a world map with countries colored in. Users are offered an option to change how the countries are color-coded. The options provided are pre-determined and must be selected from the list available for example countries that drive on the left, countries which border France or countries that start with the letter W.  
*Input* : A single option must be selected, at one moment, from the various options available. A radio button may be utilized for this purpose.  
*Output* : Changes in color coding of the countries within the world map.
- c) *Clickable Countries on Map* : Each country is clickable and leads to a new page that will reference a pre-existing database to retrieve information related to that particular country.  
*Output* : Upon clicking a country, information regarding them is displayed on the screen in alphabetical order. Upon reaching the end of the page, the user can easily go back to the top of the page or to the world map. These paths will be embedded as links into the footer of the page.
- d) *Bookmarking* : At the top right corner of each country's respective page, is an option to bookmark the page in the form of a no-fill, outlined bookmark symbol.  
*Output* : Once bookmarked, the symbol is filled in.

- e) *Dark Mode* : The app will allow users to turn on dark mode.  
*Input* : In order to turn dark mode on/off users must use the toggle switch in the menu.  
*Output* : Certain colors, like font color, will change in dark mode to promote visual accessibility to users of all needs.
- f) *Menu* : The app includes a hamburger menu that will open to reveal the app's navigation system comprising of *Home*, *Settings* and a toggle switch for turning on/off dark mode.

## 2. Website

- a) *App Details* : Includes why, where and how (download link embedded in text or using a QR code) to get the app
- b) *Mobile Compatibility* : Includes which platforms the app is compatible with
- c) *Banner* : Includes a pop-up giving users (returning or new) to sign up for a newsletter/updates
- d) *Web Design* : The website will have a responsive web design that adjusts its layout according to the viewport

## 3. Database

- a) *User Table* : This table will consist of a user's username, email address, phone, country of residence, age and a Boolean that records if the user has signed up for updates/a newsletter. This Boolean should acknowledge how much the user would like to be informed for example updates relating to only xyz topic or all updates.
- b) *Countries Table* : This table will consist of the country, tourist attractions, tourist tips, most visited cities, a cultural fact and how long a tourists typically stays there for.
- c) *Preferences Table* : The color coding of the countries on the world map can be changed by the user depending on which option they choose to select. A preferences table should be constructed that includes the country along with columns highlighting different and unique options. The data stored for that option will be a Boolean.