

Mini Project Report

on

Virtual ticketing system

Project title: Virtual
Ticketing System.

Student Reg. No & Name: K .NIVEDINI

KUB23CSE055

SAI DEEPTHI .T

KUB23CSE122

ANJALI

KUB23CSE009

K.V. SAMANA

KUB23CSE060

K.S. DEVIKA

KUB23CSE058

Department Name: Department of Computer Science and
engineering

Year of Study: 2024-27

Submission Date: 28-09-2024

Table of Contents:

1. Introduction:

Virtual Methods and Virtual Functions:

In object-oriented programming (OOP), a *virtual method* is a method that can be overridden by derived classes. While Python does not have the "virtual" keyword like some languages (e.g., C++), methods in Python classes are virtual by default, meaning that they can be overridden in child classes. This is achieved through inheritance and polymorphism.

- Virtual ticketing system is a software application that helps business. To manage and track customer requests and support inquiries.
- Virtual ticketing system is designed to streamline the process of managing and issuing tickets for various events, transportation services, or customer services
- It includes: Virtual tickets, digital tickets, virtual help desk

In OOP, Python methods are virtual by default and can be overridden

Virtual environments provide isolated workspaces for project dependencies.

Virtual properties or attributes are computed dynamically.

Scope of Project:

1. Convenience: Fans can purchase tickets online anytime, eliminating the need to visit physical outlets.
2. Contactless Entry: Mobile or digital tickets reduce physical contact, enhancing safety and efficiency during entry.
3. Real-Time Updates: Event organizers can easily communicate changes, updates, or important information to attendees.
4. Data Insights: Collects valuable data on ticket sales, attendee demographics, and preferences, aiding in marketing and planning.
5. Scalability: Easily manage ticket sales for events of varying sizes without the need for extensive physical infrastructure.
6. Reduced Costs: Lowers costs associated with printing and distributing physical tickets.
7. Ticket Creation and Management: Using Python's object-oriented programming (OOP) capabilities, different ticket classes can be created for event ticketing and support tickets. This includes creating new tickets, assigning unique IDs, tracking status, and categorizing them by priority or type.

8. Dynamic Routing and Assignment: Implement rules to automate ticket routing based on specific criteria using Python-based logic or machine learning models for ticket prioritization.

9. Event Ticket Sales and Purchases: Implement interactive forms and shopping cart mechanisms.

Limitation of Virtual Ticketing Systems for Sports Events

1. Technical Issues: System outages or glitches can disrupt ticket sales and entry processes, frustrating fans.

2. Accessibility: Not all fans may have reliable internet access or be comfortable using technology, potentially excluding some demographics.

3. Fraud Risk: Digital tickets can be subject to counterfeiting or scalping if not adequately secured.

4. Refund and Cancellation Challenges: Handling refunds for virtual tickets can be more complex than traditional methods, leading to customer dissatisfaction.

5. Impersonal Experience: Lacks the personal touch of in-person ticket sales, which some fans may prefer.

6. Learning Curve: Fans unfamiliar with digital ticketing may struggle with the process, leading to confusion and delays at entry points.

7. Access Problems: Users in areas with poor or intermittent internet access may face difficulties purchasing tickets, submitting support requests, or checking ticket status.

8. Impact: Users in regions with low bandwidth or unreliable internet may experience difficulties in accessing the system,

leading to frustration during ticket purchase or support interactions.

Purpose of the Project:

A virtual ticketing system for sports events envisions a seamless, engaging, and secure experience for fans, organizers, and venues. Key components include:

1. User-Friendly Platform: An intuitive interface for easy navigation, ticket selection, and purchase
2. Mobile Access: A dedicated app for ticket purchases, event details, and real-time updates, allowing fans to access their tickets anywhere.
3. Dynamic Pricing: Adjustable pricing models that respond to demand, maximizing revenue while providing fans with competitive prices.
4. Secure Transactions: Advanced encryption and payment security to ensure safe transactions.
5. Purpose of a virtual ticketing system is to streamline the management of tasks, issues, requests, or events by creating, tracking, and resolving tickets (or cases) in a digital format. These systems are widely used across various industries for a variety of reasons, including:
 6. Organized Task Management:
Centralized Tracking: Every task, issue, or request is logged as a "ticket," which can be tracked and managed efficiently.

