islington college
(इस्लिङ्टन कलेज)

LONDON METROPOLITAN UNIVERSITY

CS6P05 Final Year Project Computing

Interim Report

Alumni (Student) tracking system

Year and Semester

2019-2020

Student Name: Saugat Poudel

London Met ID: 17031123

College ID: np01cp4a170093

Assignment Due Date:

Assignment Submission Date: 2020/01/07

Word Count (Where Required):

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

External Supervisor:

Mr. Ishwor Shrestha

Internal Supervisor:

Miss. Subekshya Shrestha

Abstract:

The structure of this document is categorized into different part. First, it starts with introduction. Introduction part includes overall report scenario discussion. The main introduction to the project topic, the purpose of the topic and problem statement.

Similarly, next chapter is background which is totally based on research about the project which contains survey of the project, information gathered from different source about this project. It contains a review of similar system and review of technical aspects. Likewise, selection of software development methodology, their advantages and disadvantages are discussed with diagram.

Table of Contents

Table of Figures:

Table of Tables:

Chapter 1: Introduction

1.1: Topic discussion:

Every organization need growth, and inorder to grow an organization the system is required which will contains some rules and plans to be followed. System are defined by the organization holders and its entities. System includes data's and rules. As an organization grows, it must adapt to its new size inorder to succeed and continue growing. It must provide new structures to guarantee efficiency in managing the workflow and workload. It should provide a pattern for every works that are to be completed and are already completed. One of the most important things for an organization is information and human resource management. As more information are generated and more personal employed, the paper record keeping becomes unmanaged and old fashioned which can give bad results to the organization system management. Developing a digital records system for every data of organization becomes an important step in the maturing of an organization in this digital age. (Burke & Noumair, 2015)

The main root of any organization is data and entities. The project is about college and university which is also an organization which requires data of students, teachers, alumni, faculty, administration and so on to run effectively. As we discussed above many universities are facing problems due to unmanaged database and to overcome this problem this project is made.

This project is about the alumni Tracking system. The project is based on mobile application. The project also includes a database where all the records of users are stored. Tracking the alumni can be done by both college and students of that community. the use of this application is to keep records of students and alumni inorder to track them in future when necessary and make a bond with them. Generally, university only tracks the enrolling students, but it is necessary to track the

alumni's too so that they can be in touch when they need any kind of helps. This application helps university to create a separate database for alumni and students.

1.2: Purpose

A university has done a lot of effort to make their student capable in their field. Collecting the best equipment's and information inorder to teach them and make their student more capable enough in their field is the main motto of college. But after graduations there are no more links between college and alumni. They start focusing on their carrier and college start focusing on graduating new students.

The Purpose of this project is to make a bond between college their students and alumni's even after they are graduated. To maintain the records and make a link between college students and alumni this project is build.

1.3: Problem statement

Most of the university only keeps the data record of the students who are currently enrolling in their college. They simply delete the data once the students are graduated or they left the university to maintain fresh data. They start focusing on only the present students. Due to such bad practice college are not able to identify their own students after some years.

If communication between college and alumni stops once alumni leave an institution, their understanding of the university become stale. Instead they should be kept informed so they can remain engaged and keep abreast on the progress of the university.

Good alumni relationship brings many benefits to both the institution and the alumni.

Every student is better on their own faculty and has done many achievements in their field. But when the university need them in future, they already become stranger to them. Not only university, students also need help from their university in future even after their graduation. Then that time due to lack of records of student they will not be able to be in touch with each other. They will not be able to recognize their own students.

Students who are currently enrolling in the university are also facing many problems due to unproper link between alumni. Most of the time during study syllabus and guidance students want to communicate with the alumni. Sometimes students don't get the guidance of teacher and may want to get help from alumni because they are already gone through that phage while knowing about college rules and study materials.

1.4: Project as solution

This project is about tracking the alumni and keeping their records so that college and students of their respective college can be in touch with each other. In this project there will be a separate database of alumni with their current address, the jobs they are engaged in and their achievement. These details can be visualized by both students and college of the same community. I have also taken a survey before making this application. To know about what things are hampering to students and alumni during such improper management system. The survey was online question answer form to all the students and alumni about this project. (The survey result is attached in Appendix section B).

1.5: Structure of report

Chapter 2: Background: This chapter contains the overall research of the project. To build a project research is the most important things. To research about the problems and its feasibility many websites, journals and books are used. this chapter includes research work which is related to the project with the respective citation and references of different books and journals. This chapter also contains comparison between similar projects, and a brief explanation of how it is different from others.

Chapter 3: Development methodologies: Every project is done according to software development methodology. This chapter of the project gives the brief description of which methodology was used and how it is suitable according to this project scenario.

Chapter 4: Diagrams and prototype: After the research the next phage is designing and prototyping phage where the entities and attributes are listed, and workflow of the project is shown. This chapter contains different UML diagrams like Use cases, Entity relationship diagram, and the prototype of the project.

Chapter 5: Testing and analysis: The testing section of the report provides details about the tests conducted for the project. It shows all the test cases and defines their applicability in the project. Whereas, the analysis section provides a breakdown of the developed prototype with respect to the user response.

Chapter 6: Conclusion:

The conclusion of this report provides a review of how people were facing problems legally, socially, and ethically due to not having such system in college and what will be the scope of this type of project in brief.

Chapter 2: Background

2.1: About the end users

A targeted group of users are admin, college, alumni and students who can view their data. Users experiences are different from usability where the main objectives of it are to fulfill needs among the users. Every individual has their own thought in everything however the result from the survey has helped in understanding every individual opinion.

Alumni Management system is for organizational establishment to manage data, easily tracking of alumni's progress and maintain a long-time relation. The questions in the survey is set up according to the need of students, college and alumni's so that this application can be applicable to the wide range of users which will used to track alumni's records and their interaction. According to the survey most of the students are facing the problems due to not having proper interaction with alumni. the survey is also for alumni so most of the alumni response was there are no such medium so that they can be in touch with their college and students. When they come to know that this kind of system is going to be developed in future, I found them very happy and got supportive to do this project.

2.2: Understanding the System:

1. Software Architecture:

The architecture of alumni tracking application is implemented on MVC. I have used Laravel for my backend API creation. "Laravel is a web application framework of PHP with expressive elegant syntax. Laravel is accessible, yet powerful, providing powerful tools needed for large, robust applications. A superb inversion of control container, expressive migration system, and tightly integrated unit testing support give you the tools you need to build any application with which you

are tasked. Laravel attempts to take the pain out of development by easing common task used in most web projects, such as authentication, routing, session and caching. It follows MVC pattern." (laravel, 2011).

I used this framework because I already have knowledge of php. Laravel helps us creating web application where we can link it with different popular databases like MySQL. this also supports bootstrap from which we can make pleasing web user interface. Laravel is easier for making application programming interface from which we give our data access to not only web application but also to the mobile application. As my project is also based on mobile application i used Laravel for backend.

About Model View Controller (MVC):

The MVC triad of classes were used to build user interfaces in smalltalk-80 in 1988. MVC is an architectural pattern that is used in software engineering, whose fundamental principal is based on the idea that the logic of an application should be separated from its presentation. It divides a given software application to three interconnected parts, as Model, View and Controller. MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects.

MVC Components:

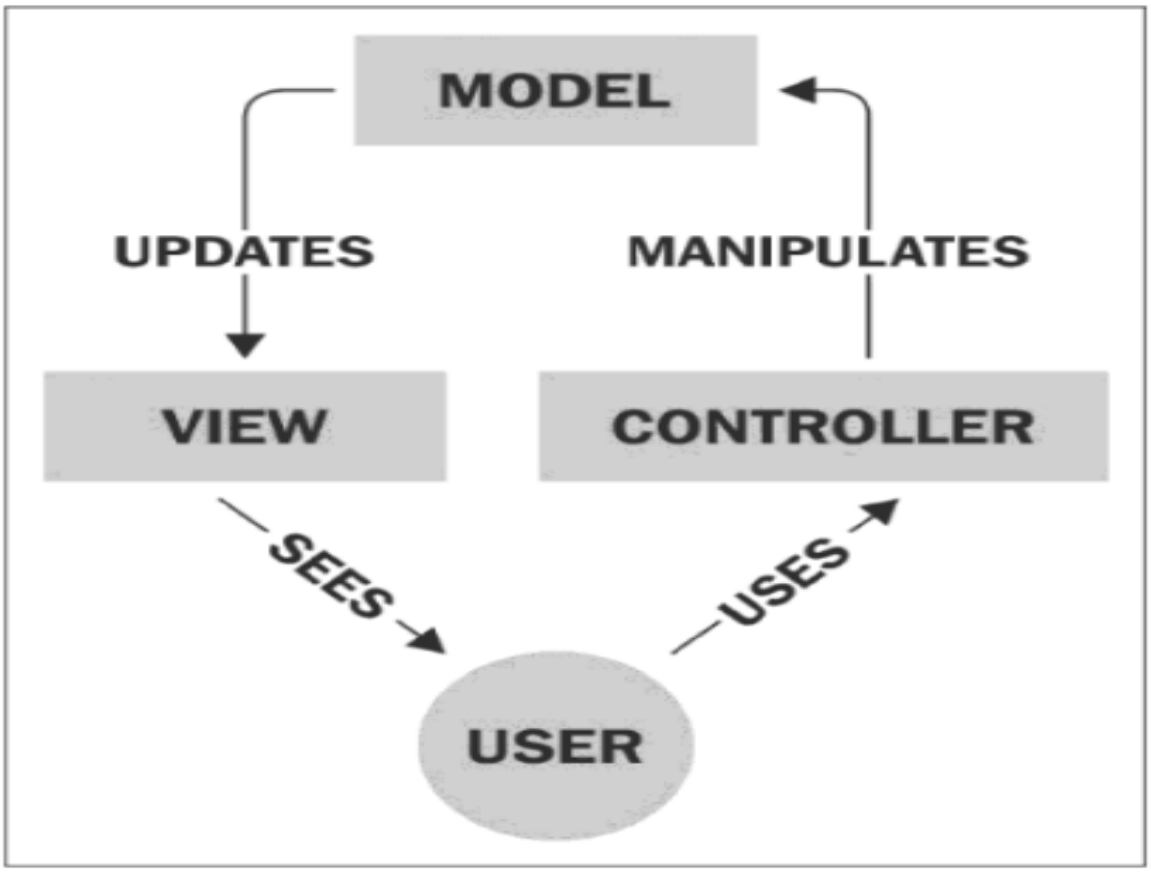


Figure 1 MVC structure.

Model: The model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the view and controller components or any other business logic-related data. For example, in the case of this project alumni object will retrieve the alumni data from the database, manipulate it and update it data back to the database.

View: The view component is used for all the UI logic of the application. For example, when a student or alumni or a college want to visit each other profile then they will visit different UI pages which contains UI components like text boxes, images, buttons.

Controller: Controller act as an interface between model and view component to process all the business logic and incoming requests. Manipulate data using the model component and interact with the views to render the output. For example, user's controller will handle all the interacts and inputs from the users view and update the database using the user model. The same controller will be used to view the user's data.

The benefits of using MVC patterns on Laravel are as follows:

- Different views and controllers can be substituted to provide alternate user interfaces for the same model.
- It provides multiple simultaneous views of the same model.
- The change propagation mechanism ensures that all views simultaneously reflect the current state of the model.
- Changes affecting just the user interface of the application becomes easier to make.
- It is easier to test the core of the application, as it is encapsulated by the model.
- One of the great benefits of the MVC pattern is that it allows you to recycle the application's logic when you want to implement an external API inside a part of your application's logic. If the MVC approach of Laravel is followed thoroughly, you will only need to modify the controller to render many different templates/views (Kilicdagi & Yilmaz, 2014).

