Product Design
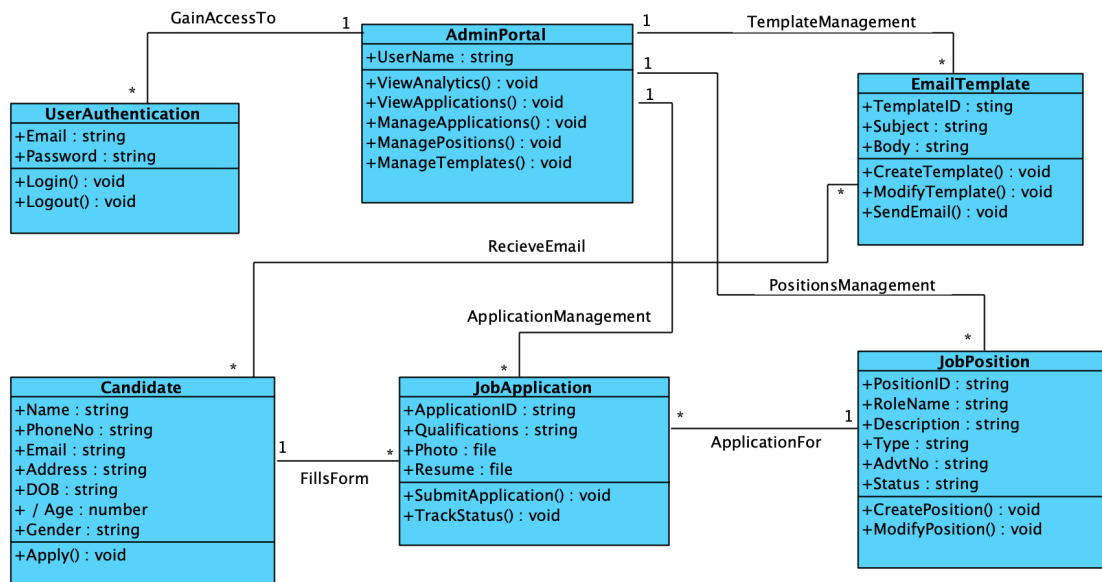
Team 8 – Recruitment Portal

Members:
Gaurav Behera (2022111004)
Amey Karan (2022111026)
D Venkata Vishnu Sai Reddy (2023121009)
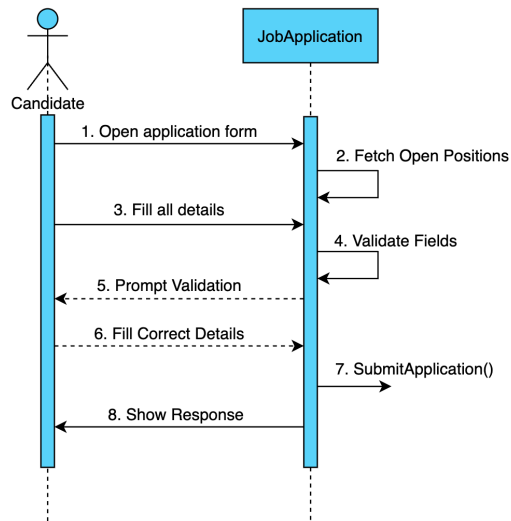**Chitturi Meghana Sai Rama Lakshmi (2021101006)**

Design Model

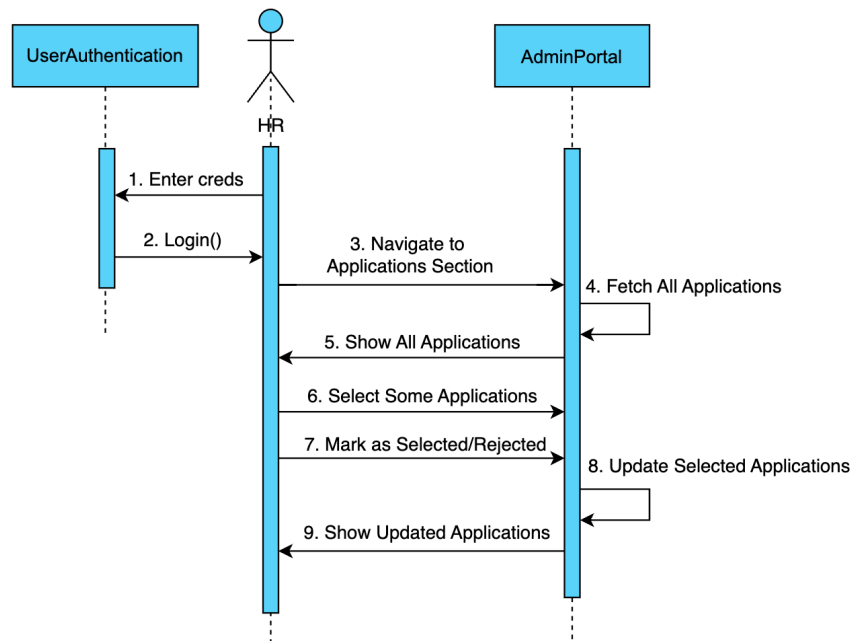| Candidate	Class state
	• Name
	• PhoneNo
	• Email
	• Address
	• DOB
	• Age
	• Gender
	Class behavior
	• Apply()
UserAuthentication (HR or Department	Class state
	• Email

