

**JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY**

**B. SC. COMPUTER SCIENCE**

**ICS 2209: DESIGN AND IMPLEMENTATION OF COMPUTER APPLICATIONS**

TITLE: **CONCERT MANAGEMENT SYSTEM**

GROUP MEMBERS:

1. Mark Horris Maina-SCT211-0704/2022
2. Emmanuel Yator SCT211-0622/2022
3. Sajjad Abdirahim Gedow SCT211-0065/2022
4. George Githigi Ngugi SCT211-0460/2022
5. Abdimajid Dakane Muktar SCT211-0028/2022
6. Njoroge Kanyagia SCT211-0009/2022

**Table of Contents**

[**1. PROBLEM STATEMENT** 3](#_Toc159554921)

[**1.1 Introduction** 3](#_Toc159554922)

[**1.2 Potential Clients and Users** 3](#_Toc159554923)

[**2. JUSTIFICATION** 4](#_Toc159554924)

[**3. REQUIREMENT SPECIFICATION** 5](#_Toc159554925)

[**3.1. FUNCTIONAL REQUIREMENTS** 5](#_Toc159554926)

[**3.1.1 Input Requirements** 5](#_Toc159554927)

[**3.1.2 Output Requirements** 5](#_Toc159554928)

[**3.1.3 Computations** 5](#_Toc159554929)

[**3.1.4 Data Management** 5](#_Toc159554930)

[**3.2 NON-FUNCTIONAL REQUIREMENTS** 5](#_Toc159554931)

[**3.2.1 Software Requirements** 5](#_Toc159554932)

[**3.2.2 Hardware Requirements** 6](#_Toc159554933)

[**4. FEASIBILITY STUDY** 6](#_Toc159554934)

[**4.1 SCHEDULE FEASIBILITY** 6](#_Toc159554935)

[**4.2 TECHNICAL FEASIBILITY** 6](#_Toc159554936)

[**4.3 OPERATIONAL FEASIBILITY** 6](#_Toc159554937)

[**4.4 ECONOMIC FEASIBILITY** 7](#_Toc159554938)

[**5. SYSTEM DESIGN** 7](#_Toc159554939)

[**5.1. DATABASE DESIGN** 7](#_Toc159554940)

[**5.2 INTERFACE DESIGN** 7](#_Toc159554941)

[**5.3 PROGRAM DESIGN** 8](#_Toc159554942)

[**5.4 FLOW CHART** 11](#_Toc159554943)

[**5.5 SEQUENTIAL DIAGRAM** 13](#_Toc159554944)

[**5.6 DATA FLOW DIAGRAM** 15](#_Toc159554945)

[**5.7 ARCHITECTURAL DESIGN** 16](#_Toc159554946)

# **1. PROBLEM STATEMENT**

## **1.1 Introduction**

A concert management system is to be developed to provide a booking system for both artists and their fans. It also acts as a link between the businesses and/or venue owners and the artists without necessarily having to meet physically. The system locates, contacts and confirms addresses of these venues knowing the rental price, capacity and location. They keep tabs with the business owners to easily alert artists of available venues and services to host concerts. The system closes the gap from the hectic work of planning concerts or fan meets and does the work for you.

Important things such as (electricity bills, food, hospitality, venue payment, transport) are all catered for. The company has a group of employees to plan, manage and oversee these meet-ups, including (bouncers, event managers, workers, etc.)

For businesses and/or venue owners; The system facilitates venue advertising on their platform using captivating pictures and videos, easy venue payment by the artists/company, easy payment of other services offered e.g. videography, food, sound, etc.; makes it easier to communicate with the artists themselves and provides opportunities for feedback which may increase their service ratings thus being favored in the systems’ algorithm.

For artists; The system helps in the advertisement/alert people of upcoming events, provides a list of available venues and businesses able to cater for their needed services for the events and their details such as: contacts, venue capacity, venue location, type of service, etc.; allows them to view pre-bookings to decide which venue (according to capacity) to rent and makes payment easy and secure (all-in-one place)

For the fans; The system shows the concert previews (the posters, venue, price) and allows them to view and book venue seats, the system also provides easy, secure payment of these tickets and generates unique identifier ID’s to confirm tickets at the venue entrance.