Prioritization: Tickets can be categorized and prioritized based on urgency or importance, helping teams focus on what matters most.

7. Improved Communication:

Collaboration: Virtual ticketing systems often allow multiple people or teams to work on resolving the same issue while maintaining clear communication.

8. Transparency: All stakeholders (employees, customers, etc.) can see the status and updates on the ticket, improving visibility and accountability.

9. Automated Workflow

Automation: Tickets can trigger workflows or notifications automatically, reducing manual intervention and improving response times.

10. Assignment Rules

Systems can automatically assign tickets to the right team or individual based on predefined criteria.

11. Customer Support and Service

Efficient Resolution: In customer support, virtual ticketing allows customer queries, complaints, or requests to be logged, tracked, and addressed systematically.

12. Self-Service Options: Customers can submit their own tickets or track their resolution status, enhancing their experience.

13. Metrics and Reporting

Analytics: Ticketing systems often provide reports and analytics on the number of tickets, response times, resolution rates, and customer satisfaction.

14. Continuous Improvement

This data helps organizations identify areas for improvement, trends, and recurring issues.

15. Scalability and Flexibility

Handling Volume: As businesses grow, a virtual ticketing system can handle larger volumes of requests or issues without overwhelming teams.

16. Customization

These systems can often be customized to suit the specific needs of different industries, such as IT, HR, event management, or customer service.

17. Cost-Effective and Time-Saving

Resource Optimization: By automating and streamlining the ticketing process, organizations can save both time and resources.

18. Reduction of Errors:

Virtual ticketing minimizes the risk of lost or forgotten issues, ensuring that everything is tracked and managed properly.

2. Objective:

Purpose of a ticketing system is to process, prioritize, and manage customer issues from end to end.

Centralized Repository: Collects and stores all service requests in single location

Automated Processes: Streamlines the creation and management of tickets, including automated responses and routing

Visibility and Transparency: Allows both customers and internal teams to track the status of requests, enhancing accountability and transparency

Data Reporting and Analytics : Offers insights into performance metrics and helps in identifying areas for improvement.

Automate Task Assignment: Ensure that tasks or issues are

automatically assigned to the appropriate individuals or teams, reducing manual intervention and speeding up processes.

Prioritize and Organize Tasks: Categorize and prioritize tasks to ensure that critical issues are addressed promptly and systematically.

Improved Communication and Collaboration

Centralize Information: Provide a single platform where all relevant stakeholders can access information and updates regarding a specific ticket, eliminating communication silos.

Foster Collaboration: Facilitate collaboration among different teams or departments to resolve complex issues more efficiently.

Enhanced Customer and User Experience

Quick and Transparent Issue Resolution: Improve response and resolution times by tracking customer queries or support requests and keeping users informed of progress.

3. Methodology:

Hosting Technology : opt for cloud-based systems for flexibility and reduced maintenance costs.

.Usability: ensure the system is user-friendly for both staff and customers.

.Customization and Integrations: choose a system that offers customization options and integrates well with other software tools

Alignment with ITIL: ensure the system aligns with ITIL best practices for efficient IT best practices for efficient IT service management.

Software used:

1. Web Application framework: VS (visual studio) Code, Python 3.12(64-bits) , Microsoft Powerpoint.

2. Payment Processing Payment Gateway Integration:
Support for services like Pay Pal, or Square to handle secure transactions.

Identity Management: Implement OAuth or JWT for secure user login and registration.

3. Role-Based Access Control: Manage permissions for users, admins, and organizers.

4. Customer Support Tools

Chatbot/Live Chat Software: Integration with services like Intercom or Zendesk for customer inquiries.

Email Marketing Software: Tools like Mailchimp for communication and promotions.

5. Backup and Recovery Software

Data Backup Solutions: Automated backup software for regular data protection.

Scalability: Ensure the software can handle increased traffic during peak sales periods.

Storage: SSD storage for faster data access.

RAID Configuration: Use RAID-10 or RAID-5 for data redundancy and fault tolerance.