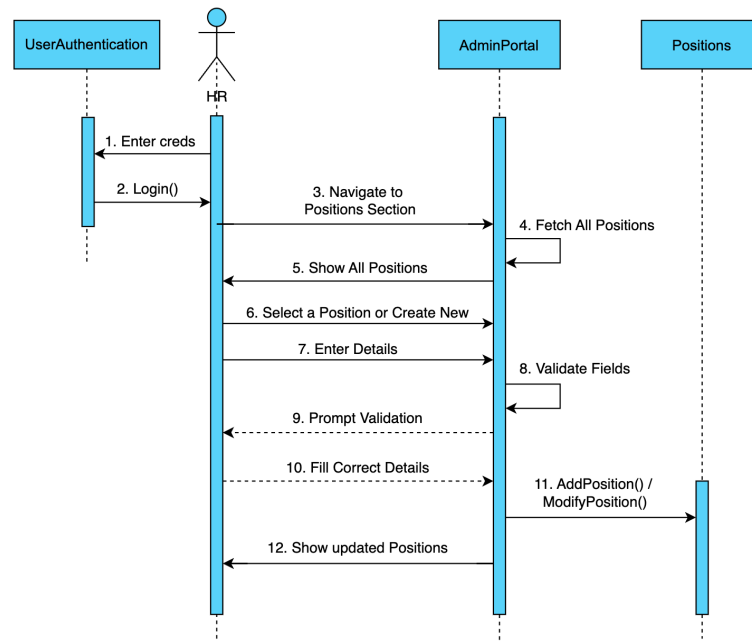|	
Heads)	• Password Class behavior • Login() • Logout()
JobApplication (Candidate Application Form Responses)	Class state • ApplicationID • Qualifications (ReasonToHire, PastExp, TechExp) • Photo • Resume Class behavior • SubmitApplication() • TrackStatus()
JobPosition (Positions for which candidates can apply)	Class state • PositionID • RoleName • Description (Requirements, Responsibilities, Salary) • Type • AdvtNo • Status Class behavior • CreatePosition() • ModifyPosition()
EmailTemplate (Templates for emails to be sent to candidates by the HR)	Class state • TemplateID • Subject • Body Class behavior • CreateTemplate() • ModifyTemplate() • SendEmail()
AdminPortal (For recuiters – HR and Department Heads)	Class state • UserName Class behavior • ViewAnalytics() • ViewApplications() • ManageApplications() • ManagePositions() • ManageTemplates()