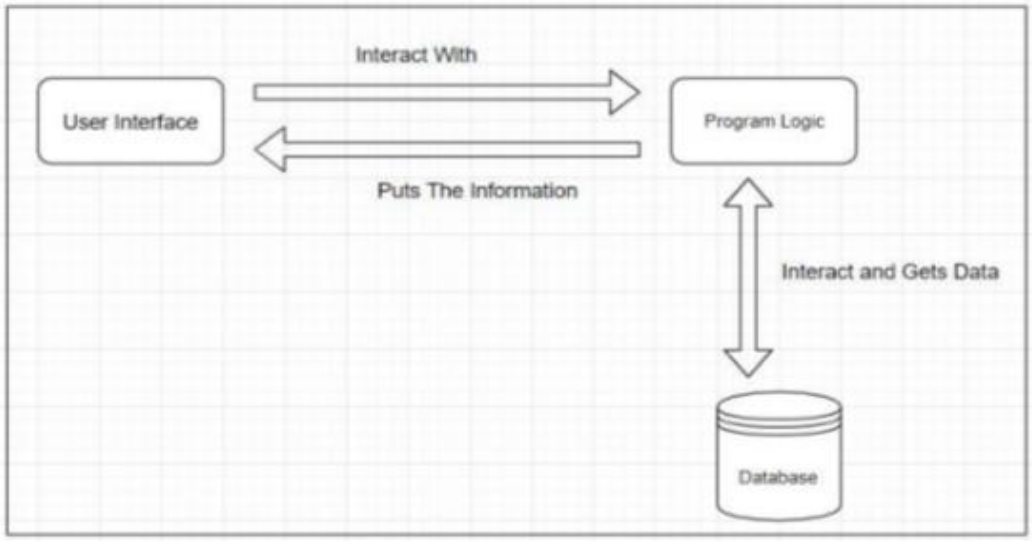


Figure 2 MVC working mechanism

Let's take an example of Alumni module which I have implemented in MVC architecture. When the Alumni wants to add a new achievement, he first goes to his Profile and sends a request with the required data to the controller. The controller handles the request with the business logic and map the data to the model and save the achievement to the user's database and sends response to the alumni with the status code of 200 i.e. Achievement added successfully. (Sinha, 2019)

2.3: Review of Similar System

Alumni management application by Sales for Education:

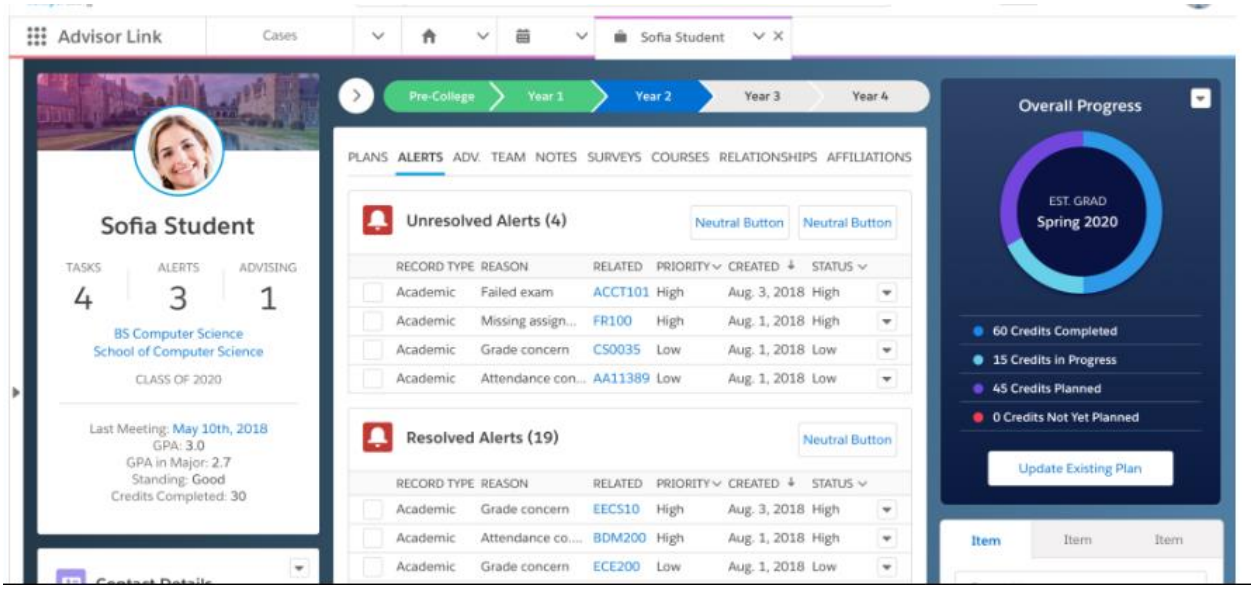


Figure 3 Similar system 1

This is the web-based application about alumni. this application records the data of alumni and students. And this only the web-based application and cannot be used from mobile. And it is only free for demo and if we want to buy it then it is very much expensive in compare to other such application. In this application if user want to update some records then must contact admin which I found time consuming and old fashioned (salesforceforeducation, 2000) .

Alumni management application by alma student information:

Figure 4 similar system 2

Alma is the web application mainly focused on student and alumni management system. While reviewing their software I found them giving more importance to student registration rather then focusing on alumni. this software register's student gives report cards and view alumni. their User interface is good, but their focus is towards student not alumni. (www.getalma.com, 2012)

Comparing the above application with this project:

After researching about 2-3 more similar projects I found that they are mainly focusing on student management rather then alumni management. They have named it as alumni management, but they have only included basic registration of alumni. as their application is web based only it might not be feasible to all the users because most of the people are now using mobile application for convenience.

This application will not only keep the records of alumni but will track their location, send events and will contact each other with mobile application. The application is made with flutter technology so it will be useful for both android and IOS users. They can register through mobile visit alumni with just mobile application and watch details.

So, in contrary think alumni management system will be different and more useful then other similar products found in markets.

Chapter 3: Development

3.1. Considered Methodologies

3.1.1: Waterfall methodology:

The waterfall model originated in 1970 largely through the efforts of dr. Winston Royce. It worked well at the time and has undergone many subsequent changes and revisions. The methodology is currently one of the most widely used software development methodology.

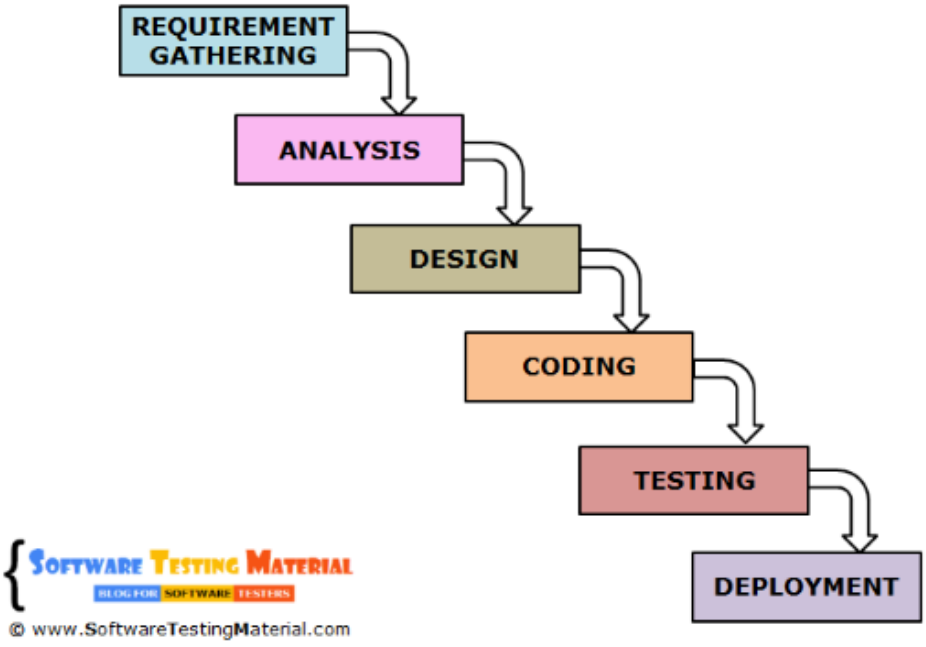


Figure 5 Waterfall model.

Phases of waterfall methodology are:

- Requirement analysis.
- Design
- Code and unit testing
- Subsequent testing
- System testing

Benefits of a waterfall methodology:

- It provides phase-by-phase checkpoints for the project.
- You may need to proceed to only the following phase after the previous phase has been completed.
- It can also be applied to an iterative approach.

Disadvantages of waterfall methodology:

- There is minimal feedback between project phases.
- You start seeing results only later in the life cycle.
- Each phase is tracked with far too many hard dates and milestones (Charvat, 2003).

3.1.2: Spiral methodology:

Spiral methodology is the combination of waterfall and iterative methodology. Each phage in spiral model begins with a design goal and ends with the client reviewing the progress.

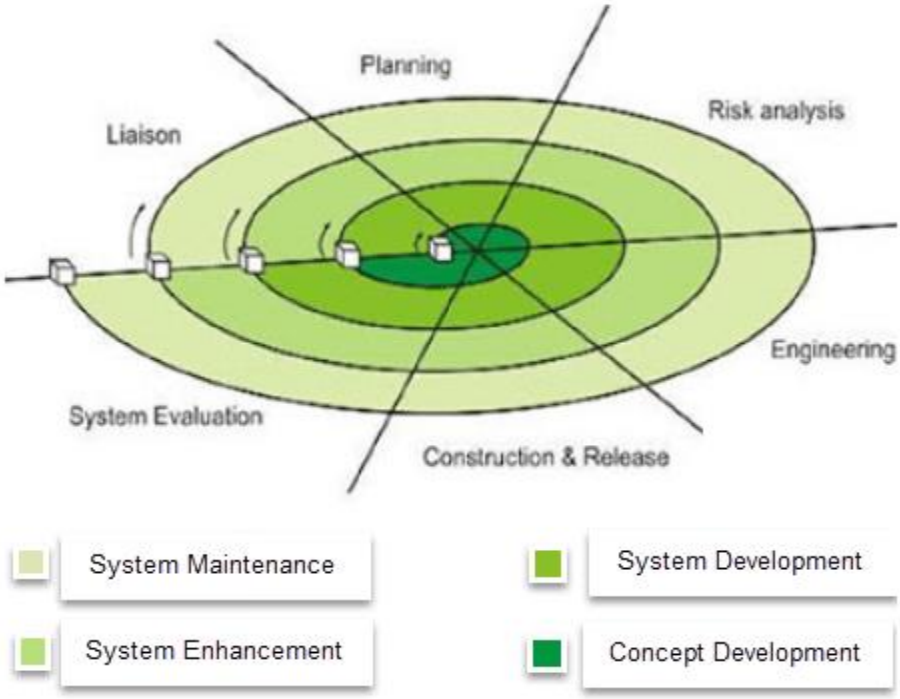


Figure 6 Spiral model.

Phases of spiral methodology:

- Planning
- Risk analysis
- Engineering
- Evaluation

Advantages of spiral methodology:

- Additional functionality or changes can be done at a later stage.
- Cost estimation becomes easier as the prototype building is done in small fragments.
- Continuous or repeated development helps in risk management
- Development is fast and features are added in systematic way.

