ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse189932020.pdf
https://doi.org/10.30534/ijatcse/2020/189932020

MUSICHUB: A Web and Android Based Rehearsal Studio Locator and Reservation System in Davao City Utilizing Geolocation API and Rabin-Karp Algorithm

Sushmita M. Gomez[1], Anacel P. Guantero[2], Joeharie A. Bulgao[3], Jan Carlo T. Arroyo[4], Allemar Jhone P. Delima[5]
[1-5]College of Computing Education, University of Mindanao, Davao City, Davao del Sur, Philippines
[5]College of Engineering, Technology and Management, Cebu Technological University-Barili Campus, Cebu, Philippines
ssushigomez18@gmail.com[1], yaszguantero@gmail.com[2], joeharie22@gmail.com[3], jancarlo_arroyo@umindanao.edu.ph[4], allemarjpdjca@yahoo.com[5]

ABSTRACT

This study aims to provide a search platform for band rehearsal studios where interested customers can locate and reserve the nearest studio. In the proposed system, the band rehearsal studio owners can advertise their studios along with the other services they offer. Through this, the search and reservation process is made more accessible. Android Studio, Java Programming Language, Google Geolocation API, Google Maps Android API, Google Directions API, Photoshop CS6, and PHP, among others, were utilized in the development of the system. Further, the Rabin Karp Algorithm is instrumental in generating the category and list of rehearsal studios. Furthermore, the Agile methodology was observed throughout the development of the system.

Key words: Geolocation API, Rabin-Karp algorithm, rehearsal studio, reservation system

1. INTRODUCTION

The rapid spread of the internet is very well known today, where technology has gone beyond client-server applications. New ways of software products are transforming people's work, organizations, and society. Technological variables such as growing global access to the Internet, the penetration and adoption of smartphones, along with the maturity of technology; is putting more music in the hands of people around the world.

Through technology evolution, an increased amount of listening, discovery, and consumption of music is perceived. As technology emerged and music studios became software enabled, it allowed musicians and singers to record at a more efficient pace. This resulted in more players investing in building studios for people with musical inclination.

An interview regarding rehearsal studios is conducted in Davao City, Philippines, where 100 random individuals were asked about the importance of having an application that will cater to the needs of people in terms of reservation and location of rehearsal studios.

Based on the findings, there are lots of difficulties and complications in terms of locating rehearsal studios. It is also distressful for musicians and clients to make reservations, especially if the availability of the studios or equipment is unknown.

Therefore, this study proposes a web and mobile application that solves the abovementioned problems with some added features. The primary goal of this study is to develop a web-based and mobile application system for various music studios in Davao City, Philippines. The application displays all rehearsal studios in Davao City on a map. The app also locates and suggests the nearest path going to the rehearsal studio. Specifically, this study (1) creates a reservation system for rehearsal studios by utilizing PHP and Firebase, (2) locates the nearest rehearsal studios using Geolocation API, (3) allows the user to provide ratings and feedback comments by using Firebase, (4) enables rehearsal studio owner to register their business using Firebase, (5) generates a list of rehearsal studios by category using Rabin Karp Algorithm, (6) allows rehearsal studio owners to advertise their businesses online, and (7) allows freelance musicians to post hiring ads by using Firebase.

2. RELATED SYSTEMS

2.1 Locator Parking System Providing Variably Priced Parking Fees

In [1], a parking locator system is made that provides priced parking for more parking kiosks. The system is set to provide directions on a map to one or more unoccupied parking spaces in response to a request for vacant parking spaces from one or more mobile devices.

2.2 HUEQUITAS: Web Application for Location of Low-Cost Restaurants

The study presented the development and implementation of a web application for the geolocation of "Huequitas," a low-cost restaurant. The Google Maps API was used to improve the search experience of locating all the inexpensive restaurants along with Geolocation API to help the users find the easiest route to reach them [2].

2.3 Real-Time Location Extraction for Social-Media Events in Qatar

The study utilized Google APIs, particularly the Geolocation API, to request the geographic coordinates of the map location entities into geographic coordinates. The application uses the coordinates to monitor real-time traffic and geo-localized traffic-related incidents [3].

