WEB TECHNOLOGIES

**REPORT TITLE**

BIT TOURS TOURISM CENTER

horizontal line

# 

# TEAM MEMBERS:

# Edel Togobo

# David Adu-Amoani

# Tracy Poku

# Joseph Prince - Agbodjan

# Introduction

The problem Bit Tours seeks to solve is the gap created when tourists seek the perfect and most suitable destination and lodging location after they have decided to make the best out of any visit. We are the meeting point between the realization of a perfect vacation and the idea of enjoying one, be it taking care of location, site, tour guiding, community review and personal recommendation.

Our foremost operation is giving advice on the best tourist locations in the world and tagging them according to the best time of visit. We do this through or website which aims presents visitors with information on the best places to visit at the time which we be as a result of the weather at the time, the events happening within a culture and the kind of food present at the time.

Also, Bit Tours gives customers and tourists the opportunity to thoroughly review, critique and recommend places of the visit after a vacation. There are chat forums which allow visitors to share experiences and inform new visitors of what it is like to be at a tourist site. Tourists can upload pictures of their activities and build an experience visualizer. A visitor, after seeing a recommended location, can add it to his/her wish list should they please.

Moreover, the Bit Tours website serves as a guide in itself. It provides mapping services for customers to enable them to track their movement and find their way around a new location. It will inform of the nearest service points like restaurants, call booth, info centres and recreational points. Bit Tours primarily seeks to be a tourist’s best companion right before, during and after an experience.

***PROJECT STRUCTURE***

**Front - End**

The design of the business model of Bit Tours is one that requires an eccentric but very familiar human-computer interaction. As such, the task of designing the front-end is as important to the core processes of the business as the profit model, especially for a business model which almost solely depends on the web presence. The front-end structure was built on HTML5, CSS, JavaScript and PHP languages. These frameworks enabled the use of the most recent technologies in making carousels, Bootstrap cards, front-end validation and the like.

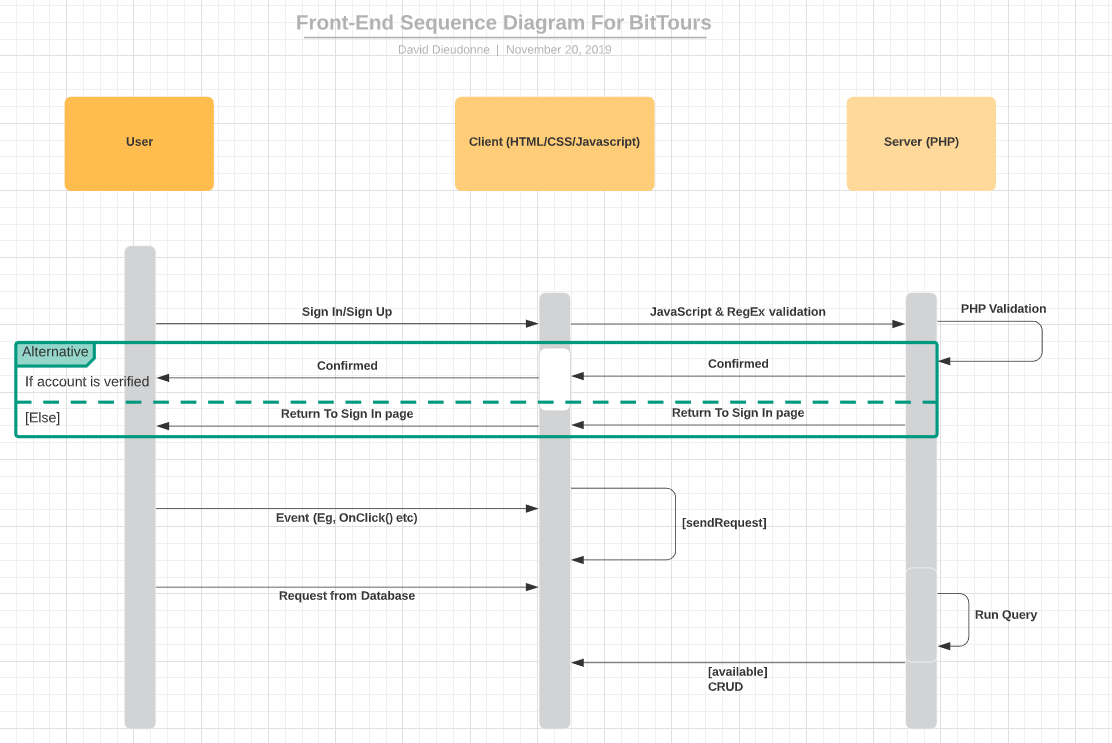
We made use of ajax programming to enable us to design a quick and time-efficient website which delivered on request without going serving a visible request to the server. There is regular expression validation on sign-in and sign-up pages. This is necessary because the account system is one very critical to our user experience and therefore needs to be as secure as possible so that user data is not breached and compromised. The mark-up frameworks used were Materialize and Bootstrap as well as jQuery and Popper JS for the JavaScript foundations.