Disadvantages of spiral methodology:

- Risk of not meeting the schedule or budget.
- It works best for large projects only also demands risk assessment expertise.
- For its smooth operation spiral model protocol needs to be followed strictly
- Documentation is more as it has intermediate phases.
- It is not advisable for smaller project, it might cost them a lot. (Guru99, 2020)

3.1.3: Rational Unified Process (RUP) Methodology:

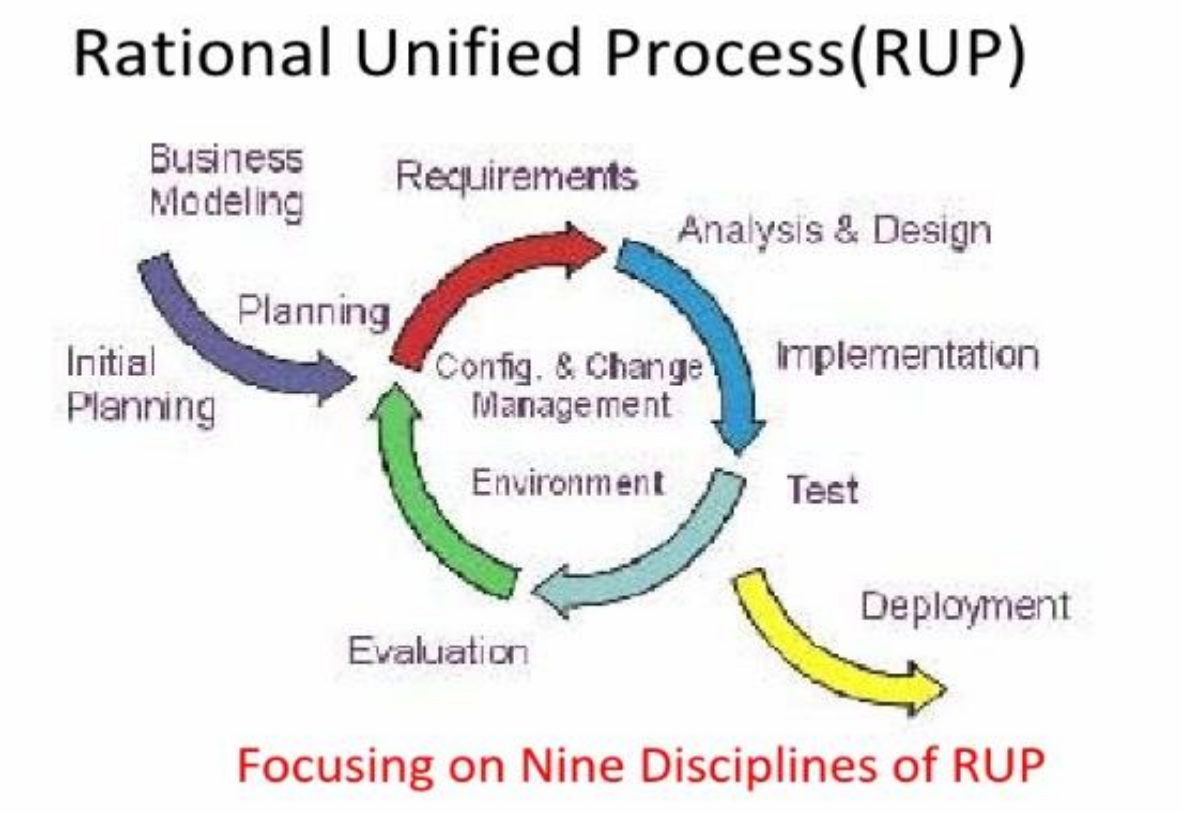


Figure 7 Rational Unified Process.

(Kumar, 2009)

The Rational Unified Process framework was initially created by the Rational Software Corporation, which was bought out by IBM in 2003. RUP is based on few fundamental ideas, such as the phases of development and building blocks, which defines who, what, when, and how development will take place (IBM, 1998).

The Rational Unified Process Best Practices:

- Develop Software Iteratively: Encourages iterative development by locating and working on the high-risk elements within every phase of the software development life cycle.
- Manage Requirement: describes how to organize and keep track of functionality requirements, documentation, tradeoffs and decisions and business requirements.
- Use Component-Based Architectures: Emphasizes development that focuses on software components which are reusable through this project and most importantly within future projects.
- Visually Model Software: Based on the Unified Modeling language (UML) the Rational Unified Process provides the means to visually model software including the components and their relationship with one another.
- Verify Software Quality: Assists with design, implementation and evaluation of all manner of tests throughout the software development life cycle.
- Control Changes to Software: Describes how to track and manage all the forms of change that will inevitably occur throughout development, inorder to produce successful iterations from one build to the next. (IBM , 2003)

3.2: Selected methodology

3.2.1: Rational Unified Process (RUP):

After learning all the above methodology. I came to conclude that rational unified process is best for this project. It is easily to understand. We can change the requirement throughout the development cycle. In my project I have added unnecessary features and data's in the project later in development I realized that some of the data are unnecessarily repeating so as my methodology is Rup, I was able to change the requirement. The project requirement is not necessary to know in beginning when using this methodology.

Advantages of Rational Unified Process:

- Allows for the adaptive capability to deal with changing requirements throughout the development life cycle, whether they be from customers or form with the projects itself.
- Emphasizes the need (and proper implementation of) accurate documentation.
- Diffuses potential integration headaches by forcing integration to occur throughout development specifically within the construction phase where all other coding and development is taking place. (Morse, 2017)

The Four Life Cycle Phases of Rational Unified Process:

- Inception Phase: During this phage the basic idea and structure of the project is determined. The team will sit down and determine if the project is worth pursuing at all based on the proposed purpose of the project, the estimated costs (monetary and time) and what resources will be required to complete the project.
- Elaboration Phase: The purpose of the elaboration phase is to analysis the requirements and necessary architecture of the system. The success of this phase is particularly critical, as the final milestone of this phase signifies the transition of the project from low-risk to high-risk.
- Construction Phase: Construction phase occurs when the coding and implementation of all application features will take pace. This period is also where integration with other services or existing software should occur.
- Transition phase: Easier thought of deployment, the transition phase is when the finished product is finally released and delivered to customers. However, the transition phase is more than just the process of deployment; it must also handle all post-release support, bug fixes, patches and so forth.
- Iterations: The RUP also recommends that each of the four above phases be further broken down into iterations, a concept taken from agile and other common iterative development models. Just as with other models, in the context of RUP an interatom simply represents full cycle of the core phases, until a product is released in some form internally or externally. From this baseline the next iteration can be modified as necessary until, finally, a full and complete product is released to customers. (Morse, 2017)

Normalization of the system:

Un- Normalized Form (UNF):

(User_id, Name, email, phone, address, achievement, {role} faculty, job) (event_id, event_name, event_venue, event_time)

here event table don't have any link with other table, so it is already normalized. here a user can have multiple role because a student can be alumni in future.

First Normal Form (1NF):

Users-1: User_id(PK), role_id(FK), Name, email, phone, address, achievement, faculty, job

Role-1: role_id(PK), role

Events-1: event_id(PK), event_name, event_venue, event_time

Second Normal Form (2nf):

Users-2: User_id(PK), role_id(FK), Name, email, phone, address, achievement, faculty, job

Events-2: event_id(PK), event_name, event_venue, event_time

Roles-2: role_id(PK), role

Third Normal Form (3NF):

No anomalies are found in 2nf. hence the table is already in 3nf...

Tools used:

1: Programming:

1.1: Laravel:

Laravel is accessible, yet powerful, providing powerful tools needed for large, robust applications. A superb inversion of control container, expressive migration system, and tightly integrated unit testing support give you the tools you need to build any application with which you are tasked. Laravel attempts to take the pain out of development by easing common task used in most web projects, such as authentication, routing, session and caching. It follows MVC pattern." (laravel, 2011). I used this programming for developing backend creating API's and tables.

1.2: Flutter:

Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. Its main features are faster development, expressive and flexible UI, Native performance. (flutter, 2020)

1.3: Firebase:

A comprehensive app development platform which can build apps fast, manage infrastructure, where functionality like analytics, databases, messaging and crash reporting is given. It is made by google. I am using this platform for sending notifications. (Firebase, 2020)

3.3: Diagram and Prototypes:

3.3.1: Use Case Diagram:

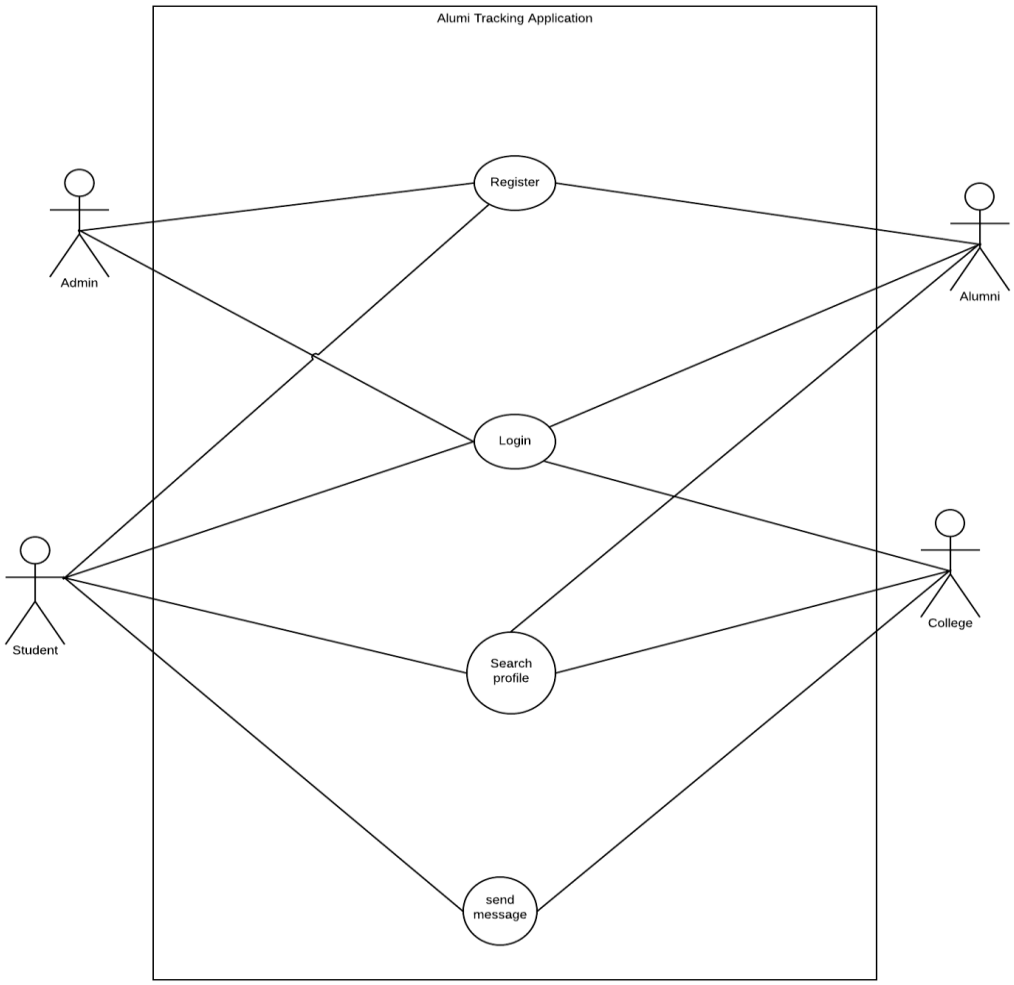


Figure 8 Use case diagram

Research on Use case diagram:

Steps involved while making use cases:

- Are started by an actor
- Are provided by the system

- Can involve more than one actor
- Describe how a system and its actors collaborate to the actor's goals
- Provide a coherent picture of how the system will be used and what it does.

Connecting actors and Use cases:

The system and the actors interact with each other by sending signals or messages to one another. To make such interaction we use arrow line by connecting it with the use case (Bittner & Spence, 2003).

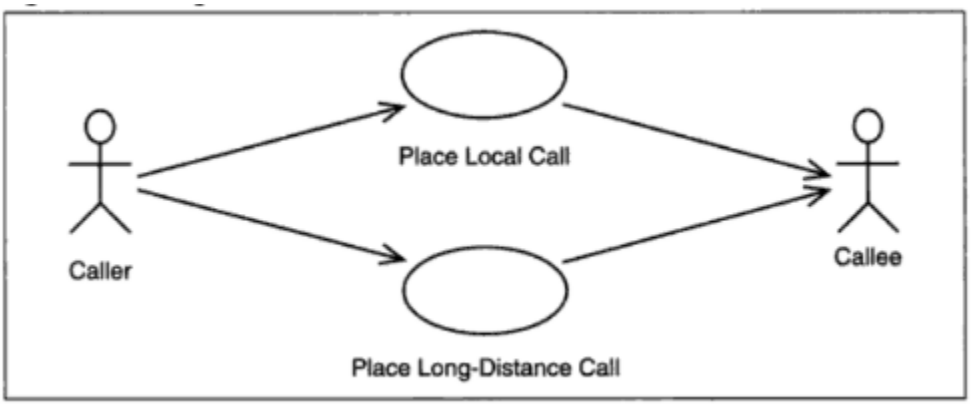


Figure 9 Use case research 1

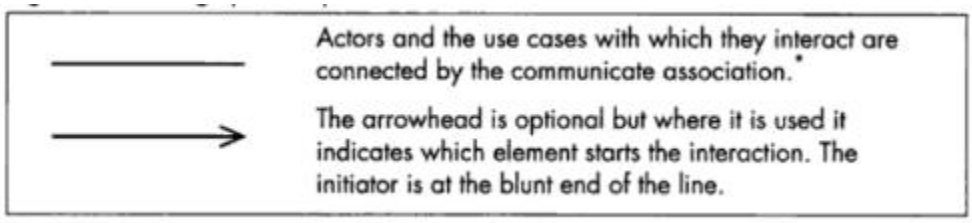


Figure 10 Use case research 2

Use case description:

The above use case shows the overall workflow of alumni tracking application. Users will use application to interact with the program. Here users are divided into 4 categories as actors. Alumni,

college, students, admin. The above use case is general view of the application. The more detailed use case is described in the high-level use case diagram. All the users can register each other through the application. Every user except admin can interact in application whereas admin will control other users from the web application. Student, college and alumni can view each other's profile and send message.

