University Institute of Information Technology,

PMAS-Arid Agriculture University,

Rawalpindi Pakistan

Product Lifecycle Management

By

Abdullah Talat	**18-ARID-2595**
Sharjeel Wakeel	**18-ARID-2709**

Supervisor

Mr. Zeeshan Javed

Bachelor of Science in Computer Science (2018-2022)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software documentation and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Abdullah Talat Sharjeel Wakeel

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS(CS) “**Product Lifecycle Management**” was developed by “Abdullah Talat**, 18-arid-2595”** and “Sharjeel Wakeel**, 18-arid-2709”** under the supervision of “Mr. Zeeshan Javed” and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Science.

Supervisor

External Examiner

Administrator UIIT

Executive Summary

Software is delivered to the end-user when it performs their task successfully but the management of software is too tough because software delivers after the completion of several phases. In our project, we manage these phases. In our project, there are two main users one is an admin who is able to create products and add members. Admin also has the authority to edit or delete the member and a product. Admin assign project to a member. And the other user is members who analyze what cost occur on a project which risk factor are faced during deployment how to increase the quality of the product and what resources need to accomplish the project. Member also check the feasibility of a product. In the existing system, we faced configuration problems even member isn’t able to create an account or manage his project. He needs some training then he is able to use those systems. He tries to overcome this issue through our project.

Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Dr. Zeeshan Javed” for personal supervision, advice, valuable guidance and completion of this project. We are deeply indebted to him for encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Abdullah Talat Sharjeel Wakeel

Abbreviations

SRS	Software Requirement Specification
PC	Personal Computer

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

In this chapter we will discuss over view of our whole project, its brief introduction, and how it is relevant to the courses which we have studied during our degree. We will also discuss project literature review and its analysis and methodology that we will use in project.

1.1. Brief

A process that is suitable to carry out a software project usually consists of several phases going far beyond the implementation of the software. These phases include, among others, project management, requirements analysis and management, quality assurance, and of course the technical phases such as implementation, testing, and maintenance. A holistic development process should address all these relevant phases of the software lifecycle, which we also refer to as project disciplines. Jira and FlinkISO are the software that provides the platform where you manage the product lifecycle but these software have some limitations. In our project, we overcome discussing software issues. In our system, there are three actors one of the project admin who is able to create projects, add members, view projects, view member's detail, and assign projects to members. Project Admin is also able to edit and delete the project and member detail and chat with members. On the other hand, members are able to see assigned projects and the history of recent assign project detail. Member able to risk analysis, schedule management, procurement management, resources management, cost management, review proposal, integration management, stakeholder management, and communicate with the admin. The last actor who is a web manager who manages a website adds new features and if a new language introduces in the market, then he adds in the select box field if a stakeholder of a system faces some issue in the website then he overcomes them. He is able to view the overall traffic of the system.

1.2. Relevance to Course Modules

Almost everything which we are using in our project “Product Lifecycle Management” is totally relevant to our course materials.

- The website which we are developing in our project is relevant to the subject “Web Development".
- The data of the users which will be stored in Database and status will be shown on screen, is relevant to the subject Database Management System.

1.3. Project Background

Generally, when we want to develop something, first of all, we make a strategy where we start first and where we end, where we focus most you answer these questions. After that, you start working on your project to achieve the desired goal but what if a strategy doesn't support you to achieve your desired goal? what if the platform interface is difficult and you want to make a strategy or take a guess? What if you aren't able to create an account on these platforms?. The existing system has a limitation which we raise in the form of a question in the above discussion. So our aim is to provide users simple and user-friendly interface where he easily does their work and easily configure his account and communicate with each other.

1.4. Literature Review

Most of the prevailing product lifecycle management doesn’t have a significant system. Some of them don't have a communication system between the team and some of them don’t have a user-friendly interface. Some of them need to give training for use because of the difficult interface. and some of them are difficult to configure so these issues occur with the existing system. Every system is good but incomplete to provide complete facilities.

Flinkiso is an existing system in which we manage, audit the project. We also assure project quality but this platform has a problem that is following.

- It doesn’t have defect tracking
- It doesn’t have a feature of managing the equipment.
- It doesn’t manage the risk.
- It doesn’t not manage maintenance.

Jira is a software application used for issue tracking and project management. The tool, developed by the Australian software company Atlassian, has become widely used by agile development teams to track bugs, stories, epics, and other tasks. But it has following limitations.

- The tool is hard to set up and get used to.
- It’s complicated user interface can make managing tasks difficult.
- No built-in timeline to track your project progress.
- No collaboration features to communicate with your team.
- It’s mainly built for engineering and software development teams.
- No idea management features to keep track of your ideas and plans.
- The tool can be expensive.
- It’s known for being a slow tool with long query load times.

1.5. Analysis from Literature Review (in the context of your project)

The goal is to provide user-friendly interface users easily interact with our system and do their work. They don't need the training to use this system. They easily create an account on this system because this system provides an easy method to create an account. This system has communication where teammates communicate with each other.

1.6. Methodology and Software Lifecycle for This Project

We use Agile methodology. The proposed system is going to follow the agile development method because this method assists in responding to the unpredictability of constructing software. This method offers a light framework and focus on rapid delivery of the software. Moreover, agile method facilitates us to create and respond to change in an uncertain and turbulent situation. Our focus is to collaborate with stakeholders during project to ensure the product quality which is impossible without using agile methodology.

1.6.1. Rationale behind Selected Methodology

- Creativity and innovation
- Lower costs
- Improved quality
- Customer satisfaction
- Focus on users
- Early and predictable delivery
- Predictable costs and schedule
- Allows for change

Chapter 2: Problem Definition

2.1. Purpose

Existing systems are good but they have some weaknesses like their interface is not user-friendly and user account configuration is complex. Teammates don't communicate with each other. Sometimes users need the training to use these software because of the complex interface. These software have some limitations so they perform their task in some limitations so users faced some difficulty by using that software. Our aim is to provide a user-friendly interface that helps users to use the software easily. they easily configure their account on this software. They communicate with teammates and project admin.

2.2. Product Functions

The product functions of Product Lifecycle Management are, Register**:** In registration the project admin can add member personal information and the information can be verified and store in to the database and the member account created **Login:** The user provide their account credential after that he see his profile .**Manage Database:** The web admin can manage database i.e., update user, add user, view user, search users, delete users etc. **Create Project:** Project admin can add a new project he adds the basic idea of a project with a proposal and submit after submission form validate if it validates successfully then it project detail store in database and shows success message. **View Projects**: Project admin can view all project detail also able to edit and delete operations on project details. **View Member:** Project admin can view all project detail also able to edit and delete operations on project details. **Assign project:** Project admin able to assign a project to a member he put a project and work detail and submit form after submission form validate then store into database and system show success message. **Add Cost:** Member can add module cost of assign project after addition he submit form after validation store into database then system show success message. **Add Schedule:** Member can add module schedule of assign project after addition he submit form after validation store into database then system show success message. **Add Risk:** Member can add module risk of assign project after addition he submit form after validation store into database then system show success message. **Add Resources:** Member can add resources of assign project after addition he submit form after validation store into database then system show success message. **Add Procurement:** Member can add module procurements of assign project after addition he submit form after validation store into database then system show success message. **Chat:** Teammates communicate to each other. **Notification:** Admin and member receive notification if action perform on project.

- login
- Manage Database
- Create Project
- View Project
- View Member
- Assign Member
- Add Cost
- Add Schedule
- Add Risk
- Add Resources
- Add Procurement
- Chat
- Notification

2.3. Proposed Architecture:

3Tier Architecture has been used in our project.

We have three layers in 3Tier Architecture:

1. Presentation Tier
2. Application Tier
3. Data Tier

- **Presentation Tier:**

We will develop Web Application in which we will create interfaces by using HTML,CSS and BOOTSTRAP . These interfaces will display all the information of User.

- **Application Tier:**

In this layer we will apply logics in our Web Application by using JavaScript and PHP Language.

This tier- also called Middle tier, Logic tier and Business tier, pulled from presentation tier.

- **Data Tier:**

Data is this tier is kept independent of web servers and Business logic.
In this layer we will use Mysql which will manage to store the data of users.

2.4. Project Deliverables:

Following are the deliverables and development requirements:

2.4.1 Projects Deliverables:

In this project the deliverables will consist of the input as equipment (i.e., hardware components and software components) then there will be a process (i.e., development phases in which development will be completed) applied on the input and then as a result there will be an output (Project being completed “Product Lifecycle management”). In this case product deliverables are the completed parts or modules of the project. Input will be software and software-based components. The project is divided into in different modules and each module is major milestone in the project.

2.4.2 Development Requirements:

Development requirements are the requirements needed for the development purposes without which the development is not possible. It can be hardware, software or any kind of requirements. These include the software and hardware equipment’s, time and date constraints, budget, planning, following the SDLC etc. development requirements are met accordingly to make sure that the end result does not differ from what is expected and that it can perform its functionality accurately and perfectly without any glitches.

2.5. Operating Environment:

Operating environment for the Product Lifecycle Management is as listed below:

Operating System: Window 10

Database: Mysql

Platform: Visual Studio Code in which web application will be developed using PHP language. Front end will be designed in HTML,CSS,BOOTSTRAP and backend will be designed in PHP.

2.6. Assumptions and Dependencies:

Assumptions:

The Web Application developed from this technique is more efficient than others. This software gives perfect results in real-time for long queries. user don’t face difficulties that they faced in an existing system.

Dependencies:

This software is dependent on internet connectivity.

Chapter 3: Requirement Analysis

In this chapter we will define all the requirements of proposed system that include functional and non-functional requirements. We will also discuss about use cases of the system and see how our system will respond to various use cases.

3.1. Functional Requirements

For our system to work and facilitate the user number of functional requirements have been are needed. These functional requirements also be presumed as interface requirements as they are all but interface.

- The system will authenticate the user through login functionality.
- The system will display the previous projects.
- The system will display the member list to project admin.
- The system will allow project admin to create new projects .
- The system will allow project admin to perform edit delete operations on created projects.
- The system will allow to perform edit delete operation on member.
- System allow admin to assign project to different members.
- The system will allow member to test the feasibility study.
- The system will allow member to add cost.
- The system will allow member to add risk.
- The system will allow member to add Integration.
- The system will allow member to add stakeholder.
- The system will allow member to add module.
- The system will allow member to add procurement.
- The system will allow member to add Resources.
- The system will allow member to view proposal.
- The system will allow member to chat with its teammate and project admin.
- Member receive notification if teammate or project admin send message.

3.1. Non-Functional Requirements

- The system should be able to handle the concurrent requests from different users.
- The system should provide confidentiality for user data.
- The system should be stable and reliable enough to handle the exceptions.
- Interface and the system itself should be user friendly so that the user will feel it easy to use.
- The system will authenticate the user by verifying the credentials to database.
- If user fill form completely then he will be able to submit otherwise system give error message on a specific fields.