Due to the main interaction pages necessary, we have the home, about, services, reviews and contact pages to enable users to make the most of the interaction with the page. The home page is the initial landing page for a regular user when they follow the domain name. It consists of our offers on the carousel that is first seen, our primary services placed on Bootstrap cards and a site map which is mainly a materialize template fitted with our email feedback powered by original PHP code. The about page gives information about the company and leads users to the review page which enable users to comment and rate travel sites as per our business model.

Primary Functionality

The primary functionalities of the front-end of the application can be analysed through the sequence diagram below.

**Sequence Diagram**



Areas of Challenges

The major area of difficulty was the regular expression implementation. This is because of the wide range of possible inputs accepted for each input section in the form for review and sign up. We had to cover every instance to maintain high fidelity and integrity of our data.

In solving this, we beta tested several instances of undesired inputs to enable us to get a wholesome and close to perfect representation of every possible case.

**BACK - END**

For the backend development of the bit tours full-stack web application, there are different stages that form this section of the project. For our project, we made use of Classes, Pages, Validation, Ajax API POST and GET requests and some simple database functionalities such as reading, writing, updating and deleting information from the database. With the use of PHP, we were able to show and hide content based on user account privileges.

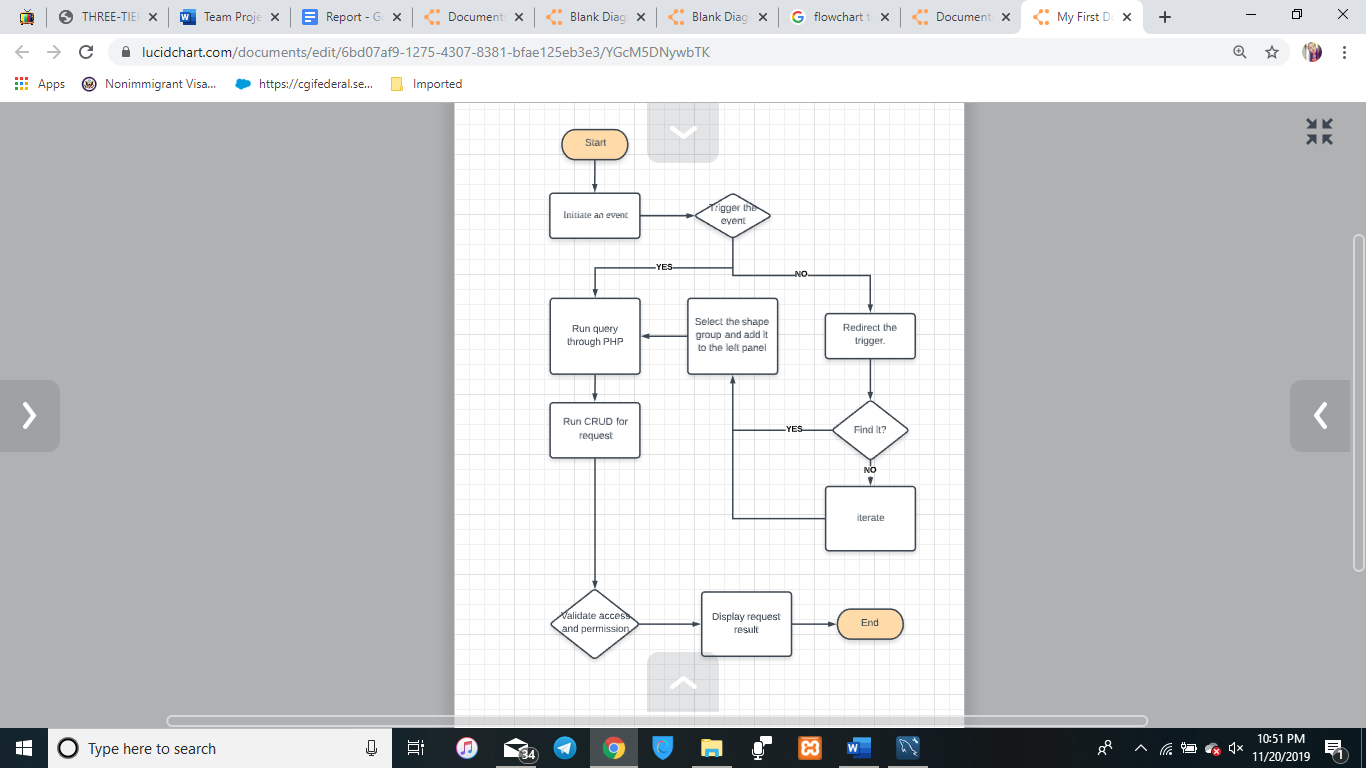
CLASSES:

