



Sri Lanka Institute of Information Technology

Project Report

GadgetBadget

Programming Application Framework Project 2021

Project ID: 14

Submitted by:

Student Registration Number	Name
IT18077698	Thennakoon T.M.B.C.K
IT19015422	Perera B.A.A.W.S
IT19015040	Rasuni Wageesha H.A
IT19120430	Gamage M.S.S.K

Table of Content

1. Cover page	01
2. Table of contents	02
3. Members' details	03
4. Clickable link of GitHub	04
5. SE methodology/methods	04
6. Time schedule (Gantt chart)	04
7. Requirements	
a. Stakeholder analysis (onion diagram)	05
b. Requirements analysis (Functional, Non-functional, Technical requirements)	05
c. Requirements modelling (Use case diagram)	06
8. System's overall design	
a. Overall architecture	07
b. Overall DB design	07
c. Activity diagrams	08
9. Individual sections	8-26
10. System's integration details	
11. References	
12. Appendix	

Introduction

This project is about an online platform to sell innovative projects of young researchers and also this will allow researchers to get funds from investors. Using those investments researchers can develop their new inventions. That is the main idea of this GadgetBadget system.

There are four main functions that we hope to develop in this system. These are User management, Product management, Fund management and Research document management. This system will manage all the above-mentioned features with all the crud functions. This will definitely help for young innovators to achieve their goals.

Members' details

Student ID	Student Name	Micro Service	Work allocated
IT18077698	Thennakoon T.M.B.C.K	Product Management	<ul style="list-style-type: none">• Add products• Delete products• Update products• View products
IT19015422	Perera B.A.A.W.S	Fund Management	<ul style="list-style-type: none">• Add funds• Delete funds• Update funds• View funds
IT19015040	Rasuni Wageesha H.A	Research project Management	<ul style="list-style-type: none">• Add Research project• Delete Research project• Update Research project• View Research project
IT19120430	Gamage M.S.S.K	User Management	<ul style="list-style-type: none">• User Registration• View, Update and Delete Users

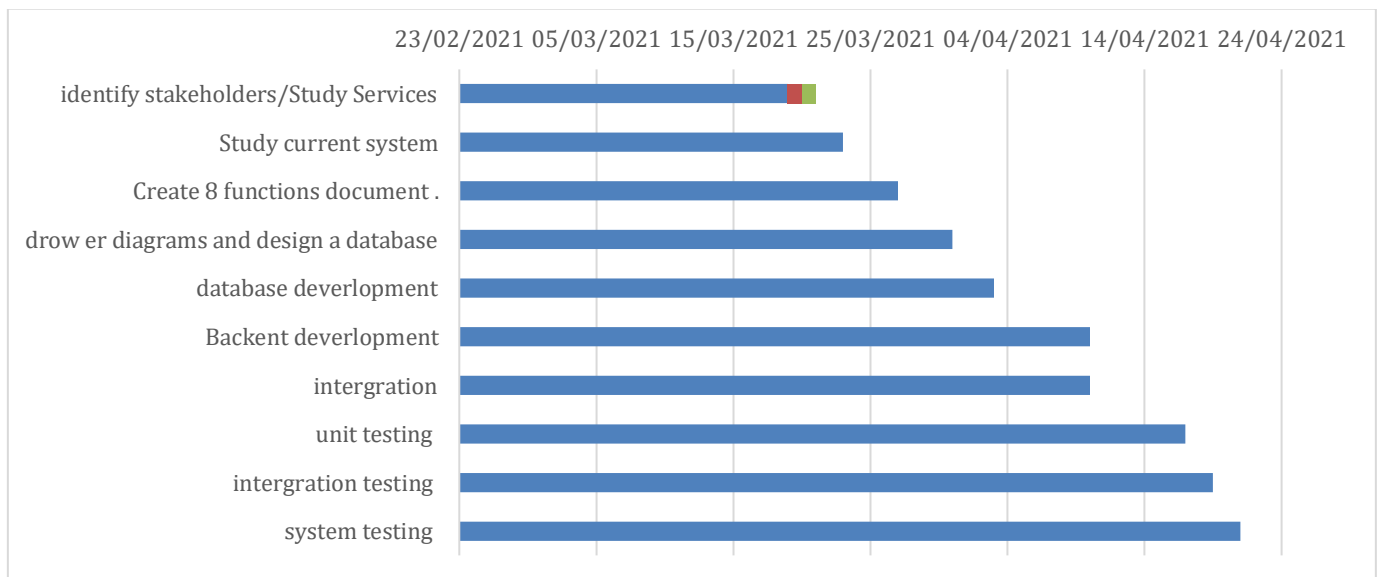
Clickable link

https://github.com/Buddhi1998/PAF_Project

SE Methodology

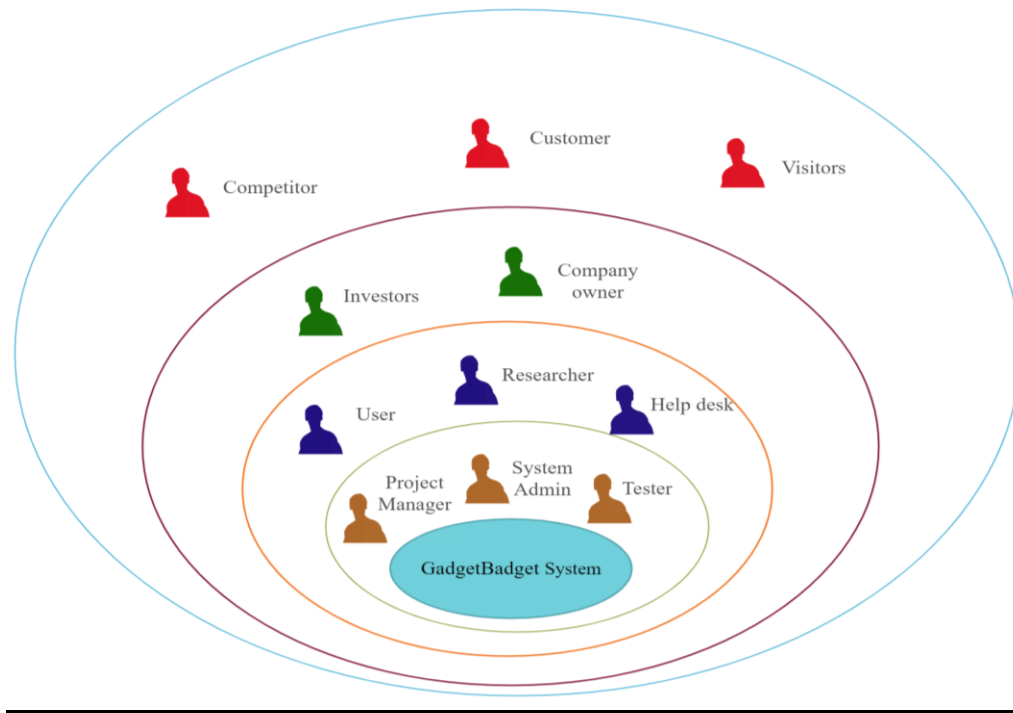
In this project waterfall method was used to develop the system. We selected this methodology because it is Simple enough to handle as model is rigid and easy to understand and functional. In waterfall model it is easy for testing and analysis. Our project was a small project. But we had a limited time period to develop the system. Waterfall method was selected as it saves significant amount of time.

Time schedule (Gantt chart)



Requirements

a. Stakeholder analysis (Onion diagram)



b. Requirements analysis

a) Functional Requirements

- I. Product Management – Add, delete, update and view Products
- II. Fund Management – Add, delete, update and view Funds
- III. Research document Management - Add, delete, update and view Research details
- IV. User Management – Add, delete, update and view Users

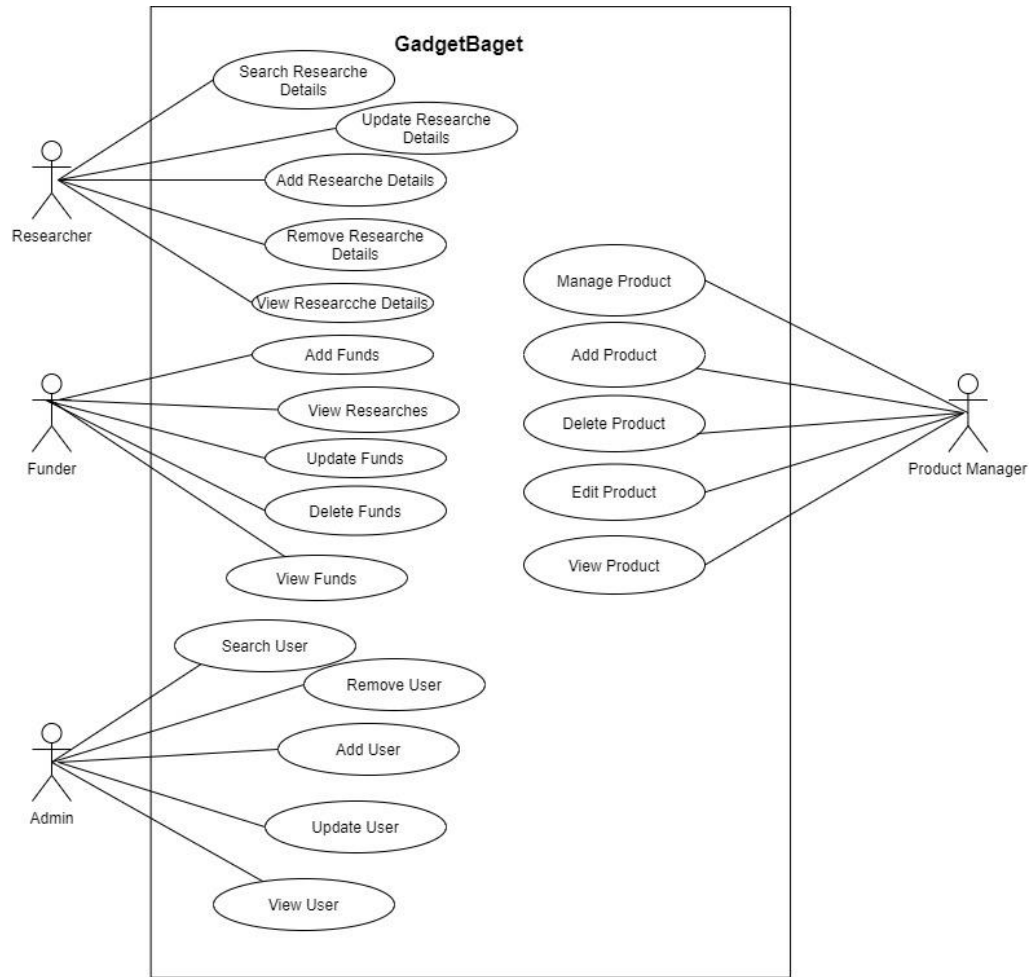
b) Nonfunctional Requirements

- Quality
- Efficiency
- Maintainability
- Reliability
- Privacy
- Accessibility
- Compatibility

