Automated Movie Theater Usher

By: Gage Bays, Jackson Hyche, Raiden Greene, and Whit Brewer

System Analysis and Design

Dr. Davis

12/9/2024

Table of Contents

I. Problem Statement and Feasibility study……3

II. Brief Review of Requirements Analysis and Specification……5

III. Preliminary Design……15

IV. Detailed Design……21

V. Project Management……..26

VI. Mockup of Prototype……28

I. Problem Statement and Feasibility Study

Problem Statement:

Having a single person as the Usher for a movie theater is not helpful, as this person will be doing nothing most of the time and could be behind the counter instead. Since there is no real security, people can scam their way into other movies or a cheaper movie ticket. The Automated Usher System would allow for non-manned security and ticket checking, allowing for more safety on scamming and allowing for more personnel at the ticket counter.

Feasibility Study:

Economic Feasibility:

The AUS, Automated Usher System, will not be very cost consuming, but it will not be cheap. The purchase of the AUS will cost approximately $25,000 USD. Although this amount of money is not small, this is an amount of money easily made back through time. The system will be available in all US states and Canada, so shipping cost would also not be expensive.

Technical Feasibility:

Implementing the AUS for AMC Theaters is technically feasible and would be very difficult in action. The database for tickets is simple to make, and can be created in any software applicable. The database previously mentioned will hold all ticket attributes and customer attributes and compute the ticket sales for each day and allow for the printing of these pages for physical documentation. Once a scan has been made on the AUS, the ticket database will be used as a validation for the ticket scanned.

Operational Feasibility:

The operational manpower for the theater will increase after the implementation of the turnstile system. The system will allow for more people at the ticket selling counter or walking around cleaning and restocking, meaning there is less stress on other workers at the end of the shift.

Legal Feasibility:

There are no legal issues with setting up the AUS system in a movie theater. The door on the side of the AUS allows for exit for all persons and entry for people who are not able to go through the turnstile. The only possible argument for a legal issue is in the case of an emergency, however there are already emergency/fire exits in the back of all movie theaters and in the electrical room.

Time Feasibility:

The only part of the implementation of the system that would be time consuming is the setup and placement. Once the turnstile is inside the theater and the turnstile is fully set up, there will be no more time consuming parts of the implementation. The database creation could be argued as a problem, however, it will not be as difficult as assumed, as the database will be set up before the physical system.

II. Brief Review of Requirements Analysis and Specification

Requirements:

Functional system requirements for the turnstyle for movie theaters:

- The turnstyle will be one way and clockwise turning(also known as left handedness for full height turnstiles)
- The QR scanner will be on the right side of the turnstile located 4’ from the ground
- The turnstile will be at minimum 5’5” in length, 5’ in width, and 7’5” in height
- To the left of the turnstyle will be two doors that allow for push on one side and handle on the other.
- The the center bar of the doors will have the ability to be removed
- The handle side of the door will have an electronic lock that can be unlocked with an employee QR code
- All employees will have an QR code that works for both the door and turnstile
- The QR scanner will be able to handle both paper and digital QR codes
- The QR scanner will connect to the database for logging ticket in times
- It should not take a user more than 6 seconds to move through the turnstile
- The turnstile cannot operate without electricity
- The scanner will deny entry for tickets until 45 before the movie starts
- The scanner will deny entry for tickets 10 minutes before movie ends

The operational system requirements for the turnstile subsystem are:

- The overall subsystem should not cost more than $25,000 for initial implementation.
- The overall subsystem should not cost more than $4,500 per year to operate.

The subsystem maintenance and support requirements for the turnstile subsystem are as follows:

- There will be glass on the panel where the QR scanner is located
- The turnstile will be made out of 304 stainless steel
- The front scanner will need cleaning every week to ensure a clean read
- The ground will be a smooth tile leading into carpet underneath the turnstile
- The turnstyle itself will only require maintenance once every few months, however it will need cleaning every week for the sake of its appearance
- If it needs to be fixed, the parts are shippable and easy to locate
- The turnstile can only break via the turning system jamming or stopping, in which case the gate will be open for entry and exit
- The way to fix that issue would be to unstick the turnstile in any way, slightly pull back, and continue pushing forward

Analysis:

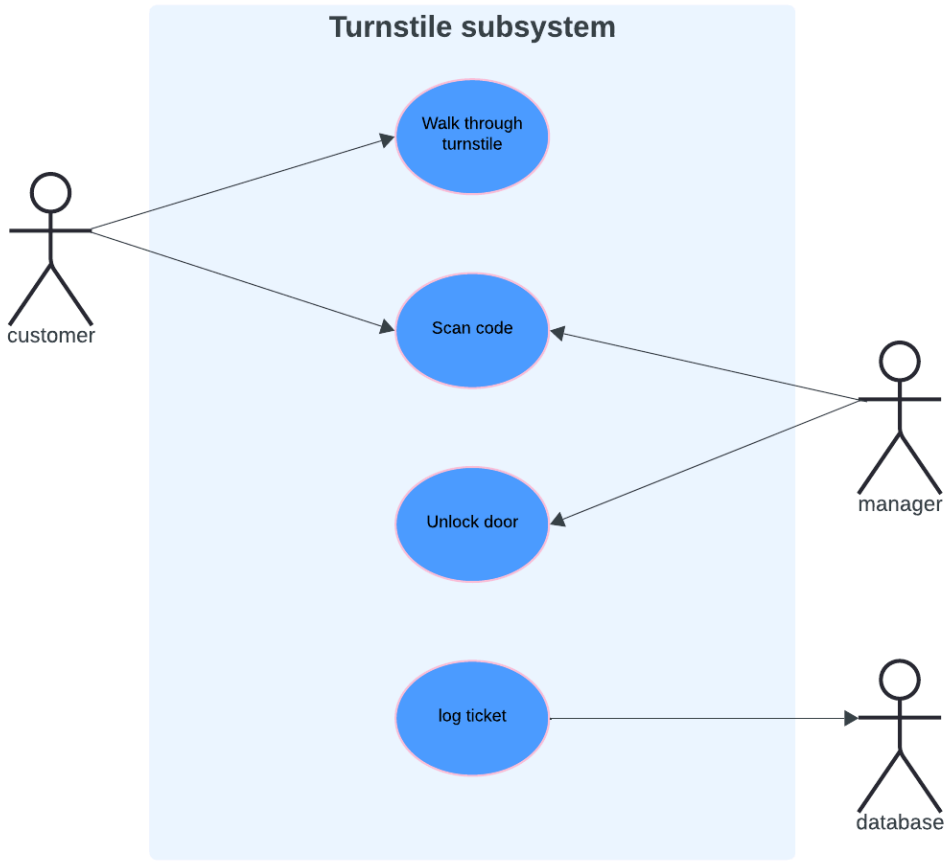
1. Identification of Use Cases:

- Scan QR code = the user scan either the paper ticket with a QR code or the digital ticket with a QR code
- Walk through turnstile = user walks through to the other side
- Unlock door = an employee unlocks the door using the QR scanner to get to the other side
- Log ticket = when a ticket is scanned, the database logs the info of the ticket
- Clean scanner = the QR code scanner is cleaned to ensure quality scans

2. Determine the actors:
   - Managers
   - Movie theater employees
   - kids
   - Adults
   - Mechanics
   - Janitors
   - Emergency services
   - Movie theater database

Use Cases Diagram:

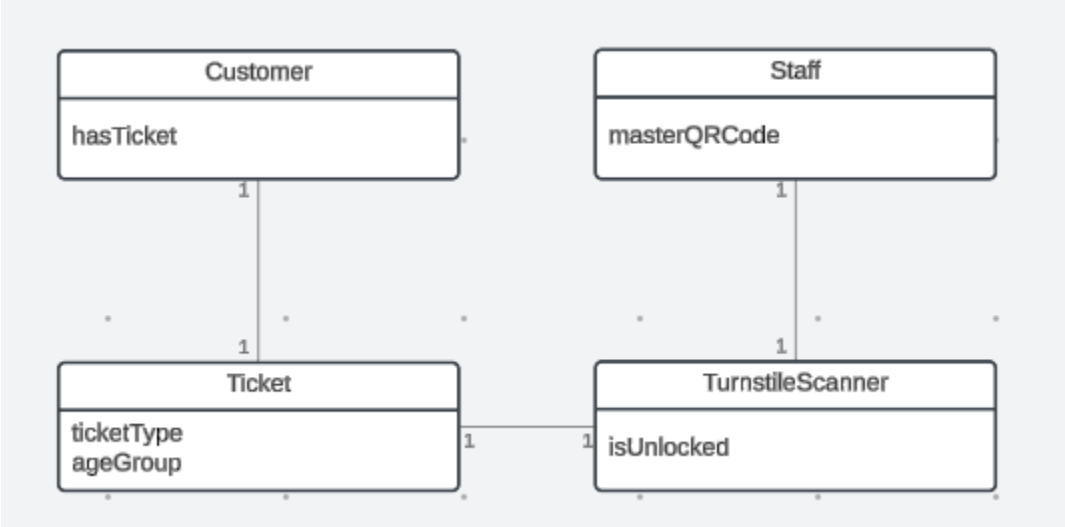
The actors are the objects and people that interact with the diagram and the use cases are simple verb noun statements of how the system works. The use case diagram is to show what actors will have certain use cases. All use cases deal with the main part of the system, the turnstile.



Domain model class diagram:

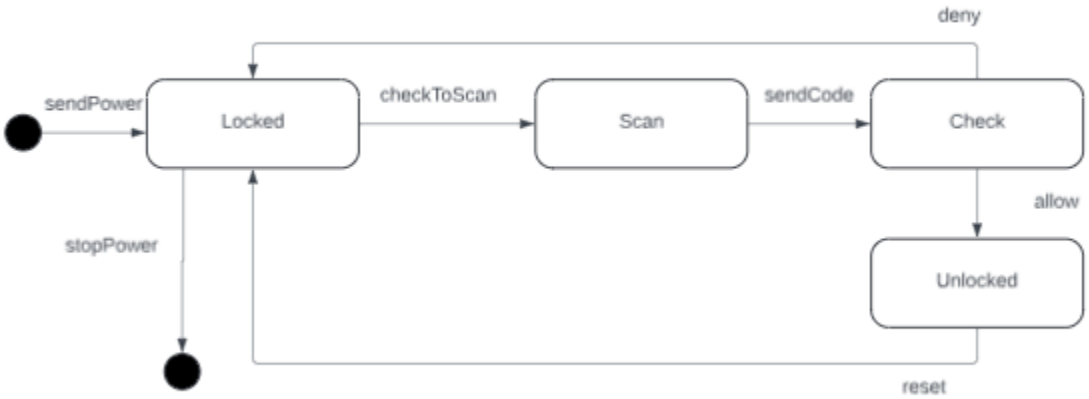
This diagram shows the components and relationships within the turnstile system at a movie theater. It shows how customers, tickets, staff, and the turnstile interact with one another. The Customer must have a ticket with specific details, like ticketType and ageGroup to pass through the system. The ticket is scanned by the TurnstileScanner and determines if the ticket is valid. If it is valid, the scanner unlocks the turnstile, which is represented by the isUnlocked attribute. This will allow the customer to enter.

The system accounts for management by the Staff for the theater. The Staff class includes the masterQRCode attribute that allows staff members to override the turnstile, unlocking it manually if it is necessary to do so. The relationships between the components ensure that every customer has exactly 1 ticket, and each ticket interacts with a singular turnstile scanner.



Simple State Machine Diagram:

The state machine diagram shows a few of the different states the system can be in while in use. This is the initial diagram to show the states of the system. The system's main states are being locked or unlocked. The system will start in the locked state when turned on and after it is scanned a ticket, the system will go into the check state where the ticket information is being validated. It will then either be unlocked or be locked.



Use Case Description:

The use case description describes all the elements of a use case. This is just one use case description, but the format can be used for all use cases associated with the system.

<table>
<tr><td>Use Case Name:</td><td colspan="2">Use the QR scanner.</td></tr>
<tr><td>Scenario:</td><td colspan="2">Use the QR Scanner to activate the turnstile.</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">Person wants to use the turnstile to access part of the theater.</td></tr>
<tr><td>Brief Description:</td><td colspan="2">A person must use a QR code at the QR scanner to activate the turnstile and access the inner part of the theater.</td></tr>
<tr><td>Actors:</td><td colspan="2">Customers and Staff.</td></tr>
<tr><td>Related Use Cases:</td><td colspan="2">Invoked by the employee unlock door use case.</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Customers and Staff.</td></tr>
<tr><td>Preconditions:</td><td colspan="2">Online tickets or printer must be available to make code. Scanner must be operational.</td></tr>
<tr><td>Postconditions:</td><td colspan="2">Tickets must be logged into the database by scanner. Scanner must activate the turnstile.</td></tr>
<tr><td rowspan="2">Flow of Activities:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Customer/Staff scan QR Code

2. Customer walks through turnstile</td><td>1.1 System checks if code is a staff code or customer code

1.2 System checks if customer movie starts within 45 minutes

1.3 System logs QR Code usage to database

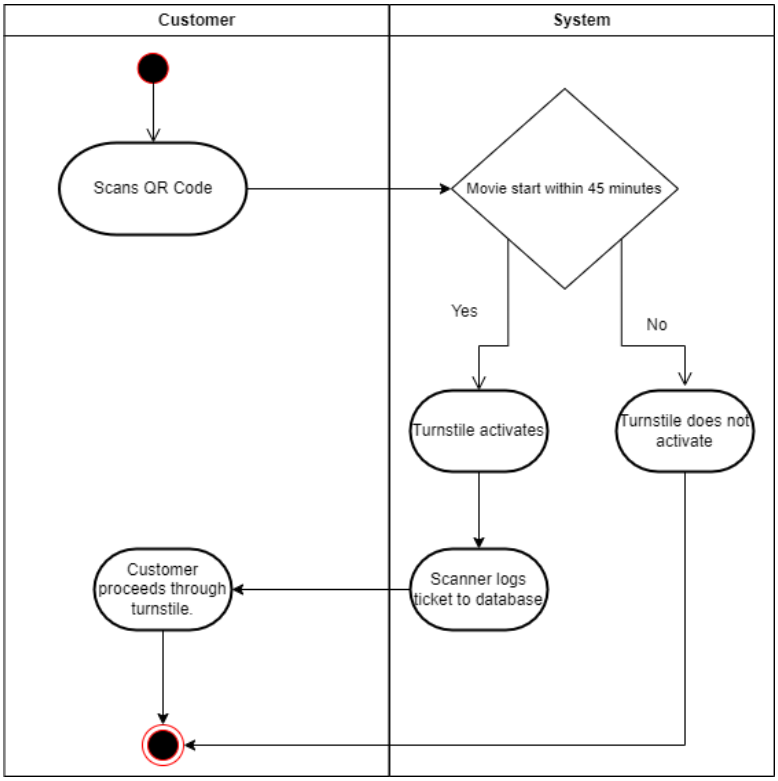
1.4 System unlocks turnstile

2.1 System locks turnstile</td></tr>
<tr><td>Exception Conditions:</td><td colspan="2">1.1 If QR Code is neither staff or customer.

