

MEDALLION THEATRE: TICKET RESERVATION SYSTEM

An Analysis

Presented to

Mrs. Villafuerte, Minabelle

In Partial Fulfillment

of the Requirement for

Software Design Lecture (CPET8)

BY:

Arenas, Joseph C.

Estrada, Adrienne Cyruz

Gutierrez, Geo Kenzter

Pacis, Lian Gil

Recaña, Jordan

BET-CPET-2A

MARCH 2025

TABLE OF CONTENTS

I.	INTRODUCTION AND BACKGROUND.....	1
II.	FLOW OF THE SYSTEM.....	2
III.	ISSUES AND CHALLENGES.....	4
IV.	REFERENCES.....	5
V.	APPENDIX	
	APPENDIX “A”.....	5

I. INTRODUCTION AND BACKGROUND

In today's world, the use of technologies and smart devices is common. Almost everything is done virtually, such as us attending virtual meetings or booking a theatre ticket. In the case study of Fox, T. (2021) entitled, "Case Study : Systems Analysis, Design, and Development Case Study: Medallion Theater - Ticket Sales System" the character Dr. Thomas Waggoner encountered an issue at Medallion Theatre in which his reserved ticket was mistakenly sold to another patron due to manual tracking process. Due to this incident, Dr. Waggoner proposed to his students to develop a computerized system to manage tickets sales efficiently.

Building on this, the researchers conducted an analysis to develop a Ticket Reservation System. The need for this kind of system was not only limited to Medallion Theatre. Nwakamma, Etus, Ajere, and Agomuo (2015) developed OBTRS which stands for Online Bus Ticket Reservation System. It is a web-based application designed to replace the traditional manual booking with a computerized approach. In this application, the user is able to book a seat reservation, cancellation and if the user has route inquiries. The system was developed using Structured Systems Analysis and Design Methodology (SSADM), PHP for the front end and MySQL for its back end.

Nwakamma, Etus, Ajere, and Agomuo (2015) concluded that having a computer application would be vital and significant in every field of human endeavor. The use of an online ticket reservation would help the people and increase efficiency. Given its positive approach, OBTRS will serve as a guide for the researchers to further develop the Ticket Reservation System for the Medallion Theatre.

II. FLOW OF THE SYSTEM

The researchers Agomuo et al. of the feasibility study entitled "Online Bus Ticket Reservation System" utilized a software process model known as the Waterfall Model. The researchers in particular adopted a variant of the aforementioned model known as the "System Structured Analysis and Design Methodology (SSADM)." The SSADM is a waterfall model variant mandated by the United Kingdom British Standard BS7738 in government-related application software development projects. It adopts a stage-by-stage software development scheme for software engineers to utilize as a reference for a systemized and well-documented program (Lutkevich, 2022).

A Data Flow Diagram (DFD) was provided in the feasibility study to present the operational flow of data from end-user to application software, admin-user to application software, and application software to end-user and admin-user. The study also provides a "Case-Use" diagram that exhibits the potential situations of when the end-user and admin-user interact with the system and if those "cases" are accessible to them. Referencing the DFD at level 0, the general point of view of the system is presented. Each user's inquiry, booking, and payment are then submitted within the reservation system, ready to be processed by the admin. The processed inquiry of the users will then be sent back to the users containing service details. At Level 1, the presentation of the admin's view of the system is shown. It begins by logging into the system, then an admin can do maintenance for the master tables (e.g., the Bus Details Table, Route Table, Schedule Table, Pricing Table, and Customer Details Table), then update details within the database when needed. The database will then generate reports regarding the system to notify an admin of problems to troubleshoot. Following the level 2 end-user DFD, the flow of data is as follows.

As stated, the front-end development of the Online Bus Ticket Reservation System was formulated through Hypertext Processor (PHP). As such, the graphical interface in which the end-user and admin-user enable program interaction is rooted in this programming language. The end user is first prompted with a home interface that shows the admin login prompt as well as the ticket reservation prompt. The interface also presents the end-user with other graphical elements such as pictures, font colors, and font styles, which were all generated using Adobe Dreamweaver CS6 and Adobe Photoshop CS6. Prompts such as the history of the company, a contact page, routes interface, and terms and conditions are also presented to the end-user upon program execution.

Upon the end-user interacting with the "Reservation Menu," the various routes and schedules of transportation utility vehicles (TUVs) are presented for the user to make a bus seat reservation. The real-time, updating schedule list is stored in the MySQL v.5 database of the application software. According to the DFD, once the end-user has prompted their reservation and invoice details, they are asked by the program first to verify the information they have placed. Once the end-user verifies their data, the database is updated, prompting the admin-user and other end-user that a certain individual has paid for a reservation on their respective bus seat number. In the reservation menu (after payment), the updated database prompts the end-user that their payment is successful.

Using PHP, the program presents a printable ticket invoice receipt, which then asks the end-user to print for documentation purposes. The end-user also has the option to cancel their reservation, which is then updated through the database of the system once confirmed. In light of the level 2 end-user DFD, the use case scenarios of the end-user are limited to bus routes and schedule inquiries, reservation options, and ticket printing (Agomuo et al., 2015). As for the admin ser, the use case scenarios are managing the master table wherein they can make a reservation,

cancel that reservation, print a ticket, log in for admin privileges, update the route of TUVs, update the payment, and update the bus details (Agomuo et al., 2015). The system's performance, security, flexibility, and accessibility requirements are yet to be investigated since the researchers did not provide a simulation to demonstrate how the system would handle multiple reservations at the same time, how passwords or payments can be made secure, how the system will adapt, and how it can be used for both desktop and mobile devices.