3.1. Use Case Model

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. Following are the use cases of the Product Lifecycle Management

3.3.1 Use Case Diagram:

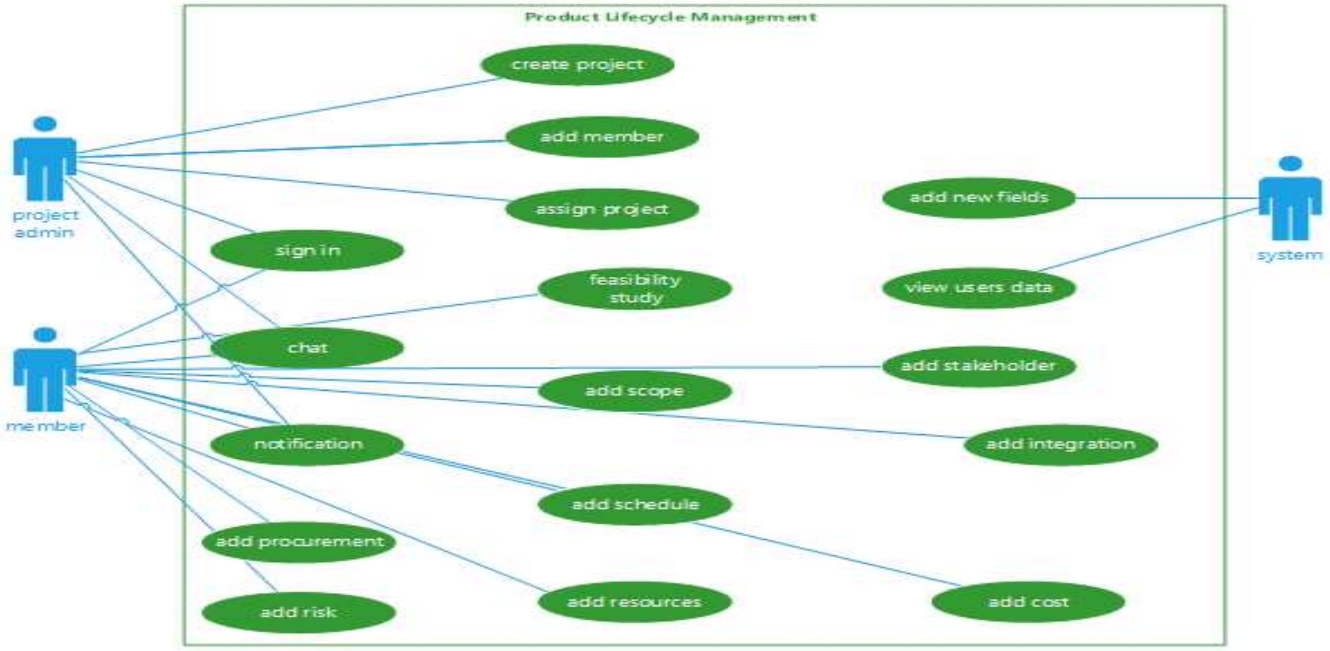

Fig 3.1: Use Case diagram

Actors Description:

We have four types of actors.

Project manager:

Project manager is a primary actor which directly interacts with a website. He can add projects ,add users, assign projects and is also able to edit and delete project, users.

User:

User is also the primary actor .He can view assigned projects and is able to accept or reject projects and perform some operation on the project.

System:

System can add new fields and manage the website.

Database:

Database stores all the information and data fetched through it as well.

Use Case Description

User Case Name:	ID-01
Use Case Name:	Sign in
Actors:	Project admin, user
Description:	User will provide its credentials and be authenticated by the system through the database.
Trigger:	When the user clicks on the sign in button
Preconditions:	User should be sign in page
Postconditions:	User will login successfully and redirect to home page
Normal Flow:	1.user enter email and password. 2.system validates the email and password. 3.system match email and password through database 4.system redirect to home page
Alternative Flows:	2a. Error show please fill required fields 3a.Error show please enter valid email and password 4a.don’t redirect to the home page

Table 3.1 sign in

User Case ID:	ID-02
Use Case Name:	Create Project
Actors:	Project admin
Description:	Project admin clicks on create project option and fills the field after the validation form is submitted and gets a successful message.
Trigger:	When the admin clicks on create project.
Preconditions:	Admin should be on the home page.
Postconditions:	Fill the form and submit an admin get success message.
Normal Flow:	1.click on the add project option. 2.fill all fields. 3.validate all fields. 4.store in the database. 5.get a success message.
Alternative Flows:	3a.Error show please fill the required field.

Table 3.2: Create Project

Use Case ID:	ID-03
Use Case Name:	Add User
Actors:	Project admin
Descriptions:	Project admin clicks on add user option and fills the field after the validation form is submitted and gets a successful message.
Trigger:	When the admin clicks on add user.
Preconditions:	Admin should be on the home page.
Postconditions:	Fill the form and submit an admin get success message.
Normal Flow:	1.click on the add project option. 2.fill all fields. 3.validate all fields. 4.store in the database. 5.get a success message.
Alternative Flows:	3a.Error show please fill the required field.

Table 3.3: Add User

Use Case ID:	ID-04
Use Case Name:	View Projects
Actors:	Project Admin
Descriptions:	Project admin click on view project and view projects
Trigger:	When a user clicks view project.
Preconditions:	Admin should be on the home page.
Postconditions:	View all projects with project id and project title with little description.
Normal Flow:	1.click on the view project. 2.View all projects project id and project title with little description.

Table 3.4: View Projects

Use Case ID:	ID-05
Use Case Name:	View User
Actors:	Project Admin
Descriptions:	Project admin click on view users and view users.
Trigger:	When a user clicks view user.
Preconditions:	Admin should be on the home page.
Postconditions:	View all user with user id and user name with designation.
Normal Flow:	1.click on view user. 2.View all user with user id and user name with designation.

Table 3.5: View Users

Use Case ID:	ID-06
Use Case Name:	Assign Project
Actors:	Project Admin
Descriptions:	Admin click on assign button then fill fields and submit get success message
Trigger:	When a user clicks the assign button.
Preconditions:	Admin should be on the view profile page.
Postconditions:	Admin fill fields press the assign button get success message
Normal Flow:	1.click on the assign button. 2.fill the form. 3.Validate form field 4.store in database 5.system gives a success message
Alternative Flows:	3a.Error show please fill required fields

Table 3.6: Assign Project

Use Case ID:	ID-07
Use Case Name:	Notification
Actors:	Project Admin, User
Descriptions:	Users get notification through the system if someone performs some activity on a given project.
Trigger:	When user clicks on notification
Preconditions:	User should be on the view profile page.
Postconditions:	User check notifications.
Normal Flow:	1.click on notification. 2.check notifications.
Alternative Flows:	None

Table 3.7: Notification

Use Case ID:	ID-08
Use Case Name:	Chat
Actors:	Project Admin, User
Descriptions:	Users get notification through the system if someone texts.
Trigger:	When a user clicks the message button.
Preconditions:	User should be on the home page.
Postconditions:	See who text him and answer him
Normal Flow:	1.get alert 2.click on the message button. 3.see who sent a message to him. 4.answer him. 5.store answer in the database.
Alternative Flows:	None

Table 3.8: Chat

Use Case ID:	ID-09
Use Case Name:	Feasibility study
Actors:	User
Descriptions:	User click on feasibility option fill field.
Trigger:	When a user clicks the feasibility option.
Preconditions:	User should be on the home page.
Postconditions:	User fills the field and clicks the submit button to get success message.
Normal Flow:	1.click on the feasibility button. 2.fill the form. 3.validate form. 4.store in the database. 5.get a success message.
Alternative Flows:	3a.Errro show please fill required field.

Table 3.9: Feasibility study

Use Case ID:	ID-10
Use Case Name:	Add Scope
Actors:	User
Descriptions:	User clicks on the add scope option to fill field.
Trigger:	When a user clicks the add scope option.
Preconditions:	User should be on the home page.
Postconditions:	User fills the field and clicks the submit button to get success message.
Normal Flow:	1.click on the scope button. 2.fill the form. 3.validate form. 4.store in the database. 5.get a success message.
Alternative Flows:	3a.Errro show please fill required field.

Table 3.10: Add Scope

Use Case ID:	ID-11
Use Case Name:	Add Risk
Actors:	User
Descriptions:	User click on add risk option to fill field.
Trigger:	When a user clicks the feasibility option.
Preconditions:	User should be on the home page.
Postconditions:	User fills the field and clicks the submit button to get success message.
Normal Flow:	1.click on the risk button. 2.fill the form. 3.validate form. 4.store in the database. 5.get a success message.
Alternative Flows:	3a.Errro show please fill required field.

Table 3.11: Add Risk

Use Case ID:	ID-12
Use Case Name:	Add schedule
Actors:	User
Descriptions:	User click on schedule option to fill the fields.
Trigger:	When a user clicks the add schedule option.
Preconditions:	User should be on the home page.
Postconditions:	User fills the field and clicks the submit button to get success message.
Normal Flow:	1.click on the schedule button. 2.fill the form. 3.validate form. 4.store in the database. 5.get a success message.
Alternative Flows:	3a.Errro show please fill required field.

Table 3.12: Add Schedule

3.4Entity Relational Diagram (ERD):

