TEZ

ONE STOP SOLUTION FOR STUDENT TRANSPORT NEEDS

Members:

Ibtehaj Haider (22i-0767) - *PRODUCT OWNER*

Muaz Ahmed (22i-1125) - *SCRUM MASTER*

Usman Haroon (22i-1177) - *SCRUM MEMBER*

INTRODUCTION OF SYSTEM

Purpose:

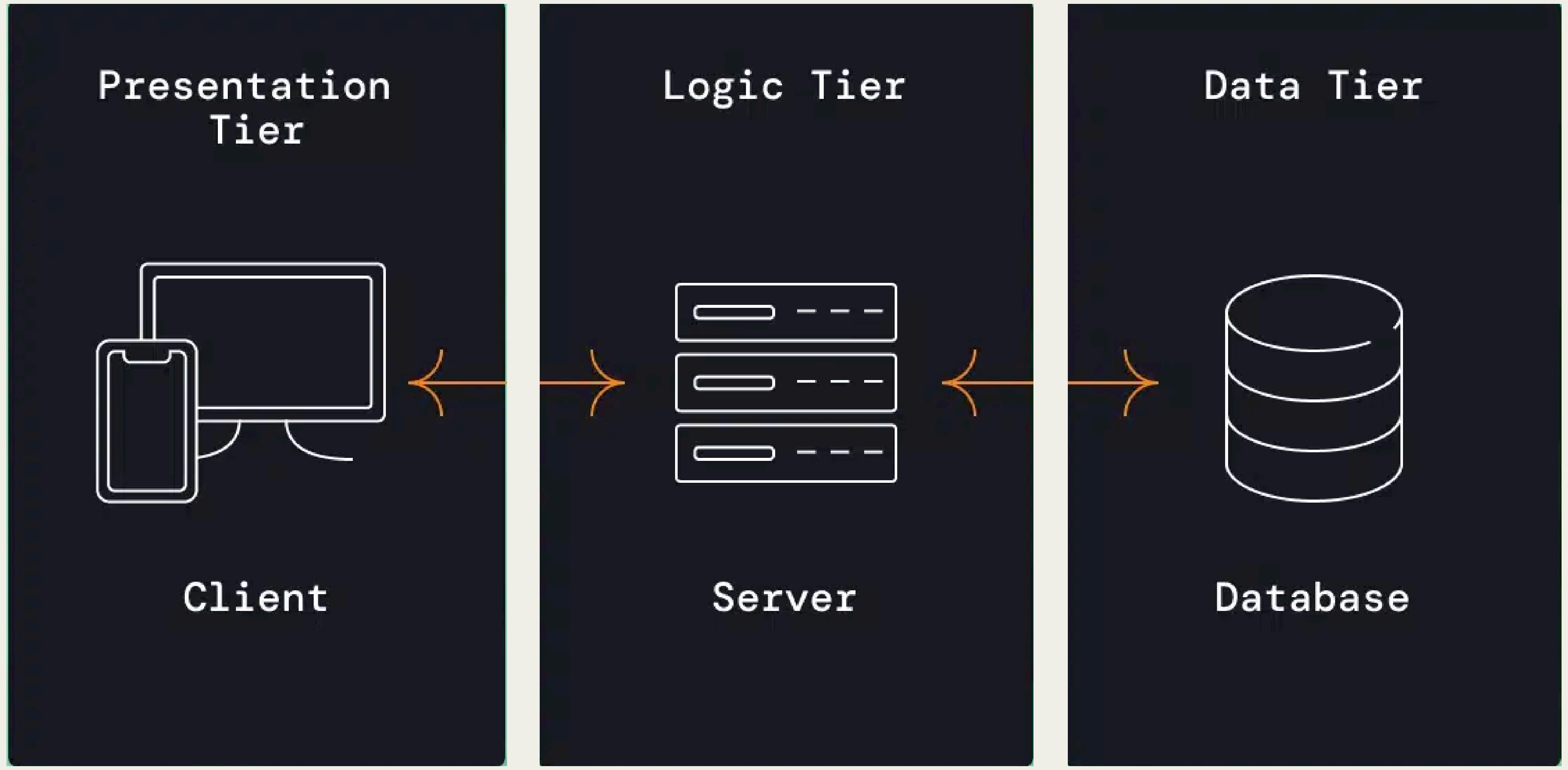
- Streamline and digitize the public transportation experience for educational institutions

Goals:

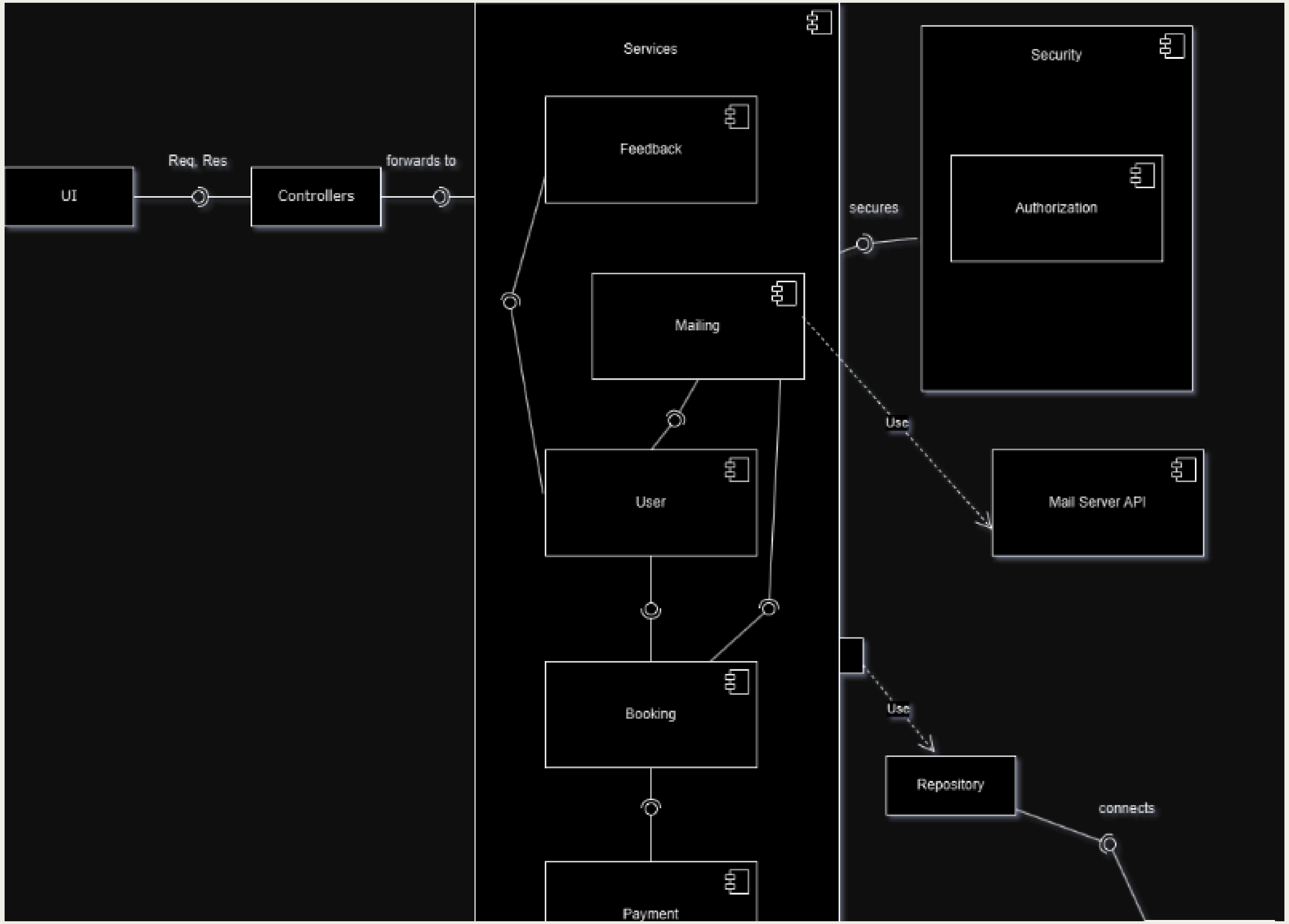
- Real-time bus tracking
- Automated seat management
- Secure authentication
- Improved communication between students, drivers, and administrators.