c) Technical requirements

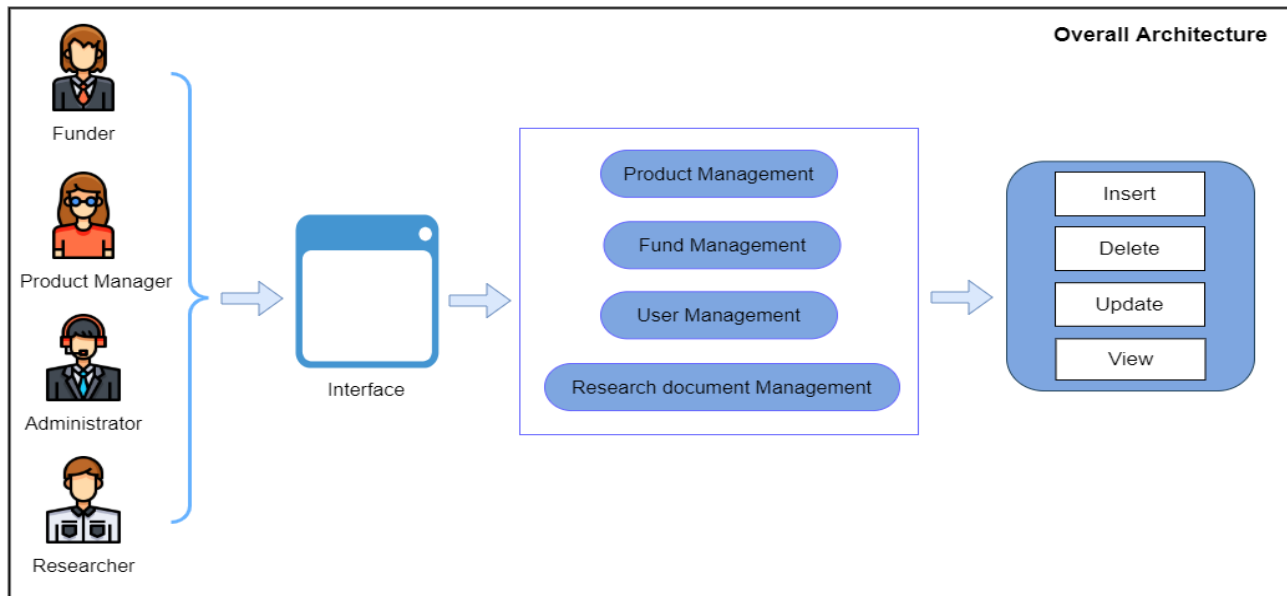
- The backend is developed using **JAX-RS, Jersey, Java** via Eclipse IDE. PhpMyAdmin is used to make database connection.

c. Requirements modelling (Use case diagram)

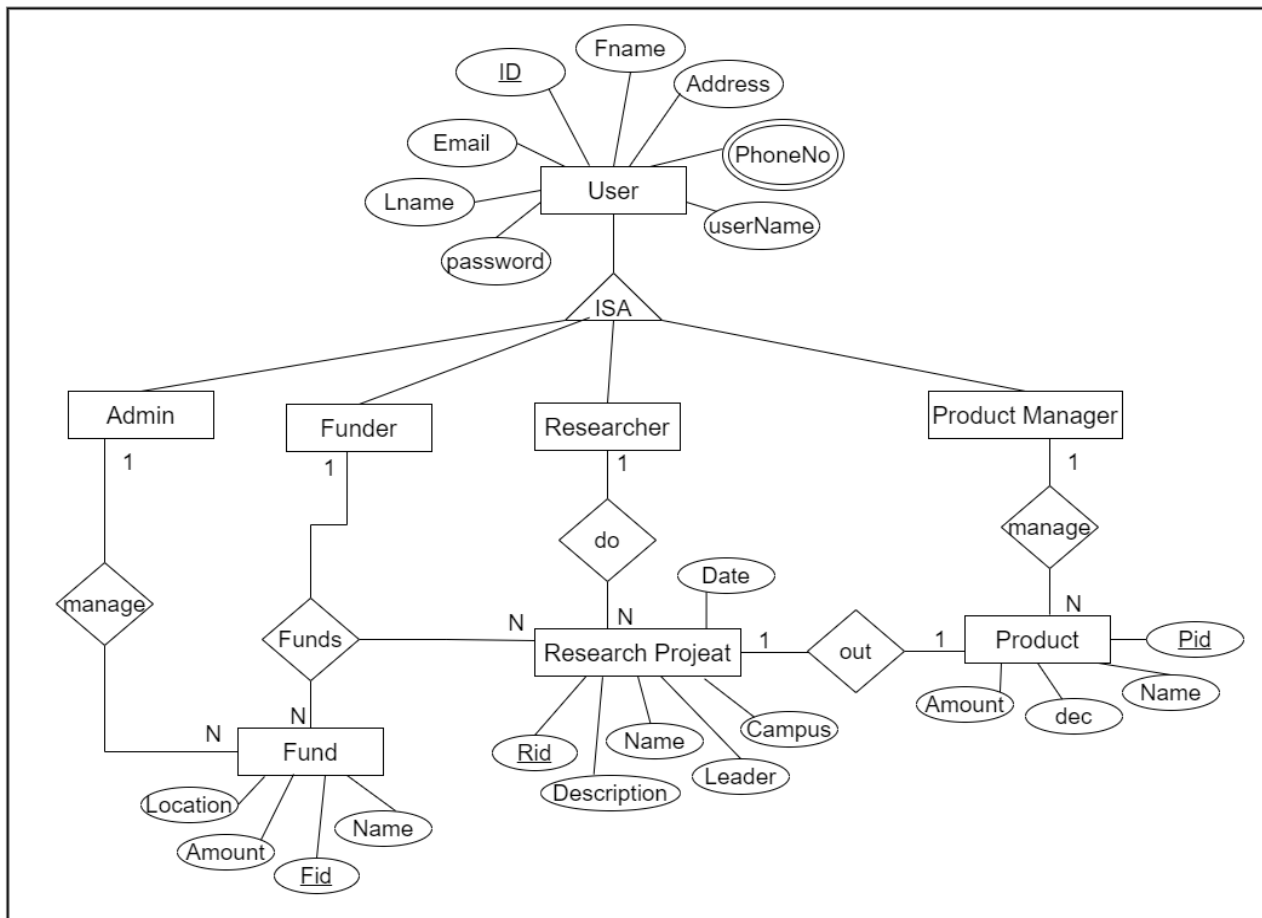


System's overall design

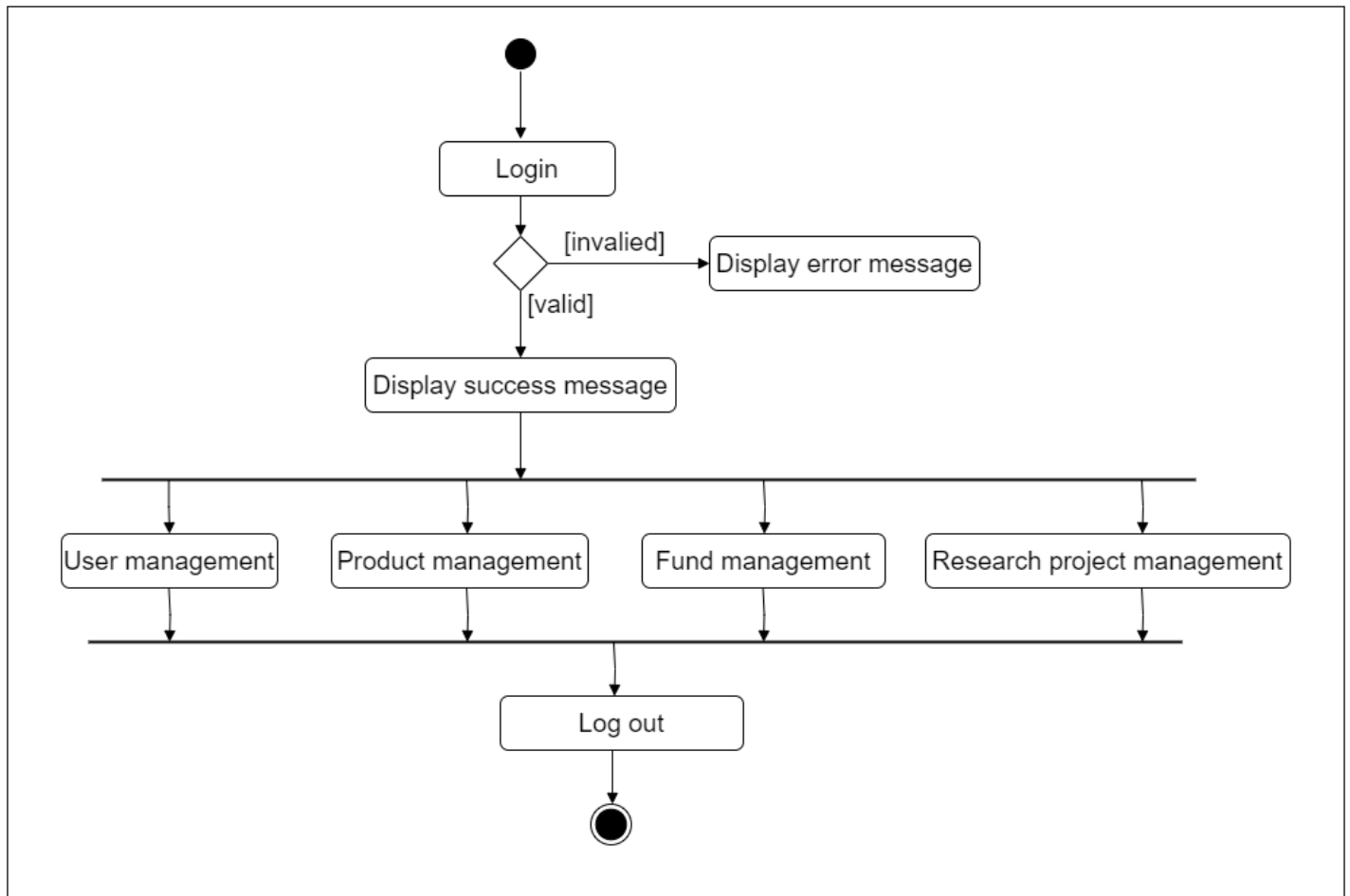
a) Overall architecture



b) Overall DB design (ER)



