ResuMate

Your Resume, Perfected

➢ User Scenario:

Anan is a fresh graduate aiming to land his first job in a software company. He has heard about Resumate, an automated CV generator that can create professional-looking resumes in minutes.

Anan decides to try out Resumate to create his resume. He navigates to the Resumate website and creates an account by entering his name, email and password. He is then directed to the CV creation page, where he is prompted to enter his personal and professional details, including his education, work experience, skills and achievements.

Anan chooses a template that best fits his style and preference, customizes the layout and previews the final version.He can also view the changes in real time. Finally he exports his resume in a PDF format.

After downloading his resume, Anan is impressed with how professional and visually appealing his resume looks. He decides to apply for several job openings and attaches his newly created Resumate resume with his applications.

➢ Identifying Functional and Non-Functional Requirements

Functional Requirements:

User Registration: The application should allow users to register by providing basic information such as name, email, and password.

Resume Creation: The application should allow users to create a resume by providing personal and professional details, including education, work experience, skills, and achievements.

Resume Templates: The application should provide multiple extensible/modifiable templates l for users to choose from.

Adding Templates: The application should provide options for users to add their own templates..

Preview and Edit: The application should allow users to preview their resume and make any necessary changes before exporting it.

Saving CV: The application should allow users to save their CV so that they can view it later.

Export and Share: The application should allow users to export their resume in a shareable format, such as PDF.

User Management: The application should allow users to manage their profile, view their resumes, and edit or delete them as needed.

Non-Functional Requirements:

Usability: The application should have a user-friendly interface that guides users through the resume creation process.

Compatibility: The application should be compatible with a variety of web browsers and devices.

Security: The application should store user details and resumes securely, protecting them from unauthorized access and data breaches.

Reliability: The application should be available 24/7 with minimal downtime, and backups should be created regularly to prevent data loss.

Scalability: The application should be designed to handle a growing user base and additional features in the future, without compromising performance or usability.

➢ Proposed Solution:

Our solution is a **web application** which will cover the entire process of resume building.
An user will be able to **sign up** using his/her email address and personal information. Then that individual can **log in** using his/her credentials. After that the user will be able to **view his resumes** built previously. The he/she will have the ability to **edit the resumes**. And the user can also start **making a new resume**. Then it will redirect to the resume building page. There will be the information panel, where the user can **input his/her resume information**. And the changes will be reflected in the **preview section** on the same page. And also the user can select **pre-built templates** to make his/her resumes. After completing the resume, the user can **download the pdf**. On the other side of the application, there will be an admin section where the admin **can add new templates** and **manage the users**.

Target Platform: Web
Technologies: MERN (MongoDB, Express, React, Node)

➢ Use-case Diagram:

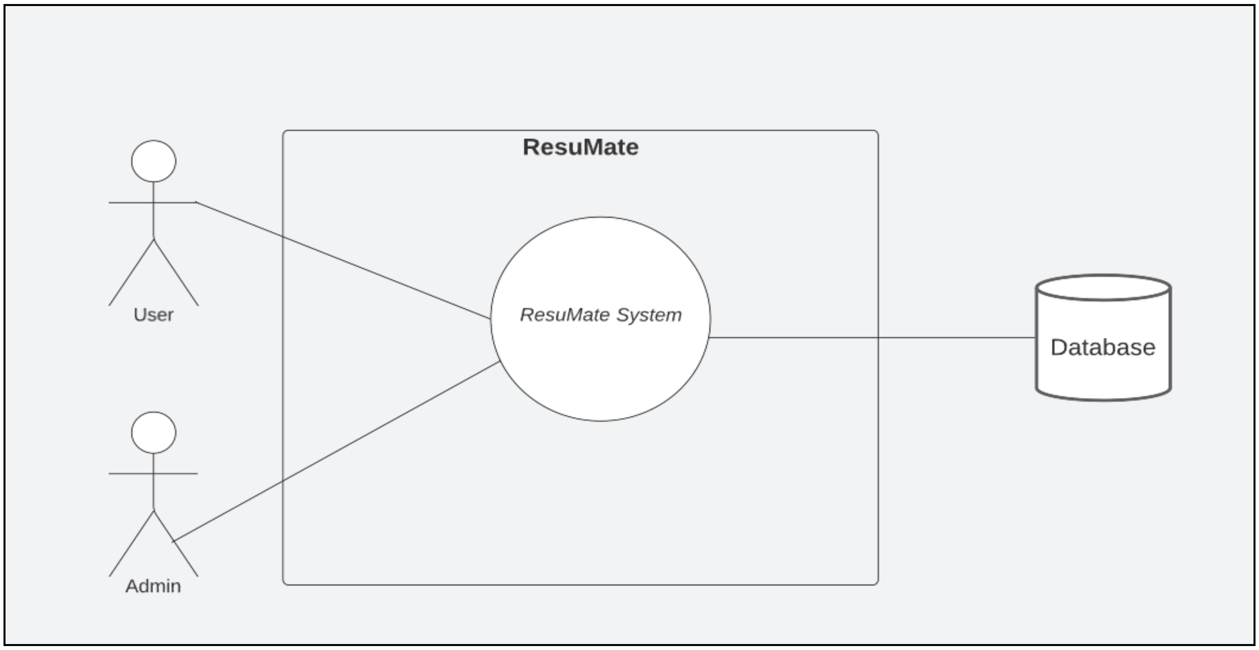


Fig: Level 0 Use case Diagram

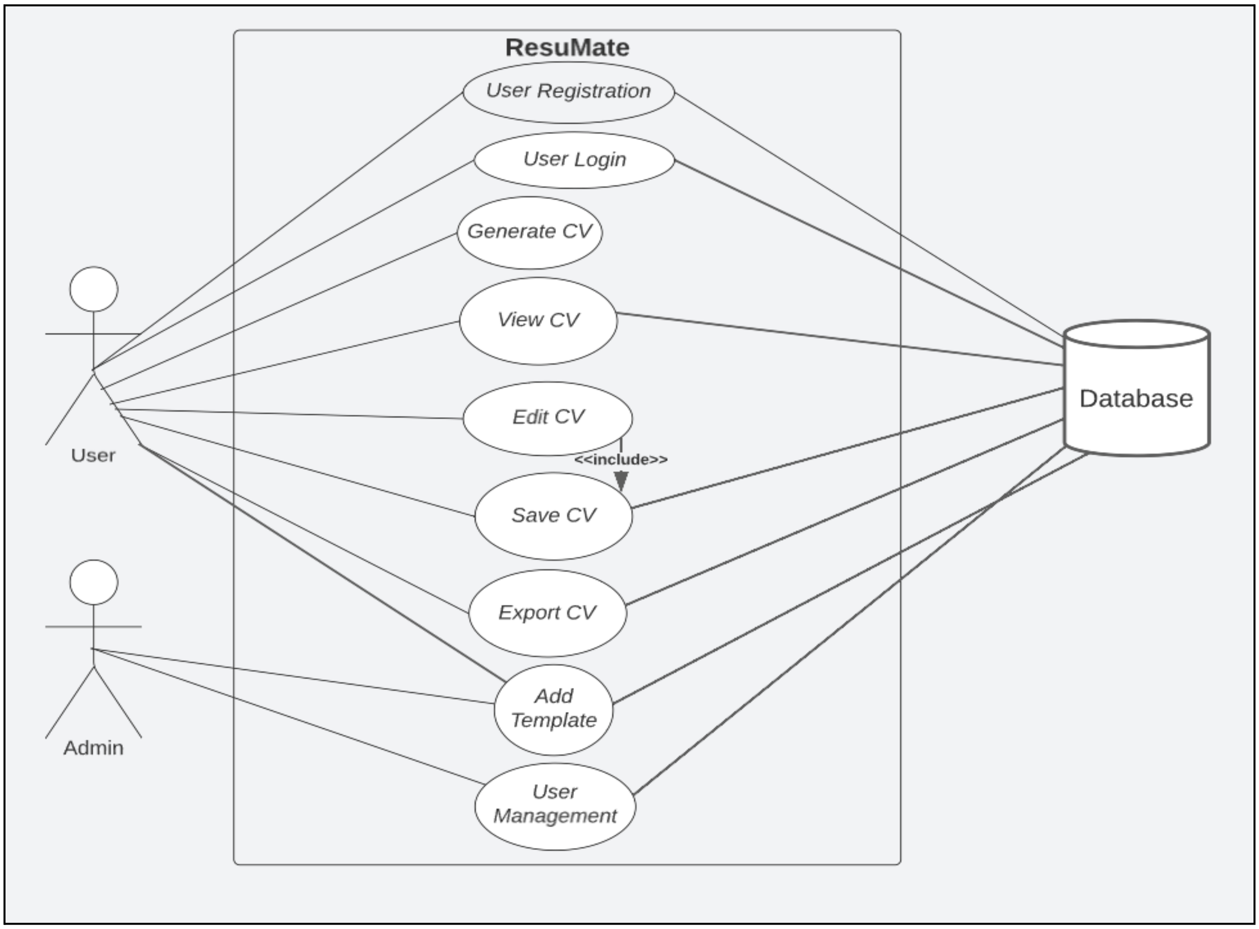


Fig: Level 1 Use Case Diagram

➢ Use-cases in Detail:

Title: User registration

Primary actor: User

Goal in Context: To create a new account on Resumate and access the system's features.

Preconditions:

1. The user has access to a device with an internet connection.
2. The user has not already registered an account on Resumate.

Main scenario:

1. The user navigates to the Resumate homepage.
2. The user clicks on the "Register" button.
3. The system displays the registration form, which includes fields for the user's name, email address, password,phone number and any other required information.
4. The user fills out the form and clicks on the "Submit" button.
5. The system verifies the user's email address and checks for any errors or missing information in the form.
6. If there are no errors, the system creates a new account for the user and displays a confirmation message.
7. The user logs in to Resumate using their email address and password.

Postconditions:

1. The user has a registered account on Resumate.
2. The user can access the system's features and create, edit, and export their resume.
3. The system stores the user's registration information securely and can retrieve it when needed.

Exceptions:

1. If the user enters an invalid email address, the system displays an error message and prompts the user to enter a valid email address.

2. If the user leaves any required fields blank, the system displays an error message and prompts the user to fill in the missing information.

3. If the user enters a password that does not meet the system's security requirements, the system displays an error message and prompts the user to enter a stronger password.

Priority:

This use case has a high priority, as it is essential for users to register an account to access the system's features.

Frequency of use: High

Title: User Login

Primary actor: User

Goal in Context: The registered user is able to access their account by providing valid login credentials.

Preconditions:

1. The user must already be registered in the system.

2. The user must have a valid email and password to login.

Main Scenario:

1. The user opens the website and clicks on the login button.

2. The system presents a login form to the user.

3. The user enters their registered email and password in the login form.

4. The system verifies the email and password.

5. The system authenticates the user and logs them into their account.

6. The user is presented with their dashboard/homepage.

Postconditions:

1. The user is successfully logged into their account.

2. The user can access the features and functionalities available to them based on their account type and permissions.

Exceptions:

1. Invalid Email or Password: If the user enters an invalid email or password, the system displays an error message and prompts the user to enter the correct information.

2. Network Connectivity Issues: If there are network connectivity issues, the system displays an error message indicating that there is a problem with the network connection. The user is prompted to check their network connection and try again.

3. Server Downtime: If the system is down for maintenance or other reasons, the user is notified with a message indicating that the server is unavailable. The user is prompted to try again later.

Priority: High

Frequency of Use: Very High

Title: Generate CV

Primary actor: User

Goal in Context: The job seeker is able to generate a professional and visually appealing CV with their personal and professional details.

Preconditions:

1. The job seeker must be logged into their account.

2. The job seeker must have entered their personal and professional details in the system.

Main Scenario:

1. The user clicks on the "Generate CV" button.

2. The system presents the user with a list of available CV templates and allows them to select a preferred template.

3. The user selects a template and the system generates a preview of the CV.

4. The user reviews the preview and makes any necessary changes to the details or layout.

Postconditions:

1. The user has a professional-looking CV.
2. The user can save and export the CV.

Exceptions:

1. Invalid Data: If the user enters invalid data or omits required information, the system displays an error message and prompts the user to correct the data.

