**Report**

**Graduation Projects Management System**

**What is the graduate project?**

* It is a graduation project that is carried out by undergraduate students in their last year with the aim of applying what has been studied during the bachelor's years.

**Who should do the graduate project?**

* Students who are about to graduate to prove their academic achievement.

**How does the process of managing the graduation projects go in the Faculty of CS and IT?**

* Groups are formed and these groups are sent to the project committee officer in the department, then the project committee chairperson presents these names to his doctors and the groups are assigned to the doctors.

**Project Idea:**

* The idea of the project is, to develop a system for managing the gradation projects, in the Faculty of Computer Science and Information Technology at AL Baha University.
* The system aims to ease the process of assigning the graduation projects to the professors.
* Also, the project will ease the process of forming student groups based on certain criteria, E.g., the area of interest of either the professor or the students.
* The system will be used as an electronic repository. This repository will include all the projects and the ideas that have been introduced by the students of the faculty in the previous years.

**Problem area:**

All the graduation project related paper works and tasks in the departments of the Faculty of Computer Science and Information Technology are done manually by the graduation projects chair committee, supervisors, examiner professors, and students. Some Examples for these papers and tasks are the following:

* Assigning supervisor professors to student groups.
* Grouping students to create student teams.
* Showing the project ideas proposed by the professors to all students.
* Submitting the evaluation forms by the examiner professors.

**Objectives:**

* Facilitate the process of forming the graduation projects groups between professors and students
* Facilitate the process of managing the graduation projects for the graduation project committee chair
* create an electronic archive that includes all graduation projects.
* Enable the examiner professors to evaluate the groups using electronic evaluations forms, and hence ease the process of the students’ evaluation.

**Related work:**

**After surveying and research, we found two systems that simulate or simulate the idea of our system.**

**The first website**: The digital repository at Taibah University

* They show the project ideas that have been done by Taibah University's students

**Means:**

We have here a website for Taibah University that displays project ideas that students will develop, and archives graduation ideas on Taibah University website to serve students. **As shown in Figure 1**

صورة تحتوي على نص

تم إنشاء الوصف تلقائياً

**[1]**

[**repository.sustech.edu**](http://repository.sustech.edu/)

**Second website**: Management system for graduation projects in the College of Computer and Information Technology.

* Browse the projects by rank, topic, area, or

the awards that got.

* A profile of all the professors who are willing to supervise students is displayed

**Means:**

* Browse projects according to the order by topic or awards you have
* A profile of all professors wishing to supervise students is displayed. **As shown in Figure 2**

****

**[2**]

[www.fcit-ye.com/gp/index.php](http://www.fcit-ye.com/gp/index.php)

**what is the difference between our work**

**and the related works?**

* Archive of group assessments and easy reference.
* Create student groups by doctors.
* Display a list of students who do not have groups for both committee chairs and doctors.
* Allow students to register in groups.
* Requirement's specifications (software & hardware)

We used techniques or requirements in creating our system:

**Software:**

* **Backend:**

MySQL:

"Is a Database Service is a fully managed database service to deploy cloud-native applications." [[8]](https://www.mysql.com/)

Django:

 "is a high-level Python Web framework that encourages rapid development and clean, pragmatic design".[[9](https://www.djangoproject.com/)]

* **Frontend**

HTML:

"Hypertext Markup Language (**HTML**) is the standard markup language for documents designed to be displayed in a web browser."[10]

CSS: "is the acronym of “Cascading Style Sheets"[11].

Bootstrap:" Quickly design and customize responsive mobile-first sites with **Bootstrap"** [12]

* **version control code‏:**

Git: "is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency"[13]

GitHub: is where over 56 million developers shape the future of software, together. Contribute to the open-source community."[14]

**hardware:**

PC

**System analysis**

When analyzing the system, we used three models of diagrams:

**Use Case Diagram**

**State Transition Diagram**

**Entity-Relationship Diagram**

**Use case of Graduation projects management system**

"A **use case** is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each **use case** is represented as a sequence of simple steps, beginning with a user's goal, and ending when that goal is fulfilled."[1]

* **Committee chairs from all departments** have four special characteristics in them Create new group, Add new idea, Assign Professor to groups, Assign examiner professors to groups
* **Doctors** have two properties: create a group, show groups, and they also have one property, and my groups also contain two properties, rate my group, and show an examiner's evaluation.
* **Students** have three properties: Create a group, Show Proposed Ideas, Archive the project

**As shown in Figure 3**

Diagram

Description automatically generated

**[3]**

**State Transition Diagram**

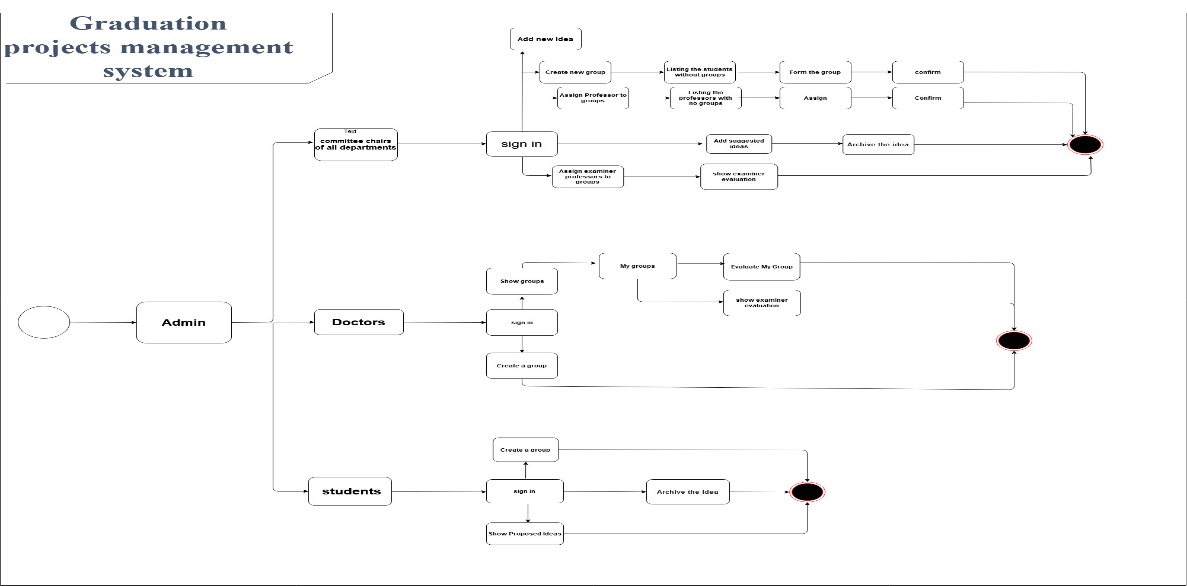
It is a diagram describing the method of transition from one state to another, the software system the project consists of three users.

the first user is, **committee chairs of all departments,** He enters the personal account through the login page after that, five stages appear for him. The first stage is the addition of an idea, the second stage is the creation of a group, the third stage is the addition of doctors to the groups, the fourth stage is the addition of the ideas suggested from section, the fifth stage is adding examiners to the group.

the second user is, **Doctors**, He logs into the personal account through the login page after that, two of the stages appear. The first case is the show groups the second case is the establishment of groups.

the third user is **Students**, He logs into the personal account through the login page after that, and three cases appear. The first case is create a group, the second case is archive the idea, and the third case is show proposed ideas.