Architecture:

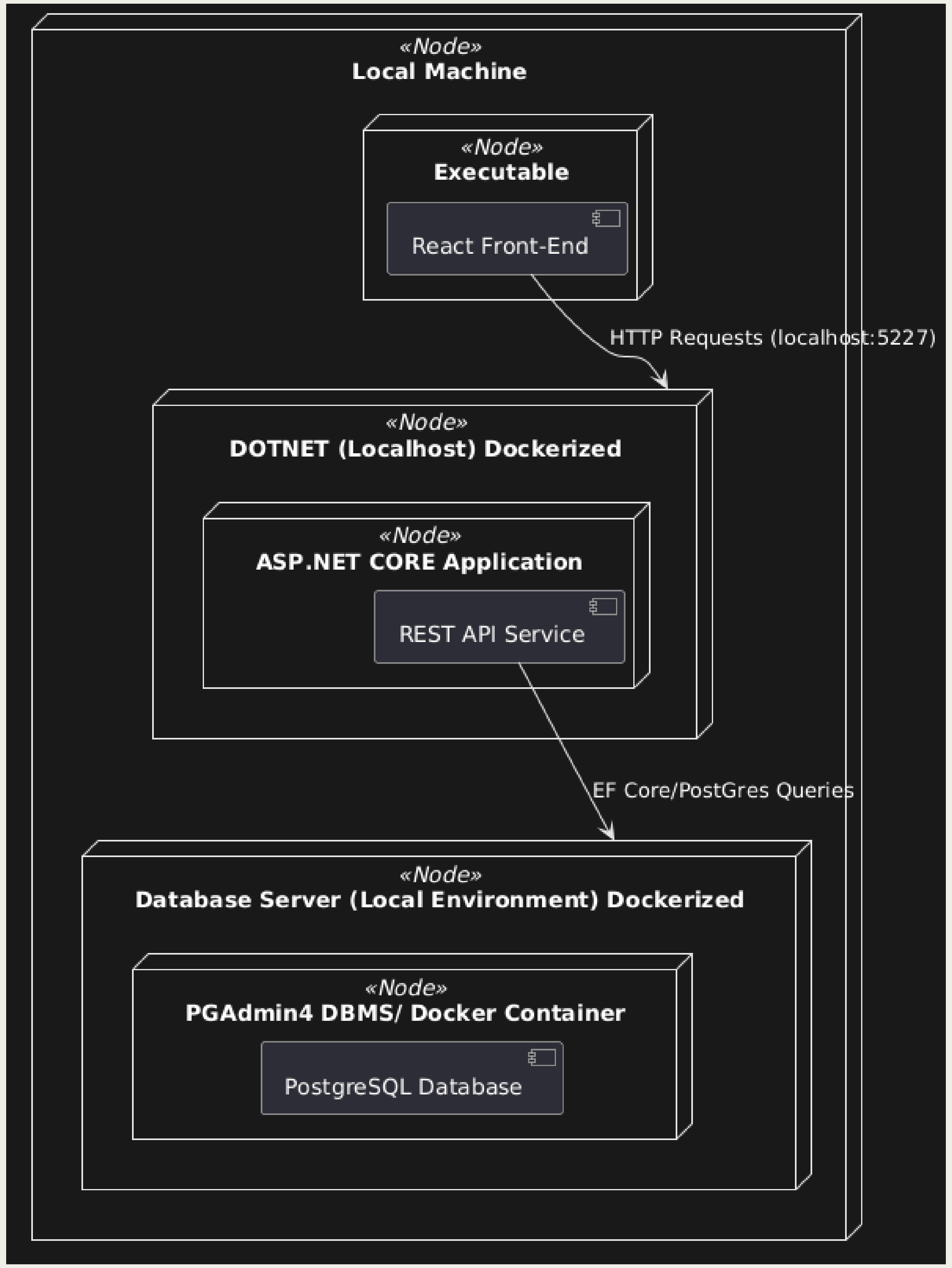
3- Tier Architecture



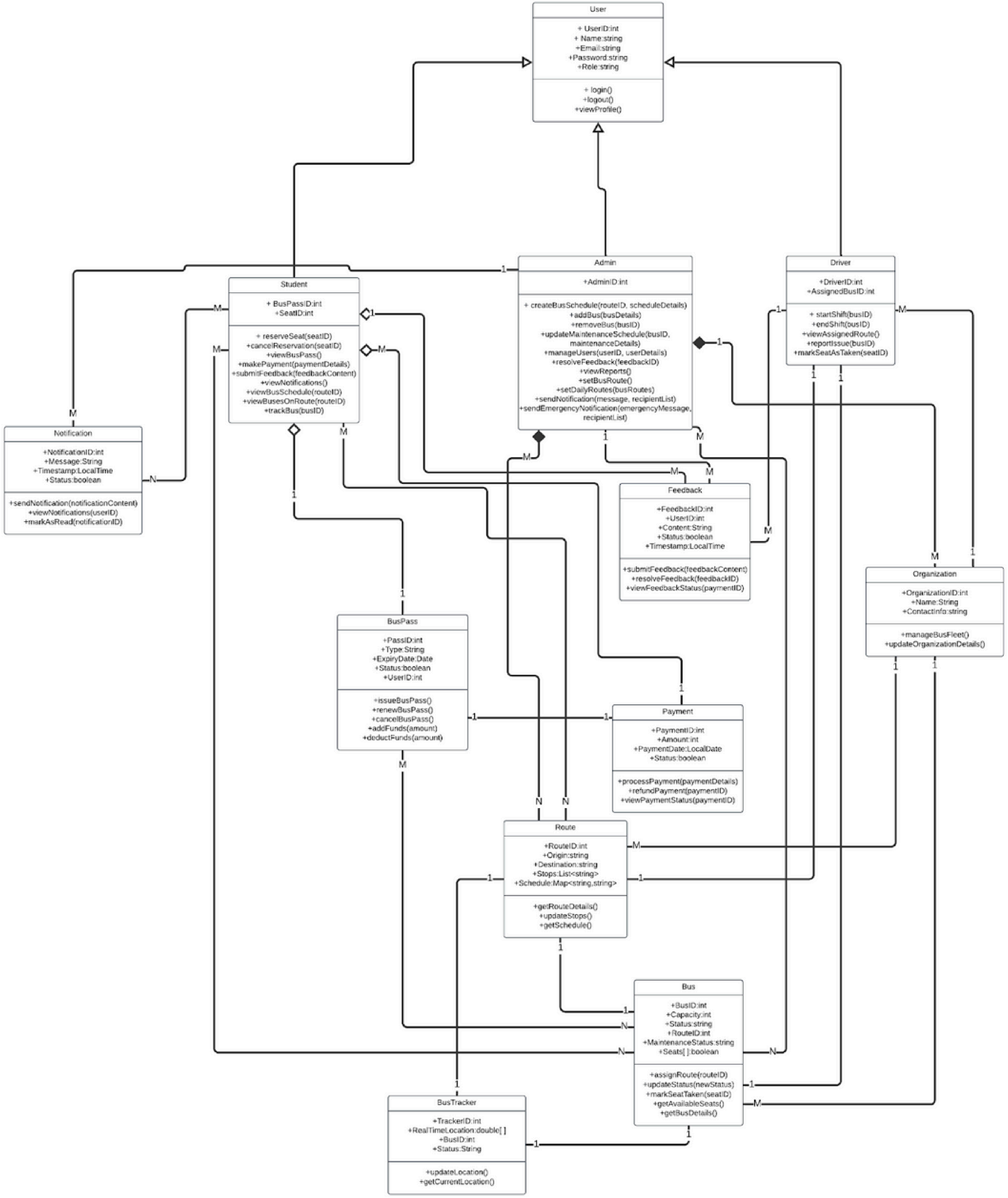
Component Diagram



Deployment Diagram



Class Diagram



User

+ UserID:int
+ Name:string
+Email:string
+Password:string
+Role:string

+ login()
+logout()
+viewProfile()

Admin

+AdminID:int

+ createBusSchedule(routeID, scheduleDetails)
+addBus(busDetails)
+removeBus(busID)
+updateMaintenanceSchedule(busID, maintenanceDetails)
+manageUsers(userID, userDetails)
+resolveFeedback(feedbackID)
+viewReports()
+setBusRoute()
+setDailyRoutes(busRoutes)
+sendNotification(message, recipientList)
+sendEmergencyNotification(emergencyMessage, recipientList)

Driver

+DriverID:int
+AssignedBusID:int

+ startShift(busID)
+endShift(busID)
+viewAssignedRoute()
+reportIssue(busID)
+markSeatAsTaken(seatID)

Student

+ BusPassID:int
+SeatID:int

+ reserveSeat(seatID)
+cancelReservation(seatID)
+viewBusPass()
+makePayment(paymentDetails)
+submitFeedback(feedbackContent)
+viewNotifications()
+viewBusSchedule(routeID)
+viewBusesOnRoute(routeID)
+trackBus(busID)

Notification

+NotificationID:int
+Message:String
+Timestamp:LocalTime
+Status:boolean

+sendNotification(notificationContent)
+viewNotifications(userID)
+markAsRead(notificationID)

Feedback

+FeedbackID:int
+UserID:int
+Content:String
+Status:boolean
+Timestamp:LocalTime

+submitFeedback(feedbackContent)
+resolveFeedback(feedbackID)
+viewFeedbackStatus(paymentID)

Organization

+OrganizationID:int
+Name:String
+ContactInfo:string

+manageBusFleet()
+updateOrganizationDetails()

BusPass

+PassID:int
+Type:String
+ExpiryDate:Date
+Status:boolean
+UserID:int

+issueBusPass()
+renewBusPass()
+cancelBusPass()
+addFunds(amount)
+deductFunds(amount)

Payment

+PaymentID:int
+Amount:int
+PaymentDate:LocalDate
+Status:boolean

+processPayment(paymentDetails)
+refundPayment(paymentID)
+viewPaymentStatus(paymentID)

Route

+RouteID:int
+Origin:string
+Destination:string
+Stops:List<string>
+Schedule:Map<string,string>

+getRouteDetails()
+updateStops()
+getSchedule()

Bus

+BusID:int
+Capacity:int
+Status:string
+RouteID:int
+MaintenanceStatus:string
+Seats[ ]:boolean

