International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

EBUS-Rout Direction

Sachin Virendra Dubey[1], *Vikas Bindoo Chauhan*[2]

[1] Student, Department of Computer Science, Faculty of MCA, ASM IMCOST, Thane, Maharashtra, INDIA
[2] Student, Department of Computer Science, Faculty of MCA, ASM IMCOST, Thane, Maharashtra, INDIA
Email id: sachin.dubey21@asmedu.org , vikas.chauhan21@asmedu.org

ABSTRACT :

The main perspective of this documentation is to create or improve the M-indicator. The M-indicator System is a web-based design to improve flexible vast services to provide information about Mumbai's local train, bus, and metro schedules, as well as other essential information, improving the difficulties which will be faced by users, Mumbaikars people face as it simplifies their daily commute.

The M-indicator app's bus map feature needs improvement to make it more user-friendly and efficient. This documentation shows how easy is to view the map feature for individuals by storing it on the web. The browser-based platform seeks to limit admin and eliminate manual techniques, In this research paper, we will explore the current issues with the bus map feature and propose ways to improve it. E-BUS is a web-based management framework that empowers Map-controllers to deal with cluttered interfaces and difficult-to-navigate buses in M-indicator.

The bus map feature's interface needs to be simplified and made more user-friendly. To make the bus map feature more useful, real-time updates should be provided. This can be achieved by integrating the app with GPS trackers on the buses.

Users should be able to search for bus routes by number, starting point, and destination. The search results should show the different bus routes available and their schedules. users to access their most frequently used bus routes. Users should be able to save their favorite bus routes and access them quickly from the main menu.

integrating with Google Maps. This will allow users to plan their journey using a combination of local trains, buses, and metro, and get directions in real-time. Additionally, users should be able to search for bus stops and get real-time updates on the buses that stop there.

Users should be able to see the safety rating for each bus route and get information on how to stay safe while traveling. Additionally, users should be able to report safety concerns and provide feedback to improve the safety of the bus routes

Introduction

The introduction section of the research paper on "EBUS: Rout direction" is to create or improve the M-indicator. The M-indicator System is a web-based design to improve flexible vast services to provide information about Mumbai's local train, bus, and metro schedules, as well as other essential information, improving the difficulties which will be faced by users, Mumbaikars people face as it simplifies their daily commute. The M-indicator app's bus map feature needs improvement to make it more user-friendly and efficient. This documentation shows how easy is to view the map feature for individuals by storing it on the web. The browser-based platform seeks to limit admin and eliminate manual techniques, In this research paper, we will explore the current issues with the bus map feature and propose ways to improve it. E-BUS is a web-based management framework that empowers Map-controllers to deal with cluttered interfaces and difficult-to-navigate buses in M-indicator.

The current bus map feature in the M-indicator app has several issues. Firstly, the interface is cluttered and difficult to navigate. There is clustered content on the screen, which will be overwhelming for users to find what they need. Secondly, the map is not user-friendly as it does not show the bus routes in a clear and concise manner. This makes it challenging for users to plan their route and understand the different bus routes available to them. Finally, the current bus map feature lacks real-time updates, which means users cannot see if a bus is delayed or if there is a route change due to traffic or other reasons.

Technology

Frontend:

1. HTML and CSS: HTML will be used for defining the structure and content of the page, while CSS will be used for styling the page.

2. JavaScript: JavaScript will be used for adding interactivity and dynamic functionality to the page.

Backend:

1. ASP.NET: ASP.NET is a web application framework developed by Microsoft for building dynamic web applications.

2. Web API: Web API is a framework provided by ASP.NET for building RESTful web services that can be accessed using HTTP protocols.

Database:

1. SQL Server: SQL Server is a relational database management system (RDBMS) developed by Microsoft for storing and retrieving data.

2. SQL Query Language: SQL is a standard language used for querying and manipulating data in relational databases.

Programming Languages:

1. C#: C# is a programming language developed by Microsoft that is used to build web applications in ASP.NET.

2. JavaScript: JavaScript can be used for adding interactivity and dynamic functionality to the page.

In summary, the frontend of the web-based project will be built using HTML, CSS, JavaScript, and jQuery, while the backend will be built using ASP.NET and Web API. The data will be stored in SQL Server and accessed using SQL query language. The programming languages used will be C# for the backend and JavaScript for the frontend.