c) Activity diagrams



Individual sections

1) Thennakoon T.M.B.C.K - IT18077698

URL: https://github.com/Buddhi1998/PAF_Project/tree/main/IT18077698_ProdMNG2

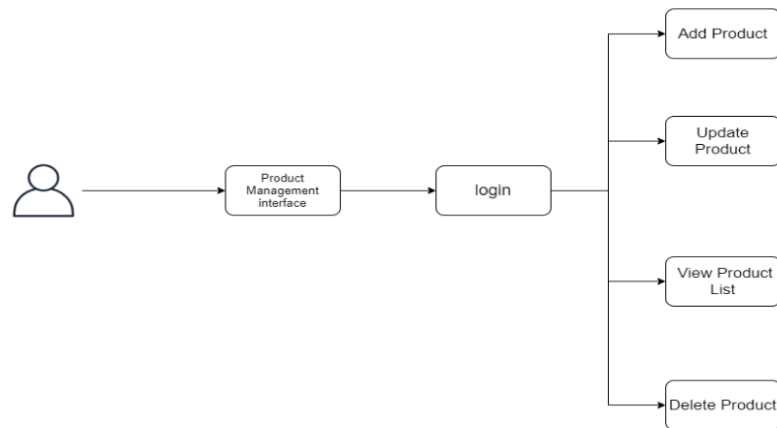
GitHub: IT18077698_ProdMNG2

Product Management

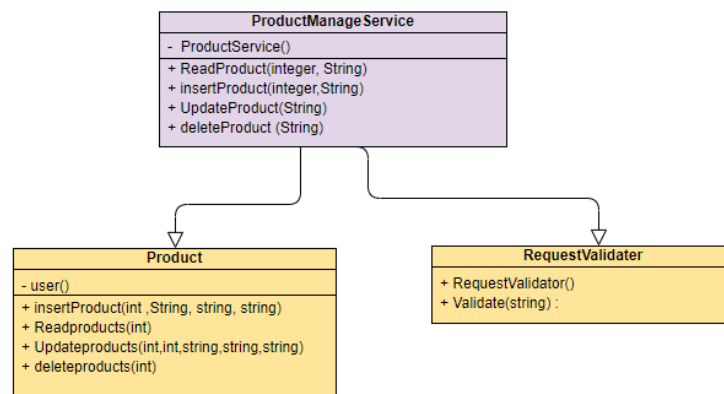
All the Products Of this system is manage by the Product Manager and he is the actor of this part. Product Manager can handle add, update, delete, view Products.

