UPRIGHT PUBLICATIONS
https://upright.pub/index.php/ijrstp/

Original Contribution

Optimizing Web Performance: Front End Development Strategies for the Aviation Sector

Rahimoddin Mohammed[1], Srinivas Addimulam[2], Manzoor Anwar Mohammed[3], Raghunath Kashyap Karanam[4], Sai Sirisha Maddula[5], Prasanna Pasam[6], Vineel Mouli Natakam[7]

Keywords: Web Performance, Front End, Optimization, Aviation Sector, User Experience, Caching, CDN

International Journal of Reciprocal Symmetry and Theoretical Physics

Vol. 4, Issue 1, 2017 [Pages 38-45]

This study investigates front end development strategies to optimize web performance for airline websites, aiming to enhance user experience and operational efficiency. The primary objectives include identifying critical performance metrics and implementing effective optimization techniques. The methodology involved analyzing key metrics such as page load time, TTFB, FCP, LCP, CLS, TBT, and TTI, followed by the deployment of best practices like minimizing HTTP requests, optimizing images, leveraging lazy loading, and using content delivery networks (CDNs). Continuous monitoring and user feedback were integral to the iterative improvement process. Principal findings indicate that these strategies significantly improve web performance, resulting in faster load times, reduced latency, and enhanced user satisfaction. Ensuring HTTPS implementation and robust security measures further bolstered performance and trust. The study concludes that a holistic approach to web performance optimization can provide airlines with a competitive edge, fostering better customer experiences and driving business success.

INTRODUCTION

In the rapidly evolving digital landscape, the aviation industry faces unique challenges and opportunities in enhancing web performance. With the increasing reliance on online platforms for booking, customer service, and information dissemination, airlines must prioritize web performance to ensure a seamless user experience (RA?UL et al., 2015). The introduction of robust front end development strategies is crucial to achieving this goal, as it directly impacts the speed, responsiveness, and overall functionality of airline websites (Aviation sector needs a sensible policy, 2014). This chapter delves into the importance of optimizing web performance in the aviation sector and explores various front end development strategies that can be employed to achieve this.

Airlines operate in a highly competitive environment where customer satisfaction is paramount. A slow or unresponsive website can lead to frustration, abandoned bookings, and a tarnished brand reputation. Conversely, a fast, efficient, and secure website enhances user experience, fosters customer loyalty, and drives revenue growth. In this context, optimizing web performance is not merely a technical necessity but a strategic imperative for airlines looking to maintain a competitive edge.

[1] Lead .Net Developer, Cognitiv, 8100 OLD Georgetown, RD Bethesda, MD 20814, USA [rahimoddinm501@gmail.com]
[2] Project Manager, Wipro (Best Buy), USA [addimulam.srinivas@gmail.com]
[3] Oracle Applications Developer, Brake Parts Inc., 4400 Prime Pkwy, McHenry, IL – 60050, USA [manzooranwarm@gmail.com]
[4] Senior Associate Consultant CISCO, Infosys Ltd., Cisco Systems, CA 95134 USA [raghunathkaranam9@gmail.com]
[5] Independent Researcher, USA [saigc94@gmail.com]
[6] Independent Researcher, USA [prasannapasam82@gmail.com]
[7] Independent Researcher, USA [natakamvineel@gmail.com]

The aviation sector presents distinct challenges for web performance optimization. Airline websites must handle high traffic volumes, especially during peak travel seasons and promotions (Bekavac & Pranicevic, 2015). They must also integrate complex functionalities such as real-time flight updates, booking engines, loyalty programs, and customer support services. Ensuring that these features operate smoothly and efficiently requires a multifaceted approach to front end development that balances performance with functionality and security.

One of the primary factors influencing web performance is the efficiency of the front end code. This includes HTML, CSS, JavaScript, and other assets that are rendered in the user's browser. Efficient coding practices, such as minimizing file sizes, optimizing images, and reducing the number of HTTP requests, can significantly enhance page load times and responsiveness (Parks et al., 2016). Furthermore, leveraging modern front end frameworks and libraries can streamline development processes and improve performance.

In addition to coding practices, the implementation of performance optimization techniques such as lazy loading, code splitting, and caching can further enhance the user experience. Lazy loading defers the loading of non-critical resources until they are needed, reducing initial load times. Code splitting breaks down large code bundles into smaller, manageable chunks that can be loaded on demand. Caching stores frequently accessed resources locally, reducing server load and accelerating page rendering.