**Figure 4 shows the State Transition Diagram**

****

**Figure 4**

**Entity-Relationship Diagram:**

The **ERD** is used to explain the Entity and its relationship with other Entities**.**

**As follows:**

Entity committees' chairs

It has more than one relationship with an entity such as:

1. Entity Groups
2. Entity Evaluation
3. Entity Projects
4. Entity Examiners
5. Entity department

**And the other entity is:**

Entity Doctors

It has more than one relationship with an entity such as:

1. Entity Groups
2. Entity Evaluation
3. Entity Projects

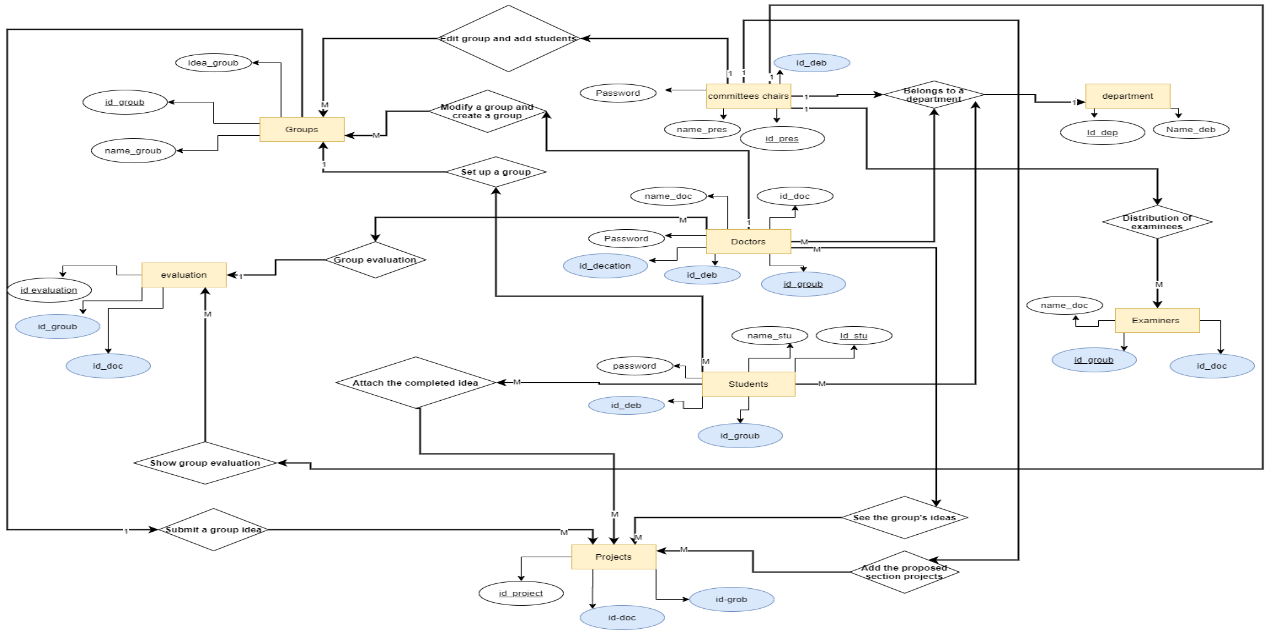
**The last entity is:**

Entity Students

It has more than one relationship with an entity such as:

1. Entity Groups
2. Entity Projects
3. Entity department

**Figure 5 shows the Entity-Relationship Diagram**

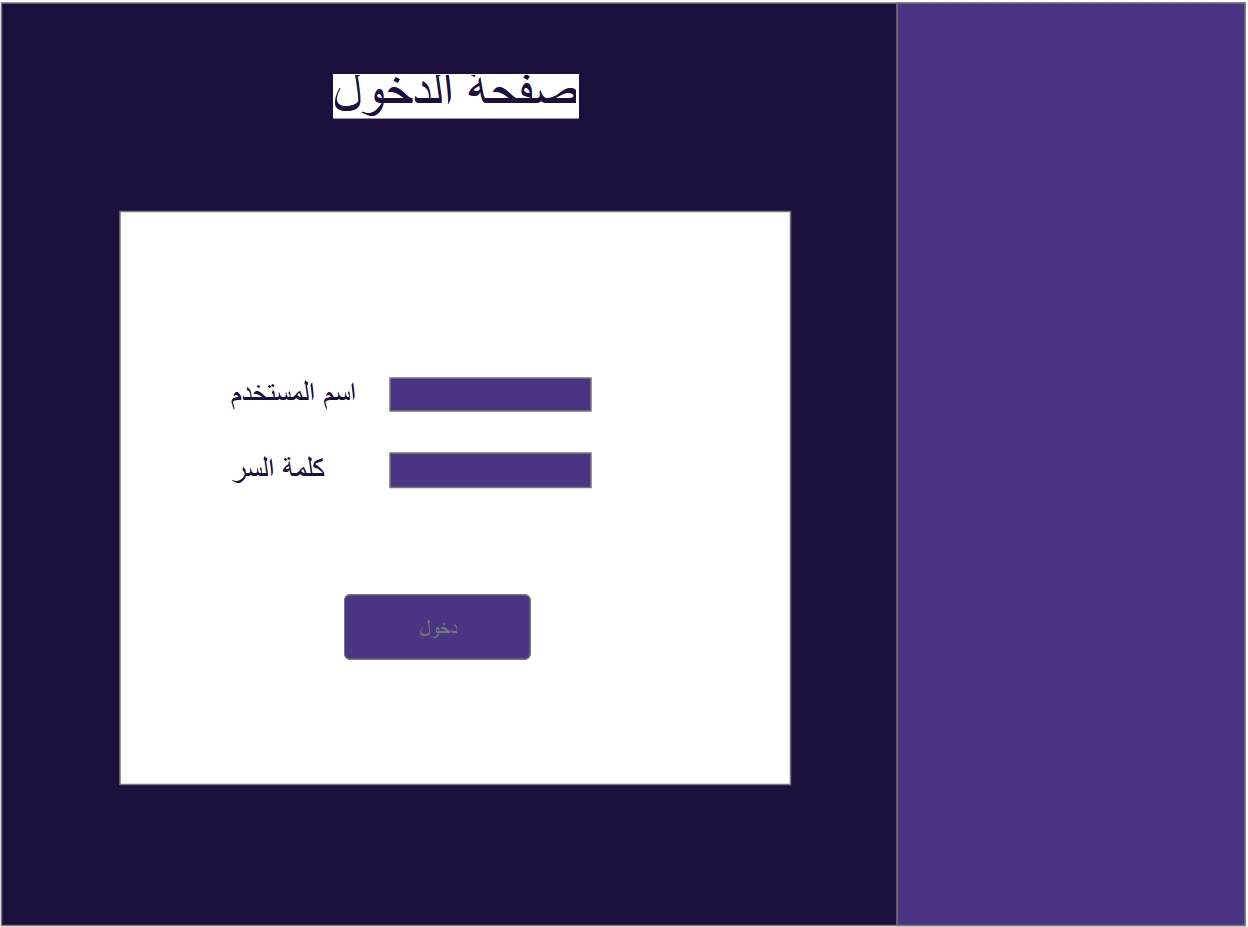
****

**[5]**

**The main pages**

A general and initial visualization of the system's user pages was made:

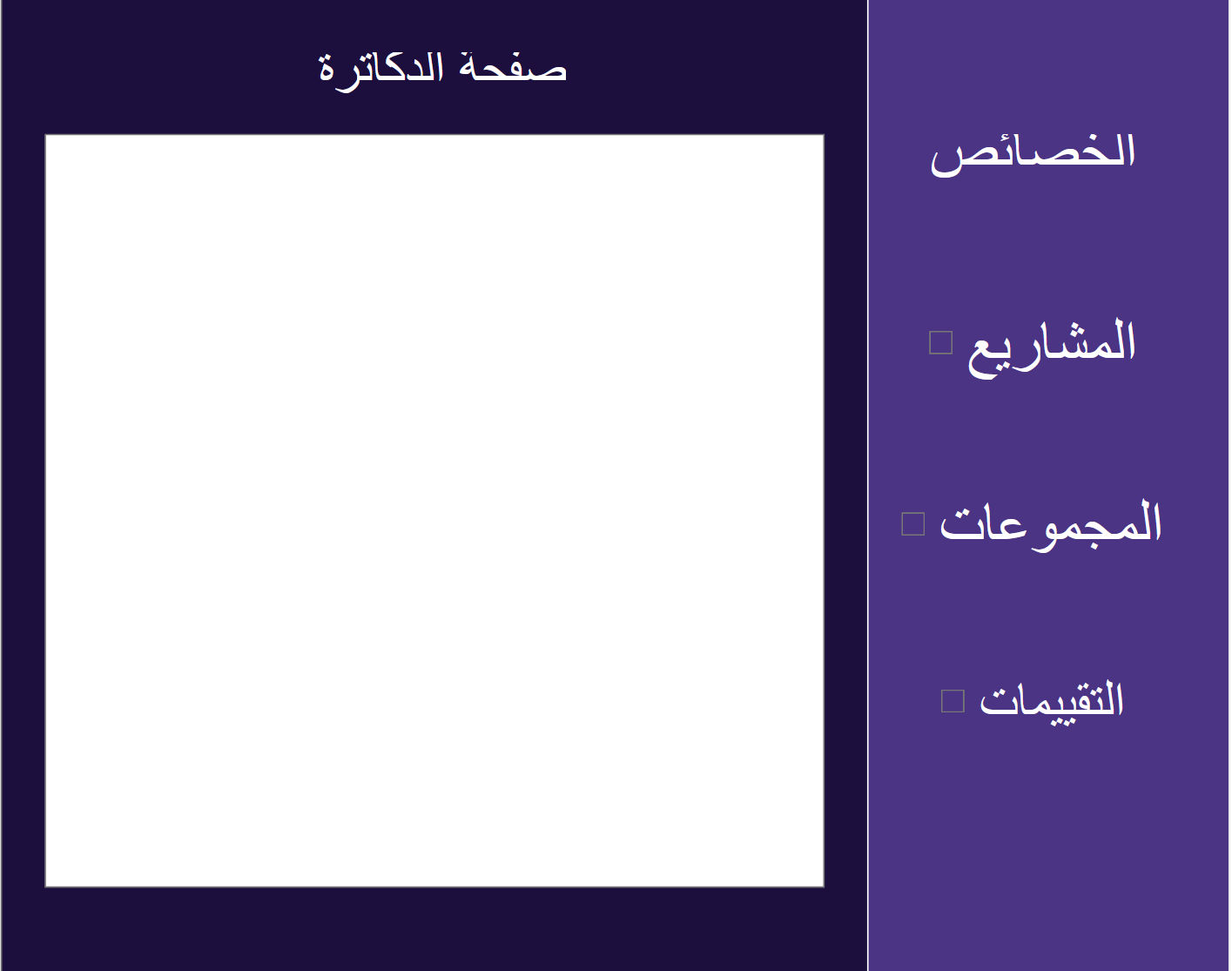
**Figure 6,7,8,9 shows the main pages**



**[6]**

****

**[7]**

****

**[8]**

****

**[9]**

* **Description of the ER diagram**

**Entity committees chairs**, the primary key is (id\_committees chairs) and the foreign key is (id\_department)

* the relationship, for each of the committees chairs belongs to a section of the college

**Entity committees chairs**, primary key is (id\_committees chairs)

Has a relationship with **Entity Examiners** and the primary key is (id\_Examiners) And the foreign key is (id\_groubs, id\_dcotors)

* The relationship is that committees chairs distribute the examiners to groups of students

**Entity committees chairs** (primary key is (id\_committees)

Has a relationship with **Entity Groups** and the primary key is (id\_Groups)

* The relationship is committees chairs Allowed him Edit group and add students

**Entity committee chairs**, primary key is(id\_commissions)

Has a relationship with the **entity evaluation** and the primary key is (id\_evaluation) and the foreign key is (id\_group, id\_doctor)

* The relationship is the committee chairs View group evaluation

**Entity committee chairs**, primary key is (id\_commissions)

Has a relationship with the **entity projects** the primary key is (id\_projects) and the foreign key is (id\_groubs, id\_doctors)

* The relationship is committee chairs Add the proposed section projects

**Entity Doctors**, primary key is(id\_docotors)

The foreign key is (id\_evaluation, id\_department, id\_groups)

Has a relationship with the **entity department**

* The relationship is for each Doctors to belong to a section of the college

**Entity Doctors**, primary key is (id\_docotors)

The foreign key is (id\_evaluation, id\_department, id\_groups)

Has a relationship with the **entity Groups** and the primary key is (id\_groups)

* The Relationship is the doctors Modify a group and create a group

**Entity Doctors**, primary key is (id\_docotors)

The foreign key is (id\_evaluation, id\_department, id\_groups)

Has a relationship with the **entity evaluation** and the primary key is (id\_evaluation) and the foreign key is (id\_groups, id\_doctors)

* The Relationship is the doctors, Group evaluation

**Entity Doctors**, primary key is (id\_docotors)

The foreign key is (id\_evaluation, id\_department, id\_groups)

Has a relationship with the **entity projects** and the primary key is (id\_projects) and the foreign key is (id\_groups, id\_doctors)

* The Relationship is the doctors See the group's ideas

**Entity students**, primary key is (id\_students)

And the foreign key (id\_department, id\_groups) Has a relationship with the **entity department**

* The Relationship is for each student belongs to a section of the college.

**Entity students**, primary key is (id\_students)

And the foreign key (id\_department, id\_groups) Has a relationship with the **entity groups** and the primary key is (id\_groups)

