CRIS: A Web-based Crowdsourced Commute Routing Information System

Samuel Tadeo L. Bautista and Monina Gazelle Charina B. Carandang

I. Introduction

A. Background of the Study

The Philippines is known to be a commute heavy country as majority of its citizens commute to either school or work everyday. The lack of a centralized government-moderated source of information for the commute routes within the country poses an inconvenience for first-time commuters in new locations. There are other online user-moderated sources that can be utilized by the public such as Sakay.ph, ph-commute, ParaSaTabi, TripBarker, etc. However, these online sources tend to be either outdated or limited in scope. A dynamic, nationwide information system accessible and editable by the public will be beneficial to the masses.

B. Statement of the Problem

Some of the Filipino people are currently facing two problems - learning how to commute and passing on that knowledge. Students, job seekers and travellers have to commute to places normally not familiar to them; such knowledge or prior experience from others will greatly aid in their commute.

C. Significance of the Study

The application will help by guiding its users in possible commute routes, and to let the users pass on that knowledge through the application by specifying routes that they know themselves.

D. Objectives

The general objective of the study is to develop a web application that guides its users in commuting in the Philippines. The specific objectives of this study are:

1) to create a Single Page Progressive Web App;
2) to let the app display the shortest, fastest and cheapest commute routes between two locations;
3) to allow users to input their own commute routes in the application; and
4) to support both Desktop and Mobile device layouts.

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

E. Scope and Limitations

Since there are various modes of transportation in the Philippines, the application will only focus on:

1) Walking
2) Jeepneys
3) Tricycles
4) Buses
5) Trains
6) Pedicabs
7) Shuttle services
8) UV Expresses

II. Review of Related Literature

Word of mouth is the predominant mode of information sharing of commute routes in the Philippines. The inclusion of technology to this process was only introduced in 2001 [1]. Many other web services have sprung throughout the years, each matching with the available technology in those times.

iKodeko Software Inc.'s ParaSaTabi was the first among the commute web services to be launched in the Philippines. The web service is free and simple. The service asks the user where to commute to and from, then show a series of straightforward steps in text on how to commute there. It would occasionally include descriptions for each step to clarify the directions. The commute routes data is also crowdsourced with no moderation from administrators. The web service however lacked a visual aspect in showing the directions [1].

Lance Lim and Mary Ann Ngo's ph-commute blog showcases a collection of places within the country, their description, some fun facts, and popular routes on how to commute there. The blog is free for viewing. Each route contains detailed information on each step of the commute. The blog however, being a static website, does not provide any service for adding new routes from its viewers and is only maintained by the authors [2].

EACOMM Corporation's TripBarker was the first of the web services to employ a map service, namely Google Maps. Like the web services mentioned above, TripBarker is also simple and free. It employs the same procedure as the others, which lets you input origin and destination locations, and then shows the possible commute routes from the origin to the destination. The service also shows an estimate of how each route would cost. Users can also report the weather, traffic, and other events nearby so that other users can be notified real time. The service, however, does not crowdsource its commute routing data to its users, leaving the maintenance

of the data to the administrators [3]. TripBarker won the Best Transit App from the Philippine Transit App Challenge [4].

98Labs Inc.s Transit.com.ph shares many similarities with TripBarker, however it optionally crowdsources its commute routes to its users through a suggestion form which is filtered by the administrators. One drawback of the app is that the commute routes are constrained to Cebu only [5]. Transit.com.ph won the Transport Agency Award from the Philippines Transit App Challenge [4].

The most recent of the services in the genre is a web app called Sakay.ph, made by By Implication, which also was an entry to the Philippine Transit App Challenge and won the Open Community Award [4]. It has similarities with TripBarker and Transit.com.ph in a way that it also employs a map service. Though from the rest of the aforementioned applications, only Sakay.ph is being actively maintained and updated with the newest technologies. The application also provides short descriptions per route, and the cost for each ride. Sakay.ph also employed a text messaging system to let users communicate with the app without having to connect to the Internet. One limitation of the app is that it only has commute routes for Manila, and only accepts reports on outdated routes, meaning the app is not truly crowdsourced [6].

III. Methodology

A. Architecture and Technologies

The application will consist of the server where the Application Programming Interface (API) will be exposed, the database to hold the necessary persistent data, and the client which will be the interface for the users.