1. Service design

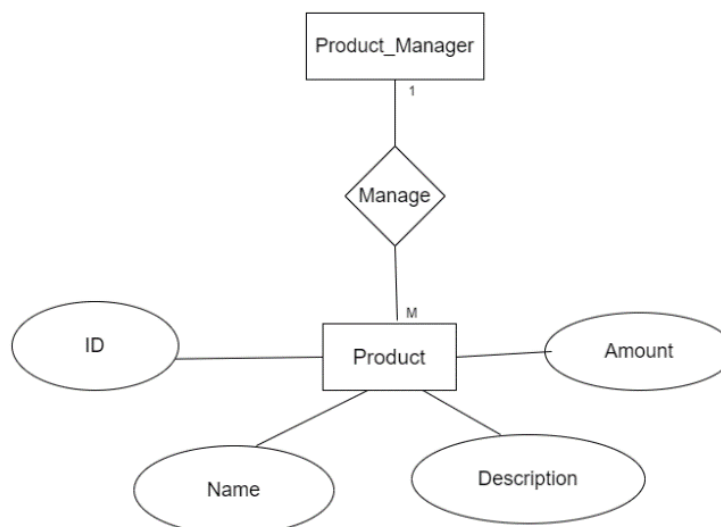
API of the service



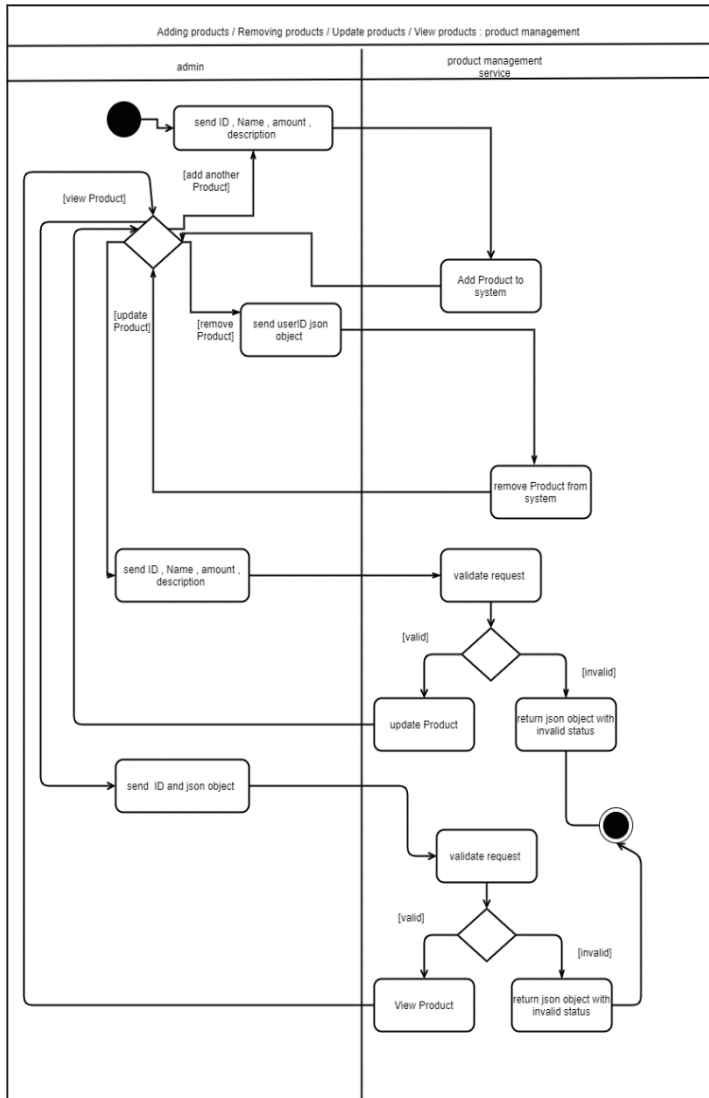
Class Diagram



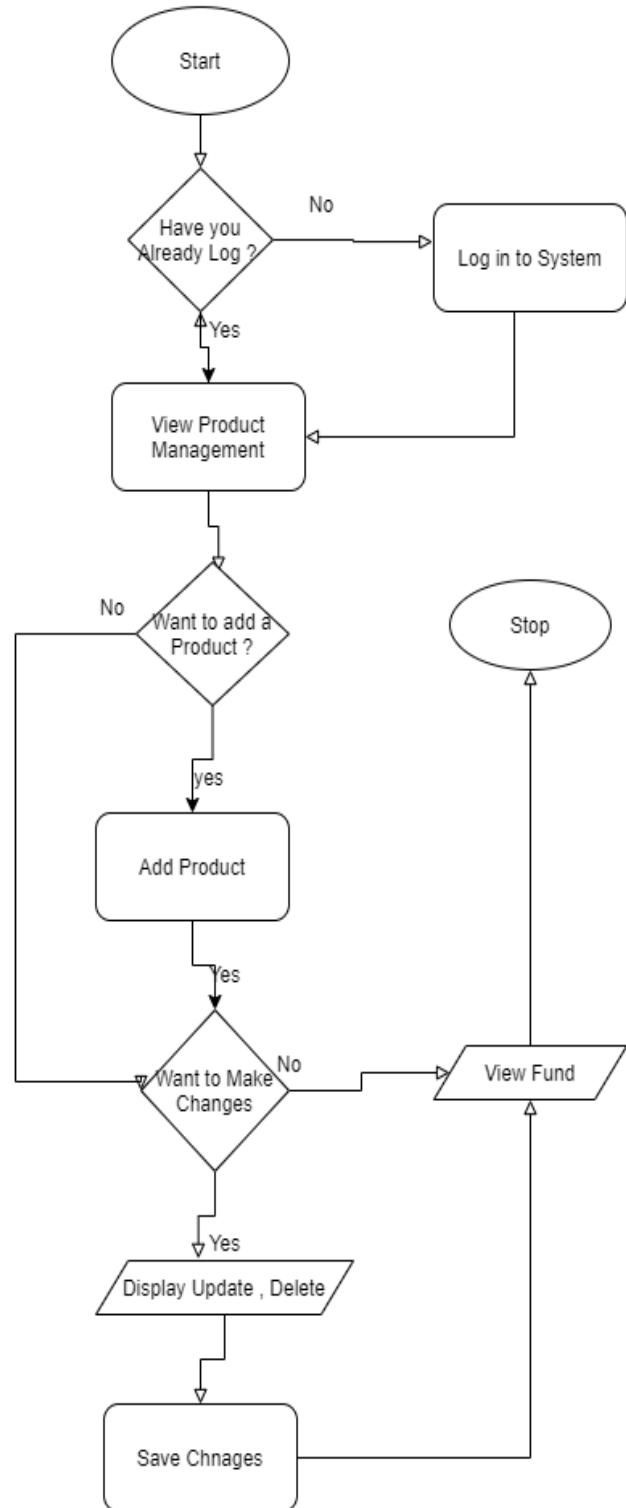
ER Diagram



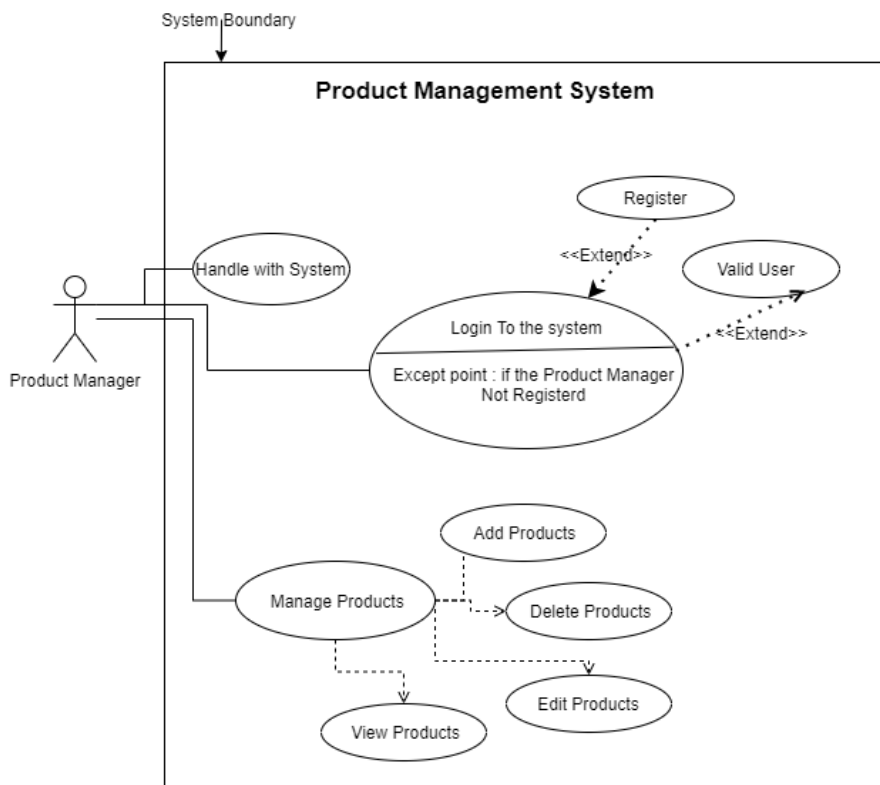
Activity Diagram



Flow Chart



Use Case Diagram



1) Service development and testing

- **Tools**

- ✓ Postman: Third party development tool
- ✓ Java-JAX-RS (Jersey): Used for backend development
- ✓ Eclipse: Use as IDE for developing this project
- ✓ Tomcat: Used as the server
- ✓ PhpMyAdmin: Used to database connection

- **Testing methodology and results**

- a) View – GET method

- Test input = URL
 - Expected output = Display the table
 - Actual output = Display the table

b) Insert – POST method

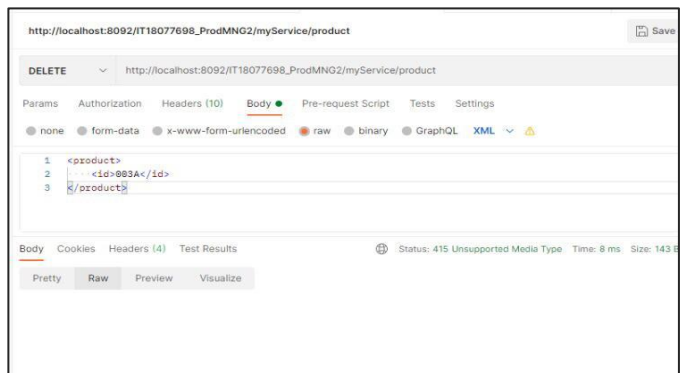
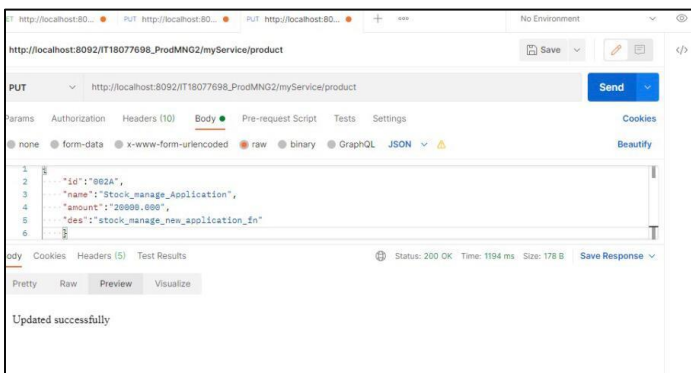
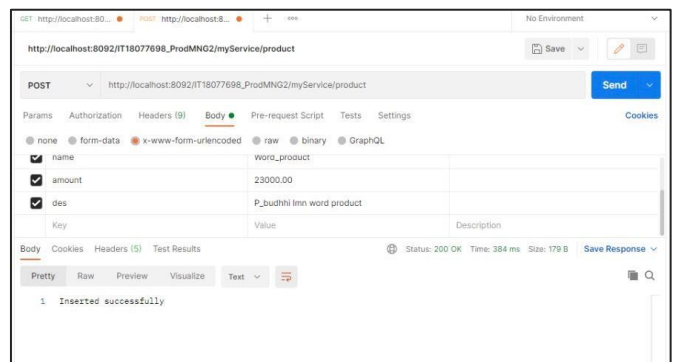
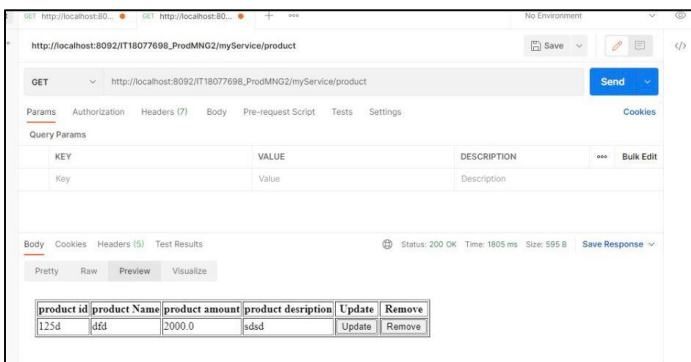
- Test input: id = "002A", name: "Word product", Amount: "23000.00", Description: "p_buddhi inn word product"
- Expected output = Inserted Successfully
- Actual output = Inserted Successfully

c) Update – PUT method

- Test input: id = "002A", name:" Stock manage application", Amount: "20000.00", Description: "Stock manage new application_fn"
- Expected output = Updated Successfully
- Actual output = Updated Successfully

d) Delete – DELETE method

- Test input: id = "003A"
- Expected output = Deleted Successfully
- Actual output = Deleted Successfully



2) Perera B.A.A.W.S – IT19015422

URL: https://github.com/Buddhi1998/PAF_Project/tree/main/IT19015422_FundMng

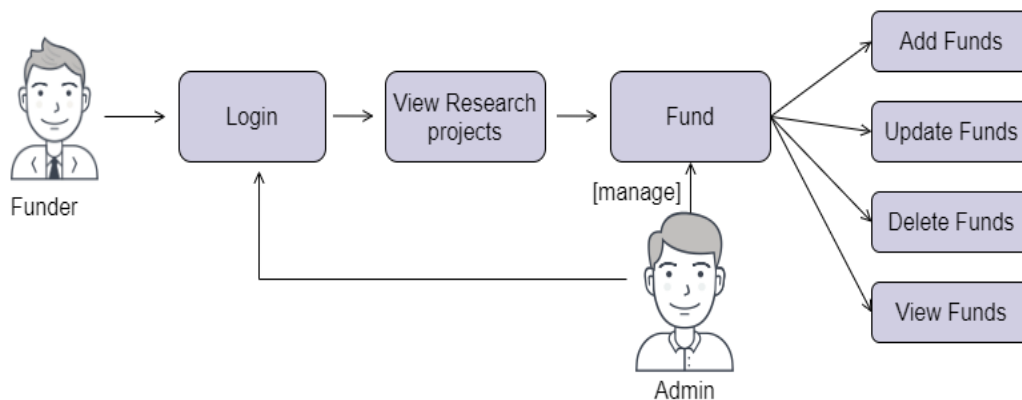
GitHub: IT19015422_FundMng

Fund Management

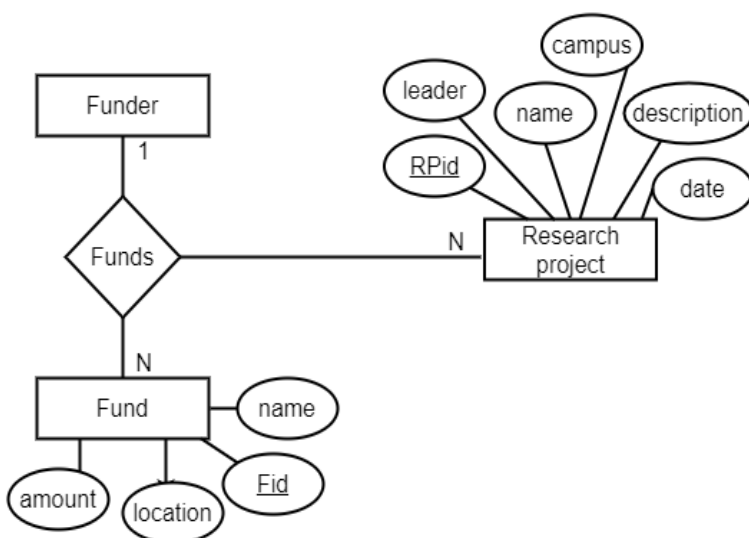
From this will manage all the funds provided by funders to develop the researcher's projects. Admin can manage all the funds from funders. And also, funders can do update and cancel their funds according to the performance of the researcher.