1.2 If the customer's movie does not start within 45 minutes.</td></tr>
</table>

Activity Diagram:

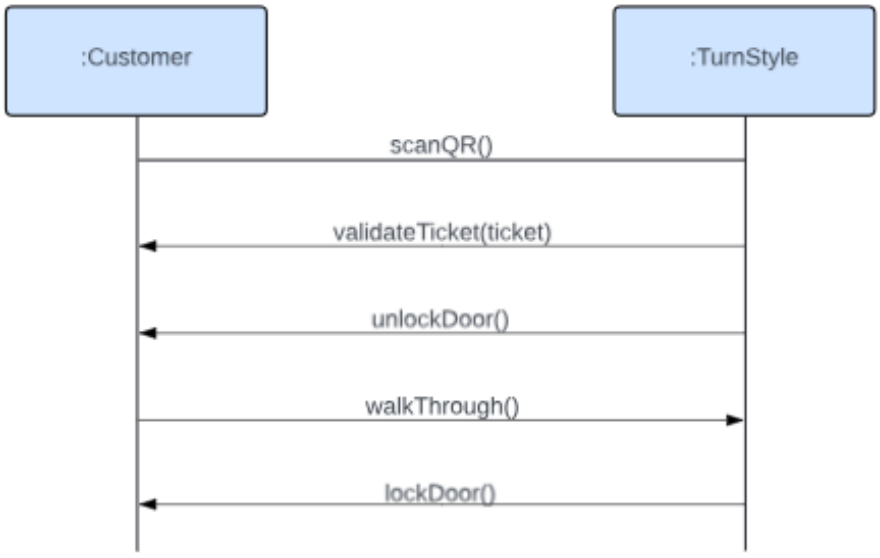
The activity diagram shows the step by step process of how a customer uses the turnstile system to access the theater. The diagram shows the interaction between the Customer and the System through various decisions and activities. The process begins with the Customer scanning their QR code at the turnstile. The System evaluates whether the movie start time is within 45 minutes. If the movie is starting within that time, the turnstile will activate and allow the customer to go through the system. At the same time, the system logs the ticket into a database to track the entries. If the movie is not starting within 45 minutes, the turnstile will not activate, and the customer will be denied entry. This diagram shows the flow and rules of the turnstile system and ensures that only valid ticket holders that are close to their movie time can enter. It also shows that the system records entries for activity for better security and management.



System Sequence Diagram:

The system sequence diagram shows the interaction between a Customer and the Turnstile when using the system to gain entry into the theater. The diagram represents the step-by-step process of how the system and the customer communicate for validation of a ticket and allow access. The sequence starts with the customer scanning their QR code at the turnstile which is represented by scanQR(). The system then validates the ticket, represented by validTicket(ticket), to ensure the ticket is legitimate and meets the necessary criteria like the correct movie and time.

If the ticket is valid, the turnstile will unlock, represented by unlockDoor(), and will allow the customer to walk through the turnstile, represented by walkThrough(). When the customer passes, the turnstile will lock again, represented by lockDoor(), to prepare the next person. This process ensures that authorized customers with valid tickets are the only customers who can enter.



CRUD Matrix:

The CRUD matrix explains the roles and responsibilities of different components within the turnstile system for various cases. CRUD stands for Create, Read, Update, and Delete, which describes how data is handled throughout the system processes. The matrix shows how components like Ticket, QR Scanner, Turnstile, and Customer interact in actions like scanning a QR code, walking through the Turnstile, unlocking the door, logging tickets, and purchasing a ticket. Each entry in this table specifies if a component is responsible for creating, reading, or updating data in a specific use case.

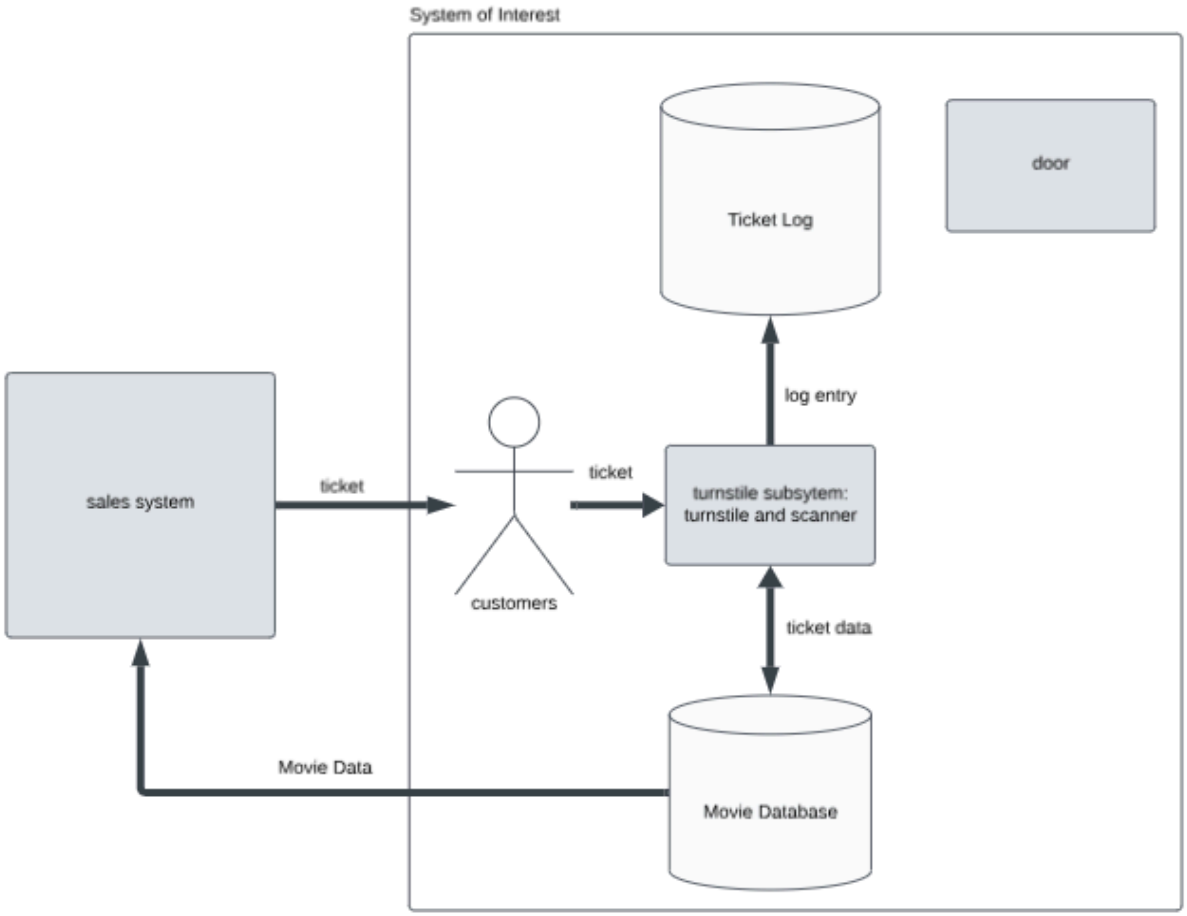
For example, in the Customer Scans QR code use case, the Customer does the action while the QR Scanner reads the data for the QR code. In the Walking through Turnstile use case, the Customer moves through the turnstile, and the Turnstile is updated to reflect the action. During ticket logging, the Turnstile is updated, while the Ticket data is read and recorded.

