



ASHESI UNIVERSITY COLLEGE

**THE FORMATION AND EVALUATION OF TEAMS IN INSTITUTIONS OF
EDUCATION**

APPLIED PROJECT

B.Sc. Computer Science

Festus Emmanuel Jartu

2016

ASHESI UNIVERSITY COLLEGE

The Formation and Evaluation of Teams in Institutions of Education

Applied Project

Applied Project submitted to the Department of Computer Science, Ashesi University College

in partial fulfilment of the requirements for the award of Bachelor of Science

Degree in Computer Science

Festus Emmanuel Jartu

April 2016

DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

.....

Candidate's Name:

.....

Date:

.....

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied project laid down by Ashesi University College.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date:

.....

ACKNOWLEDGEMENT

I would like to express my profound gratitude to my project supervisor, Ms. Grace

Sallar for her dedication and guidance through the successful completion of the project.

I would also like to appreciate my mentor Frank Asamoah and my mum Vida Lansana for their constant support, motivation and encouragement throughout my university program.

A special appreciation goes to my consultant and advisor, Dr. Nathan Amanquah for his support and inspiration throughout the project.

The completion of the project was especially made possible based on knowledge acquired over the years. Hence a special appreciation goes to all my lecturers at Ashesi University College, my friends and colleagues. To all who supported me directly or indirectly through the completion of the project, I say thank you.

ABSTRACT

The introduction of technology in any aspect of an institution's operation improves efficiency and limits the misuse of resources as technology has proven to be one of the most efficient ways to execute tasks. The use of technology in institutions of education for team formation has greatly impact team formation techniques for learning.

Team formation can be a challenging task for both students and lecturers. It becomes even more tedious when there is a need to keep track of history of who have previously worked together in a team or when a variety of criteria must be considered in the team formation process.

To achieve the formation of teams based on the different criteria, research and discussions had to be conducted around the topic for a dipper understanding of the concepts. The different criteria were chosen based on the manner in which groups are formed in various institutions of education and the ones that yield better results were outlined to be employed in the project.

Furthermore, the ability to evaluate or assign grades to teams and team members can be time consuming and inefficient if done on paper or in cases of multiple evaluators. Therefore, the essence of this project is to develop a smart team maker application. This application puts into account all these factors to change the concept and experience of team formation and evaluation in institutions of education

ACRONYMS

PHP - Hypertext Preprocessor

HTML - Hypertext Markup Language

CSS – Cascading Style Sheet

UI –User Interface

AJAX – Asynchronous JavaScript and XML

API – Application Programming Interface

HTTP – Hypertext Transfer Protocol

ER – Entity Relationship

SQL – Structured Query language

JSON – Java Script Object Notation

JS – Java Script

Contents

Chapter 1: Introduction	1
1.1 Problem Description & Motivation.....	1
1.2 Related Work & Existing Solution	2
1.3 Aims & Objectives.....	4
1.4 Overview of Report.....	5
Chapter 2: Requirement Specification	7
2.1 Product Overview	8
2.2 Product Functions.....	9
2.3 User Classes and Characteristics.....	9
2.4 Operating Environment	9
2.5 Functional Requirement	9
2.5.1 User Type: Supervisor.....	10
2.5.2 User Type: Student	11
2.6 Non-Functional Requirements.....	12
2.6.1 Product Requirements.....	12
2.6.2 Reliability Requirement	12
2.6.3 Organizational Requirements	13
Chapter 3: Design & Architecture.....	15

3.1	Overview	15
3.2	Architectural Pattern	15
3.3	Design & Platforms	17
3.4	Interface Design	17
3.5	Database Design	20
3.6	Data Design	20
3.7	System Models.....	23
3.8	Activity Diagram.....	26
Chapter 4: Implementation		28
4.1	Overview	28
4.2	System Organization	28
4.3	System Description	28
4.4	Major Tasks	28
Chapter 5: Implementation Support		30
5.1	Hardware, Software, Facilities, and Materials.....	30
5.2	Personnel	30
5.3	Performance Monitoring	30
5.4	Tools and libraries.....	30
Chapter 6: Testing & Results		34
6.1	Development Testing.....	34

6.1.1 Unit Testing.....	34
6.1.2 Component testing	36
6.1.3 System Testing.....	37
6.2 User Testing	37
Chapter 7: Conclusion & Recommendation	39
7.1 Conclusion.....	39
7.2 Limitation.....	39
7.3 Recommendation	40

Chapter 1: Introduction

1.1 Problem Description & Motivation

The process of forming teams for the purpose of project execution is one that comes with lot of hurdles especially in an academic institution. Educational institutions require a lot of team work to help teach and prepare students for real life situations, as such the essence of working with teams cannot be overemphasized (Coacher, 2011). Lots of argument and debates have risen among academicians on the best ways to form teams that generate minimum issues and yield maximize output (Barbara Oakley & Imad, 2004). However, a defined manner of doing so is yet to be agreed upon. Team work can have negative impact on the progress of a team if there is bad structure and management of teams. Inappropriate grouping to form teams leads to a number of issues such as inability to meet deadlines as a result of different schedules of team members, poor contribution of some members due to extreme unbalanced skill set, regular conflict and little output due to little contribution from some members of the team (Queensland, 2016).

In addition to grouping, it is sometimes difficult to quantify and qualify the amount of work done by each member of the team and their level of dedication to the project. Over the years, it has been realized that when there is no structured approach of evaluating individual contribution to the project work, some team members shy away from work (Konak, Konak, & Esparangoza, 2014). Also it is sometimes expensive and non-efficient to use manual approaches in doing team evaluation.

The motivation for this project stems from the challenges students experience when working together in different teams. Horror stories have been told of how the performance of students has been greatly affected by group work to the extent that over 60% of students in academic institutions rather work alone than in teams (Gallies, 2003).

Groups can however be formed to be more efficient and deliver better outputs. Depending on the criteria used, student can love working in groups and enrich themselves academically while benefiting from the social aspects (Chiriac, 2014). In addition, a paperless evaluation approach for both students and lecturers is a major goal.

1.2 Related Work & Existing Solution

Effective team work is a critical component of engineering education. This is because the search for efficient ways of forming team is still in motion. Since academic institutions depend largely on team work to develop students mentally and socially, it is necessary to have measures in place that makes the process and activity both effective and implementable (Potosky & Duke, 2007). It is a tradition among academicians to ascribe high value and credit to individual work. This has made people more independent and competitive. However, the benefits of collaborative learning have become outstanding in the field of education. Team work allows the individual team members to explore and develop their skills through others. It also fosters relationship building and encourages communication. In addition, team work helps students achieve a wide scope of knowledge, learn at a deeper level and retain information longer (Barbara Oakley & Imad, 2004). The process of team formation is one that has gone through series of trends over time. There has been a transition from the traditional approach of forming teams to a gradual introduction of technology in the process.

One of the major ways to form teams is the traditional team formation system. This is divided into self-select and randomly assigned approach. Self-select is where the students choose members of the team according to the required number. This method has proven to be inefficient as it pressurizes friends to try to choose each other to be in the same team or students with greater academic strength most likely reach out to others of their kind. This method might lead to sets of strong and weak teams in that regard (Queensland, 2016). The familiarity among group members could be an advantage or a disadvantage to the outcome of the project. This is because one may expect that due to the relationship already in existence, the project execution process would be easier. However, research has shown that this rather makes it hard sometimes to maintain discipline and order and it creates a negative attitude towards the assignment leading to excess homogeneity (Konnikova, 2010). The randomly assigned approach on the other hand is where students are randomly assigned into teams mostly by instructors. This works to a large extent but when put into keen consideration, it is thought to cause lot of issues, affecting the output of the project due to scheduling conflicts of team members (Hansen, 2006). Both approaches make use of paper evaluation to measure individual member contribution to the project which is also less efficient.

