ASHESI UNIVERSITY

Sprint Planning

Career Services On Finger Tips (CSOFT)

Jemima N. Antwi

Ronelle H. Cudjoe

Nana K. Amoako

Edward O. Mensah

Kelvin K. Ahiakpor

CS414: Software Engineering

Dr. Umut Tosun

May 2, 2025

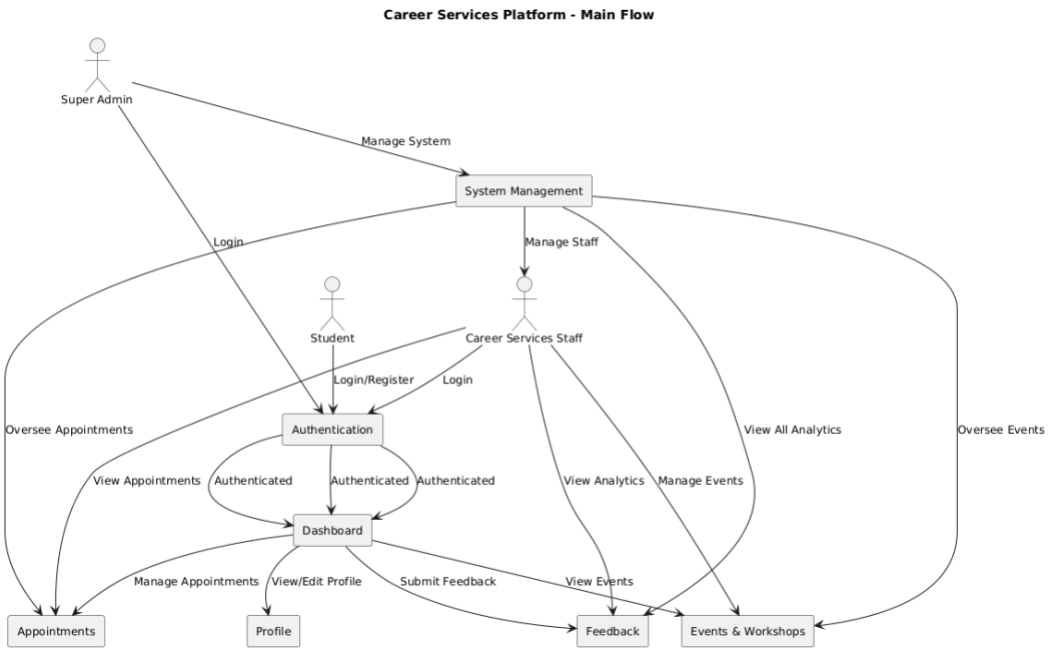
Career Services On Finger Tips (CSOFT)

Project Overview

Career Services On Finger Tips (CSOFT) is a web application designed to streamline career service operations, automate attendance tracking, and simplify the management of internship requests and approvals. Through research and first-hand experience at Ashesi, we identified that many career services departments face challenges in managing their operations efficiently. Traditional methods of tracking student attendance, managing internship letter requests, and issuing these letters are often manual, time-consuming, and prone to errors.

Project Objectives

- Enhance operational efficiency of career services
- Improve student engagement and satisfaction
- Facilitate data-driven decision making
- Streamline the internship request process
- Provide comprehensive career resource management



Functional Requirements

FR01: User Management

The system should allow administrators to create, update, and delete user accounts for students and career advisors.

Requirements:

- **FR 1.1.1:** Administrators can create, update, and delete user accounts.
- **FR 1.1.2:** Role-based access control for system settings, internship requests, and approvals.

FR02: Attendance Tracking & Event Management

The platform should enable students to check in to career development sessions using a QR code or unique identifier, and allow administrators to create and manage events.

Requirements:

- **FR 1.2.1:** Students can check in to sessions using a QR code or unique identifier.
- **FR 1.2.2:** Career advisors can view and export attendance reports for students and sessions.
- **FR 1.2.3:** Administrators can create events and generate QR codes for attendance tracking.