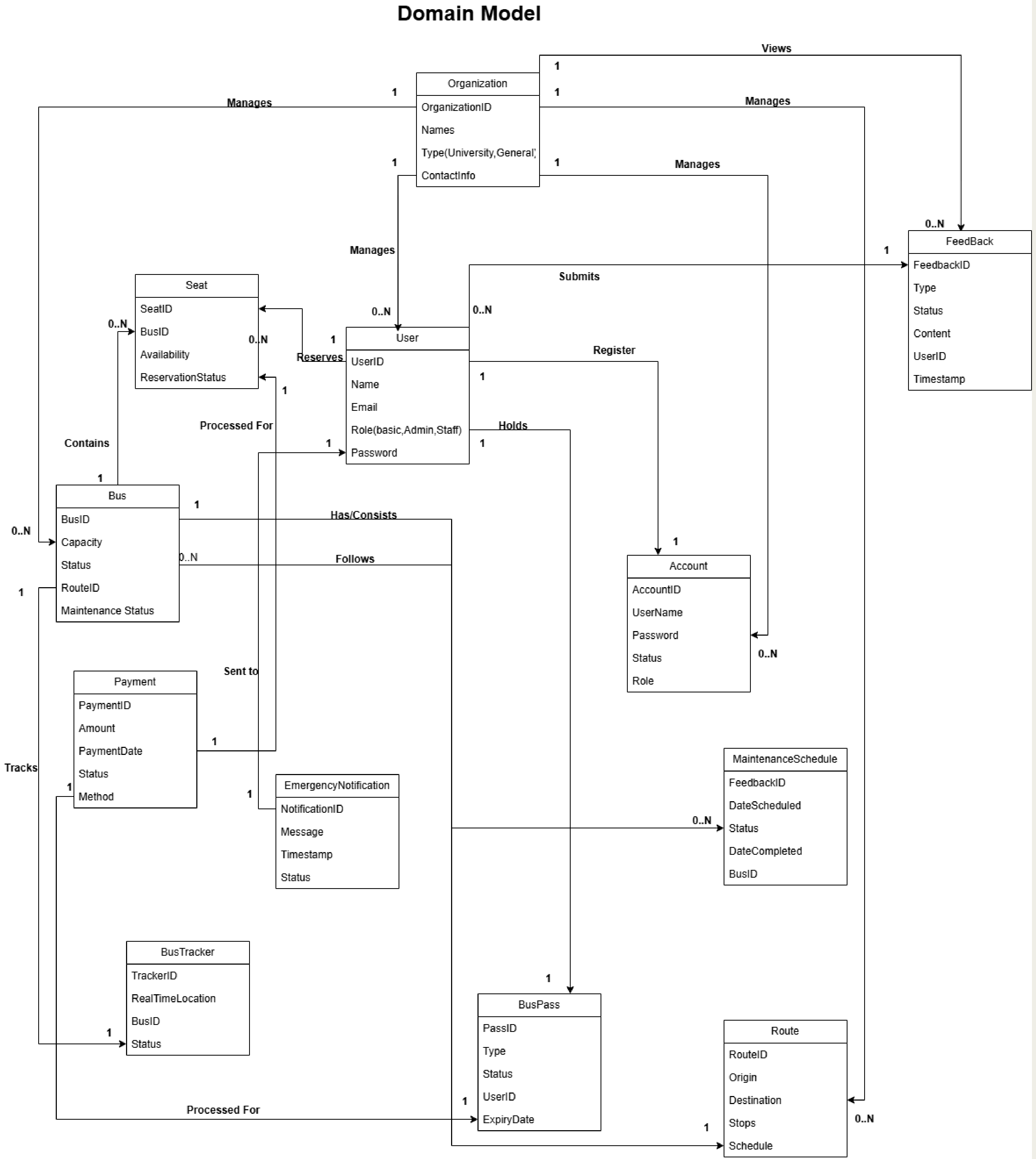
+assignRoute(routeID)
+updateStatus(newStatus)
+markSeatTaken(seatID)
+getAvailableSeats()
+getBusDetails()

BusTracker

+TrackerID:int
+RealTimeLocation:double[ ]
+BusID:int
+Status:String

+updateLocation()
+getCurrentLocation()

Domain Model



Organization
- OrganizationID
- Names
- Type(University,General)
- ContactInfo

Seat
- SeatID
- BusID
- Availability
- ReservationStatus

User
- UserID
- Name
- Email
- Role(basic,Admin,Staff)
- Password

FeedBack
- FeedbackID
- Type
- Status
- Content
- UserID
- Timestamp

Bus
- BusID
- Capacity
- Status
- RouteID
- Maintenance Status

Account
- AccountID
- UserName
- Password
- Status
- Role

Payment
- PaymentID
- Amount
- PaymentDate
- Status
- Method

EmergencyNotification
- NotificationID
- Message
- Timestamp
- Status

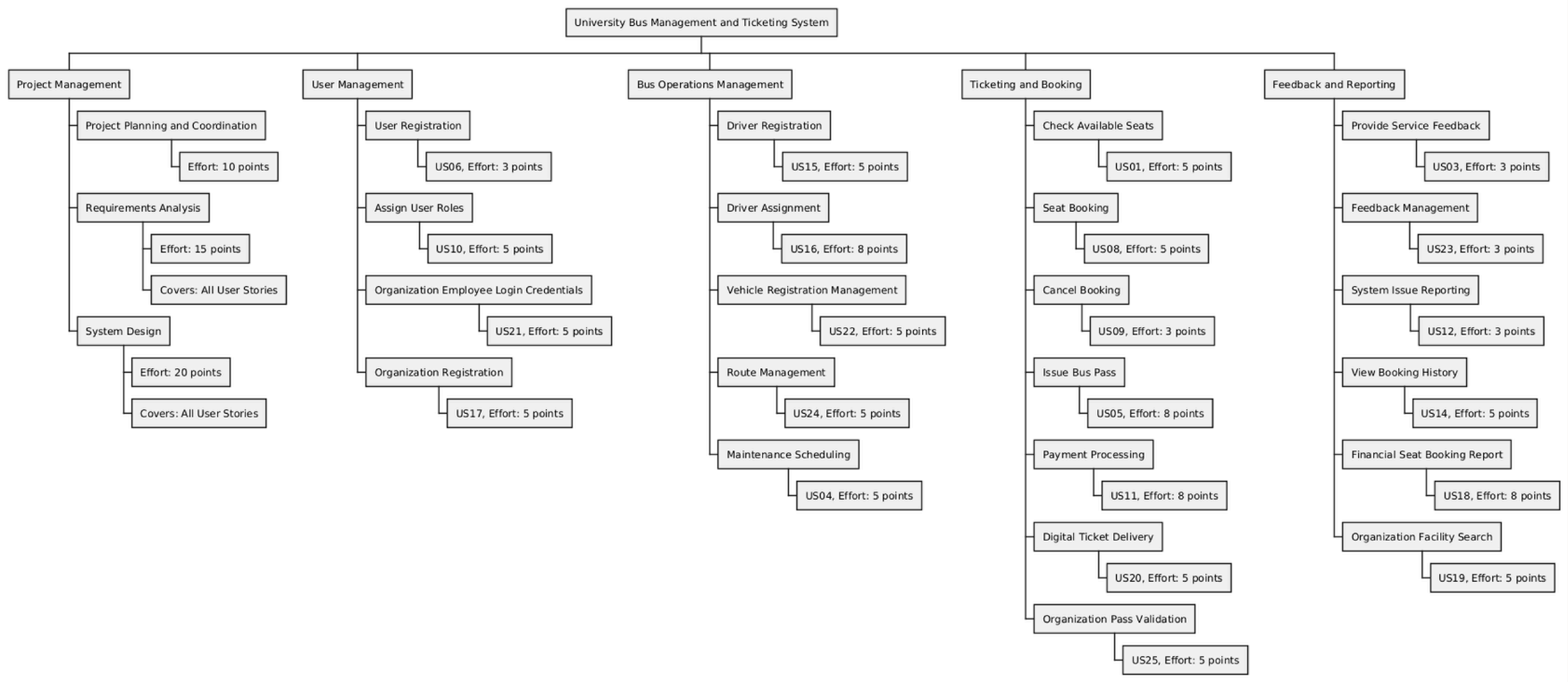
MaintenanceSchedule
- FeedbackID
- DateScheduled
- Status
- DateCompleted
- BusID

BusTracker
- TrackerID
- RealTimeLocation
- BusID
- Status

BusPass
- PassID
- Type
- Status
- UserID
- ExpiryDate

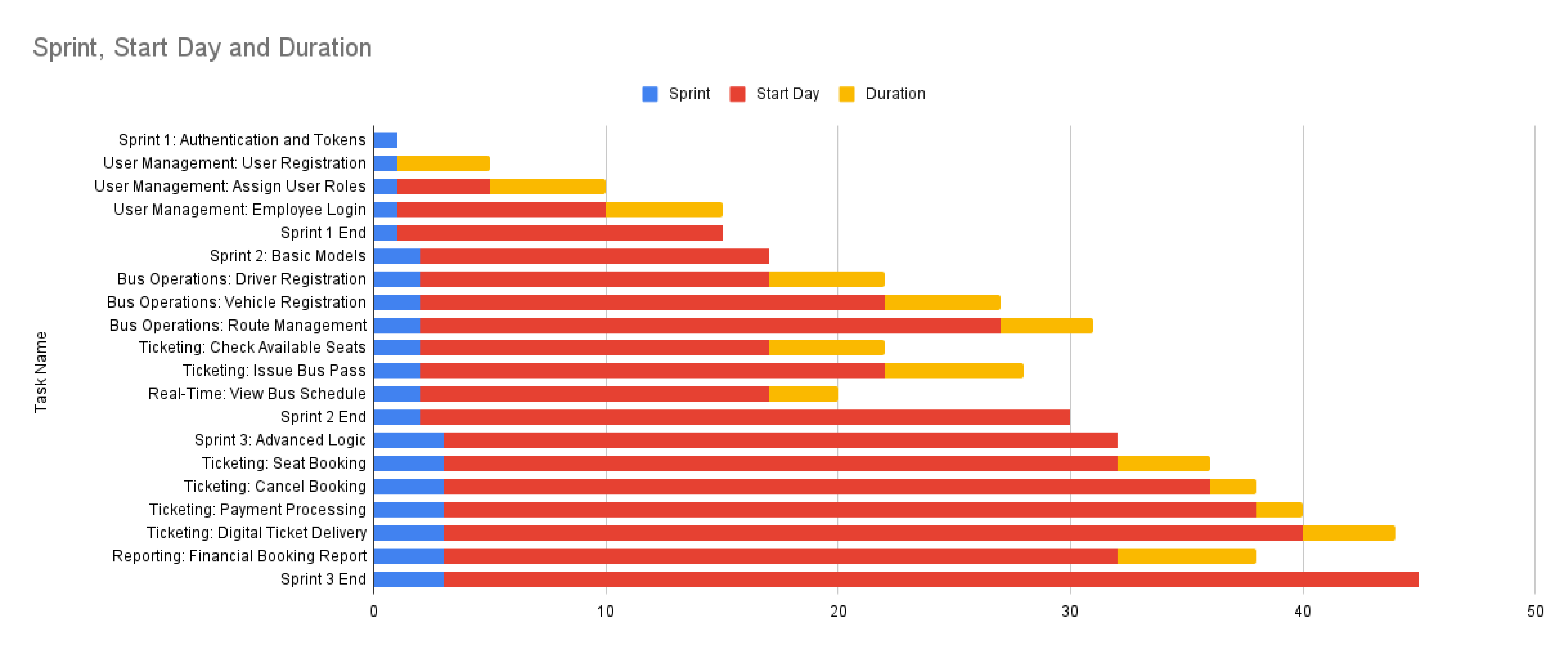
Route
- RouteID
- Origin
- Destination
- Stops
- Schedule