3.3.2: High level use case diagram

High level use case diagram for Register:

Use case: Register

Actor involved: Admin, Student, Alumni

Description: Admin can register a college. Whereas student and alumni can register themselves with the application.

High level use case diagram for Login:

Use case: Login

Actor involved: Admin, College, Student, Alumni

Description: Every user can login themselves with application, but admin will only login with web.

High level use case diagram for Update profile:

Use case: Update Profile

Actor involved: Alumni, college, student, admin

Description: Every user will be able to update their own profile

High level use case diagram for Add achievement:

Use case: Add achievement

Actor involved: Alumni

Description: Alumni can only add the achievements they have done but it can be viewed by all the users.

High level use case diagram for Events:

Use case: Events

Actor involved: College, Alumni

Description: College can add new events and alumni can view the events.

High level use case diagram for Search profile:

Use case: Search Profile

Actor involved: College, students, Alumni

Description: All the users can view, and search profiles accept of admin. Admin can only view through web.

High level use case diagram for send message:

Use case: Send message

Actor involved: Alumni, college, student

Description: All users expect admin can send message to each other.

3.3.4: Extended Use Case Diagram

Use case: Register

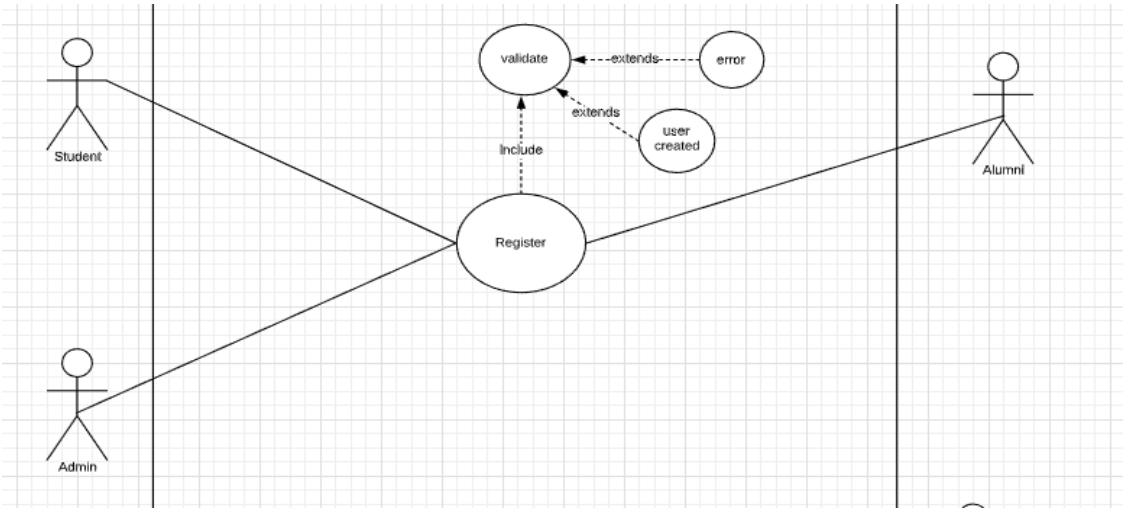


Figure 11 extended use case register

Actor involved: Admin, Student, Alumni

Description: Admin can register a college. Whereas student and alumni can register themselves with the application.

User interaction	System response
1. User (Admin, Student, Alumni) click on register button from application dashboard.	2. System shows a option to register either as alumni or student.
3. User selects any one option.	4. System lands the user to the appropriate registration page.
5. User enters all the details and clicks on register button.	6. System checks the user's validation and field validation and give appropriate response.

Table 1 Extended use case register.

Extended use case diagram for Login:

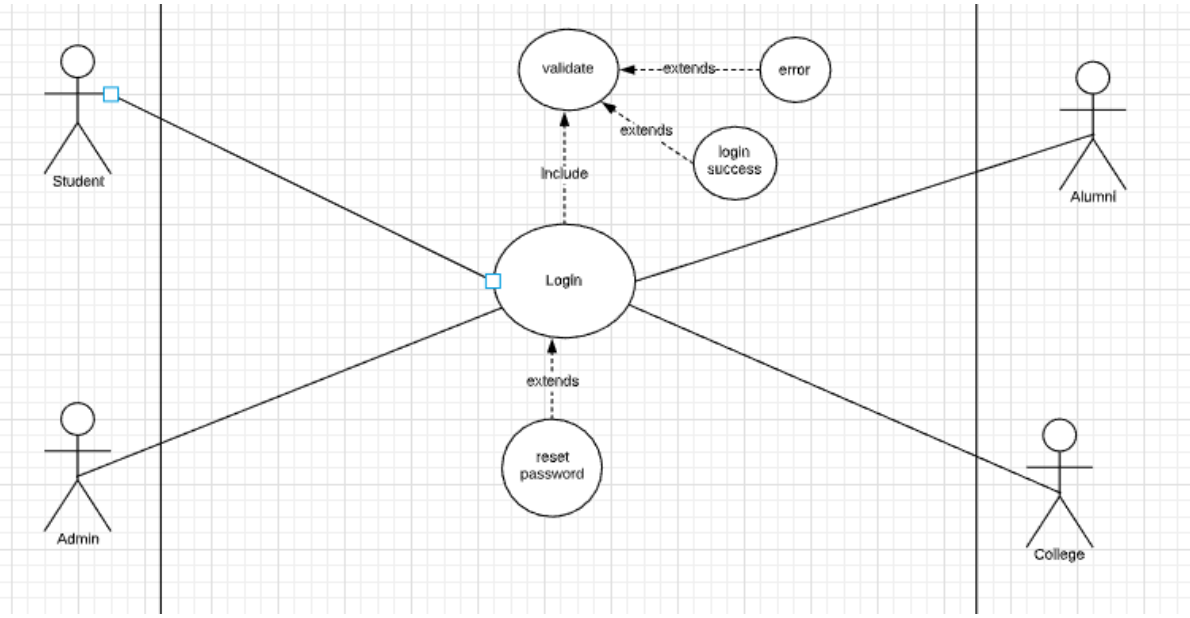


Figure 12 extended use case login

Use case: Login

Actor involved: Admin, College, Student, Alumni

Description: Every user can login themselves with application, but admin will only login with web.

User interaction	System Response
1. User (all) clicks on login page.	2. System Shows login page to users.
3. User enters the valid login credentials.	4. System checks the user's identity and verify user if user verified lands the user to appropriate page. If not shows error response.

Table 2 Extended use case login.

Extended use case diagram for Update profile:

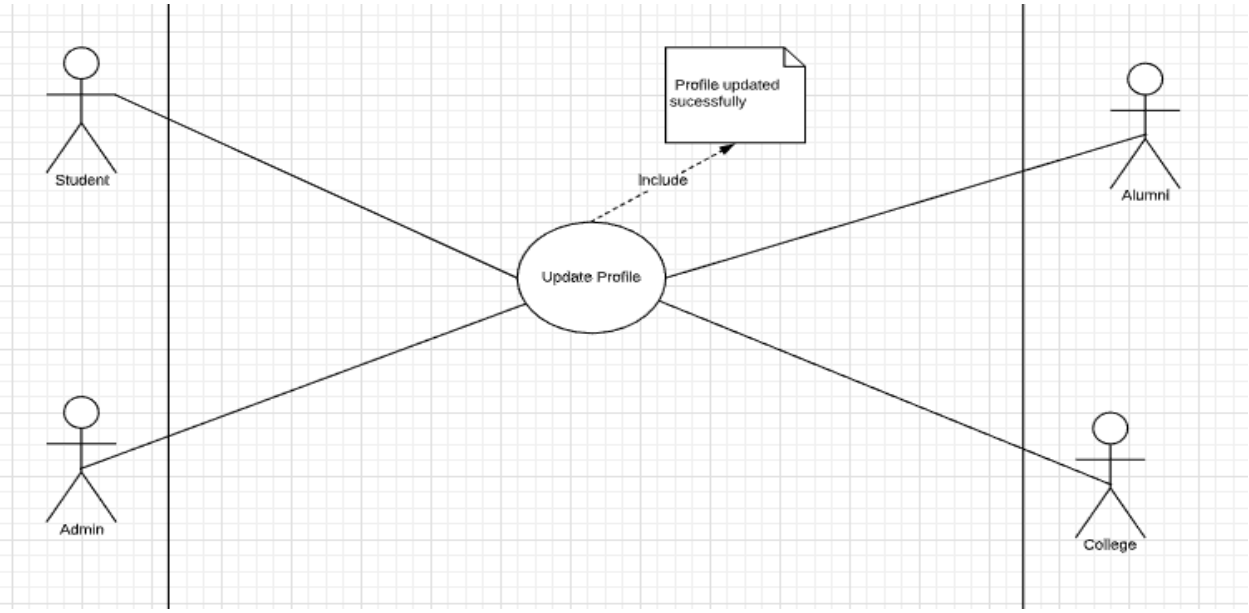


Figure 13 extended use case update profile.

Use case: Update Profile

Actor involved: Alumni, college, student, admin

Description: Every user will be able to update their own profile

Users Interaction	System response
1. User click on view profile option.	2. System shows profile of that particular user.
3. User click on update profile option.	4. System shows update profile page.
5. User edit the data and press update.	6. System updates the user data.

Table 3 Extended use case update profile.

Extended use case diagram for Add achievement:

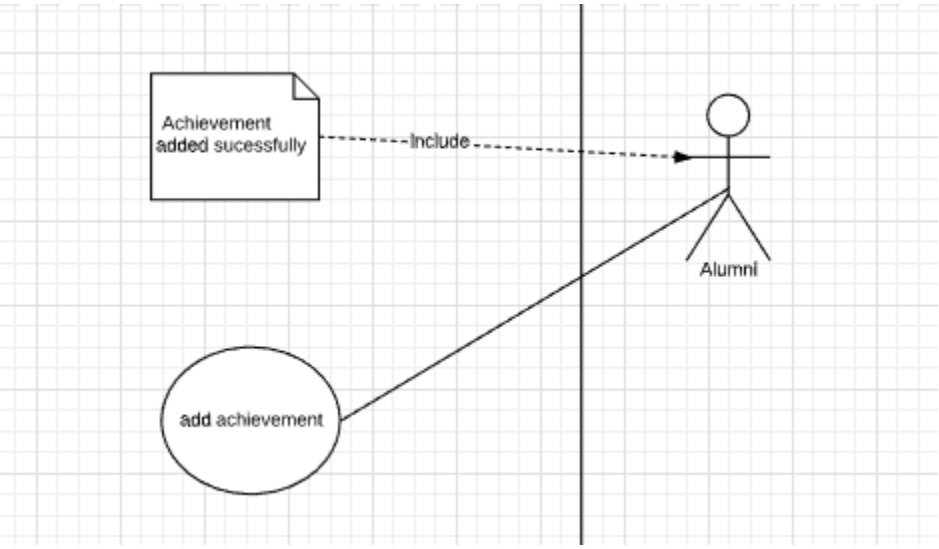


Figure 14 extended use case add achievement

Use case: Add achievement

Actor involved: Alumni

Description: Alumni can only add the achievements they have done but it can be viewed by all the users.

Users interaction	System response
1. User (Alumni) click on view profile option.	2. System shows profile of that alumni.
3. User click on add achievements.	4. System shows achievement page.
5. User add achievements and press submit.	6. System add the achievement to the profile.

Table 4 Extended use case add achievement.