2. Template Issues: If there are any issues with the selected template, such as formatting errors or missing sections, the system displays an error message and prompts the user to select a different template.

Priority: High

Frequency of Use: High

Title: View CV

Primary actor: User

Goal in Context: The user is able to view and access their saved CV in the system.

Preconditions:

1. The user must be logged into their account.

2. The user must have previously saved a CV in the system.

Main Scenario:

1. The user clicks on the "View CV" button.

2. The system displays the CV for the user to view.

Postconditions:

1. The user is able to view the selected CV.

Exceptions:

1. No Saved CVs: If the user has not saved any CVs in the system, the system displays a message indicating that there are no saved CVs.

Priority: Medium

Frequency of Use: Moderate

Title: Save CV

Primary actor: User

Goal in Context: The user is able to save their created CV in the system for future access and editing.

Preconditions:

1. The user must be logged into their account.
2. The user must have created a new CV or edited an existing CV.

Main Scenario:

1. The user clicks on the "Save CV" button.
2. The system prompts the user to enter a name for the CV.
3. The user enters a name for the CV.
4. The system saves the CV with the entered name and displays a message indicating that the CV has been saved.

Postconditions:

1. The user's created or edited CV is saved in the system with the entered name.

Exceptions:

1. Duplicate Name: If the user enters a name for the CV that already exists in the system, the system prompts the user to enter a different name.

Priority: Medium

Frequency of Use: High

Title: Edit CV

Primary actor: User

Goal in Context: The user is able to access and edit their previously created CV.

Preconditions:

1. The user must be logged into their account.

2. The user must have previously created a CV and saved it in the system.

Main Scenario:

1. The user navigates to their dashboard where they can see all their saved CVs.
2. The user selects the CV they wish to edit.

3. The system displays the selected CV in an editable format.

4. The user makes the desired changes to the CV.

5. The user clicks on the "Save" button to save the edited CV.

Postconditions:

1. The edited CV is saved in the system with the changes made by the user.

Exceptions:

1. The user does not have any previously saved CV.

Priority: High

Frequency of Use: Moderate

Title: Export CV

Primary actor: User

Goal in Context: The user is able to export their created CV into a shareable format.

Preconditions:

1. The user must be logged into their account.

2. The user must have previously created a CV and saved it in the system.

Main Scenario:

1. The user navigates to their dashboard where they can see all their saved CVs.
2. The user selects the CV they wish to export.
3. The system displays the selected CV in a preview format.
4. The user reviews the CV preview and confirms that they want to export it.
5. The user selects the export option and chooses the desired format (e.g. PDF).
6. The system generates and downloads the CV in the selected format.

Postconditions:

1. The CV is downloaded in the selected format and is ready for sharing or printing.

Exceptions:

1. Invalid Format: If the user selects an unsupported export format, the system displays an error message indicating that the selected format is not valid.

Priority: High

Frequency of Use: Moderate to High.

➢ Sequence Diagram:

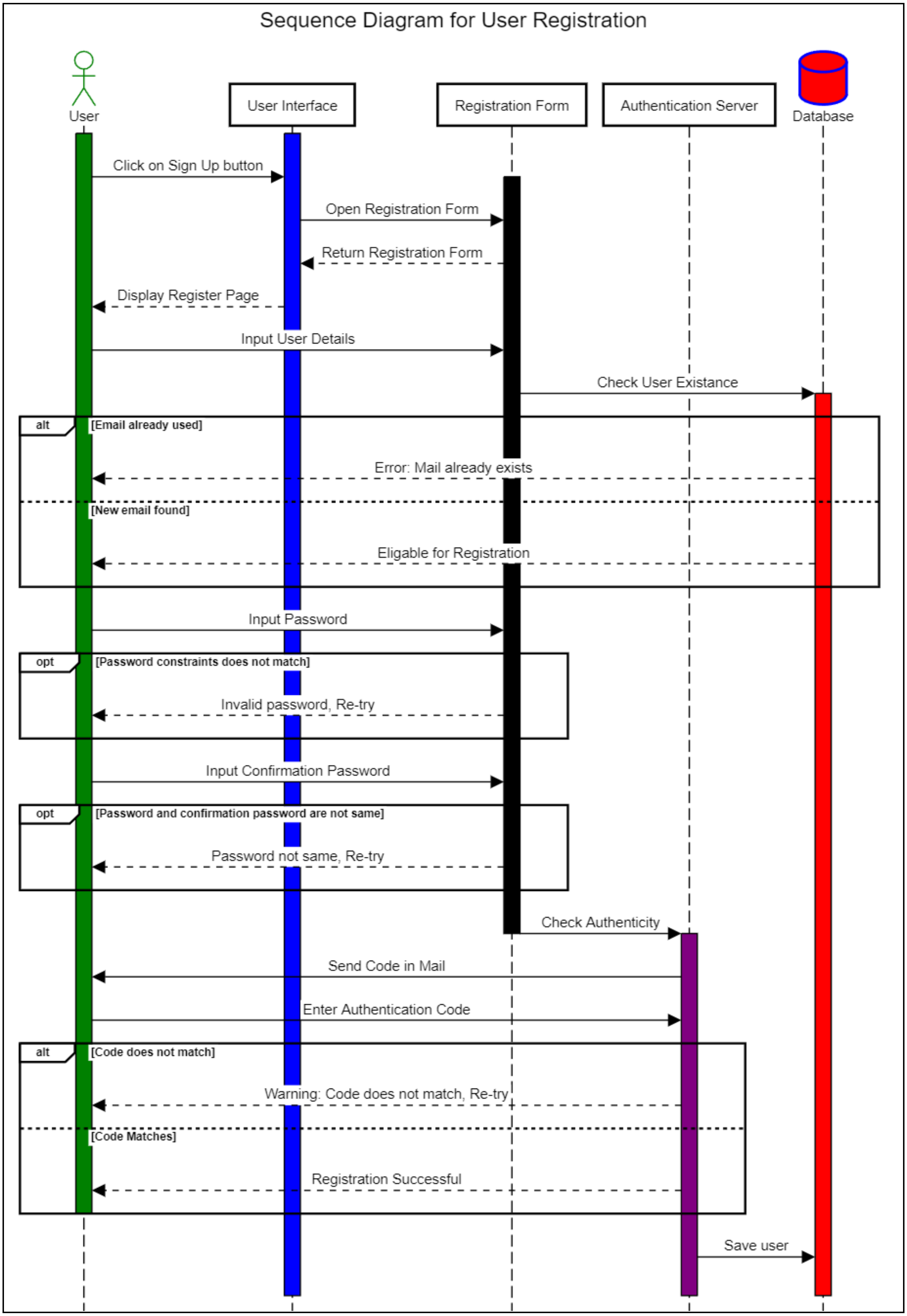


Fig: Registration Sequence Diagram

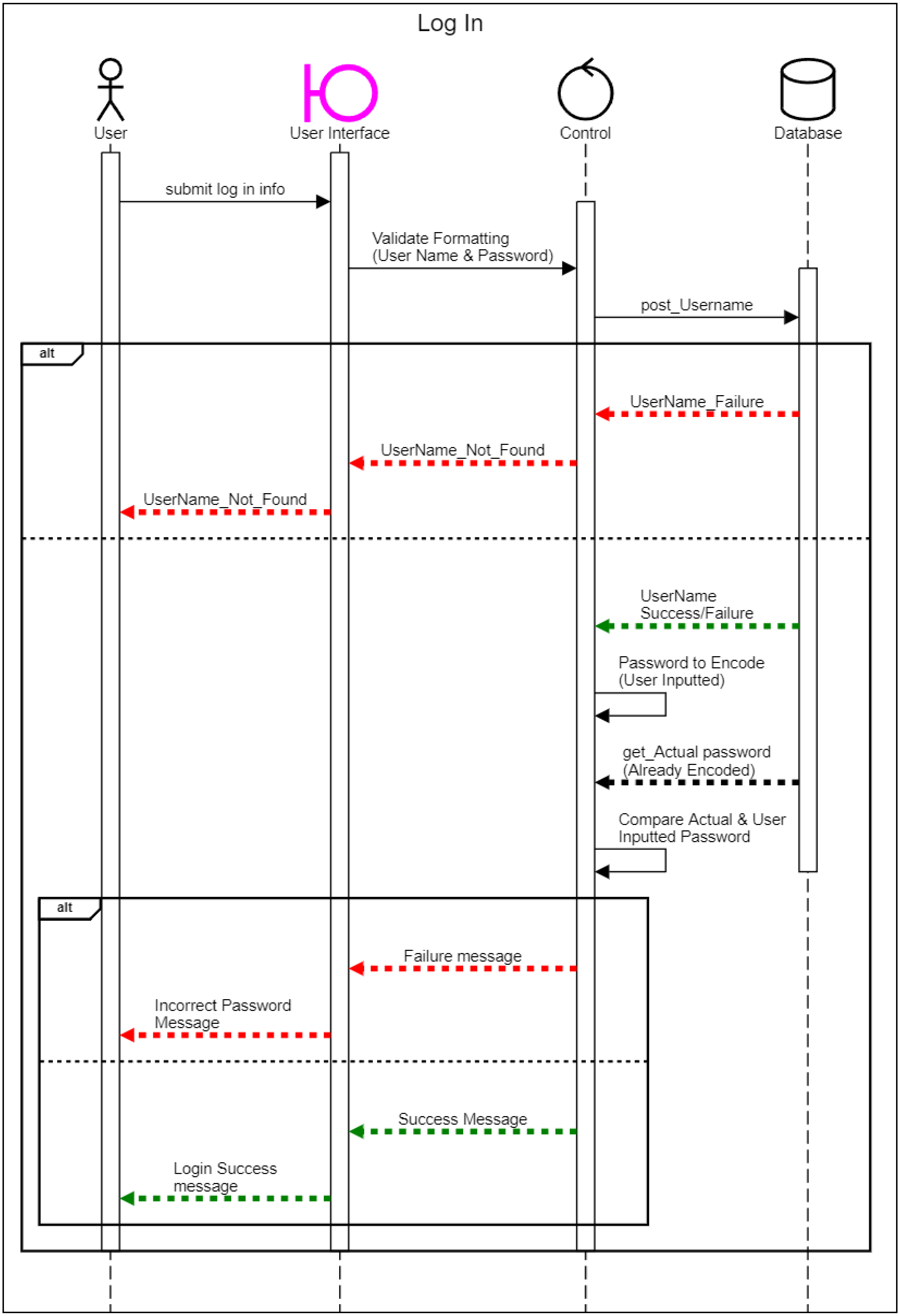
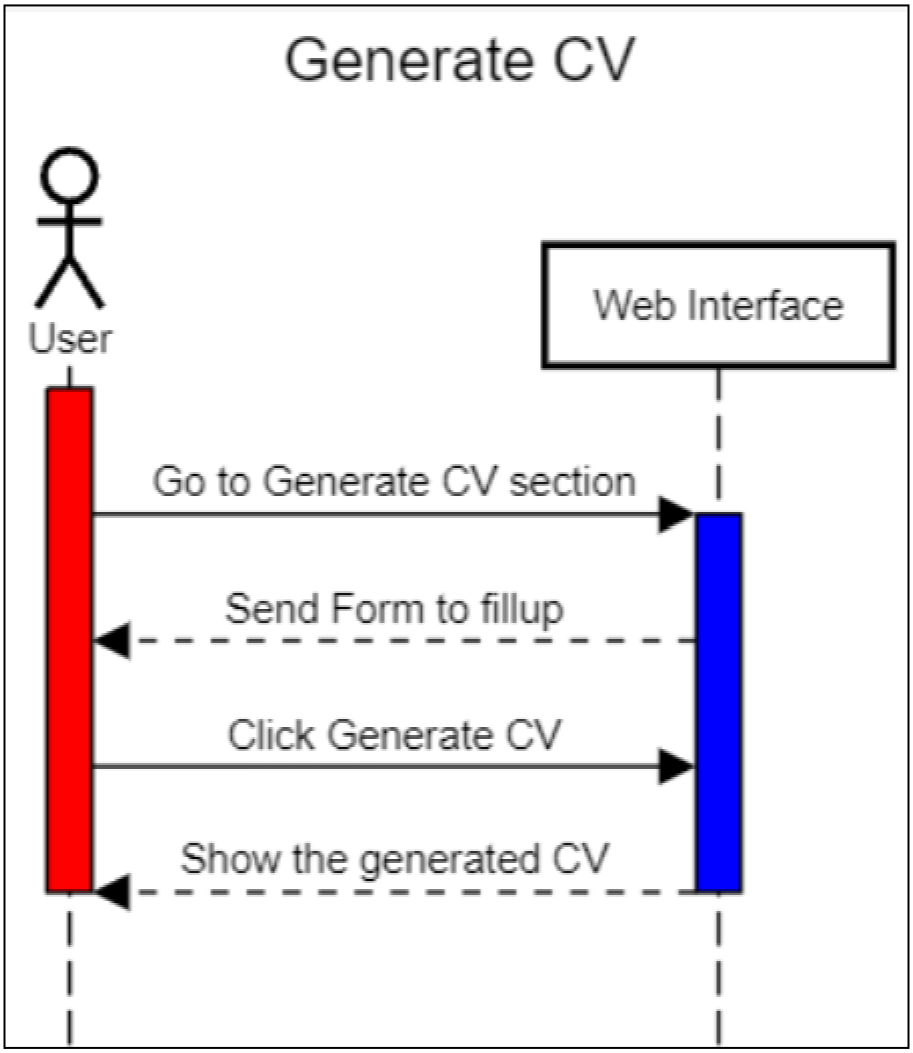