## **1.2 Potential Clients and Users**

The potential clients and users who could benefit from the **Concert Management System** include:

* **Event Managers and Promoters:** Event managers responsible for organizing concerts, festivals, and live performances can utilize the system to streamline venue booking, logistics, and ticketing. It simplifies their tasks, enhances communication, and ensures successful events.
* **Venue Owners and Managers:** Venue owners and managers, whether operating arenas, theaters, or outdoor spaces, can list their venues in the system. It provides them with exposure to high-profile artists and facilitates transparent negotiations.
* **Artists, his/her Crew and Staff:** The system benefits artists, crew members—sound engineers, lighting technicians, stage managers, and security personnel. It centralizes travel arrangements, and schedules, making their work more efficient.
* **Fans and Concertgoers:** Fans eagerly awaiting artist's concerts can benefit from streamlined ticketing. They can purchase tickets online, receive e-tickets, buy merchandise and stay informed about tour updates. The system enhances their overall concert experience.

# **2. JUSTIFICATION**

The purpose of creating this application is:

* The system streamlines venue selection, automates negotiations, and provides real-time availability information. It saves time and ensures fair terms for both parties.
* The system centralizes logistics, tracks crew schedules, and facilitates efficient communication. Crew members can focus on their tasks without administrative hassles.
* The system automates ticket sales, prevents fraud, and integrates merchandise offerings. Fans get a seamless experience, from ticket purchase to concert memorabilia.
* The system tracks expenses, revenue, and profitability. Financial reports aid decision-making, ensuring efficient resource allocation.
* The system allows the company staff to easily maintain information such as artist details, business details and artist orders.
* The system allows company staff to maintain records of company employees, expenses and profits.
* The system allows the company staff to easily maintain information such as artist details, business details and artist orders.
* The system generates reports to be used for decision making.

# **3. REQUIREMENT SPECIFICATION**

## **3.1. FUNCTIONAL REQUIREMENTS**

### **3.1.1 Input Requirements**

The system should take inputs to maintain records for: registered artists, businesses and the services they offer, venue owners and their details, venues capacities and their prices, registered fans, records of employees.

### **3.1.2 Output Requirements**

The system generates reports for the following: registered artists, registered fans, businesses, venue, owners, venues, venue prices, service expenses, tax payable, total expenses incurred by the company, total revenue generated by the company, overall profit the company has incurred.

### **3.1.3 Computations**

The system should calculate the total revenue from ticket sales. It should compute the total expenses. The system should calculate the profitability of each tour by subtracting the total expenses from the total revenue.

### **3.1.4 Data Management**

The system should store and manage data related to venues, including their availability, cost, and technical specifications. It should manage data related to ticket sales. The system should store and manage data related to the artists, fans and business owners like food vendors.

## **3.2 NON-FUNCTIONAL REQUIREMENTS**

### **3.2.1 Software Requirements**

* **Computing Platform:** The system should be a desktop application, compatible with major operating systems such as Windows, macOS, and Linux.
* **Technology to be used:** The system should be developed using VB.NET for programming, MySQL and ADO.NET for database connectivity and should be able to connect to a relational database for data storage and retrieval.

### **3.2.2 Hardware Requirements**

* **Processor:** The system should be able to run on a computer with a minimum of a 2.0 GHz processor.
* **Memory:** The system should require a minimum of 4GB RAM for smooth operation.
* **Storage:** The system should require a minimum of 1GB of free disk space for installation and additional space for data storage.

# **4. FEASIBILITY STUDY**

## **4.1 SCHEDULE FEASIBILITY**

|  |  |
| --- | --- |
| **ACTIVITY** | **DURATION** |
| System Analysis | 1 WEEK |
| System Design | 1 WEEK |
| System Coding and Construction | 4 WEEKS |
| Testing and debugging | 1 WEEK |
| System Implementation | 1 WEEK |
| TOTAL | 2 MONTHS |

## **4.2 TECHNICAL FEASIBILITY**

Technically, the system development requires developers proficient in the programming languages used to implement the system.

The people operating the system need to be computer literate and understand basic functioning of the system.

