TECH-RESU-MAKE

Presented by
Sharon Galela

About Us

👋 Hi, I'm Sharon Galela

I'm a Junior Software Engineer passionate about building smart, efficient, and user-friendly software. With a strong foundation in Python, C++, Java, and SQL, I enjoy developing applications, automating tasks, and diving into data to extract valuable insights.

💻 **Technical Skills**
- Languages: Python, Java, C++, C#, SQL
- Frameworks & Tools: Java Swing, UiPath, ZOHO Creator (Deluge), Advanced Excel
- Areas of Interest: UI Development, Data Analysis, Automation, Technical Support

🚀 **Projects & Contributions**
- Carbon Footprint & BEE Analysis Calculator
- Designed UI and implemented logic using ZOHO Creator + Deluge with Python and Java integrations.

🌐 Websites Built (No-Code)
- Demisize Website
- Association for the Physically Disabled - Johannesburg

🎓 **Education**
- BSc in Information Technology – North West University (NQF 6 completed)
- BSc in Computing – UNISA (In Progress)

🏆 **Certifications**
- Golden Key Certificate (Top 15% Academic Achievement)
- UiPath Automation Explorer
- Data Analyst Certificate
- Facilitator and Assessor Certification

🔍 I'm always learning and open to collaborating on innovative, impact-driven projects. Let's build something great together!
📫 **n.sharongalela@gmail.com**

NON-FUNCTIONAL REQUIREMENTS

- The system must load resume previews and pages within 2 seconds for a smooth user experience.
- The application should support thousands of concurrent users without crashing (high scalability).
- All user data must be encrypted in transit (HTTPS) and at rest (e.g., database encryption).
- The platform should be available 99.9% of the time with minimal downtime.
- User data should never be lost – auto-save and backup mechanisms must be in place.
- The app must be responsive and work across all device sizes (mobile, tablet, desktop).
- The interface must be intuitive, simple to navigate, and beginner-friendly.
- The application should be accessible to users with disabilities (keyboard navigation, screen reader support).
- Code should be modular, well-documented, and easy to maintain or update in the future.
- The app must run smoothly on major browsers (Chrome, Firefox, Safari, Edge).
- Support for both web and mobile platforms must be consistent and seamless.
- Resume data must be automatically saved frequently to prevent accidental loss.
- The system must comply with local and international data privacy laws (e.g., POPIA, GDPR).
- The design should support future localization to allow for multi-language support.

Actors and Use Cases

🎭 Actors

- Tech Student / User – The main user who creates and customizes resumes.
- System – The application itself performing automatic tasks (e.g., scoring, autosave).
- (Optional) Resume Buyer / Client – A user who purchases or receives a resume design from another user.

📌 Use Cases

For Tech Student / User:

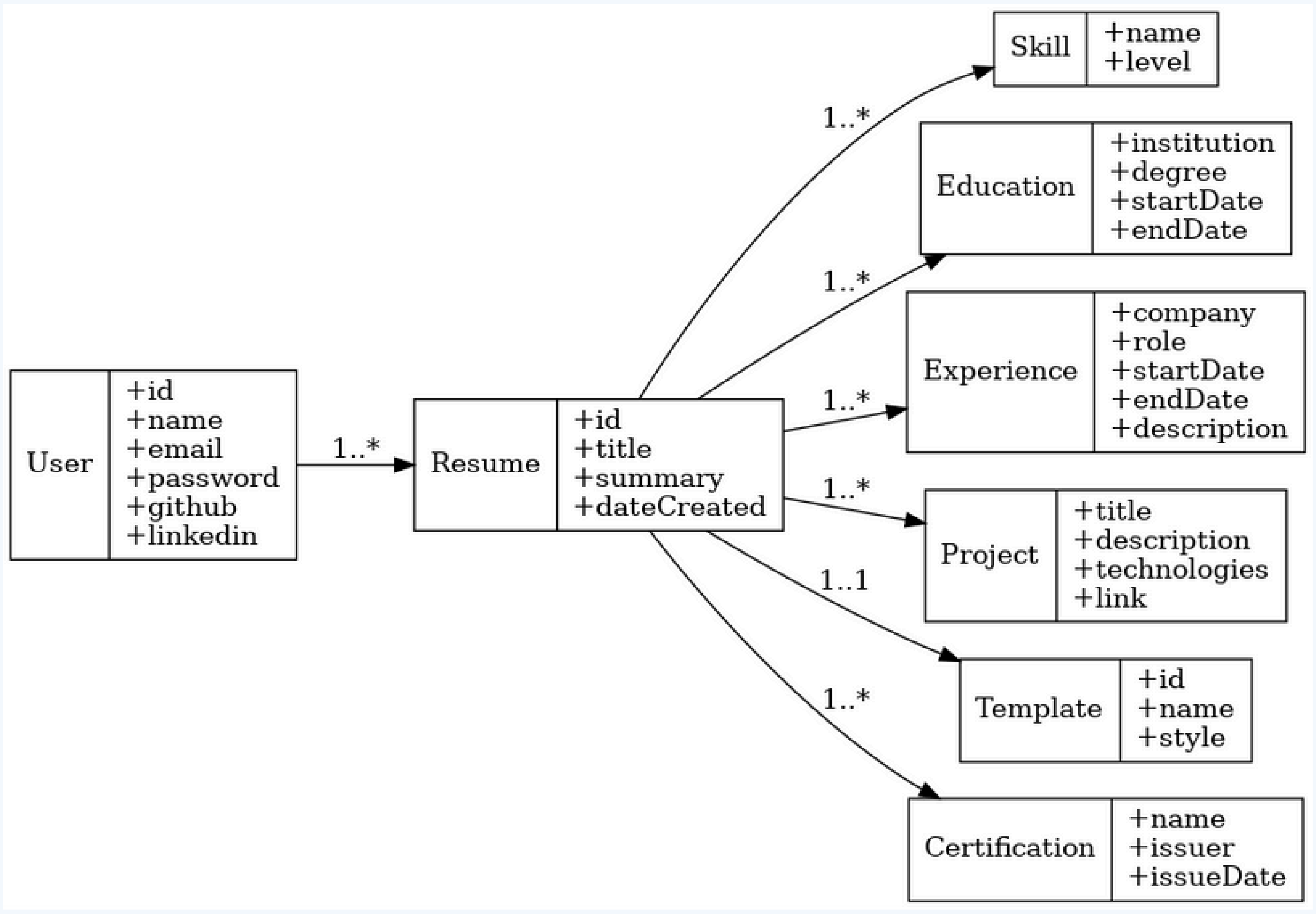
- Register / Log in / Log out
- Edit profile (add GitHub, LinkedIn, etc.)
- Create a new resume
- Add personal details
- Add technical skills
- Add education background
- Add work experience
- Add project details
- Add certifications
- Choose and apply a template
- Preview resume in real-time
- Export resume as PDF or Word
- Save resume (autosave or manual)
- Manage multiple resumes (duplicate, delete)
- View resume score and suggestions
- Offer resume services to others (optional)

For System:

- Score resume completeness automatically
- Provide suggestions for improvement
- Autosave resume data periodically
- Format resume into selected template
- Handle PDF/Word export
- Send email verification or password reset