2.4 Location, Location, Location: The impact of Geolocation on Web Search Personalization

In this paper, a novel methodology to explore the effect of location on google search results was conducted. The related study uses the Javascript Geolocation API to present arbitrary GPS coordinates to the mobile version of Google search. Similar research targeted the mobile version of Google Search by overriding the Geolocation API in their PhantomJS Script. It can feed the coordinates specified on the command line to Google Search. Thus, giving the ability to run queries that will appear as if they are coming from any location, they have chosen [4].

2.5 Accelerating Rabin Karp on a Graphics Processing Unit (GPU) using Compute Unified Device Architecture (CUDA)

Rabin Karp is a string searching algorithm that can do a single pattern and multiple pattern matching. In this study, the Rabin Karp is used in plagiarism detection applications. It uses hashing to find out any given pattern strings of a text. Aside from plagiarism detection applications, Rabin Karp is also used in various applications like text editors, bioinformatics, spell checkers, network intrusion detection systems, and search engines [5].

2.6 Text Documents Plagiarism Detection using Rabin-Karp and Jaro-Winkler Distance Algorithm

In this paper, the Jaro-Winkler Distance algorithm and Rabin-Karp were used for detecting plagiarism in text documents. The Rabin-Karp algorithm has the ability to search for multiple string patterns, which is more effective than other string-matching algorithms [6].

3. METHODOLOGY

3.1 Conceptual Framework

Figure 1 shows the conceptual framework of the proposed system. As seen in the diagram, MusicHub is accessible through mobile and web-based platforms. Although each platform features similarities and differences in terms of design and functionalities, both utilize a centralized database architecture through Firebase.

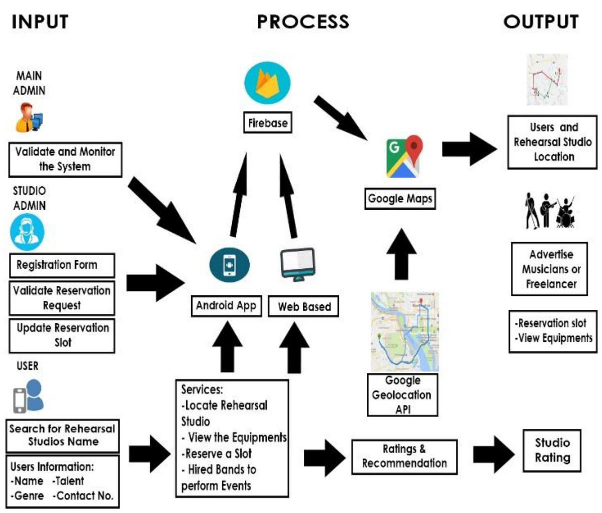

INPUT
PROCESS
OUTPUT
MAIN ADMIN
Validate and Monitor the System
STUDIO ADMIN
Registration Form
Validate Reservation Request
Update Reservation Slot
USER
Search for Rehearsal Studios Name
Users Information: -Name -Talent -Genre -Contact No.
Firebase
Android App
Web Based
Services: -Locate Rehearsal Studio - View the Equipments -Reserve a Slot - Hired Bands to perform Events
Google Maps
Google Geolocation API
Ratings & Recommendation
Users and Rehearsal Studio Location
Advertise Musicians or Freelancer
-Reservation slot -View Equipments
Studio Rating

Figure 1: Conceptual framework of the system

3.1.1 Input

To use the application, users need to access Wi-Fi or use mobile data to load the map and specify their current location. The users of the system need to sign-up and identify themselves as walk-ins, freelancers, or music studio owners. For walk-ins and freelancers, information such as first name, last name, gender, email address, phone number, username, and password are required. For music studio, owners, information such as the studio name, location, gender, email address, phone number, username, and password, among others, are required.

3.1.1.1 Walk-in Clients

After accomplishing the sign-up form and before signing-in to the application, walk-in accounts must be verified first by the administrator. After the verification, the clients can now use the application. Users can go to the dashboard, where they can view the different features of the app, such as the Search, View Music Studios, Show Nearby Studios, and Show Freelancers. The Search function is used for locating a specific studio. The View Music Studio function is where the user can view the information of a studio. Show Nearby Studio button is used for displaying nearby studios relevant to the location of the user. Lastly, the Show Freelancers function, which is for viewing the details of the freelancers. After choosing a studio, the clients can review specific information, such as the price per hour, studio equipment, rooms, reservation slot, or vacant slot by date, ratings, and feedback.