The use of some sort of technology has also been put in place to help enhance the efficiency of groups like the use of Microsoft Excel for team formation. Although teams can be formed in a number of different ways on this platform, it is limited to form teams only based on quantitative data. Microsoft Excel cannot also be used to do individual team member evaluations, which is an integral part of team work.

Applications have also been developed to help build better teams and do evaluation. Most of the applications are algorithm specific, that is, they are used to form teams based on a

particular criteria. For example, an application that form teams only based on GPA, or only based on gender, etc. Most team-forming applications also overlook the union between forming groups and evaluation since they are more concerned with team formation and pay less attention to evaluation. One example of applications that have been created to help form teams is the CATME tool (Loughry & Ohland, 2014). Although it combines team formation and evaluation, it still requires a lot of improvement on the Graphical User Interface and requires other essential features like team reshuffling, supervisor's comment, presentation evaluation etc. (Simmons, 2015). The Team Maker software is also a tool for forming random groups. Since it is specific to a particular criteria (random formation) and does not cater for evaluation, a better application could be developed to outmatch its use (O'Connel, 2010). A third example of existing applications is the *Grouper*, an application built to group specifically based on the schedules of members. It is also limited to just one method of grouping and does not include evaluation.

1.3 Aims & Objectives

The project develops a multi-platform application christened “Smart Team Maker”. The application runs on web and mobile platforms. It serves as a tool to help instructors and students have a better team work experience. It helps instructors comfortably create efficient, diverse and customized teams. This project is intended to develop multiple criteria on which efficient teams can be formed. The design supports the acquisition of data about individuals.

From the data provided, it devices a way to group certain individuals together to form a team. It goes beyond forming teams by also providing a platform where instructors can view the ratings and quantify how much effort is put in by each team member. The application has separate views for students and supervisor hence different users have access to different types

of information depending on the permission. The main objective of this application is to achieve the following:

- Make the process of team formation easy and efficient.
- Provide a platform where team members can rate each other's individual contribution to project work.
- Run the application on both web and mobile platforms to ensure availability and ease of work.
- Have an appealing and user friendly interface which can be easily used.
- Replace the paper evaluation approach in most institutions and provide a more efficient way to do peer evaluation.

1.4 Overview of Report

This is a representation of the structure of the report to enable the user to properly utilize this application. The first chapter introduces the reader to the essence and relevance of the report. It highlights some of the background information around the topic, points out the motivation for embarking on the project and also highlights some solutions presently used to handle the situation. It also possesses information on the project and maps out some of the aims and objectives of the author as he executes the project.

In Chapter 2, the author gives detailed information on the requirement specification. The author goes through a series of requirement gathering process and conversations with the project stakeholders before arriving at the functional and non-functional requirement. This is enabled by assessing the viability and implementation of the project.

Chapter 3 gives a detailed description of the project. This is where the real elements of the project are described. It includes user classes, the team formation criteria that are deployed and even the documentations to make future updates easy.

Chapter 4 covers the design and implementation of the software. This provides detailed information on the database design and the graphical user interfaces. It also looks at the evolution of these designs over the course of development. Explanation is given concerning the various interfaces, the interaction between the interfaces, and the database. It also defines the various models of the project. Finally, the author describes and explains the hardware and software interfaces of the system.

Chapter 5 covers the implementation and testing of the application. The author describes the different tools and approaches used in the implementation process. The different testing techniques used to verify the system are presented. In addition, challenges and setbacks during the implementation process are discussed in this section of the report.

Finally, the concluding chapter (Chapter 6) of the report talks about the overall work that has been done coupled with some recommendations for future work. It also gives a span of extension to be able to conveniently implement the project for future work.

Chapter 2: Requirement Specification

To be able to efficiently develop and execute the project, different requirements had to be understood and gathered. To gather the requirements, stake holders, libraries and other sources of project information were identified. In addition, strategies and means of information gathering were mapped out. The strategies used for requirement gathering stems from interviews, questionnaires, observations, and surveys amongst many others.

Table 2.1 Requirement Plan Table

Resource	Information	Method
Students	What are the basic functionalities they require the software to perform? What it required to make the formed teams works and make peer evaluation efficient	Questionnaires, emails, open discussions, social platform
Faculty	What do they look out for in the software? How can the system support them to transition from the traditional team formation approach?	Questionnaires, open –ended interviews, one on one discussion, Observation
Administrators and maintenance staff	What features should the software provide that makes maintenance easy? How to ensure continuous use of the system and service availability	Conduct a focus study, prototype trials, look at sample applications.
Supervisor	What areas of the development better suit your expertise, what advice can be given on acquisition of other skills needed for the development? How can this project expand your	Regular meetings, discussions and research.

Educational analyst	What professional approaches and inclusions do we need to bridge the gap between the customers and the developers	Talk to support teams
---------------------	---	-----------------------

In order to arrive at functional and non-functional requirements, the user requirement was thoroughly studied and understood.

2.1 Product Overview

The smart team maker is available to help make teams and do evaluations. It is expected to form groups based on a particular criteria chosen by the supervisor and then evaluations can be done accordingly. Figure 2.1 shows the main actors and systems and how they relate.