Extended use case diagram for Events:

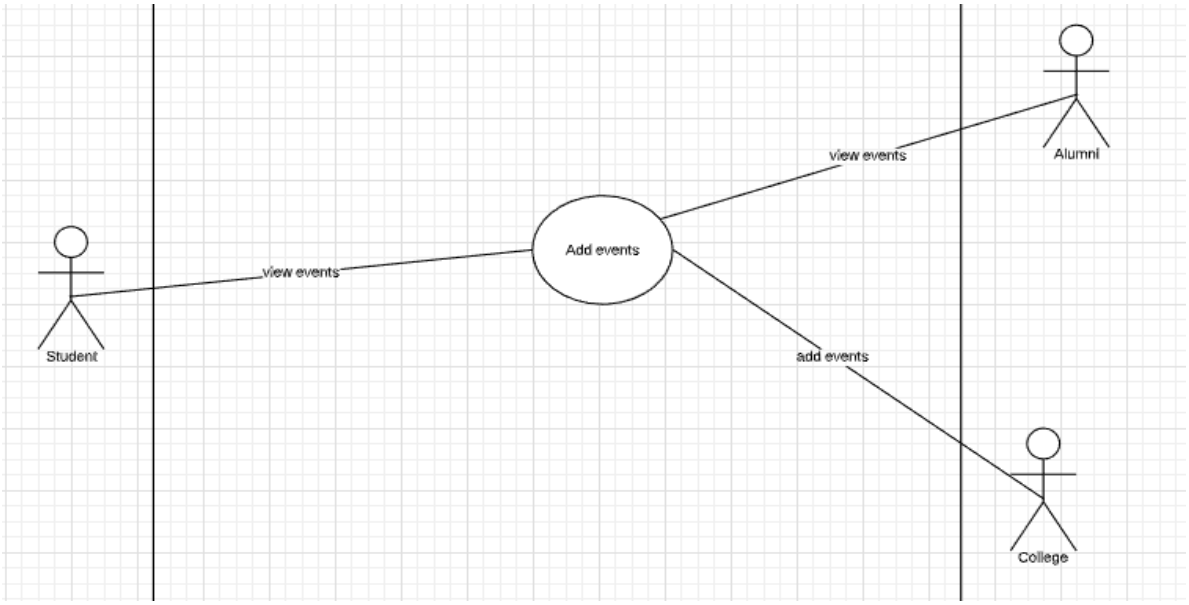


Figure 15 extended use case events

Use case: Events

Actor involved: College, Alumni, Students

Description: College can add new events and alumni can view the events.

Users Interaction	System response
1. User (College) open event tab from app.	2. System shows event page.
3. College adds events and press submit.	4. System gets the event and notify to alumni and students.
5. Alumni and student open event page.	6. System shows new events.

Table 5 Extended use case events.

Extended use case diagram for Search profile:

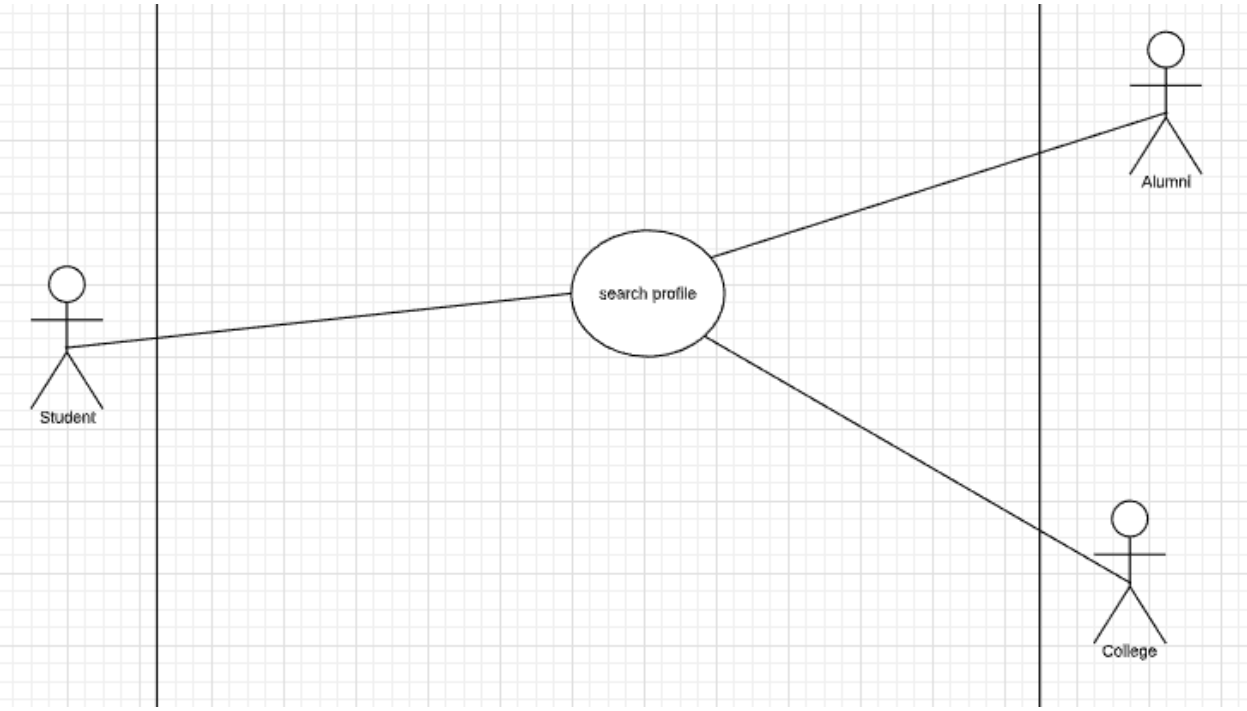


Figure 16 extended use case search profile.

Use case: Search Profile

Actor involved: College, students, Alumni

Description: All the users can view, and search profiles accept of admin. Admin can only view through web.

Users interaction	System response
1. User (College, student, alumni) clicks on search alumni.	2. System shows all the list of alumni.
3. User type alumni name or search and clicks on it.	4. System opens the respective alumni's profile page.

Table 6 Extended use case search profile.

Extended use case diagram for send message:

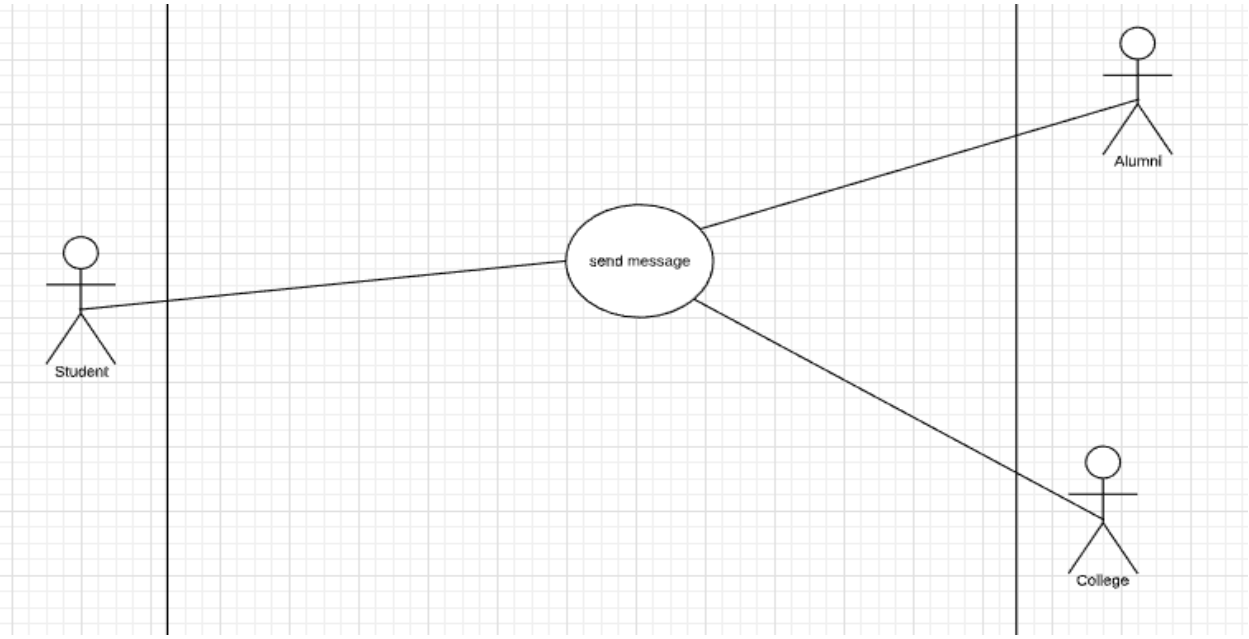


Figure 17 extended use case send message.

Use case: Send message

Actor involved: Alumni, college, student

Description: All users expect admin can send message to each other.

User's interaction	System response
1. User clicks on message tab.	2. System shows message page.
3. User select one person and send message.	4. System forward that message to appropriate user.

Table 7 Extended use case send message.

3.3.5: Entity Relationship Diagram

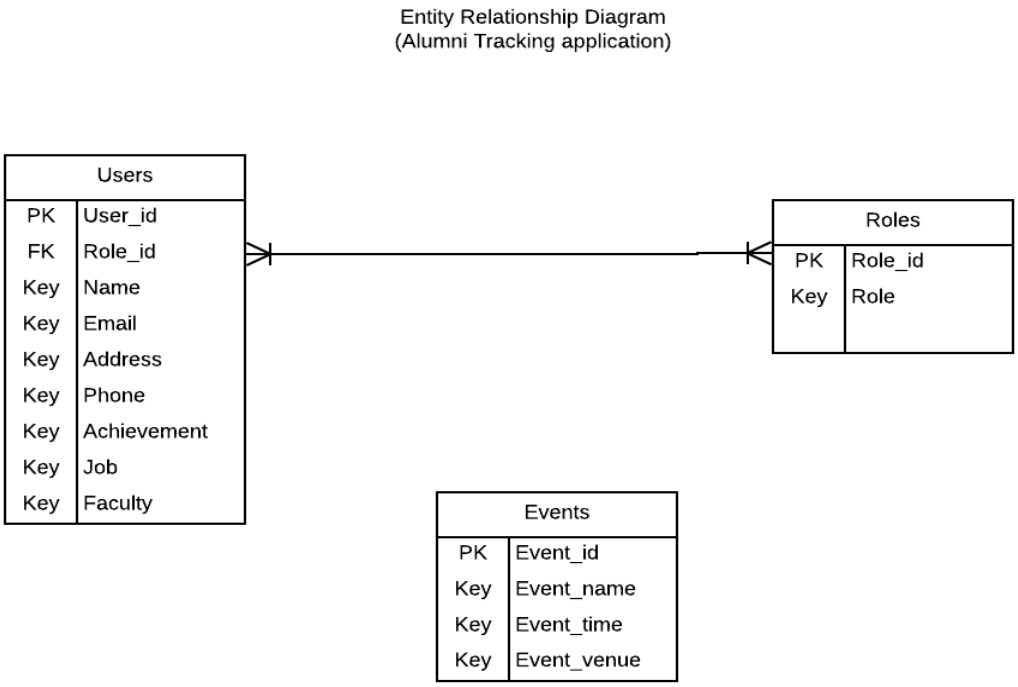


Figure 18 Entity relationship diagram

Description of Entity Relationship Diagram:

Here, after the normalization up to 2nf three tables are separated. Users table include all data of alumni, student, college and admin. Roles table includes the role and role id where alumni, student, college are separated with the code. A user can have multiple role because a student can become alumni in future in that case a user has multiple role, so role table is inherited with users table.

As we don't have any connection with event table because everybody gets event notifications and no necessary of having links between tables.

Research on Entity Relationship Diagram:

The entity relationship diagram is a semantic data modeling tool that is used to accomplish the goal of the making a structured data. Entity relationship diagram models data as entities and relationships, and entities have attributes. The thing for which we record the data are called entities. It can be a person, place, object or things. The name of the entity should be generic and well defined. (Bagui & Earp, 1964)

To model Erd we should first make a normalized database so that no any redundancy and anomalies will occur because whatever we have entities according to it the program will be operated.

3.4: Wireframes and Prototypes:

4.4.1: login wireframe: this is the wireframe of login page where two text boxes are placed for username and password of the user. There is also a button for login. Two links are attached in it for forgot password and signup.