The server will be developed with Node.js v8.9.0, a JavaScript runtime environment based on Google Chromes V8 engine [7]. The server will be structured using Express.js, a minimalistic opinionated framework for developing server side web apps [8]. The framework will be used as it provides flexibility while developing and is less bloated than other similar frameworks. The API will be made RESTful, so that its endpoints will have coherence. The server will be hosted on a Digital Ocean Droplet or cloud server [9].

The database to be used for the application will be MongoDb 3.4.10, a NoSQL database [10]. Since majority of the data will only be from the routes themselves, a NoSQL database can be utilized since they are perform better compared to relational databases in larger sets of data that do not have much relationships within the data entities [11].

For the client, React v16.0.0 will be primarily used for handling the user interface [12]. create-react-app will handle the initial scaffolding [13]. Redux will be used to handle state management, and redux-thunk will be used for to handle side-effects in the state of the app [14] [15]. The application will have a material design to it, and will incorporate Material-UIs components into the user interface [16]. Google Maps will be used as the map service of the application, as some its APIs have certain functionalities that will benefit the app.

B. System Functionalities

1) Sign up: The application will not employ admin users. As such, signing up is optional. Authentication is only necessary if the user wants to manipulate routes. Signing up will require a nickname, email address, and a password.

2) Login and Logout: The application will contain an interface for logging in and logging out. This is necessary for authentication. Logging in will require the email address and password of the user.

3) Search Route: Users can fill up a form consisting of origin and destination locations. Each input field will make use of Google Maps Places API's autocomplete function [17]. Once the user has filled in both fields, the application will automatically search for possible routes between the two locations. If the user changes either of the inputs, the previous request will be cancelled and a new search will start. Once a search has finished, the application will plot the path with the shortest route on the map.

4) Plot Routes on Map: The application can plot existing routes on the map widget. Origin and destination locations will have unique markers on the map. Each route on the map will be color coded for easier visibility. Each segment of the route will show an icon indicating which mode of transportation will be used. If the systems data is insufficient in getting the whole route from the origin to the destination, the default mode of transportation for that segment of the route will be walking, and the route plotted will be fetched from the Google Maps Directions API [18].

5) Create new Route: This functionality requires the user to be logged in. Users can enter new routes by specifying which mode of transportation, the origin and destination locations, and optionally the fare in pesos and estimated time of travel. The user will then plot the route starting from the origin up to the destination. Each adjacent point in the route should be at least within 300 meters from one another. This will be enforced by the app. Each point will be snapped to the road by using Google Maps Roads API [19].

6) Edit existing Route: This functionality requires the user to be logged in. Users can edit the details of existing routes.

7) Report existing Route: This functionality requires the user to be logged in. Users can report existing routes if the route is totally incorrect and not fixable by editing. Once a route has reached enough reports, it will automatically be deleted by the app.

C. Testing

To test the usability and feasibility of the web app, a System Usability Scale will be used which will be integrated into the app itself. The study will employ snowball sampling as its method for gathering participants. The study will have no maximum participants. Respondents will not be forced to answer the survey, but will be shown an option in the menu to take it once a user has logged in.

IV. Initial Results

A. Landing Page

As seen in Fig. 1, the landing page consists of the bare map as well as a burger menu button on the upper left, which

triggers the menu drawer. For Fig.2, the origin and destination input fields will already be visible at the top, beside the menu button.

Fig. 1. The landing page in desktop view.

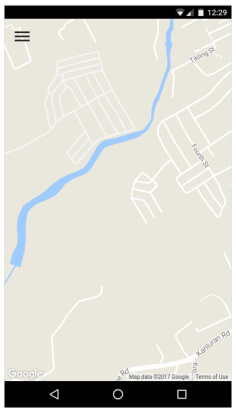

Fig. 2. The landing page in mobile view.

B. Menu Drawer

As seen in Fig. 3 and Fig. 4, the menu drawer will have a close icon that hides the drawer. For the desktop view, the origin and destination input fields are located here. The drawer will act as the main navigator for the app, consisting of links to other different functionalities depending whether the user is logged in or not.

C. Log in

As seen in Fig. 5 and Fig. 6, the login component shows a simple form consisting of an email field and a password field. Users may login using this form. Users may also opt to sign up if they do not have an account yet.

D. Sign up

As seen in Fig. 7 and Fig. 8, the sign up component consists of a simple form consisting of a nickname field, email field, and a password field. Users may sign up their account here.

