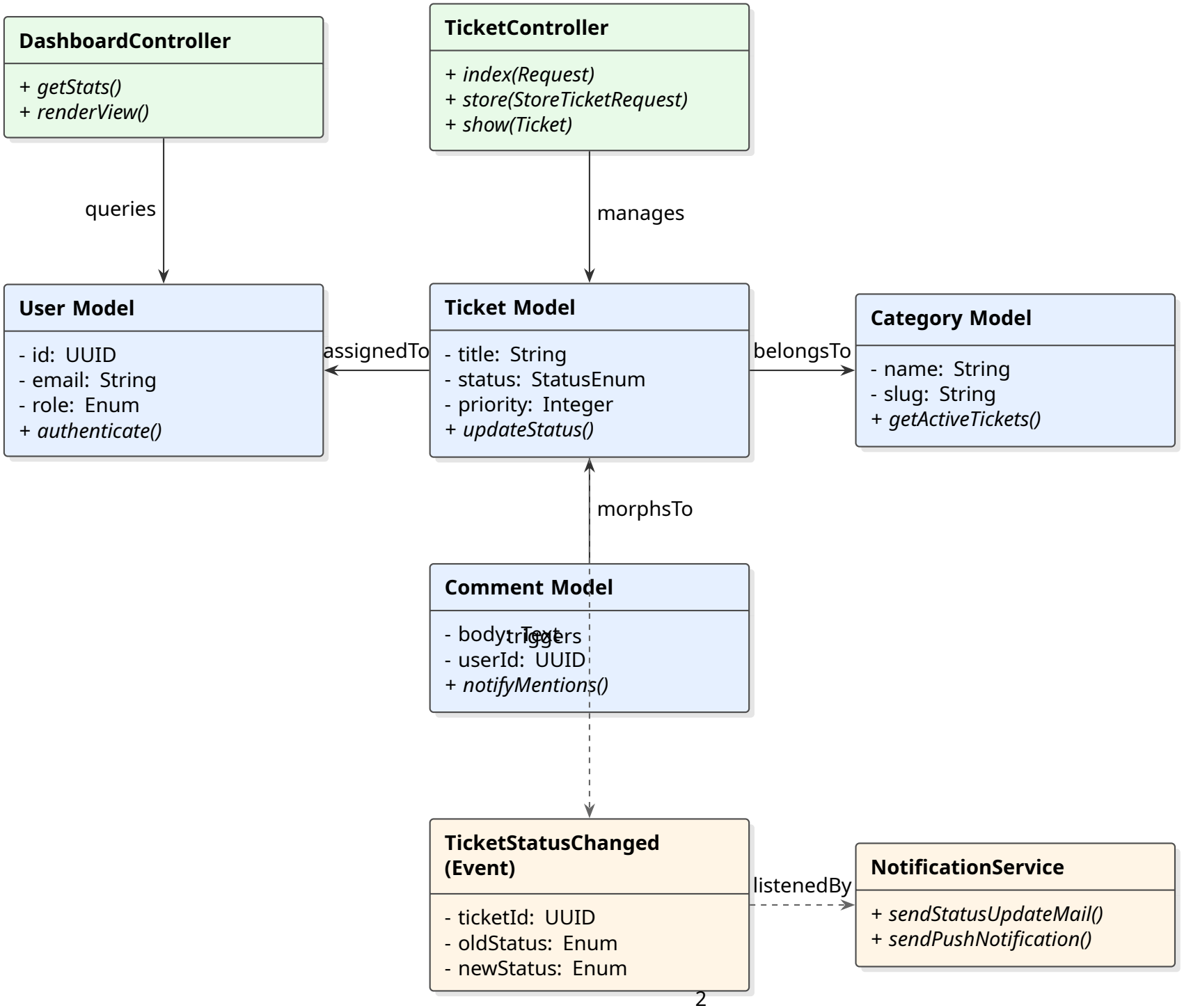
System Architecture: MVC Design Pattern Implementation

Class Relationship and Data Flow Diagram


DashboardController
+ getStats()
+ renderView()
TicketController
+ index(Request)
+ store(StoreTicketRequest)
+ show(Ticket)
queries
manages
User Model
- id: UUID
- email: String
- role: Enum
+ authenticate()
assignedTo
Ticket Model
- title: String
- status: StatusEnum
- priority: Integer
+ updateStatus()
belongsTo
Category Model
- name: String
- slug: String
+ getActiveTickets()
morphsTo
Comment Model
- body: Text
- userId: UUID
+ notifyMentions()
triggers
TicketStatusChanged (Event)
- ticketId: UUID
- oldStatus: Enum
- newStatus: Enum
listenedBy
NotificationService
+ sendStatusUpdateMail()
+ sendPushNotification()

Legend:

- Models (Database Entities)
- Controllers (Business Logic)
- Communication (Events/Mail)

Architectural Notes:

- Controllers act as the entry point for HTTP requests.
- Models represent the data layer and maintain integrity.
- Asynchronous events decouple notifications from logic.