FDRS

(Free Distribution Resources System)

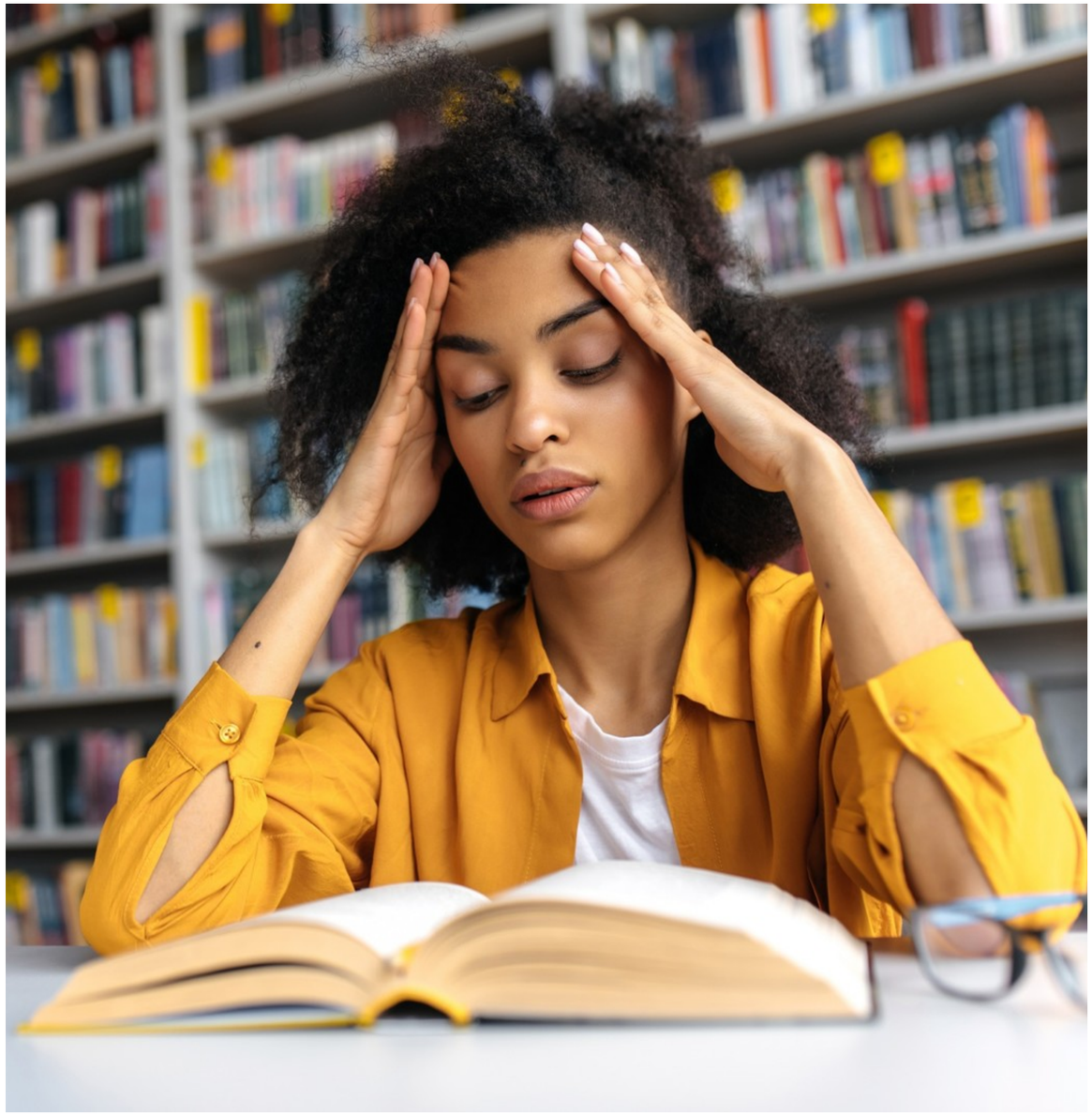
Graduation Project (1) by Wasef Joyousi Anas Alseid Saif Karborani Surpervised by Dr.Ashraf Odeh

CHAPTER ONE

INTRODUCTION

1.1PROBLEM

WHATS OUR CHALLENGE THAT WE WANT TO SOLVE ?

