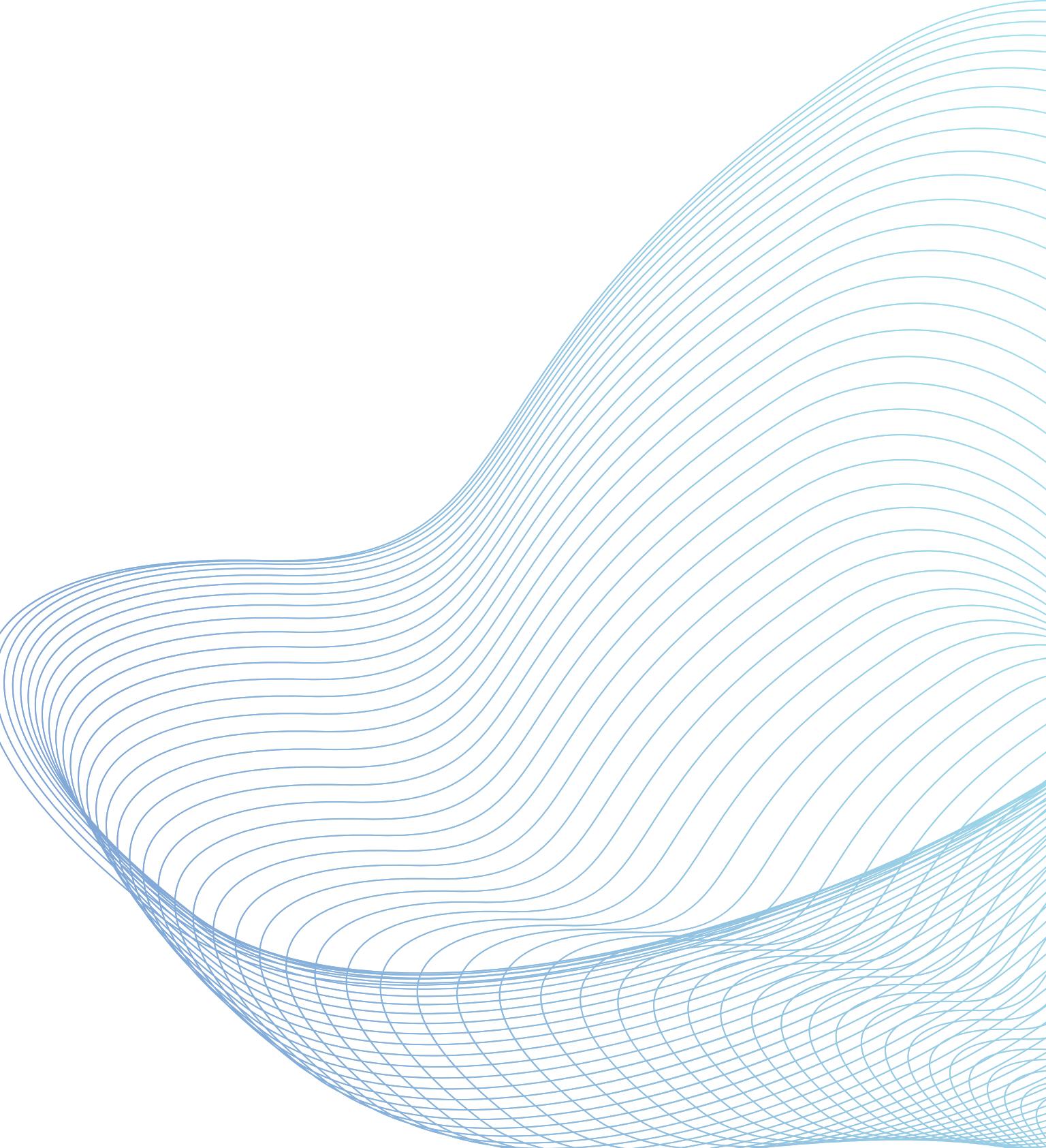




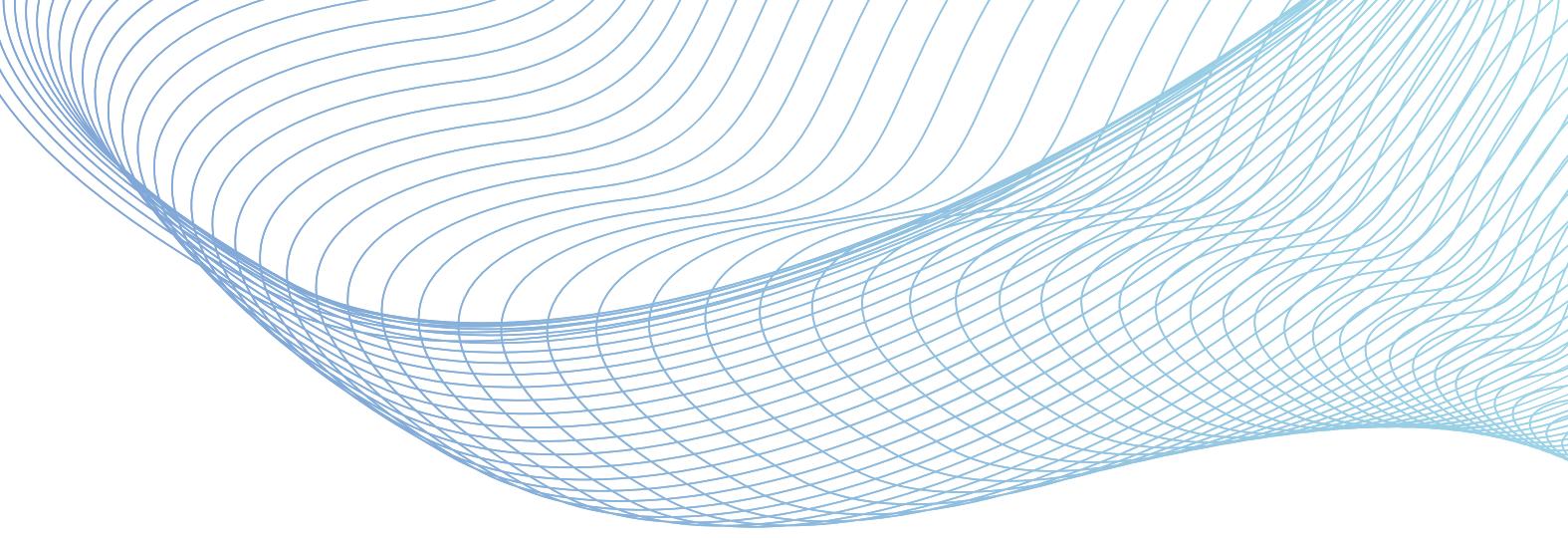
POLITECNICO  
MILANO 1863

# RASD & DD

David Gadiaga  
Andrea Pesciotti  
Simone Somazzi



# GOALS OF OUR SYSTEM

- 
1. Allow students to look for internships
  2. Allow students to improve their CVs
  3. Allow students to apply for an internship
  4. Allow a company to hire interns
  5. Allow companies to improve their internship proposals
  6. Allow users to track and monitor the status of the internship
  7. Allow users to report feedback and complaints
  8. Allow universities to review and address complaints

# SYSTEM BOUNDARIES

## World Phenomena

- Student decides to look for an internship
- Company decides to open a new internship position

## Machine Controlled

- System provides suggestions to improve the CV
- System asks students and companies to provide feedback and suggestions

## World Controlled

- Student applies for internship
- Company approves application
- University monitors internships

# REQUIREMENTS

## Profile Management

- The system allows students to insert personal information for the creation of the CV
- The system allows companies to insert information for the creation of the internship

## Internship Management

- The system allows users to apply to an internship
- The system allows companies to evaluate students' applications