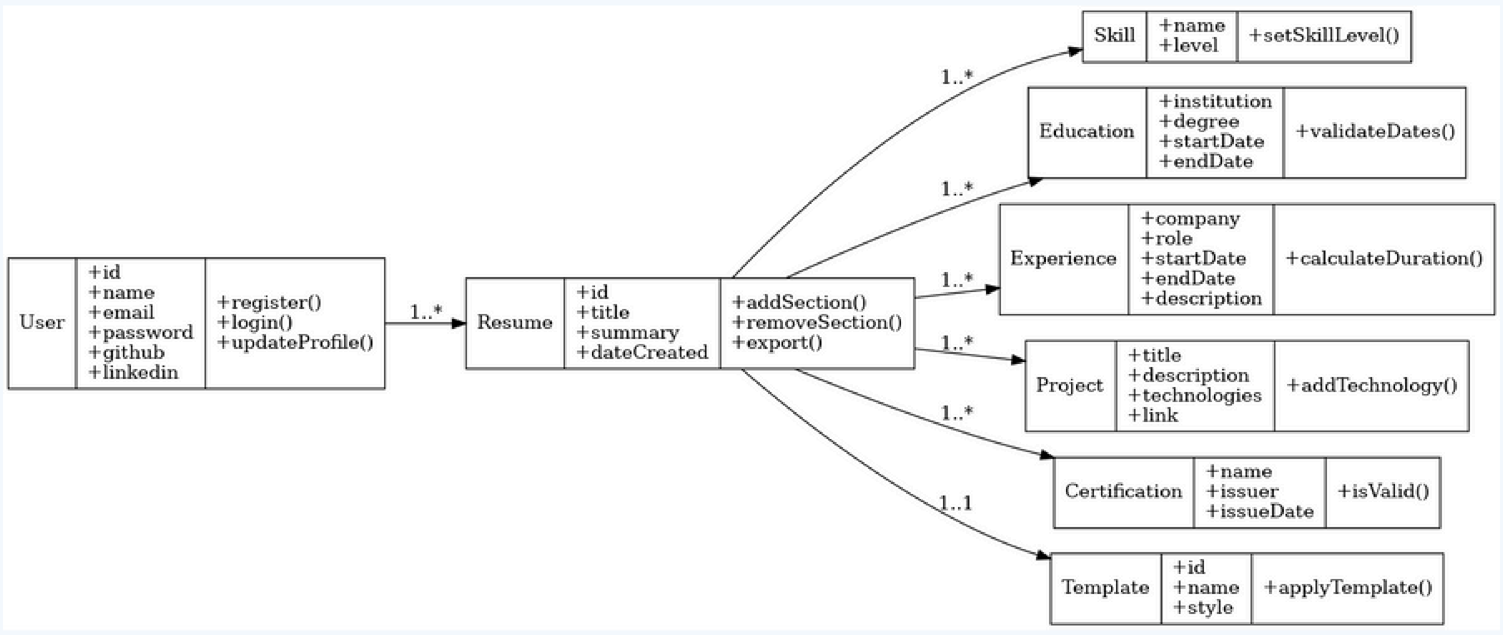
Optional: Resume Buyer / Client

- Browse or request resume templates or services
- Make payment for resume design (if monetization is enabled)
- Receive customized resume from designer


User
+id
+name
+email
+password
+github
+linkedin
1..*
Resume
+id
+title
+summary
+dateCreated
1..*
Skill
+name
+level
1..*
Education
+institution
+degree
+startDate
+endDate
1..*
Experience
+company
+role
+startDate
+endDate
+description
1..*
Project
+title
+description
+technologies
+link
1..1
Template
+id
+name
+style
1..*
Certification
+name
+issuer
+issueDate

Classes and Objects

Class Name	Description	Key Attributes	Key Responsibilities
User	Represents the tech student or user	id, name, email, password, GitHub, LinkedIn	Register, login, manage profile, manage resumes
Resume	A CV created by a user	id, title, summary, createdDate	Add/remove/edit sections, export resume
Skill	A technical skill listed on a resume	name, proficiencyLevel	Store skill name and level
Education	Educational background details	institution, degree, startDate, endDate	Store education info
Experience	Work experience entries	company, role, startDate, endDate, description	Store job history info
Project	Projects done by the user	title, description, technologies, link	Store project details
Certification	Certifications earned	name, issuer, issueDate	Store certification details
Template	Resume templates available	id, name, style	Provide formatting options



Class Relationships

1. User ↔ Resume
Relationship Type: Aggregation
Meaning: A user has one or more resumes. If the user is deleted, their resumes may also be deleted.
Cardinality: 1 User → * Resumes