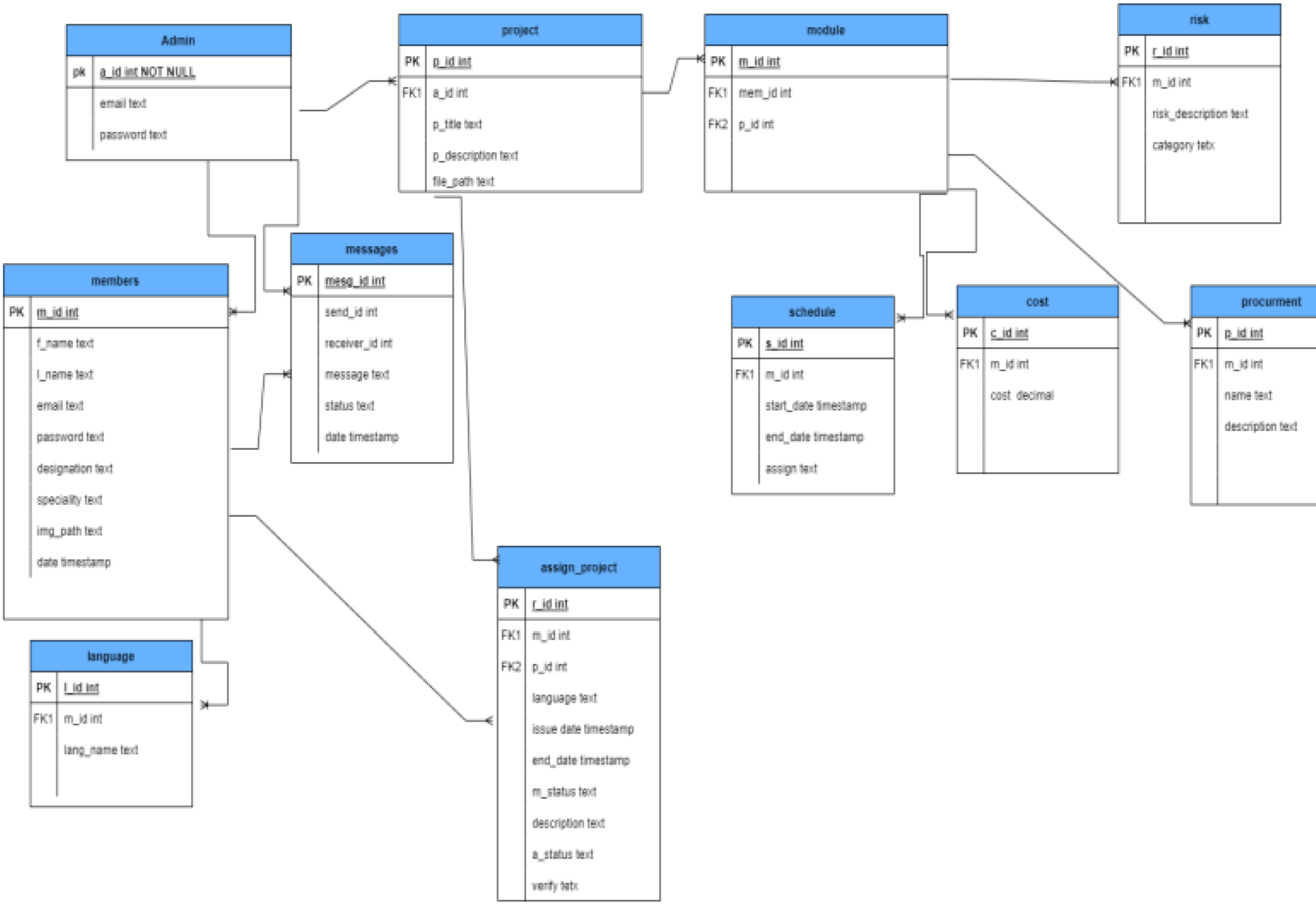

Fig 3.2: ERD

Chapter 4: Design and Architecture

In this chapter we will discuss the design and architecture of our system.

4.1. System Architecture

As system design varies from system to system, therefore user need to have the architecture view of the whole system.

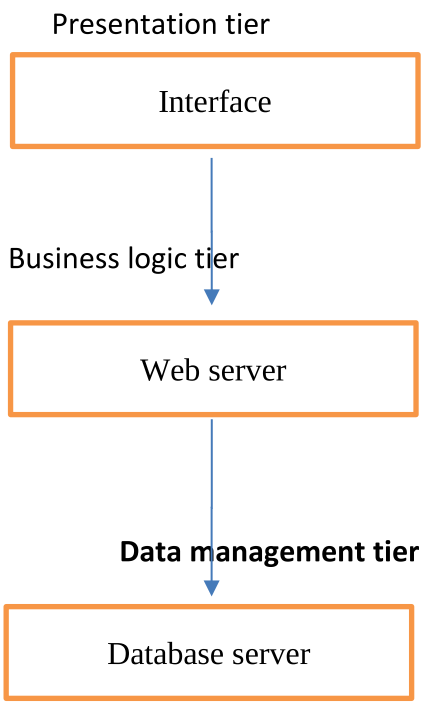

Fig 4.1: System Architecture

4.2. System Design

Systems design is the process of defining elements of a system like components, modules, architecture and their interfaces and data for a system based on the specified requirements. The purpose of the System Design process is to provide sufficient detailed data and information about the system. Following is the system design of the Product Lifecycle Management.

