



# International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





# Design and Implementation of a Restaurant Booking System

**Iroha Chioma Sylvia, Dr. Popoola, Victor Olusegun, Owolola, Lawrence Ayo,**

**Popoola, Obaloluwa Richard, Anoliefo, Chukwuma Josemaria**

Computer Science Department, Espam-Formation University, Campus 2, Porto-Novo, Benin Republic

Computer Science Department, Espam-Formation University, Campus 2, Porto-Novo, Benin Republic

Computer Science Department, Espam-Formation University, Campus 2, Porto-Novo, Benin Republic

Federal University of Agriculture, Alabata Road, Abeokuta, Ogun-State, Nigeria

Computer Science Department, Espam-Formation University, Campus 2, Porto-Novo, Benin Republic

**ABSTRACT:** The digital transformation of the restaurant industry has led to the increasing adoption of automated reservation systems to enhance customer convenience and optimize operational efficiency. This study presents the design and implementation of a restaurant booking system, focusing on streamlining the reservation process, minimizing human intervention, and improving user experience. The system integrates real-time availability tracking, secure payment processing, and automated notifications, ensuring a seamless interaction for both customers and restaurant managers. The research examines key functionalities, including user registration, table selection, booking management, and cancellation policies, while also considering system architecture, database design, and front-end/back-end integration. Additionally, testing methodologies and quality assurance measures are explored to ensure reliability and performance. The findings indicate that an AI-driven, web-based restaurant booking system can significantly enhance customer satisfaction and restaurant efficiency, thereby redefining the dining experience in the modern digital landscape.

**KEYWORDS;** Restaurant booking system; Online reservation management; User experience in dining technology, Database design and system architecture, Real-time availability tracking, Automated booking and notifications

## I. INTRODUCTION

Digital solutions have been increasingly integrated into various sectors in the modern world to simplify tasks and deliver enhanced efficiency. As such, this restaurant booking system is designed and implemented as a digital solution to booking a table at a restaurant. The solution pathway includes an overview of the importance of such a system in the modern dining landscape. It will continue by detailing the problem statement that drives the development of this system. A discussion on the implementation's scope and objectives has been incorporated to outline what the implementation seeks to achieve. There is a specific focus on the secure management of booking transactions, highlighting the integration of a payment processing system. How the use of digital solutions in modern booking processes can improve efficiency and convenience has been discussed. This specific project targets the scope of restaurant booking systems for both restaurant owners and customers. Efforts have been made to speak to each of these objectives across the scope of the system presented here. The outcomes of this implementation are anticipated to drive an increase in customer satisfaction for those dining in restaurants, as well as improvements in the operational efficiency of managing bookings for restaurant owners. The anticipated impacts across both restaurant owners and customers emphasize the need for user-friendly interfaces supporting the core aspects of the system's design. It also highlights the importance of the graphical data representation signature to both the restaurant owners' perspectives on booking trends and the customers' perspectives on booking availability. The restaurant booking system has been designed and implemented intending to achieve these outcomes. Statement restructuring is proposed followed by a decision on whether the implementation effort should be taken in its current domain or in a more general category of projects. Finally, the keywords within the paper driven by the specific scope of system development presented have been outlined. This Go or No-Go decision for the implementation project first required the evaluation of current candidate projects. (Alt, 2021)



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

### II. LITERATURE REVIEW

**Introduction** With the advancement of internet and Web technologies, most of the services are available online. People now prefer to make any bookings online, such as flight, taxi, restaurant, and hotel bookings. The proposed system provides customers with a faster and reliable way of booking a table at their desired restaurant. It eliminates the inconvenience of booking a table over the phone, waiting in a queue, or visiting the restaurant. The online restaurant booking system can be easily accessible by customers over the internet, via laptop, tablet, or smart phones (González-Ramírez et al., 2022).

