Ticket Purchase System

Justin Wasser

3/21/2023

Introduction:

The purpose of the ensuing project is to analyze and design a ticket purchase system used for concerts, then create a project plan to implement the system after the design has been completed. As such, the project is separated into three distinct phases: phase 1. requirement analysis, phase 2. system and database design, and phase 3. project plan (the scheduling section of the project plan will be submitted separately via MS Project).

Phase 1: Requirement Analysis

A. Problem definition

1. Concert tickets are currently unable to be purchased online, therefore a computer information system is needed that will enable tickets to be purchased online.

B. Issues

1. Customers' likes and dislikes concerning user interfaces are not currently known (Defining the Problem in Project Initiation, 2014).
2. It is challenging to satisfy customers without understanding their preferences.
3. It is challenging to improve revenue without pleasing customers.
4. Not known if the desired system is feasible (Le, n.d. -b; Goldsmith, 2009).
    - Perform a feasibility study.
        - Include an analysis of functional, performance, interfaces, design, and development requirements (Conger, 2008).

C. Objectives

1. The system must be able to perform the following functions:
    - Process ticket orders.
    - Process customer name and credit card information.
    - Send customer information (name, credit card number and order cost) to the correct credit card company.
    - Interface with credit card companies to inform the system whether the ticket order should be approved or declined.
    - Inform customers if their order is rejected.

- ♦ Transfer a record of the transaction to the concert management company.
- ♦ Distribute ticket(s) virtually to the customer.
- ♦ Distribute an invoice virtually to the customer.
- ♦ Store all relevant concert information.
- ♦ Store all relevant ticket inventory data.
- ♦ Store all sales history data.

D. Requirements

1. The system must be user-friendly, so customers are able to use it (Defining the Problem in Project Initiation, 2014).

2. The system must be able to protect the confidentiality of customer information collected and stored (Defining the Problem in Project Initiation, 2014).

    - ♦ Limit access to the system's database to only authorized individuals (Defining the Problem in Project Initiation, 2014).

3. The systems' technical infrastructure must only be accessed by authorized individuals (Defining the Problem in Project Initiation, 2014).

4. Rank the importance/value of different system features based on stakeholder feedback using the joint application development (JAD) method (NIH, n.d.).

    - ♦ Document all stakeholder requirements gathering sessions including the date(s), session length, and stakeholders involved (NIH, n.d.).

5. Obtain feedback from stakeholders on each component of the system during development using agile methodology to ensure that the system design is optimal (Tilley & Rosenblatt, 2017).

    - ♦ Document all alterations to system specifications, purpose, scope, etc., including the date of the alteration and its justification (NIH, n.d.).

E. Constraints

1. The tentative deadline to deliver the completed system is unknown.

- ♦ Inform business leaders that they will likely face several choices pitting nonessential system features against time to project completion (English, 2021).

2. The initial budget allocation for the creation of this system is unknown.

- ♦ Inform business leaders that they will likely face several choices pitting nonessential system features against budget constraints (English, 2021).

3. There are an unknown number of employees involved in this project.
4. Not specified who will make determinations regarding the project budget and schedule once the project has begun (NIH, n.d.).
5. Collecting, and/or storing the personal data of users is heavily regulated.

    - What type of personal data will be gathered?
        - What territories will this system be used in?
            - What are the personal data compliance laws/regulations in each of those territories for each category of personal data gathered by the system?

7. Unknown if a knowledge management strategy is within the scope of this project team.

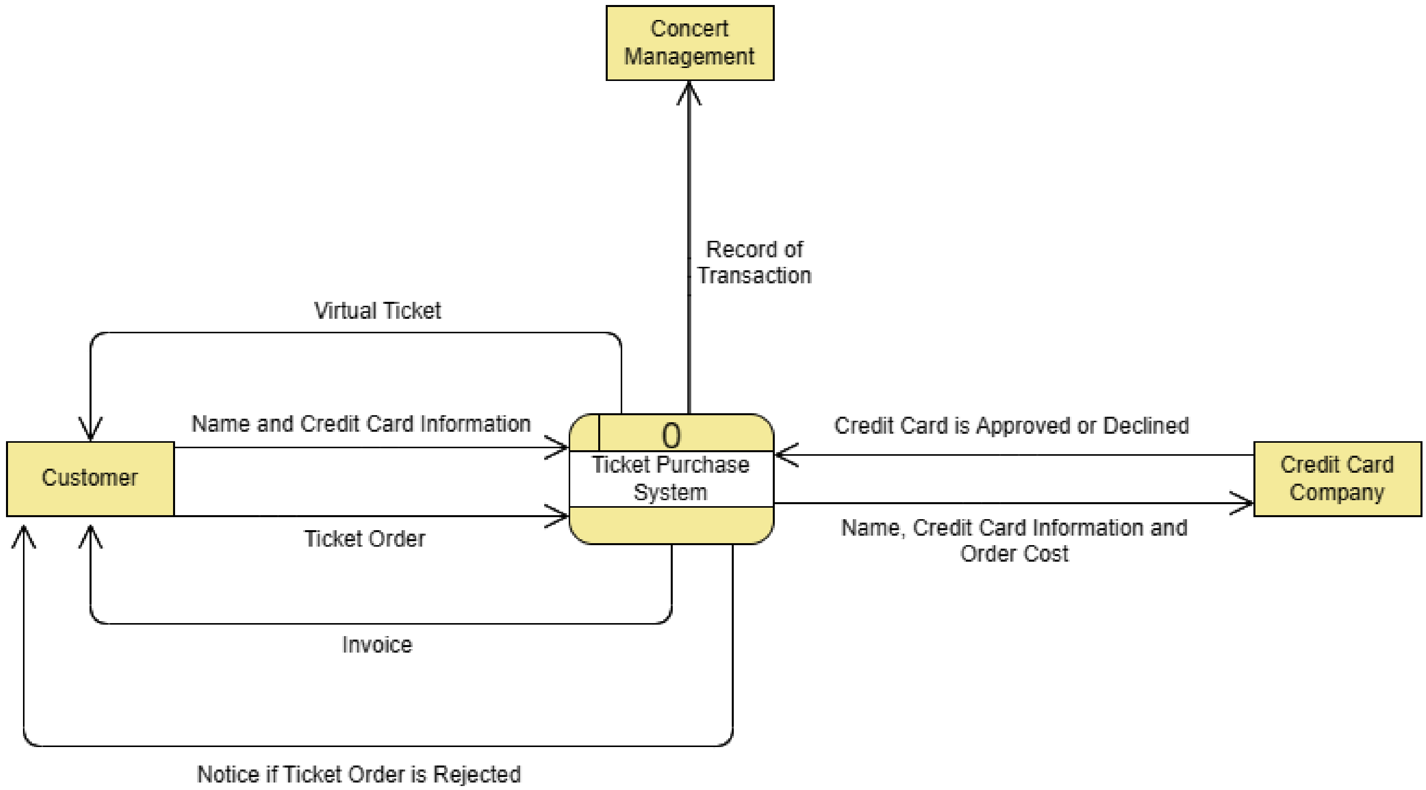
    - If yes, is there a knowledge management strategy and/or system in place to analyze data generated from the proposed system (English, 2021)?
        - Perform a "Knowledge Requirements Analysis" (English, 2021).