USE CASES	Customer	Turnstyle	QR Scanner	Ticket
Customer Scans QR Code	R		R	R
Walking through turnstyle	C	U		
Unlocking the door		C	U	R
Logging the tickets	R	U		R
Customer purchases a ticket	C			C

III. Preliminary Design

System of Interest:

The System of Interest for this project is a turnstile system at a movie theater, designed to manage entry and enhance operational efficiency. The System of Interest consists of three subsystems which include the Access Control Subsystem, the Data Management Subsystem, and the User Interaction Subsystem. The Access Control Subsystem verifies ticket validity and triggers the turnstile to allow entry based on integration with the theater's central ticketing system. This involves the use of a QR scanner upon entry into the turnstile. The Data Management Subsystem serves as the database for the turnstile, storing records of each entry, timestamp, and ticket type, and integrates with the theater’s main database to validate tickets in real time. The User Interaction Subsystem enables customers to scan mobile or physical tickets and allows staff to remotely control the turnstile for special cases, like maintenance or emergencies, through an authorized application interface. Together, these subsystems ensure secure and efficient access for theater customers.

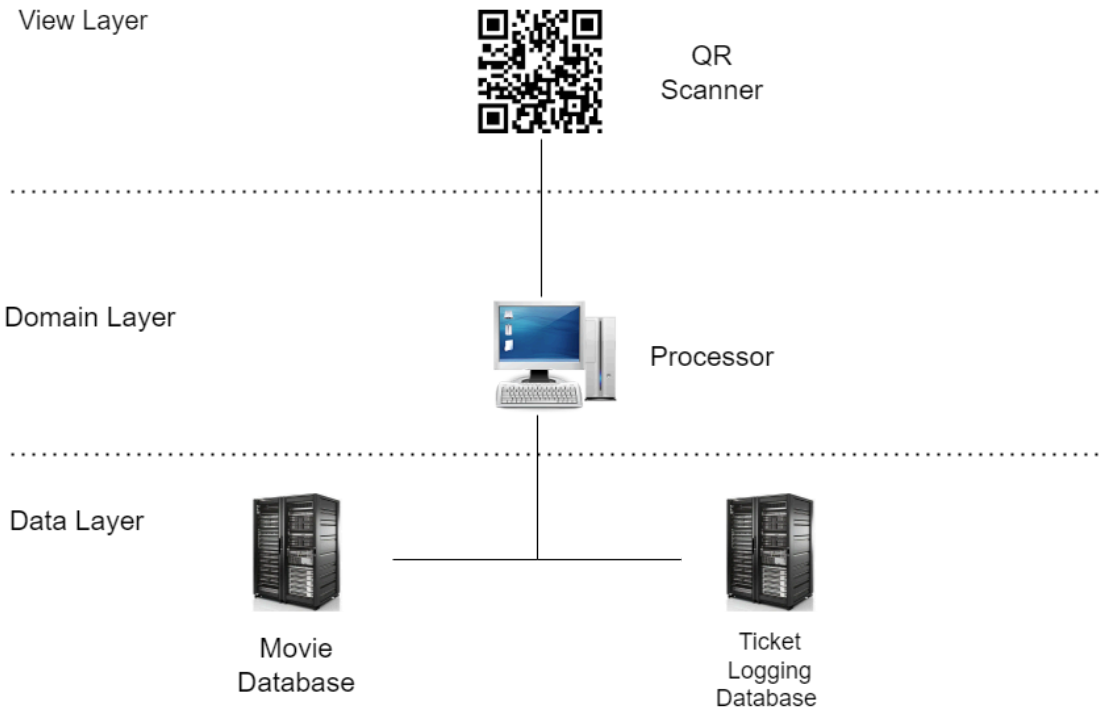

System of Interest
Ticket Log
door
log entry
sales system
ticket
ticket
turnstile subsytem:
turnstile and scanner
customers
ticket data
Movie Data
Movie Database

Describing the Environment:

In the environment of the system, a customer would go about purchasing a ticket through two means, the online app and website or the in-person ticket sales person. The ticket will be printed with a QR code on it, as well as the other ticket attributes produced normally. After purchasing a ticket(s), the customer can either scan the ticket on the QR scanner in the turnstile or end up not using the ticket. If the customer scans the ticket on the QR scanner, the scanner will send ticket data to the movie database for the count of ticket use and active persons in a theater. The scan will also log the entry for that specific ticket(s) and not allow that ticket to be scanned in anymore, as only one person can use a ticket. Next, the customer can either walk through the turnstile and push through the circle or pull it, in which case nothing would happen as turnstiles only move counter-clockwise. Once the customer has scanned the ticket(s) and walked through, the database will remove that ticket from itself, as it has been used and doesn't exist anymore, and the turnstile and QR scanner will reset themselves to the waiting stage for the next scan. The door on the side of the turnstile will act as an exit point for the movie theaters and also be used as an entrance for handicapped patrons or another entrance if needed. The inside of the door will have a simple push-to-open piece attached to it so anybody is able to open the door. On the outside, there will be a small validation scanner on the door for employee ids so the employees don't have to go through the turnstile every time they wish to get to the back. Since the employees also need to have a location for trash boats to enter and exit, this door will also serve as that as well. The very last thing that happens in the system's environment is the movie database logging another purchased ticket for a certain movie and logging it into the system.

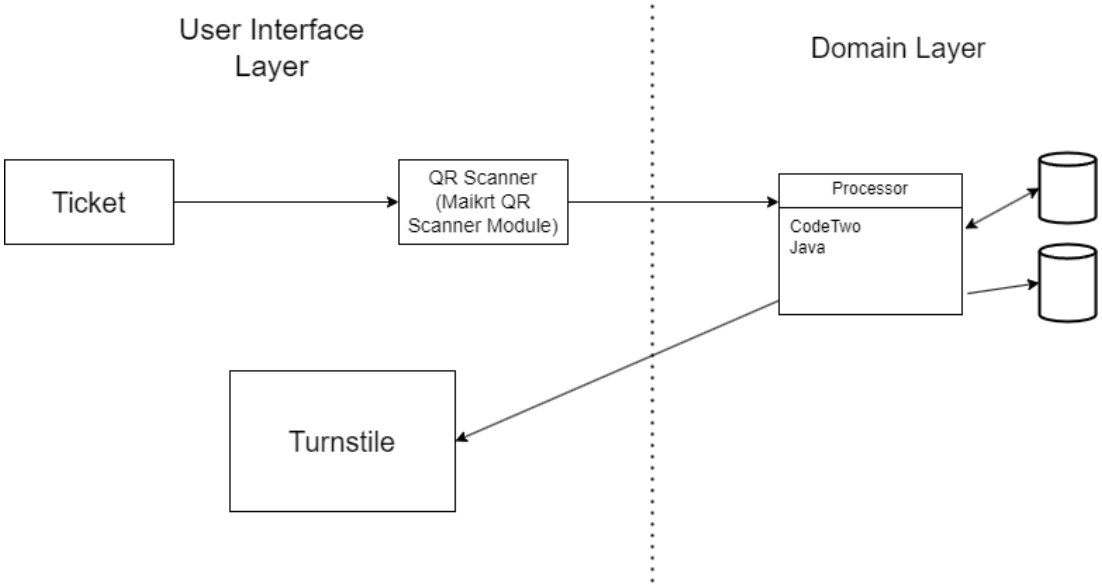
Deployment Diagram:

The deployment diagram shows how the system connects the hardware aspects of the design with the software aspects. When a ticket is scanned with the turnstiles QR scanner at the view layer, it sends the tickets information to the theaters processor at the domain layer. That processor then sends the ticket information to the ticket logging database and retrieves the relevant information from the movie database to check the tickets validity at the data layer. Once the ticket's validity is checked, the processor will send a signal back to the turnstile, telling it to either activate or stay dormant.