Project Planner - Work Breakdown Structure



- University Bus Management and Ticketing System
  - Project Management
    - Project Planning and Coordination
      - Effort: 10 points
    - Requirements Analysis
      - Effort: 15 points
      - Covers: All User Stories
    - System Design
      - Effort: 20 points
      - Covers: All User Stories
  - User Management
    - User Registration
      - US06, Effort: 3 points
    - Assign User Roles
      - US10, Effort: 5 points
    - Organization Employee Login Credentials
      - US21, Effort: 5 points
    - Organization Registration
      - US17, Effort: 5 points
  - Bus Operations Management
    - Driver Registration
      - US15, Effort: 5 points
    - Driver Assignment
      - US16, Effort: 8 points
    - Vehicle Registration Management
      - US22, Effort: 5 points
    - Route Management
      - US24, Effort: 5 points
    - Maintenance Scheduling
      - US04, Effort: 5 points
  - Ticketing and Booking
    - Check Available Seats
      - US01, Effort: 5 points
    - Seat Booking
      - US08, Effort: 5 points
    - Cancel Booking
      - US09, Effort: 3 points
    - Issue Bus Pass
      - US05, Effort: 8 points
    - Payment Processing
      - US11, Effort: 8 points
    - Digital Ticket Delivery
      - US20, Effort: 5 points
    - Organization Pass Validation
      - US25, Effort: 5 points
  - Feedback and Reporting
    - Provide Service Feedback
      - US03, Effort: 3 points
    - Feedback Management
      - US23, Effort: 3 points
    - System Issue Reporting
      - US12, Effort: 3 points
    - View Booking History
      - US14, Effort: 5 points
    - Financial Seat Booking Report
      - US18, Effort: 8 points
    - Organization Facility Search
      - US19, Effort: 5 points

Project Planner - Gantt Chart



User Stories - US01: Check Available Seats

User Story: As a student, I want to check available seats in real-time, so that I can decide whether to board the bus.

Importance: High
Estimate: 5 hours
Type: Search, Report/View

Acceptance Criteria:

- Given that I am a logged-in student, when I view the bus seat availability page, then I should see the current seat availability for all buses.
- Given that another user has just booked a seat, when I am viewing the seat availability page, then the seat status should automatically update to reflect the new booking.

User Stories - US02: View Bus Schedule

User Story: As a student, I want to view the bus schedule, so that I can plan my journey efficiently.

Importance: High
Estimate: 3 hours
Type: Report/View

Acceptance Criteria:

- Given that I am a logged-in student, when I navigate to the bus schedule page, then I should see an updated list of all bus routes with their departure and arrival times.
- Given that I am viewing the bus schedule, when I select a specific route, then I should see detailed expected arrival and departure times for all stops on that route.

User Stories - US03: Provide Service Feedback

User Story: As a student, I want to submit feedback on my bus experience, so that the administration can improve the service.

Importance: Medium
Estimate: 3 hours
Type: Manage Data

Acceptance Criteria:

- Given that I am a logged-in student, when I navigate to the feedback section, then I should see a form to submit my feedback about the bus service.
- Given that I have filled out the feedback form, when I submit the form, then I should receive a confirmation message that my feedback has been stored in the system.

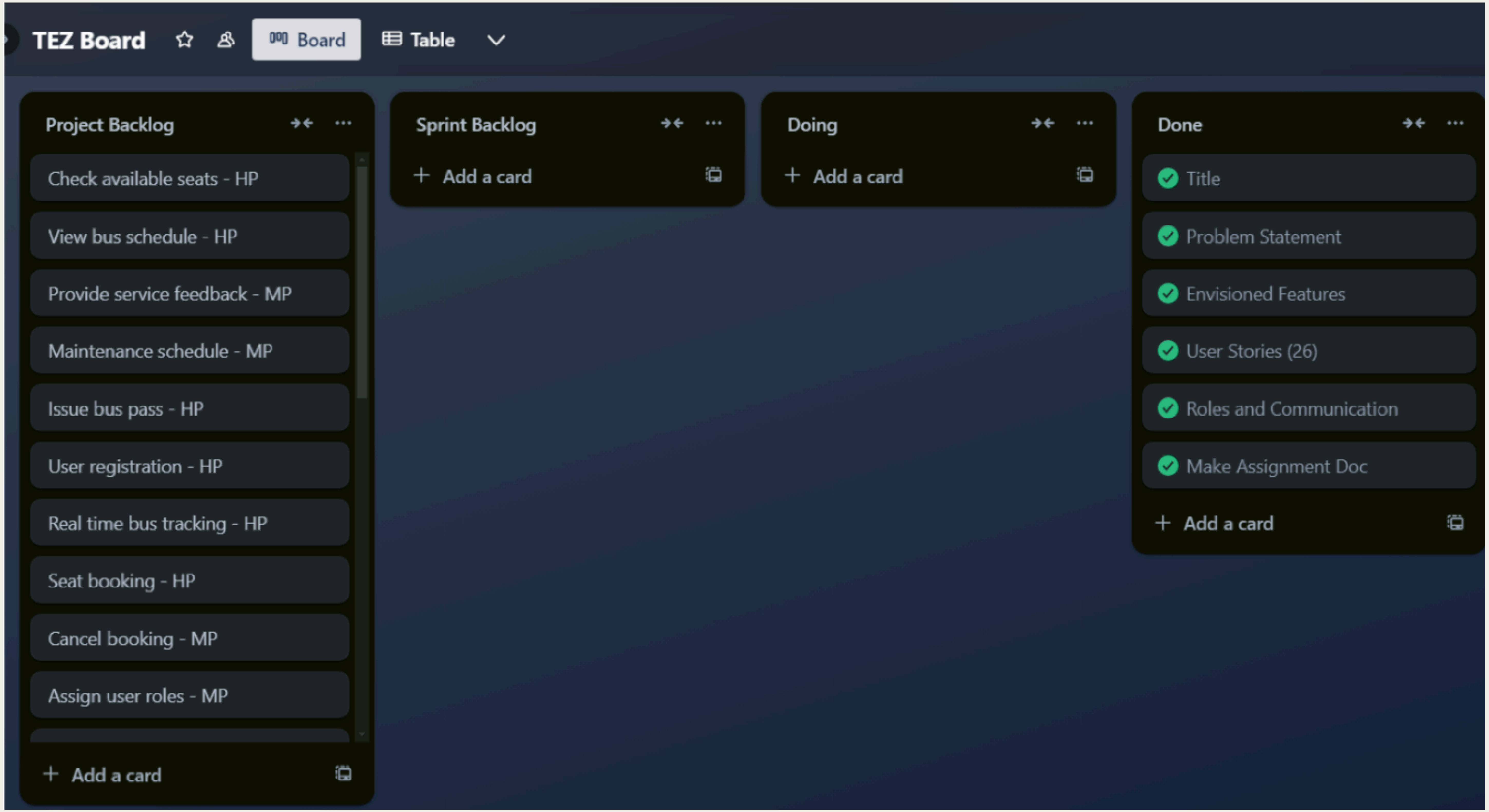
User Stories - US04: Seat Booking

User Story: As a student, I want to reserve a seat in advance, so that I have a confirmed spot on the bus.

Importance: High
Estimate: 5 hours
Type: Workflow, Manage Data

Acceptance Criteria:

- Given that I am a logged-in student, when I navigate to the seat booking page, then I should see all available seats that can be reserved.
- Given that I select an available seat, when I confirm my booking, then I should receive a confirmation of my reservation.
- Given that a seat has been reserved, when another student views the seat availability, then the reserved seat should be shown as unavailable.