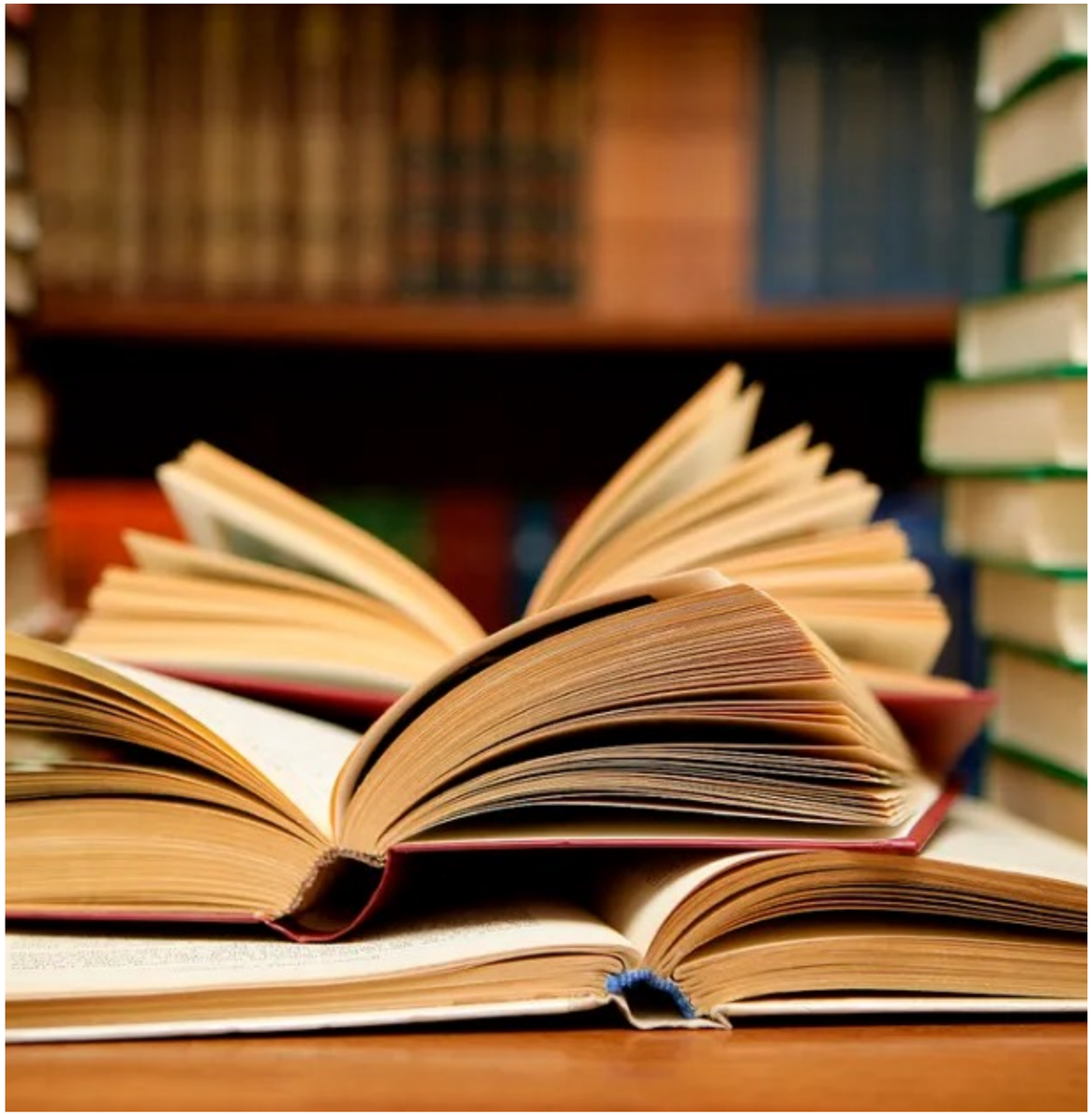


SOLUTION

WHAT DID WE DO ABOUT OUR PROBLEM?

1.3 RELATED WORK

WHAT'S NEW IN OUR SOLUTION ?