Security is another critical aspect of front end development in the aviation sector. Airline websites must safeguard sensitive customer information and ensure compliance with industry regulations. Implementing robust security measures, such as HTTPS encryption, secure cookies, and content security policies, is essential to protecting user data and maintaining trust.

This chapter will explore these strategies in detail, providing insights and best practices for optimizing web performance in the aviation sector. By adopting a holistic approach to front end development, airlines can deliver a superior online experience that meets the demands of today's digital-savvy travelers. Through case studies and practical examples, this chapter aims to equip developers and industry professionals with the knowledge and tools needed to enhance web performance and drive business success in the competitive aviation landscape.

STATEMENT OF THE PROBLEM

The aviation industry, heavily reliant on digital platforms for booking, customer service, and real-time updates, faces significant challenges in optimizing web performance. While backend optimizations and network enhancements have received considerable attention, there is a notable gap in targeted studies and practical guidelines for front end development strategies specific to airline websites (Mishra & Liu, 2014). Current research largely overlooks the unique needs of these websites, which must handle high traffic volumes, integrate complex functionalities, and ensure robust security. This gap underscores the need for a comprehensive exploration of front end development techniques to enhance web performance and user experience in the aviation sector.

Objectives of the study if to identify and analyze critical front end performance metrics for airline websites. Propose and validate specific front end development strategies to optimize web performance.

Research Question:

How can front end development strategies be optimized to enhance web performance and user experience in the aviation sector?

The significance of this study is multifaceted. Firstly, it addresses the direct impact of web performance on user experience. Airline customers expect fast, responsive, and secure online interactions, and any delay or malfunction can lead to frustration, abandoned bookings, and a negative perception of the airline's brand. By optimizing front end performance, airlines can significantly improve the user experience, leading to higher customer satisfaction and loyalty (Chauhan & Manhas, 2013).

Moreover, enhanced web performance provides a competitive advantage. In an industry where numerous airlines vie for the same customer base, a superior digital experience can be a key differentiator. Airlines that invest in advanced front end development strategies can set themselves apart, attracting more customers and increasing their market share.

Operational efficiency is another critical benefit. Optimized web performance reduces server load, minimizes downtime, and enhances scalability, which is particularly important during peak travel seasons (Korsakas & Charlamov, 2015). This not only improves the reliability of the airline's digital services but also lowers operational costs associated with maintaining and upgrading IT infrastructure.

Security is a crucial aspect intertwined with front end development. Airline websites handle sensitive customer data, including personal information and payment details. Ensuring robust front end security measures, such as HTTPS encryption, secure cookies, and content security policies, protects this data and maintains compliance with industry regulations (Karabetsos et al., 2015). This builds customer trust and safeguards the airline's reputation.

To address these issues, this study will identify key performance metrics that affect user experience on airline websites and evaluate current front end development practices. Through empirical assessments and performance benchmarking, the study will propose and validate strategies such as efficient coding practices, performance optimization techniques like lazy loading and code splitting, and integration of robust security measures.

By providing practical recommendations and actionable insights, this study aims to equip developers and industry professionals with the tools needed to enhance web performance in the aviation sector. Ultimately, the findings will contribute to creating faster, more secure, and user-friendly airline websites, fostering better customer experiences and driving business success in the competitive aviation market.

KEY METRICS FOR WEB PERFORMANCE

In the aviation sector, optimizing web performance is crucial to ensure that airline websites meet the high expectations of modern travelers. Key performance metrics provide measurable insights into how well a website is performing and highlight areas that require improvement (Tiwari et al., 2012). Understanding these metrics is the first step towards developing effective front end strategies. This chapter explores the essential web performance metrics that are particularly relevant for the aviation industry and explains their significance.

1. **Page Load Time:** Page load time is a critical metric that measures the duration from the moment a user requests a page until it is fully loaded in the browser. In the context of airline websites, where users often make time-sensitive decisions such as booking flights and checking in, a slow page load can result in user frustration and lost business. Studies have shown that users expect a page to load within two to three seconds, and any delay beyond this can significantly increase the bounce rate.