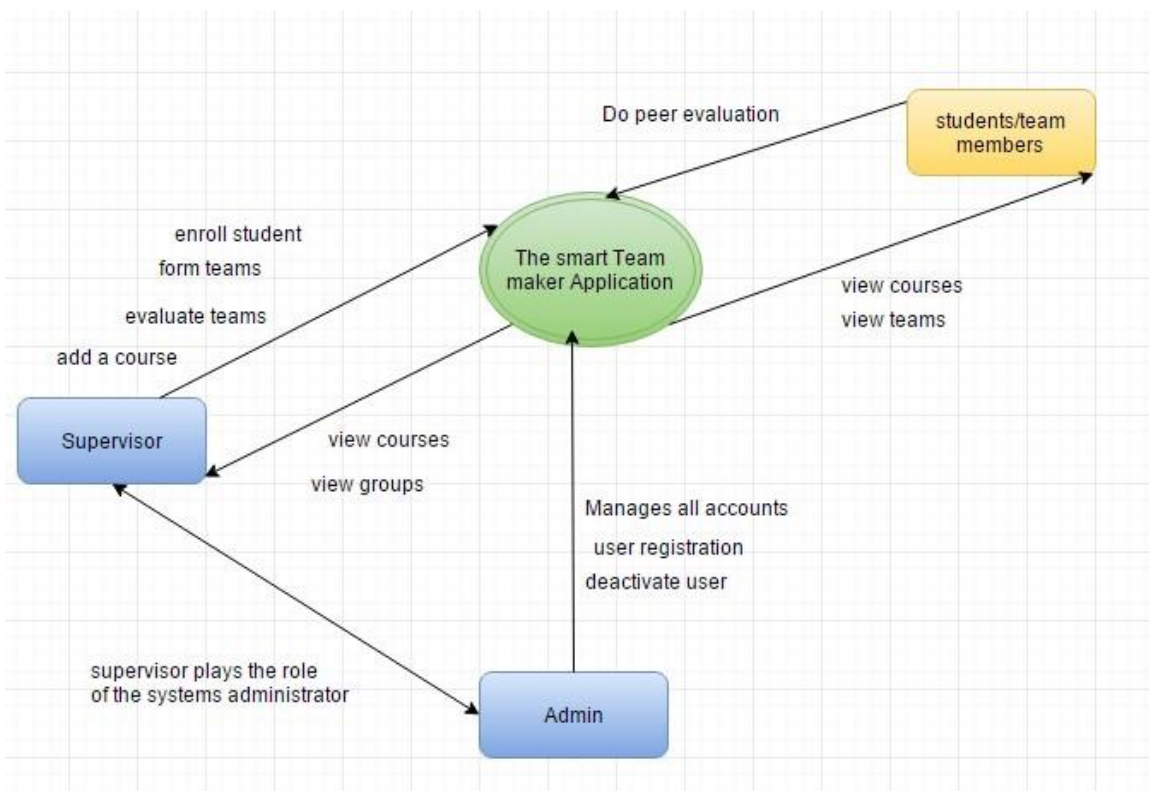


Figure 2.1 Context Diagram

2.2 Product Functions

The product enables the user to create and own an account. Once the account has been created, the user can then explore the various functionality as desired. The application can be used on the mobile and web platform thereby giving room for wide range of users.

2.3 User Classes and Characteristics

The main users of the system are faculty/supervisors and students. Supervisor performs dual tasks; plays the role of the system's administrator and course supervisor. Tasks performed as admin are to add and manage all accounts in the system. Task performed as supervisor are post and regulate projects, enroll students, form teams, evaluate team performance and project output. The second user class is the Student and the roles include executing a project, evaluating team members, viewing teams, among others.

2.4 Operating Environment

The smart team maker runs on two platforms; mobile and web. On the mobile aspect, it is incorporated on the three main existing platforms which is the IOS, Android and Windows. It supports browsers like Internet explorer, Google Chrome, Opera mini and Firefox. The application makes use of Cascading Style Sheet (CSS), Hypertext Mark-up Language (HTML), Java Script and Hypertext Preprocessor (PHP). The multi-platform of mobile and web foster the ease of use.

2.5 Functional Requirement

In order to properly understand the functional requirements, a pictorial view was created using UML class diagrams.

2.5.1 User Type: Supervisor

Figure 2.2 gives a use case for the supervisor and below is a list of functionalities the supervisor can perform.

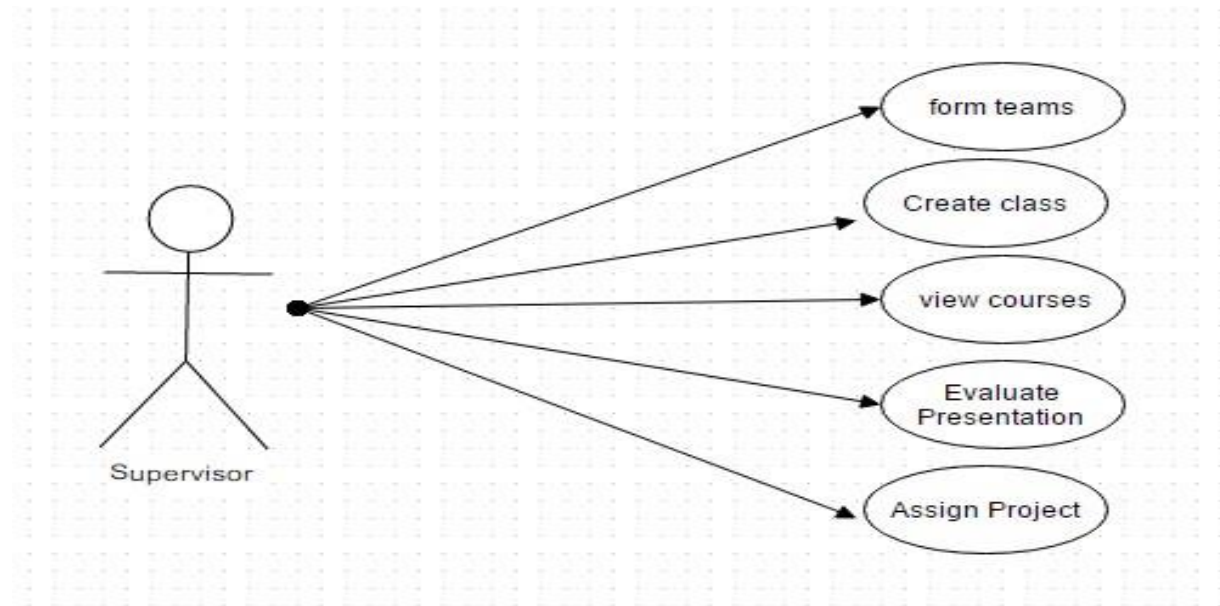


Figure 2.2 Supervisor Use Case

The main functional requirements for the supervisor are as follows:

- To form teams of students

Upon successful login the supervisor can access all enrolled students through the different courses and can form groups according to preference.

- To add course to the application

Upon successful login the user should be able to add a course by specifying the required features of that course through the provided fields.

- To view courses in the system

Once courses are successfully added the supervisor can view all courses available and can take further actions from there.

- To evaluate presentations of teams

Once a presentation is made, the user can make evaluations based on team presentation.

- To Assign project to teams

When a team is formed the user can access all teams and post projects.

2.5.2 User Type: Student

Figure 2.2 shows the use case diagram for the student and a list of actions the student can take.

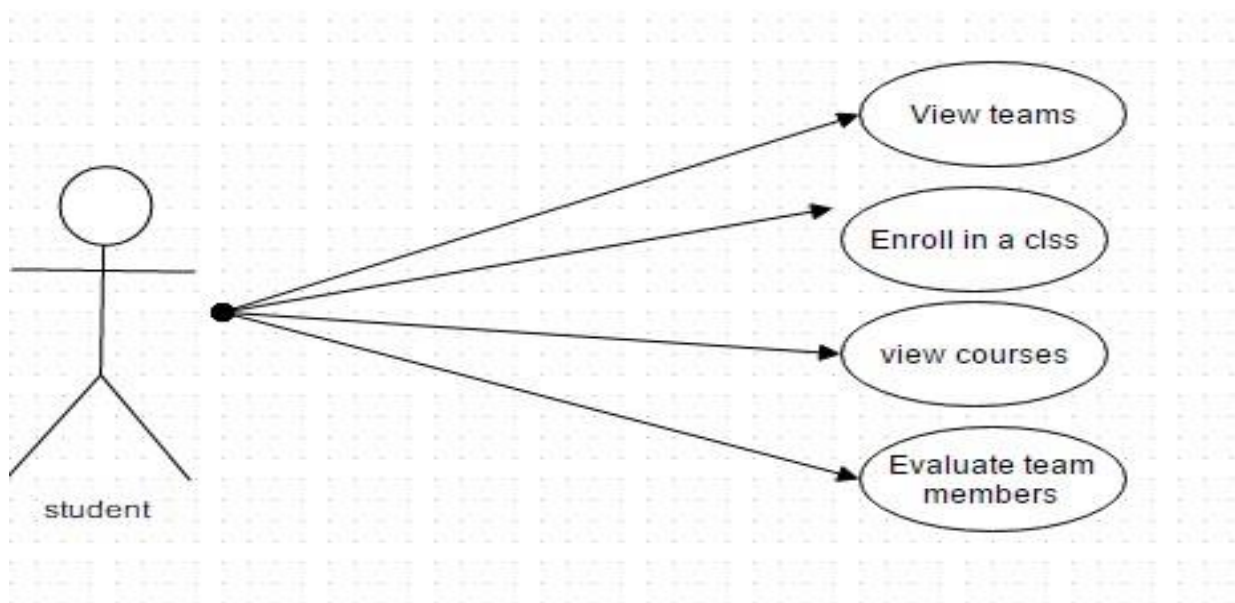


Figure 2.2 student Use Case

The main functional requirement for the student are as follows:

- To view teams already formed

When teams are formed, a student is assigned to a group. Once a student completes registration, access is granted to the teams

- To view courses in the system

The student should have access to all courses where there is active enrollment.

