### Data Plan

Server Routes

|  |  |  |
| --- | --- | --- |
| Method | Purpose | Request/Response |
| GET / | Prompts user to login to the website | Responds with a login page |
| GET /register | Prompts user to register an account | Responds with a registration page. Venue managers will also register their venue. |
| GET /user/details/*userID* | Retrieve all information about a user based on its ID. Presented as a web page. | Respond with a user object describing the user details such as firstName, lastName, address, etc. |
| GET /user/details/checkIn/*type?id?* | A user retrieves all check-in history based on its type (admin or user) and ID. | Queries the database and returns all checkins for the user ordered by time |
| GET /venue/details/*venueID* | A venue manager retrieves all information about who has checked into a venue in the past week. | Queries the database and returns all checkins for the venue ordered by time |
| GET /admin/details/recent\_visits/*type?id?* | An admin retrieves all information about who has checked into a venue in the past week. | Queries the database and returns all checkins for the user ordered by time |
| GET /hotspot\_area | Retrieve all information about each venue that is a hotspot | Respond with an array of venue objects that have a number of cases that exceed a predefined limit. |
| GET /admin/create | Prompts an admin to promote a user to admin role | Responds with a page to get the required information |
| POST  / | Authenticates the user | Sends a user object to the server. |
| POST  /register | Registers the user and/or venue | Sends a user object and/or venue object to the server |
| POST  /user/details/update | Update any necessary information about the user. | Send a user object to the server to update the database |
| POST  /venue/details/update | Venue managers can update information about their venue. | Send a venue object to the server to update the database |
| POST /admin/hotspot\_area/update | Update or add a hotspot to the database. | Send a venue object to the server to update the database |
| POST admin/user/userID/hasCovid | Update a user’s covid status | Send a boolean value to the server to update the database |
| POST  /venue/venueID | Generate a QR code for a user to scan. | Server responds with an image of the QR code |
| POST /check-in | Adds a user and a venue to the check-in record. | Respond with a boolean as to whether the data has been successfully updated. |
| POST /admin/create | Create a new account for another admin | Send a user object of the new admin to the server to update the database |
| POST /user/google\_account | Create a Google account as a user or venue manager. | Send a user object of the server to update the database |

Object Examples

User Object

{

"id":"236",

"firstName":"Antony",

"lastName":"Cameron",

"accountCreationTime":"2019-05-26T01:42:55.840+00:00",

"address":{

"id":"236",

"addressLine1":20,

"addressLine2":”Heaven Road”,

"suburb": "Adelaide",

"postcode":5432

},

"dateOfBirth":"2001-04-13T00:00:00.000+00:00",

"email":"acameron2823@gmail.com",

"mobileNumber":12345678,

"hasCovid":false,

"check-in":[

{

"id":"236",

"venue":"626",

"date":"2019-05-28T07:00:27.706+00:00"

},

{

"id":"236",

"venue":"626",

"date":"2019-05-30T13:18:23.406+00:00"

}

]

}

Venue Object

{

"id":"626",

"name":"Sam’s Business",

"address":{

"id":"101",

"addressLine1":15,

"addressLine2":”Valley Road”,

"suburb": "Adelaide",

"postcode":5082

},

"contactNo":85551234,

"long":1234.1234,

"lat":1234.1234,

"total\_cases":10,

"recent\_visits":[

{

"id":"236",

"user":**User Object**,

"date":"2019-05-28T07:00:27.706+00:00"

},

{

"id":"237",

"user":**User Object**,

"date":"2019-05-30T13:18:23.406+00:00"

}

]

"venue\_manager":[

{

"id":"236",

"firstName":"Antony",

"lastName":"Cameron",

"accountCreationTime":"2019-05-26T01:42:55.840+00:00",

"address":{

"id":"101",

"addressLine1":10,

"addressLine2":”Fife Road,

"suburb": "Adelaide",

"postcode":5085

},

"dateOfBirth":"2001-04-13T00:00:00.000+00:00",

"email":"yvonne.tansu@gmail.com",

"mobile\_number":12345678,

"hasCovid":false,

"check-in":[

{

"id":"236",

"venue":"626",

"date":"2019-05-28T07:00:27.706+00:00"

},

{

"id":"237",

"venue":"626",

"date":"2019-05-30T13:18:23.406+00:00"

}

]

}

]

}

Client/Server Interactions

|  |  |
| --- | --- |
| Page | Interaction |
| ALL | Each page (user, venue manager, admin) is set up statically by the server |
| Home | Is a login page |
| My Profile | The client updates user information. The client saves these changes by updating them to the server. |
| Manage Users | The client (in this case the admin) requests information about a certain user (e.g. a positive case). The client sends new information to be updated about the user |
| Manage Venues | The client requests information about a venue (e.g. location, recent check-ins). The client sends new information about the venue to be updated and the server responds by displaying updated information. |
| Maps | The client requests a list of recent hotspots and user check-ins. Hotspots are updated to the server and the map of current hotspots is also updated accordingly |