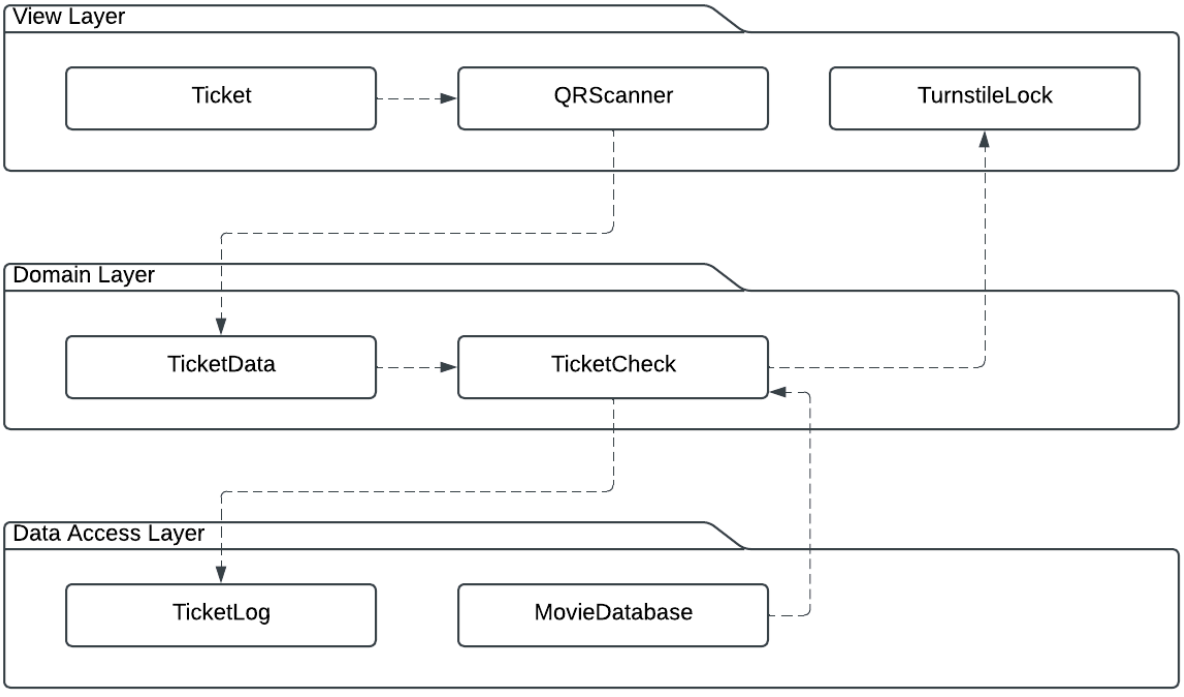
Component Diagram:

The component diagram, as the name suggests, displays the relationship between the systems' different components in more detail. The customer's ticket will be scanned by the turnstiles QR scanner, a Maikrt QR Scanner Module, in the user interface layer. The data from the ticket will be sent to the theaters processor that will be based in CodeTwo and Java in the domain layer. The processor will communicate with both the ticket logging database and the movie database (the modules to the right of the processor) to log and retrieve the relevant information. The processor will then send a signal back to the turnstile in the user interface layer, telling it to activate or not to activate.



Package Diagram:

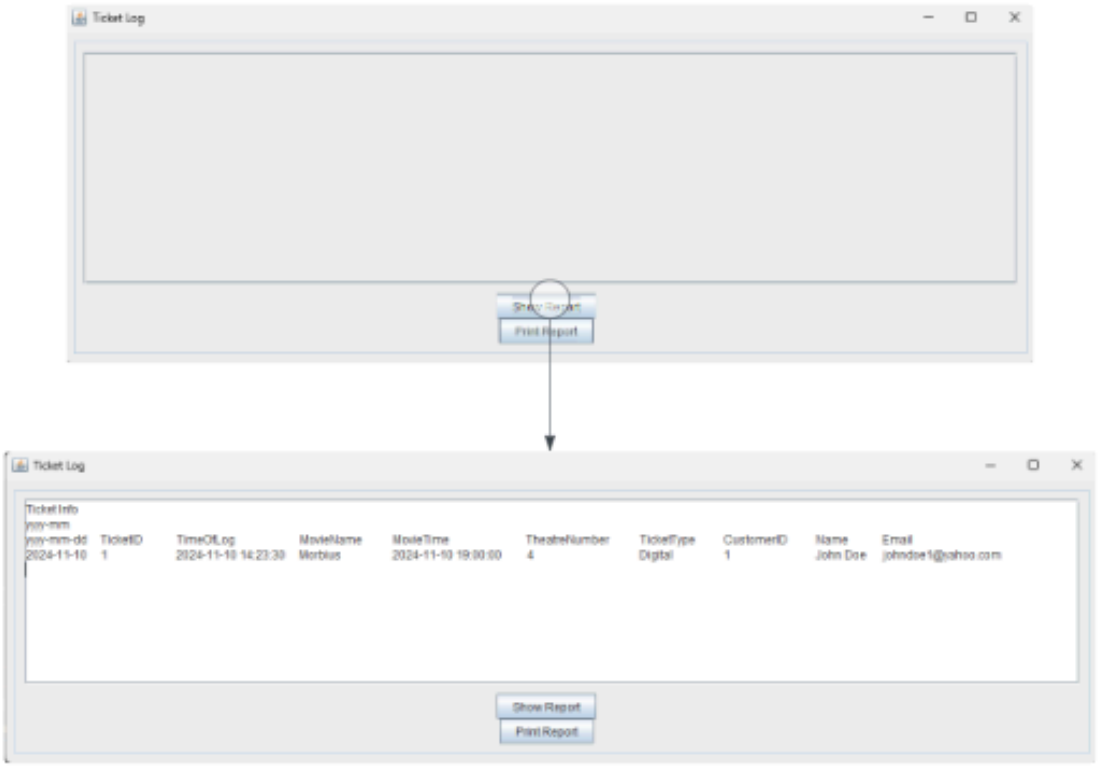
The package diagram displays the organization structure of the system's elements. In the system, the customer's ticket, the QR scanner, and the turnstile lock is in the view layer. The ticket data and the ticket check process is in the domain layer. The ticket log and movie database are in the data access layer. The ticket is scanned by the QR scanner, which sends the data to the domain layer to be checked. The data is sent to the ticket logging database and the data from the movie database is retrieved from the data access layer to validate the ticket information. The validation signal is then sent back to the turnstile lock in the view layer.



IV. Detailed Design

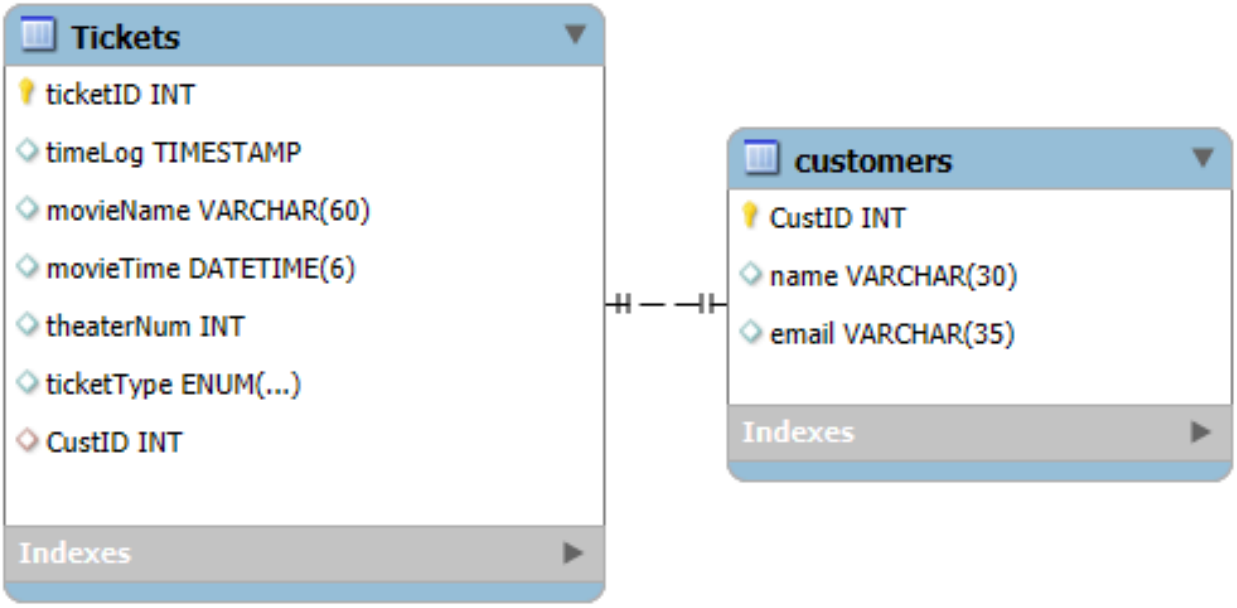
UI Design:

The UI design for this system would be for the logging database. The database info will be put in a simple format with all of the ticket info being listed. The user will be able to show reports based on what the database has. The UI does not need to be complicated and is kept simple. The User will be able to print the reports for record keeping reasons.



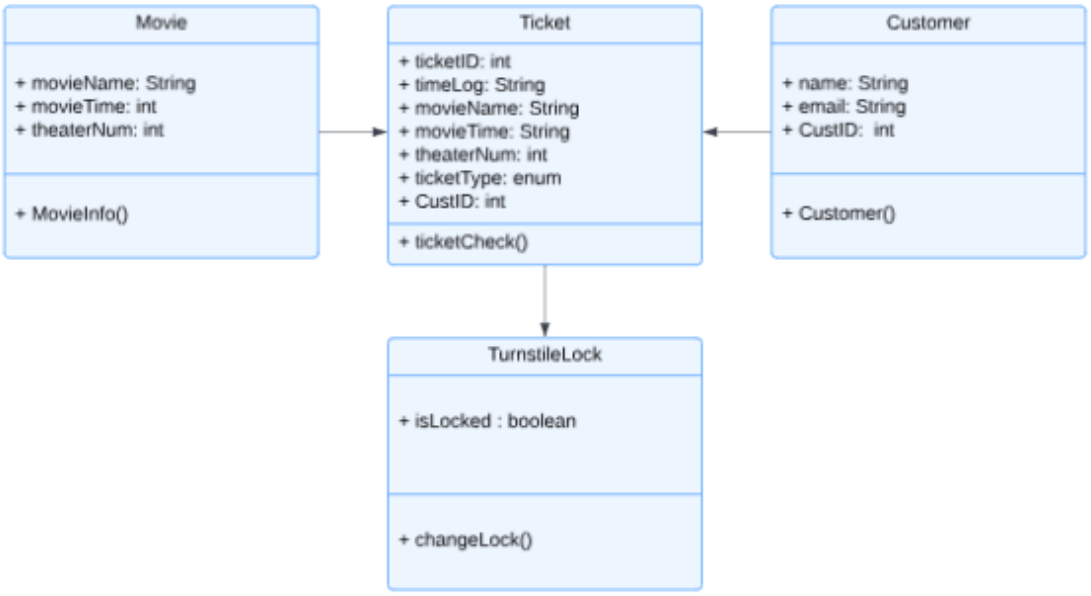
Database design:

The database is for logging the tickets that are scanned at the turnstile. The scanner will send information that a ticket has. The database will hold information from the ticket and if available, customer name and email. The database is there to help keep records of ticket scans and when customers are entering theaters that the sales and marketing might find useful. This diagram is the Entity Relationship Diagram, making it easier to understand the design and structure of the database. It shows the two entities, tickets and customers.



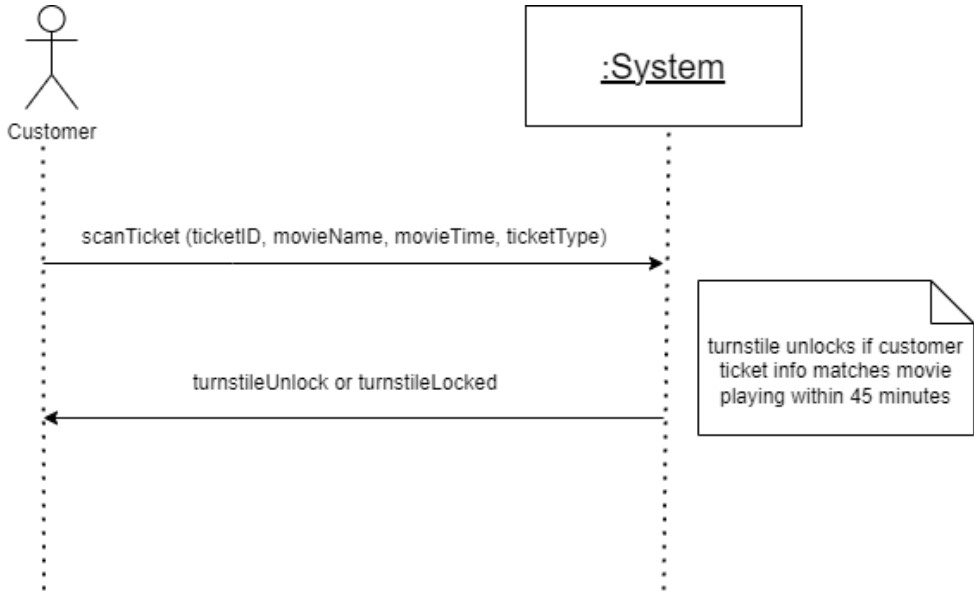
Design Class Diagram:

The design class diagram provides a detailed blueprint for the implementation of the turnstile system. The diagram outlines key components of the system such as Movie, Ticket, Customer, and TurnstileLock. Along with these are their attributes, methods, and relationships. Each class represents a certain part of the system, which shows its functionality and data requirements. For instance, the Movie class contains attributes like movieName, movieTime, and theaterNum, as well as the MovieInfo() method to obtain the movie details. The Ticket class is similar as it links a customer to a movie through its attributes which include ticketID, movieName, movieTime, and CustID with the ticketCheck() method used to validate the ticket information. The Customer class manages user data like name, email, and CustID, which links customers to their tickets. The TurnstileLock class shows the physical control of the turnstile, using the isLocked attribute to track the state and the changeLock() method to go between locked and unlocked.