- To evaluate team members after project execution

Upon completion of project execution student, team members should be able to evaluate each other's contribution to project work.

2.6 Non-Functional Requirements

2.6.1 Product Requirements

This is to give a better understanding of the smart team maker application. It is divided into reliability, security, and safety of the application.

2.6.2 Reliability Requirement

The system gives notification to each group member once they have been assigned a group. The system is available at time of use, and malfunction of one component does not lead to a crash of the entire application. To avoid denial of service, the system is properly developed to ensure availability and recovery when under attack.

- **Security Requirement**

The system keeps team member details as secure as possible. Information retrieved from members are only be used for the purpose of assigning teams and kept confidential. Also, the system operates an identification requirement to verify the authenticity of the users and also have measure for protecting itself from internal and external threat.

- **Safety Requirement and Performance Requirement**

It is almost impossible to create an error free application, however the application is tested enough and systems put in place to capture unexpected errors that may arise during use. The system is expected to perform well under various circumstances and to report any performance constraints during execution. The system employs a well

detailed convention style that the user must adhere to in order to efficiently use the system. The validity of all information fed in the system are checked to ensure safe execution and manipulation of entities.

2.6.3 Organizational Requirements

□ Operational Requirements

The application is open and available to all users of the system directly and indirectly. The software can be used via the mobile and the web platform, making it highly accessible. Supervisors, students, admin or any other account holder can utilize the functionalities of the system. The application is not restricted based on operating system or number of users in operation.

□ Ethical Requirement

The smart team maker is a reservoir of information because anytime a user is registered, personal information about that user is stored. Hence all with access to these data must maintain high confidentiality and integrity. Also the application is assured to produce high quality service in process and product.

□ Product Regulatory Requirement

The system is accessed by multiple users, however, special parts of the system can be accessed based on permission. Some users have certain privileges to change and alter data accordingly like the supervisor. The use of views and login permissions is incorporated into the app to enforce system regulation. The system includes other features to help ensure the quality of service process and final application. The features are given below:

Correctness: The smart team maker application behaves and perform as it is stated in the functional requirements.

Robustness: The team maker application acts accordingly when wrong data or commands are issued.

Maintainability: The system is modified and improved as and when the need arises

Chapter 3: Design & Architecture

3.1 Overview

This document gives an insight into the project and the development process of the application. It describes the structure, design and architecture of the first version of the application, hence it is subject to review and change as required. In addition to design and structural view, it includes the system interface and user interfaces, data design component design, database design and the different modules of the system. It briefly discusses the software and hardware interfaces of the system and their interaction with the database system.

In order to effectively execute this project, decisions had to be made concerning what development process to engage with. Due to the nature of project and all it entails, it is beneficial to combine incremental development and some sort of plan driven approach. Plan driven because the expected output is known and all processes work towards attaining that goal. Incremental because versions of the final product have to be developed and deployed so the user can give constructive feedback to satisfy requirement.

The rest of the document contains a detailed explanation of design decisions made, for example, libraries, platforms, and frameworks, design patterns, etc. It includes UML diagrams, mind maps and design modules. It also defines the design pattern considered and reasons for consideration

3.2 Architectural Pattern

The success of any application or system depends on the architecture pattern used. The architecture encompasses a set of significant decisions about the organization of a software system. By describing the general characteristics of the architecture, these patterns not only guide the designers and developer of the application, but also determine how various

components of the system interact with each other. The smart team maker employed the use of MVC and repository architecture.

The repository allows for multiple users to modify the system since it runs a centralized data system. The MVC also helps in the separation of components into model, view and the controller for easier manipulation and management of the system. One of the benefits of MVC is that it allows the user to make changes to one end of the code without affecting the other. This feature makes testing and debugging much easier.

Figure 3.1 shows how the MVC architecture interweave with the application to form an interaction. Student/faculty makes changes to model (database) by sending commands to the controller (PHP) through the view (HTML, CSS, JS) and the resulting display is shown back on the view.

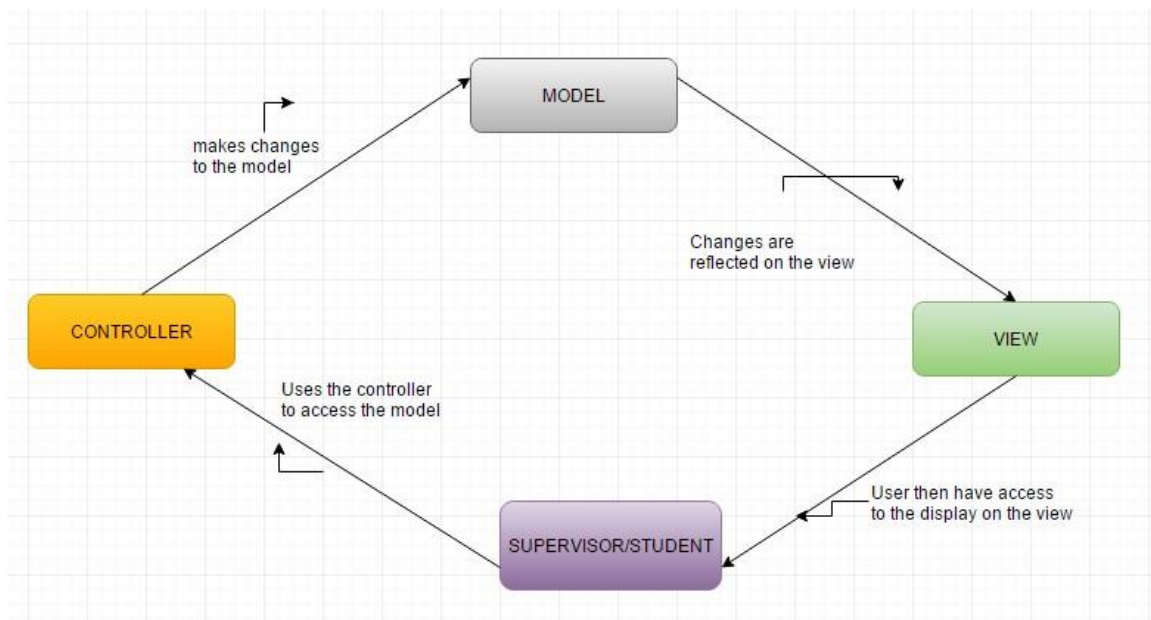


Figure 3.1 MVC Architecture

3.3 Design & Platforms

Bootstrap is the framework is for the project implementation. In addition to other design themes, it was used to develop the GUI of the Application. Bootstrap is used because it is attractive and can be easily used. These are some reasons why bootstrap was chosen amongst others:

- **It is highly responsive:** the application is mobile & web based since bootstrap has amazing tools for this they were utilized from the framework.
- **Ease of customization:** bootstrap has pages that can be customized and this will boost project development.
- **Speed of development:** There is limited time for development of the project. Bootstrap comes handy because it has readymade blocks that can be utilized to speed up the especially the interface design

3.4 Interface Design

This section provides screen shots of samples of the user interface. It gives an overview of the look and feel of the application through an interface walk through. The interfaces are made up of links, buttons for various activities, the application logo and other images.