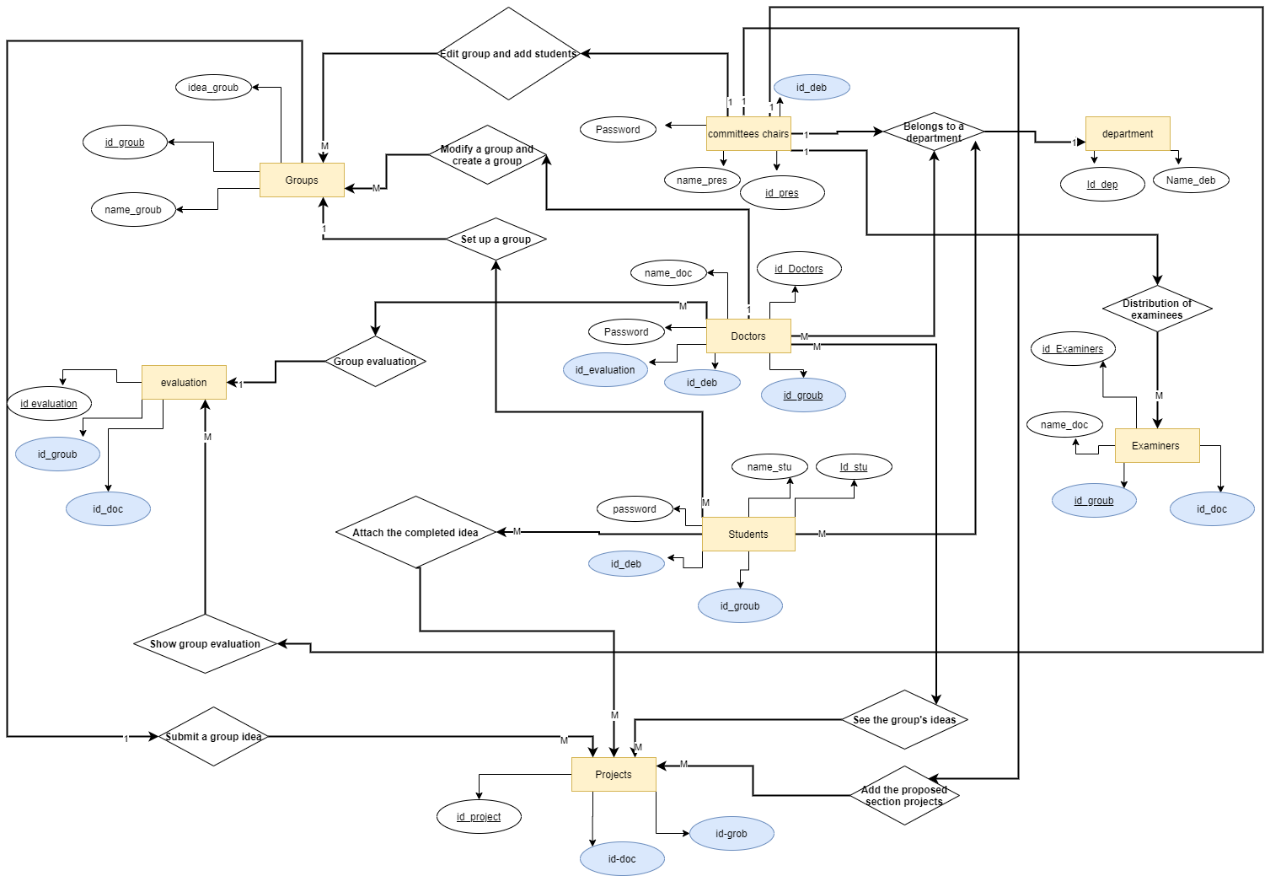
* The Relationship is Set up a group student

**Entity students**, primary key is (id\_students)

And the foreign key (id\_department, id\_groups) Has a relationship with the **entity projects**, the primary key is (id\_projects) and the foreign key (id\_doctors, id\_groups)

* The Relationship is Attach the completed idea

**As shown in Figure 1**



**[1]**

* **Data flow diagrams**

"Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage"[2]

**As shown below:**

* We have a flow of data from the **committee chairs** to the **department**:

" **Entity committee chairs** " he has the process of No. 1 "Create a record for committees chairs". Will be input to data store the " file committees chairs ", Then we have Process of No. 2 "get ID department committees chairs" which becomes the input for the **department**, Then the data store "file committees chairs" will be input to process of No.3 "Show the Department", Process No.3 will be the output for "committees chairs".

* We have a flow of data from the **doctor** to the **department**:

" **Entity doctor** " he has the process of No.1 "Create a record for doctor "will be input to data store the " file doctors", Then we have process No.2 "get ID department doctor", which becomes the input for the **department**, Then the data store "file doctors", will be input to process No.3 " Show the Department " Process No.3 will be the output for "doctors".

* We have a flow of data from the **student** to the **department**:

" **Entity student** " he has the process of No.1 "Create a record for student ", will be input to data store the " file student", Then we have process No.2 "get ID department student", which becomes the input for the **department**, Then the data store "file student", will be input to process No.3 " Show the Department " Process No.3 will be the output for "student".

* We have a flow of data from the **committees chairs** to the **examiners**:

" **Entity committees chairs** " he has the process of No.1 " create List of examiners", will be input to data store the "File of the examiners ", Then we have process No.2 " get ID examiners ", which becomes the input for the **examiners**, Then the data store " File of the examiners ", will be input to process No.3 " show the examiners " Process No.3 will be the output for " committees chairs and doctors ".

* We have a flow of data from the **examiners** to the **evaluation**:

" **Entity examiners** " he has the process of No.1 " Enter group evaluation", will be input to data store the " File of the evaluation", Then we have process No.2 " get ID evaluation", which becomes the input for the **evaluation**, Then the data store " File of the evaluation ", will be input to process No.3 " show the evaluation " Process No.3 will be the output for "examiners, doctors, and committees chairs ".

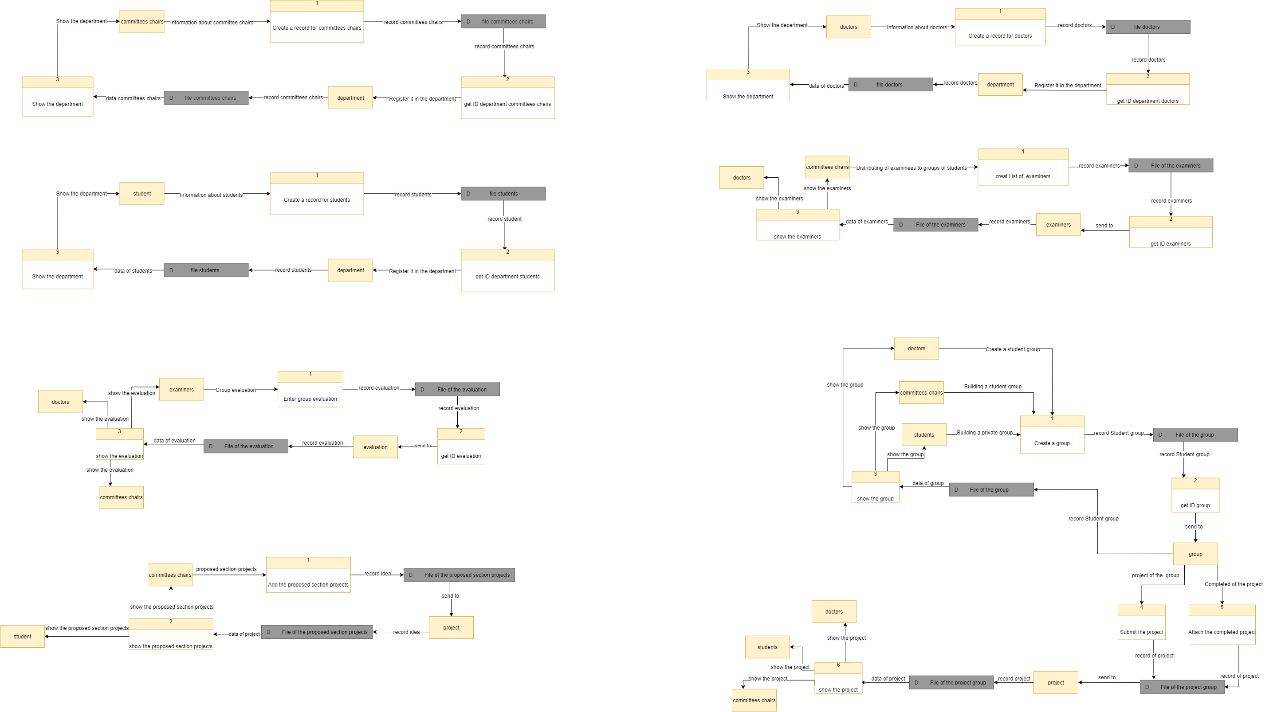
* We have a flow of data from the **doctors, committees chairs, and students** to the **group** Then we have a data flow from **group** to **project**:

"**Entity the doctors, committees chairs, and students** " he has the process of No.1 " Create a group", will be input to data store the "File of the group", Then we have process No.2 is " get ID group", which becomes the input for the **group**, Then the data store "File of the group ", will be input to process No.3 " show the group" Process No.3 will be the output for " doctors, committees chairs, and students ". "**Entity group**" he has the process of No. 4 " Submit the project" and process of No. 5 " Attach the completed project" will be input to data store the " File of the project group ", which becomes the input for the **project**, Then the data store " File of the project group ", will be input to process No.6 " show the project " Process No.6 will be the output for " doctors, committees chairs, and students ".