2. Service design

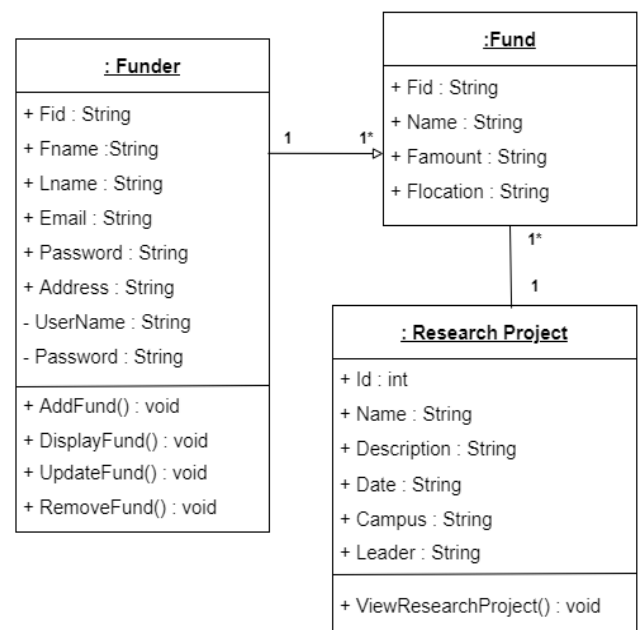
API of the service



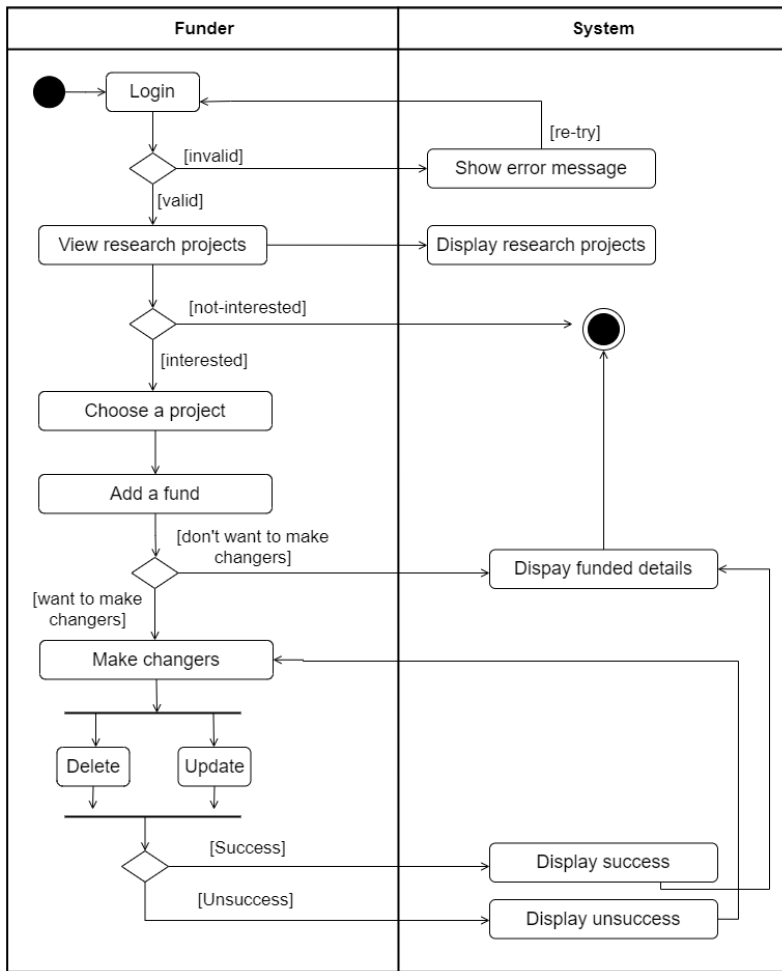
ER Diagram



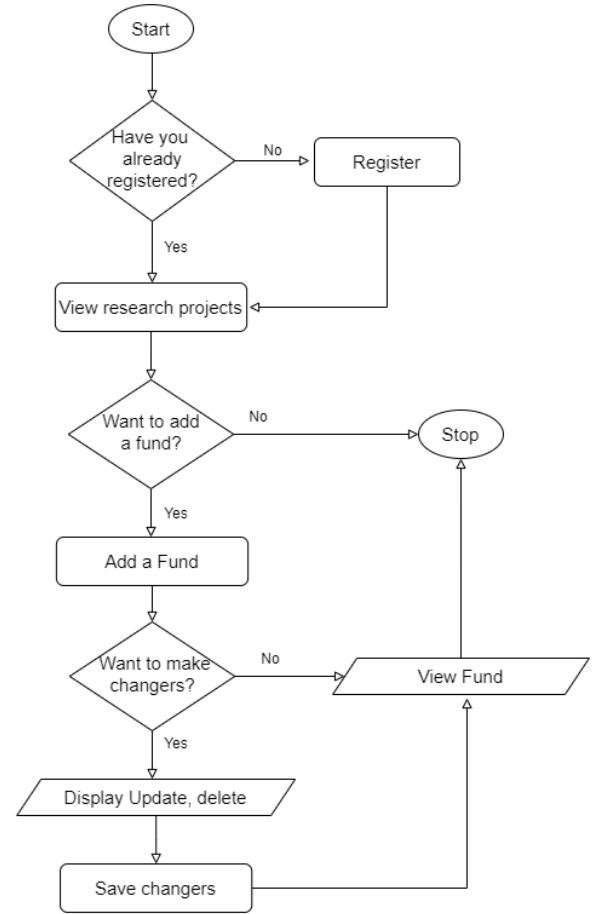
Class Diagram



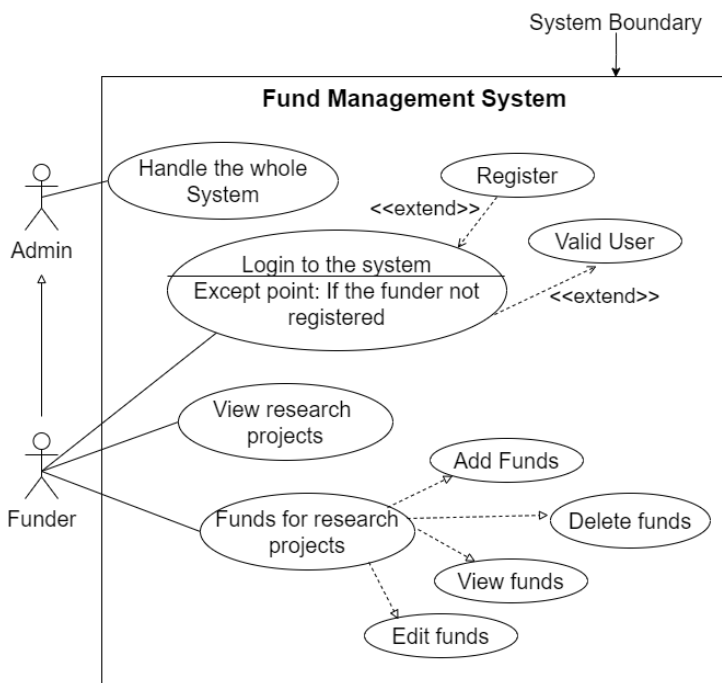
Activity Diagram



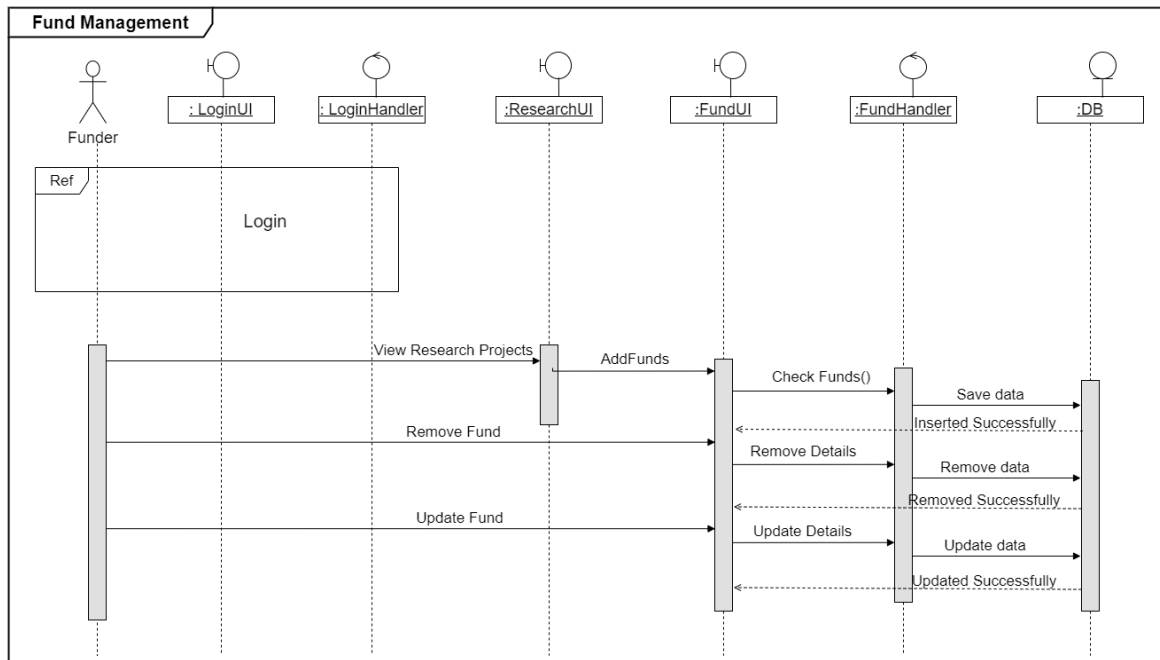
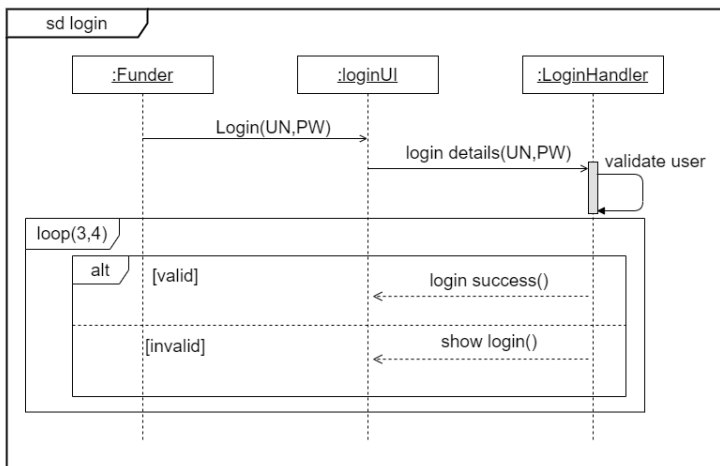
Flow Chart