FR03: Internship Letter Request Management

Students should be able to request internship letters through the platform. To be eligible, students must attend 3 events, provide feedback on them, and book a one-on-one session.

Requirements:

- **FR 1.3.1:** Students can submit internship request letters.

- **FR 1.3.2:** Career advisors can review, approve, or reject internship requests.
- **FR 1.3.3:** Automated notifications sent to students on request status updates.
- **FR 1.3.4:** System enforces eligibility requirements (3 events, feedback, one-on-one session).

FR04: Resume Review S ystem

The system should allow students to upload resumes and receive feedback from career advisors.

Requirements:

- **FR 1.4.1:** Students can upload resumes in PDF/DOCX formats.
- **FR 1.4.2:** Career advisors can annotate and provide feedback on resumes.
- **FR 1.4.3:** System tracks revision history and improvement metrics.

FR05: One-on-One Session Booking

The system should facilitate the scheduling of one-on-one sessions between students and career advisors.

Requirements:

- **FR 1.5.1:** Students can view advisor availability and book sessions.
- **FR 1.5.2:** Advisors can set their availability and manage bookings.
- **FR 1.5.3:** System sends reminders and confirmation emails.

FR06: Reporting and Analytics

The platform should provide dashboards for career advisors to view career session attendance, internship request trends, and approval timelines.

Requirements:

- **FR 1.6.1:** Dashboards display attendance rates, request trends, and timelines.
- **FR 1.6.2:** Reports exportable in PDF or Excel formats.

FR07: Notifications and Reminders

Automated email or in-app notifications for students about career sessions, pending requests, and approval statuses.

Requirements:

- **FR 1.7.1:** Students receive automated notifications on sessions, requests, and statuses.
- **FR 1.7.2:** Career advisors receive reminders for pending requests.

FR08: Resource Hub

Page with all the necessary resources like cover letter templates, resume templates, and others.

Requirements:

- **FR 1.8.1:** Students can download specific resources they need.
- **FR 1.8.2:** Students can request the addition of new resources, and career advisors can review these requests.

Sprint Schedule Overview

We are planning a total of seven sprints starting on February 3, 2025, and ending on April 16, 2025, to cover various parts of the software development lifecycle.

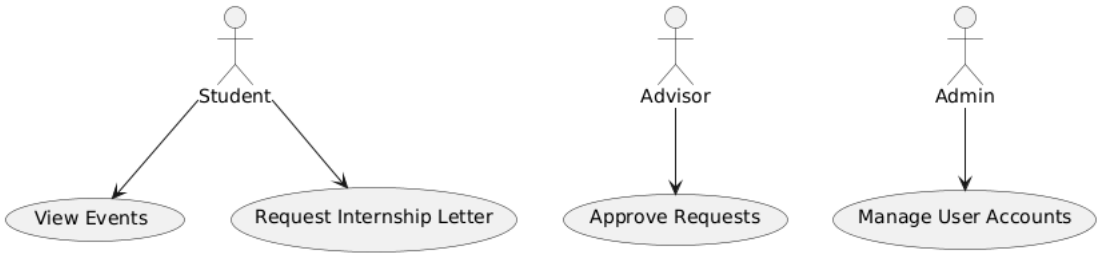
Sprint 1: Requirements Gathering & Project Setup

Duration: 2 weeks (Feb 3 - Feb 17, 2025)

Goal: Lay groundwork for development and establish both frontend and backend basics for parallel development.

User Stories:

- As a developer, I want to set up the development environment to begin coding efficiently.
- As a project manager, I want clear documentation of requirements to guide development.