* We have a flow of data from the **committees chairs** to the **project**:

" **Entity Committees chairs**" he has the process of No.1 " Add the proposed section projects ", will be input to data store the " File of the proposed section projects ", which becomes the input for the **project**, Then the data store " File of the proposed section projects ", will be input to process No.2 " show the proposed section projects " Process No.2 will be the output for "student and committees chairs ".

**As shown in Figure 2**



**[2]**

* **Class digram**

"a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects."[3]

**As shown below:**

1. **we have class "committees chairs"**

**the operation**

1. login (id, password)
2. edit committees chairs ()
3. delete committees chairs ()
4. add committees chairs ()

* Has a relationship **one to one** with the class **department**, for every committees chairs It belongs to only one **department**.
* Has relationship **one to one or many** with the class **Examiners**, Distribution of examinees to groups.
* Has a relationship **one to many** with the class **Group,** create a group or modify a group.
* Has relationship **one to many** with the class **project**, Add suggested ideas to the section.

1. **we have class "** **Doctors"**

**the operation**

1. login (id, password)
2. edit doctors ()
3. delete doctors ()
4. add doctors ()

* Has a relationship **many to one** with the class **department**, for every Doctor It belongs to only one **department**.
* Has a relationship **one to one or mor** with the class **Group**, Create a group
* Has a relationship **one to one or mor** with the class **evaluation,** View group evaluation
* Has a relationship **one to many** with the class **project,** show group ideas

1. **we have class "** **Student"**

**the operation**

1. login (id, password)
2. edit Student ()
3. delete Student ()
4. add Student ()

* Has a relationship **many to one** with the class **department**, for every student It belongs to only one **department**.
* Has a relationship **one or many to one** with the class **group**, Create a group
* Has a relationship **many to one** with the class **project**, Attach the completed project and present the project of the department

1. **we have class "** **Examiners "**

**the operation**

1. edit Examiners ()
2. delete Examiners ()
3. add Examiners ()

* Has a relationship **one or many to one or many** with the class **evaluation**, Add group evaluation

1. **we have class "** **Groups "**

**the operation**

1. edit groups ()
2. delete groups ()
3. add groups ()
4. search groups ()

* Has a relationship **one to many** with the class **project**, Send the group idea

1. **we have class "** **department "**

**the operation**

1. delete department ()
2. edit department ()
3. add department ()
4. **we have class "** **evaluation "**

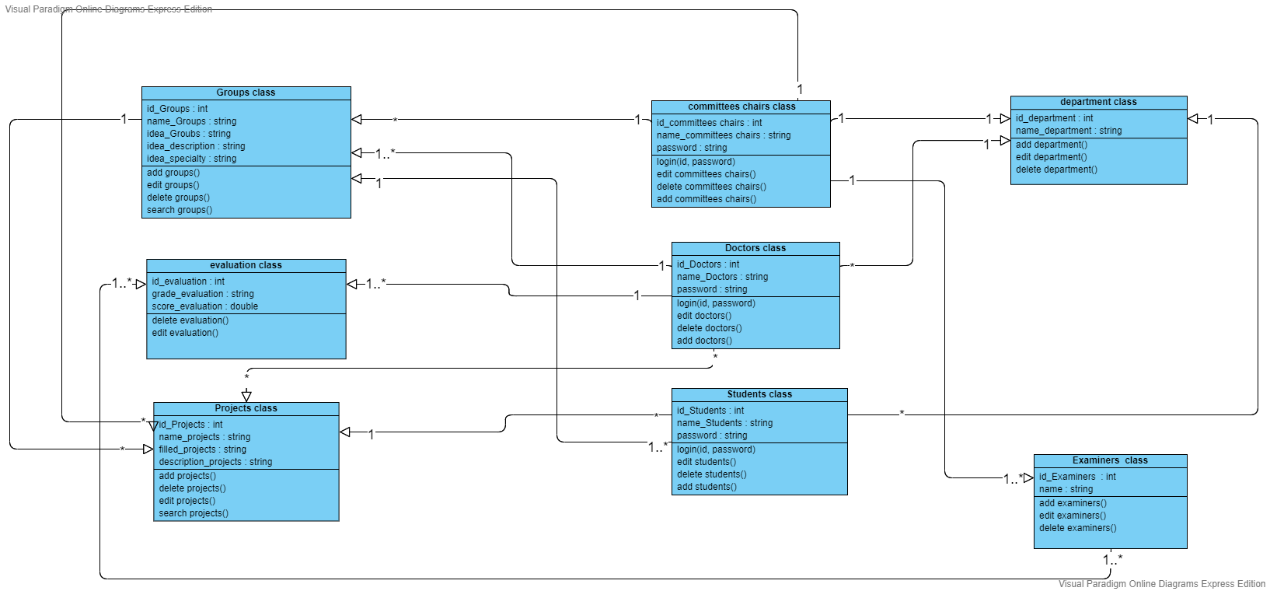
**the operation**

1. edit evaluation ()
2. delete evaluation ()
3. **we have class "** **Projects "**

**the operation**

1. edit groups ()
2. delete groups ()
3. add groups ()
4. search groups ()

