CHAPTER 3: SYSTEM REQUIREMENTS STUDY

3.1 USER CHARACTERISTICS

The CanmoreAI platform is designed for a range of users involved in proposal generation. This includes:

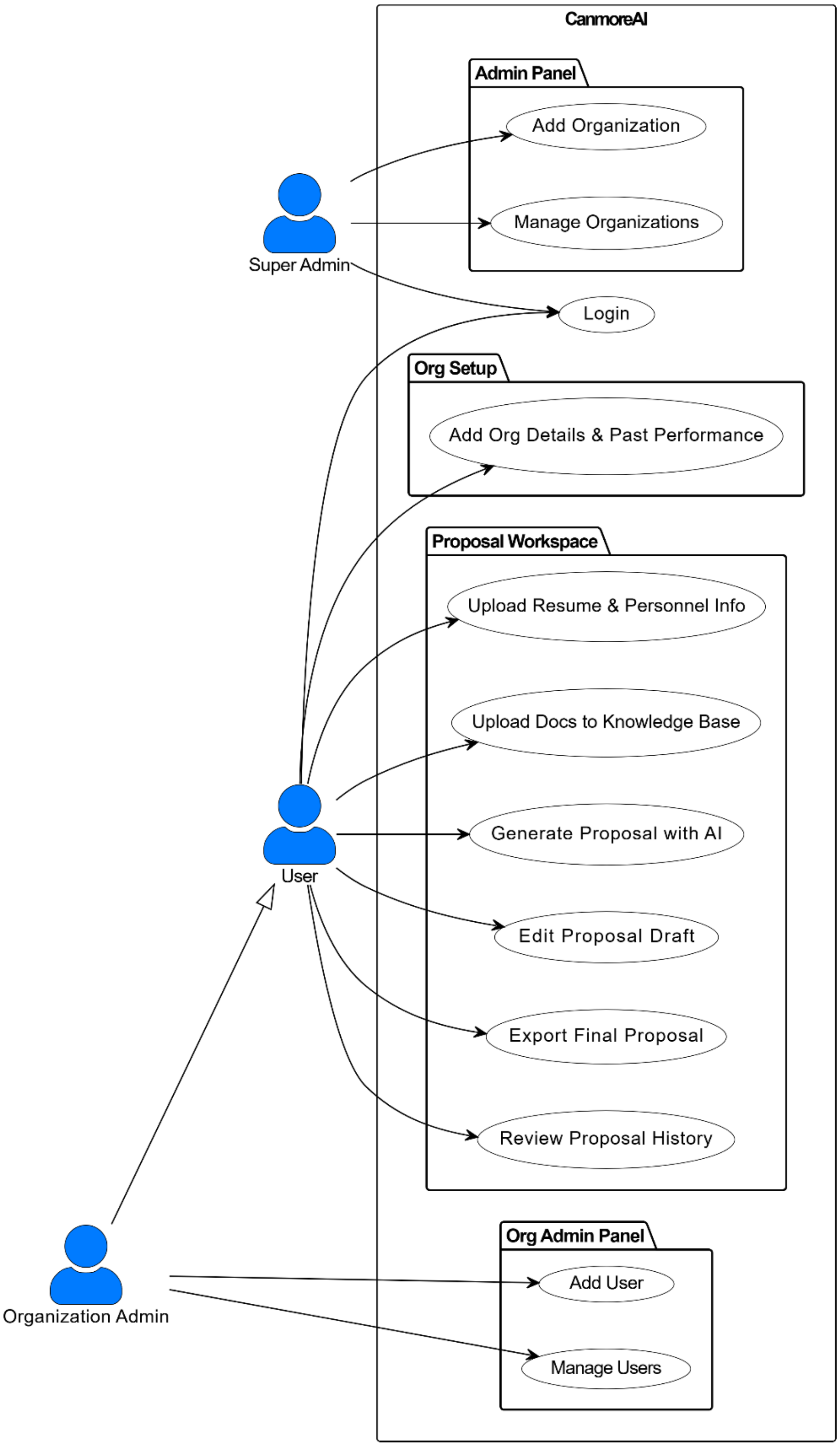

Fig 3.1 Use Case Diagram

- Proposal Writers: These users are responsible for drafting and editing proposals. They interact the most with the AI-generated content.
- Managers/Admins: These users oversee proposal quality, manage access control, and monitor submission deadlines.
- Subject Matter Experts (SMEs): Occasionally involved to validate technical or domain-specific content.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