3.1.1.2 Music Studio Owner

Upon entering the system, the owner is required to upload a business permit, birth certificate, government id, whole-body picture, and images of their studio for verification. Once approved, owners can post information such as studio name, the price per hour, available time slots for reservations, studio introductions, and images for their studio rooms and available equipment.

3.1.1.3 Freelancers

After signing up and logging in to the system, freelancers are required to upload files such as birth certificate, government id, whole-body picture, and proof that the user is a freelancer for verification. Once verified, freelancers can navigate the system to post information such as freelance type, rate per hour, instruments, pictures, and self-introduction, among others.

3.1.2 Process

All the input data from the registration module are saved in an online database called Firebase. The real-time database allows the system to store, retrieve, and upload data. Since Firebase provides Firebase AUTH feature, customers can create an account using a personal email address or through linking their Facebook and Gmail accounts.

In finding music studio locations, the Google Geolocation API and Firebase are utilized. The Google Geolocation API provides a geographic map in the Android app. This API allows the mobile application to provide the shortest path or direction between the user’s current location and the music studio destination.

3.1.3 Output

Once the Rabin-Karp algorithm has matched the user's input in searching studios, the application displays a list of possible studio suggestions and recommendations. The app displays the information of the chosen studio to wit: studio name, location of the user and music studio, date, opening, and closing time. Customers can also reserve any of the available studio time slots and view rates, feedback comments, accessible rooms, instruments, and equipment. The users may also view available freelancers based on their talent, genre, and other additional information and the services provided by the freelancers.

3.2 Rabin-Karp Algorithm

The Rabin-Karp algorithm is a string searching algorithm that uses hashing to find patterns in strings. In the proposed system, the algorithm is used to generate a list of studios searched by category, preferred schedule, and the studio title description such as cozy, acoustic, and others. All categories are stored in an array with a multidimensional index ([StuidoId] [category]). The Rabin-Karp algorithm is used to locate the studio ID and generate results when the search matches the selected category of the user. Next is the studio title description; the Rabin-Karp looks for each title string based on the length of the search string or the design, e.g., (Pattern) Cozy, (Title) (1) Music Studio with Cozy atmosphere, (2) Acoustic Soundproofing Music Studio.

Lastly, the algorithm is used in finding the preferred schedule in the module. Customers input the time (From and To), and the Rabin-Karp algorithm performs a string search in every schedule in each studio. The time (From and To) is stored in a multidimensional array [0][1](From), [0][2](To) and the algorithm performs a string search in the second index of the array and generates a list of studio showing the exact inputted schedule [0][0](Studio ID). The pattern search process using the Rabin-Karp algorithm is shown in Figure 2.

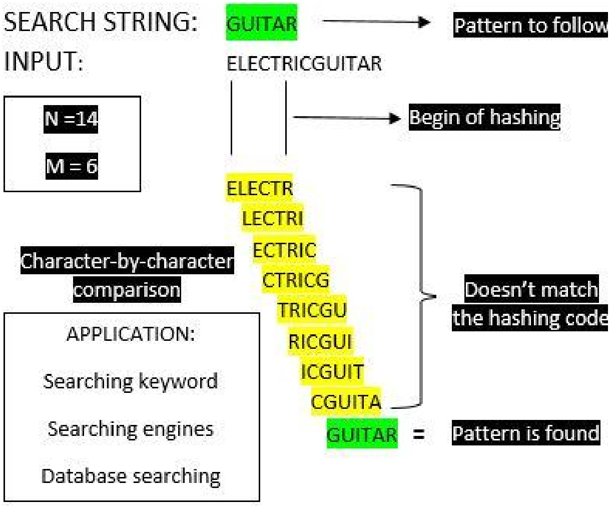


Figure 2: Pattern matching from the data entry using the Rabin-Karp algorithm

3.3 Use Case Diagram

Figure 3 shows the use case diagram of the system where system functionalities are accessed by different types of users in the application.