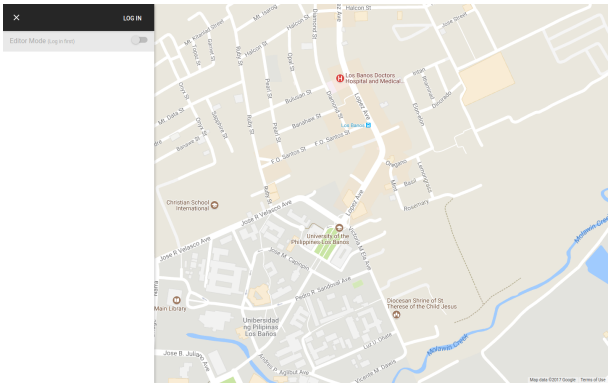

Fig. 3. The menu drawer in desktop view.

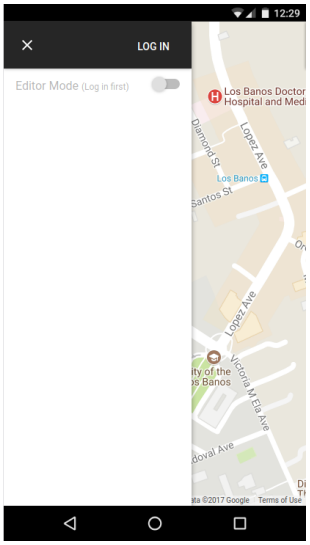

Fig. 4. The menu drawer in mobile view.

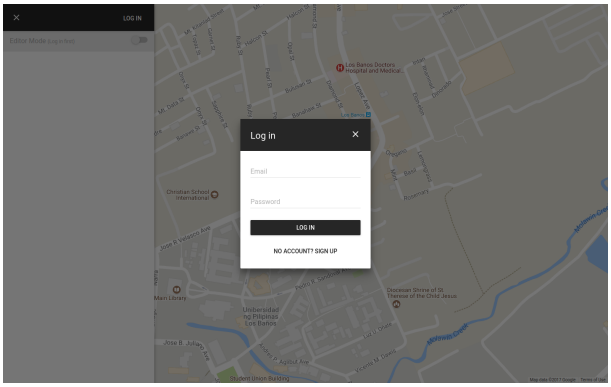

Fig. 5. The login form dialog in desktop view.

E. Drawing a Route

A map is shown in Fig. 9 and Fig. 10. The path of the route that UPLB College jeepneys take is drawn using Google Maps' built in Polyline component. Users can click on or near roads to apply more points to the path. After each click, a request to Google Maps Roads API is sent. The response of request will be the interpolated points from the last point up to the clicked point. Interpolating these points will show a smooth

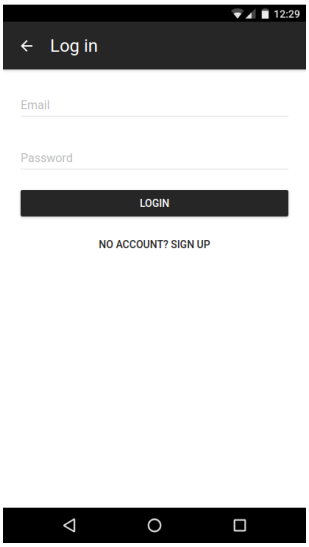

Fig. 6. The login form dialog in mobile view.

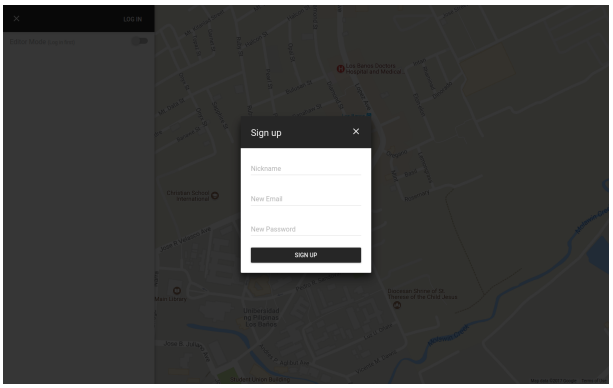

Fig. 7. The sign up dialog in desktop view.