2. **Time to First Byte (TTFB):** TTFB measures the time it takes for the browser to receive the first byte of data from the server after a request is made. This metric is crucial because it reflects the server's responsiveness. For airline websites, a low TTFB ensures that users quickly see the initial content of the page, which can enhance their perception of speed and reliability. High TTFB values can indicate server-side issues that need to be addressed to improve overall performance.

3. **First Contentful Paint (FCP):** First Contentful Paint measures the time from when the page starts loading to when any part of the page's content is rendered on the screen. This metric is significant because it marks the point at which users perceive the page to start loading (Havranek et al., 2014). For airlines, where users may be looking for quick updates on flight status or booking confirmations, a faster FCP can improve user engagement and satisfaction.

4. **Speed Index:** Speed Index is a metric that shows how quickly the contents of a page are visibly populated. It provides a comprehensive view of the loading experience from the user's perspective. A lower Speed Index indicates that the page content is loading quickly and smoothly. For airline websites, achieving a low Speed Index is crucial to keep users engaged and reduce the likelihood of them abandoning the site due to perceived slowness.

5. **Largest Contentful Paint (LCP):** Largest Contentful Paint measures the time it takes for the largest content element on the page

to become visible within the viewport. This could be a large image, video, or block of text. For airline websites, a quick LCP is essential as it ensures that the most important elements, such as flight options and booking forms, are visible to the user as soon as possible. Google recommends an LCP of 2.5 seconds or faster for a good user experience.

6. **Cumulative Layout Shift (CLS):** Cumulative Layout Shift quantifies how much the page layout shifts during the loading phase. This is particularly important for airline websites where users interact with forms and buttons. Unexpected layout shifts can lead to poor user experience, such as accidentally clicking the wrong button or entering incorrect information. Maintaining a low CLS ensures that the page remains stable and predictable as it loads.

7. **Total Blocking Time (TBT):** Total Blocking Time measures the total amount of time that a page is blocked from responding to user input. This includes the time taken by JavaScript to execute and other resources to load. For airline websites, minimizing TBT is crucial as it ensures that users can interact with the page without delays. A lower TBT contributes to a more responsive and interactive user experience.

8. **Time to Interactive (TTI):** Time to Interactive measures how long it takes for a page to become fully interactive. This means the page has loaded enough resources for the user to interact with it, such as clicking buttons or entering data into forms. For airline websites, a faster TTI is vital as it allows users to complete their tasks quickly, enhancing their overall experience.

Understanding and optimizing these key metrics can significantly enhance the performance of airline websites. By focusing on improving page load times, reducing TTFB, ensuring quick FCP and LCP, maintaining low CLS and TBT, and achieving faster TTI, airlines can provide a superior digital experience to their users. These metrics not only help in identifying performance bottlenecks but also guide developers in implementing effective front end strategies that meet the high expectations of today’s travelers.

EFFECTIVE FRONT END DEVELOPMENT PRACTICES

Optimizing web performance is essential for airline websites to provide a seamless user experience, handle high traffic volumes, and ensure security (Paterno et al., 2012). Effective front end development practices play a crucial role in achieving these goals. This chapter explores a range of best practices that can significantly enhance web performance for the aviation sector.

1. Minimize HTTP Requests: Each element on a web page, such as images, scripts, and stylesheets, requires an HTTP request. Reducing the number of these requests can significantly speed up page load times. Techniques to minimize HTTP requests include combining files (e.g., using CSS sprites), reducing the number of images, and using inline images for small graphics. Additionally, leveraging browser caching can help by storing frequently accessed files locally, reducing the need for repeated requests.

2. Optimize Images: Images are often the largest assets on a web page and can slow down loading times if not optimized. Compressing images without sacrificing quality can significantly reduce file sizes. Tools like TinyPNG and ImageOptim can help in this process. Additionally, using modern image formats such as WebP can offer better compression rates than traditional formats like JPEG and PNG. Implementing responsive images with the srcset attribute ensures that the correct image size is served based on the user's device, further improving performance.

3. Implement Lazy Loading: Lazy loading defers the loading of non-critical resources until they are needed. For instance, images and videos below the fold can be loaded as the user scrolls down the page. This technique reduces initial page load times and conserves bandwidth, enhancing the overall user experience. Lazy loading can be implemented using native HTML attributes (loading="lazy") or JavaScript libraries.