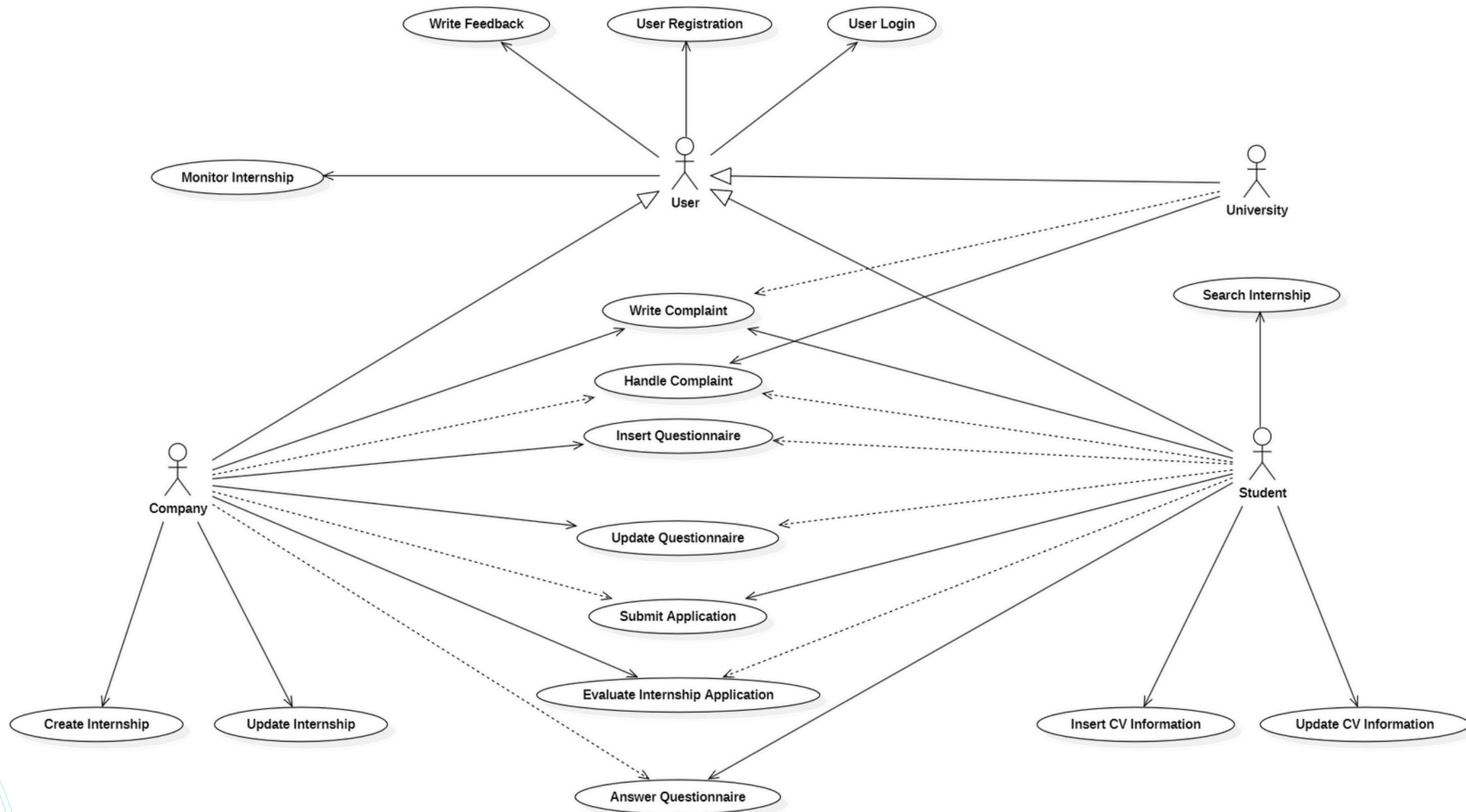
## Complaints Management

- The system allows users to name other parties involved in the complaint
- The system allows universities to close an internship