F. Description of the proposed system

1. The proposed ticket purchase system will be able to display upcoming concert information, the purchasable tickets to those concerts, and the price of those tickets. Furthermore, the system will be capable of processing customers' purchase requests, sending customer information (name, credit card number and order cost) to the correct credit card company, and then interfacing with credit card companies to inform the system whether to approve or decline customer transactions; if declined, the system will inform customers if their ticket order was declined. If the transaction is approved, the system will record the transaction history and will send a record of the transaction to the concert management company. The system will then distribute the purchased ticket(s) virtually to the customer, and distribute an invoice to the customer. Lastly, the system will update the ticket inventory data store to remove the purchased tickets from the available inventory.

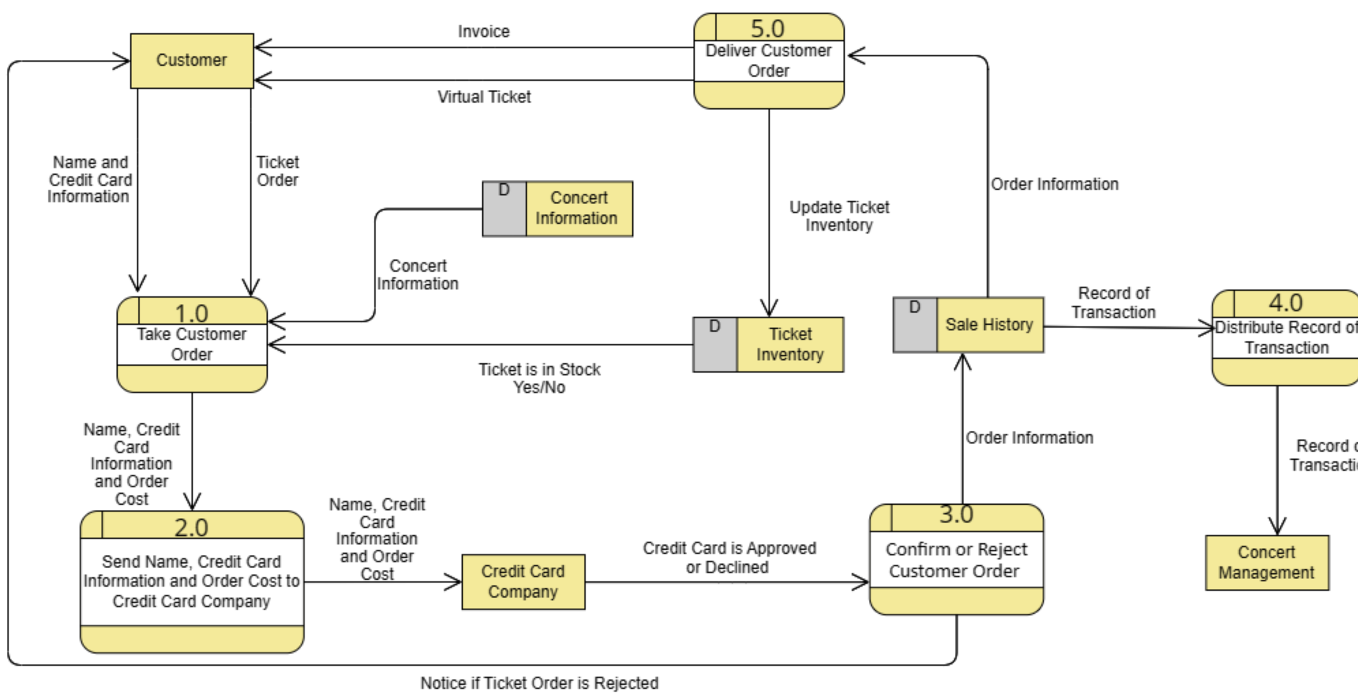
G. Logical model design

1. *Data flow diagrams*
    - **Context Diagram**



(Conger, 2008; SmartDraw, 2019, *Visual Paradigm Online*, n.d.)

- **Diagram 0**



(Conger, 2008; SmartDraw, 2019, *Visual Paradigm Online*, n.d.)

- **Descriptions of processes in each diagram**

2. **Descriptions of outputs/inputs/performance/security or controls**

Inputs:

Context Diagram:

Inputs:

1. Ticket order

♦ The customer requests to purchase one or many ticket(s) from the ticket purchase system.

2. Name and credit card information

♦ The customer inputs their name and credit card information into the ticket purchase system.

3. Credit card is approved or declined

♦ The customer's credit card company relays to the ticket purchase system whether the customer's credit card was approved or declined.

Outputs:

1. Name, credit card information and order cost

♦ The ticket purchase system relays a customer's name, credit card information and the order cost to the customer's credit card company.

2. Notice if ticket order is rejected

♦ The ticket purchase system sends a notice to the customer if their ticket order was rejected.

3. Record of transaction

♦ A record of every transaction is sent to the concert management company so they can track how much money they are owed.

4. Virtual ticket

♦ For approved transactions, the ticket purchase system distributes a virtual ticket to the customer.

5. Invoice

♦ For approved transactions, the ticket purchase system distributes an invoice to the customer for the virtual ticket.