III. ISSUES AND CHALLENGES

Although the online bus ticket reservation system is a great tool for automating ticket purchases, it may run into a number of issues that affect its overall functionality and time efficiency. Both limitations and bad system design may be the cause of these issues, which could result in dissatisfied customers and ineffective operations. Slow response time is a significant problem, which can cause delays for users when they are looking for available seats, processing reservations, or paying. Additionally, if scalability is not taken into account during development, a large number of simultaneous users may cause the system to crash or experience performance lags due to inefficient load handling.

Database latency is another possible issue that may impact the system's effectiveness when older versions of MySQL are used. It may take longer to retrieve and update information on seat availability and booking status due to its outdated database system. When several people are using the system at once during peak hours, this problem is more visible. The lack of real-time seat availability updates is another issue. If the system does not update seat availability immediately, users may choose seats that have already been reserved by someone else.

At last, the reservation process is made more difficult and time-consuming by the system's poor user interface and user experience (UI/UX) design. The overall responsiveness and security of the system may be affected by the use of outdated technologies, such as older versions of Adobe Dreamweaver, MySQL, and other development tools, which may not be compatible with latest web browsers and mobile devices. To improve user interactions, the system should integrate additional features like online payment and email for ticketing passengers.

IV. REFERENCES

- Agomuo, U., Ajere, I., Etus, C., and Nwakanma, C. I. (2015). Online Bus Ticket Reservation System. Statistics and Computing Vol. 1(2).
https://www.researchgate.net/publication/326468848_Online_Bus_Ticket_Reservation_System
- Lutkevich, B. (2022, June 17). What is SSADM (Structured Systems Analysis and Design Method)? Software Quality.
<https://www.techtarget.com/searchsoftwarequality/definition/SSADM>

V. APPENDIX A

We, the researchers for the Ticket Reservation Management System for the Medallion Theatre, hereby declare that this research work is our original creation and has been completed in accordance with academic integrity standards. Any references, sources, or materials borrowed from other authors, whether directly quoted or paraphrased, have been properly cited and acknowledged.

We affirm that this document is free from plagiarism and has not been copied, wholly or in part, from any other research, publication, or online sources without proper citation. The researchers understand that any form of academic dishonesty, including plagiarism, constitutes a serious offense and may result in disqualification or academic penalties.

We certify that all data, findings, and analyses presented in this study are genuine and have been obtained through ethical research practices. Furthermore, this document has undergone plagiarism detection and review to ensure compliance with ethical standards in research writing.

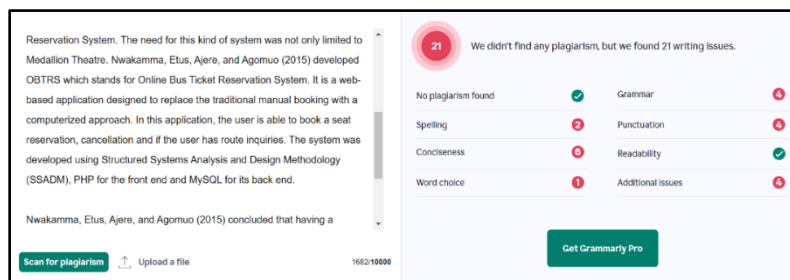


Figure 1: Introduction Plagiarism Checker

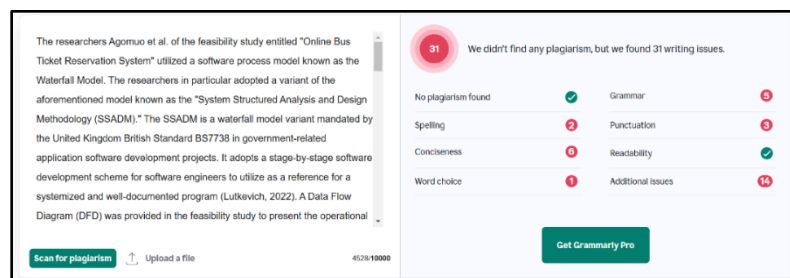


Figure 2: Flow of the System Plagiarism Checker

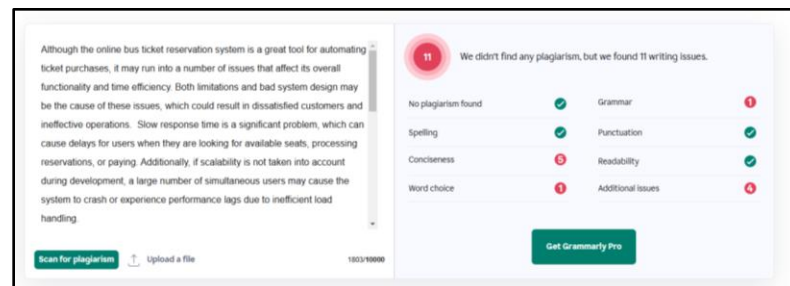


Figure 3: Issues and Challenges Plagiarism Checker