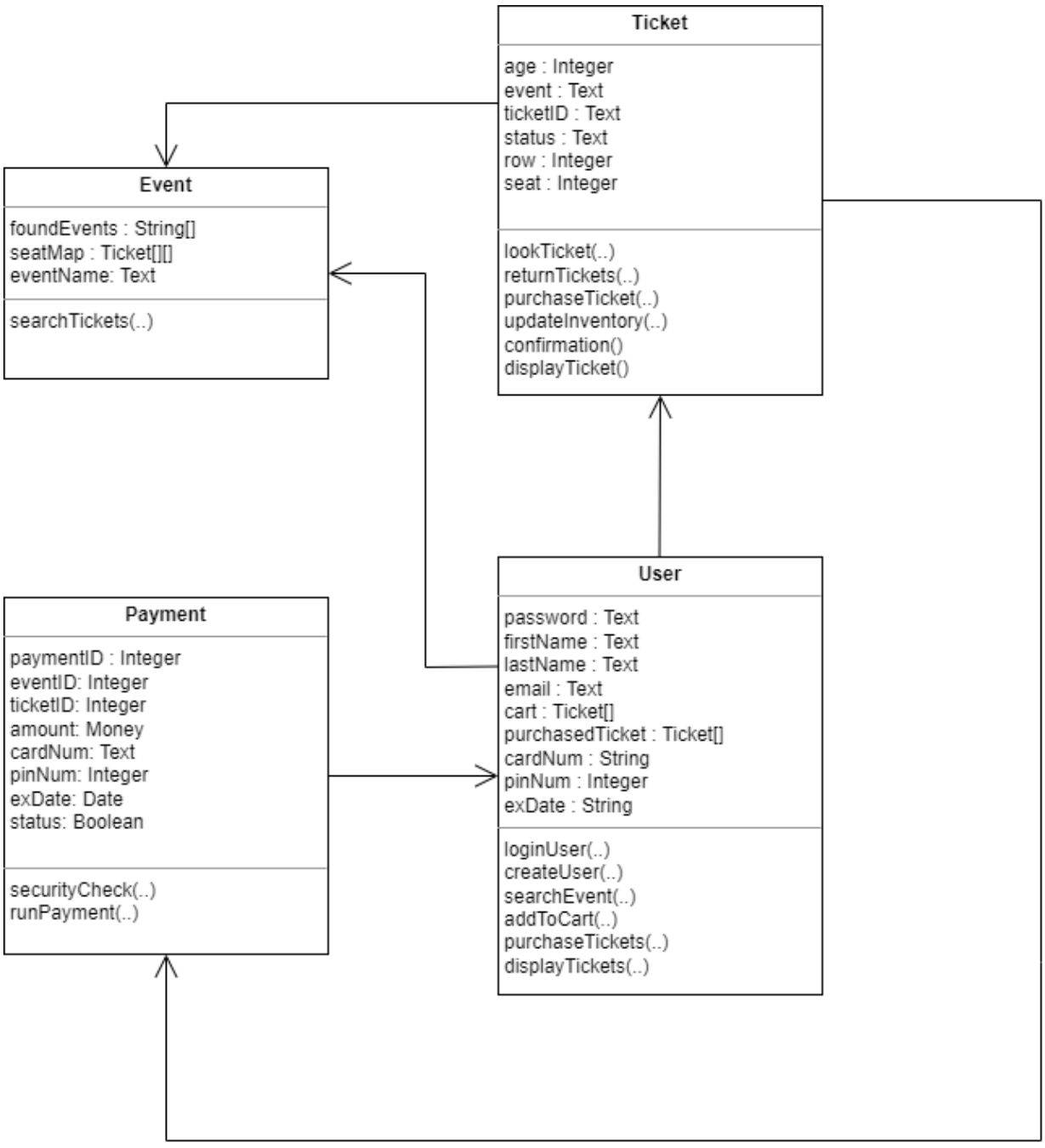