Fig: Login Sequence Diagram



<u>Fig: </u>Generate CV sequence diagram

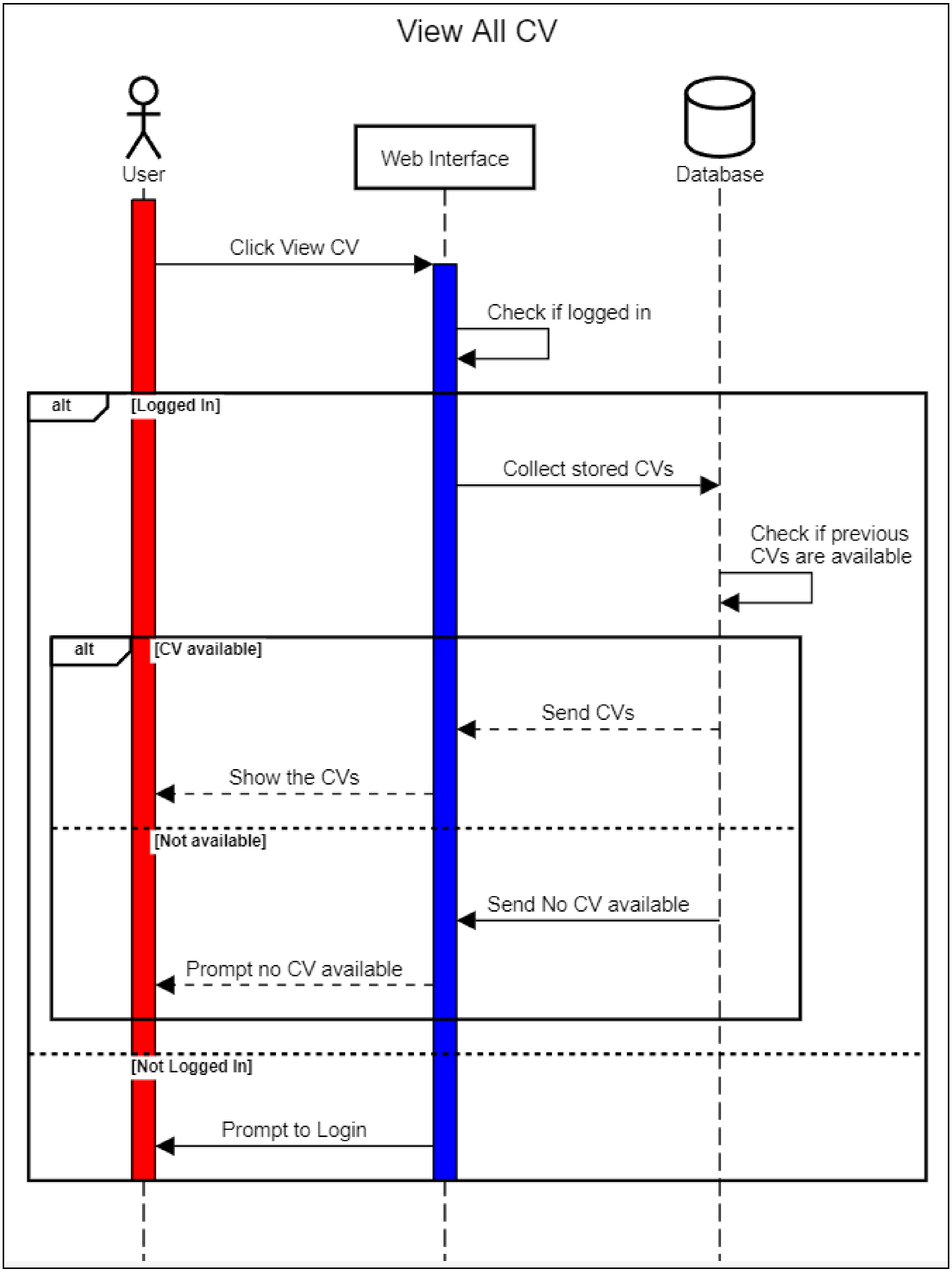