Performance:

Process Name:

1. Ticket Purchase System

   - ♦ The ‘ticket purchase system’ process takes a customer’s desired ticket order information as well as their credit card information, to pay for the desired ticket. Next, the ‘ticket purchase system’ relays the customers’ name, credit card information, and the amount of money that the customer needs from their credit card to approve their ticket order. Then ‘ticket purchase system’ process notifies the customer that their ticket order has been rejected if their credit card company declined to transfer the amount of money requested by the customer. However, if the credit card was approved by their credit card company, a record of their transaction is transferred to the external entity ‘concert management’. Next, data that constitutes the customer’s virtual ticket is sent to the external entity ‘customer’ as well as the data that constitutes a virtual invoice.

Security/Controls:

1. Require a customer’s credit card to be approved before finalizing a ticket(s) sale.
2. Require a password to log in to the ticket purchase system’s RDBMS.

♦ Require two-factor authentication, in the form of a code sent to the user’s cell phone.

3. Employ system access logging technology to track and document historical RDBMS access.
4. Employ RDBMS monitoring tools to monitor traffic for unusual activity.

Diagram 0:

Inputs:

1. Ticket order

♦ The customer requests to purchase a ticket from the ticket purchase system.

2. Name credit card information

♦ The customer inputs their name and credit card information to purchase a ticket from the ticket purchase system.

3. Credit card is approved or declined

♦ The customer’s credit card company relays to the ticket purchase system whether the customer’s credit card was approved or declined.

Outputs:

1. Name, credit card information and order cost

- The ticket purchase system relays a customer's name, credit card information and the order cost to the customer's credit card company.

2. Notice if ticket order is rejected

- The ticket purchase system sends a notice to the customer if their credit card company declined their credit card.

3. Record of transaction

- A record of every transaction is sent to the concert management company so they can track how much money they are owed.

4. Virtual ticket

- For approved transactions, the ticket purchase system distributes a virtual ticket to the customer.

5. Invoice

- For approved transactions, the ticket purchase system distributes an invoice to the customer for the virtual ticket.

Performance:

Process Names:

1. Take customer order
    - The 'take ticket order' process pulls information from the 'concert information' and 'ticket inventory' data stores, to provide the customer with concert and ticket information and pricing. The customer then selects their desired concert(s) ticket(s) and inputs their credit card information, to pay for their order.
2. Send name, credit card information and order cost to credit card company
    - The 'send name, credit card information and order cost to credit card company' relays the customers' name, credit card information and the amount of money that the customer needs from their credit card to approve their ticket order to the appropriate credit card company.
3. Confirm or reject customer order
    - The 'confirm or reject ticket order' process receives input from the external entity 'credit card company' informing the process of whether the funds needed to complete the transaction will be allocated. If the answer is no, the process notifies the customer that their ticket order has been rejected. If the answer is yes, their order is accepted and the transaction information is transferred to the data store 'sale history'.
4. Distribute record of transaction

- The ‘distribute record of transaction’ process receives an input of every sale from the data store ‘sale history’ and sends a record of every transaction to the concert management company.

5. Deliver customer order
   - The ‘deliver customer order’ process receives an input of order information from the data store ‘sale history’ and then uses that information to distribute the data that constitutes the virtual ticket to the customer, as well as the data that constitutes a virtual invoice. Lastly, the number of tickets bought and to which concert is transferred to the datastore ‘ticket inventory’ so that the ticket inventory can be updated after the purchase.

Security/Controls:

1. Require a customer’s credit card to be approved before finalizing a ticket(s) sale.
2. Require a password to log in to the ticket purchase system’s RDBMS.

- Require two-factor authentication, in the form of a code sent to the user’s cell phone.

3. Employ system access logging technology to track and document historical RDBMS access.
4. Employ RDBMS monitoring tools to monitor traffic for unusual activity.

H. Specific requirements, if any (interface, operational, resource, performance, etc.)

1. A device with an updated web browser.

Phase 2: System and Database Design

A. User interface

1st Screen (Home Screen)

The physical layout for each input:

- **Screen** – The physical layout of the 1st screen will consist of a header section at the top of the screen, a “body of the screen” (Conger, 2008) section in the middle that takes up the majority of the screen space, and a footer section at the bottom of the screen (Conger, 2008).
- **Commands** – N/A
- **Controls** – Controls options will be located on the left side of the body of the screen (Conger, 2008; Ticketmaster, n.d.). Moreover, controls will appear in the

body of the screen when users point their mouse (or finger on touch screen) at an interactable part of the screen (Conger, 2008; Ticketmaster, n.d.).

- **Features –** Features locations will coincide with the commands and control options they are attached to.

The input design and procedures:

- **Screen –** When users interface with the home screen of the ticket purchase system they will see a header that displays the name of the concerts taking place, the date of that concert, and the physical location of the concert (Conger, 2008; Ticketmaster, n.d.). In "the body of the screen" (Conger, 2008) there will be an option to select any available concert, which will display the selected concert venue's digital replica with an accompanying display of seats that are available to purchase within the same webpage (Conger, 2008; Ticketmaster, n.d.). Furthermore, areas that denote where tickets are available will be highlighted in light red, while areas that are not purchasable will be dark greyed out to indicate as such (Conger, 2008; Ticketmaster, n.d.). Finally, "the footer section" (Conger, 2008) will be used to display menu navigation controls, however, this is not applicable on the home screen (Conger, 2008).
- **Commands –** N/A
- **Controls –**User controls will include being able to select any available concert in the left section of the body of the screen by clicking on it. Furthermore, the concert options will be in displayed via black lettering against a light blue background in a rectangularly outlined button. Once in the seat selection phase, users can select any available seat (i.e., ticket) by clicking on it, after which a purchase ticket button will appear on the lower right side of the body of the screen (Conger, 2008; Ticketmaster, n.d.). Lastly, users can also search all available tickets by price (ascending and descending).
- **Features –** When the user hovers over a selectable option (such as an available seat/section, or purchase/cancel button) that selectable graphic/button will slightly darken and a border outline will appear around that graphic/button (Conger, 2008; Ticketmaster, n.d.). All buttons will be rectangular and they will all be uniform in size (Conger, 2008).