Figure 3.2 shows the supervisor's home page of the supervisor.



Figure 3.2 supervisor's Home page

SmarTeaMaker
Call: +233 548770298
E-mail: teammaker@gmail.com

Form A Group

Select Course

Select Course ▼

Submit **Cancel**

Student_id	Student_name
Student_id	Student_name

Group Size

Select Algorithm

Select Algorithm ▼

Form group **Cancel**

LOGOUT / SIGNOUT

Quick Links

- My Dashboard
- Create Class
- Form Group
- View Courses
- Enroll Student
- Evaluate Presentation
- Change Password

Figure 3.3 A view to form teams

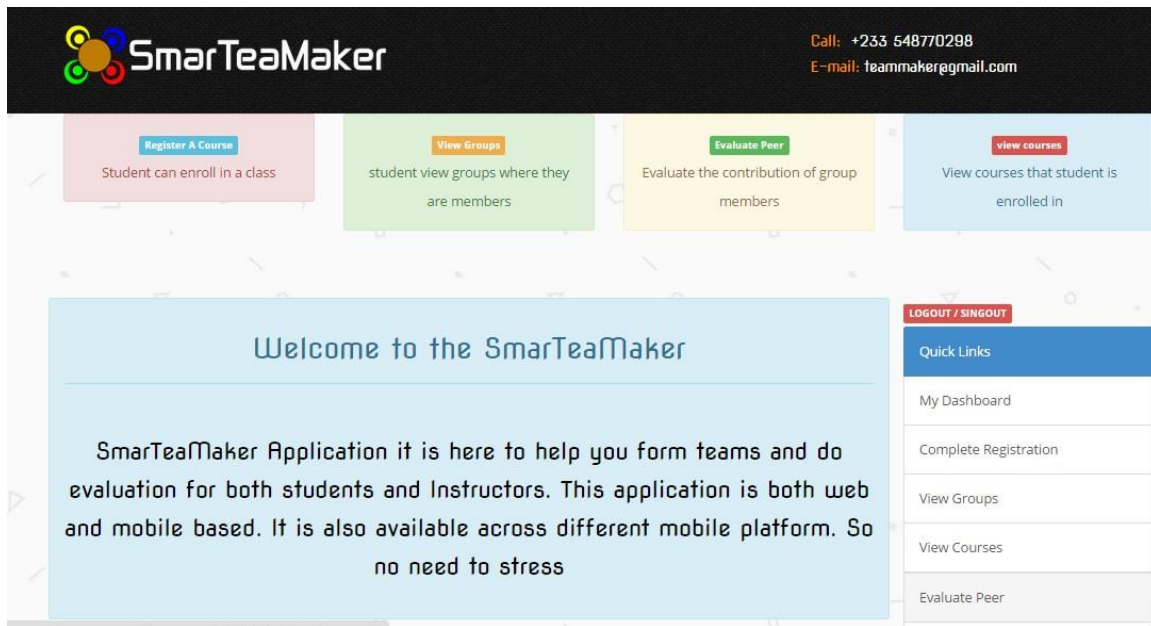


Figure 3.4 Student's Home Page

Figure 3.5 A view to help team members do evaluation

3.5 Database Design

The database design is a major part of the project and hence has to be handled carefully. For the development of the smart team maker software, five strong entities were identified from which the base of the system depend. Also some weak entities were used in order to buttress the system design. Below is the list of entities, their respective attributes, and the entity relation diagram for the project.

3.6 Data Design

The different tables show a data dictionary with different fields and their respective information. It gives a clear view of the logical database design. List of entities and their attributes.

Table 3.1 Course Table

Field Name	Data Type	Other Information	Allow Null	Description
course_id	Varchar	Primary key field	no	Used to uniquely identify a course
course_name	Varchar	Field size: 20, indexed	No	Helps to easily find the course
semester	Varchar	Field size: 20	No	Tells the semester in which the course was taught
department	Varchar	Field size: 20, indexed	No	Tell the specific field of the supervisor
instructor	Varchar	Field size: 20	No	The name of the instructoe of the course
School	Varchar	Field size: 20	No	The particular university since it may go beyond Ashesi

Table 3.2 Student Table

Field Name	Data Type	Other Information	Allow Null	Description
student_id	Varchar	Primary key field	No	Used to uniquely identify the student
student_name	Varchar	Field size: 20, indexed	No	Used commonly to easily reach a student
major	Varchar	Field size: 20, indexed	No	The special field of the student
Yeargroup	Varchar	Field size: 20, indexed	No	The graduation class of the student
Gpa	Varchar	Field size: 10	No	The academic standing
Skills	Varchar	Field size: 20	No	The special ability from a given list
Gender	Varchar	Field size: 10	No	The gender orientation of the student

Table 3.3 Supervisor Table

Field Name	Data Type	Other Information	Allow Null	Description
supervisor_id	Varchar	Primary key field	no	Identifies the supervisor
supervisor_name	Varchar	Field size: 20, indexed	no	Used to easily reach supervisors
Department	Varchar	Field size: 20	no	The specific field to get supervisor

Table 3.4 Student Enrollment Table

Field Name	Data Type	Other Information	Allow Null	Description
enroll_id	Varchar	Primary key field	no	Uniquely identifies each tuple
student_id	Varchar	Field size: 20, foreign key	no	This is to identify enrolled student

Course_id	Varchar	Field size: 20 foreign key	no	The course a student is enrolled into
dateEnrolled	Varchar	Field size: 20, indexed	no	The date the student was registered

Table 3.5 Teams Table

Field Name	Data Type	Other Information	Allow Null	Description
team_id	Varchar	Primary key field	No	It uniquely identifies each team
team_number	Varchar	Field size: 20,	No	The name of the team in numeric value
Course_id	Varchar	Field size: 20, foreign key	No	This tells the course each team belong to

Table 3.6 Team Membership Table

Field Name	Data Type	Other Information	Allow Null	Description
membership_id	Varchar	Primary key field	no	Used to number the teams
student_id	Varchar	Field size: 20, foreign key	no	Used to identify students
team_id	Varchar	Field size: 20 foreign key	no	Unique referenced from the team table

Figure 3.6 shows the entity relationship diagram with all the multiplicities. It gives a clear understanding of the database structure and how the different tables relate with each other.