1.4 TECHNOLOGY AND TOOLS TO BE USED

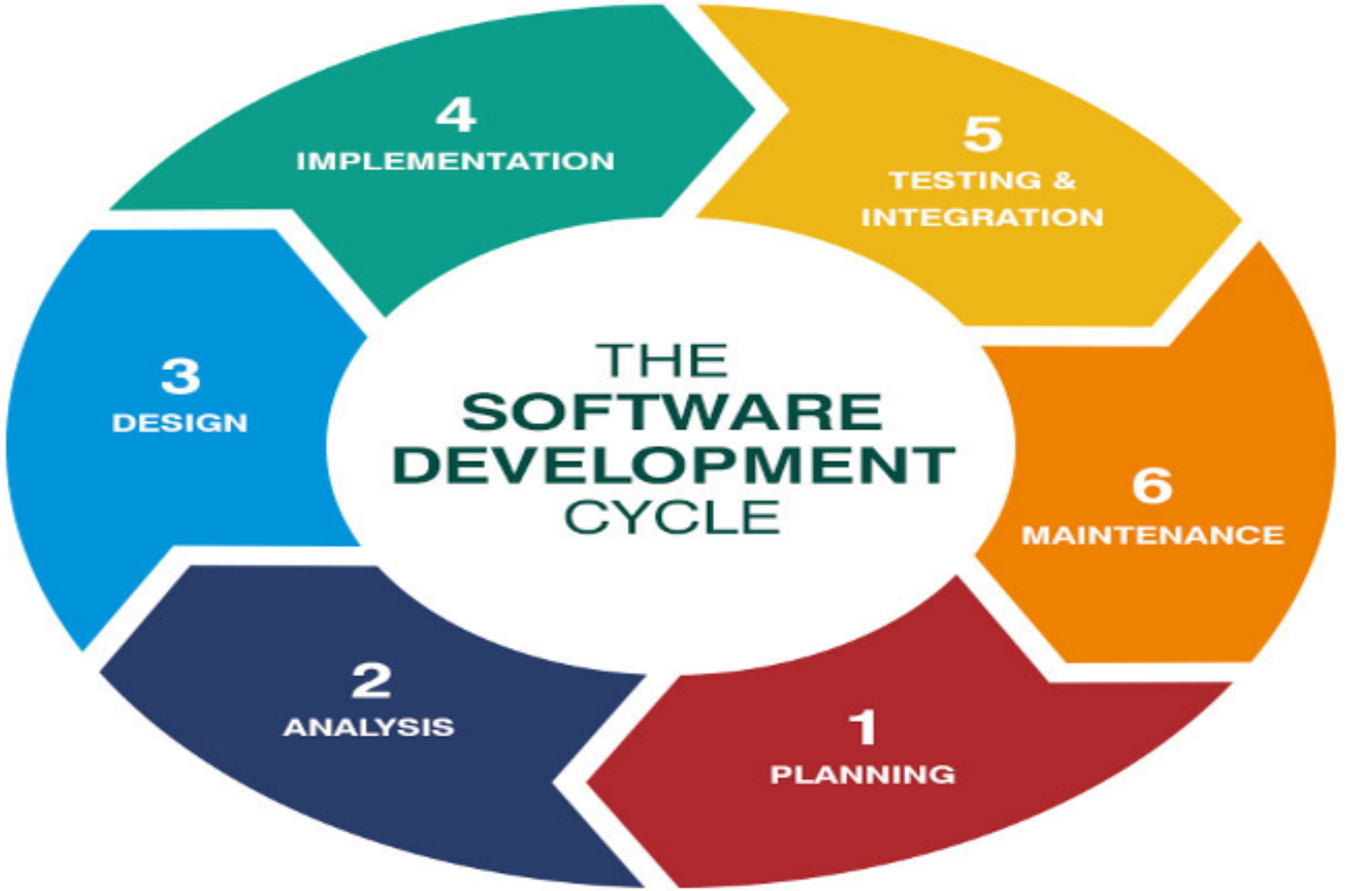
CHAPTER TWO

PROJECT PLAN

2.1 PROJECT OBJECTIVES

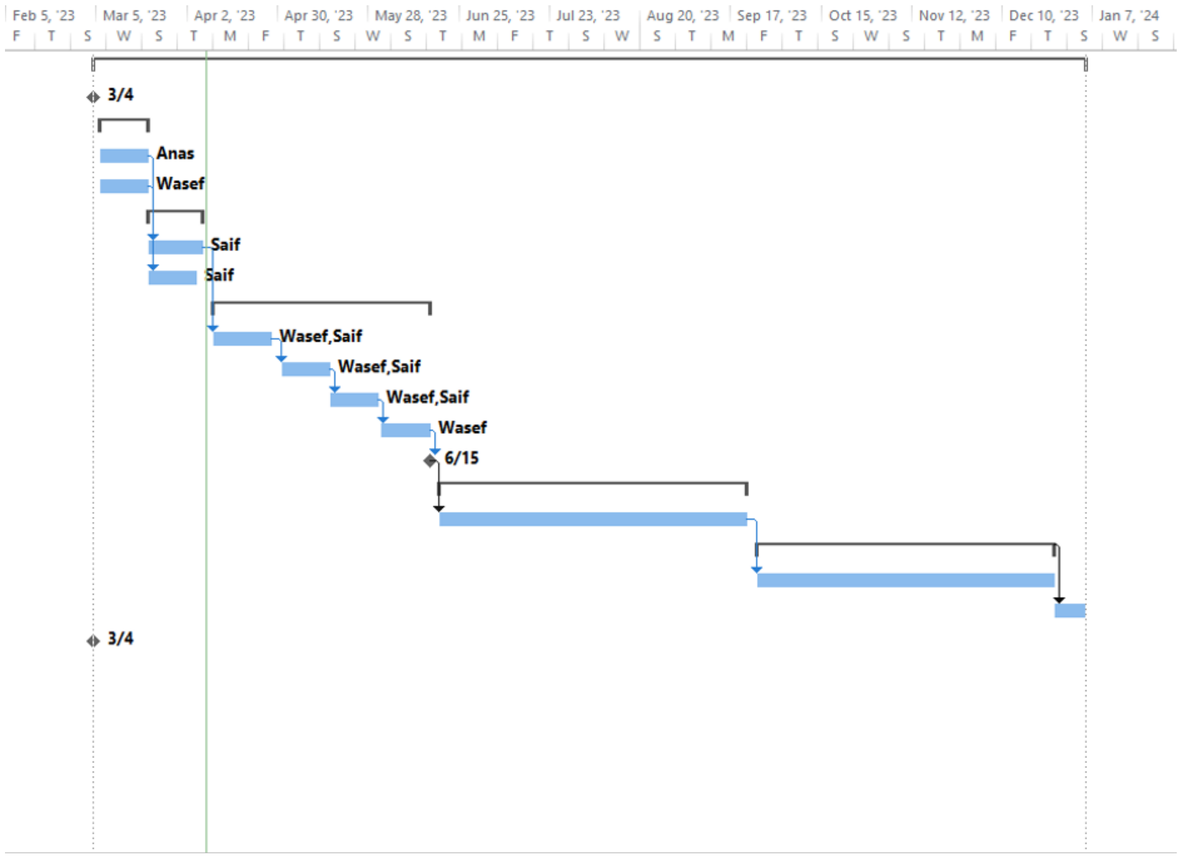
- Provide users in the university with the new up to date academic references.
- Provide users in the university to ability to share and upload their academic resources.
- Provide users to search for resources (Books...Etc.) Based on their specific needs.
- Provide users a user friendly interface that it easy to navigate.
- Provide users the ability to download the required resource if available.
- Provide users with video tutorials.
- Provide users the ability to review the Resource (book…etc.)
  - Provide a guest with less features
- An admin login page where admin can add books, videos or page sources.
- Open link for learning websites.

2.3 SOFTWARE PROCESS MODEL



2.4-2.5 PROJECT SCHEDUALE & CHART

Task Name	Duration	Start	Finish	Predecessors	Resource Names
E-library	**128 days**	**Sat 3/4/23**	**Thu 1/4/24**		
Start	0 days	Sat 3/4/23	Sat 3/4/23		