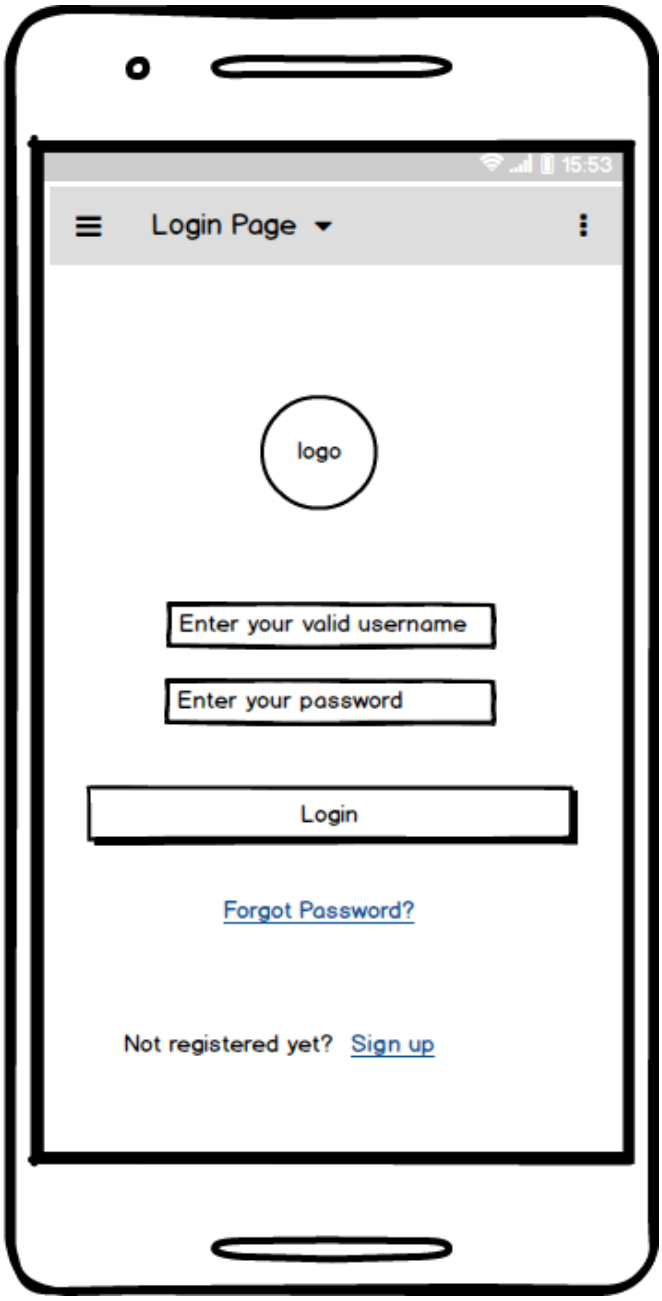


Figure 19 Login page wireframe.

4.4.2: Registration wireframe

This is the wireframe of registration page of the application. This includes all the users details some of the field are required where some are optional. There is also a button for register now when the users can submit their details.

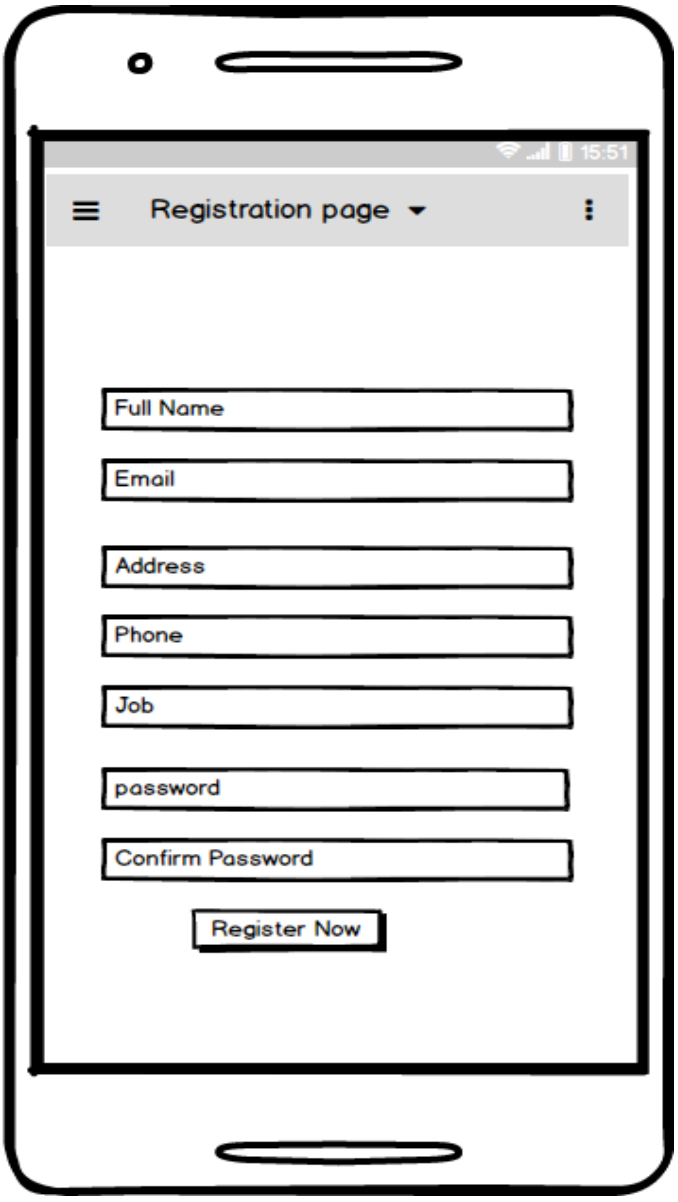


Figure 20 Registration page wireframe.

4.4.3: Profile wireframe:

This is the wireframe of profile page. Here the profile will be shown according to the user's detail. User can either be alumni student or college. The below boxes are for adding achievements. The first-round shaped field is for image user can put image of theirs in it.

Figure 21 Profile wireframe.

4.4.4: Home page:

This is the wireframe of homepage of all the users. In figure it is shown home page of alumni, but it be changed according to the user.

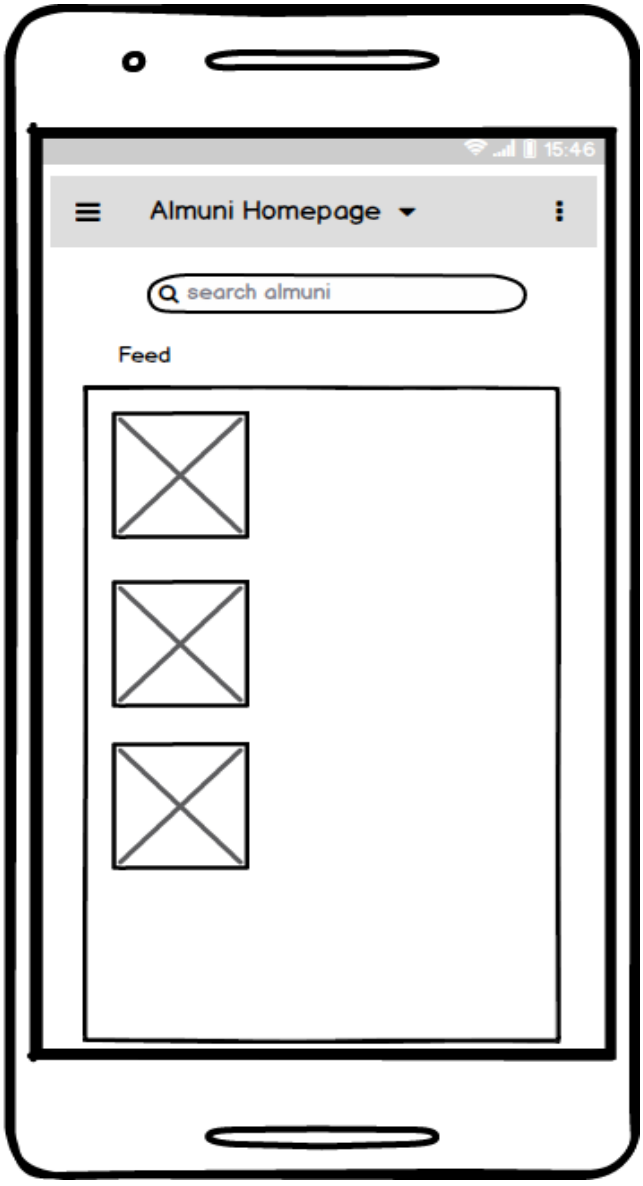


Figure 22 Homepage wireframe.

3.5: Development to date:

In first week of construction phase, I started working on mobile views like making User interface of application in flutter which is login page and registration page. It was just UI because I have not started backend development. After third week I started working on backend development using Laravel where I normalized data and started making user's table where all the data of users are stored. As I focused more on documentation part, I have not completed the user API till date. The user API is the main API of the application because all the user authentication, validation occurs in this part. I recently completed user's registration from mobile and web where a user can register themselves using mobile UI and their detail will be registered in the database. Here is the coding part that I have completed till date (Reference in Appendix-A for coding).

Mobile Application development:

Figure 23 Login page UI.

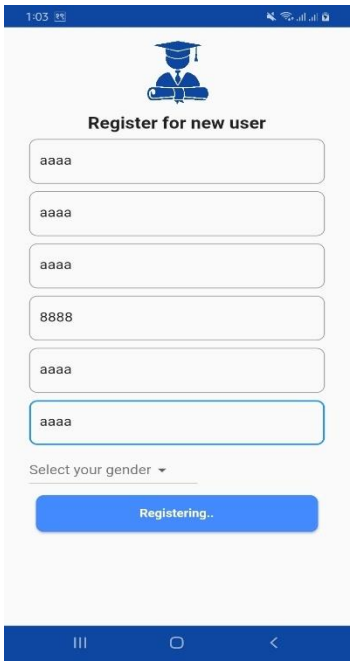


Figure 24 Registration UI

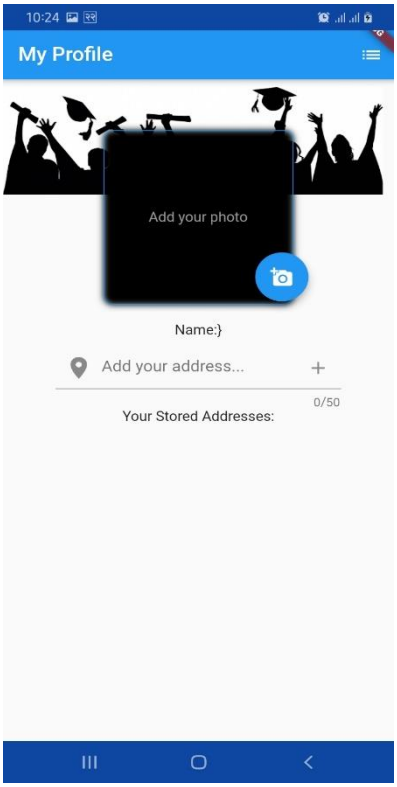


Figure 25 Profile UI.

Figure 26 Home page UI.

3.6: Analysis of progress:

(Note: Gantt Chart is in Appendix A section)

In this chapter I have shown the comparison between my actual work to date and Gantt chart.

As I have used Rational Unified Process as my software development methodology my contents in the Gantt chart divided into its four phases. Inception, elaboration, construction and transition phase. Starting from the inception phase I started it in 2019/8/20 to 2019/11/2. During this phase I collected all the features, resources and milestones. I have worked according to Gantt chart during inception phase.

According to Rational Unified Process principal we can change the requirement in future during development time, so I changed some of the features of the project like admin and its control.

Similarly, Elaboration phase is all about designing architecture, making various use case diagrams and revising risk management and requirement. I changed some of the features and development tools in this phase like I was planning to do my backend with google firebase. But according to my project scenario my supervisor suggested me to make application programming interface in web and consume it to mobile application rather then using firebase. Then I start working on backend development with Laravel. My application front is in flutter, but my backend is based on php Laravel. I also changed some of the features like student can be alumni in future. According to my Gantt chart my planning to complete elaboration phase was till 2020/03/10 and I am still working on it.

Likewise, another phase is construction phase where the maximum development part occurs. According to Gantt chart my construction starts from 2019/12/1 but I have started it bit earlier. Just after collecting requirement I started working on frontend views in mobile like login page and registration page. I am still working on construction phase.

3.7: Future work:

I am now working on application programming interface development of user. In my future work I will keep continuing this API because this is the main API where user's information will be stored. I will be working on user's login authentication where user will be verified before login into the application. After that I will be working on API development of events and roles.

After completing that my next step will be towards consuming those API's in mobile application. I will start consuming it in mobile application, showing user's detail according to the interface and features. After finishing that I will be working on testing and debugging phase which is also in construction phase of Gantt chart. Finally, after reviewing my application with my supervisors I will start working on my remaining documentation part like making sequence, collaboration diagram and after finalizing all the documents and development I will head towards transition phase and submit my project till 2020/04.