Web/Application Servers : CPU: Intel Xeon or AMD EPYC, 16-32 cores.

RAM: 32 GB – 64 GB OR 128 GB or Higher

Storage: 1 TB SSD (RAID-10 for redundancy).

Network: Dual 10 Gbps Ethernet.

Database Servers : CPU: Intel Xeon or AMD EPYC, 32+ cores.

Description of the project:

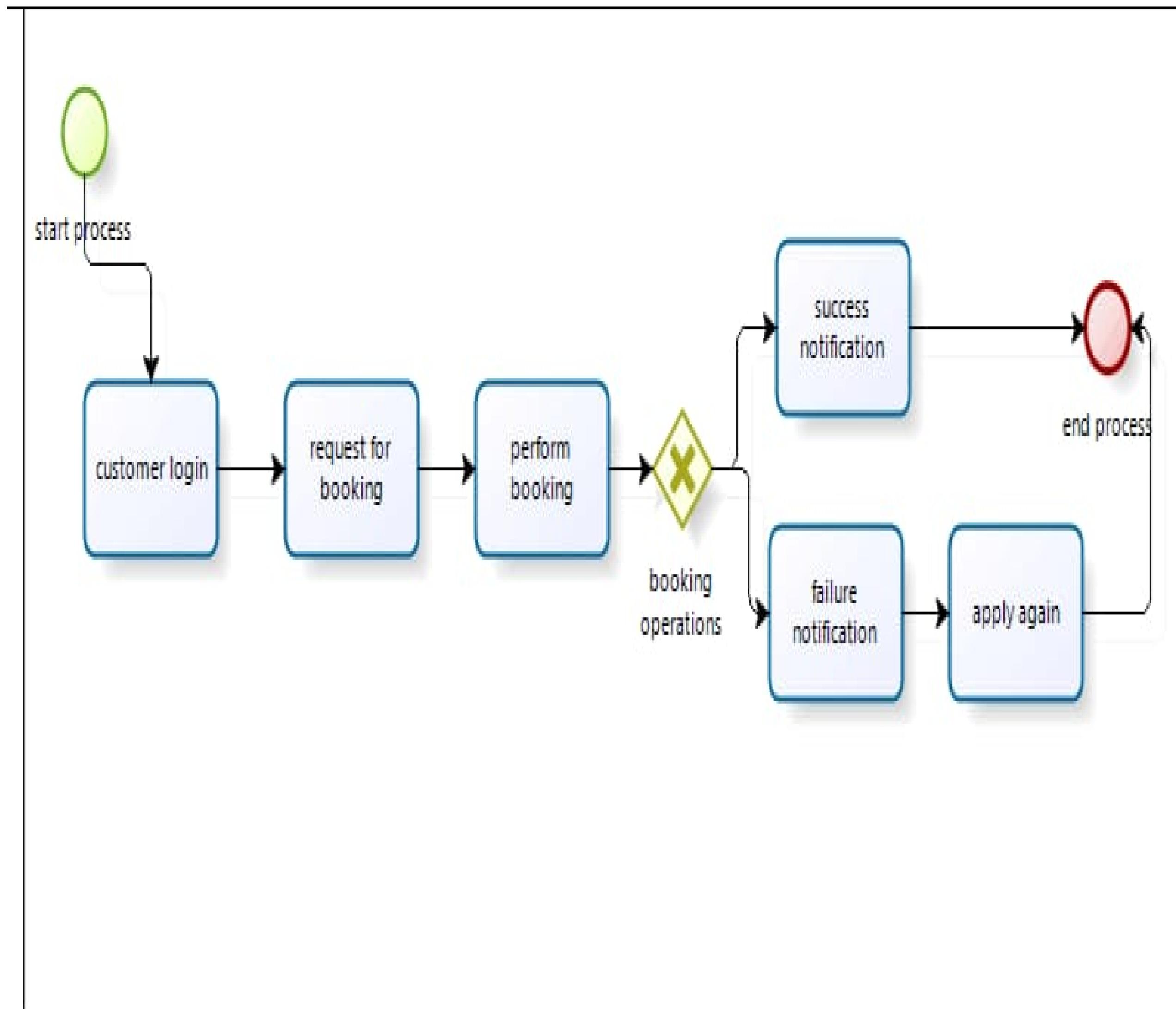
Online Ticket Sales and Distribution: Facilitates easy creation, customization, and sale of digital tickets for various events such as concerts, webinars, workshops, or conferences. Customers can purchase tickets, choose seating options, and receive e-tickets via email.