Project Backlog

- Check available seats - HP
- View bus schedule - HP
- Provide service feedback - MP
- Maintenance schedule - MP
- Issue bus pass - HP
- User registration - HP
- Real time bus tracking - HP
- Seat booking - HP
- Cancel booking - MP
- Assign user roles - MP

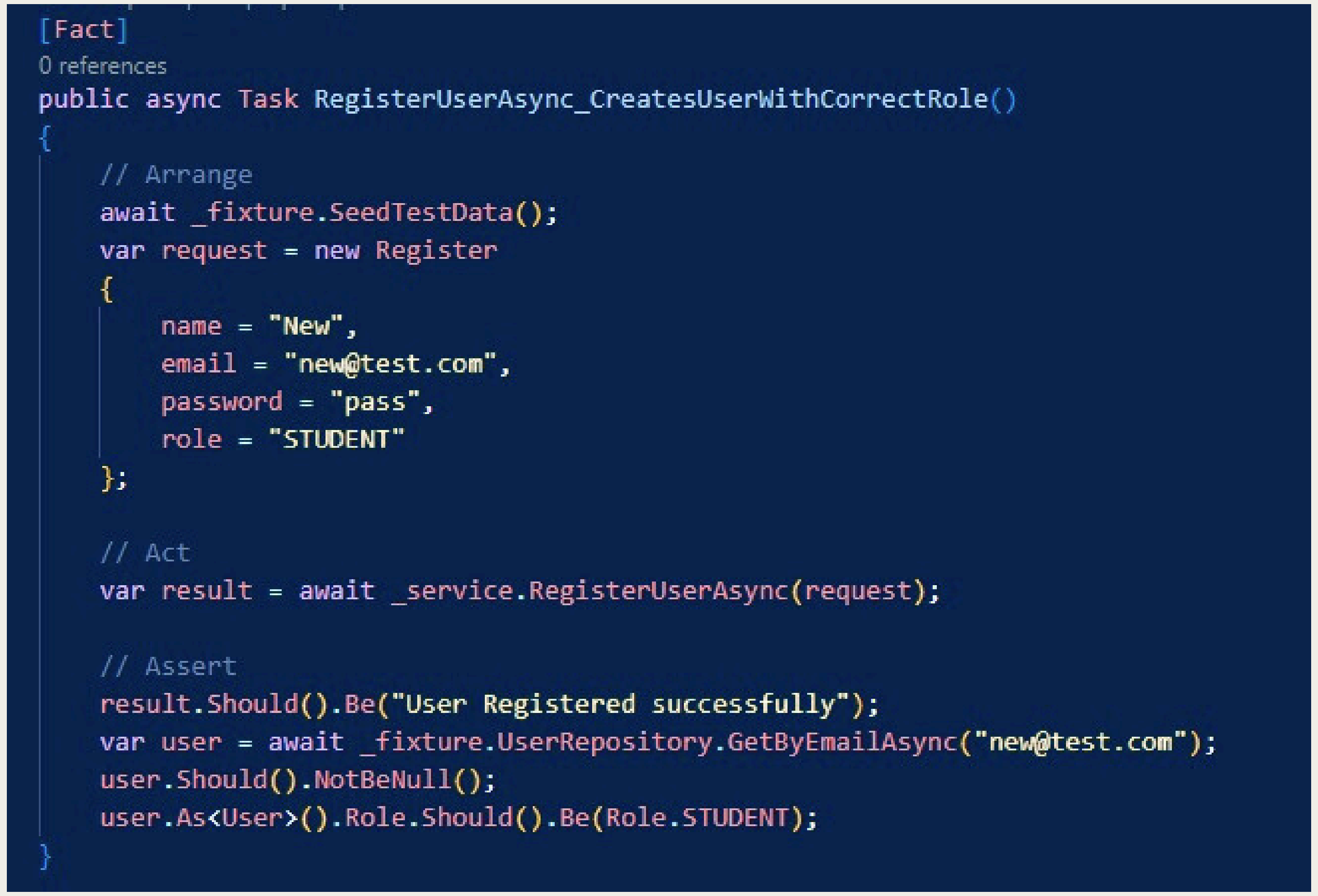
Sprint Backlog

Doing

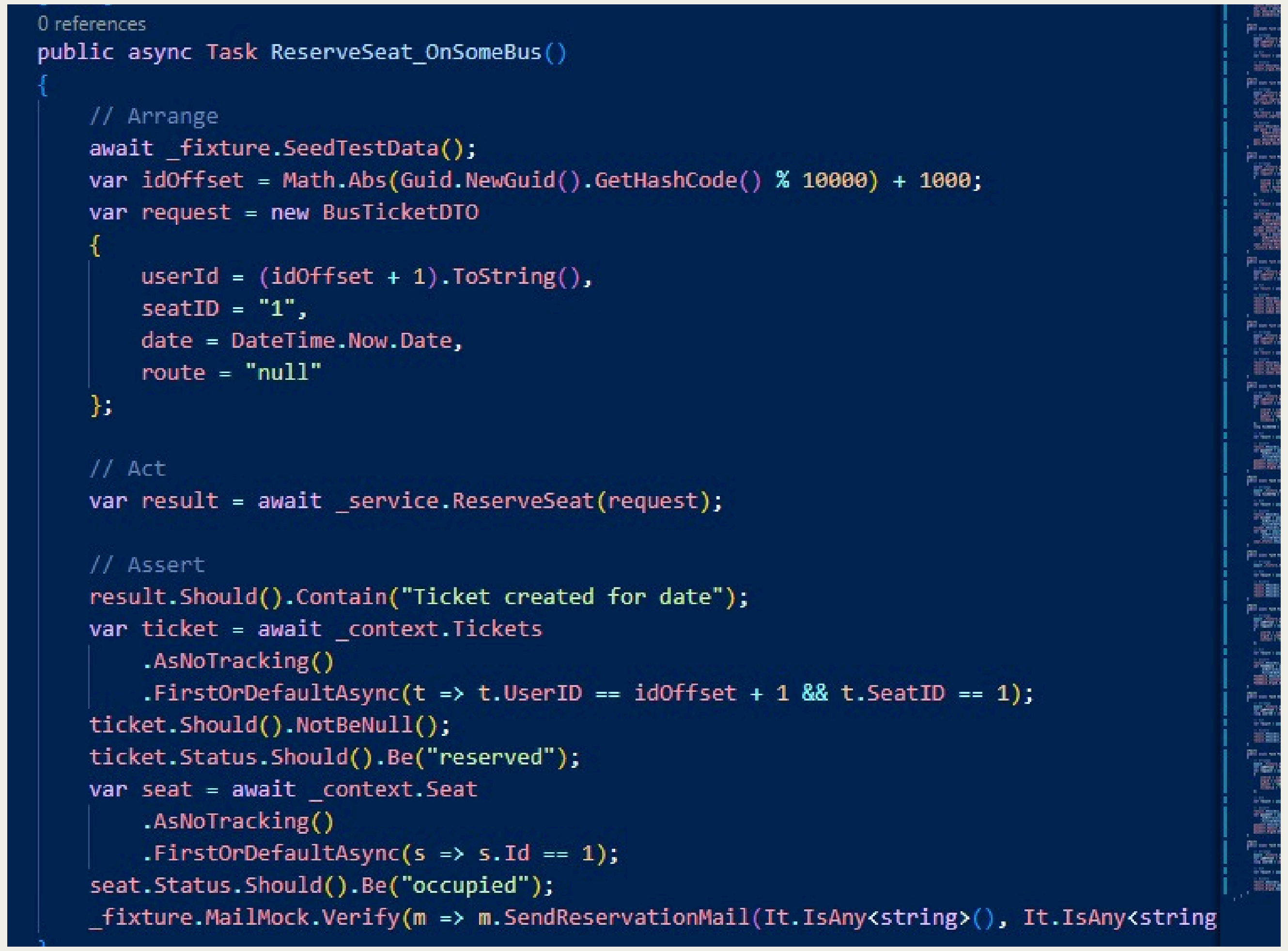
Done

- ✓ Title
- ✓ Problem Statement
- ✓ Envisioned Features
- ✓ User Stories (26)
- ✓ Roles and Communication
- ✓ Make Assignment Doc

White Box Testing.