Use Case Diagram



Sequence Diagram



3) Service development and testing

• Tools

- ✓ Postman: Third party development tool
- ✓ Java-JAX-RS (Jersey): Used for backend development
- ✓ Eclipse: Use as IDE for developing this project
- ✓ Tomcat: Used as the server
- ✓ PhpMyAdmin: Used to database connection

- **Testing methodology and results**

- a. **View – GET method**

- Test input = URL
 - Expected output = Display the table
 - Actual output = Display the table

- b. **Insert – POST method**

- Test input: id = "006", name:" Nimal", Amount: "30000.00", Location: "Negombo"
 - Expected output = Inserted Successfully
 - Actual output = Inserted Successfully

- c. **Update – PUT method**

- Test input: id = "006", name:" Sunil", Amount: "20000.00", Location: "JaEla"
 - Expected output = Updated Successfully
 - Actual output = Updated Successfully

- d. **Delete – DELETE method**

- Test input: id = "006"
 - Expected output = Deleted Successfully
 - Actual output = Deleted Successfully

GET http://localhost:8097/IT19015422_FundMng/myService/fund

Params Authorization Headers (6) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize

funderID	funderName	fundAmount	funderLocation	Update	Remove
001	Chathurya	50000.0	Malabe	Update	Remove
002	Hashini	20000.0	Kottawa	Update	Remove
003	Amal	35000.0	Kandy	Update	Remove
004	Rasuni	55000.0	Maharagama	Update	Remove
005	Buddhi	60000.0	Anuradhapura	Update	Remove

POST http://localhost:8097/IT19015422_FundMng/myService/fund

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	006	
<input checked="" type="checkbox"/> name	Nimal	
<input checked="" type="checkbox"/> amount	30000.00	
<input checked="" type="checkbox"/> location	Negombo	
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize Text

1 Inserted successfully

DELETE http://localhost:8097/IT19015422_FundMng/myService/fund

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL XML

```
1 <fundData>
2 <id>006</id>
3 </fundData>
```

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize Text

1 Deleted successfully

PUT http://localhost:8097/IT19015422_FundMng/myService/fund

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {"id": "006",
2  "name": "Sunil",
3  "amount": "20000.00",
4  "location": "JaEla"
5 }
6
```

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize Text

1 Updated successfully

3)Rasuni Wageesha H.A - IT19015040

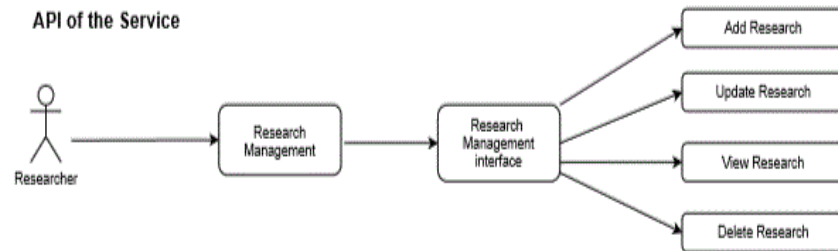
URL : http://localhost:8090/IT19015040_Research/myService/Research

GitHub: IT19015040_Research

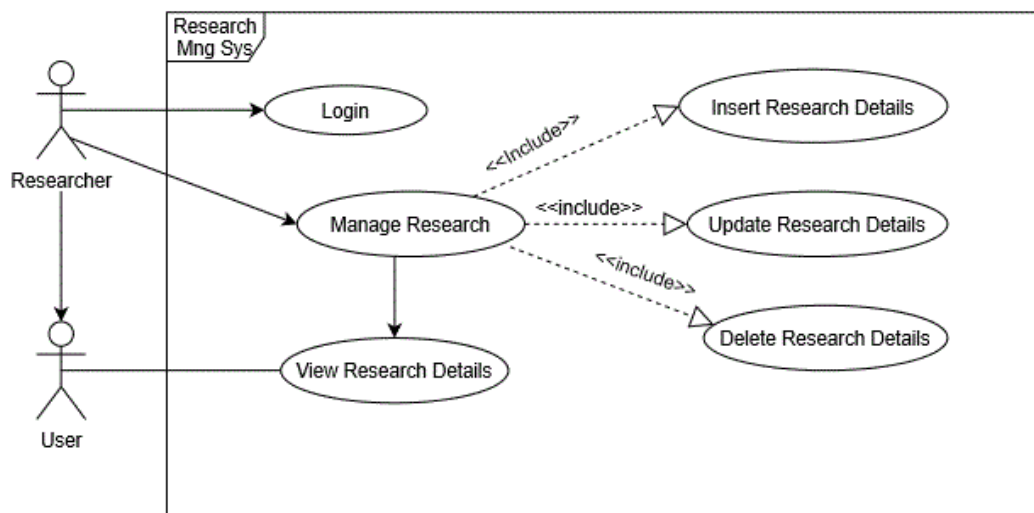
Research Management

This system has several management systems so one of the main management system is research management. This management system handle by the researcher and he is the actor of this part. Researcher can handle add, update, delete research paper details and also this adding research paper details view in the table within the system.

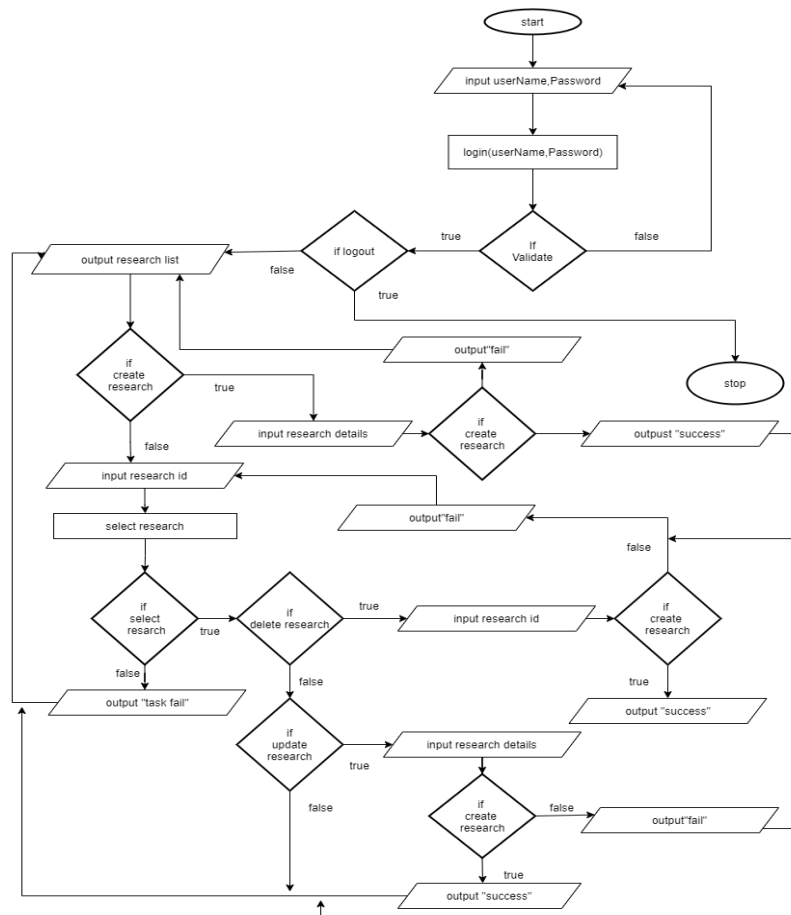
01)API of the service



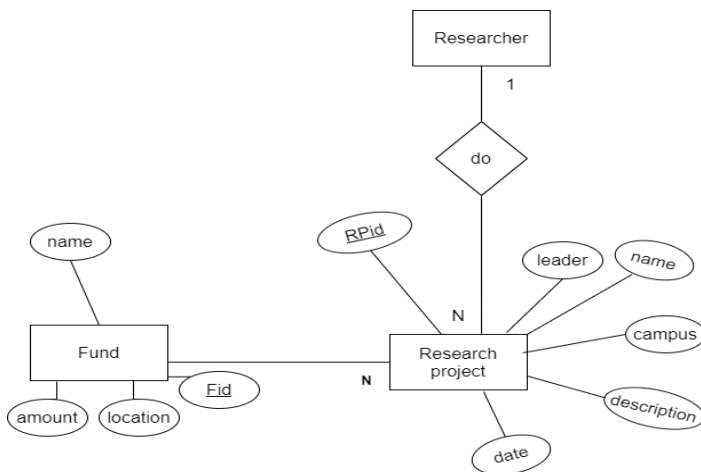
02) Use case



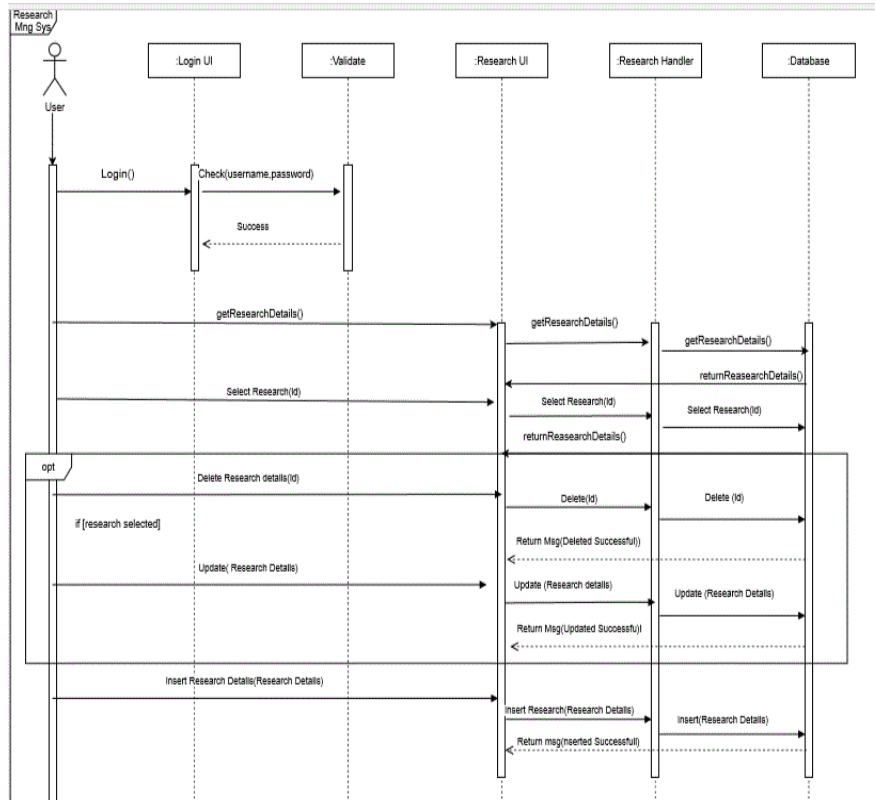
03) Flow chart



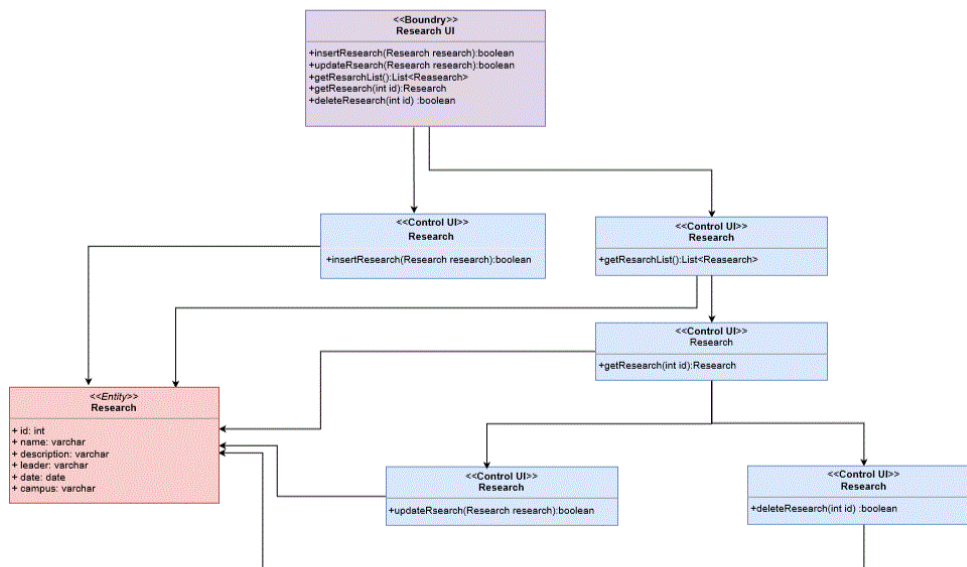
04) ER diagram



05) Sequence diagram



06) Class diagram



07) Tool used

	Tool	Reason for selection
Back end	Java-JAX-RS (Jersey development)	Used for backend
IDE	Eclipse	Use as IDE for developing this project
Server	Tomcat	Used as the server and easy configuration
database	PhpMyAdmin	Creation of database and connection is easy
Build tool	Maven	Knowledge gathered to the lab sessions
API testing	Postman	Can create better API faster

08) Service deployment and testing

	Test description	Test Input	Exception output	Actual output	Test Result
01	Insert research details	Id – 10 Name-de Description-se methodology leader- Rashini Date- 2021.04.28 Campus-SLIIT	"Inserted successfully"	"Inserted successfully"	pass
02	Delete research details	url	"deleted successfully"	"deleted successfully"	pass
03	Update research details	url	"updated successfully"	"updated successfully"	pass
04	Read research details	url	Display research details table	Display research details table	pass

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results Status: 200 OK Time: 25 ms Size: 2.51 KB Save Response

Pretty Raw Preview Visualize

Item Code	Research Title	Research Description	Member Name	Published Date	University	Update	Remove
1	economics	supply and demand	Sahan	2021-04-21	SLIT	Update	Remove
2	mobile application	how work mobile application	Ishuka	2021-04-22	NIBM	Update	Remove
11	growth	country growth	Sadun	2021-04-29	NSBM	Update	Remove
15	dsa	python	Hansi	2021-04-29	SLIT	Update	Remove
16	oop	oop cocepts	Kasuni	2021-04-26	SLIT	Update	Remove

Bootcamp Runner Trash

http://localhost:8090/IT19015040_Research/myService/Research

POST http://localhost:8090/IT19015040_Research/myService/Research Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

☒ id 20

☒ name DS

☒ description ill

☒ leader Tharushi

☒ date 2021-04-20

☒ campus SLIT

Body Cookies Headers (5) Test Results Status: 200 OK Time: 215 ms Size: 179 B Save Response

Pretty Raw Preview Visualize

Inserted successfully

http://localhost:8090/IT19015040_Research/myService/Research

PUT http://localhost:8090/IT19015040_Research/myService/Research Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "id": "16",
3   "name": "dsa",
4   "description": "data structures",
5   "leader": "Rashini",
6   "date": "2021-04-29",
7   "campus": "NIBM"
8 }
9
10