Tasks:

1. **Infrastructure Setup**
    - Create GitHub repo with branches (main, dev, feature branches)
    - Set up CI/CD (GitHub Actions for automated testing)
2. **Documentation**
    - Draft Software Requirements Specification (SRS) with prioritized user stories

- Create Figma wireframes for key screens (login, booking, CV upload)

3. **Tools & Standards**
    - Agree on coding conventions (ESLint for frontend, JSDoc for documentation)
    - Define definition of done (DoD):
        - Code reviewed + passing tests
        - Documentation updated
4. **Frontend Setup**
    - Set up React + Tailwind CSS
    - Create initial wireframes for all roles (Student/Advisor/Admin)
5. **Backend Setup**
    - Database Design:
        - Normalized PostgreSQL schema (3NF)
        - Tables: users, appointments, internship_requests, attendance
        - Relationships: Foreign keys, indexes

Outputs:

- Approved SRS + wireframes
- Ready-to-code environment
- Refined backlog for Sprint 2

Sprint 2: UI/UX Design & Frontend Development

Duration: 2 weeks (Feb 17 - Mar 3, 2025)

Goal: Redesign pages, develop a theme, and iterate front-end pages for students.

User Stories:

- As a student, I want an intuitive interface to navigate career services.
- As an advisor, I want a dashboard to view student information efficiently.
- As an admin, I want a management panel to handle user accounts.

Tasks:

1. **Design System Development**
    - Create color palette, typography, and component library
    - Develop responsive design templates
2. **Core UI Components**
    - Build reusable UI components (buttons, forms, cards)
    - Implement responsive navigation system
3. **Role-Specific Dashboards**
    - Create dashboards for each role (Student, Advisor, Admin)
    - Implement mock login/auth flow (no real backend yet)

Outputs:

- Complete design system documentation
- Role-specific dashboard prototypes
- Component library with documentation

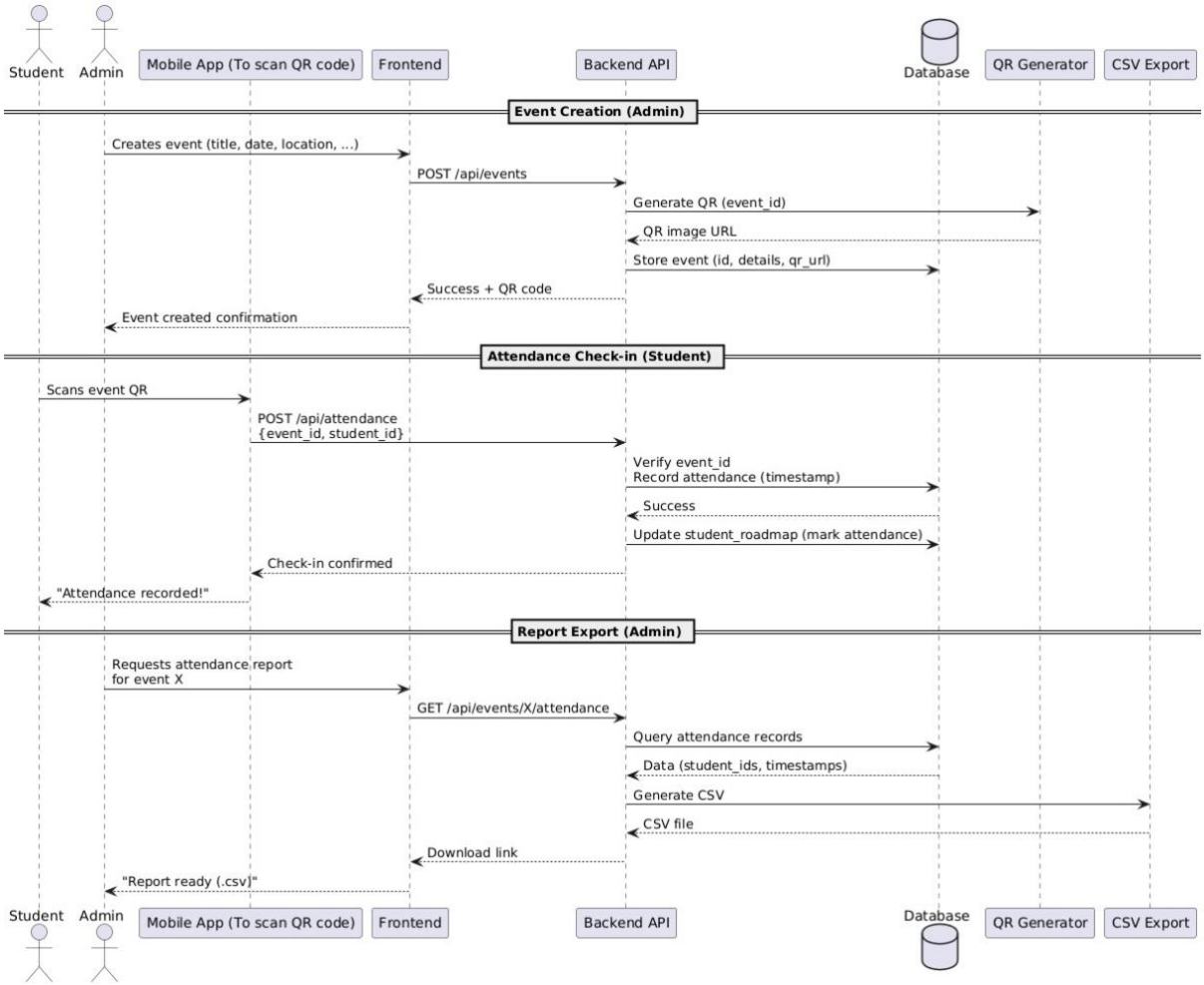
Sprint 3: Authentication & Attendance Tracking