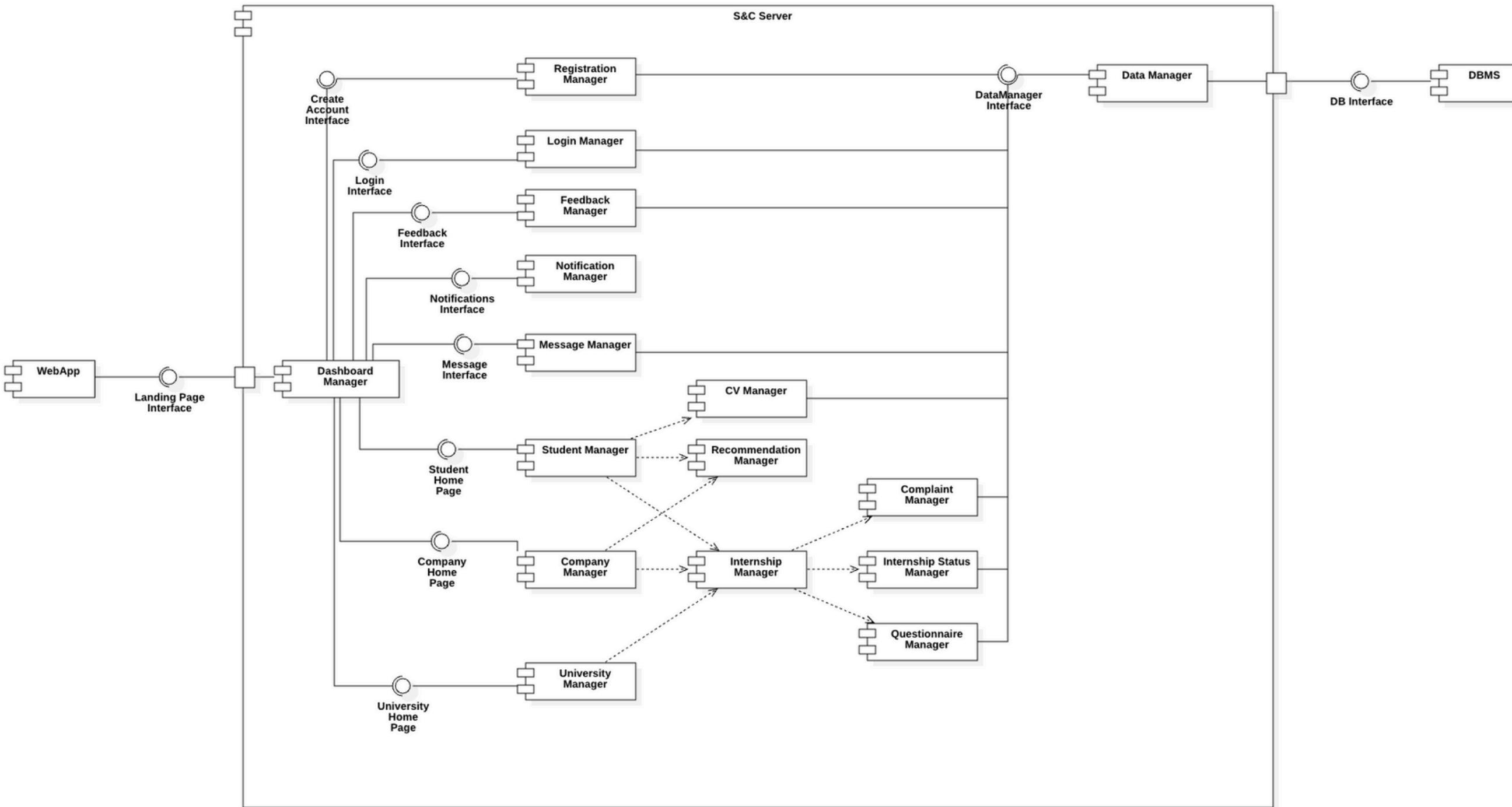
# ASSUMPTIONS

- Users must provide correct personal information at the moment of registration
- Student must be enrolled in a university
- Company must provide correct and clear information about the internships
- University must have students who are actively enrolled in its programs