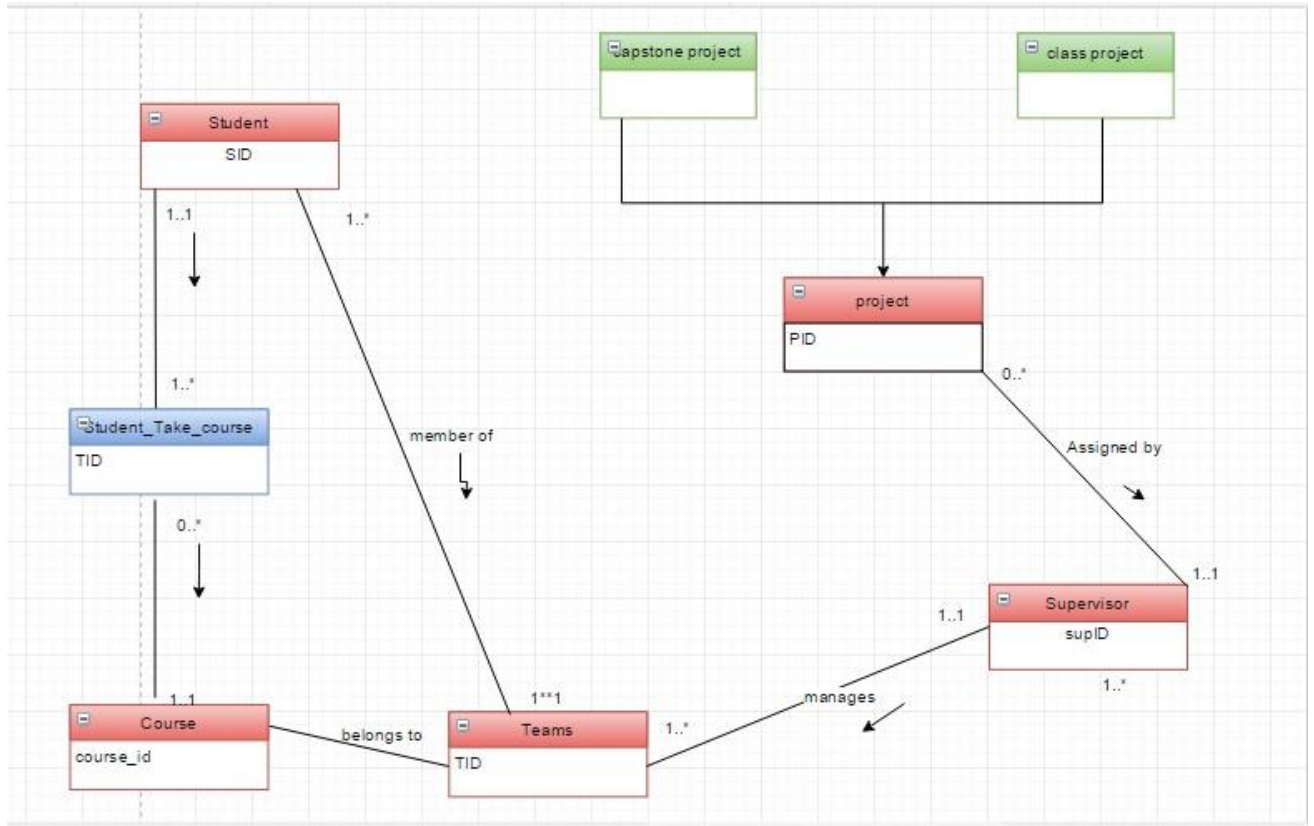


Figure 3.6 ER Diagram

3.7 System Models

The team maker application is broken down into two main modules, namely the project supervisor's module and the team member's module. Supervisor's management module is essentially concerned with all activities linked with the supervisor or faculty member. Due to the reason that the system gives administrative privileges to the systems administrator, this module handles activities such as creating accounts, assigning projects, updating projects, etc. It also handles all activities pertaining to forming teams, adding a class, enrolling students, evaluating presentation etc. In addition, constraints and errors are managed to ensure security, safety, availability and reliability. This module basically contains the

database class, the course class and all other supporting classes required including Ajax and other PHP classes. The figures show the classes attached to the supervisor module

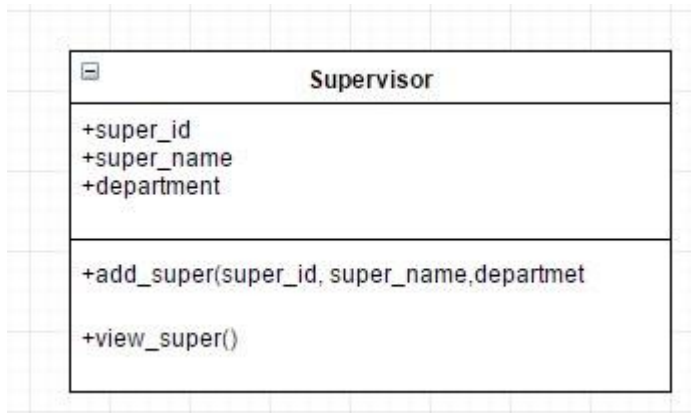


Figure 3.7 Supervisor Class

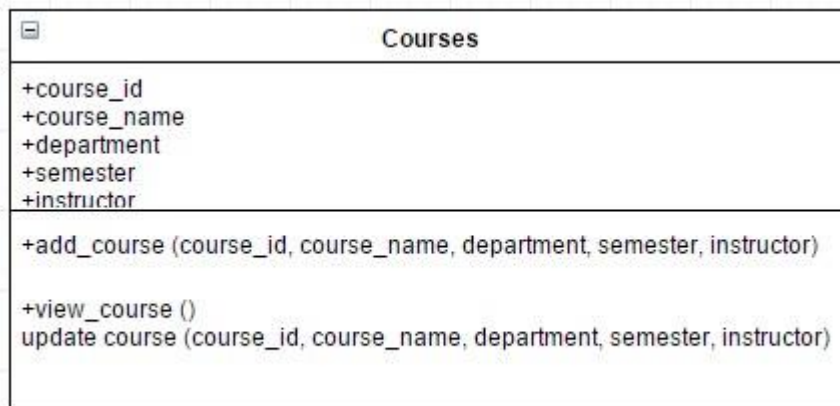


Figure 3.8 Courses Class

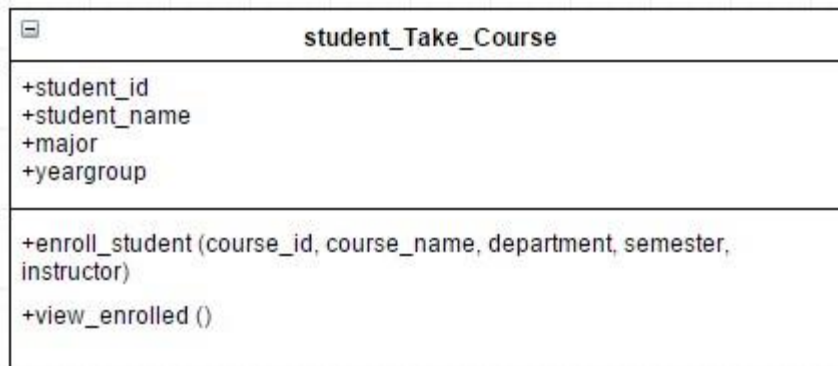


Figure 3.9 Student Tale Course Class

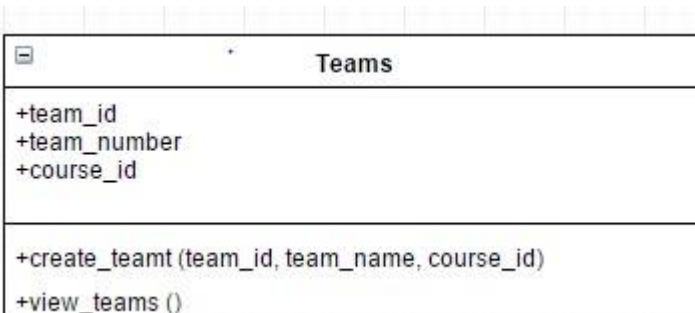


Figure 3.10 Teams Class

Similar to the supervisor management module, the project makes use of a team management module which efficiently controls all activities related to the team class. It implement functions such as join team, view groups, evaluate peer, submit project, view courses, etc. The module defines an outline on which each function implements. Since the team class is strongly dependent on the supervisor class, the module provides a correlation between these two features to foster ease of use. Most of the activities in the different modules communicate with similar tables in the database for concurrency. This is the motivation for implementing the repository system architecture. These two modules and their respective features as defined by the team are compiled to form the structure of the project.