4.2.1 Class Diagram:

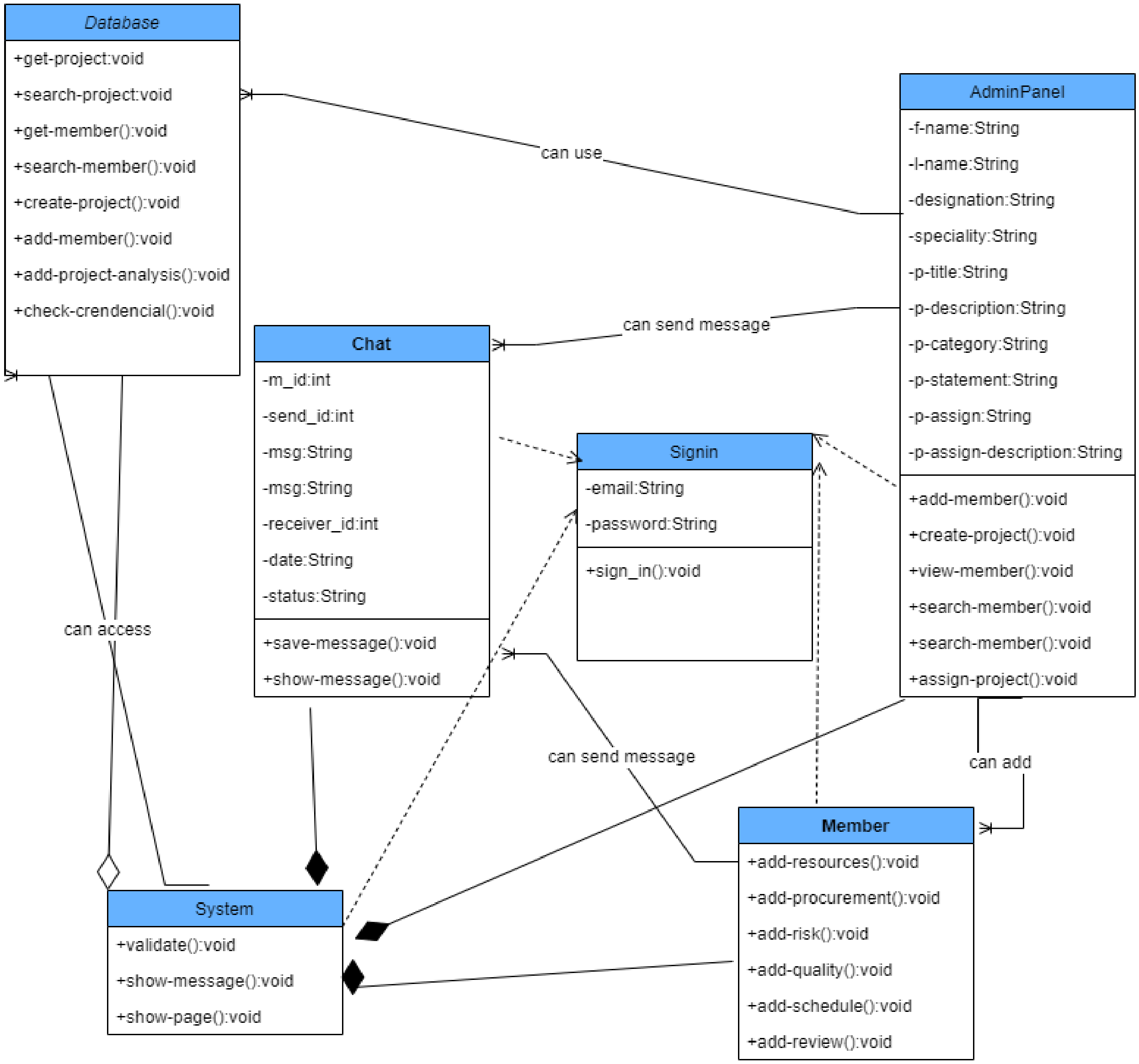

Fig 4.2: Class diagram

4.2.2 Activity Diagram:

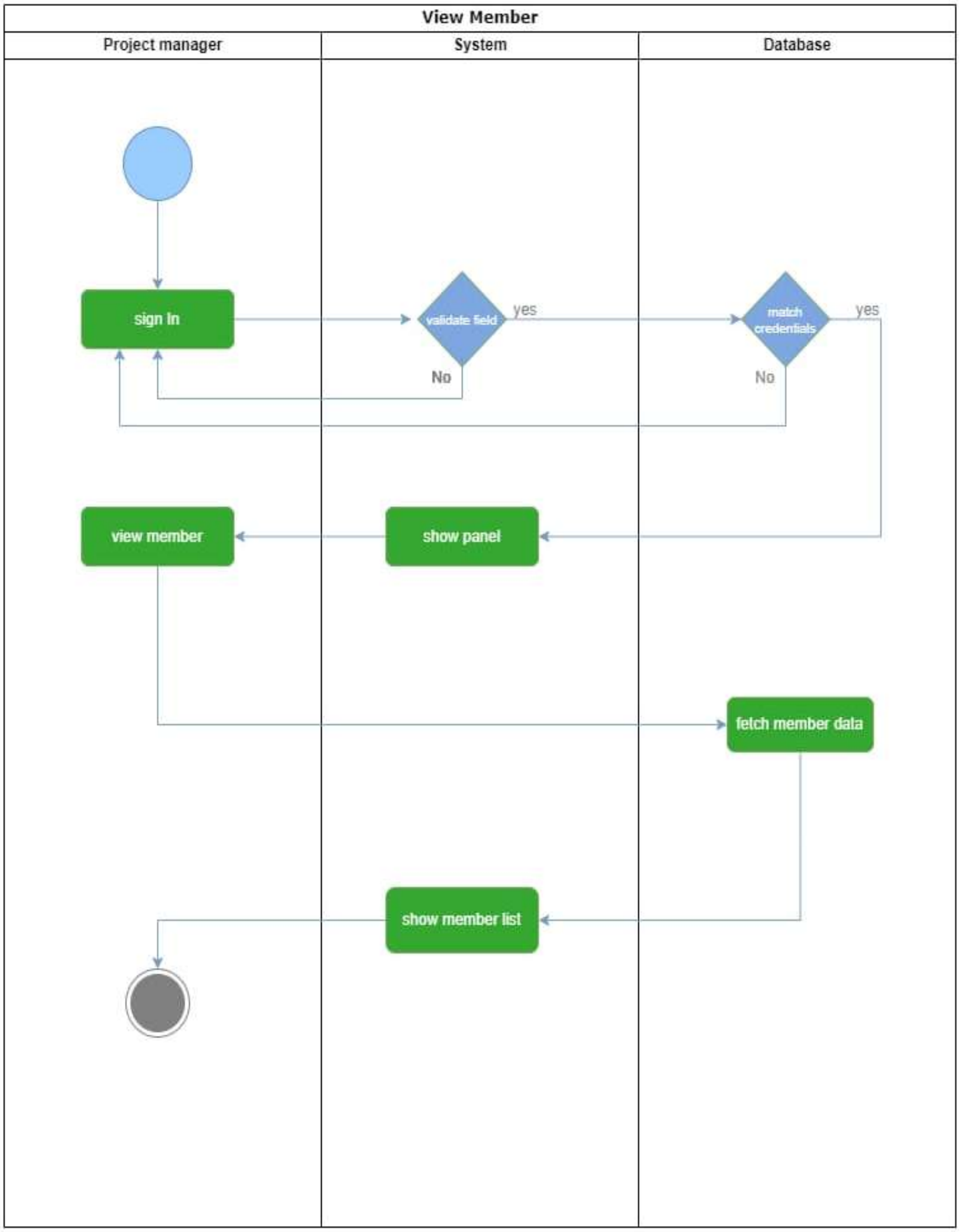

Fig 4.3: View Member Activity Diagram

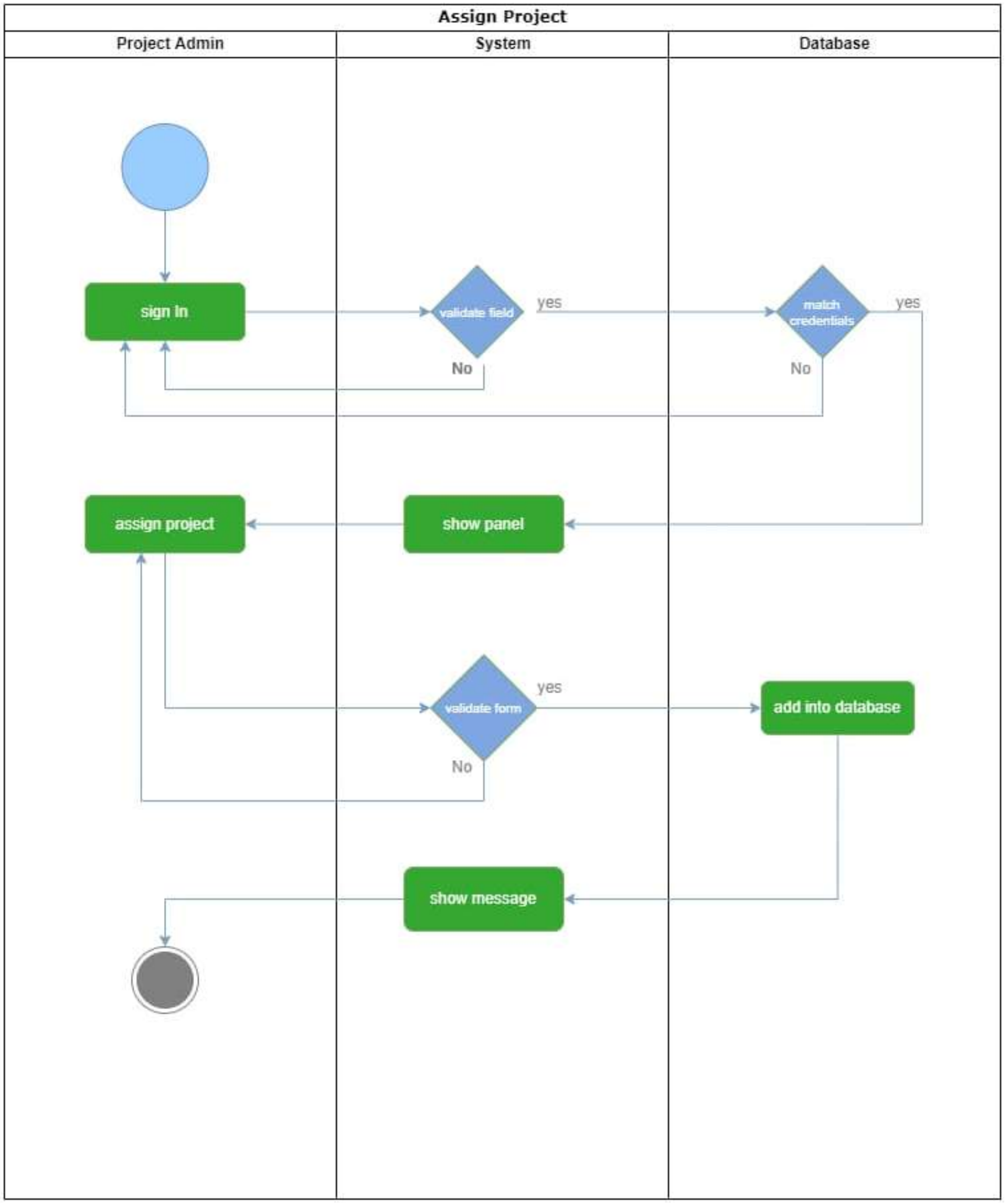

Fig 4.4: Assign Project Activity Diagram