<u>2nd Screen (Purchase Screen)</u>

The physical layout for each input:

- **Screen –** The physical layout of the purchase screen will consist of a header section at the top of the screen, a "body of the screen" (Conger, 2008) section in the middle that takes up the majority of the screen space, and a footer section at the bottom of the screen (Conger, 2008).

- **Commands –** Commands to input the user's first name, last name, and their credit card information in their respective input boxes will be located in the center of the body of the screen (Conger, 2008; Ticketmaster, n.d.).
- **Controls –** All control options will be located in the center of the body of the screen (Conger, 2008; Ticketmaster, n.d.).
- **Features –** Features locations will coincide with the commands and control options they are attached to.

The input design and procedures:

- **Screen –** The ticket or tickets that users selected to purchase on the home (1st) screen will appear on the right side of the body of the screen. The purchase input commands and controls will be present in the center of the body of the screen (Conger, 2008; Ticketmaster, n.d.). Lastly, a return to the previous screen button will be present at the bottom (footer) of the screen (Conger, 2008).
- **Commands –** The prompts for users to input their first name, last name and credit card information will be in black lettering against a white background. The 'cancel transaction' button, a 'return to the previous screen' button will be present at the bottom (footer) of the screen in black lettering against a light grey background in a rectangularly outlined button (Conger, 2008).
- **Controls –** A 'finalize purchase' button will be clickable (via user input) and the button will be in black lettering against a light green background contained within a rectangularly outlined button. The 'cancel transaction' button, and a 'return to the previous screen' button will be in black lettering against a light gray background contained within a rectangularly outlined button (Conger, 2008).
- **Features –** When the user hovers over a selectable option (such as a 'finalize purchase' button or a 'cancel transaction' button) that selectable graphic/button will slightly darken and a border outline will appear more prominently around that graphic/button (Conger, 2008; Ticketmaster, n.d.).

The physical layout, design, and procedures for each output:

1. **Name, credit card information and order cost**

- **Screen –** The physical layout of the 'name, credit card information and order cost' screen will consist of a header section at the top of the screen, a "body of the screen" (Conger, 2008) section in the middle that takes up the majority of the screen space, and a footer section at the bottom of the screen (Conger, 2008). The information will be displayed in the center of the body of the screen in black lettering against a white background. A 'copy information' button will be present in the bottom right corner of the body of the screen in black lettering against a light gray background.
- **Commands –** N/A

- **Controls –** A ‘copy information’ button will be clickable and will allow the information on the screen to be copied.
- **Features –** When the user hovers over the ‘copy information’ button that selectable graphic/button will slightly darken and a border outline will appear most prominently around that graphic/button (Conger, 2008; Ticketmaster, n.d.).

2. Notice if ticket order is rejected

- **Screen –** The physical layout of the ‘notice if ticket order is rejected’ screen will consist of a header section at the top of the screen, a “body of the screen” (Conger, 2008) section in the middle that takes up the majority of the screen space, and a footer section at the bottom of the screen (Conger, 2008). A ‘try another payment method’ button will be present in the middle of the body of the screen in black lettering against a light blue background.
- **Commands –** N/A
- **Controls –** A ‘try another payment method’ button will be clickable (via user input), that when selected will return the user to the ‘purchase screen’ (Conger, 2008).
- **Features -** When the user hovers over the ‘try another payment method’ button that selectable graphic/button will slightly darken and a border outline will appear most prominently around that graphic/button (Conger, 2008; Ticketmaster, n.d.).

3. Record of transaction

- **Screen –** The physical layout of the ‘record of transaction’ screen will consist of a header section at the top of the screen, a “body of the screen” (Conger, 2008) section in the middle that takes up the majority of the screen space, and a footer section at the bottom of the screen (Conger, 2008). The information will be displayed in the center of the body of the screen in black lettering against a white background. A ‘copy record of transaction’ button will be present in the bottom right corner of the body of the screen in black lettering against a light gray background.
- **Commands –** N/A
- **Controls -** A ‘copy record of transaction’ button will be clickable and will allow the information on the screen to be copied.
- **Features –** When the user hovers over the ‘copy record of transaction’ button that selectable graphic/button will slightly darken and a border outline will appear most prominently around that graphic/button (Conger, 2008; Ticketmaster, n.d.).

4. Virtual ticket

- **Screen –** The physical layout of the ‘virtual ticket’ screen will consist of a header section at the top of the screen, a “body of the screen” (Conger, 2008) section in the middle that takes up the majority of the screen space, and a footer section at the bottom of the screen (Conger, 2008). The virtual ticket will be displayed in the center of the body of the screen in black lettering against a white background. A ‘download virtual ticket’ button will be present in the bottom right corner of the body of the screen in black lettering against a light blue background.

- **Commands –** N/A
- **Controls -** A ‘download virtual ticket’ button will be clickable and will allow the information on the screen to be downloaded to the user’s device.
- **Features -** When the user hovers over the ‘download virtual ticket’ button that selectable graphic/button will slightly darken and a border outline will appear most prominently around that graphic/button (Conger, 2008; Ticketmaster, n.d.).

5. **Invoice**

- **Screen –** The physical layout of the ‘invoice’ screen will consist of a header section at the top of the screen, a “body of the screen” (Conger, 2008) section in the middle that takes up the majority of the screen space, and a footer section at the bottom of the screen (Conger, 2008). A ‘download invoice’ button will be present in the bottom right corner of the body of the screen in black lettering against a light gray background.
- **Commands –** N/A
- **Controls -** A ‘download invoice’ button will be clickable and will allow the information on the screen to be downloaded to the user’s device.
- **Features -** When the user hovers over the ‘download invoice’ button that selectable graphic/button will slightly darken and a border outline will appear most prominently around that graphic/button (Conger, 2008; Ticketmaster, n.d.).