Project Planning	**7 days**	**Mon 3/6/23**	**Mon 3/20/23**		
Project statement	7 days	Mon 3/6/23	Mon 3/20/23		Anas
project Objectives	7 days	Mon 3/6/23	Mon 3/20/23		Wasef
Requirements and anaylsis	**8 days**	**Tue 3/21/23**	**Thu 4/6/23**		
Funcational requirements	8 days	Tue 3/21/23	Thu 4/6/23	4,5	Saif
non-functional requrirements	7 days	Tue 3/21/23	Tue 4/4/23	4,5	Saif
Arcititcture and design	**28 days**	**Mon 4/10/23**	**Thu 6/15/23**		
use case diagrams	7 days	Mon 4/10/23	Thu 4/27/23	7	Wasef,Saif
use case flow-of-events	7 days	Mon 5/1/23	Mon 5/15/23	10	Wasef,Saif
class diagram	7 days	Tue 5/16/23	Tue 5/30/23	11	Wasef,Saif
Entity relationship diagram	7 days	Thu 6/1/23	Thu 6/15/23	12	Wasef
End of Project 1.	0 days	Thu 6/15/23	Thu 6/15/23	13	
Implementation	**40 days**	**Mon 6/19/23**	**Thu 9/21/23**		
Coding	40 days	Mon 6/19/23	Thu 9/21/23	14	
Testing & integration	**40 days**	**Mon 9/25/23**	**Mon 12/25/23**		
Testing the code	40 days	Mon 9/25/23	Mon 12/25/23	16	
Deployement and maintenance	5 days	Tue 12/26/23	Thu 1/4/24	17	
End of project 2	0 days	Sat 3/4/23	Sat 3/4/23		