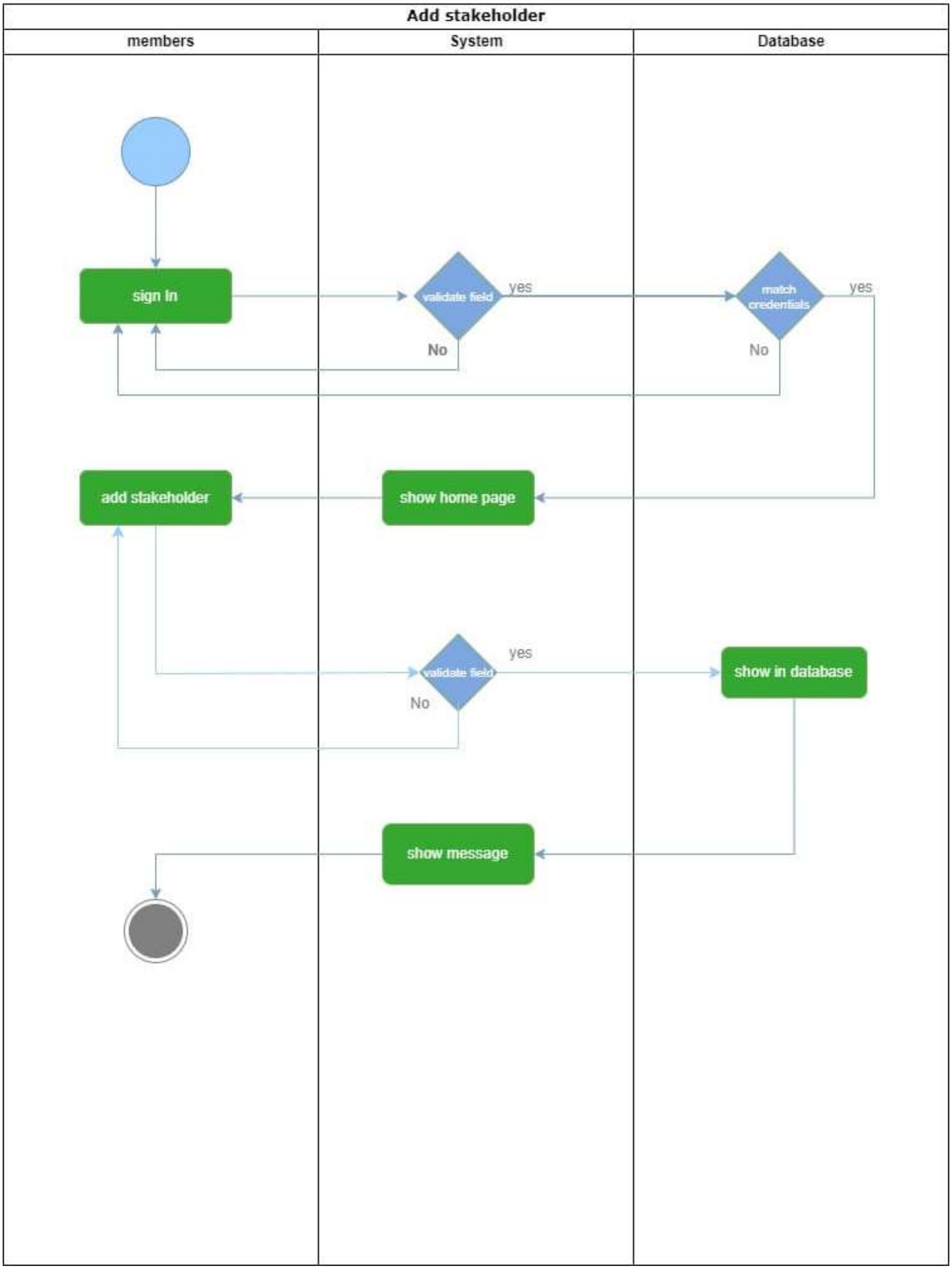

Fig 4.5: Add Stakeholder Activity Diagram

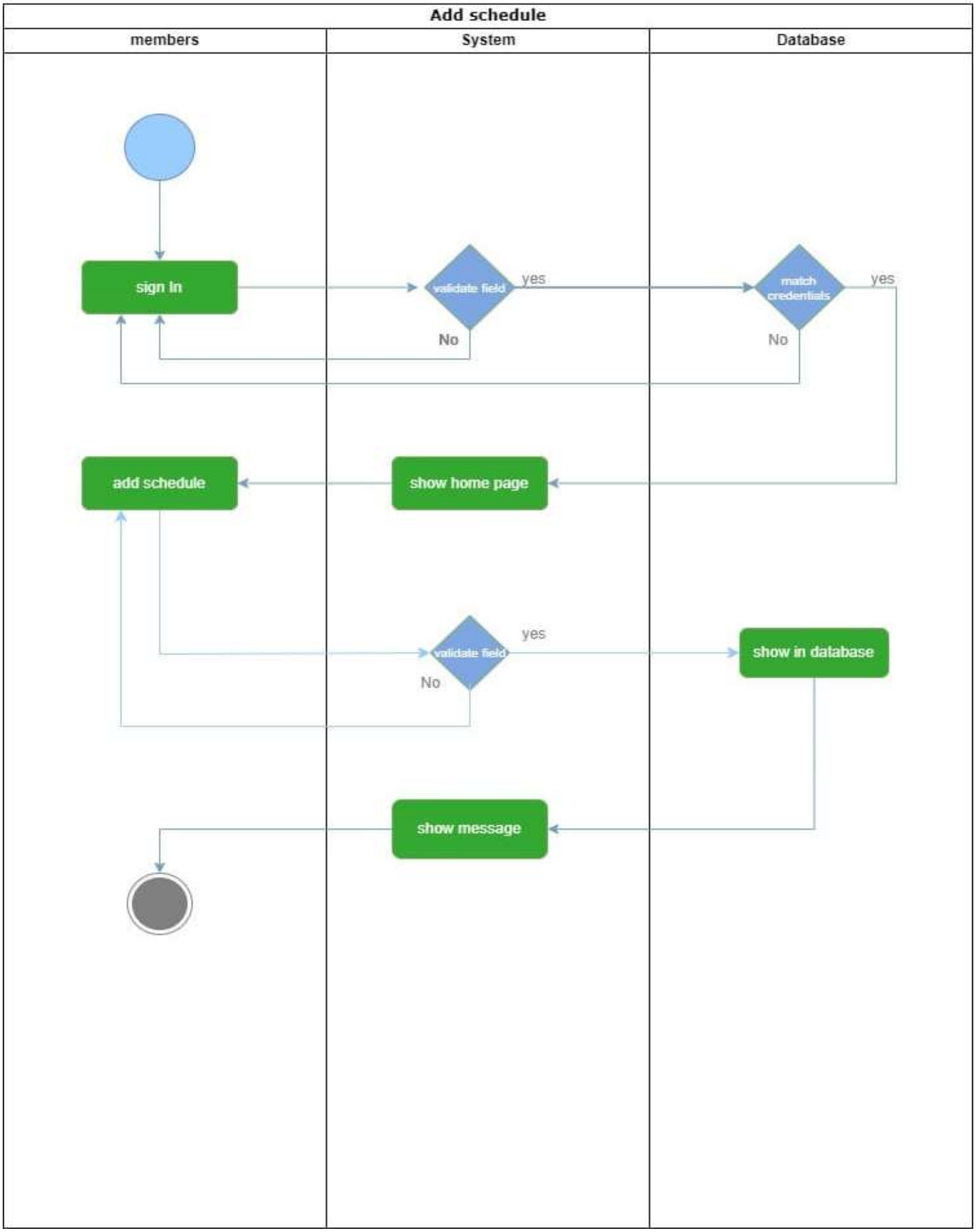

Fig 4.6: Add Schedule Activity Diagram

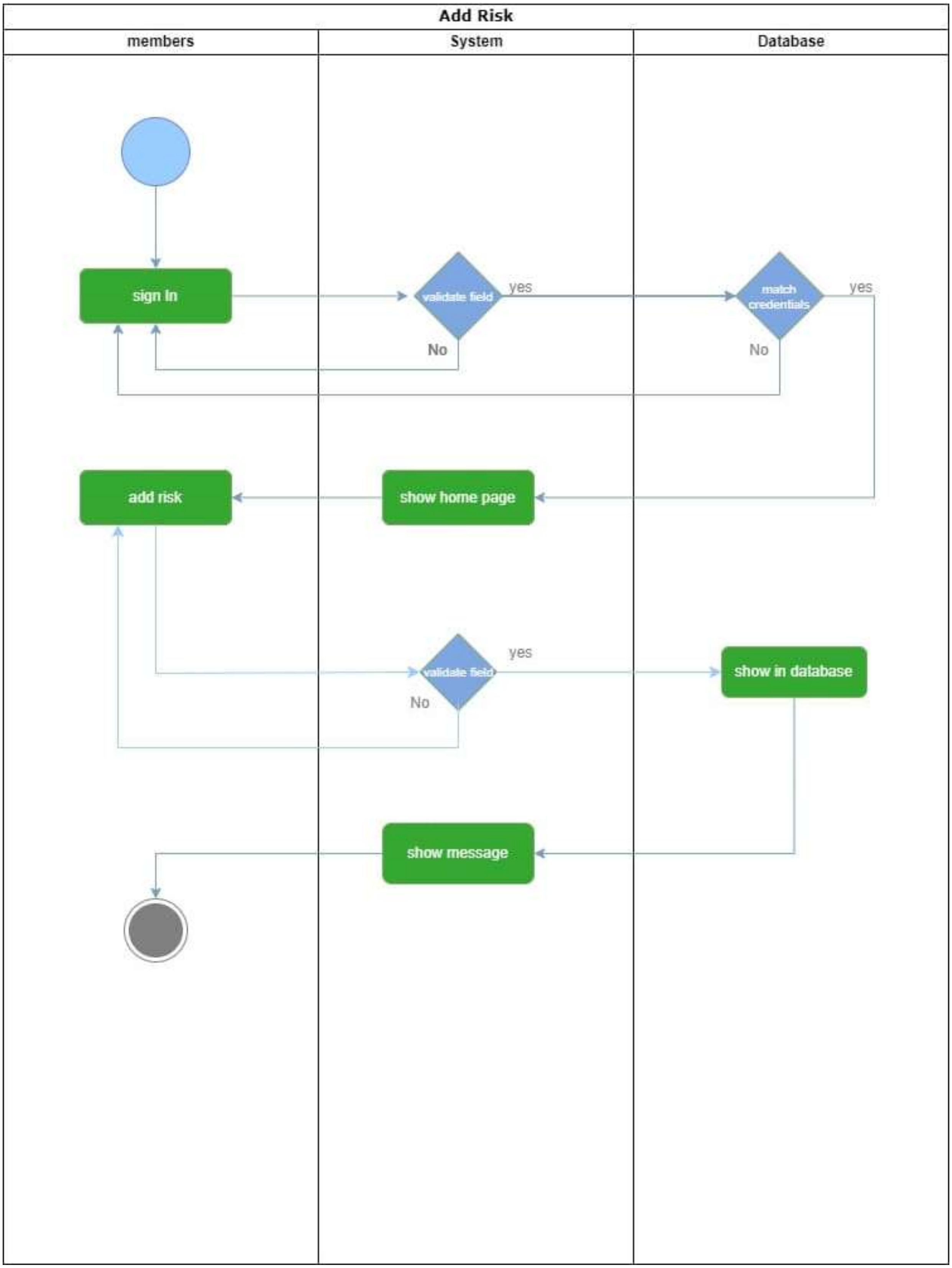

Fig 4.7: Add Risk Activity Diagram

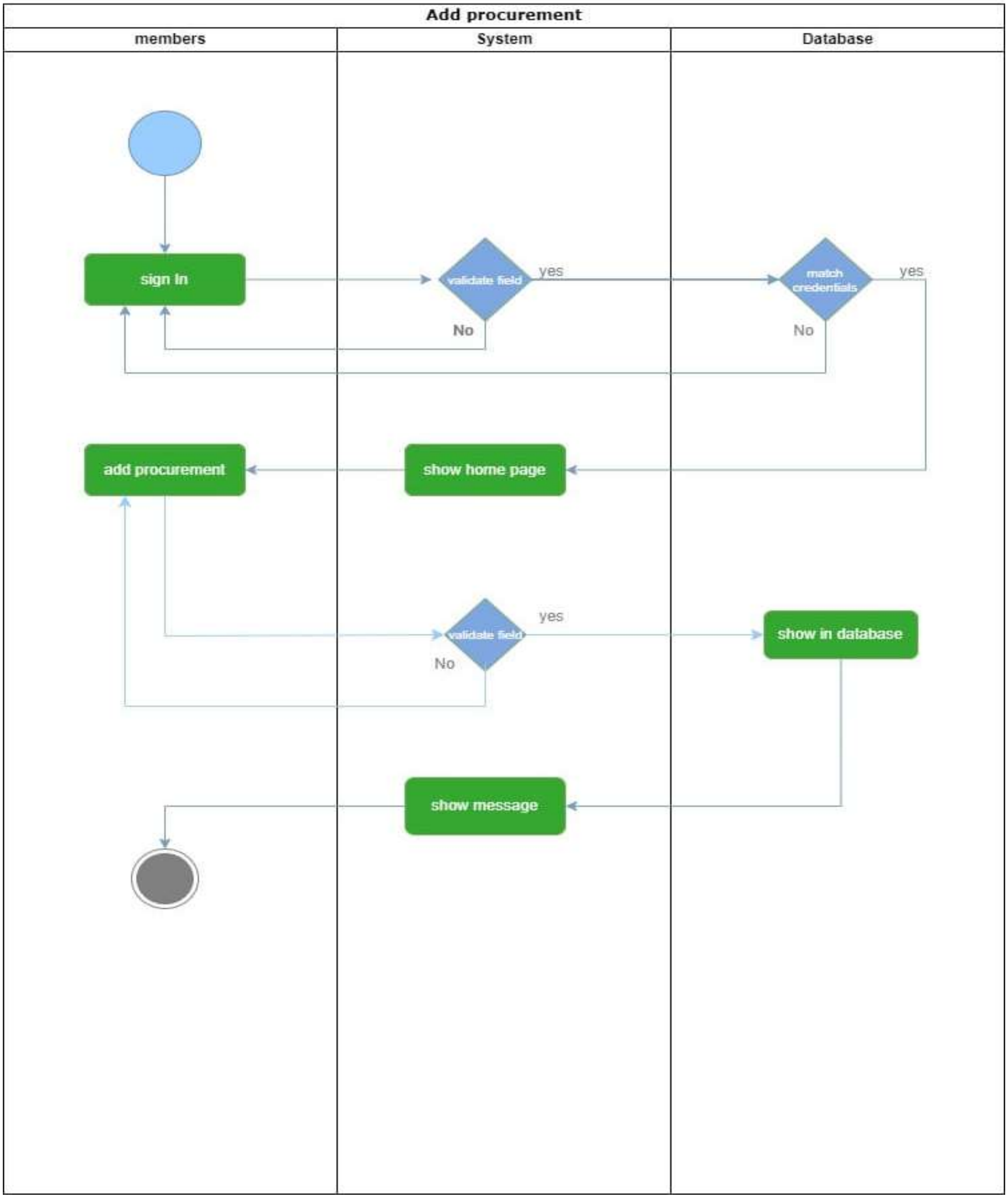