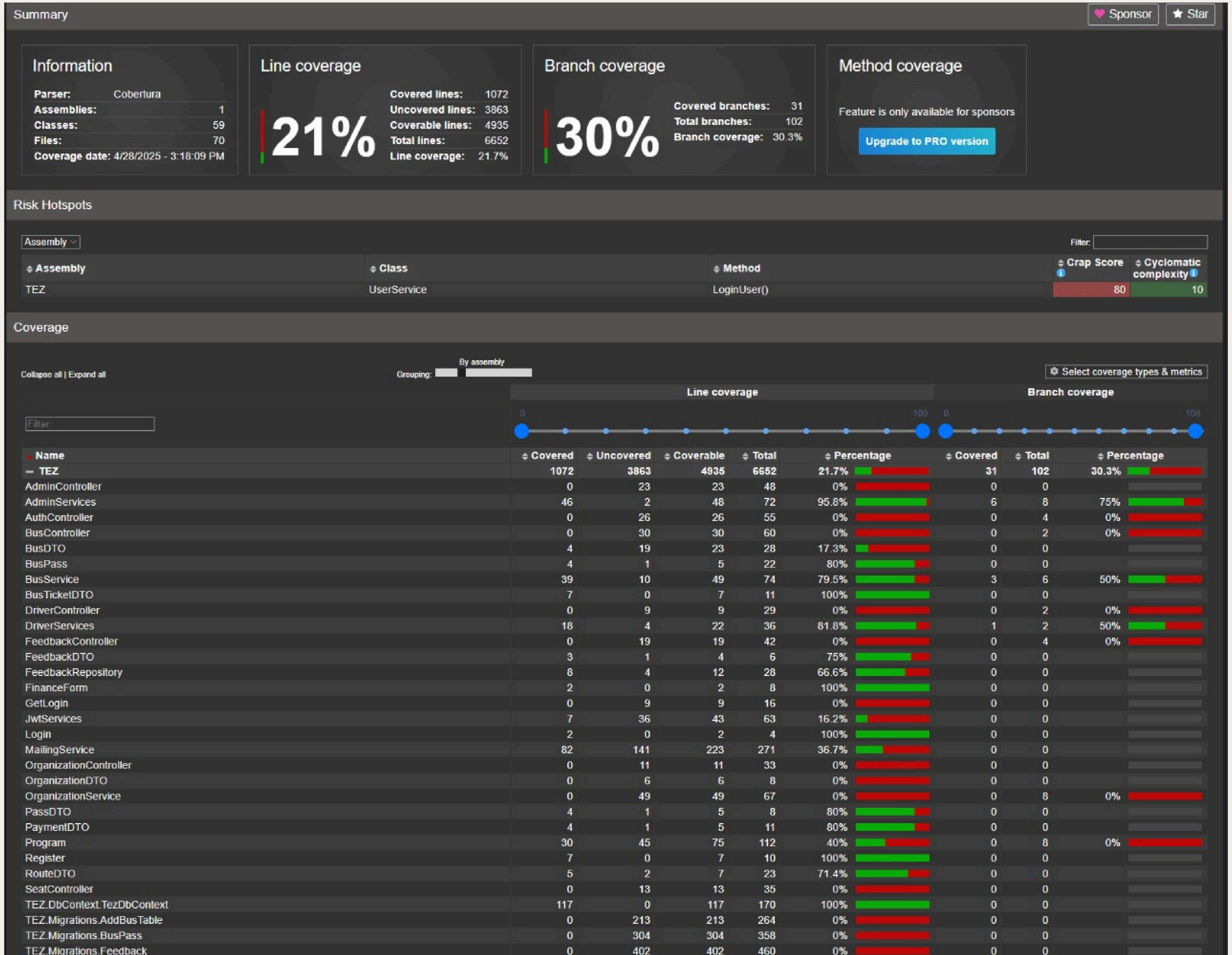


```
[Fact]
0 references
public async Task RegisterUserAsync_CreatesUserWithCorrectRole()
{
    // Arrange
    await _fixture.SeedTestData();
    var request = new Register
    {
        name = "New",
        email = "new@test.com",
        password = "pass",
        role = "STUDENT"
    };

    // Act
    var result = await _service.RegisterUserAsync(request);

    // Assert
    result.Should().Be("User Registered successfully");
    var user = await _fixture.UserRepository.GetByEmailAsync("new@test.com");
    user.Should().NotBeNull();
    user.As<User>().Role.Should().Be(Role.STUDENT);
}
```



```
0 references
public async Task ReserveSeat_OnSomeBus()
{
    // Arrange
    await _fixture.SeedTestData();
    var idOffset = Math.Abs(Guid.NewGuid().GetHashCode() % 10000) + 1000;
    var request = new BusTicketDTO
    {
        userId = (idOffset + 1).ToString(),
        seatID = "1",
        date = DateTime.Now.Date,
        route = "null"
    };

    // Act
    var result = await _service.ReserveSeat(request);

    // Assert
    result.Should().Contain("Ticket created for date");
    var ticket = await _context.Tickets
        .AsNoTracking()
        .FirstOrDefaultAsync(t => t.UserID == idOffset + 1 && t.SeatID == 1);
    ticket.Should().NotBeNull();
    ticket.Status.Should().Be("reserved");
    var seat = await _context.Seat
        .AsNoTracking()
        .FirstOrDefaultAsync(s => s.Id == 1);
    seat.Status.Should().Be("occupied");
    _fixture.MailMock.Verify(m => m.SendReservationMail(It.IsAny<string>(), It.IsAny<string
```

White Box Testing - Coverage



Information

Parser:	Cobertura
Assemblies:	1
Classes:	59
Files:	70
Coverage date:	4/28/2025 - 3:18:09 PM

Line coverage

21%

Covered lines:	1072
Uncovered lines:	3863
Coverable lines:	4935
Total lines:	6652
Line coverage:	21.7%