Duration: 1 week (Mar 3 - Mar 10, 2025)

Goal: Implement authentication, security features, and attendance tracking functionality with backend integration.

User Stories:

- As a user, I want to securely log in to access my account.
- As an admin, I want to manage user roles and permissions.
- As an event organizer, I want to create events with QR codes for attendance.
- As a student, I want to check in to events using QR codes.



Tasks:

1. **Authentication System**
    - Set up Firebase authentication
    - Implement role-based access control
    - Design and implement login/registration pages
2. **Event Management**
    - Create event creation forms for administrators
    - Implement QR code generation for events
    - Build event listing and detail pages
3. **Attendance Tracking**
    - Develop QR code scanning functionality
    - Build attendance recording backend API
    - Create attendance reports and analytics

Outputs:

- Working authentication system with role-based access
- Event creation and management system
- QR code-based attendance tracking

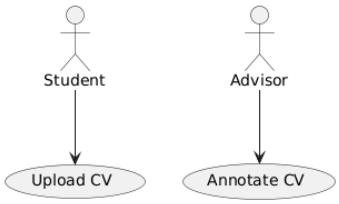
Sprint 4: CV Review & Annotation

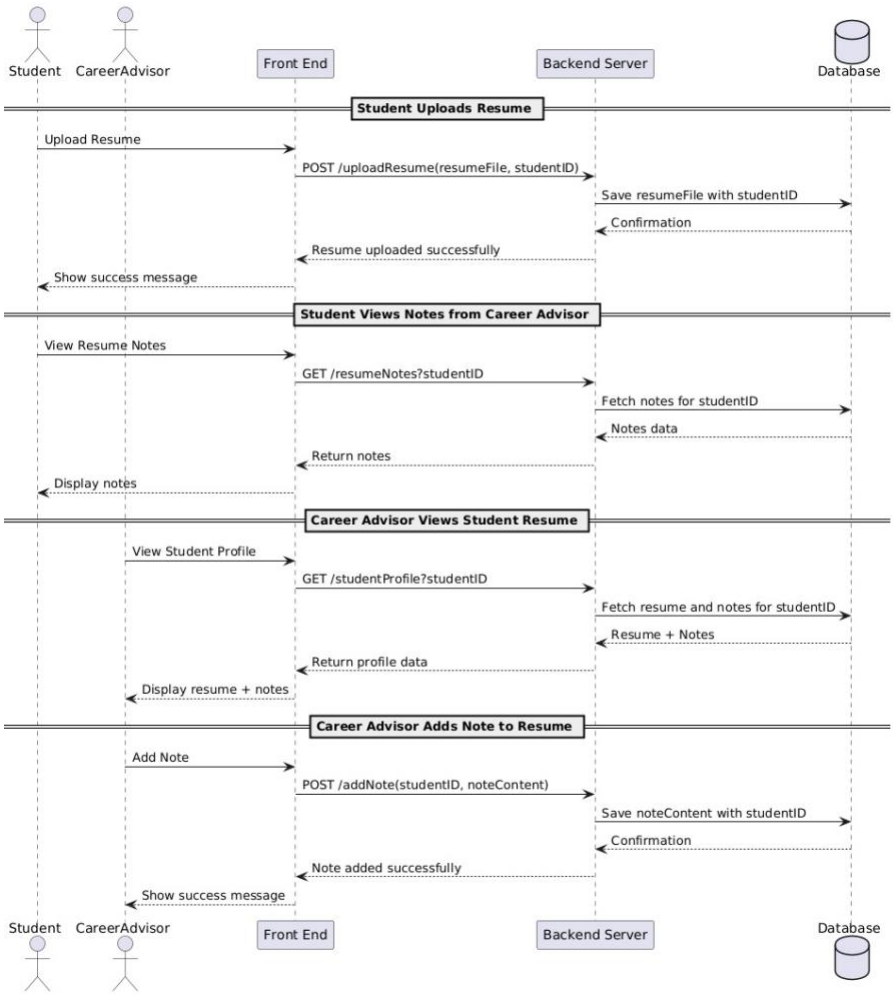
Duration: 1 week (Mar 10 - Mar 17, 2025)