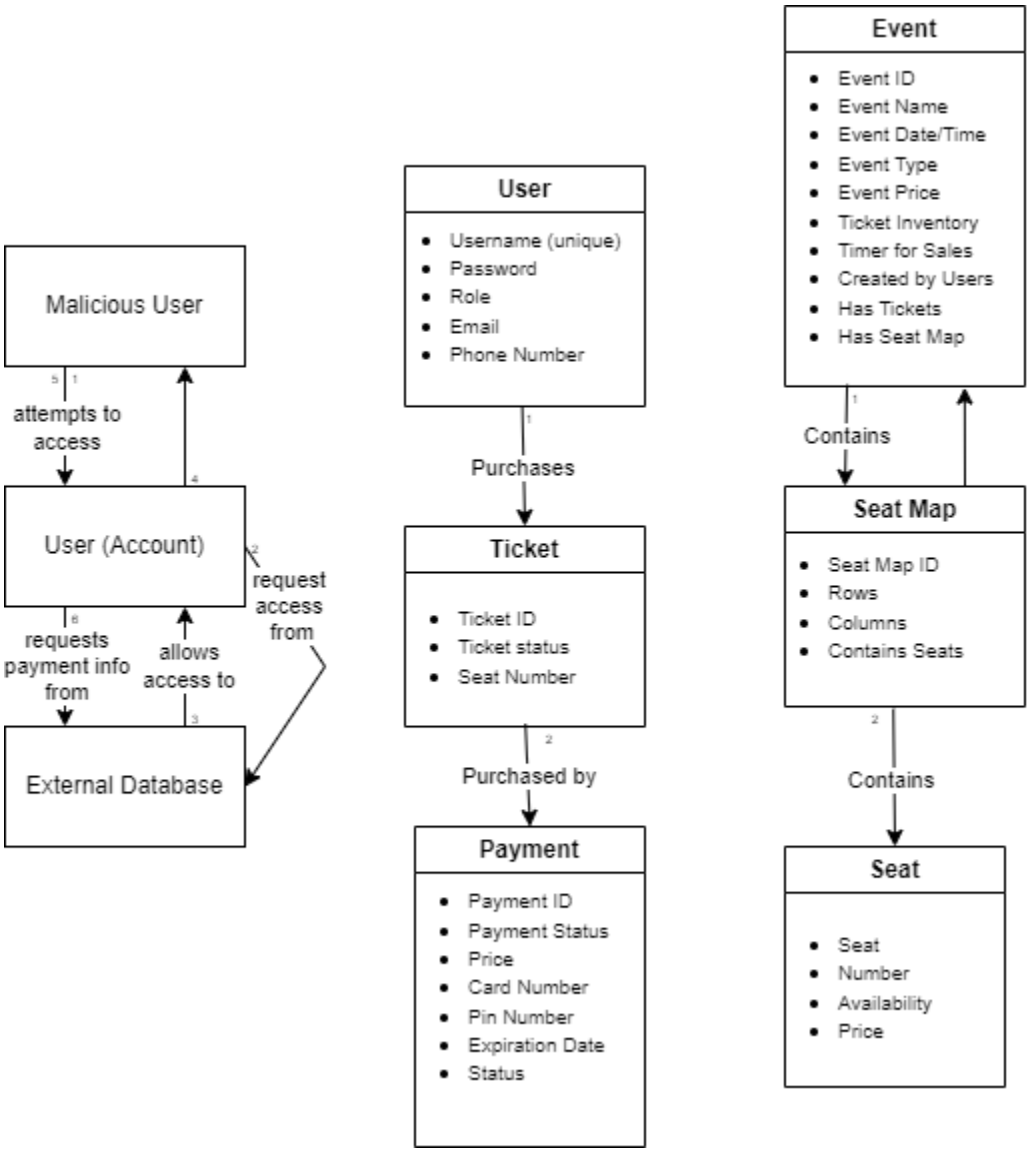
CSCI 360 T D7 SMH + A