# USE CASE DIAGRAM

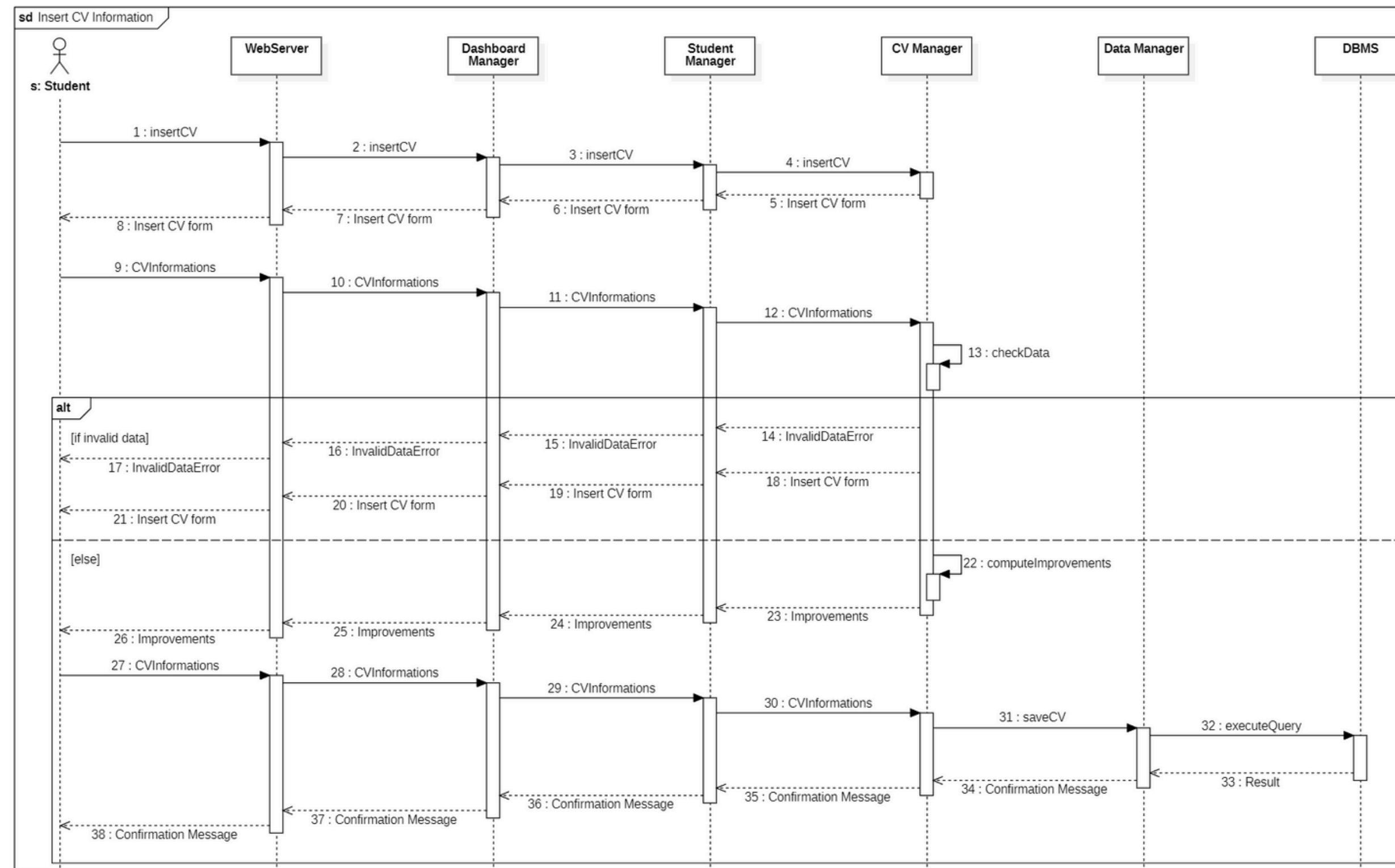


# COMPONENT DIAGRAM



# SEQUENCE DIAGRAM

## Insert CV Information



# ARCHITECTURAL STYLES AND PATTERNS

## 3-Tier Architecture

Handles interactions with the user. Includes the user interface (UI) that captures inputs and displays outputs. Accessible via web browsers or mobile devices, ensuring flexibility.

## Client-Server Interactions

The client is the front-end user interface, serving as the link between the end users and the system. The server is the back-end platform, receiving user requests, performing necessary computations, and returning the results

## Model-View-Controller Pattern

The application's internal structure follows the MVC pattern, a design principle that divides the software into three interconnected components: Model, View, Controller:

# IMPLEMENTATION PROCESS

A Bottom-Up approach will be used.