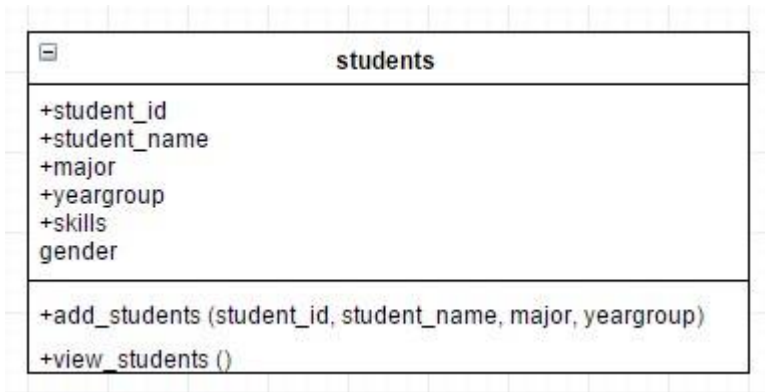


Figure 3.11 Student Class

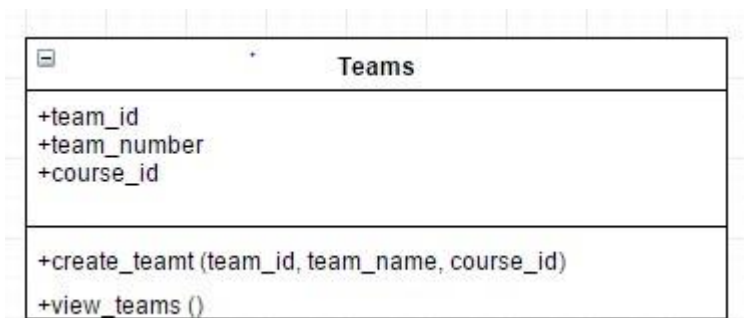


Figure 3.12 Teams Class

3.8 Activity Diagram

Figure 3.13 shows the student's activity diagram

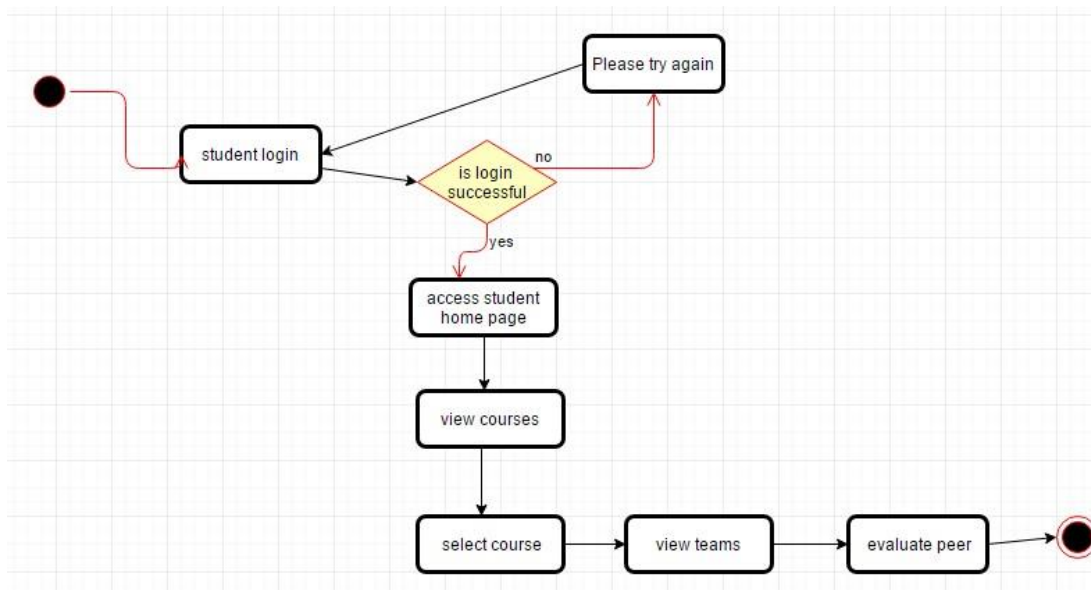


Figure 3.13 Activity Diagram for Student

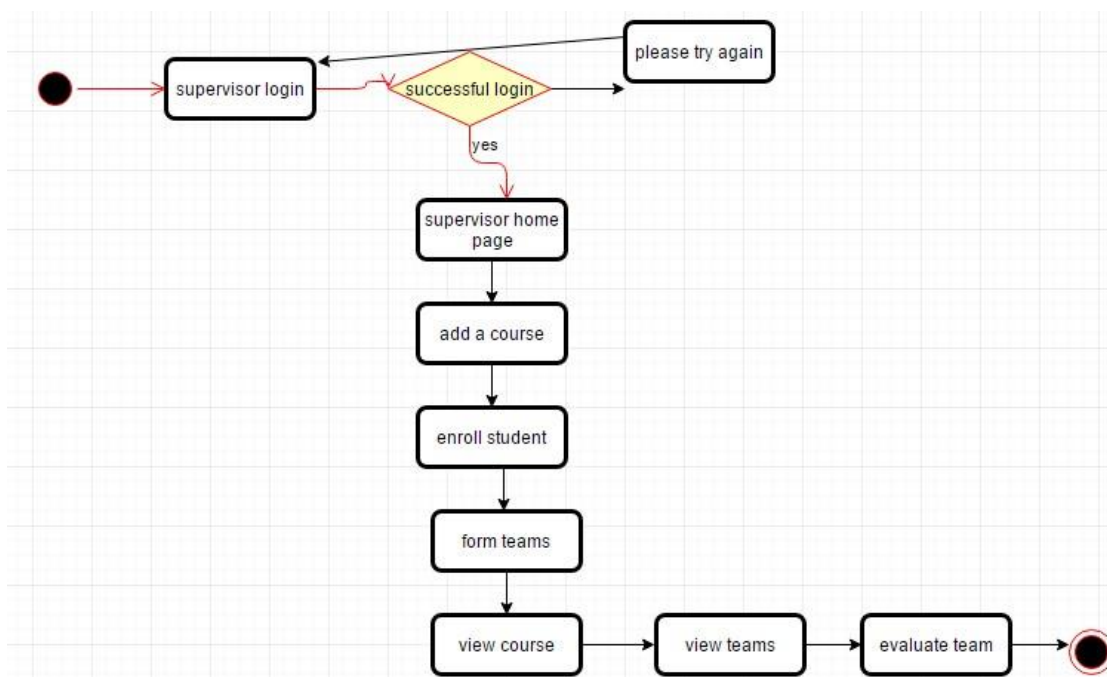


Figure 3.14 Activity Diagram for Supervisor

Chapter 4: Implementation

4.1 Overview

This document gives an overview of the system implementation for the Application. It gives details information of the system overview, system management, implementation support, system performance etc.

4.2 System Organization

The smart team maker runs on two main platforms mobile and web. The application is divided into three main component Account creation, team formation and project evaluation. The combination of these three components make up the complete application.

4.3 System Description

The system supports multiple processes to ensure reliability. The first process is for a supervisor to choose a criteria suitable for the group formation. The system provides a variety of ways in which groups can be formed. It the responsibility of the supervisor/faculty to select a suitable one depending on preference. Another important process is assigning a project by the supervisor. Once the groups have been formed the team can now access the project posted. Also supervisors can evaluate teams after the project is completed and feedback given to teams. In addition, teammates can view groups and make peer evaluations of individual team members.

