PART A.

The functional requirements for the Parking App subsystem are as follows:

- The subsystem shall allow for the application to be installed on a mobile device with a valid UNA email.
- The subsystem will require usernames and passwords and allow them to update their information.
- The subsystem shall allow users to remotely access, maintain, and update the app.
- The subsystem shall allow ticket payment via credit card, debit card, or other secure payment methods, and email a receipt of ticket payment to the user.
- The subsystem shall only show lots the user is permitted to park in.
- The subsystem shall allow users to register vehicles and renew permits.
- The subsystem shall allow students/faculty to submit visitor parking pass requests.
- The subsystem shall allow users to view a parking map.
- The subsystem shall display the busiest times.
- The subsystem shall allow users to receive real-time notifications on parking lot occupancy.
- The subsystem shall maintain user session data securely while adhering to privacy guidelines and University of North Alabama policies.
- The subsystem shall allow users to receive notifications regarding when lots become full or when new spaces become available.
- The subsystem shall automatically end sessions after twenty minutes of inactivity to free up server and network resources for other users.
- The subsystem shall be integrated with the University of North Alabama parking portal.
- The subsystem shall receive constant data from the parking lot cameras/sensors.
- The subsystem shall update parking availability for events and alert those permitted to park in those areas and notify them.
- The subsystem shall direct users to their desired lot via GPS through Google Maps or Apple Maps.

The operational requirements for the proposed subsystem are as follows:

- The subsystem shall be capable of properly functioning on Android devices running version 11 or later.
- The subsystem shall be capable of properly functioning on iOS devices running iOS version 14 or later.
- The subsystem should not cost more than one hundred and fifty thousand U.S. dollars to develop.
- The subsystem shall accommodate for high volumes of use from faculty, students and staff.

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

- The subsystem shall roll updates out to users automatically on their personal devices as needed.
- The subsystem shall automatically check for software updates on a regular basis.
- The subsystem shall have an annual uptime of ninety-nine-point nine percent which will allow 8.76 hours of downtime per year.
- The subsystem maintenance should not cost more than ten thousand U.S. dollars per year.

PART B.

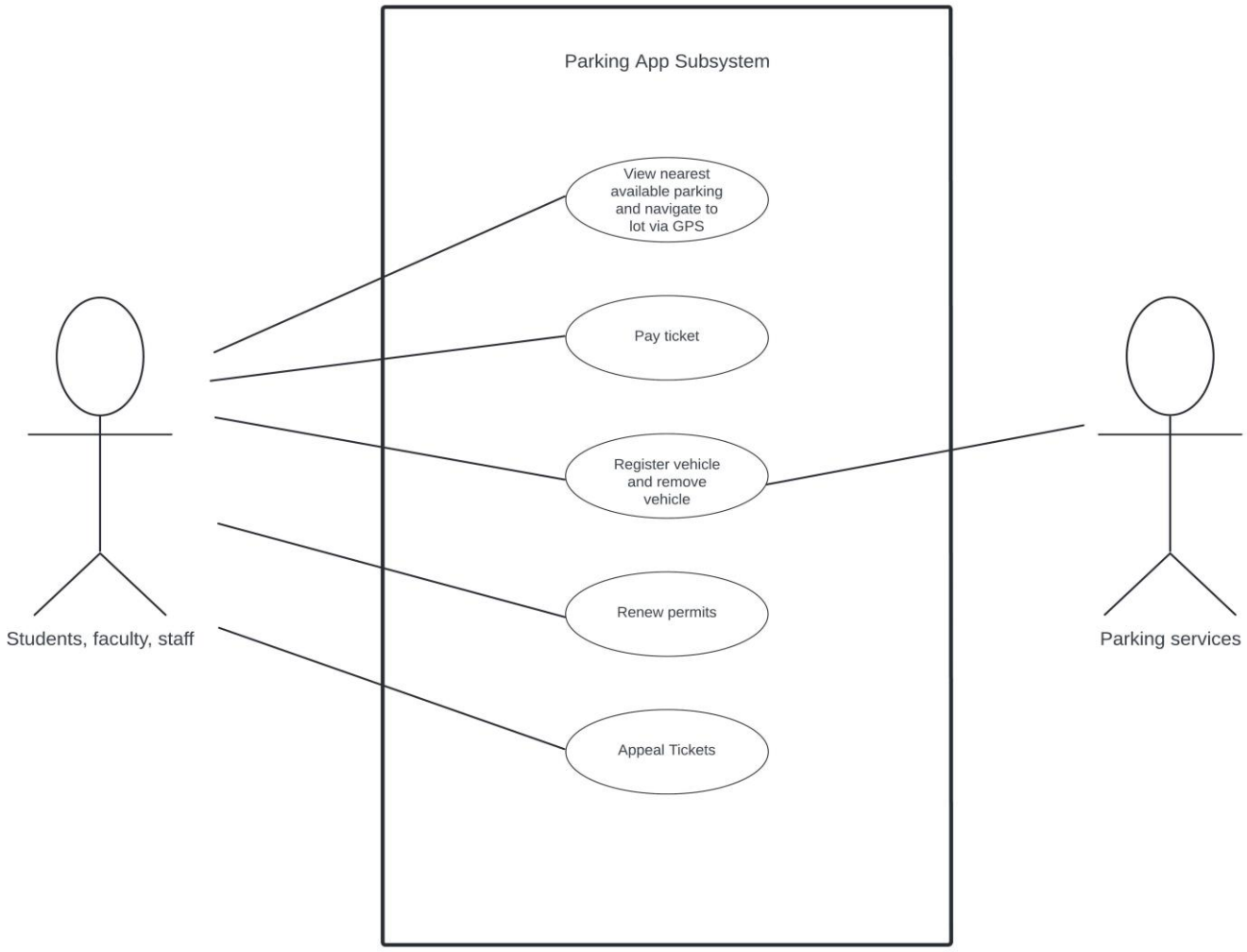
1. Identification of use cases:

Use Cases	Brief Use Case Descriptions
View parking and navigate	Users can view real-time parking lot availability and information and be given a route to nearest parking via GPS application (Google Maps, Apple Maps).
Pay Tickets	Users can pay their ticket in the app.
Register/Remove Vehicle	Users can register a vehicle and/or remove a vehicle from registration.
Renew Permits	Users can renew their permits.
Appeal Tickets	Users can appeal their tickets in the app.

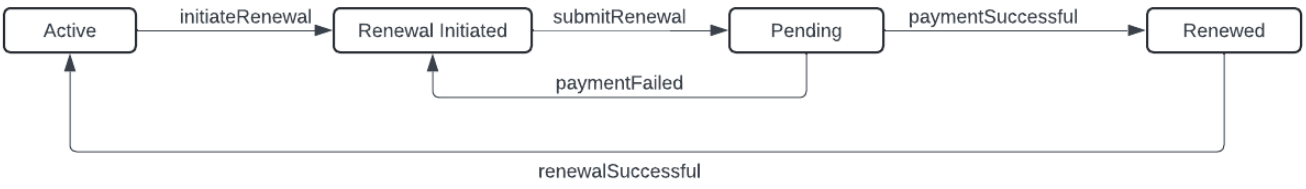
2. Determine the actors:

Use Cases	**Actors**
View parking	Students, Faculty, UNA PD
Pay Tickets	Students, Faculty
Register/Remove Vehicle	UNA PD
Renew Permits	Students, Faculty
Appeal Tickets	Students, Faculty