**Literature Review** The literature review includes a summary of research, systems, or implementations that have been studied and are related to restaurant booking systems. This includes methodology, analysis, and use of technologies in previous studies. The themes of existing research include the security of booking applications, user experience of applications, maps as a media in booking a table, and the challenges of integration with restaurant systems. Insufficient technology and on-site promotion is a weakness that can be exploited by competitors. A theoretical foundation is established that places this technological design in the right historical, competitive, and external context. In addition, it also includes models describing how variables are related to each other. Four hypotheses were developed in response to the primary and secondary objectives. (Puri et al., 2023)(Acheampong, 2021)

Information and communication technologies (ICTs) are having a revolutionary impact on firms, workforce, and society, transforming old processes and creating new alternatives. Moreover, customers are also becoming more skillful in the use of ICTs. The goal is to provide a Web based cutting-edge bibliographic database of ICT adoption in tourism in order to improve different researchers' tasks, to avoid duplications, to shed on potential gaps on the previous studies and to offer new ideas for other topics. The main topics and methodologies development are summarized. (Wendt et al., 2022)

### III. SYSTEM REQUIREMENTS

**Functional Requirements** The success of developing a Restaurant Booking System (RBS) depends on a clear and concise understanding of the requirements that will satisfy the needs of potential users. Aiming to fully understand these requirements, it is necessary to perform requirement elicitation. In light of this, all the elicited requirements were carefully analyzed, refined, and restructured for a better understanding. A comprehensive delineation of the specific requirements needed for the system to be effectively designed and implemented is provided. To break down the complexity of the requirements, they are classified as functional requirements and non-functional requirements. Functional requirements describe the features and functionality to be provided by the software system. While having an understanding of these requirements, it is recommended to consider all of the sub-sections: 1. User Registration: In order to use the system the user shall be obliged to register providing their information and email addresses. Registered users shall have the capability to be informed about the recent booking operations. It is compulsory to make a registration and obtain a password. 2. Table Reservation: Registered users shall reserve a table in a particular restaurant in a particular time and day. The user shall choose the number of persons coming for the meal and prefer to select the desired restaurant. The restaurant becomes an option after the date and time selected. It allows the users to know about the remaining tables and to book a table of their wish. 3. Table Capacity and Time Constraint: When the user has chosen the date, time, and restaurant, the system displays the available tables according to the reviewing of that time. Furthermore, the system also shows whether customers can select a new time in case tables are remaining in other times of the same day. 4. Table Reservation Success: Users shall see booking confirmation upon completion. The particular users shall receive SMS or e-mail confirmation about the meal. Booked tables can be canceled. Reversed time cancellation is not permissible. And the system shows an error message if the time for booking cancelation is reversed. (Wiilams & Ajinaja, 2019)

#### 3.1. Functional Requirements

The expected operational functionalities required for the application can be defined by identifying a set of key user interactions e.g. functionality necessary to facilitate a user successfully booking a table in a restaurant. The expected process the user will follow each interaction should then be articulated. This will provide a full outline of the expected inputs, processes, and outputs involved in each user task, assisting with thinking through the tasks in greater detail.



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A) Browsing - A list of restaurants is presented to the user within an appropriate radius - The current time is automatically input when the user visits the page to simplify browsing - Only the soonest 5 available times are displayed following user selection

B) Timeslot information - The list of available times and the selected restaurant are sent to the application - The total number of tables of each party size and the number of tables which are booked for each time are returned

C) Booking a Table - The restaurant, desired time, number of diners, and preferred table are input - The party size should be deduced from the selected table - If no table is input, the user is given one at random (preferably in the non-reserved section) - A booking is confirmed if the party size is within the range accepted by the table

D) Reservations The user has access to their reservations after logging in. They can cancel any reservation, which immediately becomes available. Feedback can be provided on too tight or too loose a time between booking and arrival, or on the availability of bookings and arriving times. The user also has the ability to see booking guidelines at any time.

E) Signing Up (Guests Only) - A user provides an email, name, and password. They can enter further details to complete their profile - They are then automatically logged in

F) Signing In - A guest user can login using only their email address - A registered user can either use their email and password, or use their Facebook login - The password can be reset using a reset email mechanism to be received by their email

Make use of any external systems that can be integrated with to improve functionality.