Goal: Implement CV annotation and review functional requirements.

User Stories:

- As a student, I want to upload my CV for review.
- As an advisor, I want to annotate CVs with feedback.
- As a student, I want to receive AI-powered suggestions to improve my resume.





Tasks:

1. **CV Upload System**
    - Build CV upload form with drag-and-drop UI
    - Set up AWS S3 bucket for file storage
    - Implement file type and size validation
2. **Annotation Tool**
    - Develop PDF/DOCX viewer with annotation capabilities
    - Create comment and feedback system
    - Implement revision history tracking
3. **AI Integration**
    - Connect to HuggingFace API for skills gap analysis
    - Implement keyword optimization suggestions
    - Build AI feedback presentation interface

Outputs:

- Working CV upload and storage system
- Functional annotation tool for advisors
- AI-powered CV improvement suggestions

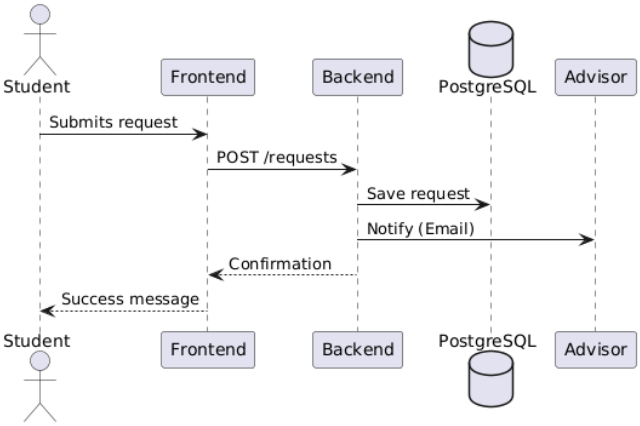
Sprint 5: Internship Letter Requests

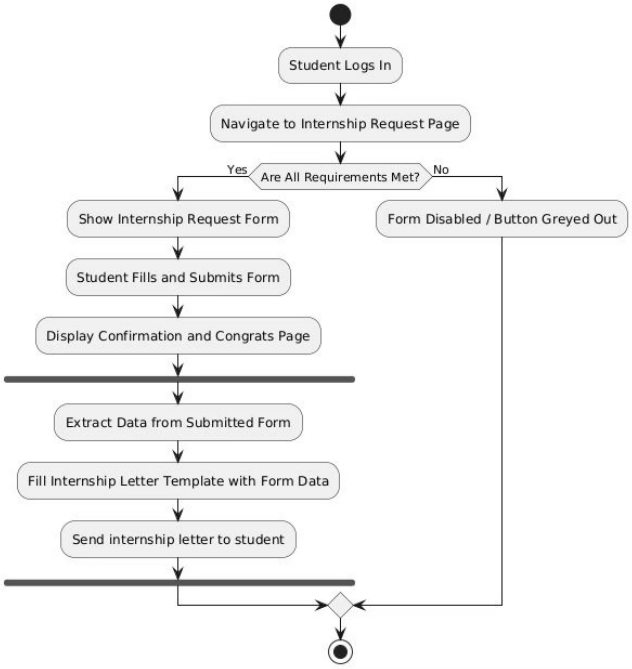
Duration: 1 week (Mar 11 - Mar 18, 2025)