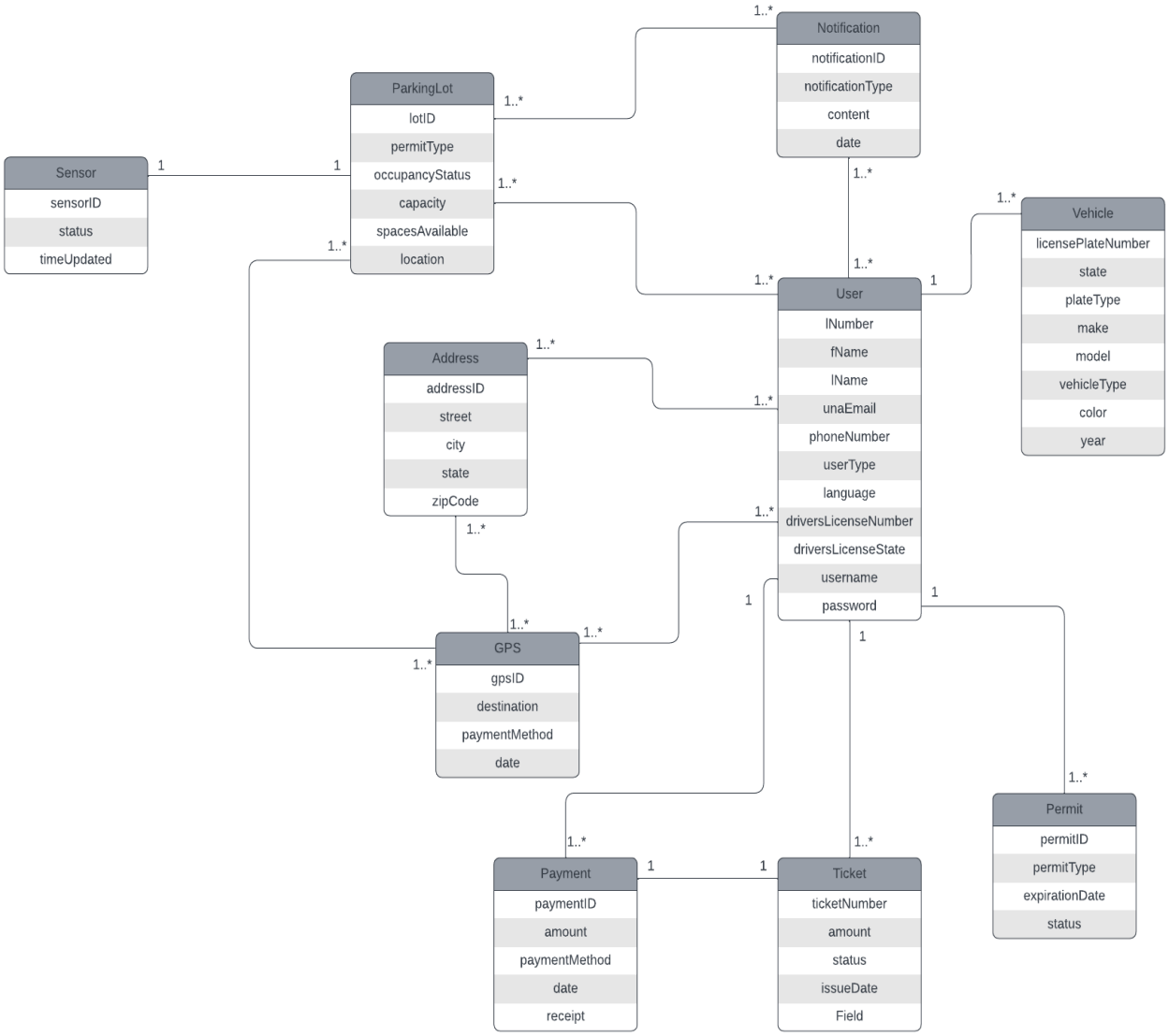
Use case diagram



State machine diagram: Permit