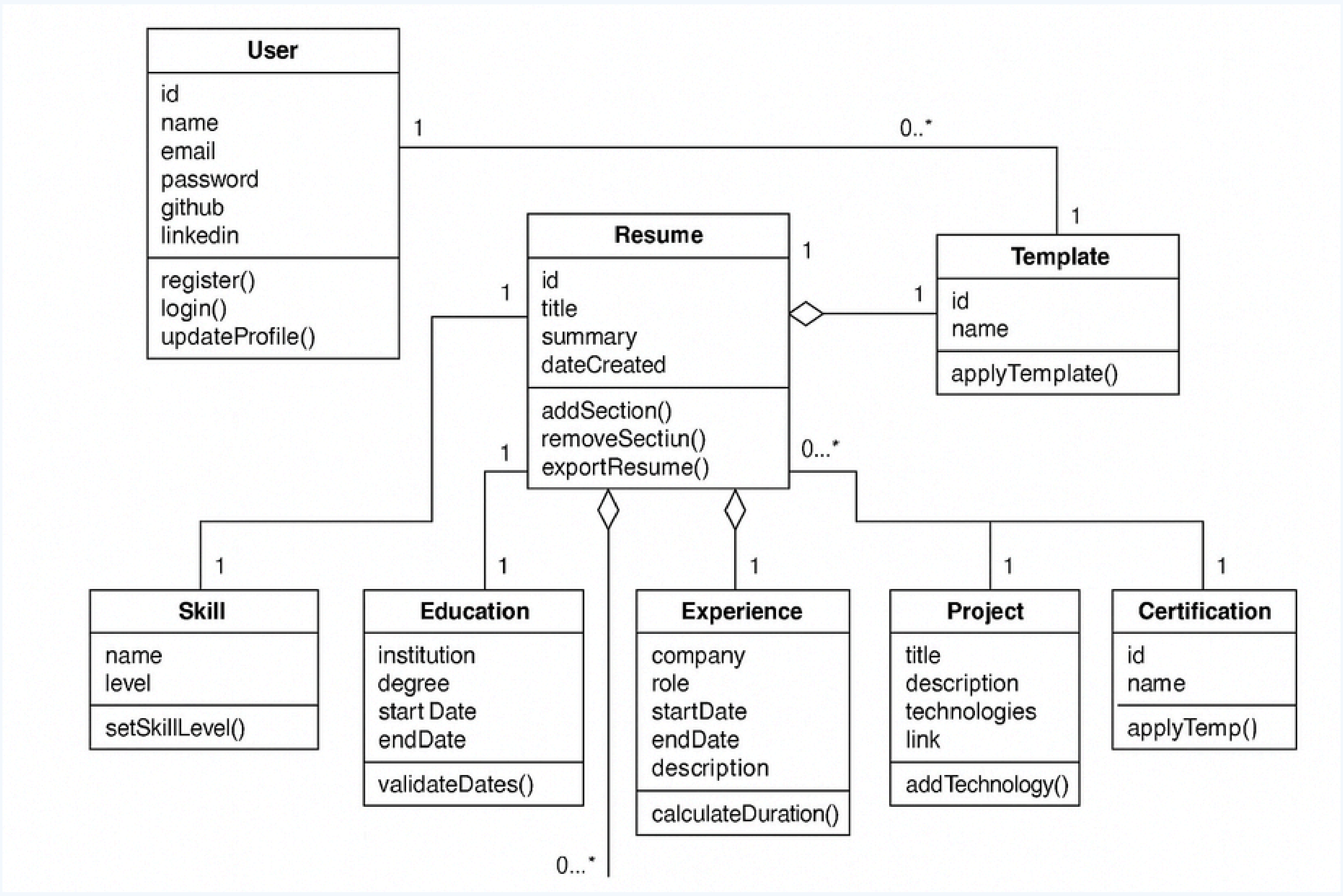
2. Resume ↔ Skill / Education / Experience / Project / Certification
Relationship Type: Composition
Meaning: A resume is composed of these sections. If the resume is deleted, the sections are deleted too.
Cardinality: 1 Resume → * Items (Skills, Education, etc.)