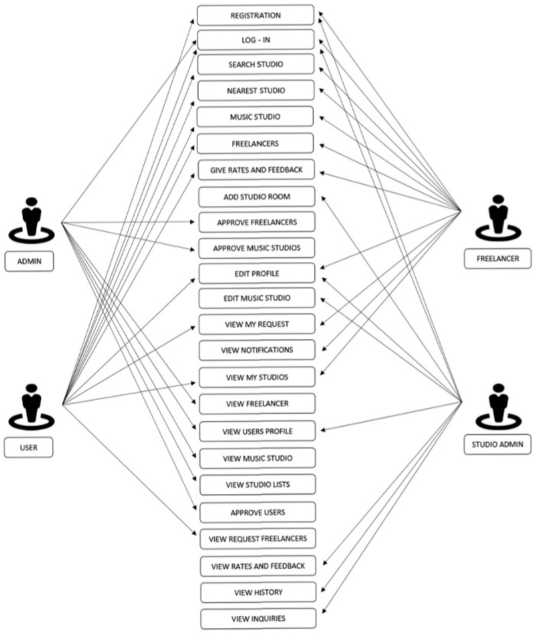


Figure 3: Use case diagram of the system

3.4 Development Tools

3.4.1 Android Studio

It is used for creating the user-interface and other functionalities of the mobile application.

3.4.2 Java Programming Language

The use of Java was observed for the application to be network centered since the MusicHub needs an Internet connection.

3.4.3 Google Geolocation API

The use of Geolocation was instrumental in accessing and locating the geographical data, which is implemented to the application's functionality. The app is able to display the source location of the user using the API. The Google Geolocation API serves as the API for mapping rehearsal studio locations. In general, the API gives a navigational path and suggests the shortest route towards the location of the rehearsal studio.

3.4.4 Google Maps Android API

The service API enables the user to access street view imagery, custom markers, and info window to provide the user with an easy way to locate the rehearsal studio.

3.4.5 Photoshop CS6

The Photoshop CS6 is used for photo editing, graphics designing, and digital imaging needed in the system. It is also used in system design and in the creation of the MusicHub logo.

3.4.6 PHP

The PHP programming was observed in developing the admin web application as PHP is specifically designed for web development. It is used for web crawling, adding, updating, deleting, and viewing data in the database

4. RESULTS AND DISCUSSION

4.1 System Design and Functionality Assessment

This section presents the functionality testing of the application. The app shows remarkable results upon development and testing, as shown in the indexed result in Tables 1-5. Hence, the objectives of the study are successfully met.

Table 1: Authentication module

Actions	Output	Results
Enter the username and password (password can be visible by clicking the eye icon).	If the inputs are correct, the application will proceed on the dashboard.	PASSED
	Incorrect inputs will remain on the login form.	PASSED
Click Sign-up to register a new user to the application.	Fill-up the text fields and choose which user type to register.	PASSED

Table 2: User module

Action	Expected Output	Actual Output	Results
Check the nearest studio location and user's location	The geolocation will show a map of the nearest studio basing on the user's location	The application did run smoothly without bugs and errors.	PASSED
Search for nearest studio	The application will show all registered studio	The application did run smoothly without bugs and errors.	PASSED
Rate Studio	The user can rate the studio with feedback comments	The application did run smoothly without bugs and errors.	PASSED
View Freelancer	All freelancer will show its profile	The application did run smoothly without bugs and errors.	PASSED
View My request	The user can view all his/her request on every studio/freelancer	The application did run smoothly without bugs and errors.	PASSED
View Request to Freelancers	The user can view the approved or declined a request from freelancer	The application did run smoothly without bugs and errors.	PASSED
View My Studios	The user can view the approved or declined a request from studios	The application did run smoothly without bugs and errors.	PASSED

Table 3: Admin module

Action	Expected Output	Actual Output	Results
View Studio	All new registered will be shown waiting for approval	The application did run smoothly without bugs and errors.	PASSED
View Freelancer	All new registered will be shown waiting for approval	The application did run smoothly without bugs and errors.	PASSED
View Users	All new registered will be shown waiting for approval	The application did run smoothly without bugs and errors.	PASSED
View Studio Lists	All new registered will be shown waiting for approval	The application did run smoothly without bugs and errors.	PASSED