Fig 4.8: Add Procurement Activity Diagram

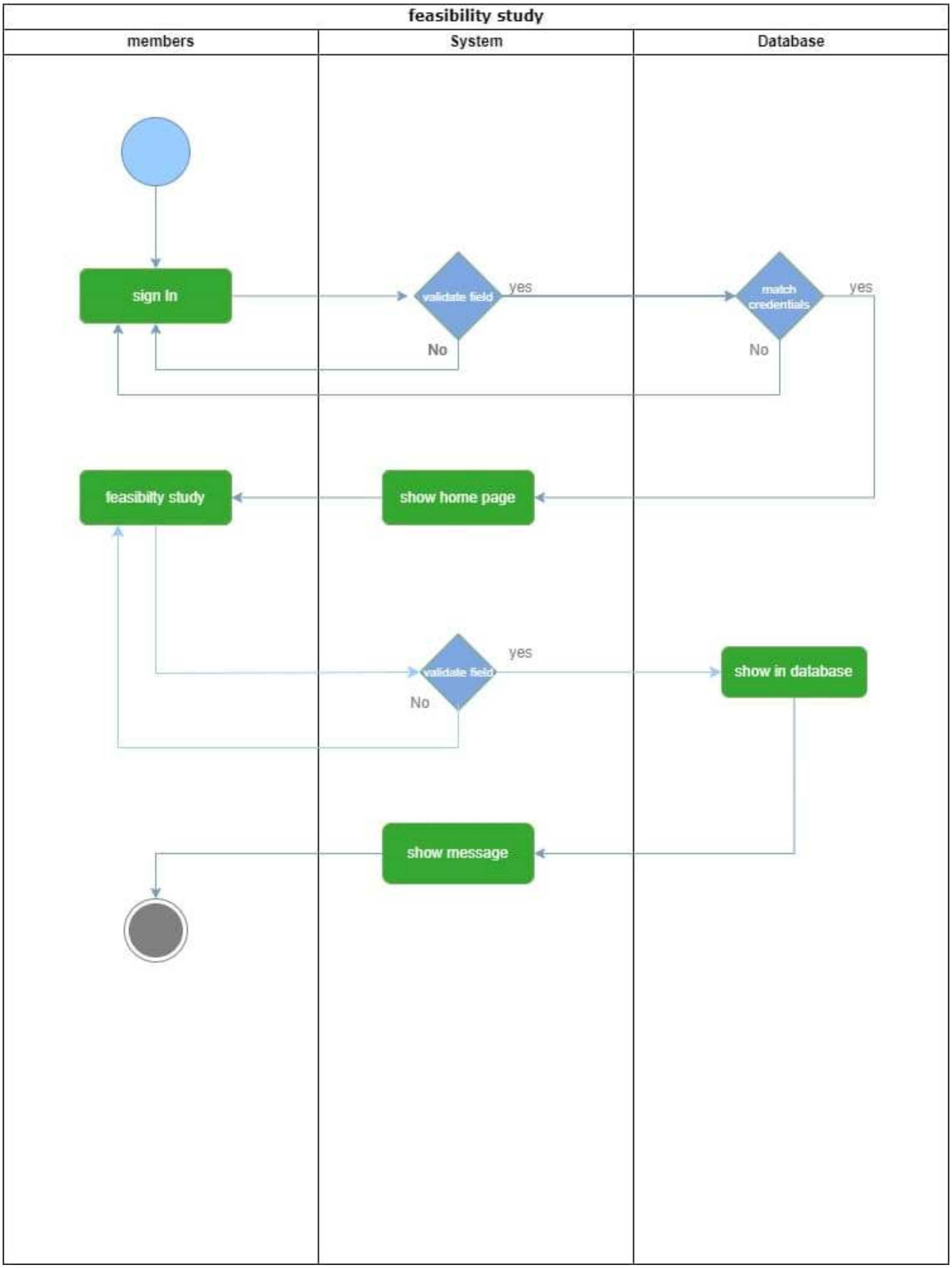

Fig 4.9: Feasibility Study Activity Diagram

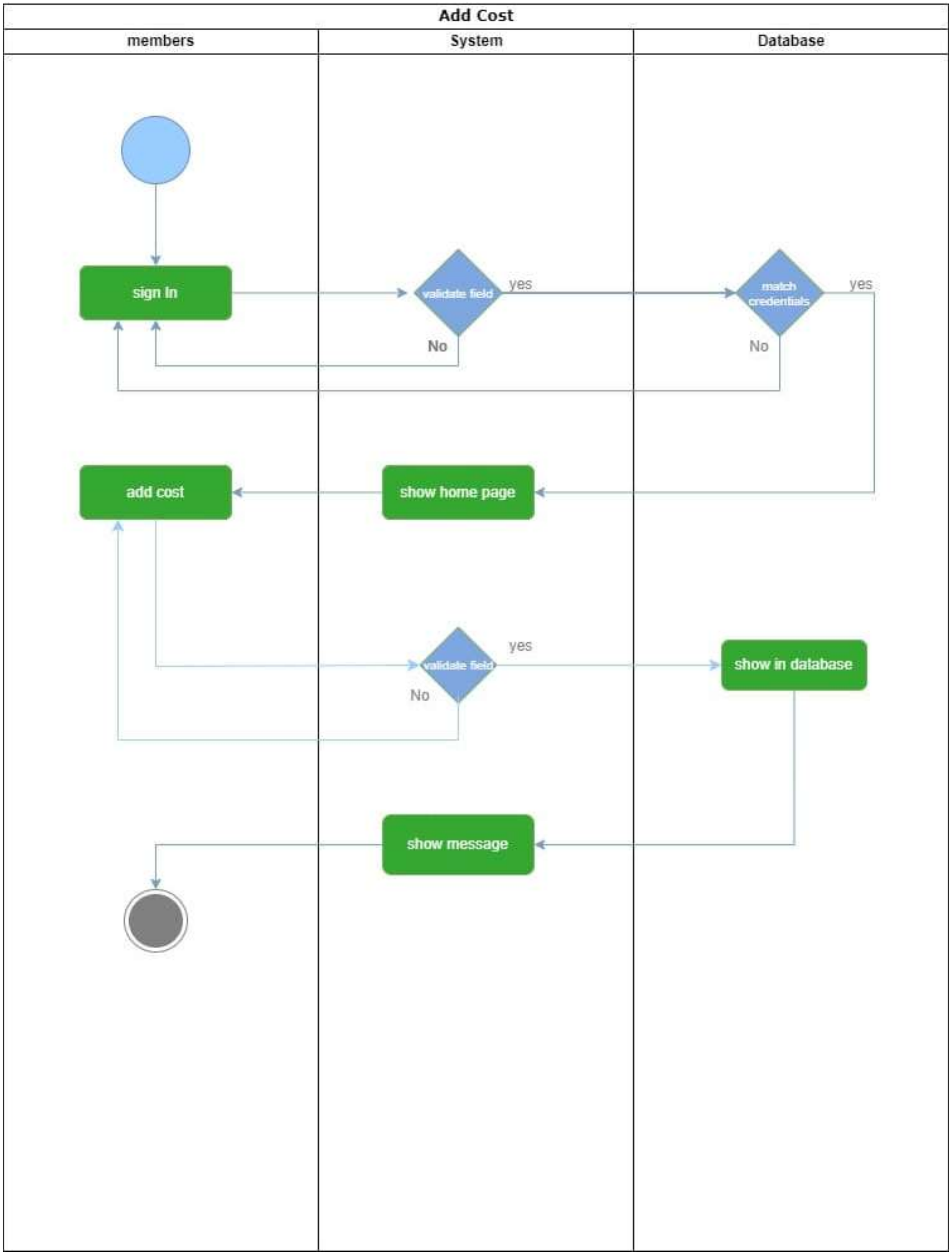

Fig 4.10: Add Cost Activity Diagram

4.2.3 Sequence Diagram:

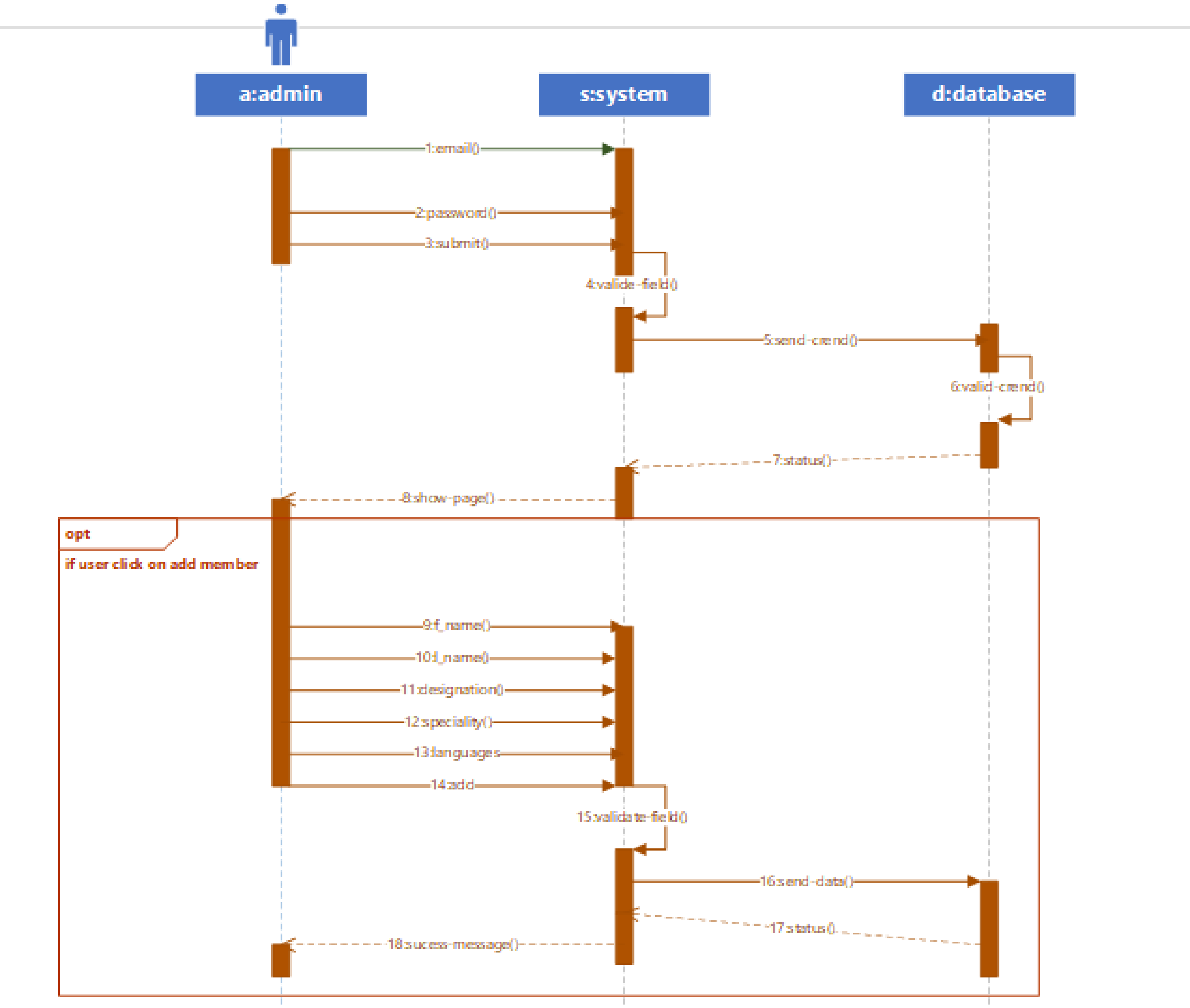

Fig 5.1: Sequence Diagram Add Member