B. Data design

Unnormalized

CONCERT-DATA

Concert ID	Concert Name	Concert Date	Ticket ID	Ticket Name	Ticket Price	Ticket Inventory	Customer First and Last Name	Customer Credit Card Number	Ticket Quantity Purchased	Sales Total
1	Shadows	3/21/23	1	General Admission	$50	100	Al, Abe	1111-1111-1111-1111	1	$50
1	Shadows	3/21/23	2	Backstage	$100	10	Bob, Burns	2222-2222-2222-2222	1	$100
1	Shadows	3/21/23	3	VIP	$500	2	Cam, Caan	3333-3333-3333-3333	2	$525
2	Final	4/4/23	4	Stadium	$25	50	Dan, Dune	4444-4444-4444-4444	1	$25

(Harrington, 2009; Le, n.d)

Primary key = N/A

First Normal Form (1NF)

CONCERT-DATA

Concert ID	Concert Name	Concert Date
1	Shadows	3/21/23
2	Final	4/4/23

(Harrington, 2009; Le, n.d)

Primary key = Concert ID

TICKET-CUSTOMER

Ticket ID	**Customer ID**	Ticket Name	Ticket Price	Ticket Inventory	Customer First Name	Customer Last Name	Customer Credit Card Number	Ticket Quantity Purchased	Sales Total
1	1	General Admission	$50	100	Al	Abe	1111-1111-1111-1111	1	$50
2	2	Backstage	$100	10	Bob	Burns	2222-2222-2222-2222	1	$100
3	3	VIP	$500	2	Cam	Caan	3333-3333-3333-3333	2	$525
4	4	Stadium	$25	50	Dan, D	Dune	4444-4444-4444-4444	1	$25

(Harrington, 2009; Le, n.d)

Primary key = Ticket ID and Customer ID

Second Normal Form (2NF)

CONCERT-DATA

Concert ID	Concert Name	Concert Date
1	Shadows	3/21/23
2	Final	4/4/23

(Harrington, 2009; Le, n.d)

Primary key = Concert ID

TICKET

Ticket ID	Ticket Name	Ticket Price	Ticket Inventory
1	General Admission	$50	100
2	Backstage	$100	10
3	VIP	$500	2
4	Stadium	$25	50

(Harrington, 2009; Le, n.d)

Primary key = Ticket ID

CUSTOMER-QUANTITY-SALE

Customer ID	First Name	Last Name	Customer Credit Card Number	Ticket Quantity Purchased	Sales Total
1	Al	Abe	1111-1111-1111-1111	1	$50
2	Bob	Burns	2222-2222-2222-2222	1	$100
3	Cam	Caan	3333-3333-3333-3333	2	$525
4	Dan, D	Dune	4444-4444-4444-4444	1	$25

(Harrington, 2009; Le, n.d)

Primary key = Customer ID

<u>Third Normal Form (3NF)</u>

CONCERT-DATA

<u>Concert ID</u>	Concert Name	Concert Date
1	Shadows	3/21/23
2	Final	4/4/23

(Harrington, 2009; Le, n.d)

Primary key = Concert ID

TICKET

<u>Ticket ID</u>	Ticket Name	Ticket Price	Ticket Inventory
1	General Admission	$50	100
2	Backstage	$100	10
3	VIP	$500	2
4	Stadium	$25	50

(Harrington, 2009; Le, n.d)

Primary key = Ticket ID

CUSTOMER

<u>Customer ID</u>	First Name	Last Name	Customer Credit Card Number
1	Al	Abe	1111-1111-1111-1111
2	Bob	Burns	2222-2222-2222-2222
3	Cam	Caan	3333-3333-3333-3333
4	Dan, D	Dune	4444-4444-4444-4444

(Harrington, 2009; Le, n.d)

Primary key = Customer ID

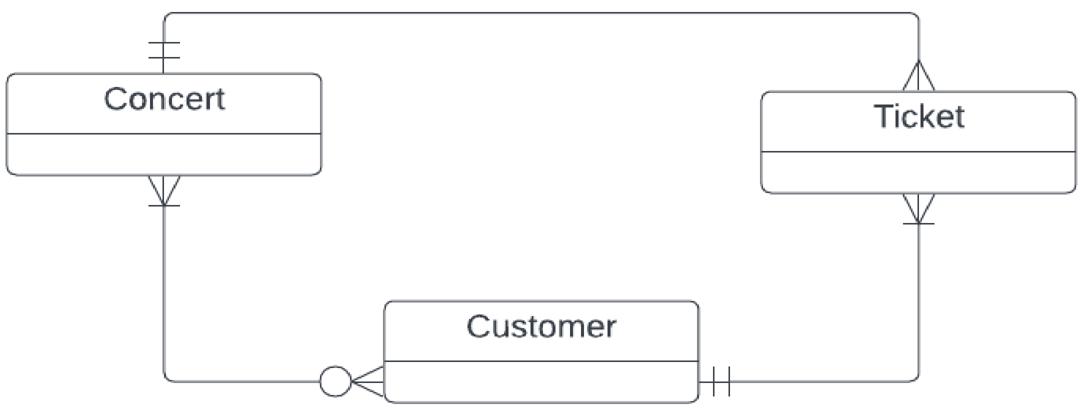
CUSTOMER-TICKET-QUANTITY-SALE

Customer ID	**Ticket ID**	Ticket Quantity Purchased	Sales Total
1	1	1	$50
2	2	1	$100
3	3	1	$500
3	4	1	$25
4	4	1	$25

(Harrington, 2009; Le, n.d)