## **4.3 OPERATIONAL FEASIBILITY**

The new system is likely to change the work environment because some staff may have to be retrained, redeployed or replaced. The company is also expected to hire a qualified system administrator and more ICT staff

## **4.4 ECONOMIC FEASIBILITY**

The following is a comparison of the average monthly costs of managing the musical world tour system:

|  |  |  |
| --- | --- | --- |
|  | MANUAL SYSTEM | COMPUTERISED SYSTEM |
| ADMINISTRATION COST | 30,000 | 30,000 |
| ELECTRICITY BILLS | 3,000 | 6,500 |
| SALARIES | 90,000 | 45,000 |
| STATIONARY | 7,000 | 3,000 |
| TOTAL | 130,000 | 84,500 |

# **5. SYSTEM DESIGN**

## **5.1. DATABASE DESIGN**

The database is required in order to hold various types of information that is necessary for the functioning of the system. It is also where data is stored when it is entered from the registration forms. The data stored in the databases will be used to manage the operation of activities of the concert and evaluate performance of how much income is generated and the costs incurred in running the concerts.

The following tables are included in the database:

* VENUE\_OWNER\_DETAILs (**Owner ID**\_, Name, Phone number, Address,)
* VENUE\_DETAILS (**Venue ID**, Owner ID, Venue Name, Venue Location, Venue Capacity, Venue Rating, Venue Pricing )
* ARTIST\_DETAILS (**Artist ID**, Name, Citizenship , Genre, Phone number ,Address)
* CONCERT\_DETAILS (**Concert ID**, Venue ID, Business ID, Artist ID, Concert Name)
* TICKET\_DETAILS (**Ticket ID**, Fan ID, Concert ID, Payment ID, Ticket name, Ticket price)
* FAN\_DETAILS (**Fan ID**, Name, Phone number, Email address)
* PAYMENT\_DETAILS (**Payment ID**, Ticket price, Payment made)
* BUSINESS\_DETAILS (**Business ID**, Employee ID, Salary ID, Business Name, Service offered, Service expenses, Number of employees, Business rating)
* EMPLOYEE\_DETAILS (**Employee ID** , Salary ID, Name, Address, Phone number)
* SALARY\_DETAILS (**Salary ID**, Work Category, Payment)

## **5.2 INTERFACE DESIGN**

Below is a depiction of a part of our user interface that shows the visual and interactive elements of our system. It is both simple and allows users to feel in control. It encompasses modern design patterns hence giving the maximum user experience

|  |
| --- |
|  |

## **5.3 PROGRAM DESIGN**

1. UserInterface *(*UI*)*:

Design an intuitive user interface for easy interaction.

Include sections for artists, venues, bookings, ticket sales, and employee management.

1. Functionality:

ArtistsManagement: Add, edit, and delete artist profiles. View artist details including performance history, contact information, etc.

VenueManagement: Add, edit, and delete venue profiles. View venue details such as capacity, location, upcoming events, etc.

Bookings: Schedule performances by selecting artists and venues. View and manage bookings, including date, time, and venue details.

Ticket Sales: Track ticket sales for each event. Generate reports on ticket revenue, number of tickets sold, etc.

Employee Management: Manage employee profiles, roles, and salaries. Track work hours and payroll calculations.

Notifications: Send notifications for upcoming events, ticket sales milestones, etc.

DataManagement: Store and retrieve data related to artists, venues, bookings, ticket sales, and employees. Ensure data security and integrity.

1. Architecture:

Use a modular architecture for scalability and maintainability. Consider implementing a Model-View-Controller (MVC) or similar design pattern to separate concerns and improve code organization. Utilize appropriate frameworks and libraries.

1. Database:

Choose a database system (e.g., MySQL, PostgreSQL) to store application data. Design database schemas for storing information related to artists, venues, bookings, ticket sales, employees, etc.

1. Integration:

Integrate APIs or services for features like payment processing, email notifications, etc., if needed. Ensure seamless communication between different components of the application.

1. Testing:

Develop unit tests to verify the functionality of individual components. Perform integration testing to ensure proper interaction between different modules.

1. Documentation:

Create comprehensive documentation covering the design, implementation, and usage of the application. Include instructions for installation, configuration, and troubleshooting.