Table 4: Studio owner module

Action	Expected Output	Actual Output	Result
Upload Photos like Government permits for authenticatio n	Government requirements like BIR, Government IDs, business permits, and others can be uploaded.	The application did run smoothly without bugs and errors.	PASSED
Upload Studio Rooms	Photos of the studio rooms can be uploaded for studio profile.	The application did run smoothly without bugs and errors.	PASSED
Input Studio Schedule and Price	The price and schedule will be shown on the studio profile.	The application did run smoothly without bugs and errors.	PASSED
View User's Reservation	Reservations of the users will show	The application did run smoothly without bugs and errors.	PASSED
View Inquiries	The studio admin can view inquiries	The application did run smoothly without bugs and errors.	PASSED
View Ratings and Feedback	Studio admin can view all the ratings and feedback of the users	The application did run smoothly without bugs and errors.	PASSED

Table 5: Freelancer's module

Action	Expected Output	Actual Output	Results
View Profile	Can view profiles such as name, contact number, etc.	The application did run smoothly without bugs and errors.	PASSED
View Notifications	Can view, accept, and decline user's requests.	The application did run smoothly without bugs and errors.	PASSED
Rate Studio	Can rate studio after their performance.	The application did run smoothly without bugs and errors.	PASSED
Search Music Studio	Can search and view music studios' information.	The application did run smoothly without bugs and errors.	PASSED
View Location and Nearest Music Studio	When accessing the application, the user can search the nearest music studio form its location.	The application did run smoothly without bugs and errors.	PASSED
View My Requests	Can view their request for the desired studio.	The application did run smoothly without bugs and errors.	PASSED
View My Studios	They can view the studios where they already had their performances.	The application did run smoothly without bugs and errors.	PASSED

Figure 4 shows the results of the search performed in the search bar. The algorithm delivered the data within 0.0006 seconds and showed 100 percent of the result. Further, Figure 5 shows the processing time results in Logcat, showing a speed ranging from 0.0001 to 0.0008 seconds. It fetches the index which the pattern found in the text and give the 100 percent result in the system.

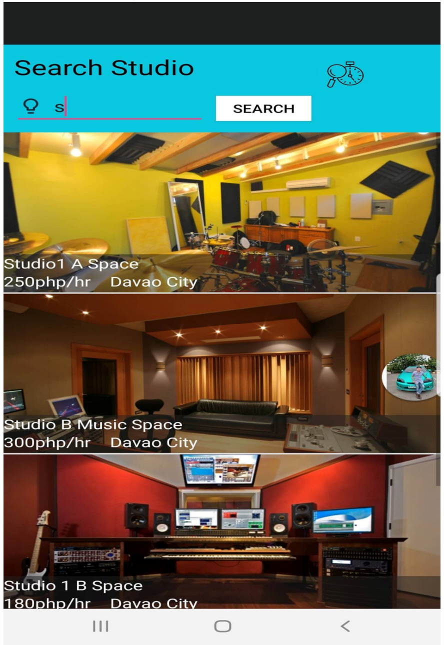
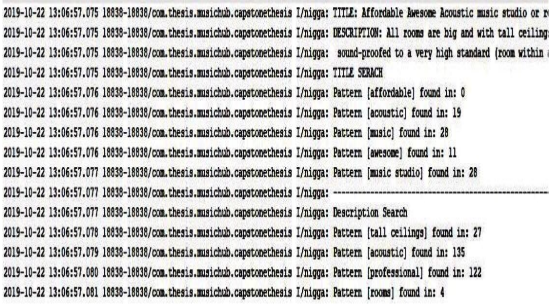