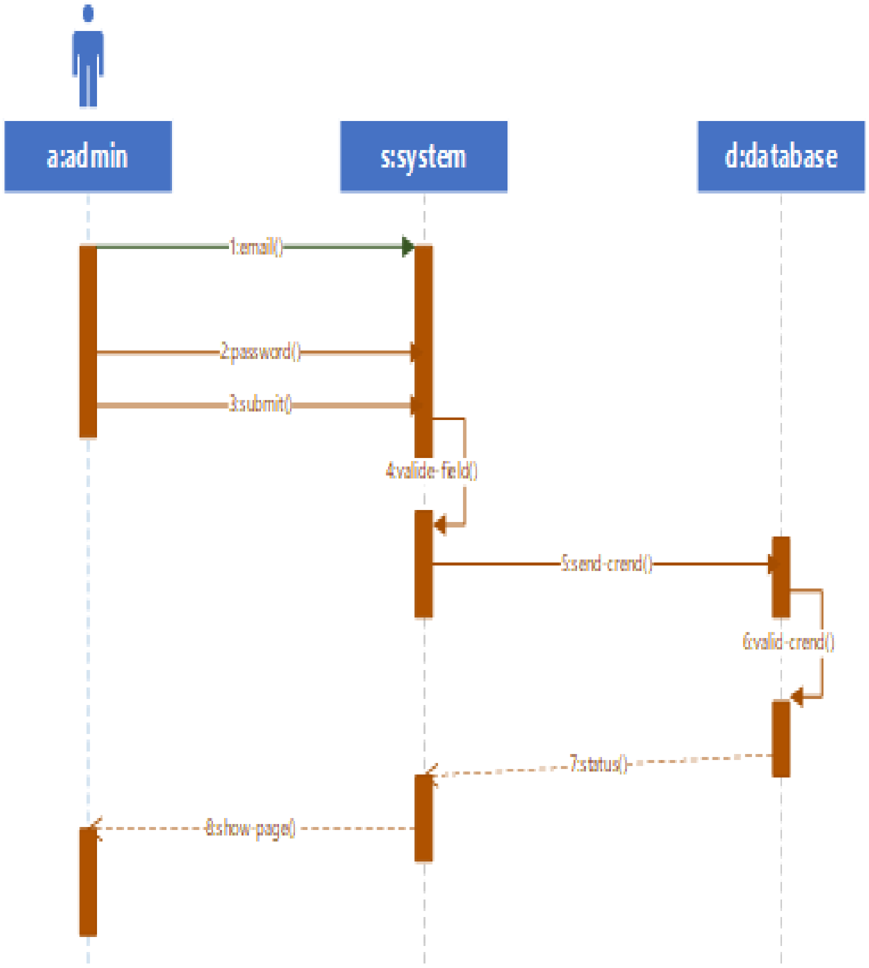

Fig 5.2: Sequence Diagram sign in

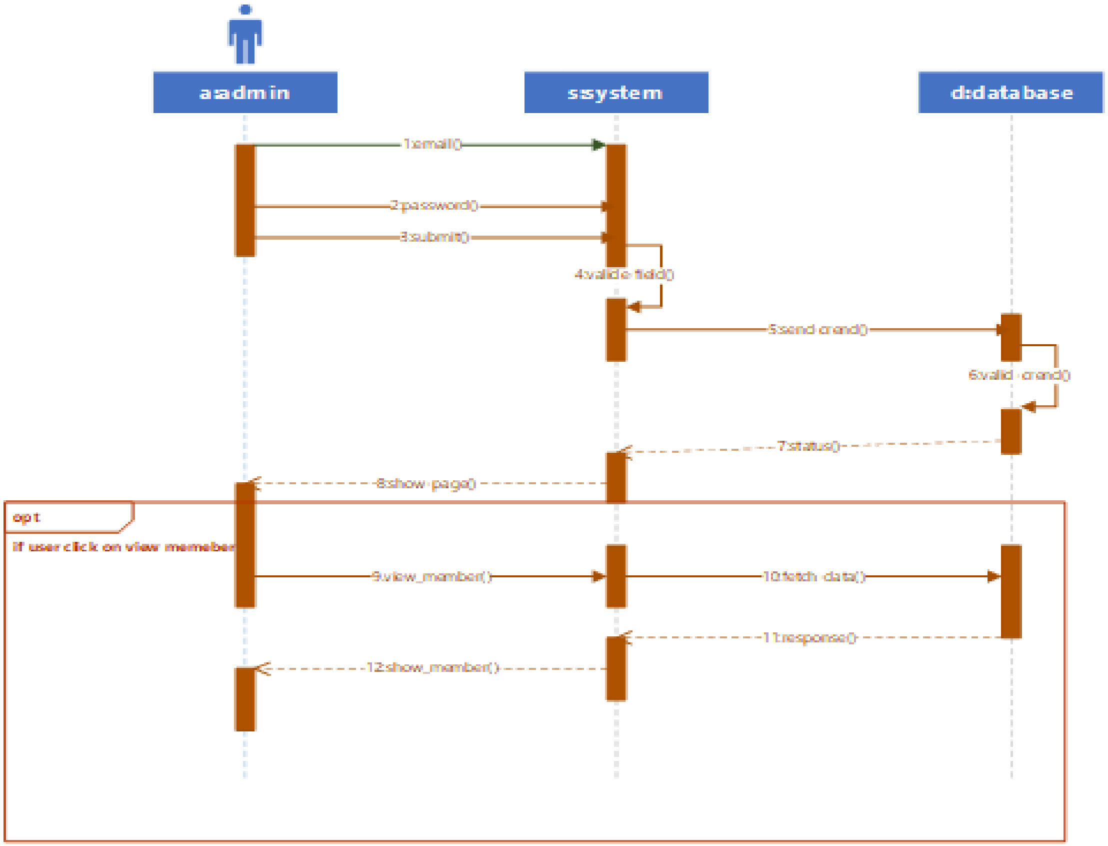

Fig 5.3: Sequence Diagram View Member

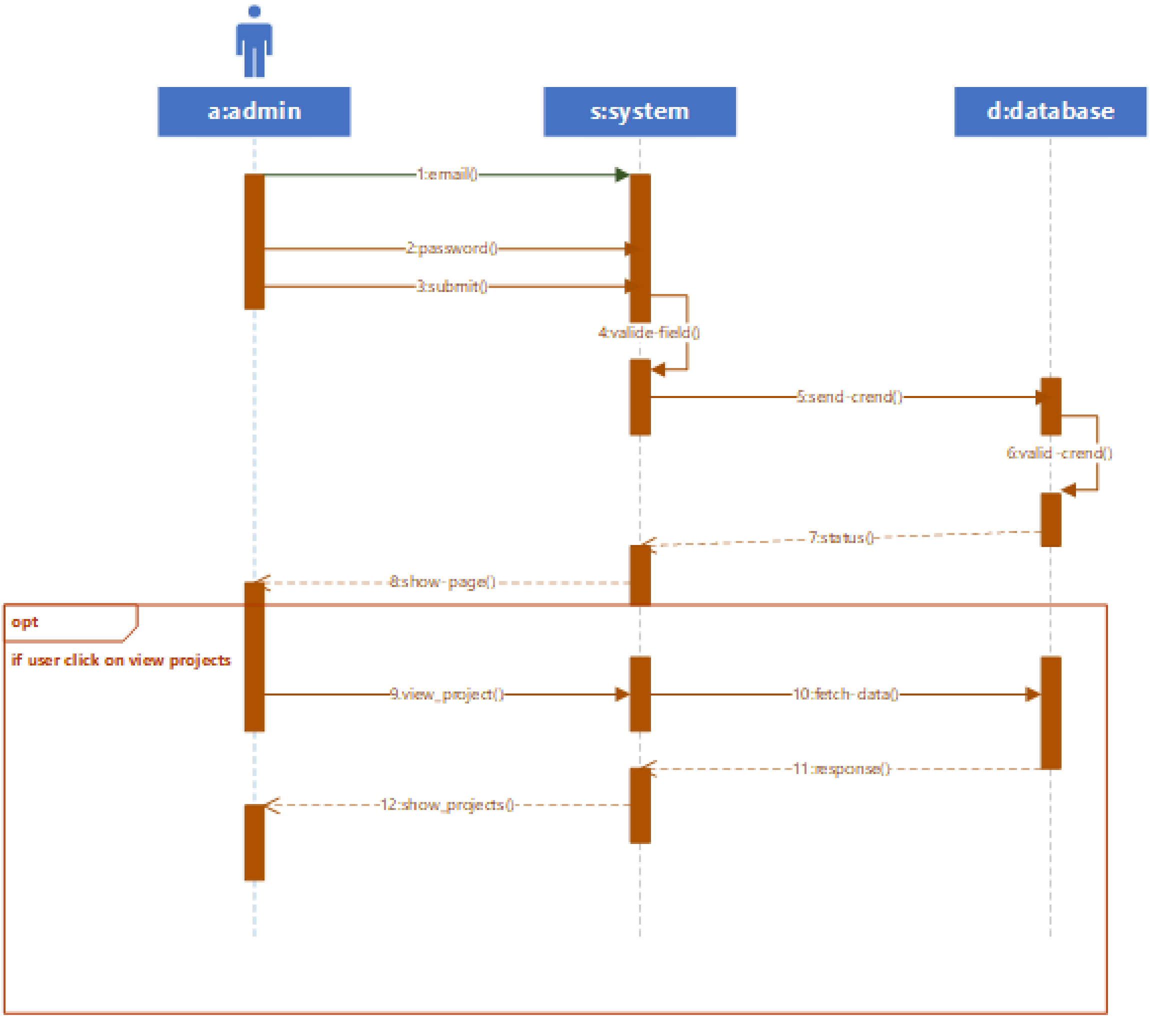

Fig 5.4: Sequence Diagram View Project

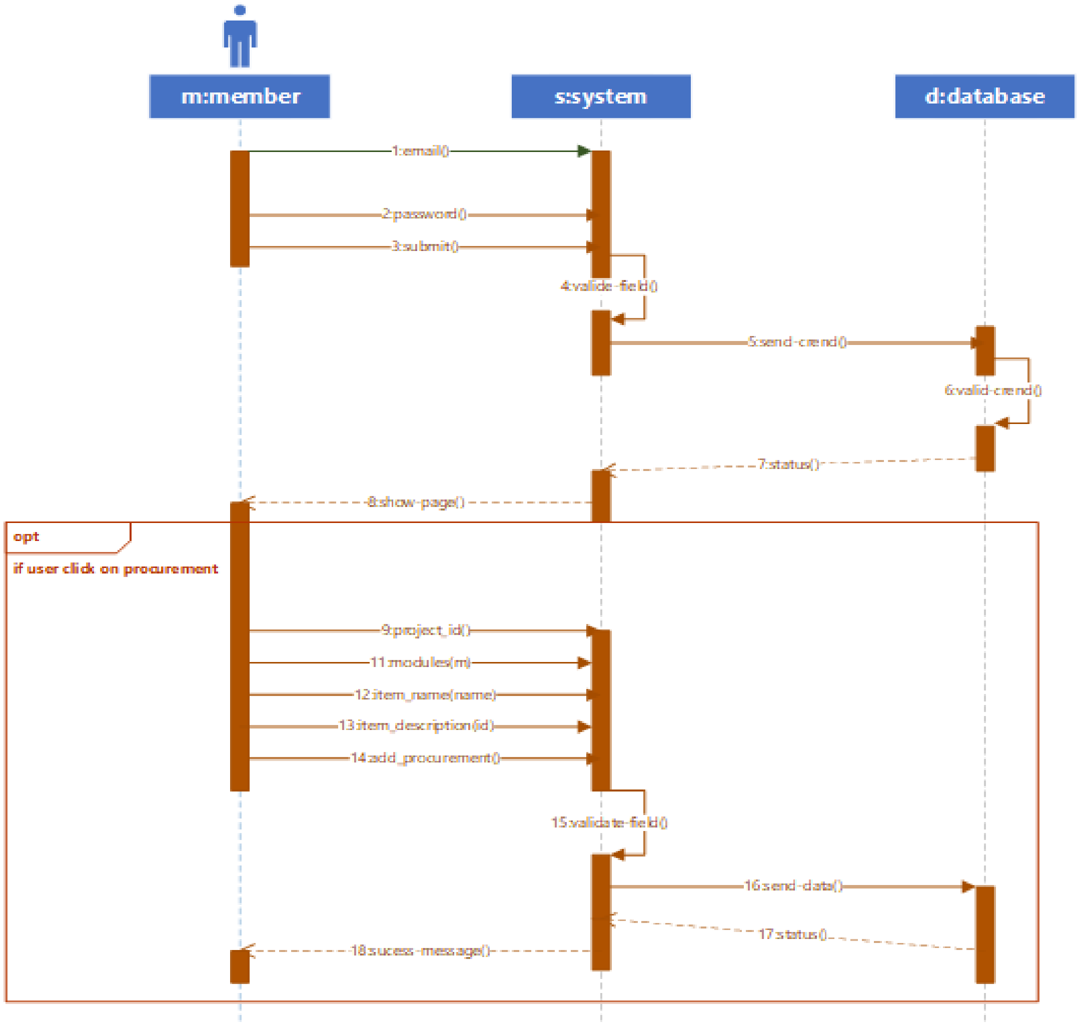