**As shown in Figure 3**



**[3]**

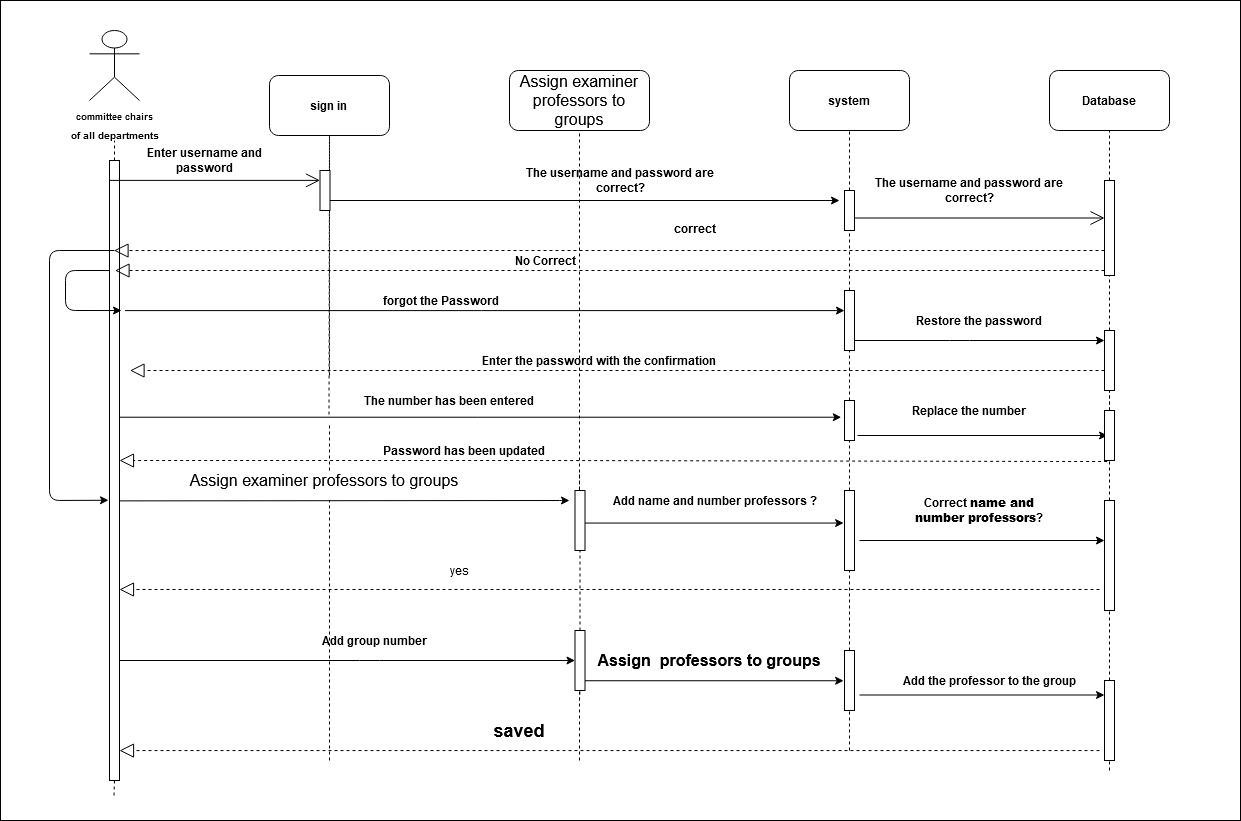
* **sequence diagram**

"A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place."[4]

* **sequence diagram of committee chairs of all departments**

A sequence diagram of the committee heads in all departments. The user logs in via the login page by entering the username and password and sending them to the system. The system requests the database. Is the entry correct There are two types of database responses. **The first answer Not Correct**, and the password will be recovered through the password reset page. The user requests the system to retrieve the password, the system requests the database to return the password, and the database will request the new password. The user enters the password and sends it to the system. Then the system sends a message to the database with the new password, and the database replies that the password has been updated, **the second response from the database is correct** and the user can log in to the account, After that, examinees move to professor groups. The user enters the professor’s name and number, then sends it to the system. The system receives it after that. Correct database and data requests. Then, the database replies as correct. After that, the system asks the database to add the professor to the group. The database is responding to the user who was added

**As shown in Figure 4**

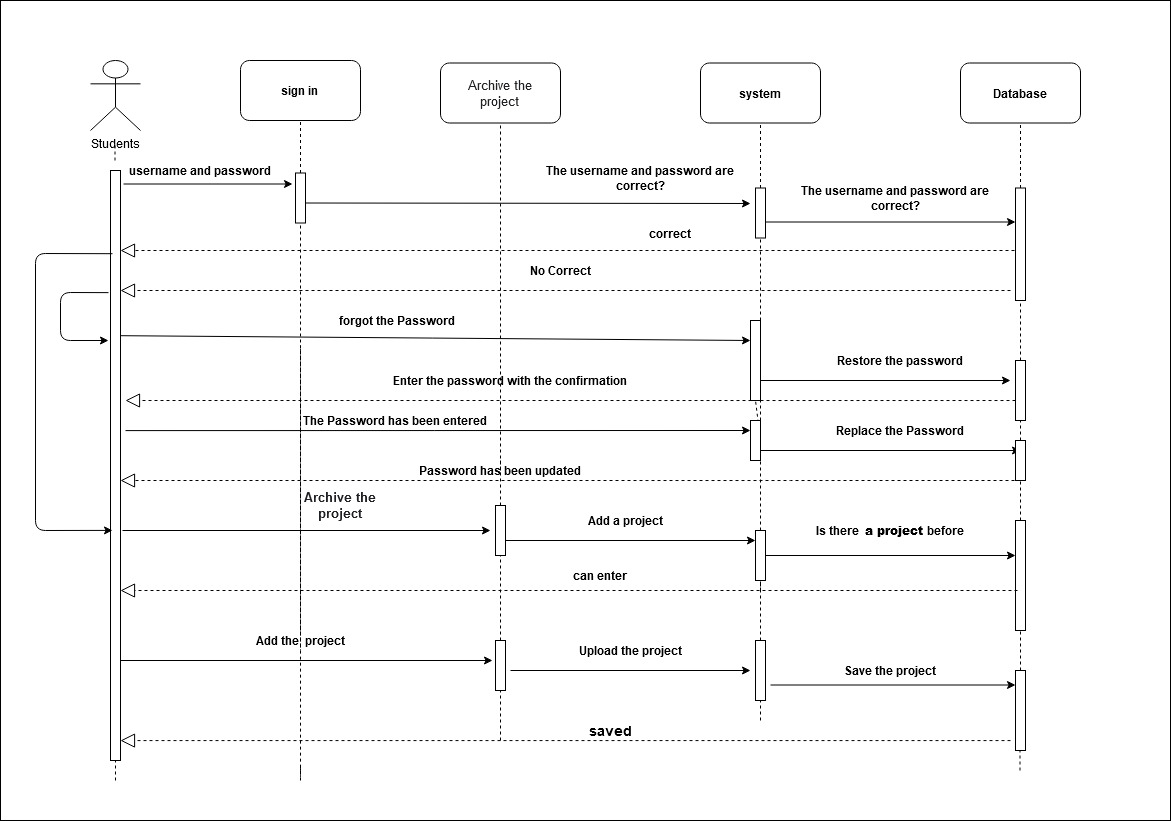
****

**[4]**

* **sequence diagram of Students**

A sequence diagram of the committee heads in all departments. The user logs in via the login page by entering the username and password and sending them to the system. The system requests the database. Is the entry correct There are two types of database responses. **The first answer Not Correct**, and the password will be recovered through the password reset page. The user requests the system to retrieve the password, the system requests the database to return the password, and the database will request the new password. The user enters the password and sends it to the system. Then the system sends a message to the database with the new password, and the database replies that the password has been updated, **the second response from the database is correct** and the user can log in to the account, After that, it goes to the project archive page and requests to archive the project and the system requests the database. Is there a project before? Can enter? The user enters the project and the system requests that the project be saved in the database