3. Resume ↔ Template
Relationship Type: Association
Meaning: A resume uses a particular template but the template exists independently.
Cardinality: 1 Resume → 1 Template

4. (Optional) Inheritance (if you add roles in future)
Admin might inherit from User
ResumeSection could be a base class for Skill, Education, etc.

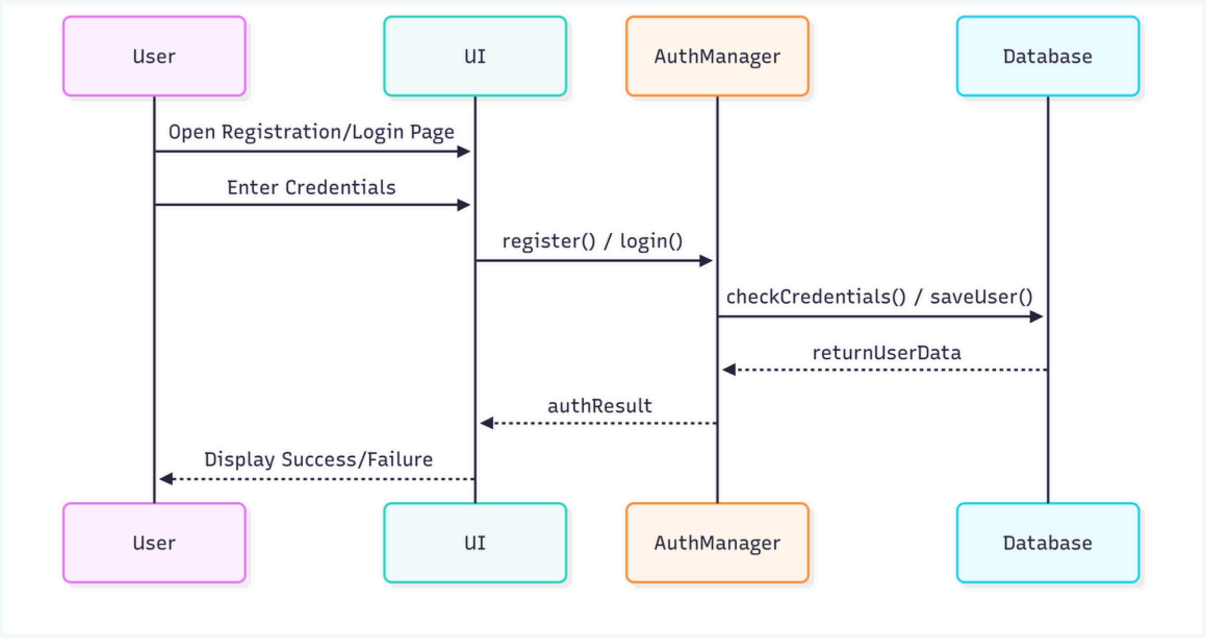
System Design (Detailed Design)

Class	Attributes	Methods
User	id: String, name: String, email: String, password: String	register(), login(), updateProfile()
Resume	id: String, title: String, summary: String, dateCreated: Date	addSection(), removeSection(), exportResume()
Skill	name: String, level: String	setSkillLevel()
Education	institution: String, degree: String, startDate: Date, endDate: Date	validateDates()