Order of Class Implementation Diagram



Domain Model



Operations Contract: Check ticket availability

Operation	searchEvent(eventName: String)
Cross Reference	Check ticket availability
Pre-conditions	The user must be logged in and on the homepage of the application. The user must have either selected or typed in the specific event they are looking for.
Post-conditions	If they have fulfilled the necessary requirements for searching for the event, then the user will be able to select the event on the application.

Operation	searchTickets(ticketID: String)
Cross Reference	Check ticket availability
Pre-conditions	There is a user, user is logged in, user searched for an event, and has selected the event of their choosing
Post-conditions	Ticket info is updated and returned if there are tickets available.

Operation	returnTickets()
Cross Reference	Check ticket availability
Pre-conditions	The ticket info
Post-conditions	The ticket info and availability is sent and displayed to the user

Operations Contract: Login Accessibility

Operation	registerUserr()
Cross Reference	Creating a login for a user (login accessibility)
Pre-conditions	The person must have a unique username and a password that must be 15 characters and include lower case letter, upper case letter, number, and special character
Post-conditions	If they have fulfilled the necessary requirements for a unique username and a password with all of the following conditions, then they will be accepted into the system

Operation	loginUser()
Cross Reference	The system has accepted the user into the system (login accessibility)
Pre-conditions	The person must have entered their unique username and password into the system.
Post-conditions	If they have fulfilled the necessary requirements for their own unique username and a password with all of the following conditions, then they will be accepted into the application.

Operation	start2FA()
Cross reference	User attempting to login
Pre-conditions	The user has entered there username and password successfully and the screen asking for an input from the user is displayed
Post-conditions	The user has inputted the information asked from the display screen and the process of verification is confirming currently.

Operation	verify2FA()
Cross reference	User attempting to login
Pre-conditions	The user has inputted the information asked from the display screen and the process of verification is confirming currently.
Post-conditions	The verification is successful and the user is able to access the application.

Operations Contract: Purchase tickets

Operation	lookTicket(event tickets : string)
Cross Reference	Purchase tickets
Pre-conditions	The user has logged in, the system is running, and the ticket provided is valid and corresponds to an existing ticket.
Post-conditions	If the operation is successful and a valid ticket is found. The ticket details contain detailed information about the ticket that will be returned.

Operation	displayTickets (list of Tickets: string) Ticket list (Ticket ID, Event Name, Ticket Price, Availability)
Cross Reference	Purchase tickets
Pre-conditions	the system is running and accessible,the user looks up for tickets with valid event names.
Post-conditions	If the operation is successful, the tickets list contains information about available tickets which is Ticket list (Ticket ID, Event Name, Ticket Price, Availability). But if the operation is not successful it will display a message stating no availability, try another event name or date for example.

Operation	AddtoCart (ticketID: int)
Cross Reference	Purchase tickets
Pre-conditions	The TicketID provided exists in the database of available tickets. The user is authorized to modify the specified shopping cart. - The ticket is not already in the user's cart (unless the system supports multiple quantities of the same ticket in a cart).
Post-conditions	If the operation is successful, the specified ticket is added to the user's shopping cart.

Operation	PurchaseTicket(ticketID: int)
Cross Reference	Purchase ticket
Pre-conditions	The user has logged in, added tickets to their cart, and then selected “Purchase Tickets” then the ticket will be saved in the system as ready for purchase.
Post-conditions	The tickets added to cart will be ready for purchase