Figure 4: The search functionality of the system using the Rabin-Karp algorithm



```
2019-10-22 13:06:57.075 18838-18838/com.thesis.musichub.capstonethesis I/nigga: TITLE: Affordable Awesome Acoustic music studio or r
2019-10-22 13:06:57.075 18838-18838/com.thesis.musichub.capstonethesis I/nigga: DESCRIPTION: All rooms are big and with tall ceiling
2019-10-22 13:06:57.075 18838-18838/com.thesis.musichub.capstonethesis I/nigga:  sound-proofed to a very high standard (room within
2019-10-22 13:06:57.075 18838-18838/com.thesis.musichub.capstonethesis I/nigga: TITLE SERACH
2019-10-22 13:06:57.076 18838-18838/com.thesis.musichub.capstonethesis I/nigga: Pattern [affordable] found in: 0
2019-10-22 13:06:57.076 18838-18838/com.thesis.musichub.capstonethesis I/nigga: Pattern [acoustic] found in: 19
2019-10-22 13:06:57.076 18838-18838/com.thesis.musichub.capstonethesis I/nigga: Pattern [music] found in: 28
2019-10-22 13:06:57.076 18838-18838/com.thesis.musichub.capstonethesis I/nigga: Pattern [awesome] found in: 11
2019-10-22 13:06:57.077 18838-18838/com.thesis.musichub.capstonethesis I/nigga: Pattern [music studio] found in: 28
2019-10-22 13:06:57.077 18838-18838/com.thesis.musichub.capstonethesis I/nigga: ----------------------------------------------------
2019-10-22 13:06:57.077 18838-18838/com.thesis.musichub.capstonethesis I/nigga: Description Search
2019-10-22 13:06:57.078 18838-18838/com.thesis.musichub.capstonethesis I/nigga: Pattern [tall ceilings] found in: 27
2019-10-22 13:06:57.079 18838-18838/com.thesis.musichub.capstonethesis I/nigga: Pattern [acoustic] found in: 135
2019-10-22 13:06:57.080 18838-18838/com.thesis.musichub.capstonethesis I/nigga: Pattern [professional] found in: 122
2019-10-22 13:06:57.081 18838-18838/com.thesis.musichub.capstonethesis I/nigga: Pattern [rooms] found in: 4
```

Figure 5: Rabin-Karp algorithm performance

5. CONCLUSION

In this study, the proposed web and mobile applications have successfully met the users' requirements and accomplished the objectives after a series of testing, verification, and validation were conducted. With the use of Google Maps API and Rabin-Karp algorithm, users can conveniently locate and book music studios in Davao City. On the other hand, music studio owners can register their studios, while freelancers can advertise their services. Further, the system was successfully deployed in two studios in Davao City, Philippines, namely Studio 1 and State of Mind Production Inc.

REFERENCES

[1] R. Rowe and J. Fiorucci, "Parking locator system providing variably priced parking fees," 2014.

[2] L. Enciso, D. Nodine, E. Cueva, P. A. Quezada-Sarmiento, and E. Zelaya-Policarpo, "HUEQUITAS: Web Application for Location of

Low Cost Restaurants," in *WorldCIST'18*, 2018, vol. 746, pp. 98–106. https://doi.org/10.1007/978-3-319-77712-2_10

[3] S. Abbar, N. Al Amadi, F. J. Guzman, F. Sebastiani, and J. Borge-holthoefer, "Real-Time Location Extraction for Social-Media Events in Qatar," in *Qatar Foundation Annual Research Conference Proceedings 2016: ICTPP3046*, 2016. https://doi.org/10.5339/qfarc.2016.ICTPP3046

[4] C. Kliman-Silver, A. Hannak, D. Lazer, C. Wilson, and A. Mislove, "Location, location, location: The impact of geolocation on web search personalization," in *Internet Measurement Conference*, 2015, pp. 121–127. https://doi.org/10.1145/2815675.2815714

[5] N. Dayarathne and R. Ragel, "Accelerating Rabin Karp on a Graphics Processing Unit (GPU) using Compute Unified Device Architecture (CUDA)," in *7th International Conference on Information and Automation for Sustainability*, 2014. https://doi.org/10.1109/ICIAFS.2014.7069589

[6] B. Leonardo and S. Hansun, "Text documents plagiarism detection using Rabin-Karp and Jaro-Winkler distance algorithms," *Indones. J. Electr. Eng. Comput. Sci*., vol. 5, no. 2, pp. 462–471, 2017. https://doi.org/10.11591/ijeecs.v5.i2.pp462-471