**As shown in Figure 5**

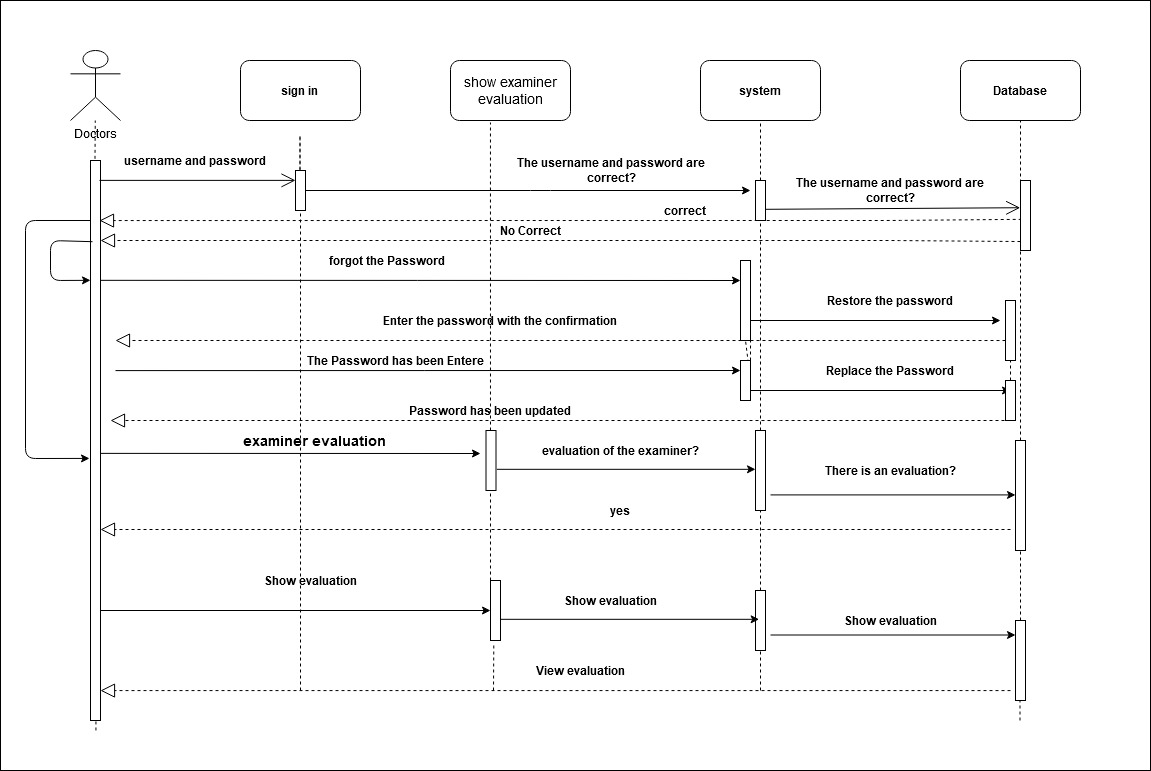
****

**[5]**

* **sequence diagram of Doctors**

A sequence diagram of the committee heads in all departments. The user logs in via the login page by entering the username and password and sending them to the system. The system requests the database. Is the entry correct There are two types of database responses. **The first answer Not Correct**, and the password will be recovered through the password reset page. The user requests the system to retrieve the password, the system requests the database to return the password, and the database will request the new password. The user enters the password and sends it to the system. Then the system sends a message to the database with the new password, and the database replies that the password has been updated, **the second response from the database is correct** and the user can log in to the account, User enters review examiner evaluation. After that, the user requests the system. Is there an evaluation of the examiner and the system requires the database? Is there an evaluation? Answer: Yes, the user requests the system to show the evaluation and the system requests the evaluation from the database.

**As shown in Figure 6**

****

**[6]**

* **Implementation ER**

"**MySQL Workbench:** is a visual [database design](https://en.wikipedia.org/wiki/Database_design) tool that integrates [SQL](https://en.wikipedia.org/wiki/SQL) [development](https://en.wikipedia.org/wiki/Software_development), [administration](https://en.wikipedia.org/wiki/Database_administration), [database design](https://en.wikipedia.org/wiki/Database_design), creation and maintenance into a single [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) for the [MySQL](https://en.wikipedia.org/wiki/MySQL) database system. It is the successor to DB Designer 4 from fabFORCE.net, and replaces the previous package of software, [MySQL GUI Tools Bundle](https://en.wikipedia.org/wiki/MySQL_Workbench#MySQL_GUI_Tools_Bundle)"[5]

We applied it to mysql and created the tables and relationships between them. The database was created, and it is ready for use.

* GitHub Repository Implementation ER

<https://github.com/NaifAliA/Implementation-ER/tree/main/Database>

* **HCI design, input, and output design:**

"**Human-computer interaction (HCI)** is a design field that focuses on the interfaces between people and computers."[6]

"The design decisions for handling input specify how data are accepted for computer processing. The **design of inputs** also includes specifying how end-users and system operators direct the system in performing actions. Output refers to the results and information that are generated by the system."[7]

* GitHub Repository Protype

<https://github.com/hamad57/final-project/tree/master/prototypeXD>

* Database Design

The database has been modified to suit the needs of the system.

Diagram

Description automatically generated

* Framework

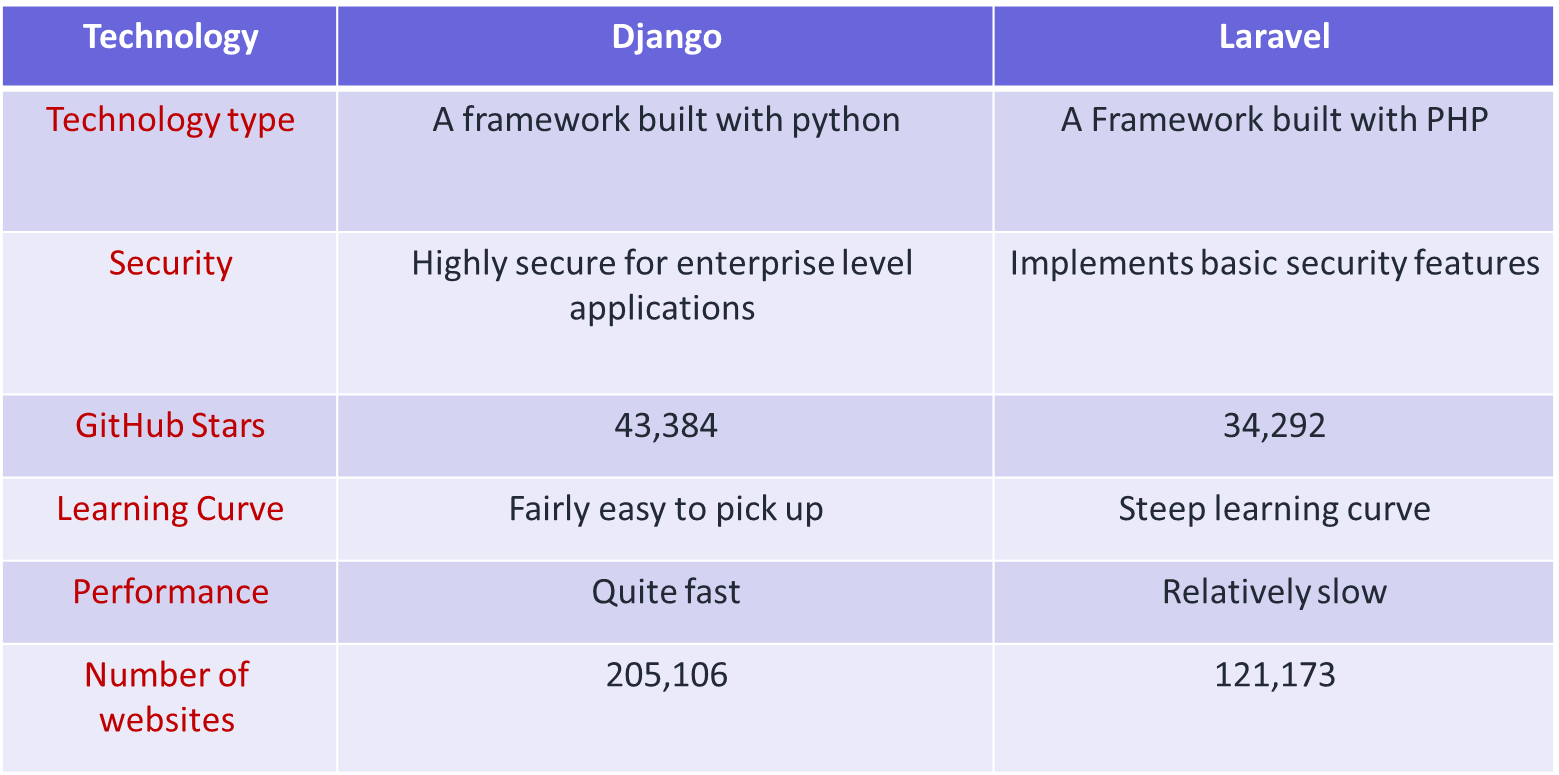
**why we use framework**.