CHAPTER THREE

REQUIRMENTS AND ANALYSIS

3.1 FUNCTIONAL REQUIRMENTS

1. User Registration and login - Users should be able to create an account, login, and access only the features that they are authorized to use.
2. Recourse Catalog - The system should provide a searchable catalog of all recourse available in the FDRS along with their metadata such as title, author, genre, publication year, publisher, and availability status.
3. Recourse Reading - Users should be able to read recourse online or download them for offline reading.
4. Recourse Search - Users should be able to search for recourse by title, author, keyword, or category.
5. User Interaction - Users should be able to rate, review, and comment on recourse, as well as share them on social media.
6. Admin Panel - The system should have an admin panel to manage users, recourse, and system settings.
7. Reporting and Analytics - The system should provide reports and analytics to track the usage of recourse, users, and the system overall.
8. User Profile Management - Users should be able to manage their profile information, update their preferences, and view their reading history.

3.2 NON-FUNCTIONAL REQUIRMENTS

1. Performance - The system should handle many concurrent users and transactions without slowing down or crashing, with fast page load times and smooth scrolling for online reading.
2. Availability - The system should be available 24/7, with minimal downtime for maintenance or upgrades, and with high availability to ensure users can access the system whenever they need it.
3. Reliability - The system should be reliable and provide accurate information, without losing data or causing errors, with frequent backups to ensure that data is not lost.
4. Scalability - The system should be able to scale up or down as needed to accommodate changes in user traffic or library collections, with the ability to add more servers and storage space as required.
5. Security - The system should be secure, with measures in place to protect user data, prevent unauthorized access, and detect and respond to security breaches, with SSL encryption and strong password policies.
6. Usability - The system should be easy to use, with an intuitive user interface and clear instructions for performing tasks, with clear and concise user documentation available.
7. Compatibility - The system should be compatible with different web browsers and operating systems, and support different languages and character sets, with a responsive design to ensure that it works on different devices.
8. Maintainability - The system should be easy to maintain and upgrade, with clear documentation and well-structured code, with change management policies in place to prevent unplanned changes.