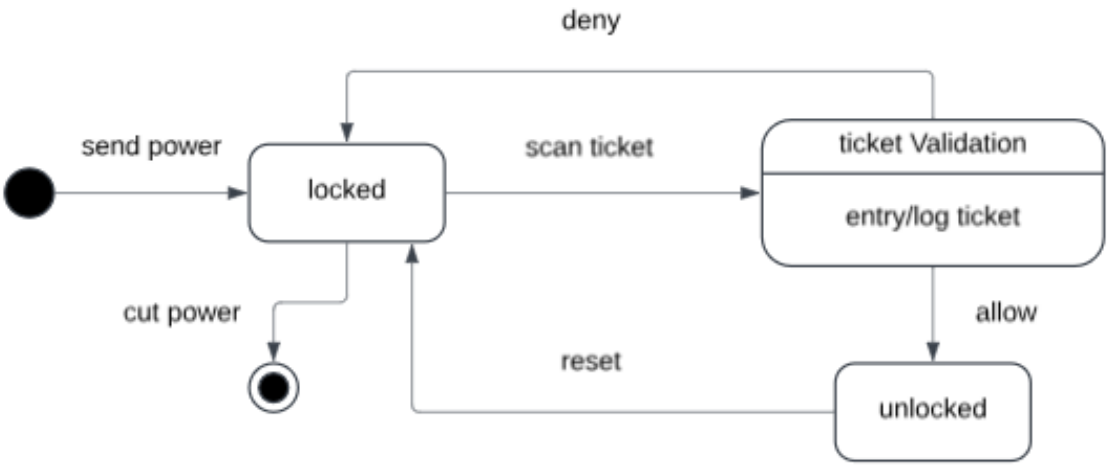
System Sequence Diagram:

The system sequence diagram below illustrates the relationship between the customer and the system, after the database has been created and implemented into the system. The first bar shows that when a customer scans a ticket, those four attributes will be passed into the database system. Those attributes are existing in each ticket object that is scanned and will all have a different movie ID, some will have the same movie name, movie time, and ticket type. Once the scan has occurred and that function has been called, the database will check if the ticket is valid or not through a sequence of simple check steps. Depending on the validation check, the system could either return turnstileUnlock or turnstileLock, which do as expected, unlock the turnstile if the ticket is valid and lock it if the ticket is not. There is also a parameter that must be met before the system will allow the turnstile to be unlocked, which is checking if it is less than 45 minutes before the movie. If it is more than 45 minutes before the movie's allocated start time, the turnstile will not unlock.



State Machine Diagram:

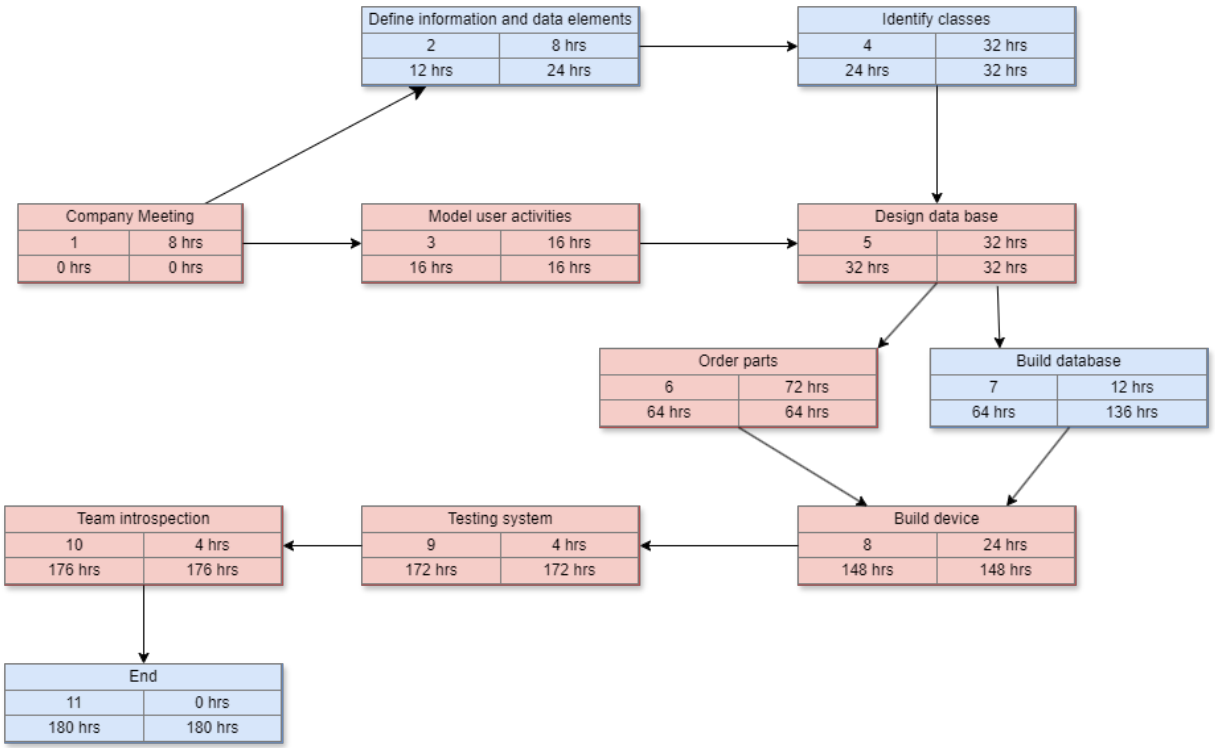
This is a more complete version of the simple state machine diagram. The system has three states: locked, ticket validation, and unlocked. The system is normally in the locked state, then when a ticket is scanned, it moves to a ticket validation state where it is waiting on the validation of the ticket. Along with that, when the ticket is scanned and upon entry to the ticket validation state, the ticket information is logged. After the ticket is validated, the ticket is either denied or allowed. If it is allowed the turnstile will become unlocked; if denied, then locked.



V. Project Management

PERT/CPM Chart:

The PERT/CPM chart visualizes the entire process of the system from conceptualization to its final implementation. In each module, there are four different values. The value in the top left represents the task i.d., which starts at one and continues upwards till the end. The value in the top right represents how long the task should take to complete. The value in the bottom left represents the earliest time a task can begin and the value in the bottom right represents the latest time a task can begin. The modules in blue represent tasks that are “not on the critical path”, meaning they are tasks that are not required to be completed before another task can proceed. The modules in red represent tasks that are “on the critical path”. According to this PERT/CPM chart, this system should take 180 hours to complete with eleven total tasks.



Net Present Value Chart:

The Net Present Value chart allows for visual representation for allocated money for the system and the total time until the system is fully paid back for. Year 0 represents the time that the system is purchased and implemented. After the first year, approximately 42% of the original spending has been made back. The money that is counting for the net benefit is the amount, on average, that an usher would be paid yearly at a movie theater, assuming they work 5 days a week. The discount factor is a number used in the multiplication for the amount of money added to the Cumulative NPV to determine an estimation on how much will be made, accounting for political and economic inflation and changeability. After the third year, a guarantee of profit is achieved. From that point on, there will only be an increase in the amount of money that is gained. The Year 5 Total describes the amount of the 3 most important numbers, the total cost of deployment, the net benefit, and the net present value. The net benefit is the amount of money saved before the application of the discount factor and the net present value is the total after said application. The final payback period will be 2.498 years, or 2 years and 182 days. The final return on investment with the discount is 87% and the return on investment without the discount is 122%.