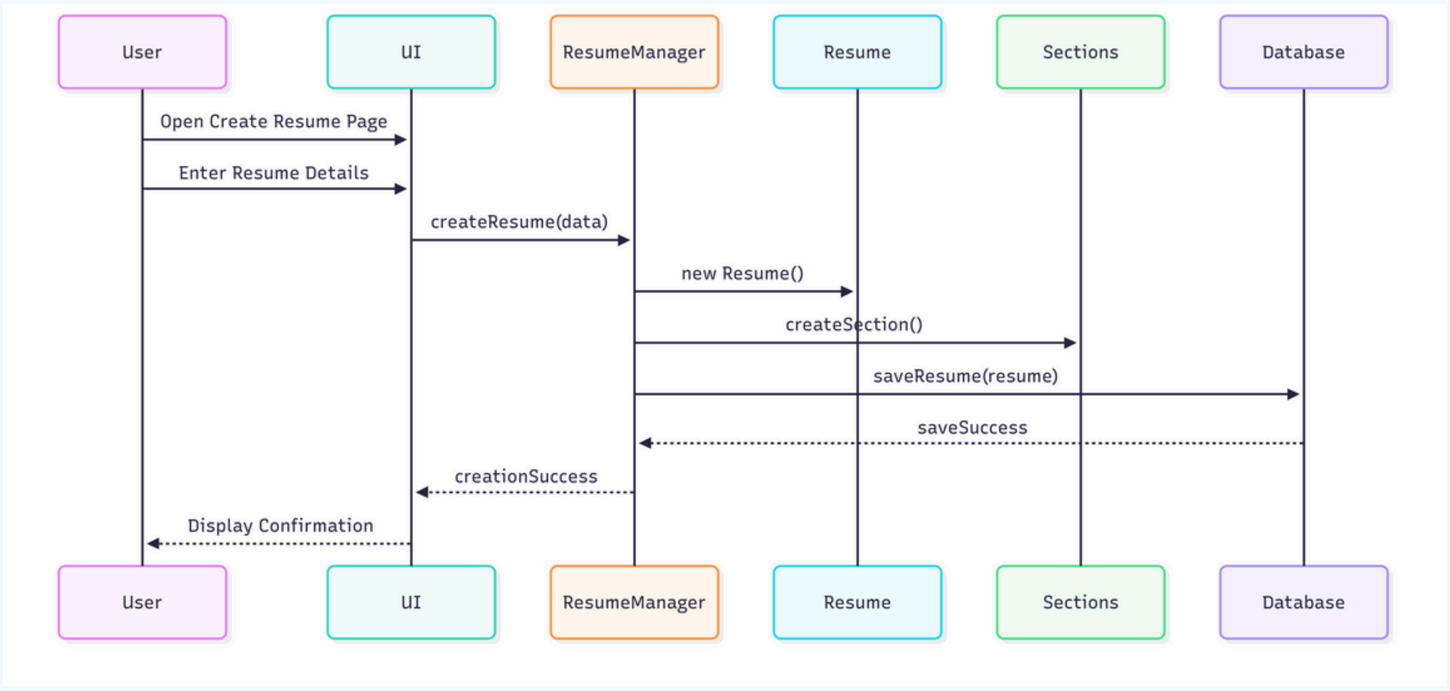


Visual sequence diagrams

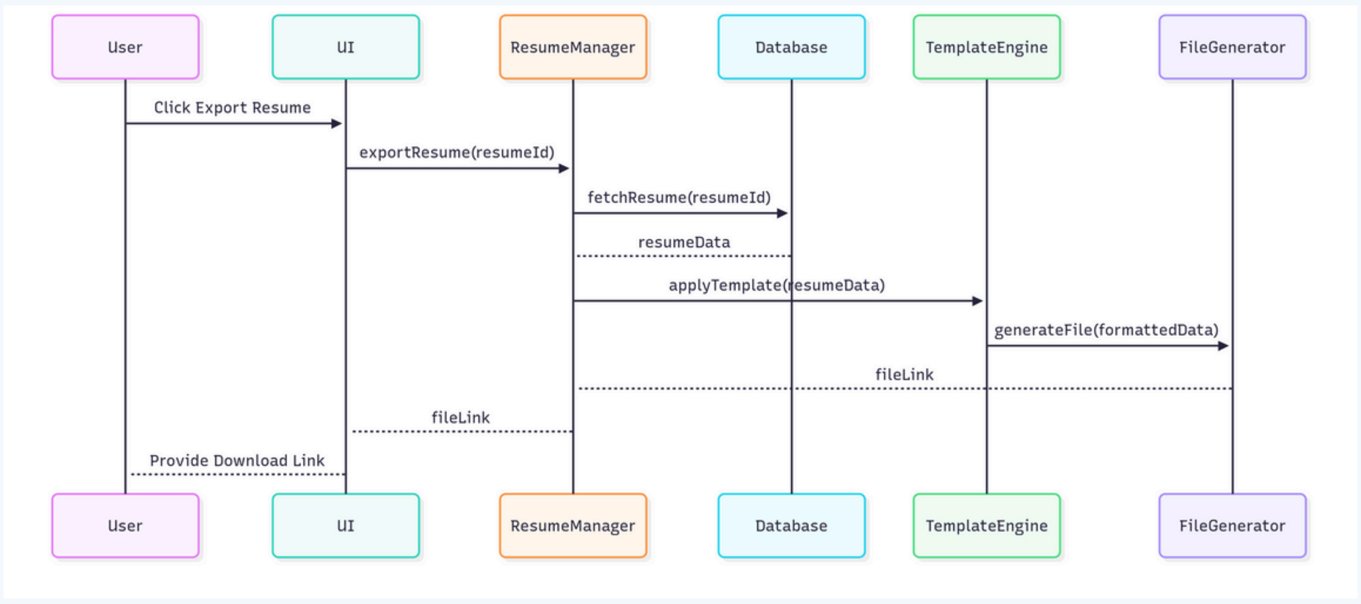
1. **User Registration & Login**



2. Create Resume



3. Export Resume



Implementation and Testing

Validation Methods

1. Email Format Validation
Checks that the email entered by the user follows a standard email pattern, ensuring it contains an '@' symbol and a valid domain. This prevents invalid or malformed email addresses from being accepted.

2. Password Strength Validation
Ensures that passwords meet security requirements by enforcing a minimum length and requiring a mix of uppercase letters, lowercase letters, digits, and special characters. This helps protect user accounts from being easily compromised.

3. Skill Level Validation
Confirms that skill proficiency ratings are within an acceptable range, for example between 1 (beginner) and 5 (expert). This prevents invalid skill levels from being recorded.

4. Date Validation
Validates that start dates and end dates for education or work experience make logical sense, specifically ensuring that the end date is not earlier than the start date. This prevents impossible timelines.

5. Mandatory Field Validation
Checks that required text fields (such as name, institution, job title) are not left empty or filled with only spaces. This ensures important information is not missing from the CV.

6. URL Format Validation