CHAPTER FOUR

ARCHITTECURE & DESGIN

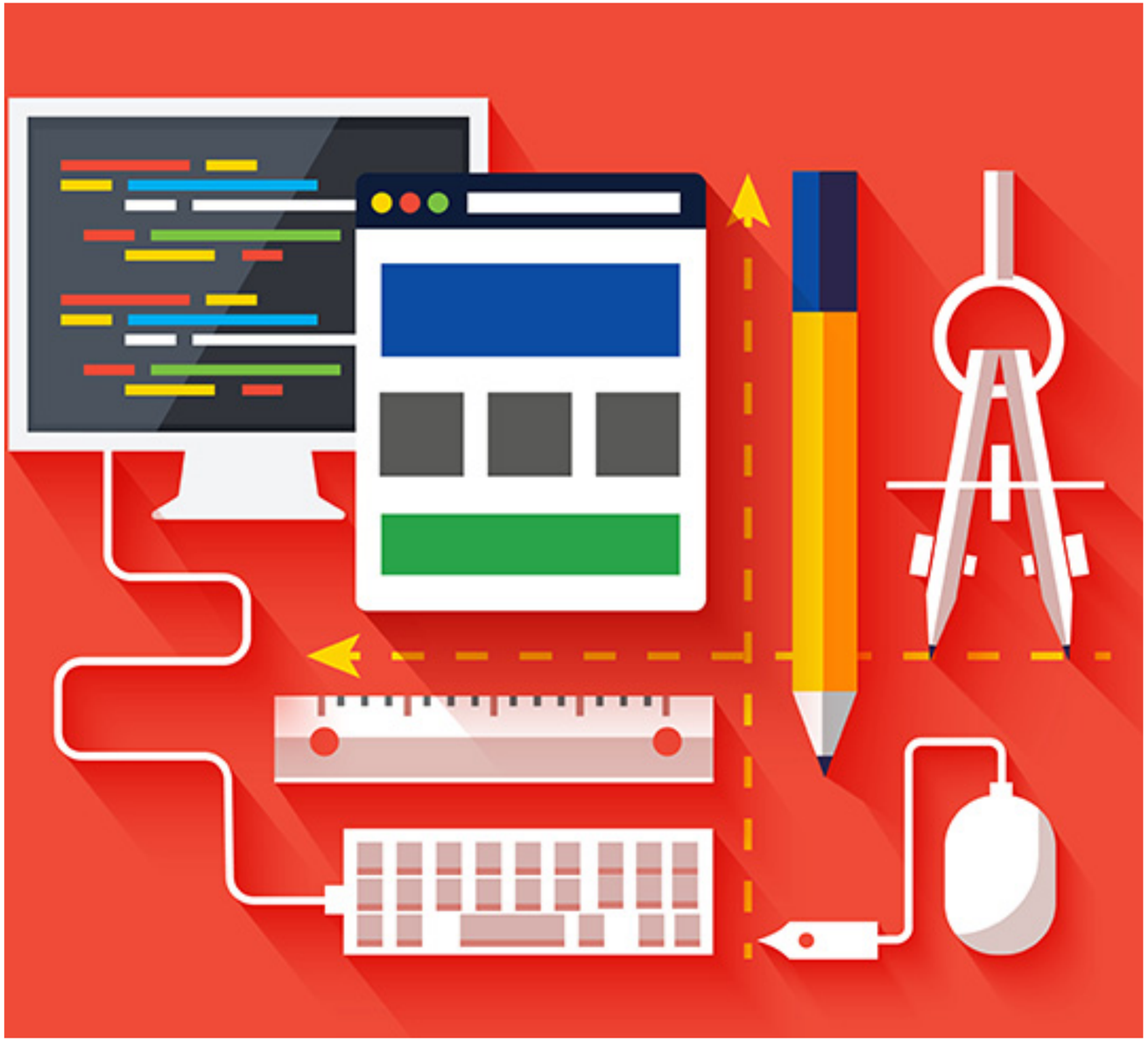
4.1 ARCHITECTURE

WE USED 3 LAYERS FOR OUR WEBSITE

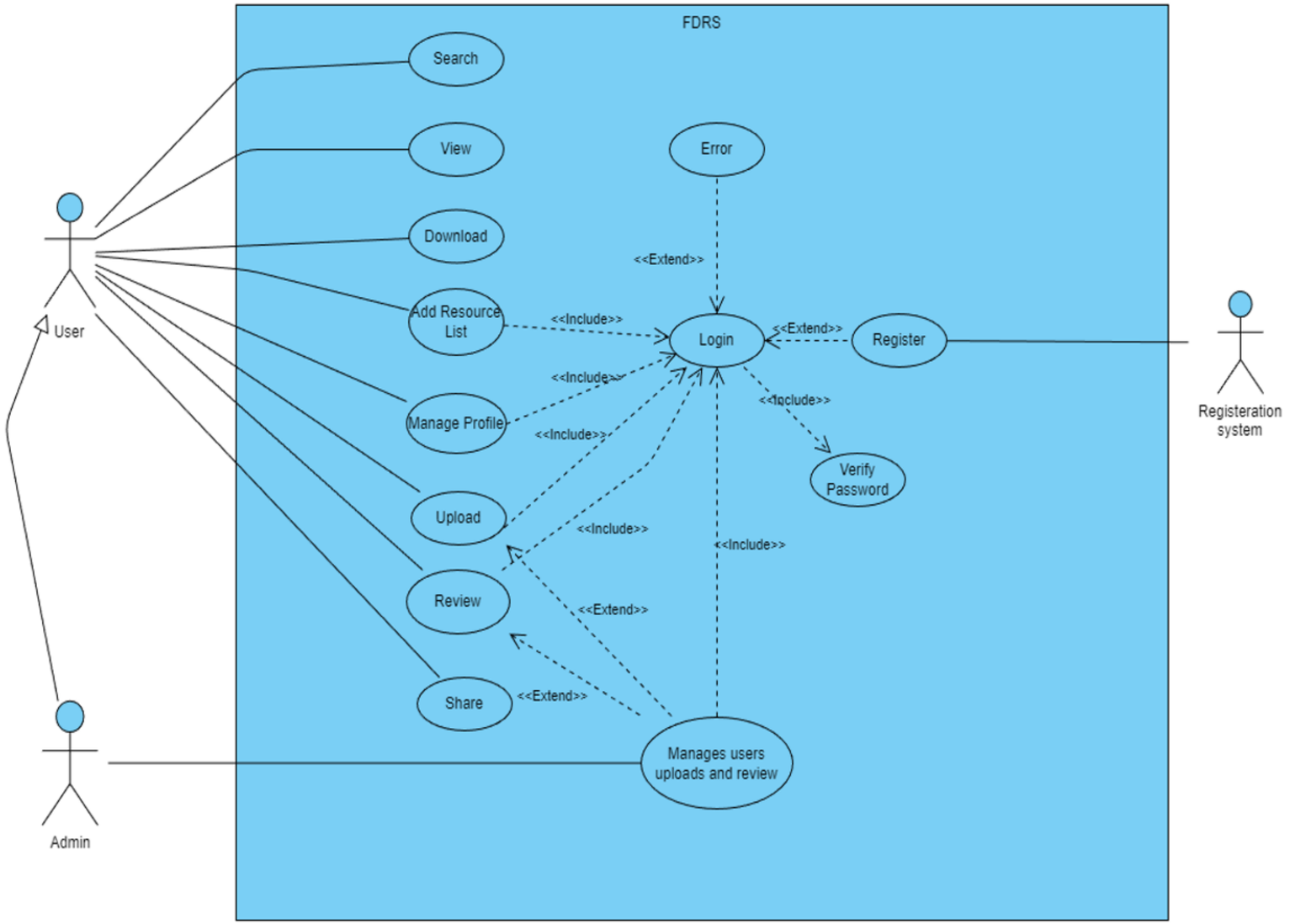
- **PRESENTATION LAYER -> FRONT END**
- **APPLICATION LAYER -> IS OUR LOGIC SERVER**
- **DATA LAYER -> DATABASE**

- **WHY DID WE USE A 3-TIER ARCHITECTURE ?**

1. **SECURITY**
2. **SCALABILITY**
3. **MODIFICATION**



4.2 USE CASE DIAGRAM



USE CASES DESCRIPTION/FLOW OF EVENTS

Use Case name:	Login	**Use Case name:**	Register(Create new account)	**Use Case name:**	Search
Use Case ID:	1	**Use Case ID:**	2	**Use Case ID:**	3
Description:	This use case describes how the actor logs into the System.	**Description:**	This use case describes how an actor creates a new user account.	**Description:**	This use case describes how the actor Search through resources.
Primary Actor:	User/admin.	**Primary Actor:**	User.	**Primary Actor:**	User/admin.
Secondary Actor:	Registration system.	**Secondary Actor:**	Registration system.	**Secondary Actor:**	...
Pre-condition:	The user must register (use case must be performed first).	**Pre-condition:**	None.	**Pre-condition:**	None.
Main flow of events:	User/admin: 1- The system requests that the actor enter his/her E-mail and password. 2- The actor enters his/her E-mail and password and presses 'login' option. 3- The system validates the entered E-mail and password and logs the actor into the system (Checks against the database).	**Main flow of events:**	1- The actor clicks on (Register) option from the home page. 2- The system prompts the actor to enter his/her full name, E-mail and password. 3- The system validates the entered E-mail and password via the registration system and then redirects the actor to the login page (A1).	**Main flow of events:**	User/admin: 1. The user uses the search bar to type the resources that he is looking for. 2. The system will scan the database based on the search type. 3. Then the system returns the desired Search resource.
Alternative Flow of events:	• The system displays an error message that the entered E-mail and/or password is invalid. • The system requests that the actor re-enters his/her E-mail and password.	**Alternative Flow of events:**	A1: • The system displays an error message that the E-mail and/or password is invalid. • The system requests that the actor re-enters his/her E-mail and password.	**Alternative Flow of events:**	• The system tells the user that the search is not Found/invalid
Post-condition:	The user that logins in will be able to do his features (Search, view, download, review, upload, track List, Share). The admin That login in will have the Extra feature of managing the users.	**Post-condition:**	Added the user to the database via the registration system actor.	**Post-condition:**	The user should then be able to see the resource.