### 3.2. Non-functional Requirements

The Softbody simulation system allows users to animate and render 3D models under the physical laws of elasticity. The Softbody simulation focuses on rendering effects applied to closed Solid Blood objects. In further sections, the System Features to be designed are recovered from the previously collected scenarios. The scenarios are organized in subsections relating to the same feature or a set of closely related features. In each subsection, the scenarios are re-listed, and the individual requirements of each scenario are itemized. Focus is now shifted to define the non-functional aspects underpinning the Service, as extracted before. It is broadly described how the Service should perform in multiple dimensions including the desired system response time, reliability, and scalability under different loads. Also detailed are how users are expected to interact with the Service in terms of usability. Finally, various external attributes, such as how to ensure the system is secure, compliant with regulations, maintainable, and easily adaptable are presented. (Sefati et al.2022)(Jafari et al., 2023)

## IV. SYSTEM DESIGN

### 4.1 System Work Flow

Whenever customer makes a reservation from website, these items must be selected: (1) Number of Customers and Information (2) Time for Reservation (3) Phone Number of Customers. If customer comes to the restaurant physically, the Items in the physical restaurant are selected. At this time, if the customer wants to modify, more data is added. If it is added, the physical restaurant is selected first. This is because the physical restaurant is selected as the first and then the Items are adjusted. The actual order of how data is added is First Item, Item Information, Customer Preferences. Items is added in as many data as there are. Item Information is entered as much as Items have been selected, and Customer Preferences is added to adjust customer's preference. Each is modified or added by selecting a button. (Tseng et al.2021) It illustrated the interaction and communication between each component of the system. "PC" at the bottom represents the server provided by the hosting company. HTML, PHP, and SQL files required for creating and interacting with webpages are hosted and files created in the programming process must be hosted here. It is written PHP scripts for interaction between the website and the SQL database. All information inputted from the website is stored in the database through these scripts and delivers the requested information to the website through the same scripting after querying the database. Hence, the PHP script acts as a connection device for websites and databases. It is created required PHP pages necessary for the functioning of the website. These PHP pages interact with the SQL file. SQL is used to create and interact with the database on the server. mappedBy.name refers to which table in the database the information is being inputted to or outputted from. After interacting with the database, the resulting information is answered in a JSON format. To check this interaction on the local computer, software is used. (Powers and Powers2022)



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

### 4.1. Architecture

Modern software systems are built utilizing various architectural approaches, patterns and techniques. The architectural framework reflects the technological background of the system and significantly influences the system's performance, capabilities and maintainability. It is essential to select a suitable architectural pattern in the early stages of system development, to ensure the potential of the system is maximized and proper resources are allocated. There is a broad selection of basic architectural designs, comprising both monolithic and distributed architectures, which can then be expanded using various design patterns and technologies. These patterns are merged to define the most fitting architectural structure corresponding to the needs of this software solution. The selected pattern suits system requirements while providing satisfactory scalability options and streamlined future maintenance. The software architecture distinguishes between the user interface, the application logic, and the data storage of the application. Each architectural layer is discussed separately to determine which architecture is most fitting for the restaurant booking system, relevant design decisions and justifications are given. The architecture design aims at modular and separable concerns to augment the reusability and maintainability of the applications. Prototypical diagrams are used to illustrate a prevalent type of architecture for the client side application and another one for the server side logic and data storage. (Dogani et al., 2023) The client communicates with the application logic through RESTful APIs. RESTful JSON APIs can be built with Laravel using several techniques. One simple technique is to use the Laravel controller response functions `Response::json()` to just pass an array of data to the Ajax request in JSON format. Another method is to use the alternative `FormRequest` class for validation and error response in a RESTful controller (Subro Banerjee, 2004). A third technique is to use Validator facade of Laravel to validate the HTTP request and use it to generate a proper response. The communication between the front end and back end applications always happens asynchronously. This offers the front end additional time to concentrate on the user interface, and creates a seamless experience for the users even while the requests are processing in the background of the application logic. Several methods can be used to allow the cloud server to easily scale these requests, such as building a cloud server independent of the file system, maintaining the data using a cloud based service, or applying asynchronous processing of background tasks using a queue servers. If the request limits of these asynchronous routes are reached, a solution could be to offload the communication transport between the client application to the cloud based service. (Gómez et al.2022)(Bui & Bui, 2023)