Problem Statement:

The problem statement section defines and outlines the specific improve the M-indicator. The M-indicator System is a web-based design to improve flexible vast services to provide information about Mumbai's local train, bus, and metro schedules, as well as other essential information, improving the difficulties which will be faced by users, Mumbaikars people face as it simplifies their daily commute. The M-indicator app's bus map feature needs improvement to make it more user-friendly and efficient. This documentation shows how easy is to view the map feature for individuals by storing it on the web. The browser-based platform seeks to limit admin and eliminate manual techniques

Problem Methodology

Improvements to the Bus Map Feature:

To improve the bus map feature in the M-indicator app, we propose the following changes:

1. Redesign the interface: The bus map feature's interface needs to be simplified and made more user-friendly. This can be achieved by reducing the clutter on the screen and organizing the information in a more logical manner. The map should be the focus of the screen, and other information such as bus routes and schedules should be presented in a clear and concise manner.

2. Show bus routes more clearly: Currently, the bus routes on the map are not easy to understand. To address this, the map should show the bus routes in a clear and concise manner. This can be achieved by using different colors and line thicknesses to differentiate between different bus routes. Additionally, the bus routes should be labeled with their respective numbers to make it easier for users to identify them.

3. Provide real-time updates: To make the bus map feature more useful, real-time updates should be provided. This can be achieved by integrating the app with GPS trackers on the buses. Users should be able to see the location of the bus they are waiting for and get updates on delays or route changes.

4. Introduce a search feature: A search feature should be introduced in the bus map feature to make it easier for users to find the bus routes they need. Users should be able to search for bus routes by number, starting point, and destination. The search results should show the different bus routes available and their schedules.

5. Introduce a favorites feature: A favorites feature should be introduced in the bus map feature to make it easier for users to access their most frequently used bus routes. Users should be able to save their favorite bus routes and access them quickly from the main menu.

6. Integrate with Google Maps: The bus map feature can be further improved by integrating with Google Maps. This will allow users to plan their journey using a combination of local trains, buses, and metro, and get directions in real-time. The integration will also help users see the different transport options available to them and choose the most efficient route for their journey.

7. Add information on bus stops: The bus map feature can be enhanced by adding information about bus stops. Users should be able to see the bus stops on the map and get details about the buses that stop at each stop. Additionally, users should be able to search for bus stops and get real-time updates on the buses that stop there.

8. Provide feedback feature: A feedback feature can be introduced in the bus map feature to enable users to share their experiences and suggest improvements. Users should be able to rate their journey and leave feedback on the bus route, schedule, and overall experience. This feedback will help improve the bus map feature and make it more user-friendly.

9. Gamification: Gamification can be used to motivate users to use the bus map feature more frequently. Users can earn points or rewards for using the app to plan their journey, take a bus instead of driving, or provide feedback. This will encourage users to make sustainable choices and contribute to reducing traffic congestion in Mumbai.

10. Integration with other modes of transport: The bus map feature can be integrated with other modes of transport, such as bike-sharing and car-sharing services. This will provide users with a complete solution for their journey planning needs and encourage sustainable modes of transport.

11. In-app advertising: In-app advertising can be used to generate revenue and provide users with relevant information. The ads can be targeted to users based on their journey history, location, and preferences. This will also help local businesses reach their target audience and promote sustainable modes of transport.

12. Safety information: The bus map feature can be further enhanced by adding safety information for users. Users should be able to see the safety rating for each bus route and get information on how to stay safe while traveling. Additionally, users should be able to report safety concerns and provide feedback to improve the safety of the bus routes.

13. Integration with emergency services: The bus map feature can be integrated with emergency services to ensure the safety of users. Users should be able to contact emergency services in case of an emergency and get real-time updates on the status of their request.

14. Multi-language support: Multi-language support can be added to the bus map feature to cater to users from different backgrounds. Users should be able to choose their preferred language and get information about bus routes and schedules in their language.