Chapter 4: References and Bibliography:

References

Bagui, S. & Earp, R. (1964) *Database Design Using Entity-Relationship Diagrams*. illustrated ed. London: Auerbach publications.

Bittner, K. & Spence, I. (2003) *Use Case Modeling*. Illustrated ed. New york: Addison-wesley.

Burke, W.W. & Noumair, D.A. (2015) *Organization Development: A Process of Learning and Changing*. 3rd ed. USA: FT Press, 2015.

Charvat, J. (2003) *Project Management Methodologies: Selecting, Implementing, and Supporting Methodologies and Processes for Projects*. illustrated ed. Canada: John Wiley & Sons, 2003.

Firebase. (2020) *Firebase* [Online]. Available from: https://firebase.google.com/ [Accessed 07 January 2020].

flutter. (2020) *flutter.dev* [Online]. Available from: https://flutter.dev/ [Accessed 07 January 2020].

Guru99. (2020) *What is Spiral Model? When to Use? Advantages & Disadvantages* [Online]. Available from: https://www.guru99.com/what-is-spiral-model-when-to-use-advantages-disadvantages.html.

IBM. (2003) Rational Unified Process. *Rational Unified Process: A Best Practicles Approach*, 1(2020/1/7), p.82.

IBM. (1998) *Rational Unified Process* [Online]. ibm Available at: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf [Accessed 07 January 2020].

Kilicdagi, A. & Yilmaz, i.H. (2014) UK: Packt Publishing.

Kumar, P. (2009) *Rational Unified Process(Rup)* [Online]. Available from: https://www.slideshare.net/pawanonline83/rational-unified-processrup [Accessed 07 January 2020].

laravel. (2011) *Introduction to laravel* [Online]. Available from: https://laravel.com/docs/4.2/introduction.

Morse, A.P. (2017) *Rational Unified Process: What Is It And How Do You Use It?* [Online]. Available from: https://airbrake.io/blog/sdlc/rational-unified-process [Accessed 14 March 2017].

salesforceforeducation. (2000) *Salesforce for Education Reviews & Product Details* [Online]. Available from: https://www.g2.com/products/salesforce-for-education/reviews [Accessed 06 January 2020].

Sinha, S. (2019) *Beginning Laravel: Build Websites with Laravel 5.8*. 2nd ed. India: Apress.

www.getalma.com. (2012) *Alma | Student Information* [Online]. Available from: https://sourceforge.net/software/product/Alma/ [Accessed 07 January 2020].

Bibliography:

Kumar, P. (2009) *Rational Unified Process(Rup)* [Online]. Available from: https://www.slideshare.net/pawanonline83/rational-unified-processrup [Accessed 07 January 2020].

laravel. (2011) *Introduction to laravel* [Online]. Available from: https://laravel.com/docs/4.2/introduction.

Morse, A.P. (2017) *Rational Unified Process: What Is It And How Do You Use It?* [Online]. Available from: https://airbrake.io/blog/sdlc/rational-unified-process [Accessed 14 March 2017].

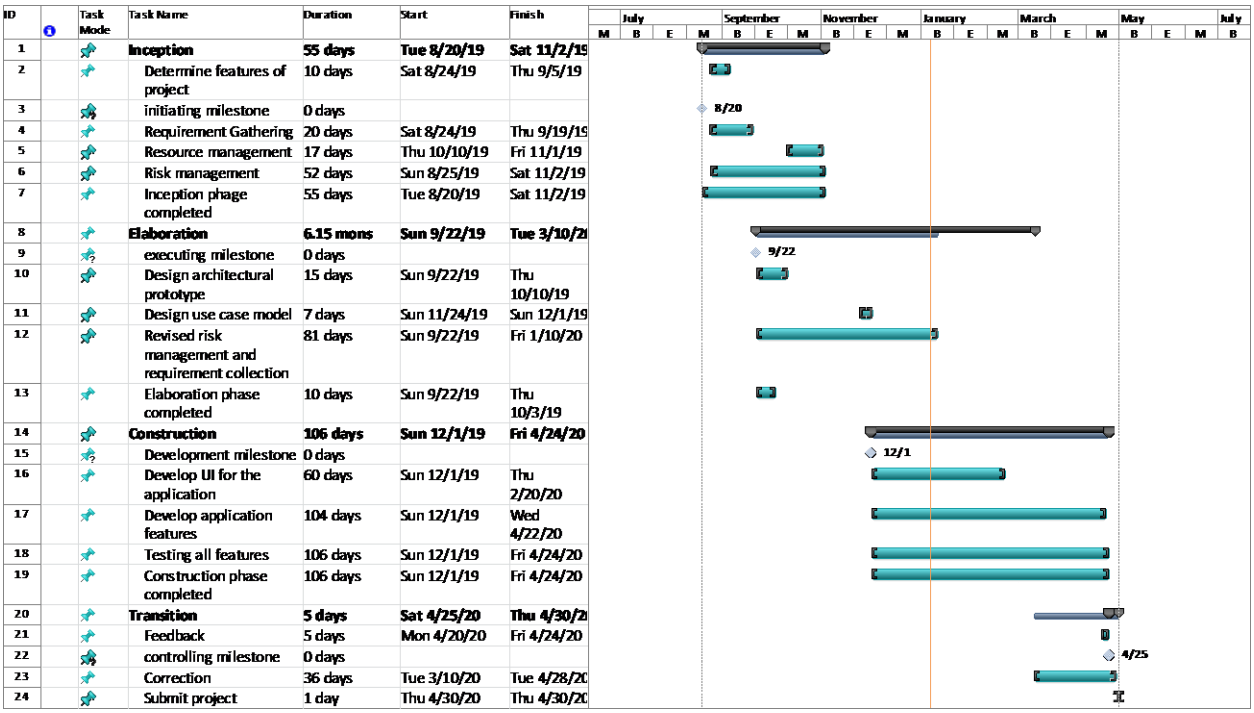
salesforceforeducation. (2000) *Salesforce for Education Reviews & Product Details* [Online]. Available from: https://www.g2.com/products/salesforce-for-education/reviews [Accessed 06 January 2020].

Sinha, S. (2019) *Beginning Laravel: Build Websites with Laravel 5.8*. 2nd ed. India: Apress.

www.getalma.com. (2012) *Alma | Student Information* [Online]. Available from: https://sourceforge.net/software/product/Alma/ [Accessed 07 January 2020].

Chapter 5: Appendix

Appendix A:



ID	Task Mode	Task Name	Duration	Start	Finish
1		**Inception**	**55 days**	**Tue 8/20/19**	**Sat 11/2/19**
2		Determine features of project	10 days	Sat 8/24/19	Thu 9/5/19
3		initiating milestone	0 days		
4		Requirement Gathering	20 days	Sat 8/24/19	Thu 9/19/19
5		Resource management	17 days	Thu 10/10/19	Fri 11/1/19
6		Risk management	52 days	Sun 8/25/19	Sat 11/2/19
7		Inception phage completed	55 days	Tue 8/20/19	Sat 11/2/19
8		**Elaboration**	**6.15 mons**	**Sun 9/22/19**	**Tue 3/10/2**
9		executing milestone	0 days		
10		Design architectural prototype	15 days	Sun 9/22/19	Thu 10/10/19
11		Design use case model	7 days	Sun 11/24/19	Sun 12/1/19
12		Revised risk management and requirement collection	81 days	Sun 9/22/19	Fri 1/10/20
13		Elaboration phase completed	10 days	Sun 9/22/19	Thu 10/3/19
14		**Construction**	**106 days**	**Sun 12/1/19**	**Fri 4/24/20**
15		Development milestone	0 days		
16		Develop UI for the application	60 days	Sun 12/1/19	Thu 2/20/20
17		Develop application features	104 days	Sun 12/1/19	Wed 4/22/20
18		Testing all features	106 days	Sun 12/1/19	Fri 4/24/20
19		Construction phase completed	106 days	Sun 12/1/19	Fri 4/24/20
20		**Transition**	**5 days**	**Sat 4/25/20**	**Thu 4/30/2**
21		Feedback	5 days	Mon 4/20/20	Fri 4/24/20
22		controlling milestone	0 days		
23		Correction	36 days	Tue 3/10/20	Tue 4/28/2
24		Submit project	1 day	Thu 4/30/20	Thu 4/30/2

Figure 27 Gantt chart.

Coding:

```
import 'package:alumniapp/post.dart';
import 'package:alumniapp/profile.dart';
import 'package:alumniapp/registrationpage.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:alumniapp/homepage.dart';
import 'dart:async';

void main() {
  runApp(Main());
  SystemChrome.setSystemUIOverlayStyle(SystemUiOverlayStyle(
    systemNavigationBarColor: Colors.blue[900], // navigation bar color
    statusBarColor: Colors.blue[900], // status bar color
  ));
```

```
}

class Main extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    print("hello");
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Chart',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: Scaffold(
        body: LoginPage(),
      ),
    );
  }
}

class LoginPage extends StatefulWidget {
  @override
  State createState() => new LoginPageState();
}

class LoginPageState extends State<LoginPage> {
  // Create a text controller and use it to retrieve the current value
  // of the TextField.
  final emailController = TextEditingController();
  final passwordController = TextEditingController();

  @override
  void dispose() {
    // Clean up the controller when the widget is disposed.
    emailController.dispose();
    passwordController.dispose();
    super.dispose();
  }

  final _formKey = GlobalKey<FormState>();
  var sizebox = SizedBox(
    height: 10,
  );

  Widget build(BuildContext context) {
    return Padding(
```

```
    padding: const EdgeInsets.all(8.0),
    child: Center(
      child: Form(
        key: _formKey,
        autovalidate: true,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            alumnilogo(context),
            TextFormField(
              controller: emailController,
              keyboardType: TextInputType.emailAddress,
              validator: validateEmail,
              decoration: InputDecoration(
                  prefixIcon: Icon(Icons.email),
                  labelText: "Email",
                  border: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(10.0))),
            ),
            sizebox,
            TextFormField(
              controller: passwordController,
              obscureText: true,
              validator: (value) {
                if (value.isEmpty) {
                  return 'Please enter your Password';
                } else {
                  return null;
                }
              },
              decoration: InputDecoration(
                  prefixIcon: Icon(Icons.lock),
                  labelText: "Password",
                  border: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(10.0))),
            ),
            sizebox,
            ButtonTheme(
              shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(10.0)),
              minWidth: MediaQuery.of(context).size.width - 10,
              height: 45,
              child: RaisedButton(
                color: Colors.blue[900],
                child: Text(
```

```
                "Login",
                style: TextStyle(
                    fontWeight: FontWeight.bold,
                    fontSize: 30,
                    color: Colors.white),
              ),
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                      builder: (context) => HomePage()),
                );
              },
            ),
          ),
          MaterialButton(
            splashColor: Colors.blue[900],
            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10.0)),
            onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                      builder: (context) =>Registrationpage()),
                );
            },
            child: Text(
              "Forgot Password?",
              style: TextStyle(fontSize: 20.0),
            ),
            minWidth: MediaQuery.of(context).size.width,
          ),
          Expanded(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.end,
              children: <Widget>[
                Row(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: <Widget>[
                    Text("Not registered Yet?"),
                    MaterialButton(
                      splashColor: Colors.blue[900],
                      shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(5.0)),
                      minWidth: 5,
```

```
                        height: 5,
                        onPressed: ()
                         {
                            Navigator.push(
                    context,
                    MaterialPageRoute(
                        builder: (context) =>HomePage()),
                  );
                         },
                        child: Text("Contact Us"),
                      ),
                    ],
                  ),
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }
}

String validateEmail(String value) {
  Pattern emailPattern =
      r'^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))@((\[[0-
9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-
Z]{2,}))$';
  RegExp emailRegExp = RegExp(emailPattern);

  if (!emailRegExp.hasMatch(value)) {
    return 'Please enter your valid Email';
  } else {
    return null;
  }
}

/*
 *
 */
Widget alumnilogo(context) {
  return Expanded(
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
```

```
      children: <Widget>[
        Expanded(
          child: Padding(
            padding: const EdgeInsets.symmetric(vertical: 20.0),
            child: Material(
              child: Image(
                image: AssetImage('images/alumnilogo.png'),
                color: Colors.blue[900],
              ),
            ),
          ),
        )
      ],
    ),
  );
}
```