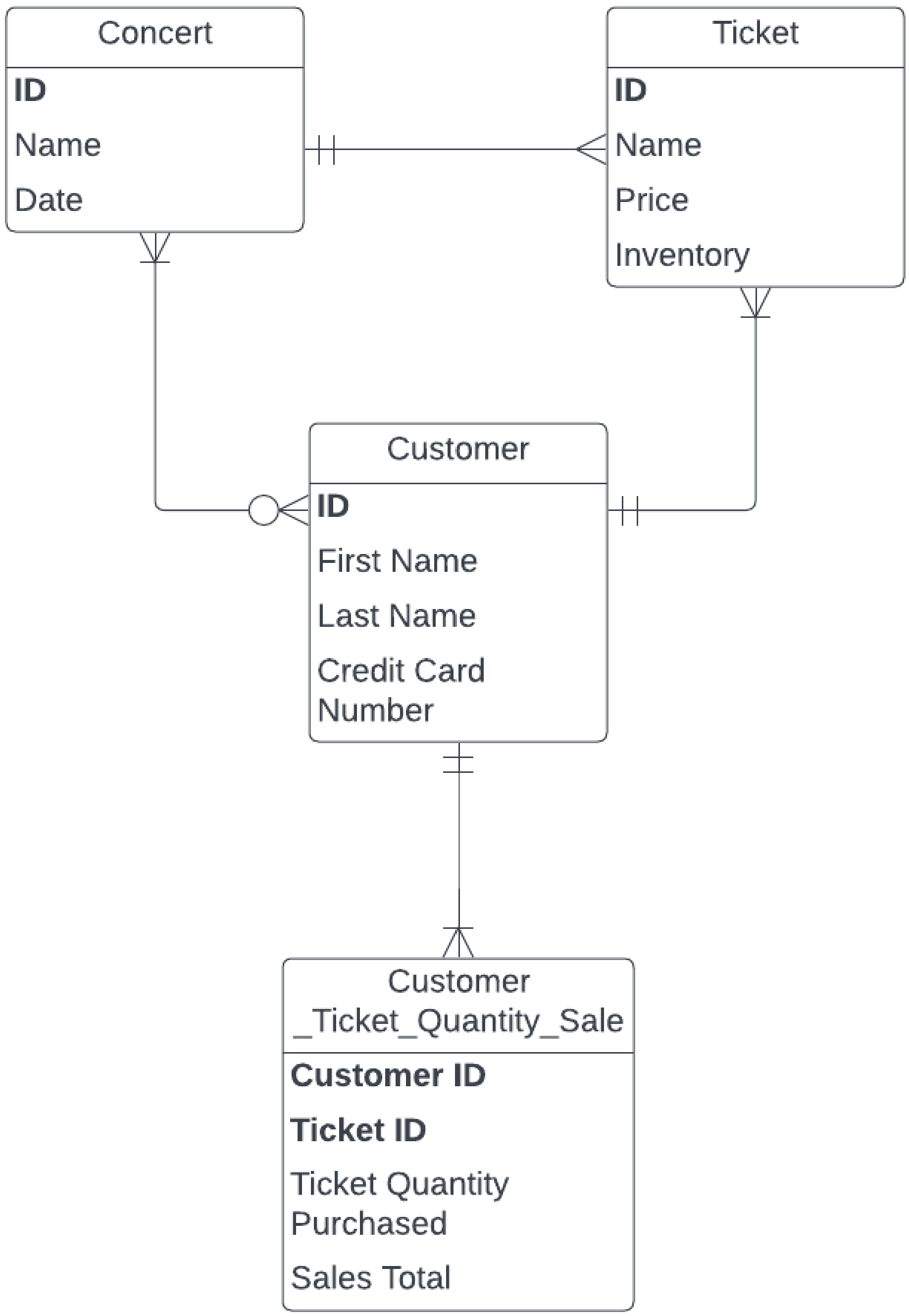
Primary key = Customer ID and Ticket ID

Conceptual ERD:



(LucidChart, 2017)

Logical ERD:



(LucidChart, 2017)

2. *Data file storage and access*

The ticket purchase system will use a relational database design, meaning data files will be stored via tables that each have a unique primary key (Harrington, 2009; Brush, 2019). Furthermore, the proposed database will utilize "sequential file organization" (Le, n.d.-b) for data file storage. Finally, files will be accessed from the relational database using a structured query language (SQL) (Brush, 2019).

C. System architecture

The ticket purchase system will use a multicloud "active-active-passive" (Elkhoja et al., n.d.) architecture to ensure "business continuity, continuous availability, and operational flexibility" (Elkhoja et al., n.d.). Furthermore, the system will utilize a web-based interface meaning a user can access the system from any device connected to the internet. Additionally, the client/server model will be utilized, meaning the computing-intensive processes of the system will be handled by the two active servers housed in the multicloud, and then that fully processed information will be delivered to the user/client machine (Intellipaat, 2022). Furthermore, as mentioned before, the middleware that is the multicloud provides increased data security and data redundancy (Intellipaat, 2022). Moreover, the multicloud will serve as the intranet that users will interface with when accessing the ticket purchase system from devices on the internet (Jive, n.d.). Lastly, the network will be configured to allow all non-malicious traffic to access the system.

Phase 3: Project Plan

A list of tasks or activities needed for implementing the proposed system

1. Choose a development methodology (which will be agile for this project) (Tilley & Rosenblatt, 2017).
    - Following agile principles, feedback from stakeholders in the form of "user stories" (Tilley & Rosenblatt, 2017) will be elicited and acted upon throughout the various iterations of each component of the system (Tilley & Rosenblatt, 2017).
2. Create all data stores (relational database)
    - Concert Information
    - Ticket Inventory
    - Sale History
3. Create all user interfaces
    - 2 input screens
        - Home screen
        - Purchase screen

- 5 output screens
    - Name, credit card information and order cost
    - Notice if ticket order is rejected
    - Record of transaction
    - Virtual ticket
    - Invoice

4. Create all system processes
    - Take customer order
    - Send name, credit card information and order cost to credit card company
    - Confirm or reject customer order
    - Distribute record of transaction
    - Deliver customer order
5. Choose whether to purchase hardware for one active section of the system architecture (on-premises). For the remaining sections of the system, choose a cloud vendor to host system services.

Estimating completion time and costs

To estimate the time required to implement the proposed system, we will employ the $\frac{B + 4P + W}{6}$ formula where "best-case estimate (B), a probable-case estimate (P), and a pessimistic, or worst-case estimate (W)" (Tilley & Rosenblatt, 2017). Furthermore, 'P' is a weight assigned to the probable case estimate which we will assign the standard value of 4 (Tilley & Rosenblatt, 2017). Finally, we will use this formula for each activity to produce individual completion time estimates and all the estimates will be added together to produce an estimated time of total project completion (Tilley & Rosenblatt, 2017).