This strategy involves starting from the lower levels of the "uses" hierarchy, beginning with the development of smaller, independent modules that do not depend on other components to function

1

## STEP 01

Features Identification

2

## STEP 02

Incremental Integration and Progressive Feature Addition

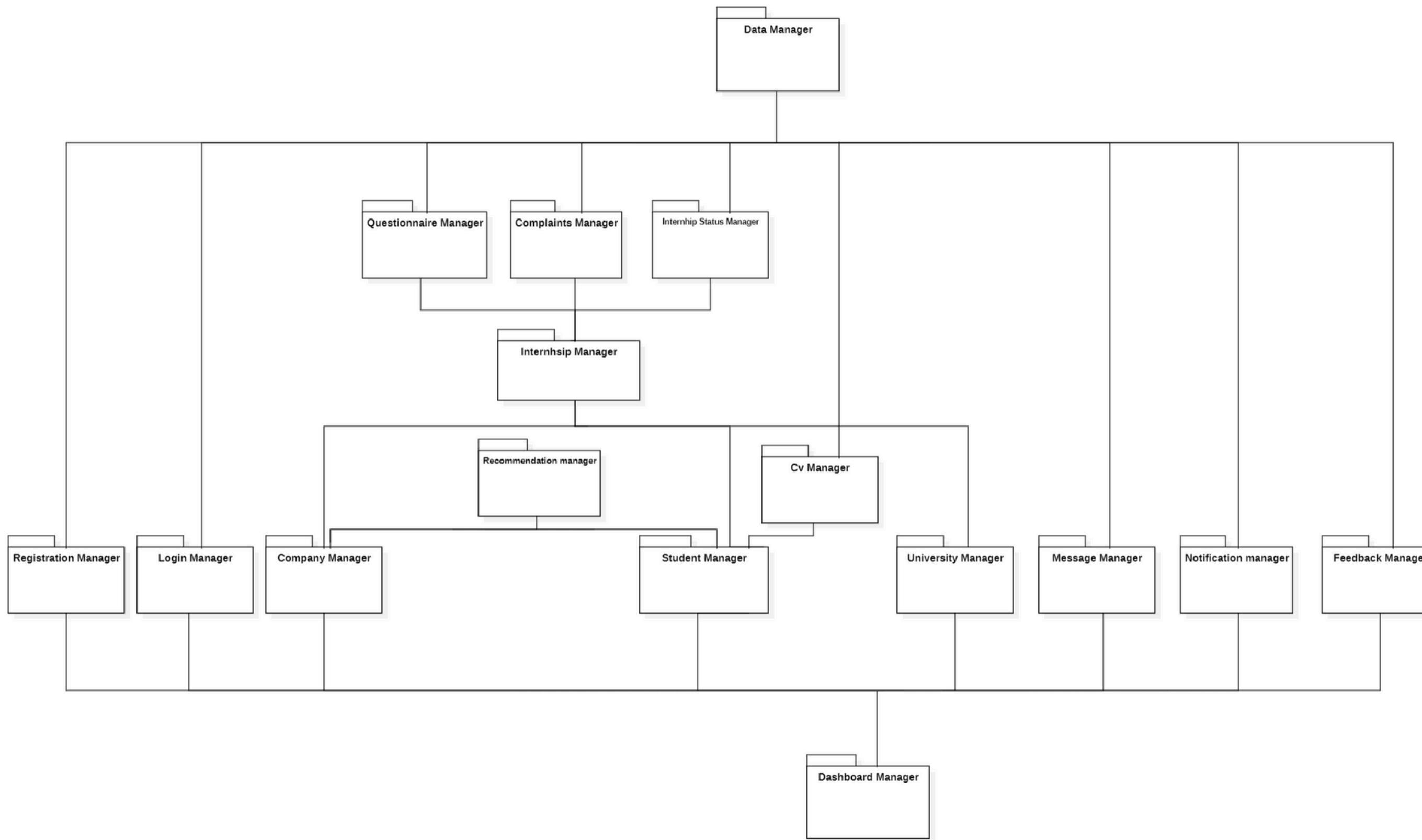
3

## STEP 03

Final Integration



# IMPLEMENTATION



# INTEGRATION STRATEGY AND TESTING

## Component-Level Testing

Ensure that individual components work as intended using stubs and drivers.

## System-Level Testing

Verify that the entire system operates as per functional and non-functional requirements after full integration

## Functional Testing

Validate that use-case scenarios are executed correctly

## Usability Testing

Ensure that the UI is user-friendly and accessible on different devices.

## Stress Testing

Examines the system's ability to recover from failures

## Performance Testing

Identify and resolve bottlenecks in response times and resource utilization

## Load Testing

Evaluates the system under increasing workloads to identify its upper limits.

## User Interface Testing

Ensures the system works easily across different platforms