4. Optimize CSS and JavaScript: Efficiently written CSS and JavaScript can greatly improve web performance. This includes:

- **Minification:** Removing unnecessary characters, such as whitespaces and comments, from CSS and JavaScript files reduces their size and improves load times.

- **Code Splitting:** Breaking down large code bundles into smaller chunks that can be loaded on demand ensures that only the necessary code is loaded initially, reducing the initial load time.
- **Deferred Loading:** Using the defer and async attributes for script tags can prevent render-blocking and ensure that scripts are loaded asynchronously or after the HTML document has been parsed.

5. Use Content Delivery Networks (CDNs): CDNs distribute content across multiple servers located around the world. By serving content from a server closest to the user, CDNs reduce latency and improve load times. For airline websites with a global audience, using a CDN can significantly enhance performance by ensuring faster content delivery.

6. Implement Caching Strategies: Effective caching strategies can drastically reduce load times by storing copies of resources locally. Browser caching, for example, stores static resources like images, stylesheets, and scripts on the user's device, allowing subsequent visits to load faster. Additionally, server-side caching can be employed to store dynamic content, reducing the load on the server and speeding up response times.

7. Prioritize Critical Rendering Path: The critical rendering path refers to the sequence of steps the browser takes to render a web page. Optimizing this path ensures that the most important content is rendered as quickly as possible. Techniques include:

- **Inlining Critical CSS:** Embedding essential CSS directly in the HTML document can speed up the rendering of above-the-fold content.
- **Deferring Non-Critical Resources:** Ensuring that non-critical CSS and JavaScript files are loaded after the initial render can improve perceived performance.

8. Reduce Redirects: Each redirect creates additional HTTP requests and adds latency, slowing down the overall page load time. Minimizing redirects, especially on critical pages like the homepage and booking flow, can improve performance. Ensuring clean and direct URL structures helps eliminate unnecessary redirects.

9. Ensure Security with HTTPS: Security is a key concern for airline websites, which handle sensitive customer data. Implementing HTTPS ensures that data transmitted between the user and the server is encrypted and secure. Additionally, modern browsers give performance benefits to HTTPS-enabled websites, such as HTTP/2 support, which offers improved performance through multiplexing, header compression, and server push.

10. Monitor and Analyze Performance: Continuous monitoring and analysis are crucial for maintaining optimal web performance. Tools like Google Lighthouse, WebPageTest, and Chrome DevTools provide detailed insights into performance metrics and highlight areas for improvement. Regular performance audits help identify and address bottlenecks, ensuring that the website remains fast and efficient.

By adopting these effective front end development practices, airlines can enhance web performance, providing users with a faster, more responsive, and secure online experience. These practices not only improve user satisfaction but also contribute to higher conversion rates, customer loyalty, and overall operational efficiency.

IMPLEMENTING AND EVALUATING OPTIMIZATION STRATEGIES

In the aviation sector, where digital interactions are pivotal to customer satisfaction and business efficiency, implementing and evaluating web performance optimization strategies is essential. This chapter outlines a structured approach to deploying these strategies and assessing their impact on airline websites.

1. Planning and Prioritization: The first step in implementing optimization strategies is thorough planning and prioritization. This involves identifying the key performance metrics that need improvement, understanding the current performance baseline, and setting realistic goals. For airline websites, critical metrics often include page load time, time to first byte (TTFB), first contentful paint (FCP), largest contentful paint (LCP), and cumulative layout shift (CLS).

Prioritization is crucial, as not all optimization techniques will have the same impact. High-priority areas typically include:

- Reducing page load times for key user journeys, such as booking flights and checking in.
- Enhancing the responsiveness and interactivity of the website.
- Improving the stability and security of the front end.

2. Implementing Optimization Techniques: Once priorities are set, the next step is implementing the chosen optimization strategies. Here are some key techniques and how they can be applied to airline websites:

- **Code Optimization:** Minify CSS, JavaScript, and HTML files to reduce their size and load times. Code splitting can help by breaking down large scripts into smaller, manageable chunks that load as needed.
- **Image Optimization:** Compress and resize images using tools like TinyPNG or ImageOptim. Use responsive images with the srcset attribute to ensure the correct image size is served based on the user's device.
- **Lazy Loading:** Implement lazy loading for images and videos to defer the loading of non-essential content until it is needed. This can significantly improve initial load times and save bandwidth.
- **Caching Strategies:** Utilize browser caching to store static resources locally on the user's device and server-side caching to store dynamic content. Set appropriate cache headers to ensure resources are reused effectively.
- **Content Delivery Networks (CDNs):** Deploy a CDN to distribute content across multiple servers worldwide, reducing latency and improving load times for users across different regions.
- **Critical Rendering Path Optimization:** Prioritize the loading of above-the-fold content by inlining critical CSS and deferring non-critical JavaScript. This ensures that the most important content is rendered quickly, enhancing perceived performance.
- **HTTPS Implementation:** Ensure that the website uses HTTPS to secure data transmission and leverage performance benefits like HTTP/2, which supports multiplexing, header compression, and server push.

3. Testing and Validation: After implementing the optimization strategies, rigorous testing and validation are necessary to ensure they have the desired effect. This involves:

- **Performance Testing:** Use tools like Google Lighthouse, WebPageTest, and Chrome DevTools to measure performance metrics before and after optimization. These tools provide detailed reports on load times, resource usage, and potential bottlenecks.
- **User Testing:** Conduct usability tests to gather feedback from real users. This helps identify any issues that automated tools might miss and provides insights into how the optimizations impact user experience.
- **A/B Testing:** Implement A/B testing to compare the performance of the optimized version of the website against the original version. This can help quantify the improvements and understand their impact on user behavior.

4. Continuous Monitoring and Iteration: Web performance optimization is not a one-time task but an ongoing process. Continuous monitoring and iteration are essential to maintain and improve performance over time. Key activities include:

- **Performance Monitoring:** Use monitoring tools to track performance metrics in real-time and set up alerts for any significant deviations. This helps quickly identify and address performance issues as they arise.
- **Regular Audits:** Conduct regular performance audits to reassess the website's performance and identify new optimization opportunities. This is particularly important after major updates or changes to the website.
- **User Feedback:** Collect and analyze user feedback to understand their experience and identify areas for improvement. This can be done through surveys, feedback forms, and user testing sessions.

5. Evaluating the Impact: Finally, evaluating the impact of the optimization strategies involves analyzing both quantitative and qualitative data. Key evaluation criteria include:

- **Improvement in Performance Metrics:** Measure the changes in key performance metrics such as page load time, TTFB, FCP, LCP, and CLS. Significant improvements in these metrics indicate successful optimizations.
- **User Satisfaction:** Assess changes in user satisfaction through feedback, surveys, and usability tests. Higher satisfaction levels indicate a positive impact on the user experience.
- **Business Metrics:** Evaluate the impact on business metrics such as conversion rates, bounce rates, and customer retention. Improved web performance should correlate with better business outcomes.

By systematically implementing and evaluating optimization strategies, airline websites can achieve significant improvements in web performance. This not only enhances user experience but also drives business success in the competitive aviation sector. Through continuous monitoring and iteration, airlines can ensure that their websites remain fast, responsive, and secure, meeting the high expectations of modern travelers.

MAJOR FINDINGS

This study on optimizing web performance through front end development strategies for the aviation sector has yielded several significant findings.

Critical Performance Metrics: The research identified key metrics essential for evaluating and improving web performance, including page load time, time to first byte (TTFB), first contentful paint (FCP), largest contentful paint (LCP), cumulative layout shift (CLS), total blocking time (TBT), and time to interactive (TTI). These metrics provide a comprehensive framework for assessing user experience and site efficiency.

Effective Optimization Techniques: Implementing strategies such as minimizing HTTP requests, optimizing images, and utilizing lazy loading significantly enhances web performance. Techniques like minifying and splitting CSS and JavaScript, employing content delivery networks (CDNs), and optimizing the critical rendering path were particularly effective in reducing load times and improving responsiveness.

Caching and CDNs: The use of browser and server-side caching, along with CDNs, proved crucial in reducing latency and ensuring faster content delivery. These strategies are especially beneficial for airline websites that cater to a global audience and experience high traffic volumes.

HTTPS and Security: Ensuring HTTPS implementation not only secures data transmission but also leverages performance benefits like HTTP/2, which enhances overall site speed and user trust.

Continuous Monitoring: Regular performance audits, real-time monitoring, and user feedback collection are vital for maintaining and further improving web performance. Continuous iteration based on these insights helps in adapting to changing user needs and technological advancements.