Fig 5.5: Sequence Diagram Add Procurement

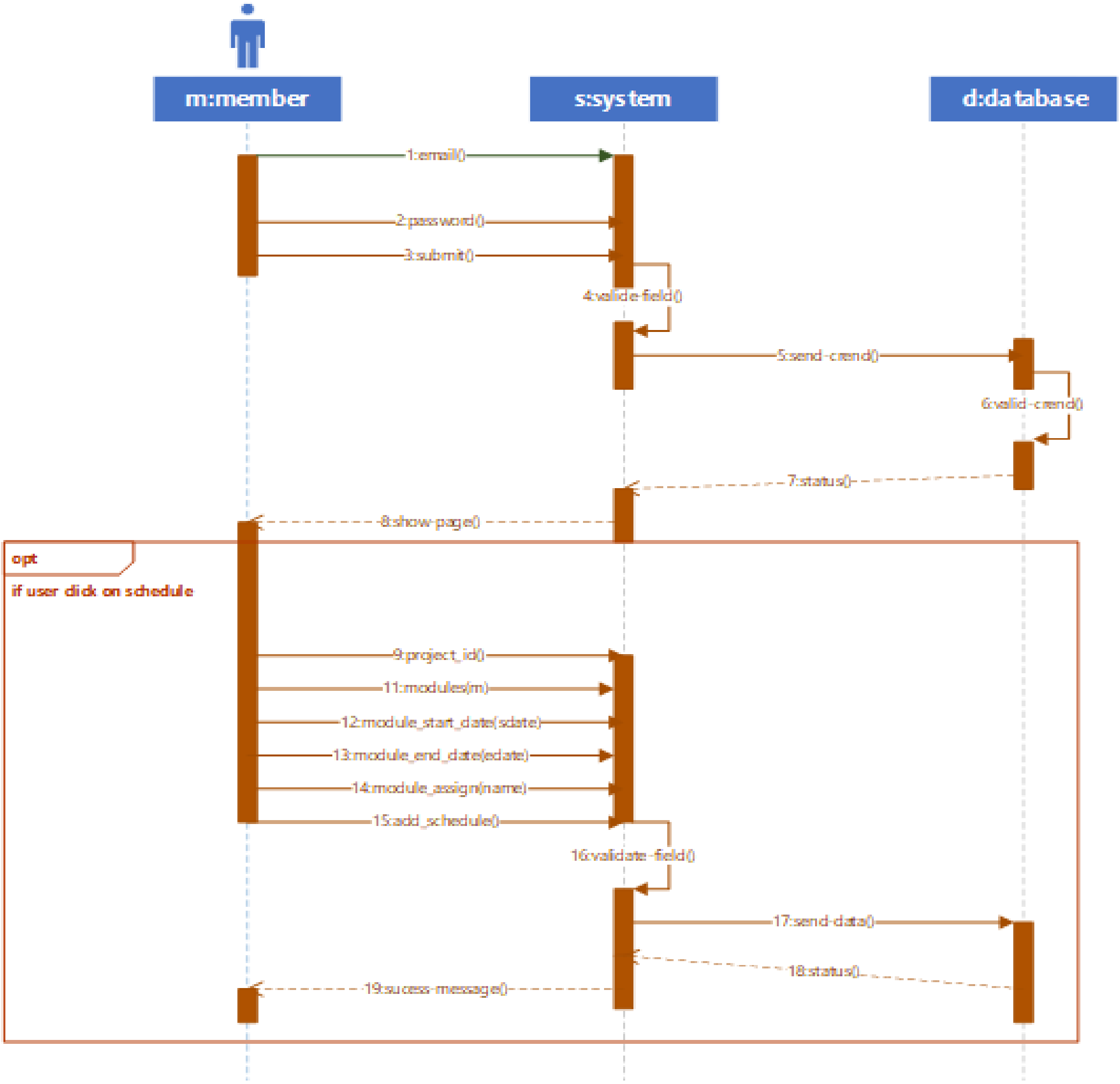

Fig 5.6: Sequence Diagram Add Schedule

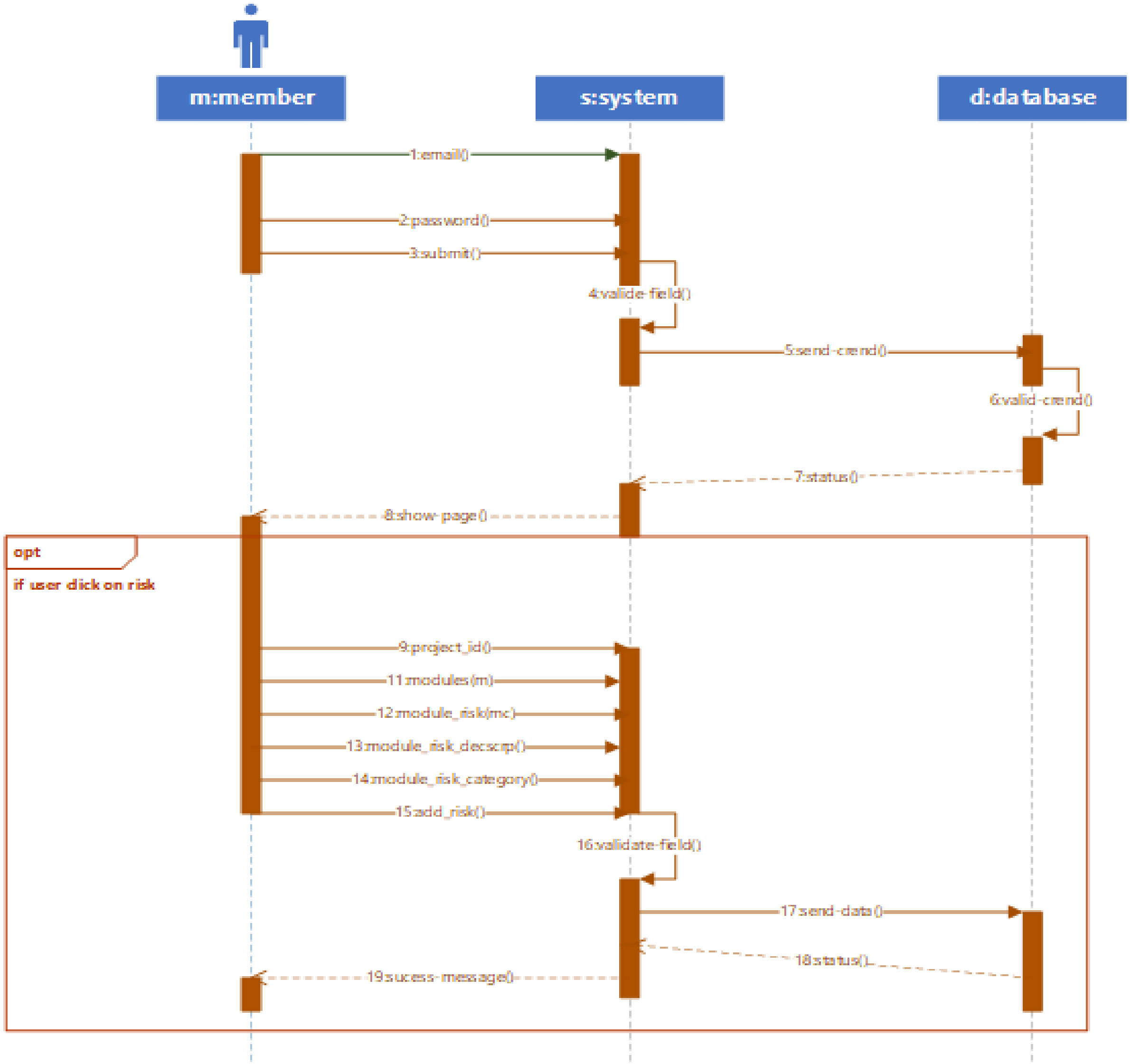

Fig 5.7: Sequence Diagram Add Risk

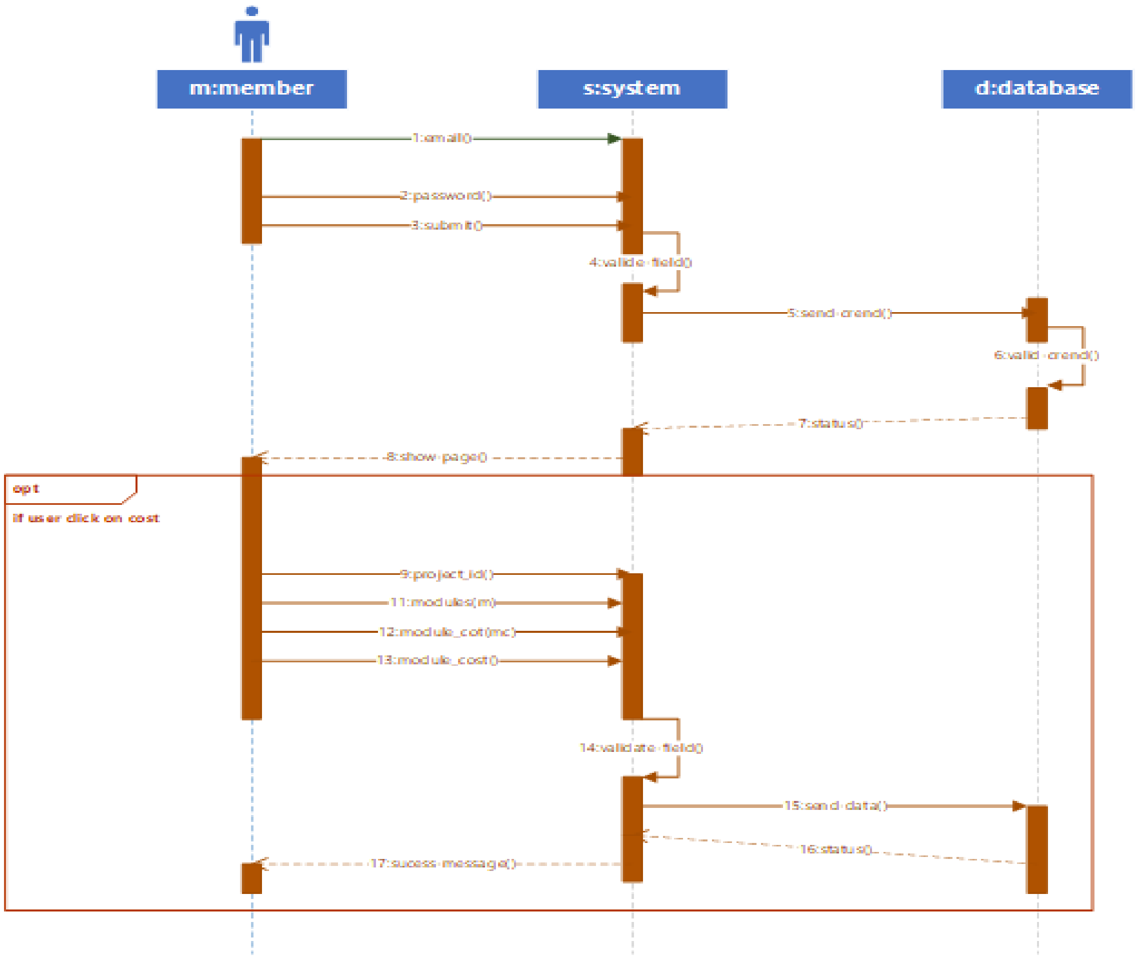

Fig 5.8: Sequence Diagram Add Cost