Coding of registration page:

```
import 'dart:convert';

import 'package:alumniapp/api.dart';
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';
import 'package:alumniapp/main.dart';
import 'dart:async';
import 'package:http/http.dart' as http;

class Registrationpage extends StatefulWidget {
  @override
  State<StatefulWidget> createState() {
    return RegistrationpageState();
  }
}

class RegistrationpageState extends State<Registrationpage> {
  TextEditingController nameController = TextEditingController();
  TextEditingController emailController = TextEditingController();
  TextEditingController phoneController = TextEditingController();
  TextEditingController symbolController = TextEditingController();
  TextEditingController addressController = TextEditingController();
  TextEditingController passwordController = TextEditingController();
  TextEditingController passwordConfController = TextEditingController();
```

```
 bool _isLoading = false;

List<DropdownMenuItem<int>> listDrop = [];
int selected = null;

void loadData(){
  listDrop = [];
  listDrop.add(new DropdownMenuItem(
    child: new Text('Male'), value: 1,

  ),
  );
  listDrop.add(new DropdownMenuItem(
    child: new Text('Female'), value: 2,

  ));
}

 @override
 Widget build(BuildContext context) {
   loadData();
   return Scaffold(
     resizeToAvoidBottomInset: false,
     body: Form(
       child: Container(
         margin: EdgeInsets.only(top: 10),
         child: Padding(
           padding: const EdgeInsets.all(30.0),
           child: Column(
               crossAxisAlignment: CrossAxisAlignment.start,
               children: <Widget>[
                 Center(
                   child: Container(
                     child: Image(
                       height: 100,
                       width: 100,
                       image: AssetImage('images/alumnilogo.png'),
                       color: Colors.blue[900],
                     ),
                   ),

                 )
                 ,
                 Padding(
                   padding: const EdgeInsets.all(8.0),
```

```
                    child: Center(child: Text('Register for new user', style: Tex
tStyle(fontWeight: FontWeight.bold, fontSize: 20),)),
                  ),
                  TextFormField(
                    controller: nameController,
                    decoration: InputDecoration(
                      border: OutlineInputBorder(
                        borderRadius: BorderRadius.circular(10),
                      ),
                      hintText: "Full Name",

                    ),
                    // validator: (String value) {
                    //   if (value.trim().isEmpty) {
                    //     return 'Name cannot be empty*';
                    //   }
                    // },
                  ),
                  SizedBox(
                    height: 10,
                  ),
                  TextFormField(
                    controller: emailController,
                    decoration: InputDecoration(
                      border: OutlineInputBorder(
                        borderRadius: BorderRadius.circular(10),
                      ),
                      hintText: "Email*",
                    ),
                    // validator: (String value) {
                    //   if (value.trim().isEmpty) {
                    //     return 'Email address cannot be empty';
                    //   }
                    // },
                  ),
                  SizedBox(
                    height: 10,
                  ),
                  TextFormField(
                    decoration: InputDecoration(
                      border: OutlineInputBorder(
                        borderRadius: BorderRadius.circular(10),
                      ),
                      hintText: "Phone*",
                    ),
```

```
                  // validator: (String value) {
                  //   if (value.trim().isEmpty) {
                  //     return 'Phone cannot be empty';
                  //   }
                  // },
                ),
                SizedBox(
                  height: 10,
                ),
                TextFormField(
                  decoration: InputDecoration(
                    border: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(10),
                    ),
                    hintText: "College Symbol No.*",
                  ),
                  // validator: (String value) {
                  //   if (value.trim().isEmpty) {
                  //     return 'Symbol No cannot be empty';
                  //   }
                  // },
                ),
                SizedBox(
                  height: 10,
                ),
                TextFormField(
                  controller: passwordConfController,
                  decoration: InputDecoration(
                    border: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(10),
                    ),
                    hintText: "password confirmation",
                  ),
                  // validator: (String value) {
                  //   if (value.trim().isEmpty) {
                  //     return 'Address cannot be empty';
                  //   }
                  // },
                ),
                SizedBox(
                  height: 10,
                ),
                TextFormField(
                  controller: passwordController,
                  decoration: InputDecoration(
```

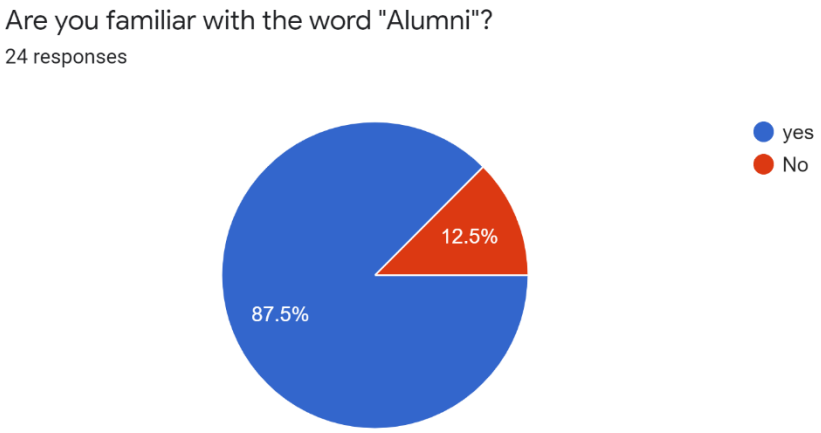
```
                    border: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(10),
                    ),
                    hintText: "Password*",
                  ),
                  // validator: (String value) {
                  //   if (value.trim().isEmpty) {
                  //     return 'Education cannot be empty';
                  //   }
                  // },
                ),
                SizedBox(
                  height: 10,
                ),
                Container(
                  height: 50,
                  width: 200,
                  child: DropdownButton(
                    value: selected,
                    items: listDrop ,
                    hint: Text('Select your gender'),
                    onChanged: (value){
                      selected = value;
                      setState(() {

                      });
                    },
                  ),

                ),


                Padding(
                  padding: const EdgeInsets.all(8.0),
                  child: Center(
                    child: Container(
                      width: 430,
                      height: 50,
                      child: RaisedButton(
                        shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(10)),
                        onPressed: () {
                          _handleSubmit();
                                                },
                                                color: Colors.blueAccent,
```

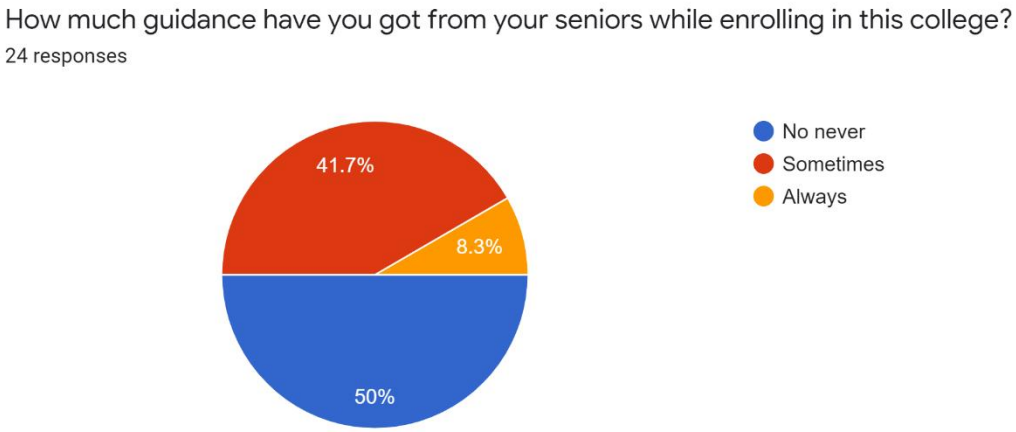
```
                                                child: Text(
                                                 _isLoading ? 'Registering.
.' : "Submit",
                                                  style: TextStyle(
                                                    fontWeight: FontWeight.
bold,
                                                    color: Colors.white,
                                                  ),
                                                ),
                                              ),
                                            ),
                                          ),
                                        ),

                                        /*
                                     * For password textfield
                                     */
                                          ]),
                                      ),
                                    ),
                                  ),
                                );
                              }

                              void _handleSubmit() async{
                                setState(() {
                                  _isLoading = true;
                                });



                                var data = {
                                  'name' : nameController.text,
                                  'email' : emailController.text,
                                  'password' : passwordController.text,
                                  'password_confirmation' : passwordConfControlle
r.text,

                                };

                                print(data);

                                var res = await CallApi().postData(data, 'signup'
);
```

```
                                var body = jsonDecode(res.body);
                                print(body);
```

Appendix B:

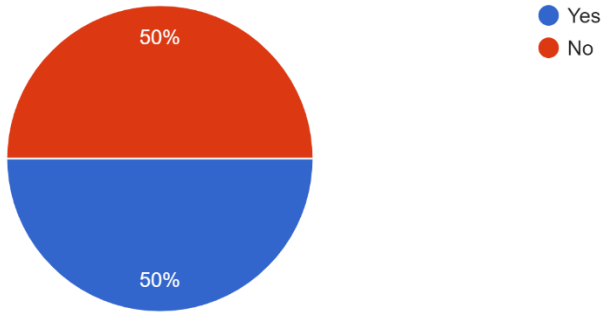
Survey for alumni tracking application:





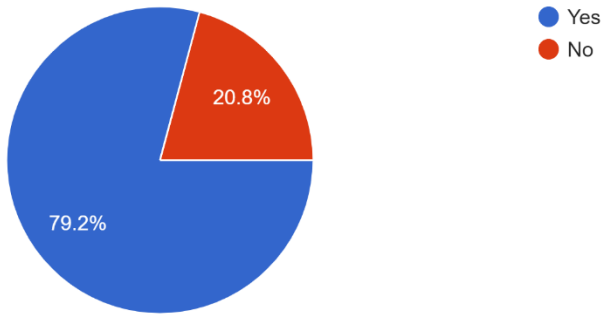
Have you expected a mobile application from where you can track your Alumni (Seniors) so that you can be in touch and watch their progress in life?

24 responses



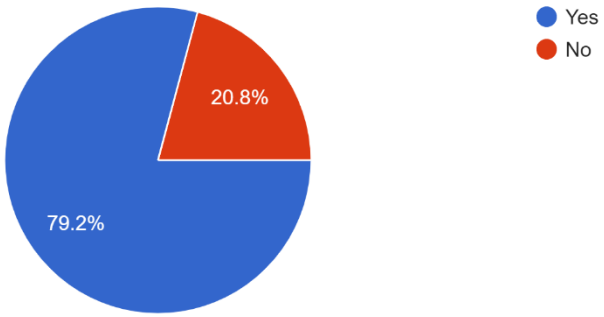
Do you think your university should be in touch with you after your graduation too?

24 responses



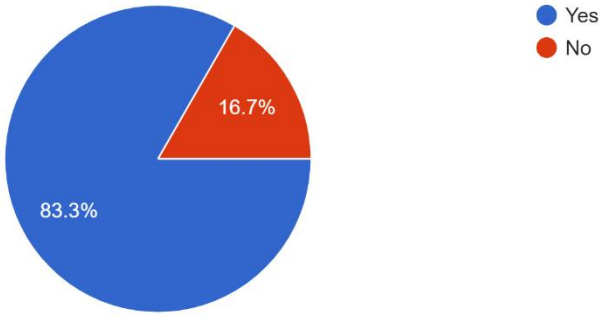
Are you facing or ever faced problems while doing college projects due to improper guidance of teachers?

24 responses



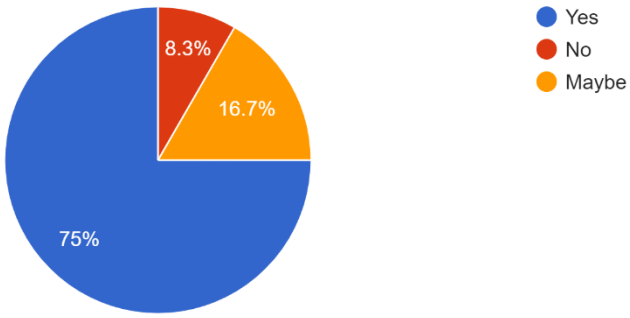
Do you think this application will be useful?

24 responses



Do you think you should be in touch with the college Alumni so that you can get help from them whenever you feel confused regarding the college policies and projects?

24 responses



End of the project….