Fig: View CV Sequence Diagram

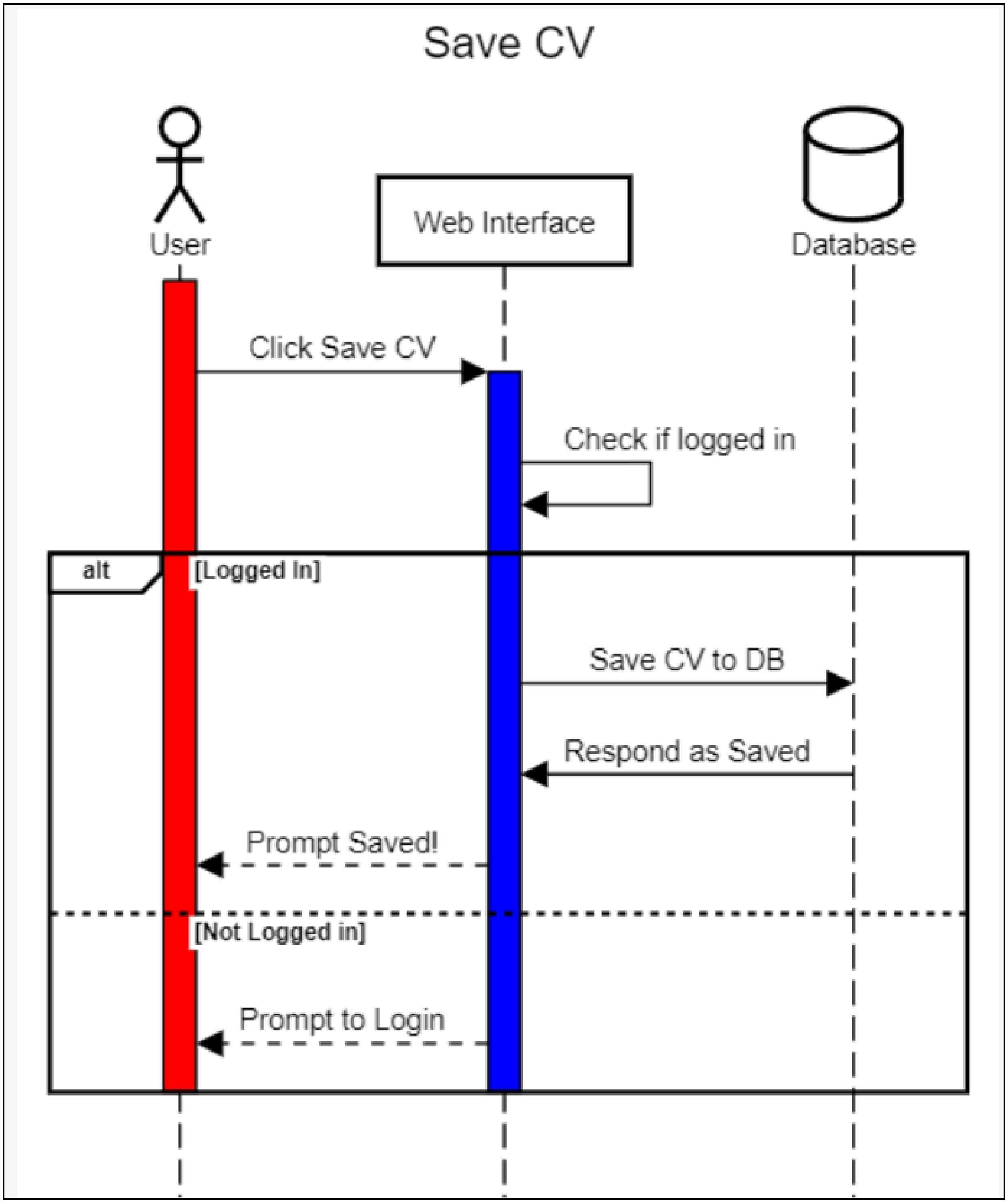


Fig: Save CV sequence diagram

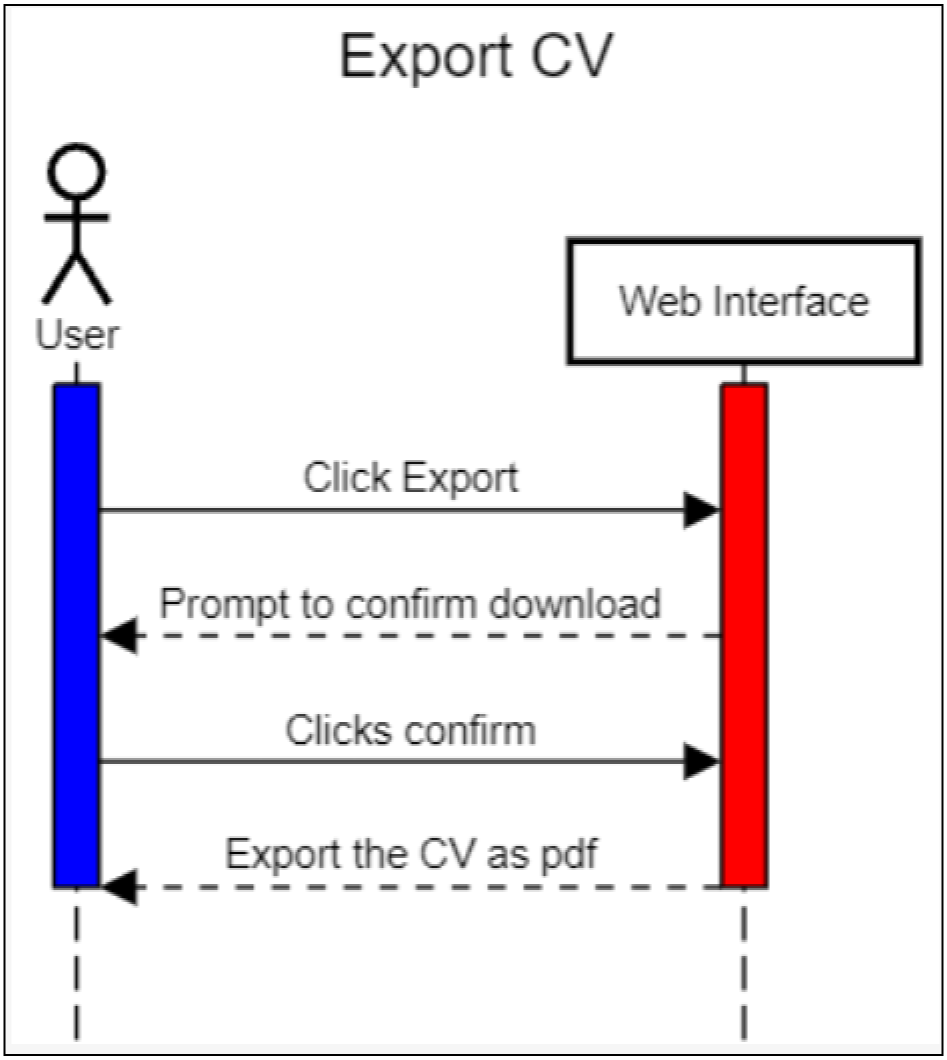