For easier and more flexible use of functions and objects, we made use of classes for our database and some other functionalities of the system. We made use of a database class to hold the function and properties that allow for creating an account, deleting an account, updating some details from an account, signing up for the web application and logging into the account that they create.

VALIDATION:

For this section of the program, we believe it to be the most important part of the entire project. To ensure that the system is fully functional and has no unwanted data being entered into the system, we made use of HTML tag attribute validation, javascript validation and PHP (Backend validation). We also made use of regex expressions to allow for strict validation from the user’s input as they are signing in, logging in, or editing existing profile details.

**Flowchart**

****

**DATABASE**

Considering the case scenario of Bit Tours Tourism Center, and the functionalities of the proposed website, we needed a database that would match the functions of the website, specifically, be able to store user information, hotels, restaurants and tourist site information that is uploaded to the site and the reviews that users make. Considering this, the following tables were created with MySQL:

The **USERS** table is where all the user information is contained. Here, when the user logs in and enters all their personal details, it will be stored in this table. As such, when the user or another user requests to a profile, the profile details will be fetched from this table.

Moreover, the **HOTELS, RESTAURANTS** and **TOURIST\_SITES** tables contain the information about the hotels, restaurants and tourist sites that will be uploaded to our site respectively. As such, when a user uploads a restaurant they visited and gives the name, address and a brief description of the restaurant, these details will be uploaded to the **RESTAURANTS** table. Also, if a user searches for a hotel, the name inputted by the user will be checked against the list of hotels inputted into the **HOTELS** table.

Finally, there are the **HOTEL\_REVIEWS**, **RESTAURANT\_REVIEWS** and **TOURIST\_SITE\_REVIEWS** tables. These tables store the reviews that users post about the particular hotel, restaurant or tourist site, respectively. The tables are able to store the users rating and comment for that particular hotel, restaurant or tourist site and can match a rating or comment to a user.

**AREAS OF CHALLENGES**

The major area of difficulty was the regular expression implementation. This is because of the wide range of possible inputs accepted for each input section in the form for review and sign up. We had to cover every instance to maintain high fidelity and integrity of our data.

In solving this, we beta tested several instances of undesired inputs to enable us to get a wholesome and close-to-perfect representation of every possible case. Also, the organisation of files for the three-tier architecture was a bit og