USE CASES DESCRIPTION/FLOW OF EVENTS

Use Case name:	View
Use Case ID:	4
Description:	This use case describes how the actor view's the resource.
Primary Actor:	User/admin.
Secondary Actor:	...
Pre-condition:	None.
Main flow of events:	1. The user clicks on the view button. 2. The system redirects the user into another page.
Alternative Flow of events:	· The system return an error.
Post-condition:	When the resources accepted from the admin the user can view it

Use Case name:	Download
Use Case ID:	5
Description:	This use case describes how the actor can Download a resource from the website
Primary Actor:	User
Secondary Actor:	
Pre-condition:	None
Main flow of events:	1- The user selects a resource to download. 2- The system transfers the resource from the database (server) to the user website downloads.
Alternative Flow of events:	· The systems return to the user that the resource isn't available to download.
Post-condition:	The user has the resource on his device, drive.

Use Case name:	Add resource list
Use Case ID:	6
Description:	This use case describes how the user can add his Favorite list of resources
Primary Actor:	User/admin
Secondary Actor:	
Pre-condition:	The user should have an account and logged in
Main flow of events:	1- The user selects the resource. 2- The user clicks on add to list. 3- The system returns that it has been added to the list.
Alternative Flow of events:	· The system returns to the user that he can't add more to the list. · The system displays an error message.
Post-condition:	The user can view his resource list.

Use Case name:	upload
Use Case ID:	7
Description:	This use case describes how the user can upload resources on the website
Primary Actor:	Users.
Secondary Actor:	Admin.
Pre-condition:	Logged in.
Main flow of events:	1- The user clicks on the upload button and selects the resource he wants to upload. 2- The website displays a form with fields for the user to upload what kind of resource. 3- The admin reviews and accepts it (A1) 4- The system goes to the database server and add the resource. 5- The system returns to the user that upload is successful and has been added to the website.
Alternative Flow of events:	· The system returns that the upload is invalid. · The admin Declines the upload (A1).
Post-condition:	Users can see the uploaded resource.

Use Case name:	Review
Use Case ID:	8
Description:	The use case describes how the user can review on the resource, in which he can put a comment or rate.
Primary Actor:	User.
Secondary Actor:	Admin.
Pre-condition:	Logged in.
Main flow of events:	1- User selects the resource to review. 2- The website displays a review form with fields for the user to enter their review information (comment, rating) and submits. 3- The website displays the review on the resource for other users to see.
Alternative Flow of events:	· The review may be reviewed by the admin and deletes it.
Post-condition:	Users can see the review on the resource.

Use Case name:	Share
Use Case ID:	9
Description:	This use case describes how the user can share the resource
Primary Actor:	User/admin
Secondary Actor:	
Pre-condition:	None.
Main flow of events:	1- The user selects the resource that he wants to share. 2- The system returns a link of the resource to the user. 3- The user either copy the link or confirm sharing it via system.
Alternative Flow of events:	· The system fails providing the link or sharing the resource.
Post-condition:	The system delivers the shared resource or user has the shared link.

Use Case name:	Manage users
Use Case ID:	10
Description:	This use case describes how the admin validates the users uploads and reviews.
Primary Actor:	Admin.
Secondary Actor:	User.
Pre-condition:	Logged in as an admin.
Main flow of events:	1- The system shows to the admin the upload's of the users. 2- Admin accepts them. 3- The system returns to the user that their upload has been approved. 1- As for the review (comments of users) 2- The admin clicks on the users comment 3- Hit delete button 4- System returns that comment has been deleted.
Alternative Flow of events:	· The admin declines. · The system returns to the users that the upload has been rejected.
Post-condition:	The admin can Reject users uploads and reviews.

Use Case name:	Manage profile
Use Case ID:	11
Description:	This use case describes how the user can edit his username or password.
Primary Actor:	User.
Secondary Actor:	
Pre-condition:	Logged in
Main flow of events:	1- The user clicks on user icon or manages profile button 2- The system displays the user information. 3- The user edits their profile information and saves. 4- The system displays to the user that the Profile changes has been done.
Alternative Flow of events:	· The system returns that the edit is invalid.
Post-condition:	User is able to change his profile settings.