Branch coverage

30%

Covered branches:	31
Total branches:	102
Branch coverage:	30.3%

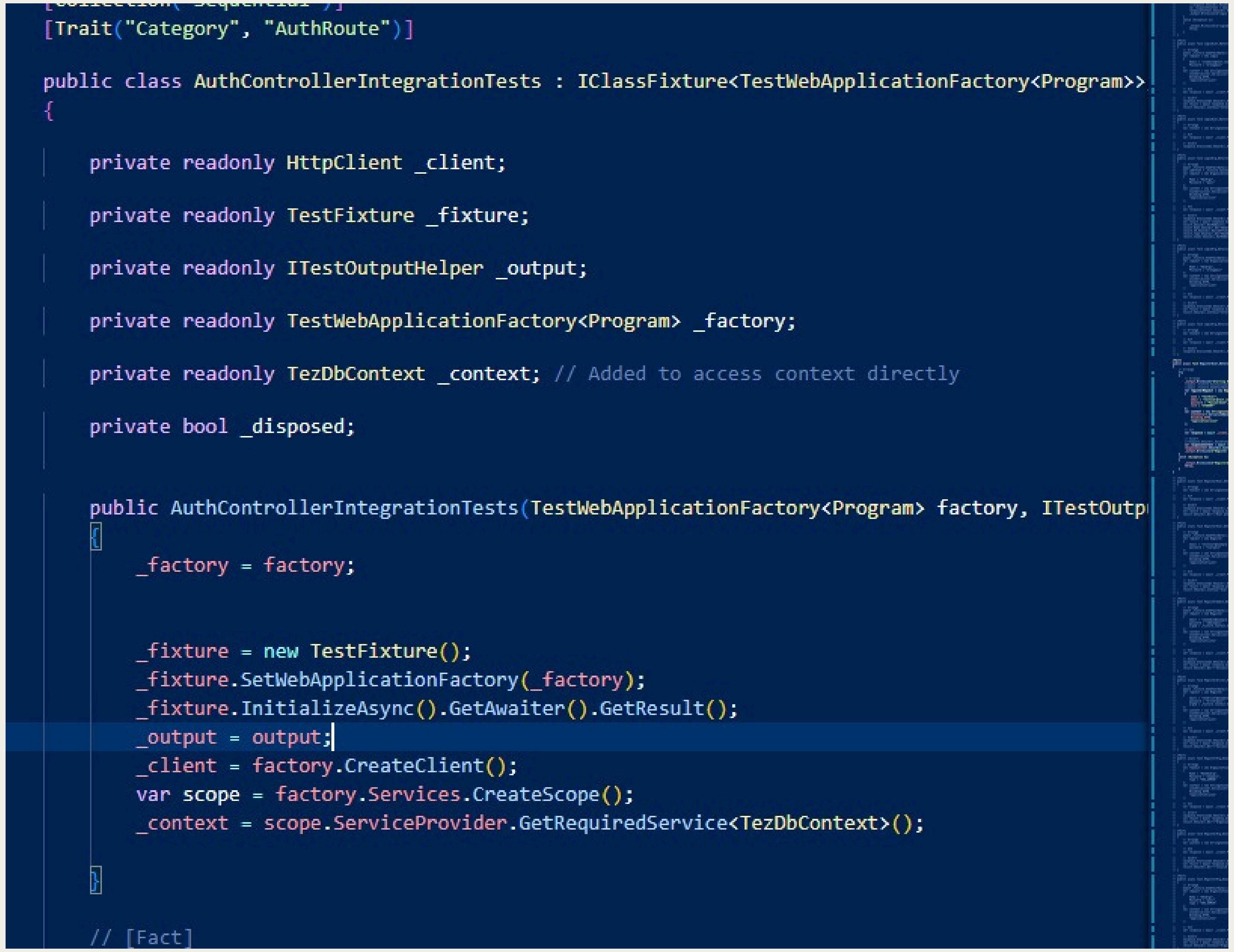
Method coverage

Feature is only available for sponsors

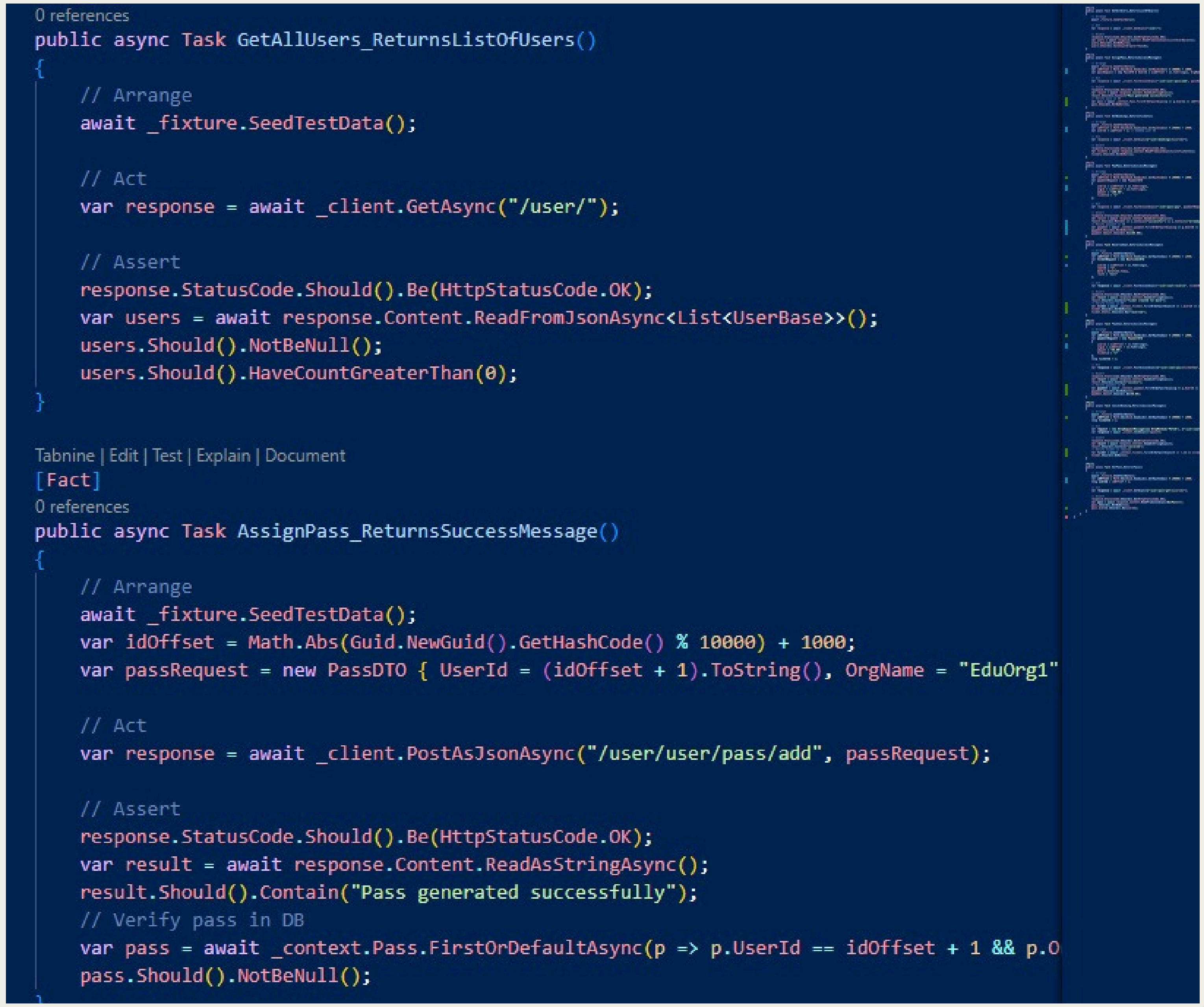
Assembly	Class	Method	Crap Score	Cyclomatic complexity
TEZ	UserService	LoginUser()	80	10

	Line coverage					Branch coverage		
Name	Covered	Uncovered	Coverable	Total	Percentage	Covered	Total	Percentage
— TEZ	1072	3863	4935	6652	21.7%	31	102	30.3%
AdminController	0	23	23	48	0%	0	0	
AdminServices	46	2	48	72	95.8%	6	8	75%
AuthController	0	26	26	55	0%	0	4	0%
BusController	0	30	30	60	0%	0	2	0%
BusDTO	4	19	23	28	17.3%	0	0	
BusPass	4	1	5	22	80%	0	0	
BusService	39	10	49	74	79.5%	3	6	50%
BusTicketDTO	7	0	7	11	100%	0	0	
DriverController	0	9	9	29	0%	0	2	0%
DriverServices	18	4	22	36	81.8%	1	2	50%
FeedbackController	0	19	19	42	0%	0	4	0%
FeedbackDTO	3	1	4	6	75%	0	0	
FeedbackRepository	8	4	12	28	66.6%	0	0	
FinanceForm	2	0	2	8	100%	0	0	
GetLogin	0	9	9	16	0%	0	0	
JwtServices	7	36	43	63	16.2%	0	0	
Login	2	0	2	4	100%	0	0	
MailingService	82	141	223	271	36.7%	0	0	
OrganizationController	0	11	11	33	0%	0	0	
OrganizationDTO	0	6	6	8	0%	0	0	
OrganizationService	0	49	49	67	0%	0	8	0%
PassDTO	4	1	5	8	80%	0	0	
PaymentDTO	4	1	5	11	80%	0	0	
Program	30	45	75	112	40%	0	8	0%
Register	7	0	7	10	100%	0	0	
RouteDTO	5	2	7	23	71.4%	0	0	
SeatController	0	13	13	35	0%	0	0	
TEZ.DbContext.TezDbContext	117	0	117	170	100%	0	0	
TEZ.Migrations.AddBusTable	0	213	213	264	0%	0	0	
TEZ.Migrations.BusPass	0	304	304	358	0%	0	0	
TEZ.Migrations.Feedback	0	402	402	460	0%	0	0	

Black Box Testing

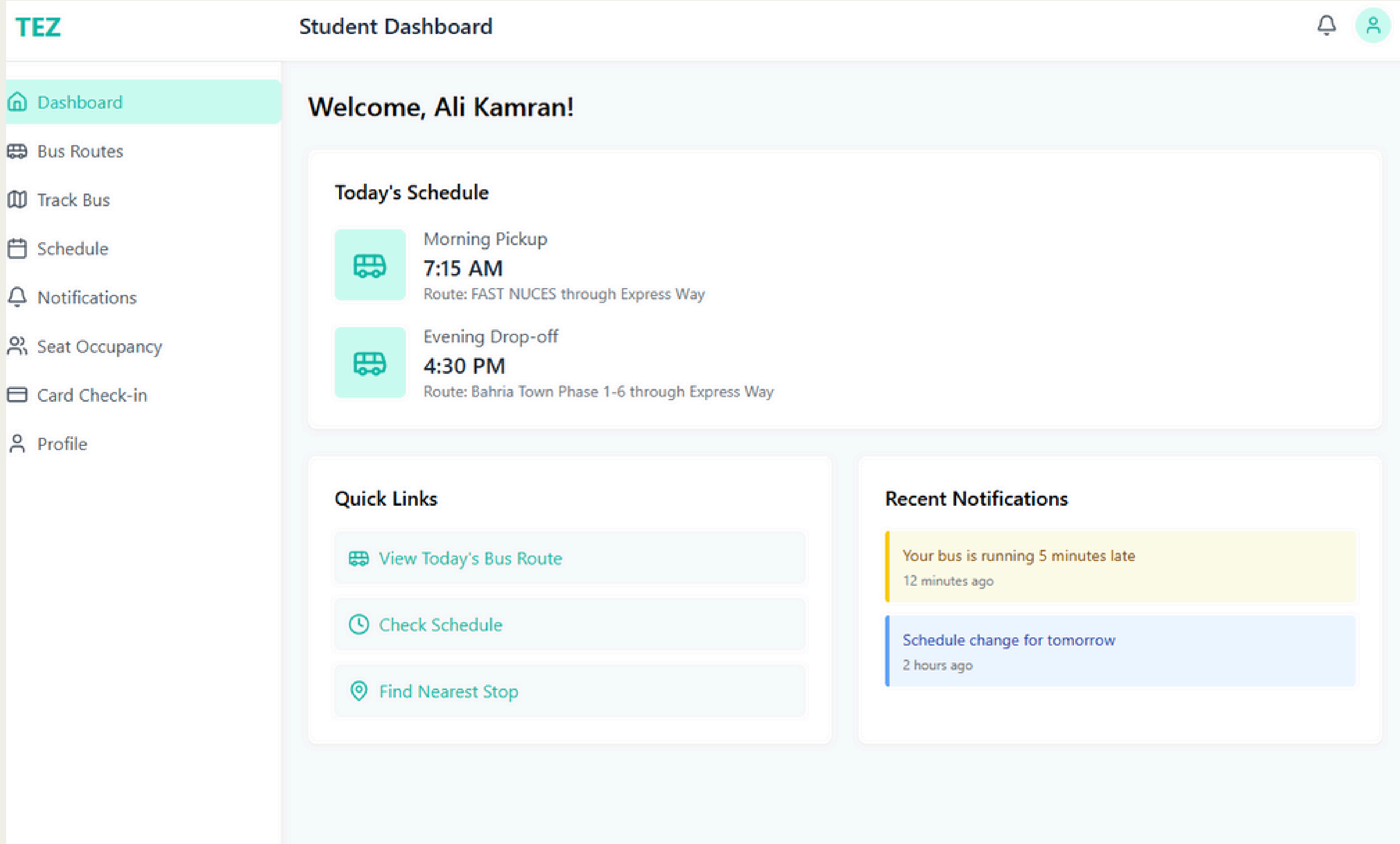


```
[Collection("Sequential")]
[Trait("Category", "AuthRoute")]

public class AuthControllerIntegrationTests : IClassFixture<TestWebApplicationFactory<Program>>
{

    private readonly HttpClient _client;

    private readonly TestFixture _fixture;

    private readonly ITestOutputHelper _output;

    private readonly TestWebApplicationFactory<Program> _factory;

    private readonly TezDbContext _context; // Added to access context directly

    private bool _disposed;


    public AuthControllerIntegrationTests(TestWebApplicationFactory<Program> factory, ITestOutp
    {
        _factory = factory;


        _fixture = new TestFixture();
        _fixture.SetWebApplicationFactory(_factory);
        _fixture.InitializeAsync().GetAwaiter().GetResult();
        _output = output;
        _client = factory.CreateClient();
        var scope = factory.Services.CreateScope();
        _context = scope.ServiceProvider.GetRequiredService<TezDbContext>();

    }

    // [Fact]
```



```
0 references
public async Task GetAllUsers_ReturnsListOfUsers()
{
    // Arrange
    await _fixture.SeedTestData();

    // Act
    var response = await _client.GetAsync("/user/");

    // Assert
    response.StatusCode.Should().Be(HttpStatusCode.OK);
    var users = await response.Content.ReadFromJsonAsync<List<UserBase>>();
    users.Should().NotBeNull();
    users.Should().HaveCountGreaterThan(0);
}

Tabnine | Edit | Test | Explain | Document
[Fact]
0 references
public async Task AssignPass_ReturnsSuccessMessage()
{
    // Arrange
    await _fixture.SeedTestData();
    var idOffset = Math.Abs(Guid.NewGuid().GetHashCode() % 10000) + 1000;
    var passRequest = new PassDTO { UserId = (idOffset + 1).ToString(), OrgName = "EduOrg1"

    // Act
    var response = await _client.PostAsJsonAsync("/user/user/pass/add", passRequest);

    // Assert
    response.StatusCode.Should().Be(HttpStatusCode.OK);
    var result = await response.Content.ReadAsStringAsync();
    result.Should().Contain("Pass generated successfully");
    // Verify pass in DB
    var pass = await _context.Pass.FirstOrDefaultAsync(p => p.UserId == idOffset + 1 && p.O
    pass.Should().NotBeNull();
}
```

Design and Implementation



TEZ

Student Dashboard

Welcome, Ali Kamran!