For a project like this, we'll be using agile methodology. Here's how you can incorporate Agile into the program design:

AgileMethodology

- Adopt an Agile approach to facilitate iterative development and quick adaptation to changes.

- Break down the project into manageable chunks or "sprints" with specific goals for each iteration.

- Conduct regular meetings, such as daily stand-ups, to discuss progress, address issues, and plan upcoming tasks.

- Encourage collaboration between team members, including developers, designers, and stakeholders, to ensure alignment with project goals.

- Emphasize flexibility and responsiveness to feedback, allowing for adjustments to be made throughout the development process.

IterativeDevelopment:

- Plan development cycles (sprints) with defined goals and prioritize features based on customer needs and feedback.

- Begin with a minimum viable product (MVP) that includes essential functionality and gradually add additional features in subsequent iterations.

- Continuously gather feedback from stakeholders, users, and team members to refine and improve the application over time.

User Involvement:

- Involve users and stakeholders in the development process through regular feedback sessions, demos, and user acceptance testing.

- Solicit input on features, usability, and overall satisfaction to ensure that the application meets their needs and expectations.

Adaptability:

- Remain open to change and ready to adjust project priorities or requirements based on new information or shifting business needs.

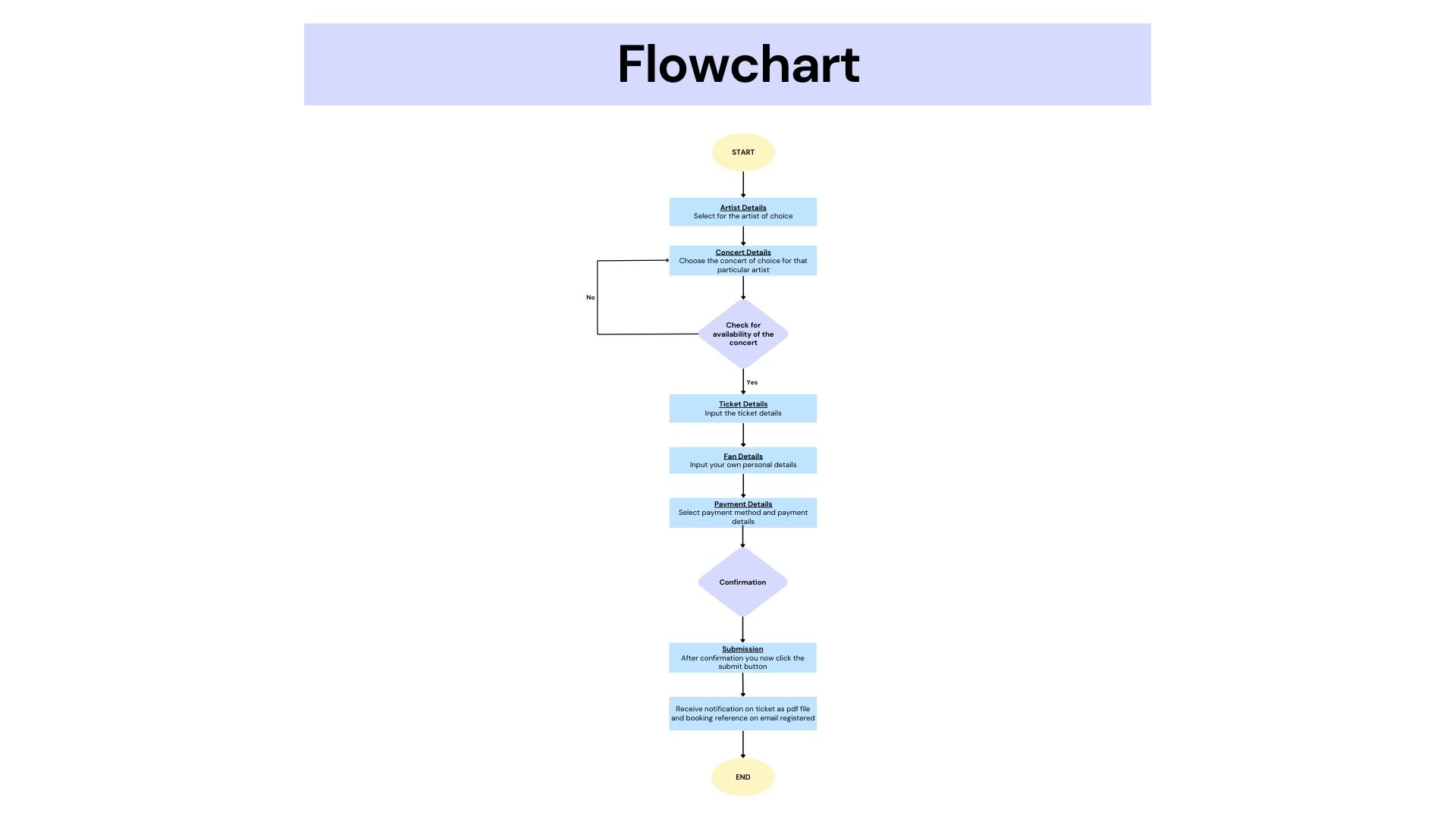
- Prioritize tasks based on their importance and potential impact, allowing the team to focus on delivering the most value with each iteration.