### 4.2. Database Design

This sub-section discusses database design for the system. An appropriate database management system is chosen. The database design itself is explored using schema design, data normalization, and entity-relationship diagrams. To ensure a baserate of understanding of the context in which this database will be used, the data requested by the system is detailed, and authoritative relationships between database tables are described. Following this basic theoretical foundation, a practical exploration of database operations is conducted. The key implementation is detailed, including how to create or modify a table, insert data into a table, query a table for data, join two or more tables, update or delete a data record, and create a new table using data from at least one existing table. This implementation is presented in the form of concrete examples. Consistent with best practice when designing a database system, queries against the database are not embedded within the client applications' code. Conversely, all such queries are directed towards the database management system, thereby having the added benefits of easier maintenance, better security, and increased reliability. (Gadde2022)

The proposed system can only be as effective as the data it administers. Given the functioning of the system, such data must relate to: the restaurants themselves, bookings made at each restaurant, and users who make bookings. In turn, each of the camps of data requires the recording of associated auxiliary data as a means of ensuring database integrity. For instance, anonymous users should not be permitted to leave restaurant reviews; an inherent relationship between the Users table and the Reviews table is ascertained. All this data is now detailed. (Pang et al.2021)

## V. IMPLEMENTATION

The objective of the implementation section is to describe, in great detail, the actual execution of the system development based on the design principles defined in the previous sections. This section details how the design principles translate into pragmatic endeavors. The implementation section is divided into several subsections in order to provide a comprehensive elucidation of the technical aspects of the system development. The first subsection will describe the setup environment needed to develop the system, both hardware and software. The environment is then operationalized to be a front-end development and a back-end development, as explained in the subsequent sections. The development



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

process of the front end and back end will be explained. Finally, the integrated result, challenges found during the development phase, the solution for those challenges, and version control as a way to manage the changes of the project are also elaborated. (Ma, 2022)

Software implementation is mainly concerning how the design is implemented towards the development of running stream. It explains the detailed specification about how certain design or algorithms can run properly. Implementation is the phase where the project is separated by each subsystem to specify specific detail which will make it possible to realize the whole project. After needing to specify and gather all requirements and conducting a design phase, a development is the embodiment of the system that has been established towards a running and working system. It is desired that the restaurant booking system has been reachable as a running system according to objectives stated in Chapter 1 enter from customer ID and click button 'Change User Preferences'. Then they will select their preferences which are choice between Breakfast Type, Lunch Type, Dinner Type, Drink Type, and Activity type. Finally, enter in new amount for that preference and then clicking button 'Change User Preferences', customer's preferences will be updated. (Ardiansyah et al.2021)

### 5.1. Frontend Development

This sub-section focuses on the design and development of the frontend of a restaurant booking system (Lin, 2006). As the face of the platform, the frontend is the primary point of interaction for users, with a responsive and eye-catching user interface representing the key goal. Various front-end technologies and frameworks can be used in the ongoing development process to build an elegant and highly functional user interface (UI). User experience (UX) design principles should guide the design of the front-end, aiming to make it intuitive to navigate and inclusive to all. Emphasized the importance of features like responsive design, critical for ensuring that the UI adjusts smoothly for various devices and screen sizes, allowing users to access the platform and engage with it comfortably. The platform's color schemes are what users will notice and will subconsciously guide their feelings or thoughts. Additionally, typography and the correct use of typefaces can play a significant role in enhancing user experience. In general, the UI needs to be engaging and consistent, incorporating visually pleasing elements to promote a good first impression from users. A series of user testing sessions should also be planned in advance. They will act as an important element in gathering feedback on the use of the platform, identifying existing issues, and subsequently iterating on design improvements. Performance optimization is also important. The optimizations outlined later in this sub-section target fast loading times for the frontend, ensuring that the platform is accessible as soon as possible, thus enhancing user satisfaction. Ultimately, the UI should aim for seamless functionality and an engagement with the backend aspects, enhancing the user experience by providing the latter in an appealing, easy-to-use manner. (Unger & Chandler, 2023)

### 5.2. Backend Development

The core part of the system on which the restaurant booking system is going to run is the server side. Essentially, it is the back end side of the system which includes all data processing logic, database queries, and server side functions that power the website. The user facing reaction is the front end and the components that users interact with. In the back end, it talks about the technologies and programming languages used to write server side code. (Nebaba et al., 2024)