"A Framework is a platform used as a foundation for developing application software frameworks are versatile, robust, and efficient."[15]

* Django VS Laravel

**"Django** is a model view template framework built with python. It’s a free and open-source framework that encourages rapid development and helps developers write efficient cleaner code. used by some of the enterprises in . Pinterest, Udemy, NASA, and Instagram"[[16](https://hackr.io/blog/django-vs-laravel)]

**"Laravel** is model-view-controller framework built with PHP which is one of the most famous languages of the web. It’s also a powerful framework used by 9GAG, UNION, Toyota Hall of fame"[16]



[16]

* Why we use Django framework ?
* Easy to Use.
* It’s fast and simple‏.
* Excellent Documentation for real-world application‏
* It’s secure‏.
* It suits any web application project.‏
* Why did we ignore Laravel ?
* Lack of continuity between versions
* Some promotions may be a problem.
* Not easy to use.
* Sources limited.
* Lack of artistic talent
* Difficulty with some updates
* **Django Framework Difficulties Solutions**
* **Modifying user data through the form**

calling library that is responsible for manipulating the input data > from Django import forms.

* **Establishing the connection between database and framework**

we switch to use MySQL database after facing some difficulties with establishing the connection between SQLite and Django framework.

* **sending / retrieving data from database**

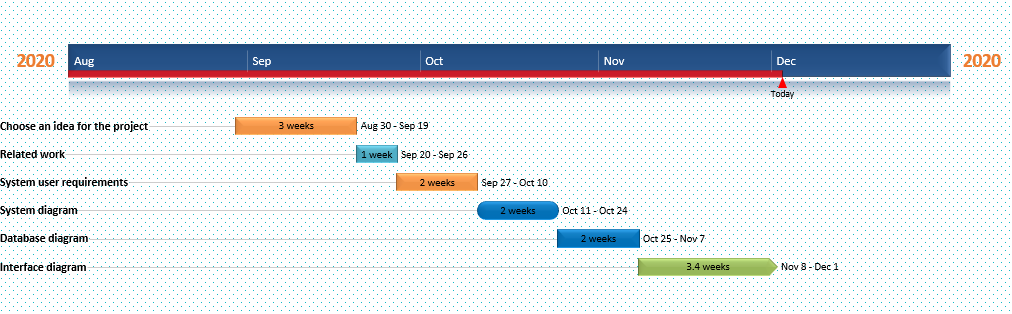
Applying some properties on some of the columns in our database to resolve the problem of retrieving and sending the data to/from the database. For example, applying the auto-increment properties with primary keys.

* Demo The Project

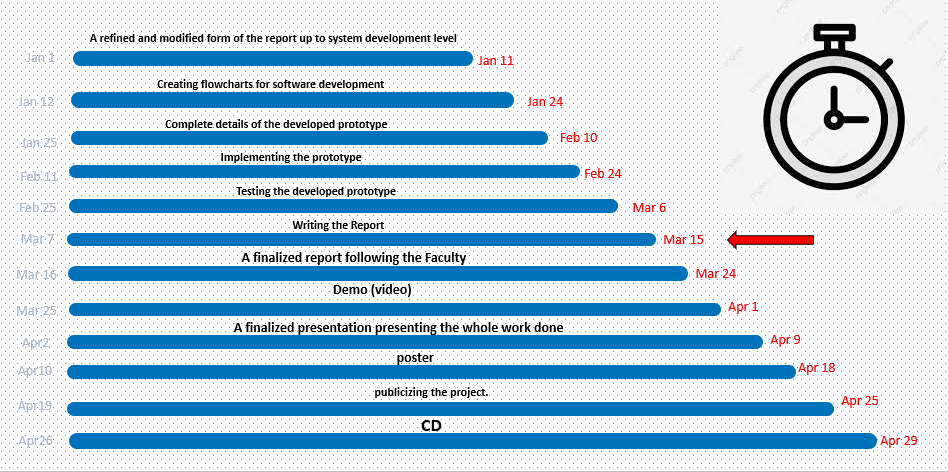
<https://github.com/hamad57/final-project/tree/master/>

**Gantt chart**

**First term**

****

**Second term**



* **References**

1. **Use Cases, usability.gov, November 27, 2020 U.S. General Services Administration Technology Transformation Services, from** [**https://www.usability.gov/how-to-and-tools/methods/use-cases.html**](https://www.usability.gov/how-to-and-tools/methods/use-cases.html)
2. **What is Data Flow Diagram, 2020 by Visual Paradigm from** [**https://www.visualparadigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/**](https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/)
3. **Class diagram, From Wikipedia the free encyclopedia, last edited on 10 November 2020, at 02:19 (UTC) from** [**https://en.wikipedia.org/wiki/Class\_diagram**](https://en.wikipedia.org/wiki/Class_diagram)
4. **Unified Modeling Language (UML) | Sequence Diagrams, GeeksforGeeks, 5th Floor, A-118, Sector-136, Noida, Uttar Pradesh – 201305, from** [**https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/**](https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/)
5. **What is MySQL Workbench? From**  [**https://en.wikipedia.org/wiki/MySQL\_Workbench#References**](https://en.wikipedia.org/wiki/MySQL_Workbench#References)
6. **HUMAN COMPUTER INTERACTION, Copyright © 2019-2020 Adobe., from** [**https://xd.adobe.com/ideas/principles/human-computer-interaction/**](https://xd.adobe.com/ideas/principles/human-computer-interaction/)
7. **Input(S) And Output(S), Copyright © Freetutes.com, from** [**https://www.freetutes.com/systemanalysis/sa5-inputs-and-outputs.html**](https://www.freetutes.com/systemanalysis/sa5-inputs-and-outputs.html)
8. **© 2021, Oracle Corporation and/or its affiliates,from** [**https://www.mysql.com/**](https://www.mysql.com/)
9. **2005-2021 Django Software Foundation and individual contributors. Django is a registered trademark of the Django Software Foundation, from** [**https://www.djangoproject.com/**](https://www.djangoproject.com/)
10. **This page was last edited on 28 February 2021, at 17:50 (UTC),from** [**https://en.wikipedia.org/wiki/HTML**](https://en.wikipedia.org/wiki/HTML)
11. **2021 AT Internet® - All rights reserved, from** [**https://www.atinternet.com/en/glossary/css/**](https://www.atinternet.com/en/glossary/css/)
12. **by the Bootstrap team with the help of our contributors. Currently v5.0.0-beta2. Code licensed MIT, docs CC BY 3.0,from** [**https://getbootstrap.com/**](https://getbootstrap.com/)
13. **Git is a member of Software Freedom Conservancy, from** [**https://git-scm.com/**](https://git-scm.com/)
14. [**https://github.com/**](https://github.com/)
15. **The Importance of Frameworks,Collection of JavaScript Frameworks : Copyright by Jose Aguinaga,from** [**https://benyoss4.medium.com/the-importance-of-frameworks-2c4a04d20ac5**](https://benyoss4.medium.com/the-importance-of-frameworks-2c4a04d20ac5)
16. **Django Vs Laravel: Which framework to choose?,Youssef Nader**

**Last Updated 01 Nov, 2020,from** [**https://hackr.io/blog/django-vs-laravel**](https://hackr.io/blog/django-vs-laravel)

* Some Django Sources Like Video And Text

1. [**https://youtu.be/Itr3aJAEWI0**](https://youtu.be/Itr3aJAEWI0)
2. [**https://youtu.be/\_lGpdAxYlxc**](https://youtu.be/_lGpdAxYlxc)
3. [**https://docs.djangoproject.com/en/3.1/**](https://docs.djangoproject.com/en/3.1/)