```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 132 ms Size: 179 B Save Response

Pretty Raw Preview Visualize

Updated successfully

http://localhost:8090/IT19015040_Research/myService/Research

DELETE http://localhost:8090/IT19015040_Research/myService/Research Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL XML

```

1 <itemData>
2 <itemID>16/</itemID>
3 </itemData>
4
5
6

```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 361 ms Size: 179 B Save Response

Pretty Raw Preview Visualize

Deleted successfully

4) Gamage M.S.S.K - IT19120430

URL: https://github.com/Buddhi1998/PAF_Project/tree/main/user_managment/src/main/java/com/project/user_managment.

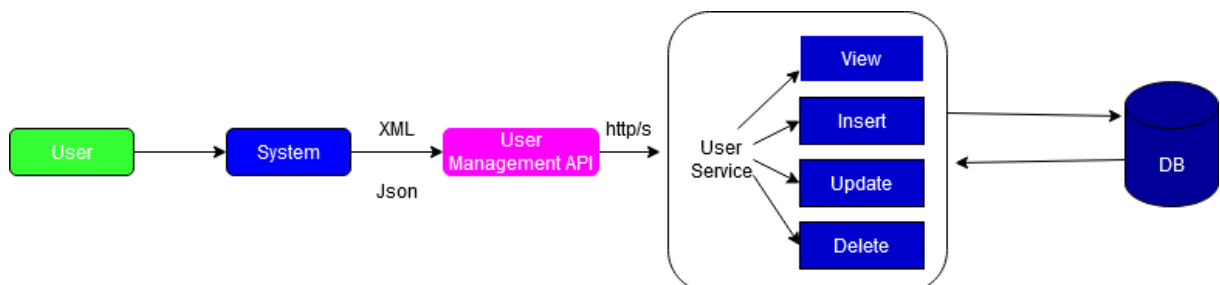
GitHub: user management.

User Management

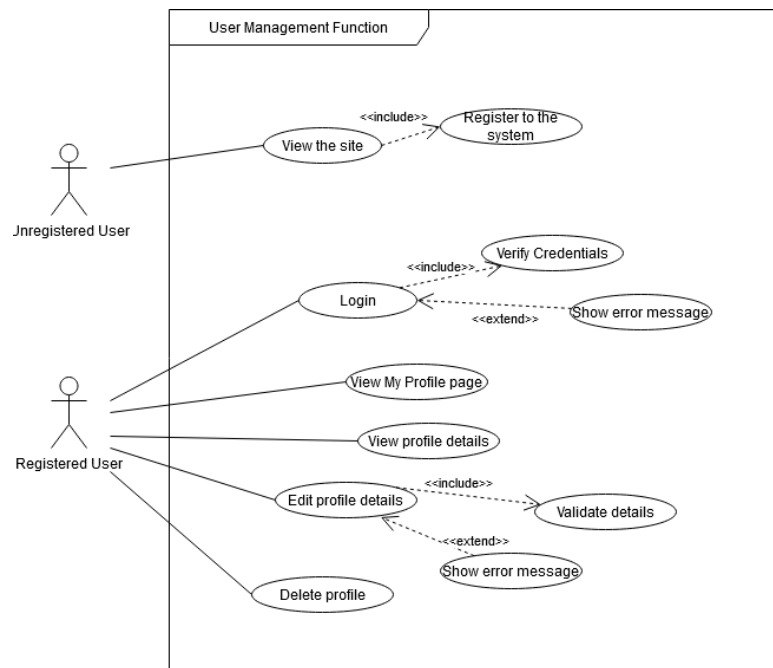
This system is the key of the implemented system in the API system. can be able to add, update, delete and view the users details. user has to login into the system with a valid account. then user can see his profile and if he wants to update profile he can easily update or he want to delete that account he can also delete account.