The system needs to facilitate the storage, manipulation, and retrieval of large amounts of data. This is done by database systems that specialize in those tasks. Typically, this is done with relational databases, but it might also involve querying data structured with various NoSQL databases. All of the server side functions are an API. This is a protocol adopted by the back end that front end can use to communicate with it. The back end receives a request, processes it, and sends some sort of response. (Doricchi et al.2022)

This is where the command line options can come into play. It should write robust server logic that satisfies the previously specified functional and non-functional requirements. This includes creating secure functions to guard against malicious database manipulation, as well as creating functions that validate incoming data and prevent malformed data from entering the database system. The secure API is constructed. This will consist of methods written in the runtime environment that interface with the database. These methods will be mounted onto a platform to allow for documentation generation and testing of API calls. Additionally, the back end code itself will be written using a framework with the runtime environment. Moreover, the server will be optimized for performance through load balancing and caching, as well as through the use of a reverse proxy. (Ibrahim et al.2021)



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

### VI. TESTING AND QUALITY ASSURANCE

6.1 Overview This section describes the methodology which is adopted while developing the Restaurant booking System. The following types of testing are done in the Restaurant Booking System: unit testing, then integration testing and then system testing. After testing of code is done, quality control processes are used in form of achieving code optimization and code review. All this testing and quality methods are discussed in detail for each module of the project. (Pramudito et al.2024)

6.2 Unit Testing This type of testing is performed to independently test each of the component or module or each unit of the architecture. After a module is developed, test classes are created to test that module. These test classes help a developer understand the functioning of module and also to debug that module. Test classes are used not only to catch the bugs during development but also to keep check on the quality of code. (Langr, 2024)

6.3 Integration Testing This type of testing focuses on checking the interface between modules. After each of the modules passes the unit testing, then interface between the modules are tested to check the information flow between the modules. Each of the modules is combined together (say two at a time) and these collective modules are tested for any errors. After checking the interface of all the modules, the architecture as a whole is tested and application is tested for different scenarios (M. O'Connor, 1990). This type of testing is done to ensure that all the modules work perfectly when combined to function to create a sub set of the system.

6.4 System Testing This type of testing focuses on checking the architecture for a system as a whole. Application is tested with different types of inputs for different scenarios and is thoroughly checked for the bugs.

6.5 Quality Assurance Apart from testing, to ensure better quality of the code, certain processes are followed like: code review, code optimization, well definition of coding standards and constant checks to ensure that the code adheres to the coding standards. Code reviews are done to check the logic of the code, to check if the code the optimal or not, checks are done to ensure that the each of the function follows the correct class (Sharma, 2017). Code review is also done to check if there is any unnecessary code or any errors in the code. There are tools available to automatically check the code. There are various tools available which helps in maintaining the quality of code. Code optimization is done to ensure the maximum that is to make the code more easy and efficient.

### REFERENCES

1. Alt, R. (2021). Digital transformation in the restaurant industry: Current developments and implications. *Journal of smart tourism*. sagepub.com
2. González-Ramírez, R., L. Gascó, J., & Llopis, J. (2022). Information and communication technologies in food services and restaurants: a systematic review. [PDF]
3. Puri, V., Mondal, S., Das, S., & Vrana, V. G. (2023). Blockchain propels tourism industry—an attempt to explore topics and information in smart tourism management through text mining and machine learning. *Informatics*. mdpi.com
4. Acheampong, R. A. (2021). Societal impacts of smart, digital platform mobility services—an empirical study and policy implications of passenger safety and security in ride-hailing. *Case Studies on Transport Policy*. manchester.ac.uk
5. Wendt, C., Adam, M., Benlian, A., & Kraus, S. (2022). Let's connect to keep the distance: How SMEs leverage information and communication technologies to address the COVID-19 crisis. *Information Systems Frontiers*. springer.com
6. Wiilams, K. & Ajinaja, M. (2019). Design and Implementation of Reservation Management System - Case Study: Grand Ville Hotels. [PDF]
7. Sefati, S., Mousavinasab, M., & Zareh Farkhady, R. (2022). Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation. *The Journal of Supercomputing*, 78(1), 18-42. researchgate.net
8. Jafari, M., Kavousi-Fard, A., Chen, T., & Karimi, M. (2023). A review on digital twin technology in smart grid, transportation system and smart city: Challenges and future. *IEEE Access*. ieee.org