For the task create a 'concert inventory' data store the best-case estimate will be 2 weeks, the probable-case estimate will be 3 weeks and the worst-case estimate will be 4 weeks. Plugging those figures into the above-referenced formula, the estimate required to implement the proposed system is 3 weeks (Tilley & Rosenblatt, 2017). For the task create a 'ticket inventory' data store the best-case estimate will be 2 weeks, the probable-case estimate will be 3 weeks and the worst-case estimate will be 4 weeks equaling an estimated time to completion of 3 weeks (Tilley & Rosenblatt, 2017). For the task create a 'sale history' data store the best-case estimate will be 2 weeks, the probable-case estimate will be 3 weeks and the worst-case estimate will be 4 weeks equaling an estimated time to completion of 3 weeks (Tilley & Rosenblatt, 2017).

For the task create a 'home screen' for UI input the best-case estimate will be 2 weeks, the probable-case estimate will be 4 weeks and the worst-case estimate will be 6 weeks equaling an estimated time to completion of 4 weeks (Tilley & Rosenblatt, 2017). For the task create a 'purchase screen' for UI input the best-case estimate will be 2 weeks, the probable-case estimate will be 4 weeks and the worst-case estimate will be 6 weeks, therefore according to the formula, the estimate required to implement the proposed system is 4 weeks (Tilley & Rosenblatt, 2017). For the task create a ‘credit card information and order cost’ output the best-case estimate will be 1 week, the probable-case estimate will be 2 weeks and the worst-case estimate will be 3 weeks equaling an estimated time to completion of 2 weeks (Tilley & Rosenblatt, 2017). For the task create a ‘notice if ticket order is rejected’ output the best-case estimate will be 1 week, the probable-case estimate will be 2 weeks and the worst-case estimate will be 3 weeks, equaling an estimated time to completion of 2 weeks (Tilley & Rosenblatt, 2017). For the task create a ‘record of transaction’ output the best-case estimate will be 1 week, the probable-case estimate will be 2 weeks and the worst-case estimate will be 3 weeks, equaling an estimated time to completion of 2 weeks (Tilley & Rosenblatt, 2017). For the task create a ‘virtual ticket’ output the best-case estimate will be 1 week, the probable-case estimate will be 2 weeks and the worst-case estimate will be 3 weeks, equaling an estimated time to completion of 2 weeks (Tilley & Rosenblatt, 2017). For the task create an ‘invoice’ output the best-case estimate will be 1 week, the probable-case estimate will be 2 weeks and the worst-case estimate will be 3 weeks, equaling an estimated time to completion of 2 weeks (Tilley & Rosenblatt, 2017).

For the task create the system process ‘take customer order’ the best-case estimate will be 5 weeks, the probable-case estimate will be 8 weeks and the worst-case estimate will be 11 weeks, equaling an estimated time to completion of 8 weeks (Tilley & Rosenblatt, 2017). For the task create the system process ‘send name, credit card information and order cost to credit card company’ the best-case estimate will be 5 weeks, the probable-case estimate will be 8 weeks and the worst-case estimate will be 11 weeks, equaling an estimated time to completion of 8 weeks (Tilley & Rosenblatt, 2017). For the task create the system process ‘confirm or reject customer order’ the best-case estimate will be 5 weeks, the probable-case estimate will be 8 weeks and the worst-case estimate will be 11 weeks, equaling an estimated time to completion of 8 weeks (Tilley & Rosenblatt, 2017). For the task create the system process ‘distribute record of transaction’ the best-case estimate will be 5 weeks, the probable-case estimate will be 8 weeks and the worst-case estimate will be 11 weeks, equaling an estimated time to completion of 8 weeks (Tilley & Rosenblatt, 2017). For the task create the system process ‘deliver customer order’ the best-case estimate will be 5 weeks, the probable-case estimate will be 8 weeks and the worst-case estimate will be 11 weeks, equaling an estimated time to completion of 8 weeks (Tilley & Rosenblatt, 2017). For the task ‘determine multicloud deployment architecture & vendor’ the best-case estimate will be 1 week, the probable-case estimate will be 2 weeks and the worst-case estimate will be 3 weeks equaling an estimated time to completion of 2 weeks (Tilley & Rosenblatt, 2017). For the task ‘activate multicloud services’ the best-case estimate will be 1 day, the probable-case estimate will be 1 day and the worst-case estimate will be 2 days equaling an estimated time to completion of 1 day (Tilley & Rosenblatt, 2017). Finally, after adding all of the tasks together the probable-case estimate of the time required to implement the proposed system will be 11 weeks.

As for costs, each development task will have two programmers assigned to it as specified by the extreme programming (XP) concept of “pair programming” (Tilley & Rosenblatt, 2017). Moreover, each task can be performed concurrently, apart from creating the data stores and the system processes. Furthermore, using an average hourly salary of $47 for a database Developer, $49 for a Full Stack Developer, $73 for a VP of Technology, and $38 for an IT project manager using 8-hour, 5-day work weeks, we can calculate the estimated workforce cost of this project (Salary.com, n.d.-a; Salary.com, n.d.-b; Salary.com, n.d.-c; Zippia, 2020)

To begin, the data stores are projected to take 3 weeks to develop, employing 6 Database Developers for each of the 3 separate data stores, which equals a projected cost of $33,840 (Salary.com, n.d.-a). Furthermore, developing the ‘home screen’ and ‘purchase screen’ tasks are estimated to take 2 months to develop, employing 2 Full Stack Developers per task, coming out to a projected cost of $62,720 (Salary.com, n.d.-b). Developing the remaining 5 UI tasks is projected to take 2 weeks to complete, employing 2 Full Stack Developers per task, coming out to a projected cost of $39,200 (Salary.com, n.d.-b). Next, creating all 5 system processes is projected to take 40 days to complete, and employing 2 Full Stack Developers per task will cost an estimated $156,800 (Salary.com, n.d.-b). Completing the task ‘determine multicloud deployment architecture & vender’ is projected to take 2 weeks, will be conducted by the VP of technology, and will cost an estimated $5,840 (Zippia, 2020). There will also be a dedicated IT project manager working on this project for the entire project life cycle (11 weeks) at a cost of $16,720 (Salary.com, n.d.-c). After adding all the costs of the tasks together, the estimated total workforce cost totals $315,120 (Salary.com, n.d.-a; Salary.com, n.d.-b; Salary.com, n.d.-c; Zippia, 2020). Lastly, the average annual cost of multicloud IaaS deployment cannot be reliably estimated at this point as vendor prices vary and the scale of the services required by this system also cannot be reliably estimated at this point in the project.