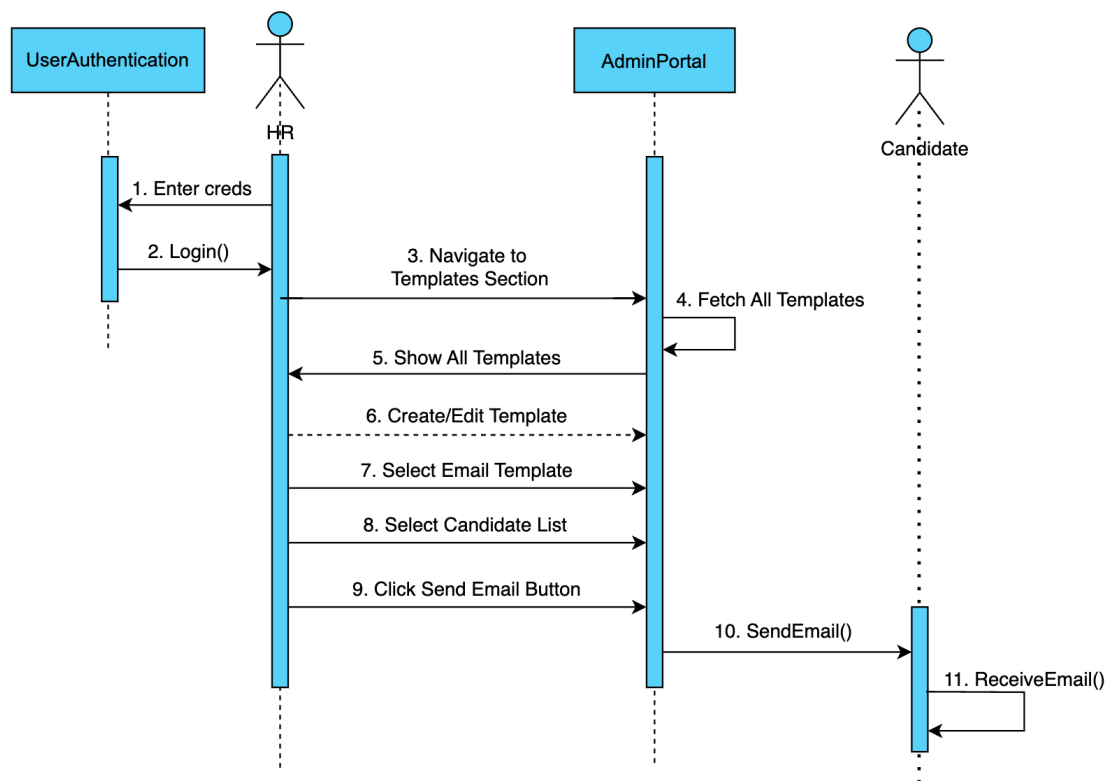# Sequence Diagram(s)

1. Candidate Submits Application Form
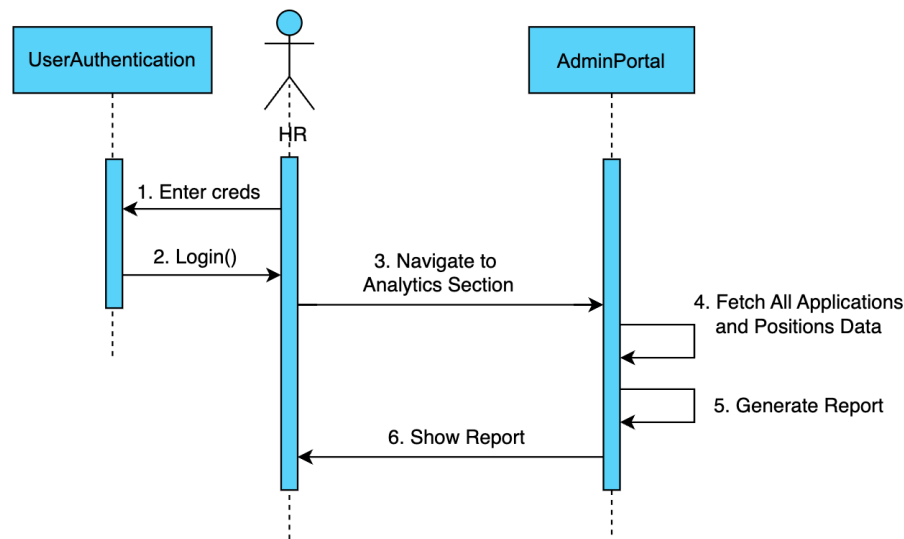

2. Candidate Management and Application Tracking

3. Position Management

4. Template Management and Sending Email

5. View Analytics

Design Rationale

- **Database Choice**: Selected MongoDB, a NoSQL database, due to its flexible schema which suits the dynamic nature of job applications and user profiles. This choice prioritizes scalability and flexibility over ACID transactions and structured relational query capabilities typical in SQL databases.

- **User Authentication**: Implemented CAS for its high security, especially since the admin portal is exclusively for IIITH employees. Although CAS offers strong security, it may lack flexibility and customization compared to a custom solution, ultimately reducing development effort.

- **Email Service Integration**: Chose a third-party API service over a custom SMTP solution for reliability, scalability, and pre-built templates. While this decision streamlines development and ensures reliability, it sacrifices some customization and flexibility.

- **Frontend Choice**: Selected React.js due to its component-based architecture and extensive features, making it well-suited for scalable and maintainable web applications. However, React.js may require additional libraries and has a steeper learning curve compared to Angular/Vue.