Proposed Algorithm:

For Redesign the interface:

Step 1: Analyze user feedback and conduct user research to identify pain points and areas of improvement

Step 2: Simplify the user interface by removing unnecessary elements and reducing clutter.

Step 3: Organize information logically, ensuring the map is the central focus of the screen.

Step 4: Implement responsive design principles to optimize the interface for different screen sizes and devices.

For Show bus routes more clearly:

Step 1: Retrieve bus route data from the backend database or API.

Step 2: Implement a visualization algorithm to draw bus routes on the map using different colors and line thicknesses.

Step 3: Label each bus route with its respective number for easy identification.

For Provide real-time updates:

Step 1: Integrate GPS trackers on buses to obtain real-time location data.

Step 2: Implement a mechanism to fetch and update bus locations in real time.

Step 3: Calculate and display estimated arrival times based on the current bus locations and historical data.

For Search Feature:

Step 1: Design a search functionality that allows users to input bus route numbers, starting points, and destinations.

Step 2: Implement a search algorithm to query the backend database or API for relevant bus route information.

Step 3: Display search results with available bus routes, schedules, and other relevant details

For User favorites feature:

Step 1: Implement a user authentication system to allow users to create accounts and save their favorite bus routes.

Step 2: Store user preferences and favorites in a user profile database or secure storage.

Step 3: Provide a user interface for managing and accessing favorite bus routes.

Performance Analysis:

The performance analysis section evaluates the effectiveness and efficiency of the proposed methodology and algorithm. It involves conducting experiments or simulations to measure various performance metrics such as response time, throughput, resource utilization, and scalability. The analysis aims to assess the performance impact of implementing the proposed designed interface,bus routes on the map, GPS integration, search algorithm for retrieving bus routes based on different criteria (number, starting point, destination), user authentication system, and the storage mechanism for saving and retrieving favorite bus routes and impact of integrating with the Google Maps API and retrieving directions and real-time data, in a system environment. The results of the performance analysis provide insights into the strengths and limitations of the proposed methodology and algorithm, helping organizations make informed decisions about their adoption.

Conclusion:

Improving the proposed improvements to the bus map feature in the M-indicator app offer several enhancements to improve user experience, functionality, and performance. The specific performance analysis would require a more detailed examination based on the app's architecture, technology stack, and anticipated user load. However, by following best practices for performance analysis, such as setting performance goals, conducting load testing, monitoring performance metrics, and optimizing the system, it is possible to ensure that the app meets the desired performance standards.

Some key areas to consider for performance analysis include the responsiveness of the redesigned interface, efficient rendering of bus routes on the map, real-time updates with GPS integration, search functionality, favorites feature, integration with Google Maps, and overall scalability under different loads. By thoroughly analyzing and optimizing these areas, the app can provide users with a fast, reliable, and user-friendly bus map feature.

It's important to conduct iterative testing and validation throughout the development process to identify and address any performance issues or bottlenecks. Continuous monitoring in the production environment is also crucial to proactively detect and resolve performance regressions or emerging issues.

By prioritizing performance analysis and optimization, the M-indicator app can deliver a highly functional and performant bus map feature, enhancing the overall user experience and satisfaction.

References:

- Abualigah, L. M., Al-Ayyoub, M., & Jararweh, Y. (2017). Smart Public Transportation System: Design and Implementation of a Mobile Application for Bus Transit. Procedia Computer Science, 110, 231-238.
- Chen, T., & Ren, F. (2018). Design of Bus Route Optimization System Based on Mobile Internet. 2018 IEEE International Conference on Mechatronics and Automation (ICMA), 1198-1202.
- Gao, Y., & Liu, X. (2017). Design of a Mobile Application for Bus Route Optimization. 2017 International Conference on Industrial Design Engineering and Creative Communication (IDEC), 13-17.
- Khairnar, R., & Patel, R. (2016). A Survey of Mobile Based Application for Public Transport System. International Journal of Innovative Research in Computer and Communication Engineering, 4(8), 131-136.
- Zheng, Y., Liu, L., & Zhang, C. (2021). An Intelligent Bus Transit Information Platform Based on Mobile Internet. Journal of Physics: Conference Series, 1819(1), 012016.