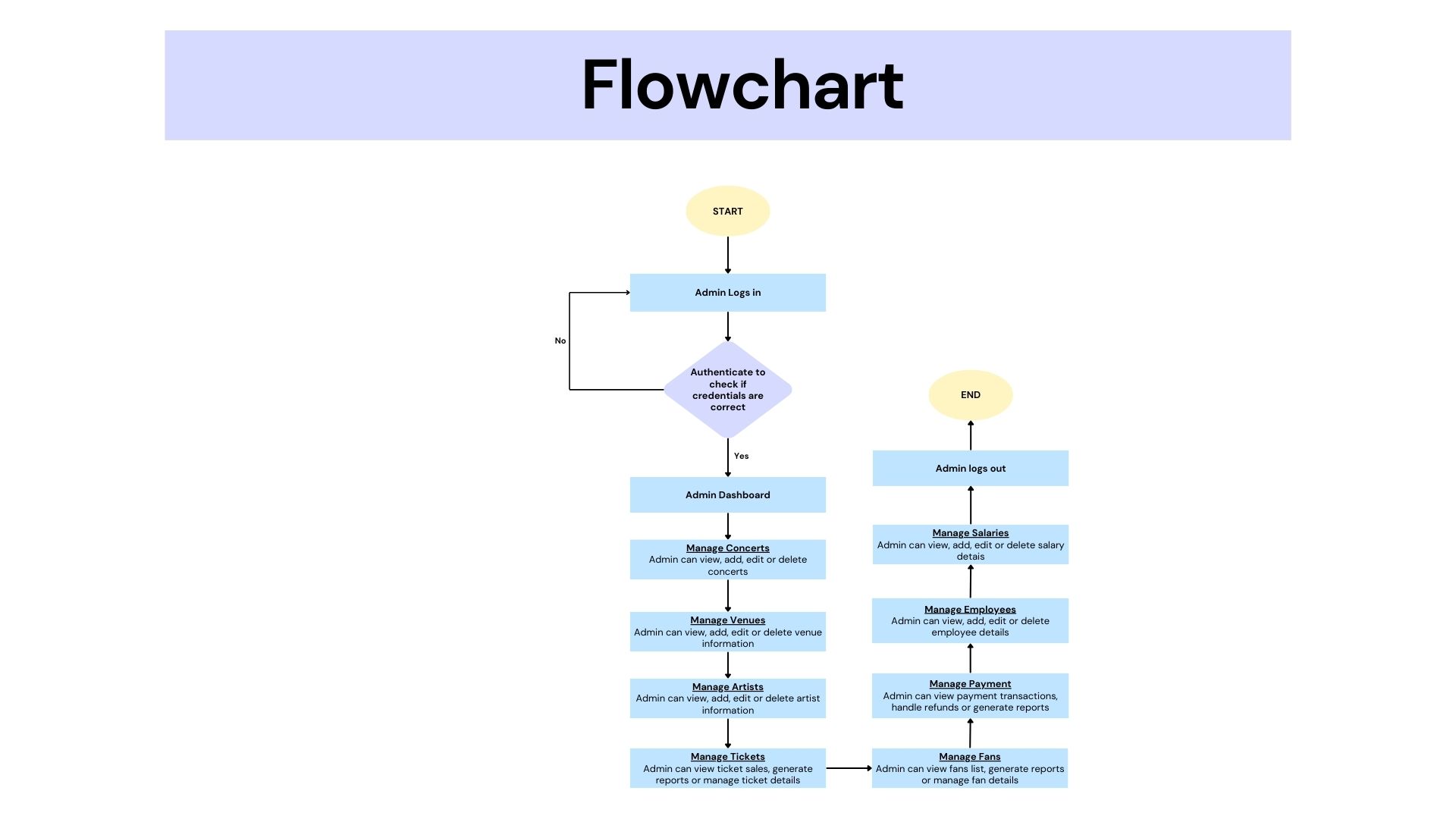
Continuous Improvement:

- Reflect on the team's performance and the effectiveness of development practices during regular retrospectives.

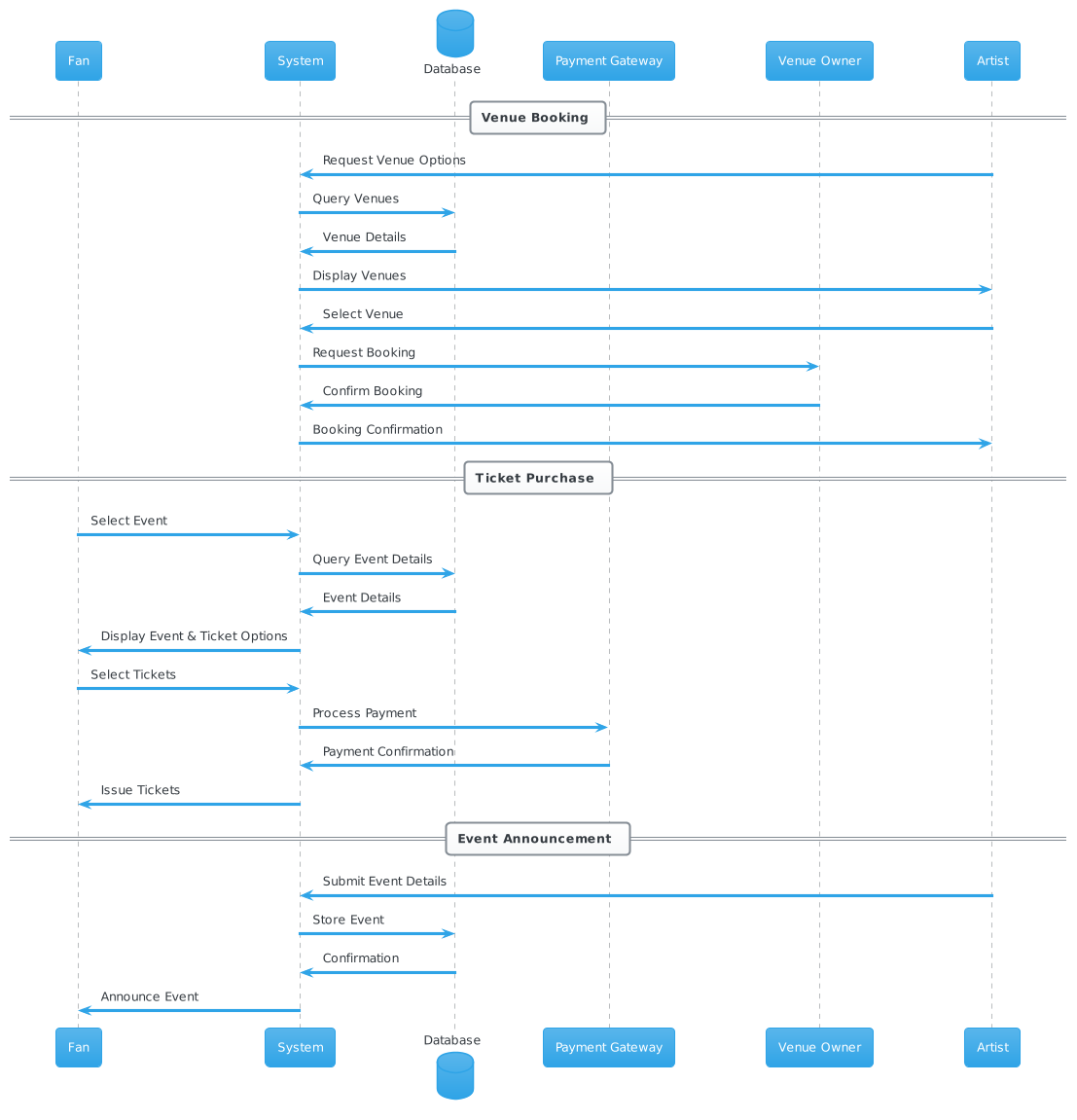
- Identify areas for improvement and implement changes to enhance productivity, quality, and collaboration