1. Service design

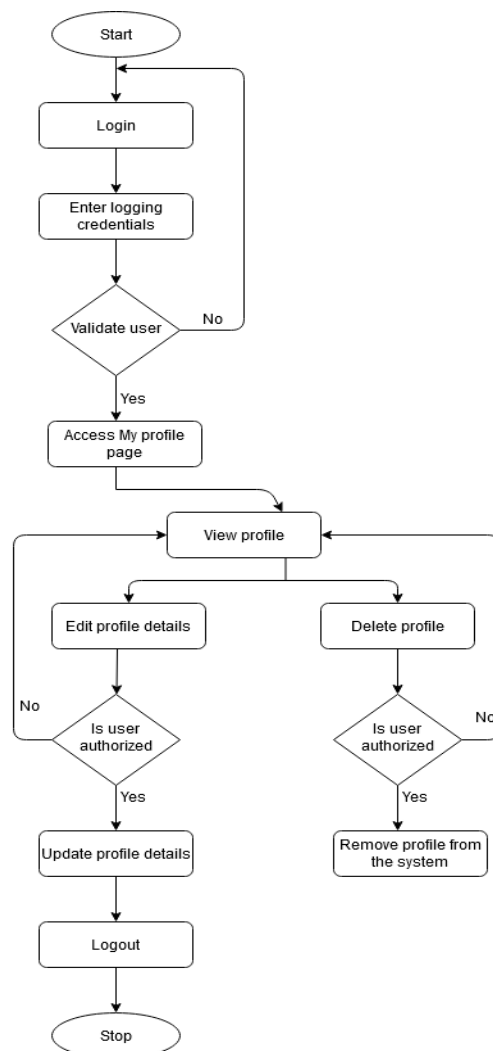
API of the service



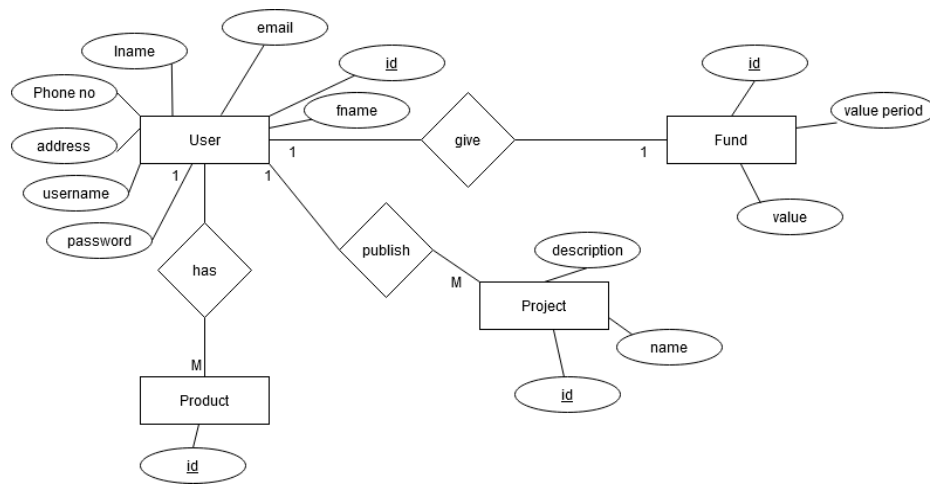
01) Use Case



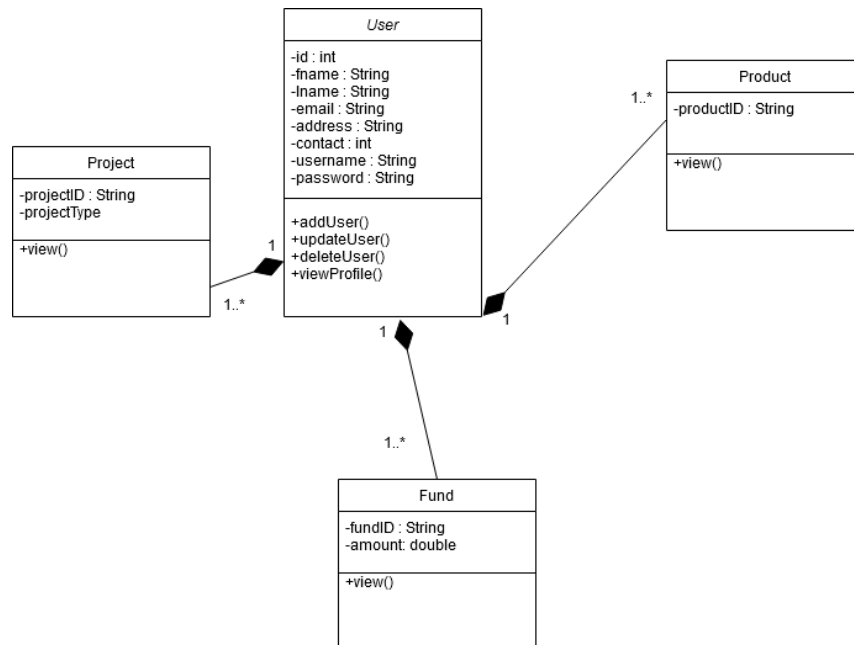
02) Flow chart



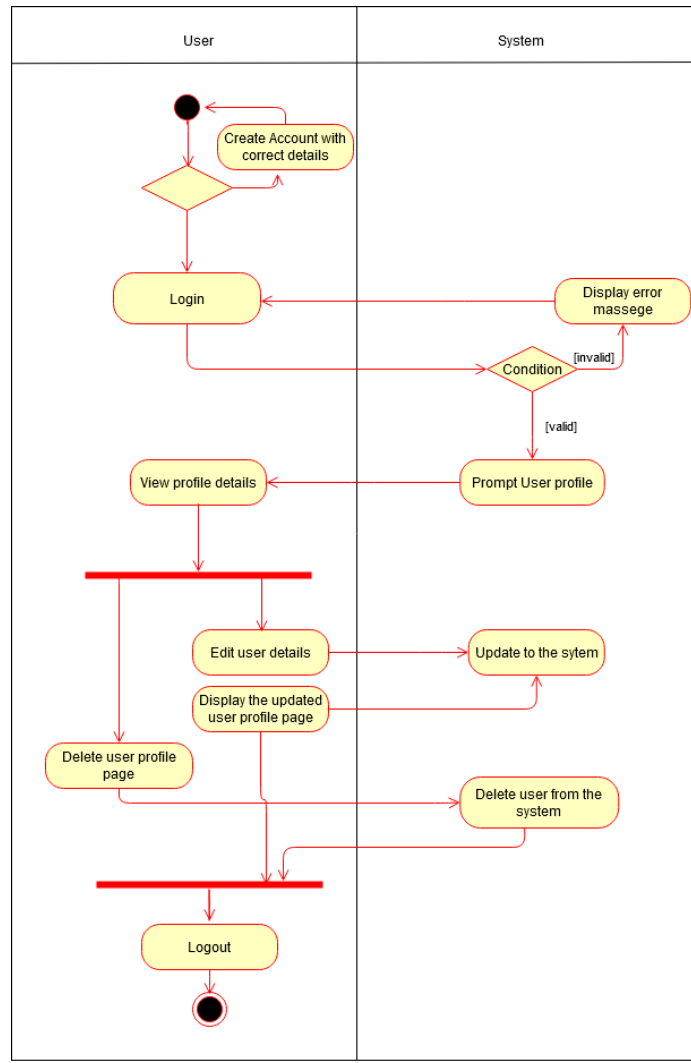
03) ER Diagram.



04) Class Diagram.



05) Activity Diagram.



03.Service development and testing

• Tools

- ✓ Postman: Third party development tool
- ✓ Java-JAX-RS (Jersey): Used for backend development
- ✓ Eclipse: Use as IDE for developing this project
- ✓ Tomcat: Used as the server
- ✓ PhpMyAdmin: Used to database connection

- **Testing methodology and results**

1-Insert

- ❖ POST method

- Test input:
 - id = "2", Fname:" gaveesha", Lname: "rajapaksha" ,Email:"rajapakshar@gmail.com", Address:"Kaduwela",PhoneNo:"0776787654",Username:"gaveesha_g",Password:"gavee123".
- Expected output = Inserted Successfully
- Actual output = Inserted Successfully

2-View

- ❖ GET method

- Test input = URL
- Expected output = Display the table
- Actual output = Display the table

3-Update

- ❖ PUT method

- Test input: id = id = "1", Fname:" sanuwana", Lname: "Gamage", Email: "sanuwana_g@gmail.com", Address:"kurunegala",PhoneNo:"077123456",Username:"sanuwana_g",Password:"sanu1234".
- Expected output = Updated Successfully
- Actual output = Updated Successfully

4-Delete

- ❖ DELETE method

- Test input: id = "1"
- Expected output = Deleted Successfully
- Actual output = Deleted Successfully

http://localhost:8090/user_management/webapi/users/user

POST http://localhost:8090/user_management/webapi/users/user

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "id": 2,
3   ... "Fname": "gaveesha",
4   ... "Lname": "rajapaksha",
5   ... "Email": "gaveesha@gmail.",
6   ... "Address": "kaduwela",
7   ... "Phone_n": 776787654,
8   ... "username": "gaveesha_g",
9   ... "password": "gavee123"
10 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "Address": "kaduwela",
3   "Email": "gaveesha@gmail.",
4   "Fname": "gaveesha",
5   "Id": 2,
6   "Lname": "rajapaksha",
7   "Phone_n": 776787654,
8   "address": "kaduwela",
9   "email": "gaveesha@gmail.",
10  "fname": "gaveesha",
11  "id": 2,
```

GET http://localhost:8090/user_management/webapi/users/read

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize

ID	First Name	Last Name	Email	Address	Contact No	Username	Password
1	Sanuwana	Gamage	sanuwana@gmail.	Kurunegala	771234567	sanuwana_g	sanu123

PUT http://localhost:8090/user_management/webapi/users/update

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "id": 1,
3   ... "Fname": "Sanuwana",
4   ... "Lname": "Gamage",
5   ... "Email": "sanuwana_g@gmail.",
6   ... "Address": "Kurunegala",
7   ... "Phone_n": 771234567,
8   ... "username": "sanuwana_g",
9   ... "password": "sanu1234"
10 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text

```
1 Updated Successful
```

http://localhost:8090/user_management/webapi/users/deleteuser/3

DELETE http://localhost:8090/user_management/webapi/users/deleteuser/3

Params Authorization Headers (6) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 Delete Successful
```