Today's Schedule

Morning Pickup
7:15 AM
Route: FAST NUCES through Express Way

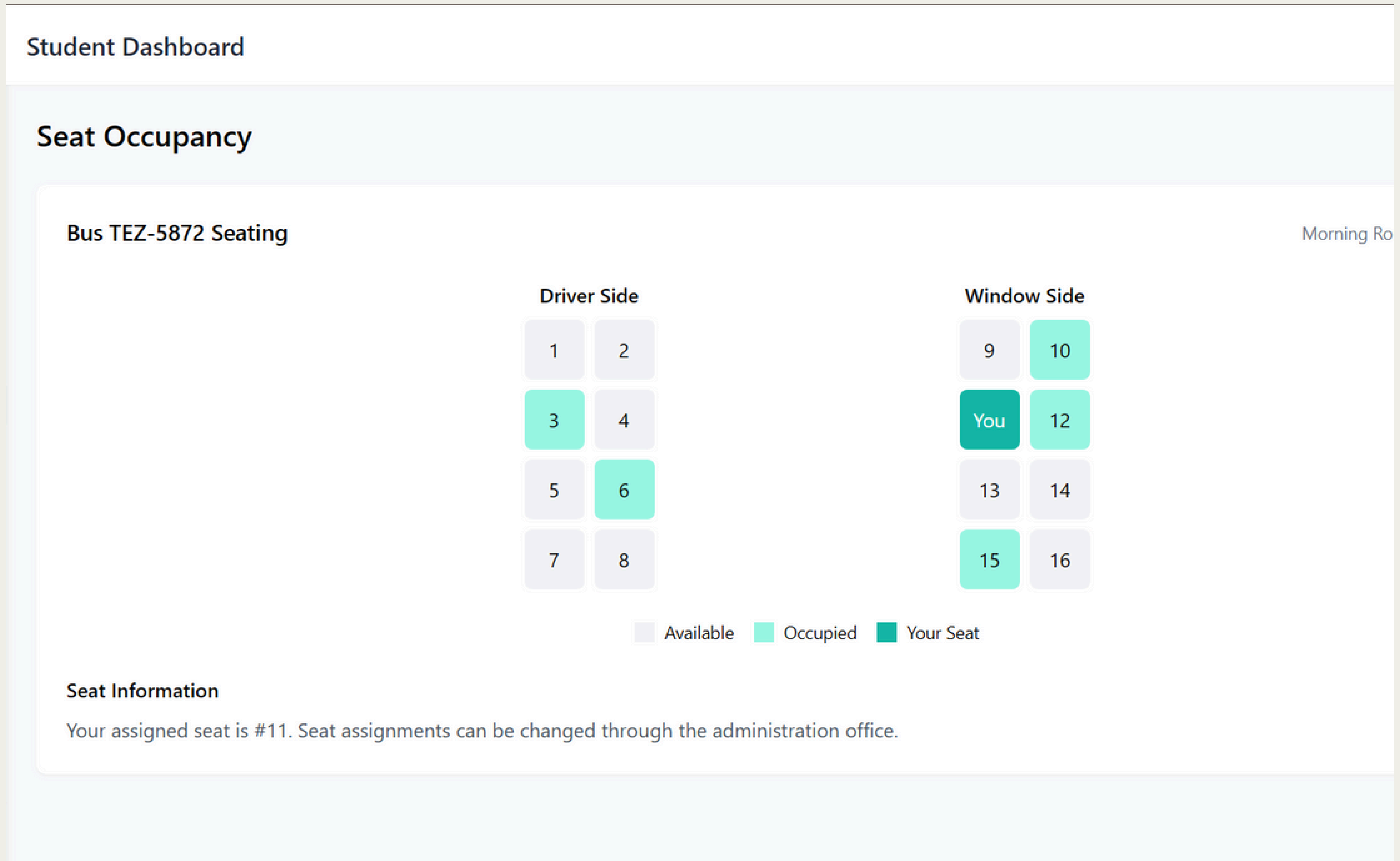
Evening Drop-off
4:30 PM
Route: Bahria Town Phase 1-6 through Express Way

Quick Links

Recent Notifications

Your bus is running 5 minutes late
12 minutes ago

Schedule change for tomorrow
2 hours ago



Student Dashboard

Seat Occupancy

Bus TEZ-5872 Seating

Morning Ro

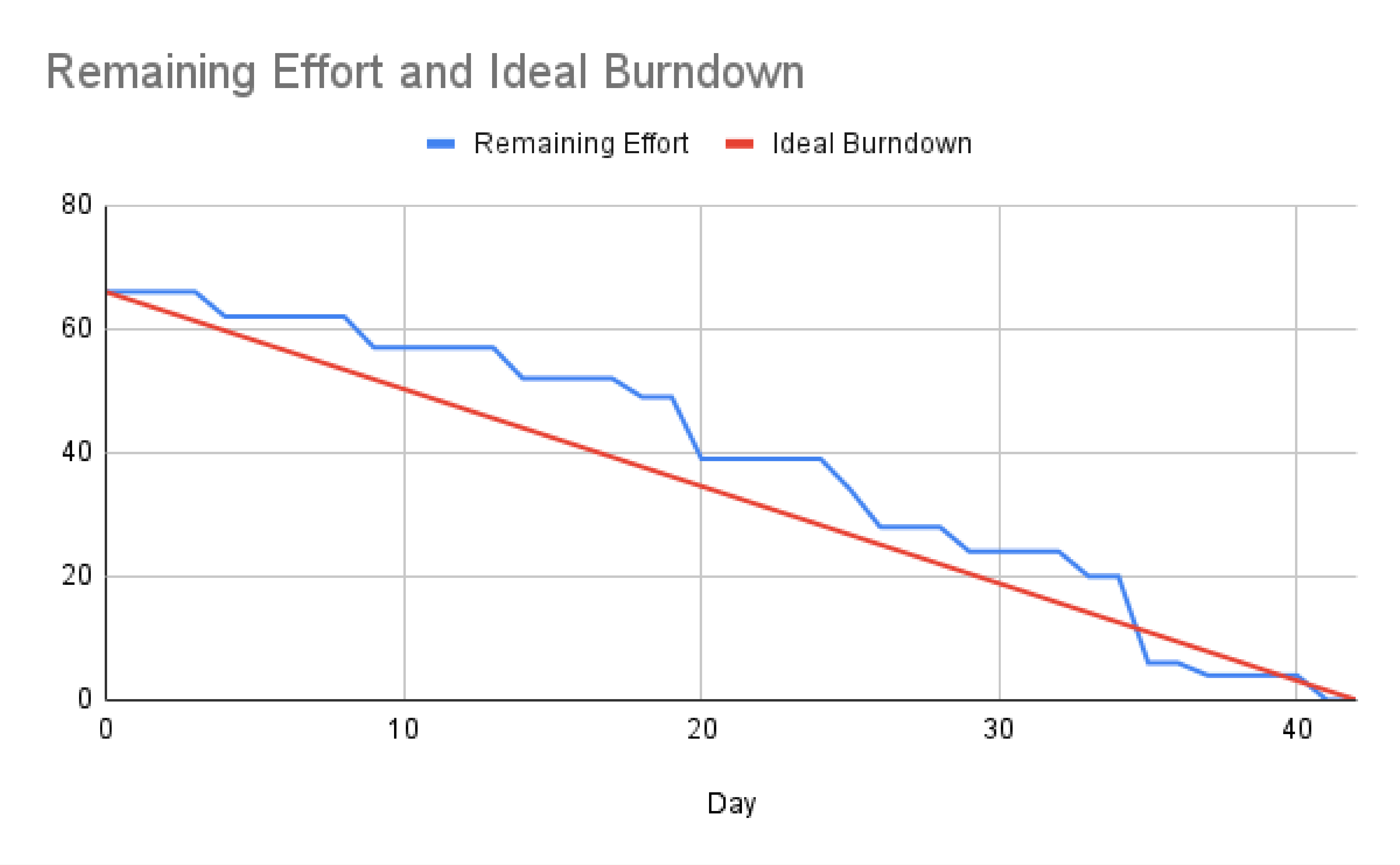
Driver Side			Window Side	
1	2		9	10
3	4		You	12
5	6		13	14
7	8		15	16

□ Available ▨ Occupied ■ Your Seat

Seat Information

Your assigned seat is #11. Seat assignments can be changed through the administration office.

Burndown Chart



Further Improvements

- AI Route Optimization

- Digital Bus Pass

- Driver Rating

Lessons Learnt

- **Scrum Model**
  - Sprints, Daily stand-up, Agile environment
- **Team Work**
  - Task Distribution, Time management, Communication
- **Importance of Testing**

Thank you!

QUESTIONS?

molarmuaz.github.io/tez