3.2.1 Development Environment:

As mentioned in Table 3.1 and Table 3.2, the project is developed using a robust set of software and hardware specifications to ensure optimal performance and seamless integration. The software stack includes Windows 10/11 as the operating system, Python with Flask for backend development, and PostgreSQL as the database.

The frontend is built using ReactJS and TypeScript, while the AI/ML capabilities are powered by Amazon Bedrock, LangChain, and Pinecone. Cloud deployment utilizes Amazon Web Services (AWS), specifically AWS Lambda, Elastic Container Service (ECS), and Amazon RDS. Version control is managed through Git and GitHub, and development tools such as VS Code, Postman, and Jupyter Notebook support the development process.

On the hardware side, the system runs on devices equipped with Intel Core i5/i7 processors, 16GB RAM, and 512GB SSD, ensuring sufficient resources for efficient computation and storage.

Table 3.1 Software Specification for Development

Name of component	Specification
Operating System	Windows 10/11
Frontend	ReactJS, TypeScript
Backend	Python, Flask, PostgreSQL
AI/ML	Amazon Bedrock, LangChain, PGvector
Cloud Service	AWS (AWS Lambda, Elastic Container Service (ECS), Amazon RDS)

Table 3.2 Hardware Specification for Development

Name of component	Specification
Processor	Intel Core i5/i7
RAM	16GB
Storage	512GB SSD

3.2.2 Execution/Hosting Environment:

As presented in Table 3.3 and Table 3.4, the execution and hosting environment for the project is entirely cloud-based, leveraging the powerful infrastructure of Amazon Web Services (AWS).

The software specifications include AWS Lambda and Elastic Container Service (ECS) for compute services, Amazon RDS (PostgreSQL) for database management, and Amazon Bedrock, LangChain, and Pinecone for AI-driven functionalities. Additionally, Amazon S3 is used for object storage, while Amazon ElastiCache supports high-speed in-memory caching for enhanced application performance.

On the hardware side, the system is hosted on AWS EC2 Instances, which are high-performance virtual machines designed to handle intensive workloads. Furthermore, AWS Elastic Load Balancer and Auto Scaling mechanisms are employed to ensure the application remains highly scalable, reliable, and responsive under varying user demands.

Table 3.3 Software Specification for Execcution

Name of component	Specification
Cloud Platform	Amazon Web Services(AWS
Compute Services	AWS Lambda, Elastic Container Service (ECS)
Database	Amazon RDS (PostgreSQL)
AI Services	Amazon Bedrock, LangChain
Storage & Caching	Amazon S3, Amazon ElastiCache

Table 3.4 Hardware Specification for Execcution

Name of component	Specification
AWS EC2 Instances	High-performance virtual machines used to host and run application components

AWS Elastic Load Balancer	Distributes incoming application traffic across multiple EC2 instances for load balancing and fault tolerance
AWS Auto Scaling Mechanism	Automatically adjusts the number of EC2 instances based on traffic/load conditions to ensure scalability and reliability

3.3 ASSUMPTIONS AND CONSTRAINTS

3.3.1 Assumptions

- Users will provide high-quality contextual data (e.g., resumes, project details).
- AWS services such as Bedrock and S3 are available and accessible with proper credentials.
- Users have basic digital literacy and can navigate web forms and file upload interfaces.

3.3.2 Constraints

- Dependence on AWS services limits offline usage.
- Budgetary constraints restrict the use of paid AI models.
- Some foundational models may not fully understand highly specialized domains without fine-tuning.