Goal: Implement internship letter request and approval system.

User Stories:

- As a student, I want to request an internship letter if I meet the eligibility criteria.
- As an advisor, I want to review and approve/reject internship letter requests.
- As a student, I want to be notified about the status of my request.





Tasks:

1. **Request Form**
    - Build internship letter request form
    - Implement eligibility validation (3 events, feedback, one-on-one)
    - Create request tracking interface
2. **Approval Workflow**
    - Develop advisor review interface
    - Implement approval/rejection system
    - Build automated letter generation
3. **Notification System**
    - Set up email notifications for status changes
    - Implement in-app notifications
    - Create notification preferences settings

Outputs:

- Complete internship letter request workflow
- Automated eligibility checking system
- Working notification system for status updates

Sprint 6: Testing & Quality Assurance

Duration: 2 weeks (Mar 18 - Apr 1, 2025)

Goal: Ensure the reliability, performance, and security of the application.

User Stories:

- As a user, I want a bug-free experience with the application.
- As a developer, I want comprehensive test coverage to ensure code quality.
- As an admin, I want the system to be secure and performant.

Tasks:

1. **Unit Testing**
    - Write unit tests for frontend components
    - Develop backend API tests
    - Implement continuous integration testing
2. **Integration Testing**
    - Test end-to-end workflows across system components
    - Validate data flow between frontend and backend
    - Test cross-browser compatibility
3. **Security Testing**
    - Perform authentication and authorization tests
    - Implement CSRF protection
    - Conduct vulnerability assessment
4. **Performance Testing**
    - Test application under load
    - Optimize database queries
    - Improve frontend performance

Outputs:

- Comprehensive test suite with high coverage
- Security audit report with resolved vulnerabilities
- Performance optimization recommendations

Sprint 7: Deployment & Documentation

Duration: 2 weeks (Apr 2 - Apr 16, 2025)

Goal: Deploy the application to production and complete documentation.

User Stories:

- As a user, I want access to the live application.
- As a developer, I want clear documentation for future maintenance.
- As a new user, I want helpful user guides to navigate the system.

Tasks:

1. **Deployment Pipeline**
    - Set up Vercel deployment configuration
    - Configure environment variables
    - Implement CI/CD for automated deployments
2. **Technical Documentation**
    - Create API documentation
    - Document database schema
    - Develop system architecture documentation
3. **User Documentation**
    - Create user manuals for each role
    - Develop help center content
    - Record tutorial videos
4. **Final Testing**
    - Perform smoke testing in production environment
    - Validate all features in deployed application
    - Fix any deployment-specific issues

Outputs:

- Deployed application on Vercel
- Complete technical documentation
- Comprehensive user guides and tutorials

Risk Management

Risk	Impact	Probability	Mitigation
Technical complexity of cv comments	High	Low	Dedicated resources in Sprint 4
Integration issues between front and backend	Medium	High	Comprehensive API testing, integration testing in Sprint 6
Time constraints for all features	High	Medium	Prioritize core features, use MVP approach
Database performance issues	High	Low	Optimize queries, implement caching, regular performance testing

Conclusion

This sprint planning document outlines our approach to developing the Career Services On Finger Tips (CSOFT) application over seven sprints. By following this plan, we aim to deliver a high-quality system that meets the needs of students, career advisors, and administrators, while ensuring a smooth development process and timely delivery.