Conclusion:

A comprehensive system requirement analysis, led to a well-thought-out system and database design, whose implementation using the agile development methodology, was then planned out in the project plan section (the actual scheduling section of the project plan will be submitted separately via MS Project). By following the project plan laid out in the MS Project-enabled Gantt chart, the proposed ticket purchase system for concerts can be developed in an estimated 11 weeks at a projected cost of $315,120 plus the average annual cost of the chosen multicloud IaaS vendor (Salary.com, n.d.-a; Salary.com, n.d.-b; Salary.com, n.d.-c; Tilley & Rosenblatt, 2017; Zippia, 2020).

References:

Brush, K. (2019, November). *What is a RDBMS (Relational Database Management System)?* SearchDataManagement. https://www.techtarget.com/searchdatamanagement/definition/RDBMS-relational-database-management-system

Conger, S. (2008). *The New Software Engineering*. https://learn.umgc.edu/d2l/le/content/732250/viewContent/30023345/View

Defining the Problem in Project Initiation. (2014, August 9). W3computing. https://www.w3computing.com/systemsanalysis/defining-problem-project-initiation/

Elkhoja, H., Pullo, R., & Buratti, P. (n.d.). *https://www.ibm.com/cloud/architecture/architectures/resilience-active-active-passive/*. Www.ibm.com. Retrieved March 7, 2023, from https://www.ibm.com/cloud/architecture/architectures/resilience-active-active-passive/

English, M. (2021). Systems Analysis and Design: Creating the Systems that Support Strategic Plans. *Salem Press Encyclopedia*.

Goldsmith, R. (2009, March 17). *Defining requirements during software project feasibility analysis | TechTarget*. TechTarget.

https://www.techtarget.com/searchsoftwarequality/tip/Defining-requirements-during-software-project-feasibility-analysis

Harrington, J.L. (2009). Relational Database Design and Implementation : Clearly Explained: Vol. 3rd ed.Morgan Kaufmann. UMGC Digital Library: https://eds-b-ebscohost-com.ezproxy.umgc.edu/eds/ebookviewer/ebook/ZTAyNXhuYV9fMzE5NjQxX19BTg2?sid=b999e8e9-b7dc-456b-b3c0-2a1088048c49@pdc-v-sessmgr01&vid=14&format=EB&rid=1

Intellipaat. (2022, December 22). *What is Client Server Architecture? - Differences, Types, Example*. Intellipaat Blog. https://intellipaat.com/blog/what-is-client-server-architecture/#:~:text=Client%20server%20architecture%20is%20a

Jaiswal, S. (2022, September 27). *The 6 Types Of Information Systems And Their Applications*. Emeritus - Online Certificate Courses | Diploma Programs. https://emeritus.org/in/learn/the-6-types-of-information-systems-and-their-applications/

Jive. (n.d.). *Cloud-Based Intranet Software & Solutions*. Jive Software. Retrieved March 11, 2023, from https://www.jivesoftware.com/intranet/solutions/cloud-intranet/#:~:text=Cloud%2DBased%20Intranet-

Le, D. (n.d.-a). *Agile Modeling and Prototyping*. UMGC. Retrieved February 1, 2023, from https://learn.umgc.edu/d2l/le/content/732250/viewContent/30023347/View

Le, D. (n.d.). *An Example of Database Normalization*. UMGC. Retrieved March 9, 2023, from https://learn.umgc.edu/d2l/le/content/732250/viewContent/30023346/View

Le, D. (n.d.-b). *Designing Databases*. UMGC. Retrieved March 12, 2023, from https://learn.umgc.edu/d2l/le/content/732250/viewContent/30023350/View

Le, D. (n.d.-b). *Lecture Notes Project Management* [Notes]. UMGC. https://learn.umgc.edu/d2l/le/content/732250/viewContent/30023343/View

LucidChart. (2017). *ER diagram tutorial*. Lucidchart.com. https://www.lucidchart.com/pages/er-diagrams

NIH. (n.d.). *REQUIREMENTS ANALYSIS DOCUMENT (RAD)*. View.officeapps.live. Retrieved February 5, 2023, from https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fhr.nih.gov%2Fsites%2Fdefault%2Ffiles%2Fpublic%2Fdocuments%2Fhr-systems%2Fwits%2Fdocx%2Frequirementsanalysisdocumenttemplate.docx&wdOrigin=BROWSELINK

Salary.com. (n.d.-a). *Hourly wage for Database Developer*. Salary.com. Retrieved March 14, 2023, from https://www.salary.com/research/salary/listing/database-developer-hourly-wages

Salary.com. (n.d.-b). *Hourly wage for Full Stack Developer*. Salary.com. Retrieved March 15, 2023, from https://www.salary.com/research/salary/listing/full-stack-developer-hourly-wages#:~:text=The%20average%20hourly%20wage%20for

Salary.com. (n.d.-c). *Hourly wage for IT Project Manager I*. Salary.com. Retrieved March 16, 2023, from https://www.salary.com/research/salary/benchmark/it-project-manager-i-hourly-wages#:~:text=How%20much%20does%20an%20IT

SmartDraw. (2019). *Data Flow Diagrams - What is DFD? Data Flow Diagram Symbols and More - YouTube*. Www.youtube.com. https://www.youtube.com/watch?v=6VGTvgaJllM

Ticketmaster. (n.d.). *Music Concert Tickets & Tour Dates | Buy Ticketmaster Verified*. Ticketmaster. Retrieved March 8, 2023, from https://www.ticketmaster.com/discover/concerts

Tilley, S. R., & Rosenblatt, H. J. (2017). *Systems analysis and design*. https://learn.umgc.edu/d2l/le/content/732250/viewContent/30023356/View

Visual Paradigm. (n.d.). *Visual Paradigm Online*. Online.visual-Paradigm.com. https://online.visual-paradigm.com/drive/#diagramlist:proj=0&diagram=list

Zippia. (2020, May 18). *Vice President, Technology Salary (March 2023) - Zippia*. Www.zippia.com. https://www.zippia.com/vice-president-technology-jobs/salary/