Fig: **Export CV Sequence Diagram**

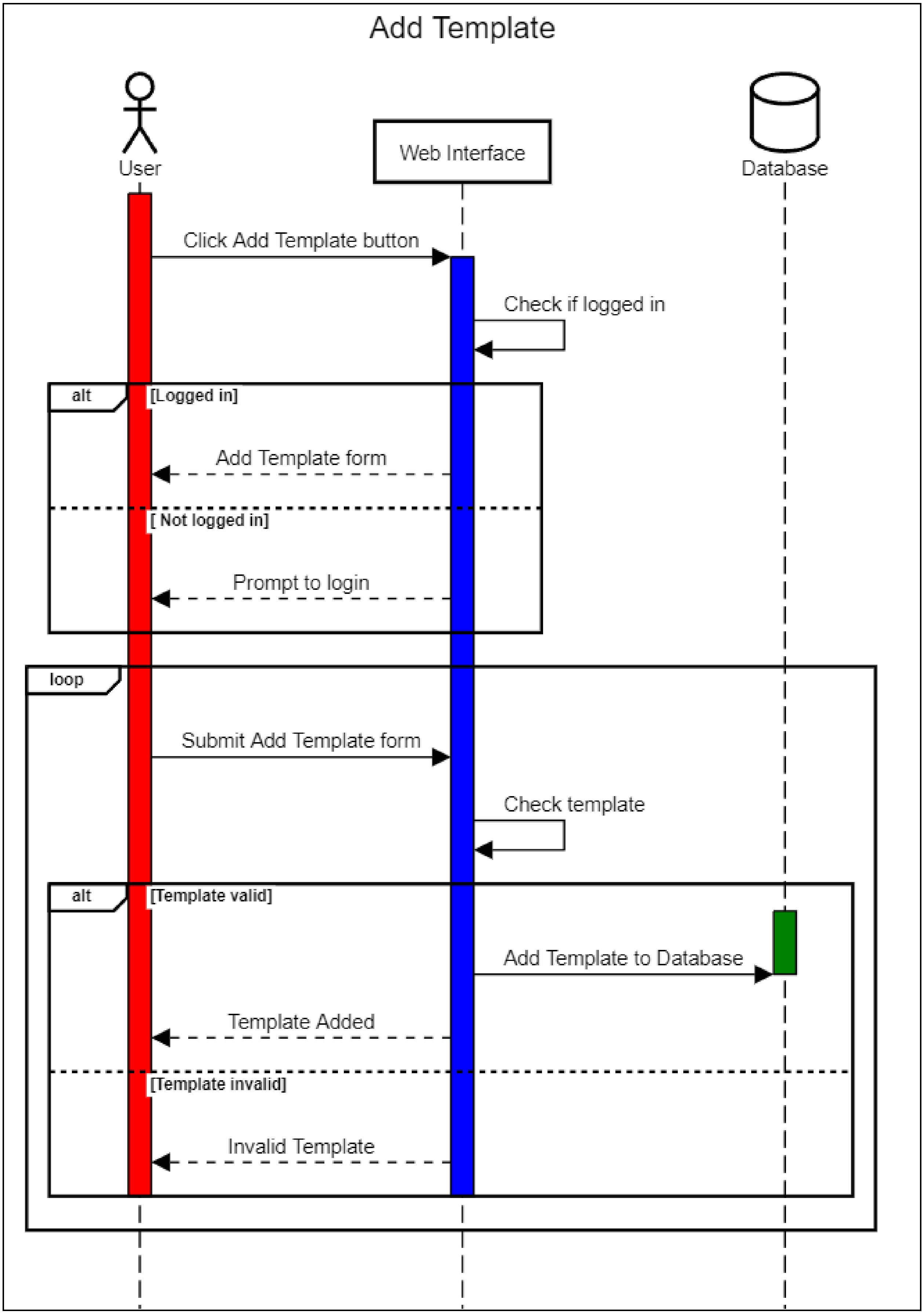
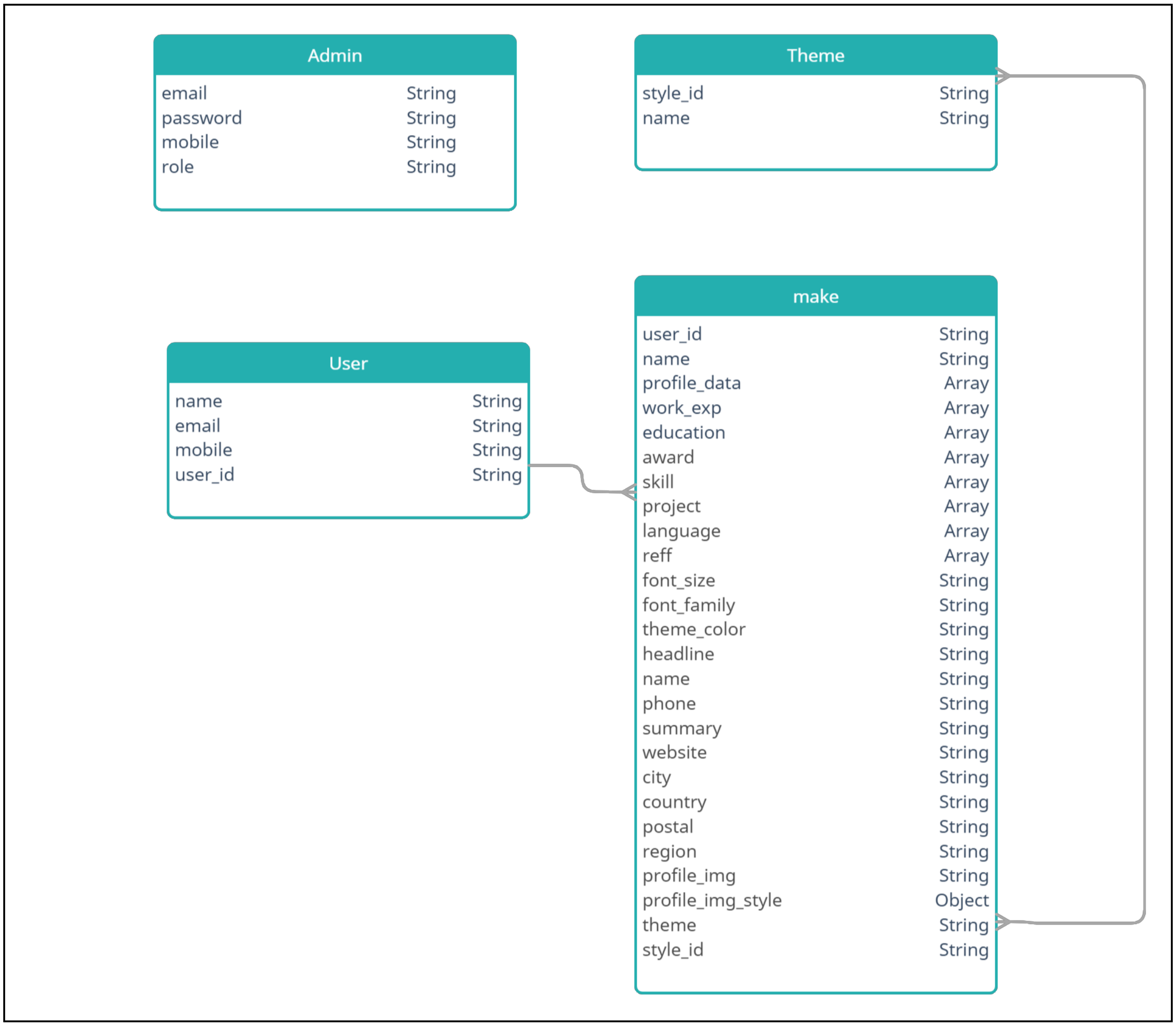