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

9. Tseng, T. H., Wang, Y. S., & Tsai, Y. C. (2021). Applying an AHP technique for developing a website model of third-party booking system. *Journal of Hospitality & Tourism Research*, 45(8), 1440-1463. [HTML]
10. Powers, D., & Powers, D. (2022). Connecting to a Database with PHP and SQL. *PHP 8 Solutions: Dynamic Web Design and Development Made Easy*, 357-394. [HTML]
11. Dogani, J., Namvar, R., & Khunjush, F. (2023). Auto-scaling techniques in container-based cloud and edge/fog computing: Taxonomy and survey. *Computer Communications*. [HTML]
12. Subro Banerjee, S. (2004). Software architecture: styles and representational schemes. [PDF]
13. Gómez, A., Iglesias-Urkia, M., Belategi, L., Mendiola, X., & Cabot, J. (2022). Model-driven development of asynchronous message-driven architectures with AsyncAPI. *Software and Systems Modeling*, 1-29. [springer.com](https://www.springer.com)
14. Bui, D. D. & Bui, D. (2023). Next.js for front-end and compatible backend solutions. [theseus.fi](https://theseus.fi)
15. Gadde, H. (2022). Integrating AI into SQL Query Processing: Challenges and Opportunities. *International Journal of Advanced Engineering Technologies and Innovations*, 1(3), 194-219. [academia.edu](https://www.academia.edu)
16. Pang, J., Huang, Y., Xie, Z., Li, J., & Cai, Z. (2021). Collaborative city digital twin for the COVID-19 pandemic: A federated learning solution. *Tsinghua science and technology*, 26(5), 759-771. [ieee.org](https://www.ieee.org)
17. Ma, X. (2022). Recent advances in mass spectrometry-based structural elucidation techniques. *Molecules*. [mdpi.com](https://www.mdpi.com)
18. Ardiansyah, M. R., Kusuma, J. F., Gumelar, I. S., & Mentari, M. (2021, September). The Website-Based Food Booking System Equipped with Real-Time Booking Status to Address Queuing Issues at Restaurants. In 2021 International Conference on Electrical and Information Technology (IEIT) (pp. 133-138). IEEE. [HTML]
19. Lin, T. (2006). Restaurant recommendation system (RRS). [PDF]
20. Unger, R. & Chandler, C. (2023). A Project Guide to UX Design: For user experience designers in the field or in the making. [HTML]
21. Nebaba, N. O., Kucher, M. M., Yazina, V. A., Saihak, Y. L., & Golovko, N. O. (2024). Theoretical aspects of information provision of restaurant enterprise management processes. [umsf.dp.ua](https://www.umsf.dp.ua)
22. Doricchi, A., Platnich, C. M., Gimpel, A., Horn, F., Earle, M., Lanzavecchia, G., ... & Garoli, D. (2022). Emerging approaches to DNA data storage: challenges and prospects. *ACS nano*, 16(11), 17552-17571. [acs.org](https://www.acs.org)
23. Ibrahim, I. M., Ameen, S. Y., Yasin, H. M., Omar, N., Kak, S. F., Rashid, Z. N., ... & Ahmed, D. M. (2021). Web server performance improvement using dynamic load balancing techniques: A review. *system*, 19, 21. [academia.edu](https://www.academia.edu)
24. Pramudito, D. K., Pettalongi, S. S., Tawil, M. R., Hermila, A., & Zein, A. (2024). Application of Rapid Application Development Method to Design E-Commerce Systems in National Expedition Company to Increase Marketing Effectiveness. *Jurnal Informasi dan Teknologi*, 144-149. [jidt.org](https://www.jidt.org)
25. Langr, J. (2024). Pragmatic unit testing in Java with JUnit. [HTML]
26. M. O'Connor, N. (1990). An object-oriented testing approach based on a rigorous model of claimed functionality. [PDF]
27. Sharma, A. (2017). Optimizing Software Quality through Automation Testing. [PDF]





INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  [ijircce@gmail.com](mailto:ijircce@gmail.com)



[www.ijircce.com](http://www.ijircce.com)

Scan to save the contact details