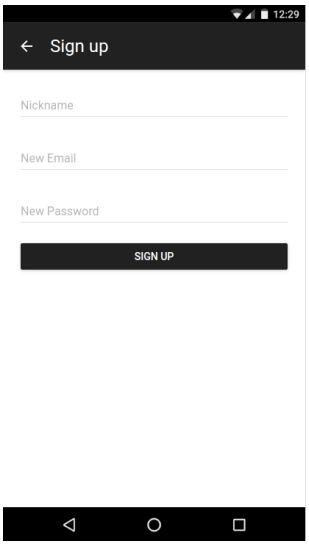

Fig. 8. The sign up dialog in mobile view.

path of the polyline, making it snap to the road.

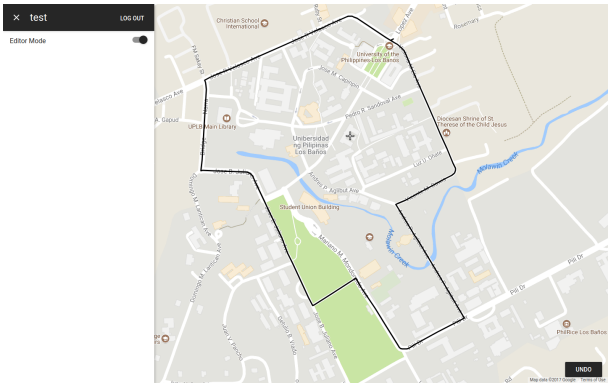

Fig. 9. The route that UPLB College jeepneys take in desktop view.

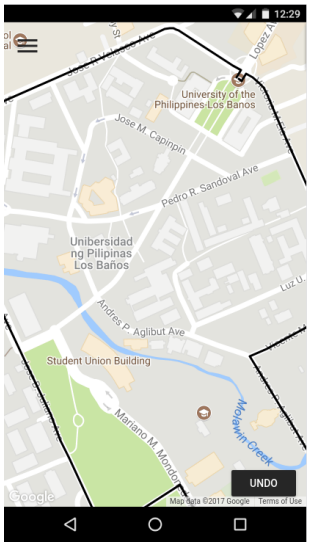

Fig. 10. The route that UPLB College jeepneys take in mobile view.

REFERENCES

[1] (2001) Parasatabi.com - public transportation guide for the philippines. [Online]. Available: http://www.parasatabi.com/
[2] (2008) ph-commute — commuting in the philippines 101. [Online]. Available: http://www.ph-commute.com/
[3] (2013) Trip barker - trip planning, traffic, weather and more for mass transit commuters. [Online]. Available: http://www.tripbarker.com/
[4] (2013) Philippine transit app challenge — create apps to solve transit problems in philippine. [Online]. Available: http://philippine-transit.hackathome.com/
[5] (2013) Transit.com.ph - plan your commute. [Online]. Available: https://transit-com-ph.appspot.com/
[6] (2013) Sakay.ph commuting in metro manila made easy. [Online]. Available: https://sakay.ph/
[7] (2009) Node.js. [Online]. Available: https://nodejs.org/
[8] (2010) Express - node.js web application framework. [Online]. Available: https://expressjs.com/
[9] (2011) Digitalocean: Cloud computing designed for developers. [Online]. Available: https://expressjs.com/
[10] (2011) Mongodb for giant ideas. [Online]. Available: https://www.mongodb.com/
[11] Z. Parker, S. Poe, and S. V. Vrbsky, "Comparing nosql mongodb to an sql db," in *Proceedings of the 51st ACM Southeast Conference*, ser. ACMSE '13. New York, NY, USA: ACM, 2013, pp. 5:1–5:6. [Online]. Available: http://doi.acm.org/10.1145/2498328.2500047
[12] (2013) React - a javascript library for building user interfaces. [Online]. Available: https://reactjs.org/
[13] (2016) Create react apps with no build configuration. [Online]. Available: https://github.com/facebookincubator/create-react-app

[14] (2015) Predictable state container for javascript apps. [Online]. Available: https://redux.js.org/
[15] (2015) Thunk middlware for redux. [Online]. Available: https://github.com/gaearon/redux-thunk
[16] (2014) Material-ui. [Online]. Available: http://www.material-ui.com
[17] Google places api. [Online]. Available: https://developers.google.com/places/
[18] Google maps directions api. [Online]. Available: https://developers.google.com/maps/documentation/directions/
[19] Google maps roads api. [Online]. Available: https://developers.google.com/maps/documentation/roads