Ensures that URLs entered for profiles like GitHub or LinkedIn start with “http://” or “https://” and follow a valid URL format. This helps avoid broken or incorrect links

Project Timeline

PHASE 1
Weeks 1-4

Planning & Basic UI Setup

- Define project goals and features
- Set up development environment (Java, IDE, GitHub repo)
- Learn Java Swing basics for GUI development
- Design and build simple input forms:
  - Personal details (Name, Email, Phone)
  - Basic sections: Summary, Skills, Education, Experience
- Implement basic input validations (non-empty, simple email format)
- Sketch UI wireframes and plan data flow

PHASE 2
Weeks 5-12

Data Display & Export Implementation

- Develop functionality to display entered information clearly in the app
- Implement export feature to save resume as a plain text (.txt) file
- Add error handling for input and file operations
- Improve input validation messages and usability
- Test all forms and export feature, fixing bugs iteratively
- Refine UI layout for better user experience

PHASE 3
Weeks 13-16

Testing, Documentation & Finalization

- Conduct comprehensive testing (functional, usability)
- Clean up and comment code for readability and maintenance
- Write detailed README.md with:
  - Project overview
  - Setup and run instructions
  - Features and usage guide
- Add screenshots or demo instructions (optional)
- Prepare GitHub repo for sharing (add LICENSE, .gitignore)
- Final bug fixes and polishing