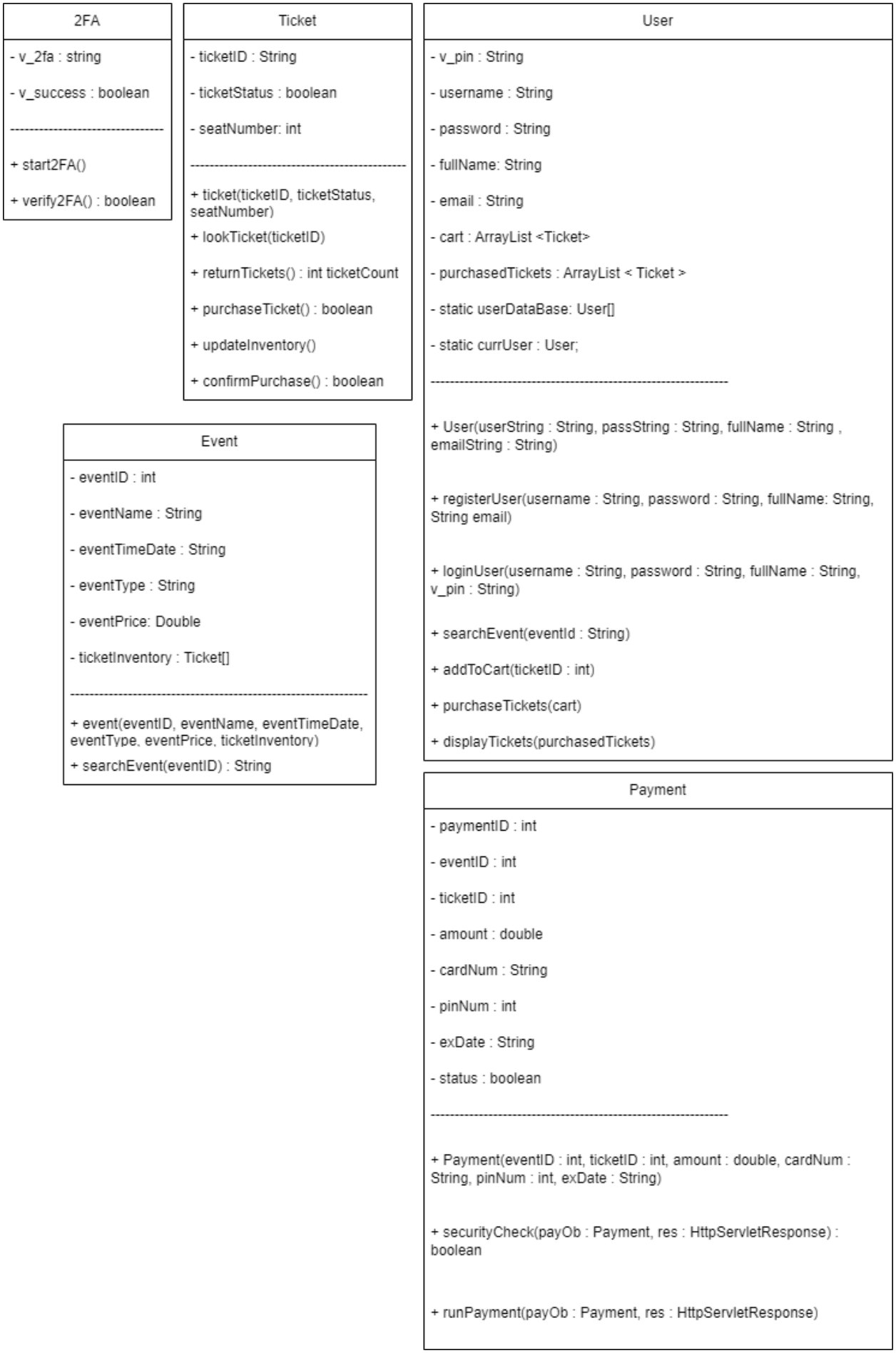
Operation	confirmPurchase () return (confirmationID: int)
Cross Reference	Purchase tickets
Pre-conditions	The system is running. The Transaction provided is valid and corresponds to an existing, pending transaction, order, or action.
Post-conditions	If the operation is successful, the specified payment transaction is marked as completed.

Operation	securityCheck(..)
Cross Reference	Purchasing the ticket
Pre-Conditions	The user has inputted their card information
Post-Conditions	The card information was confirmed the user is able to run their payment.

Operation	runPayment(..)
Cross Reference	Purchasing the ticket
Pre-Conditions	The user has successfully confirmed their card information

Post-Conditions	The tickets are now purchased and accessible by the user.

Operation	UpdateInventory(ticketID: int)
Cross Reference	Purchase tickets
Pre-conditions	The user has purchased Tickets, and the ticket inventories need to be updated.
Post-condition	The ticket inventories are updated and are confirmed.


2FA
- v_2fa : string
- v_success : boolean

+ start2FA()
+ verify2FA() : boolean
Ticket
- ticketID : String
- ticketStatus : boolean
- seatNumber: int

+ ticket(ticketID, ticketStatus, seatNumber)
+ lookTicket(ticketID)
+ returnTickets() : int ticketCount
+ purchaseTicket() : boolean
+ updateInventory()
+ confirmPurchase() : boolean
User
- v_pin : String
- username : String
- password : String
- fullName: String
- email : String
- cart : ArrayList <Ticket>
- purchasedTickets : ArrayList < Ticket >
- static userDataBase: User[]
- static currUser : User;

+ User(userString : String, passString : String, fullName : String , emailString : String)
+ registerUser(username : String, password : String, fullName: String, String email)
+ loginUser(username : String, password : String, fullName : String, v_pin : String)
+ searchEvent(eventId : String)
+ addToCart(ticketID : int)
+ purchaseTickets(cart)
+ displayTickets(purchasedTickets)
Event
- eventID : int
- eventName : String
- eventTimeDate : String
- eventType : String
- eventPrice: Double
- ticketInventory : Ticket[]

+ event(eventID, eventName, eventTimeDate, eventType, eventPrice, ticketInventory)
+ searchEvent(eventID) : String
Payment
- paymentID : int
- eventID : int
- ticketID : int
- amount : double
- cardNum : String
- pinNum : int
- exDate : String
- status : boolean

+ Payment(eventID : int, ticketID : int, amount : double, cardNum : String, pinNum : int, exDate : String)
+ securityCheck(payOb : Payment, res : HttpServletResponse) : boolean
+ runPayment(payOb : Payment, res : HttpServletResponse)