## **5.4 FLOW CHART**





## **5.5 SEQUENTIAL DIAGRAM**



Creating sequence diagrams for the entire Concert Management System as described in your document involves illustrating the interactions between different system components and actors (such as artists, venue owners, and fans) for various functionalities. Given the complexity and breadth of the system, I'll outline the steps for generating sequence diagrams for key functionalities:

1. Venue Booking Process

Actors: Artist/Event Manager, System, Venue Owner

Process:

The artist/event manager searches for venues.

The system displays available venues.

The artist/event manager selects a venue and requests a booking.

The system sends the booking request to the venue owner.

The venue owner approves or declines the booking.

The system updates the artist/event manager with the booking status.

2. Ticket Booking and Payment Process

Actors: Fan, System, Payment Gateway

Process:

The fan selects a concert and chooses seats.

The system confirms the availability and shows the price.

The fan proceeds to payment.

The system processes the payment through the payment gateway.

The payment gateway confirms the payment.

The system issues tickets to the fan.

3. Venue Listing and Advertisement Process

Actors: Venue Owner, System

Process:

The venue owner submits venue details for listing.

The system validates the information and lists the venue.

The venue owner requests to advertise the venue.

The system schedules and displays the advertisement.

The system provides feedback and analytics on the advertisement's reach.

4. Event Creation and Announcement Process

Actors: Artist, System

Process:

The artist submits details for the event creation.

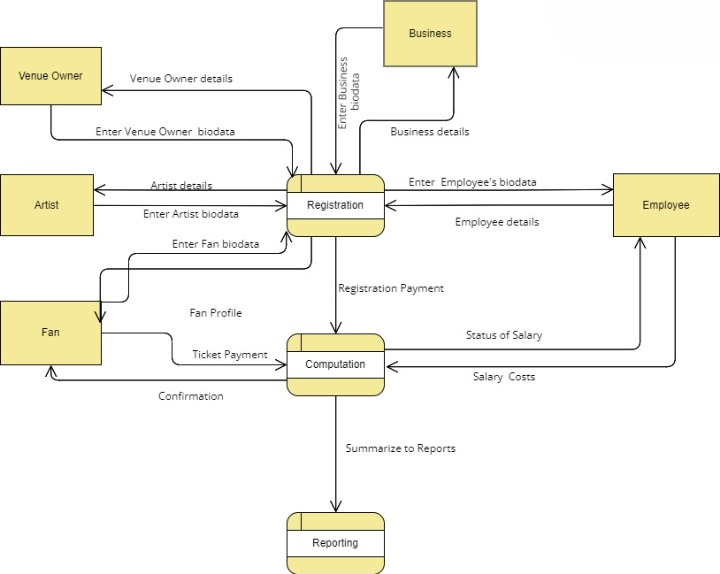
The system validates and creates the event.

The artist requests to announce the event.