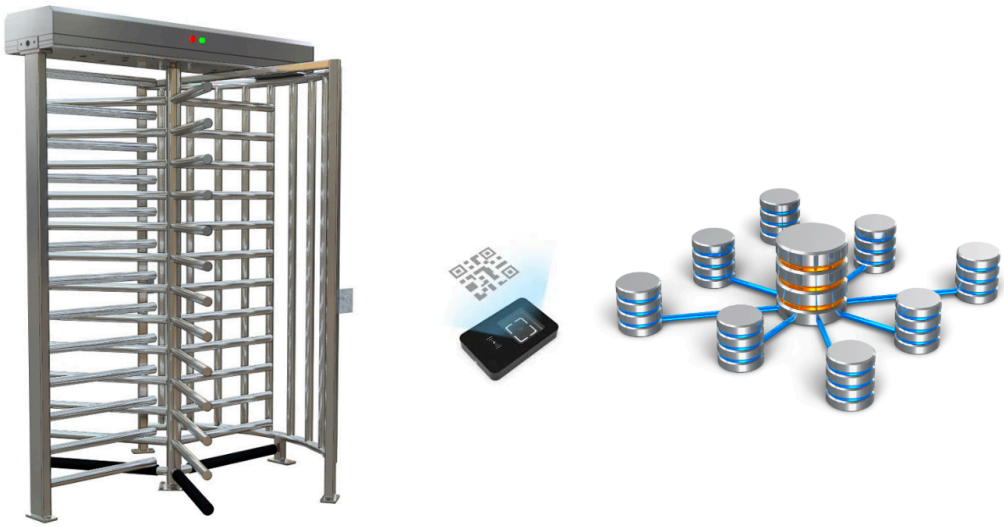
RMO Cost Benefit Analysis

	Category	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5	Year 5 Total
1	Value of benefits		$15,600	$15,600	$15,600	$15,600	$15,600	
2	Development costs	-$25,000						-$25,000
3	Annual expenses		-$4,500	-$4,500	-$4,500	-$4,500	-$4,500	
4	Net benefit/costs	-$25,000	$11,100	$11,100	$11,100	$11,100	$11,100	$30,500
5	Discount factor	1.00000	0.9434	0.8900	0.8396	0.7921	0.7473	
6	Net present value	-$25,000	$10,471.74	$9,879	$9,319.56	$8,792.31	$8,295.03	$21,757.64
7	Cumulative NPV	-$25,000	-$14,528.26	-$4,649.26	$4,673.70	$13,466.01	$21,761.04	
8	Payback period	2 years +	4649.26 / (4649.26 + 4673.70) = .498			or 2 years + 182 days (.498 x 365)		
	ROI with discount =	21,757.64 / 25,000 = 87%						
	ROI without discount =	30,500 / 25,000 = 122%						

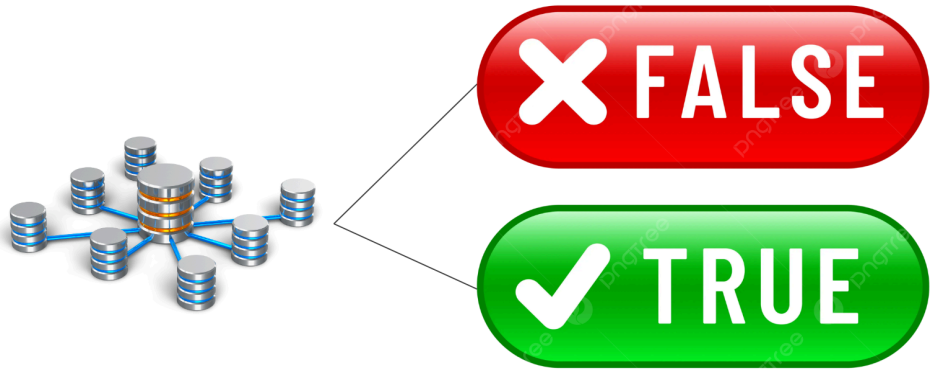
VI. Mockup of Prototype

This prototype is a showing of the system through images as it would be difficult to create a fully functional turnstile. To start this off, the actor will approach the turnstile with a ticket and scan the ticket.

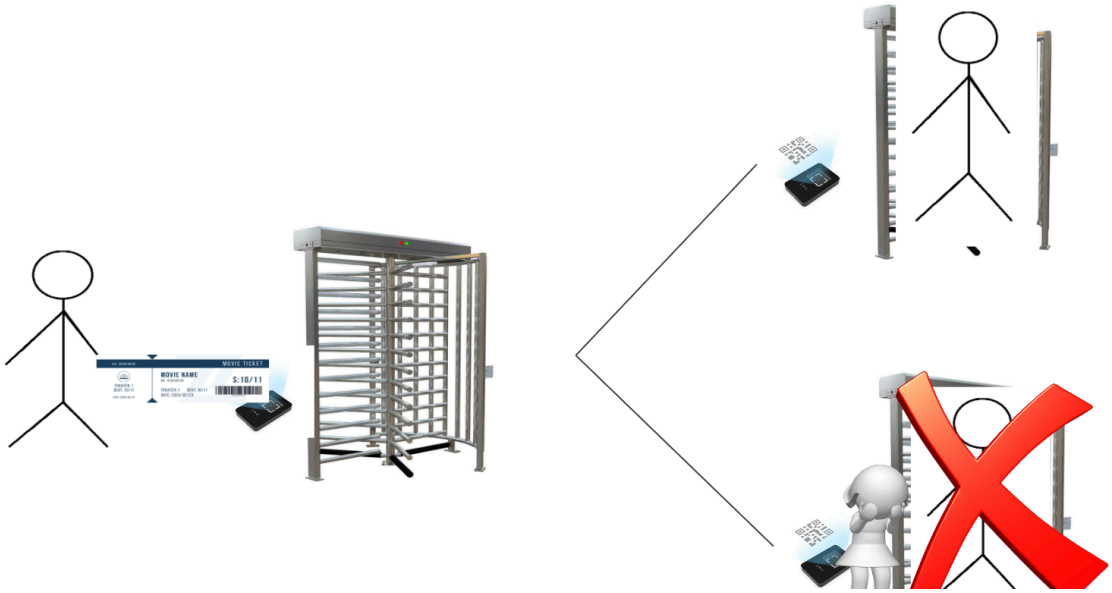
This second image shows that the information from the ticket is sent to the database. Every ticket that gets scanned will have its information sent to the database.



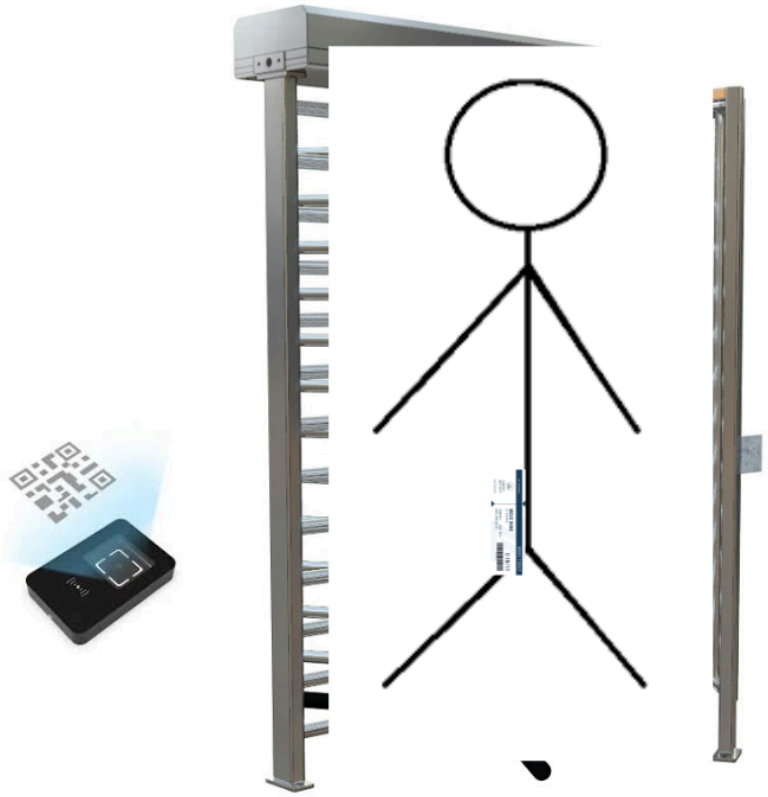
The next image is for ticket validation. Depending on the information being sent, the ticket will evaluate to either true or false or in other words, allow or deny.



This image shows the more realistic sequence of the interaction of the users and the turnstile. Either the person will be allowed through or the person will not be allowed through based on their ticket.



Assuming the ticket is valid, the customer can then proceed through the unlocked turnstile.



Once the customer walks through the turnstile, the turnstile proceeds to lock back up and the customer can continue on to their movie.