Overall, the study highlights the importance of a holistic approach to web performance optimization, focusing on both technical enhancements and user experience improvements. By adopting these strategies, airline websites can achieve significant performance gains, leading to better user satisfaction and competitive advantage in the aviation sector.

RECOMMENDATION

Based on the findings, the following recommendations are made for optimizing web performance in the aviation sector:

Prioritize Critical Metrics: Focus on improving key performance metrics such as page load time, TTFB, FCP, LCP, CLS, TBT, and TTI. Regularly monitor these metrics using tools like Google Lighthouse and WebPageTest.

Implement Best Practices: Adopt front end development best practices, including minimizing HTTP requests, optimizing images, leveraging lazy loading, and minifying CSS and JavaScript. Use CDNs to distribute content efficiently and implement robust caching strategies.

Ensure Security: Implement HTTPS to secure data transmission and take advantage of performance benefits associated with HTTP/2. Integrate strong security measures into front end development to protect user data and maintain compliance with industry standards.

Continuous Improvement: Establish a continuous monitoring and improvement cycle. Regularly conduct performance audits, gather user feedback, and iterate on optimization strategies to adapt to evolving user needs and technological advancements.

CONCLUSION

This study underscores the critical role of front end development strategies in optimizing web performance for airline websites. By focusing on essential performance metrics and implementing effective optimization techniques, airlines can significantly enhance user experience, operational efficiency, and competitive advantage. Continuous monitoring and iterative improvements are key to maintaining high performance standards in a dynamic digital landscape. Implementing these recommendations will help airlines provide faster, more responsive, and secure web experiences, ultimately driving customer satisfaction and business success.

REFERENCES

Aviation sector needs a sensible policy, fast. (2014). *Vayu Aerospace and Defence Review,* (6), 4-4, 6.

Bekavac, I., & Pranicevic, D. G. (2015). Web analytics tools and web metrics tools: An overview and comparative analysis. *Croatian Operational Research Review, 6*(2), 373-386. https://doi.org/10.17535/crorr.2015.0029

Chauhan, V., & Manhas, D. (2013). An Investigation of IT Adoption Propensity in Civil Aviation Sector. *International Journal of Hospitality and Tourism Systems, 6*(1), 23-30.

Havranek, M., Hemperek, T., Kishishita, T., Kruger, H., & Wermes, N. (2014). Pixel front-end development in 65 nm CMOS technology. *Journal of Instrumentation, 9*, 7. https://doi.org/10.1088/1748-0221/9/01/C01003

Karabetsos, S., Koulouras, G., Charamis, P., Adamidis, G., Vardiambasis, I. O., & Nassiopoulos, A. (2015). Development of the RF Front-end of a Multi-Channel Microwave Radiometer for Internal Body Temperature Measurements. *Journal of Physics: Conference Series, 637*(1). https://doi.org/10.1088/1742-6596/637/1/012010

Korsakas, E., & Charlamov, J. (2015). Development and Investigation Of An Optical Power Meter Front End. *Mokslas: Lietuvos Ateitis, 7*(3), 330-334. https://doi.org/10.3846/mla.2015.794

Mishra, S., & Liu, X. (2014). Optimizing Concurrency Performance of Complex Services in Mobile Environment. *International Journal of Web Services Research, 11*(1), 94-110. https://doi.org/10.4018/ijwsr.2014010105

Parks, R. F., & Hall, C. A. (2016). Front-End and Back-End Database Design and Development: Scholar's Academy Case Study. *Information Systems Education Journal, 14*(2), 58-63.

Paterno, F., Santoro, C., & Spano, L. D. (2012). The role of HCI models in service front-end development. *Behaviour and Information Technology, 31*(3), 231-244. https://doi.org/10.1080/0144929X.2011.563795

RA?UL PE~ NA-ORTIZ, JOS?E A., G. I. L., SAHUQUILLO, J., & Pont, A. (2015). SURFING THE WEB USING BROWSER INTERFACE FALILITIES: A PERFORMANCE EVALUATION APPROACH. *Journal of Web Engineering, 14*(1-2), 3-021.

Tiwari, R. A., Tuteja, R. R., Pund, M. A., & Dhande, M. R. (2012). Optimizing Performance of Source Code for Real Time System. *International Journal of Advanced Research in Computer Science, 3*(3).

--0--