Customer Support and Helpdesk Management: Allows users to submit queries, complaints, or service requests, which are organized into a digital queue. Support teams can assign, prioritize, and track the progress of each ticket, ensuring issues are resolved promptly.

Workflow Automation: Automatically routes tickets based on predefined rules, such as type of issue, department, or agent availability. Automated notifications keep both users and staff informed about ticket status changes.

Analytics and Reporting: Provides insights into ticket volume, response time, and customer satisfaction through detailed analytics and custom reporting. This helps businesses optimize their processes and enhance service quality.

Flow chart:



Results/Findings:

CODE:

```
# Simple Ticket Booking System in Python

# In-memory list to store ticket bookings
tickets=[]

# Function to display the menu
def display_menu():
    print("\n--- Virtual Ticketing System ---")
    print("1. Book a Ticket")
    print("2. View All Bookings")
    print("3. Exit")

# Function to book a ticket
def book_ticket():
    print("\n--- Book a Ticket ---")
    customer_name=input("Enter Customer Name: ")
    event_name=input("Enter Event Name: ")
    try:
        ticket_quantity=int(input("Enter Number of Tickets: "))
    except ValueError:
        print("Invalid number. Please enter a valid integer.")
    return

# Store ticket details in the tickets list
ticket={
    'CustomerName':customer_name,
    'Event Name':event_name,
    'Number of Tickets':ticket_quantity
}
tickets.append(ticket)
print("Ticket booked successfully!")

# Function to view all bookings
def view_bookings():
    print("\n--- All Ticket Bookings ---")
    if len(tickets)==0:
        print("No bookings found.")
    else:
        for index,ticket in enumerate(tickets):
            print(f"\nBooking {index+1}:")
            print(f"Customer Name: {ticket['CustomerName']}")
```

```
# Main function to run the ticketing system
def run_ticketing_system():
    while True:
        display_menu()
        choice = input("Enter your choice (1/2/3):")

        if choice == '1':
            book_ticket()
        elif choice == '2':
            view_bookings()
        elif choice == '3':
            print("Exiting the system. Goodbye!")
            break
        else:
            print("Invalid choice. Please select a valid option.")

# Run the ticketing system
if __name__ == "__main__":
    run_ticketing_system()
```

OUTPUT:

Booking 1:

Customer Name: ABC

Event Name: CRICKET

Number of Tickets: 2

--- Virtual Ticketing System ---

1. Book a Ticket
2. View All Bookings
3. Exit

Enter your choice (1/2/3): 3

Exiting the system. Goodbye!

PS C:\Users\saman\OneDrive\Documents\sample\ds1.py> |

GRAPHS:

TICKET TRACKING

Waiting tickets

55

Processing tickets

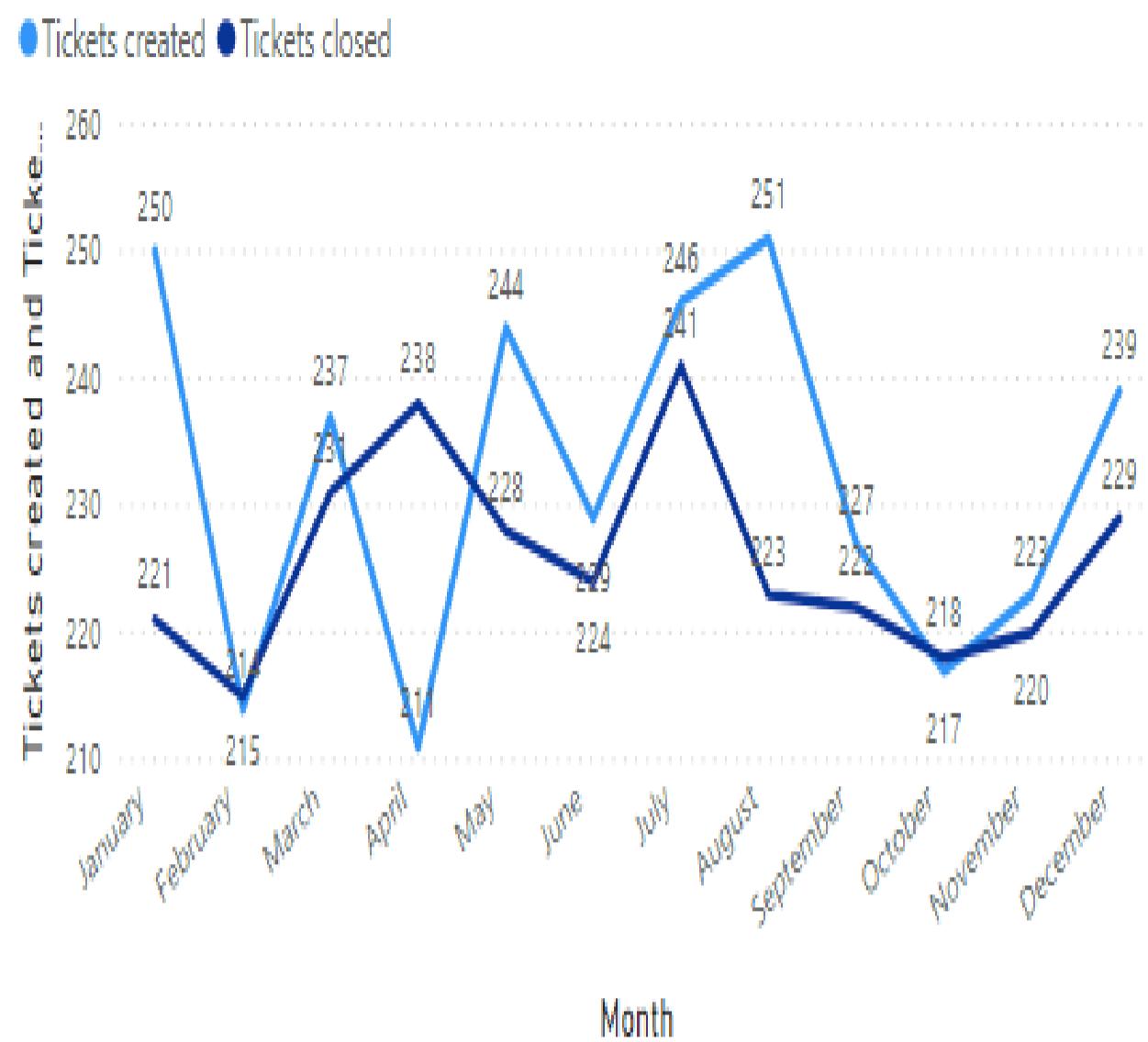
23

Closed tickets

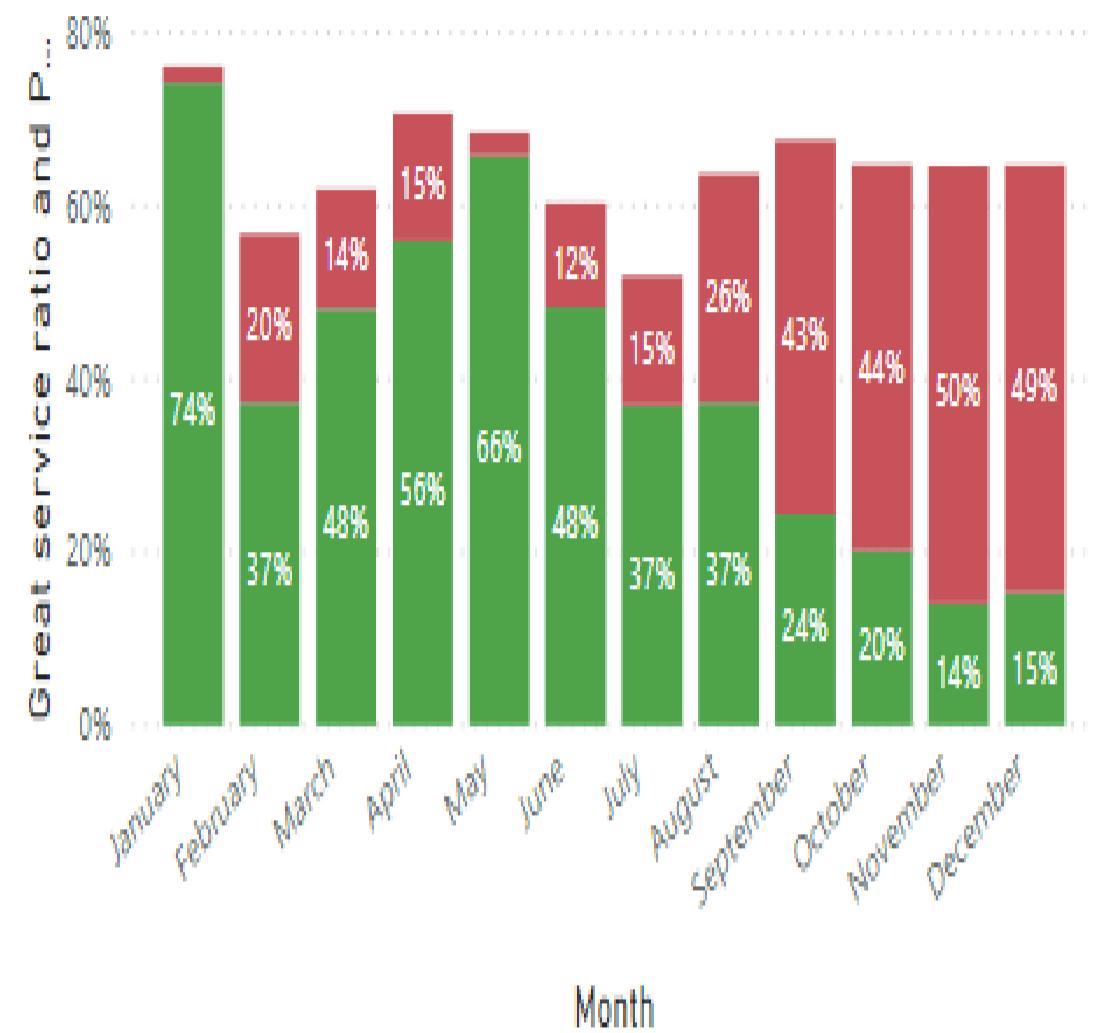
2710

FINANCE-BI.com

TICKETS BY MONTH



Great service ratio Poor service ratio



DETAIL BY AGENT

Agent	Total tickets	Tickets waiting and in process	Avg wait time (days)	Avg processing time (days)	Great service ratio	Poor service ratio
Henrik	686	21	3.30	2.30	42%	23%
Richard	665	21	2.87	2.14	45%	22%
Vincent	706	14	3.65	2.18	38%	27%
Yola	731	22	2.94	2.23	40%	23%
Total	2788	78	3.19	2.22	41%	24%

CONCLUSIONS :

The Virtual Ticketing System revolutionizes the way tickets are handled in various industries by offering a modern, user-friendly,

and efficient solution.

It caters to both users and administrators, ensuring smooth operations and providing an enhanced customer experience.

The Future Enhancement for our Project is:

- **AI-Driven Suggestions** : Use machine learning algorithms to analyze user preferences and suggest events, improving the likelihood of ticket sales.
- **Dynamic Pricing** : Implement algorithms that adjust prices based on demand, user history, and real-time data.

Social Features

- **Group Ticketing** : Allow users to purchase multiple tickets for friends and integrate with social media to coordinate events.
- **Event Sharing** : Enable users to share their tickets on social platforms, creating organic marketing opportunities.

REFERENCES:

- <https://www.greeksforgeeks.org> referred by Geeksforgeeks
- AUTOMATE THE BORING STUFF WITH PYTHON (2NDEDITION)

BY AL SWEIGART

-----THANK YOU---