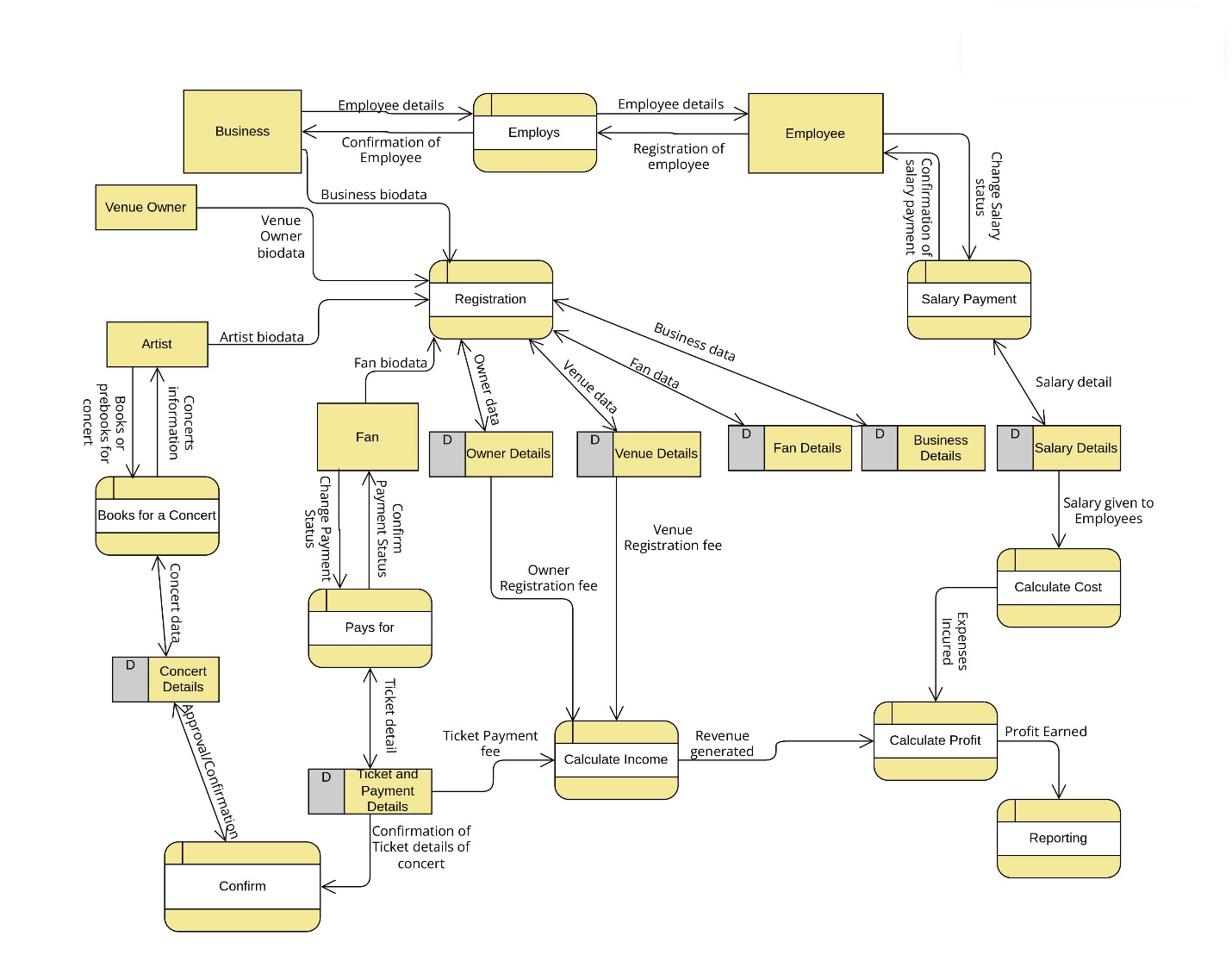
The system distributes the announcement through various channels.

The system tracks and reports on the announcement's reach and engagement.

## **5.6 DATA FLOW DIAGRAM**

On decomposition of the application, below is a representation of the level 0 dataflow diagram of the application:  


The level 1 decomposition of the application is as follows:



## **5.7 ARCHITECTURAL DESIGN**

Requirements

Person should be able to book venue/ticket based on:

1. Current city
2. Concert
3. Venues of the concert
4. See available seat and book

System should handle concurrency

System should be secure as it will be dealing with money and payment

Database should handle transactions as well as unstructured data such as reviews and comments from fans

**APIs**

GetCities

GetVenueByCity

GetAvailableSeatS

SelectSeatbyPrice

BookSeat

SendTicketByMailOrSMS

PostCommentsAndReviews