renewal use case



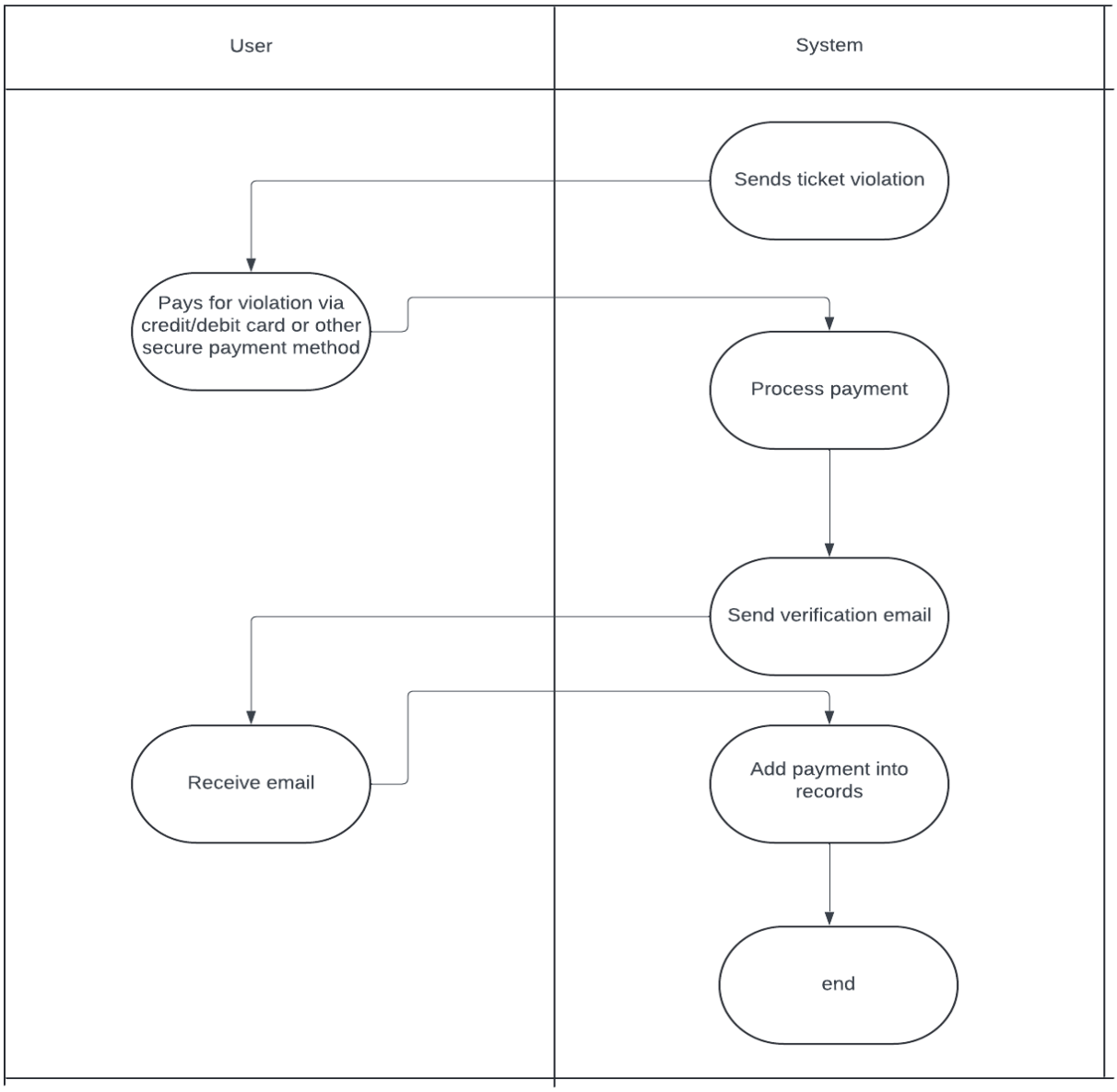
Domain model class diagram:



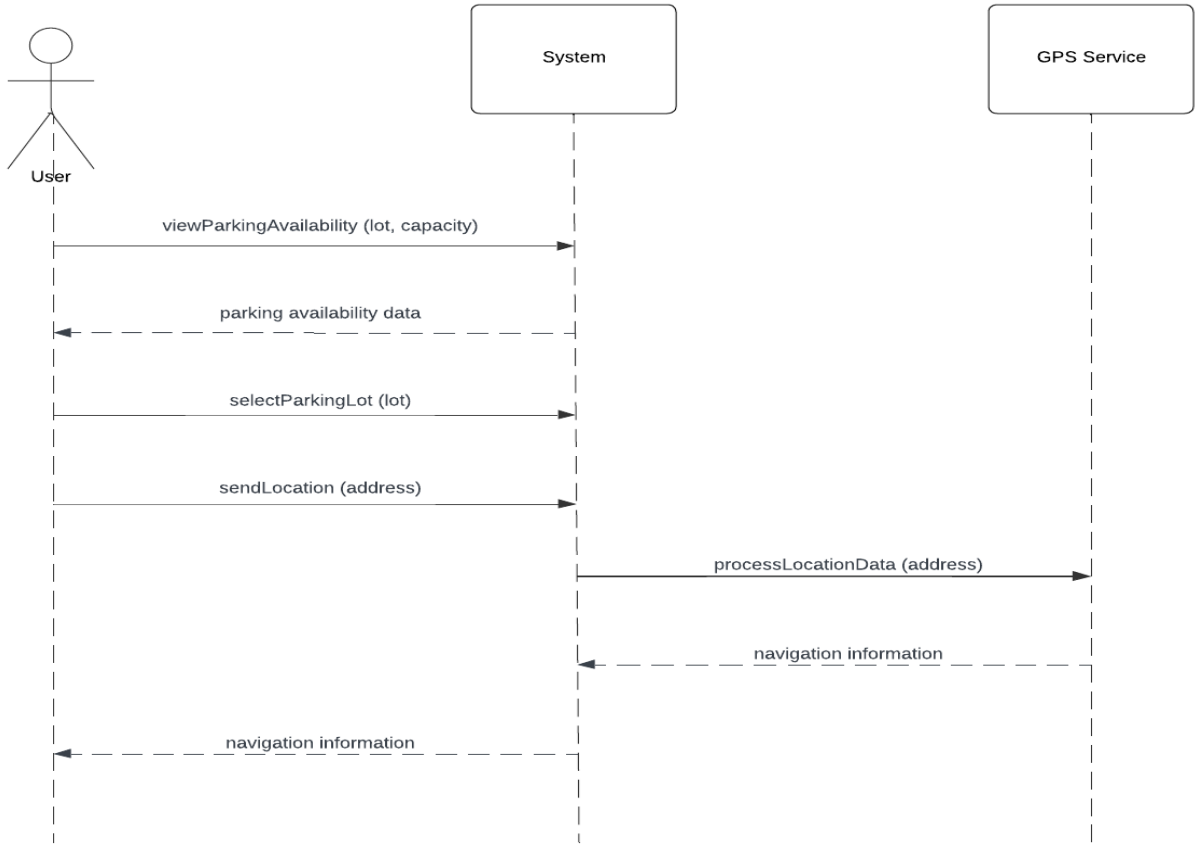
Use case description: Ticket payment use case

Use case name:	*Pay ticket.*	
Scenario:	Pay a ticket.	
Triggering event:	User wants to pay a recieved ticket on the app.	
Brief description:	User navigates to the Parking App's payment tab and pays ticket with a credit or debit card.	
Actors:	Students, faculty, and staff.	
Related use cases:	May be invoked by *register a vehicle* use case if an unregistered vehicle is added to an account thus linking the ticket to an account.	
Stakeholders:	University of North Alabama, Parking Services, UNA Police Department.	
Preconditions:	OpsCom must be fully integrated with Parking App. A payment system for credit and debit cards must be available.	
Postconditions:	Ticket must be removed in app after successful payment. OpsCom must be updated to remove ticket in its system. User account balance must return to value it had before the ticket was issued. If hold was added to user account, it must be removed upon payment of the ticket.	
Flow of activities:	**Actor**	**System**
	1. User wants to pay ticket after recieving notification of violation from the app.	1.1 System sends notification of violation to user.
	2. User opens app to payment tab.	2.1 System pulls all stored tickets on user upon request in payment tab. 2.2 System displays timestamp and potential late fee on ticket to user.
	3. User selects ticket and checkout button.	3.1 System navigates to payment API as designated by the University of North Alabama
	4. User enters credit or debit card information and completes payment.	4.1 System removes ticket from app and OpsCom upon successful payment. 4.2 System sends reciept of payment in email to user. 4.3 System removes hold from user account if neccesary. 4.4 System holds record of payment upon completion.
Exception conditions:	1.1 User does not have a ticket. 4.1 Ticket payment was unsuccessful. 4.2 User has invalid email address linked to account. 4.3 User still has pending tickets.	

Activity diagram: Ticket payment use case



System sequence diagram: Access GPS use case



CRUD:

Use case	Students	Faculty	UNA PD
Users can view parking lot availability	R	R	R
Users are guided to the nearest parking lot with availability via GPS	R	R	
Users pay their parking ticket	U	U	
Users register a vehicle	CU	CU	
Users remove a vehicle from registration	D	D	