Fig: Add Template Sequence Diagram

➢ Database Schema:



Admin	
email	String
password	String
mobile	String
role	String

Theme	
style_id	String
name	String

User	
name	String
email	String
mobile	String
user_id	String

make	
user_id	String
name	String
profile_data	Array
work_exp	Array
education	Array
award	Array
skill	Array
project	Array
language	Array
reff	Array
font_size	String
font_family	String
theme_color	String
headline	String
name	String
phone	String
summary	String
website	String
city	String
country	String
postal	String
region	String
profile_img	String
profile_img_style	Object
theme	String
style_id	String

Fig: Database Schema

➢ Data Flow Diagram:

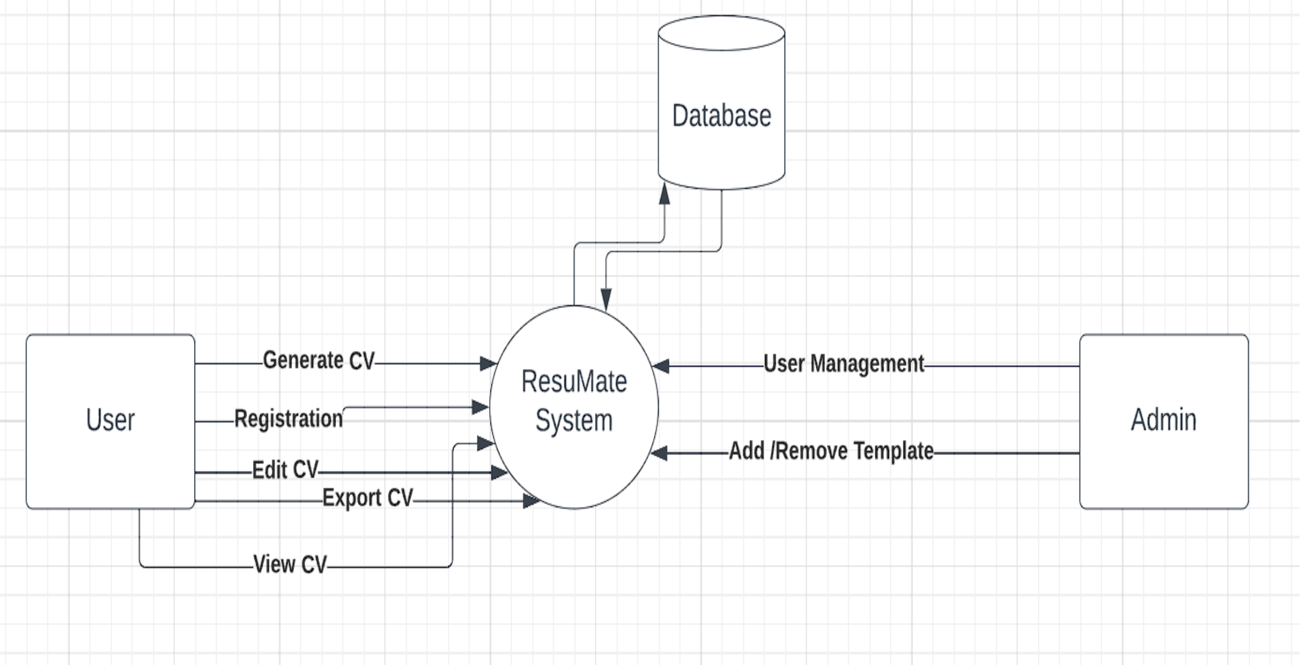


Fig: Level 0 DFD