4.4 Major Tasks

The application accomplishes a number of task for its smooth running. Some of these are managed by the system administrator/supervisor some by the general users of the system. The ability to accomplish or execute a task is dependent on the permission level of the user. Some of the major task of the system are as follows:

- The ability of users to accomplish a onetime login into the system. This is greatly be dependent on the permission of the user. The administrator provides a dummy log in information then the user can change upon access.
- The ability of the supervisor to choose a group forming criteria from which the groups are be formed. When the data is been sent to the list of the prospective group members then the task is completed.
- Supervisors evaluating teams
- Team members submitting a completed task
- Team members doing evaluations.

Chapter 5: Implementation Support

5.1 Hardware, Software, Facilities, and Materials

There are tools and equipment needed for the successful execution of the application. In order for a web application to run successfully, a hardware equipment like a personal computer is required also internet or a form of network is required since it is on web. Libraries for implementation of certain functions are all facilities required.

5.2 Personnel

For the application to run successfully certain personnel needs some sort of training to be able to easily execute the task. Administrator needs training on how to set up an account for users. Also users are introduced to the use of the app on different platforms and in different location to see the ease of use. Installation packages and read me files are given to the right resources personnel to be able to run the app.

5.3 Performance Monitoring

After the system is deployed, it will be monitored for a period of 12 months to identify errors gather feedback to improve it. It will have a side app and a database where all the errors will be log and this can be accessed by the developer to maintain availability.

5.4 Tools and libraries

These are technologies used to implement the smart team maker.

AJAX: Ajax used to make request to the other classes in the system and enhance responsiveness.

PHP: PHP is one of the most widely used languages when it comes to web development. It was chosen because of its ease of use and inclusion of simple formats. Also, PHP offers outstanding usability and efficiency when used for web development and since a good portion of the project is web, it became imperative to employ its use. It also includes fast data processing features and is compatible across operating systems like windows and Linux making it suitable for the project scope. Also there is a wide range of documentation available online for the use of PHP.

CSS: the Cascading Style Sheet was used for describing the HTML presentations. It defines the structure and give a view of elements on the interface. Thus CSS was deployed to do a better styling of the application.

MYSQL: MYSQL was employed to implement the creation of databases in the system. This is because it can serve a broad range of users and handle large volumes of data. It addition it makes deleting, updating inserting and querying of the database easy.

JAVASCRIPT: JavaScript was used to create a responsive page and compliment the static page provided by HTML. In addition it created responsive pages through the request sent by Ajax medium.

HTML: HTML is the markup language used to develop the front end design of the application. Developers are attracted to it due to its high speed and loading time. Also research has shown that all browsers support HTML. It was the base language that the JavaScript, CSS and PHP codes plug into. HTML provides the content that CSS styles, JavaScript enhances whiles PHP links to the database.

Apache Server (Xampp): Xampp is chosen because it has the ability to run the software locally and on the server. Also it is used to harness the functionality in executing the database requirement.

Libraries: In this project, Angular JS was highly utilized, this is because Angular JS framework enables the extension of HTML by acting as the controller portion of the MVC model

Formation Algorithm

This gives an overview of the algorithm used to form groups based on the chosen criteria. It explains what each criteria aims to achieve and how it is implemented to achieve that particular goal.

Random Selection: The random selection criteria aims to form groups from the list of enrolled students in the chosen course. The students in that particular course are fetched from the database and put in an array. A tracker is initialized to one which is the first item on the list and two loops are written to iterate through the list. One keeps track of the size of the list and another keeps track of the number of groups to be formed. Each is updated accordingly until it reaches its exit condition. When the outer loop starts it keeps track of the first item in the list then it enters the inner loop which updates the list based on the number of groups to be formed. E.g. if the number of groups is 4, the second iterator updates the list with 1, 2, 3&4 and then exits. The tracker is updated to start from the next unassigned element on the list. It then enters the inner loop and there is assignment based on the number of groups then exits into the outer loop the tracker is updated and this process continues until the tracker gets to the end of the list. The

values assigned to each number on the list are then re-arranged to form groups. I.e. all elements assigned 1 will be in same group, all 2 will be in the same group and so on.

GPA criteria: Another implemented criteria for forming groups is GPA this allows the formation of groups based on the cumulative GPA of the student. The GPAs of all students enrolled in a particular course are acquired and the list is ordered by GPA. Upon successful ordering of the list an iterator is initialized to the start of the list and using an inner and outer loop as in the case of the random selection, the process is used to form groups.

Temperament criteria: The temperament is an important factor for team formation. Since different individuals have different temperament a test is taken that tells the student their temperament and this information is kept in the database. When the students are fetched to form groups the list is ordered by temperament this keeps similar temperaments together hence during group formation each of the different groups becomes like a box and a temperament is placed in each box by the time the assigning of values is done and when values are re-arranged the temperaments would have been dispersed among the various groups.

Chapter 6: Testing & Results

The testing of the smart team maker was one of the most important elements in the development process. In order to efficiently test the product, two testing methods were employed. These are the development test approach and the user test approach.

6.1 Development Testing

This test was employed during the development process. It was relevant because it helped in the testing of functions and classes that had been implemented. It tested the application on three levels the unit, component and system.

6.1.1 Unit Testing

Unit testing approach is employed on the testing of functions. Below is a list of functions randomly selected on which unit test was performed.

enroll_student() - In order to test this function, the required parameters were provided that is the student_id, student_name, gender, and the course_name. When the test was done the test function returned back the data that was added indication that it runs as expected. The database was queried to confirm and the data was in the database as expected. Here is the test implementation:

```
<?php
class testEnrollment extends PHPUnit_Framework_TestCase
{
    include ("courses.php")
    public function Test_enroll_student()
    {
        // generate test username sername
        $strTestUsername=random_bytes(10)
        $obj=new courses();
```

```

$this->assertEquals(true,
    $obj->enroll_student(
        $strTest,// username sername
        "student_id", //id of the student
        "student_name", //student' name
        "course_name",//provision of the name of the course
        "gender", //gender
    ));
    $id="student_id"
    $this->assertEquals(true,$obj->query("select * from users where
student_id='$id$strTest'"));
    //count the number of fields
    $this->assertCount(7,$obj->fetch());

?>

```

The second test done was attempting to enroll a student who had already been enrolled in the same course and as soon as the parameters were set and the command sent the application sent an error message saying that student is already in the system.

form_teams() – this function allows the supervisor to be able to form teams. During the testing process, the following were provided in the field's class size, formation criteria and course.

When the command was issued it created a random group of five students per group.

view_course() – this function allows the supervisor to view all the courses in the system. Once the view course button is clicked it populates a table of courses from which the user can filter according to preference.

add_course() – the first test done for the add course function was to give it all the necessary parameters needed to add a course. The test returned a true result indicting it works well.

A second test was done to add course but instead of adding a new course some of the fields were left blank the test failed because the system refused null values.

view_groups() – this was tested by viewing all the courses and clicking the view group linked to a particular course and it listed all groups in that course.

evaluate_peer() – this function allows the student in a team to evaluate other members of the team. During the test evaluate button is clicked and the name of team member, evaluator, group number, assigned grade and comment are filled and executed. The test result was true and the data was in the database when queried.

6.1.2 Component testing

Supervisor login: the authentication of the supervisor was done by comparing the login details provided with the ones in the database. Since the account was valid the type of user was then checked because it is a supervisor the supervisor home page was. A test for this component was successful as a supervisor successfully logged in and was redirected to the supervisor home page.

Course page: this displays all the courses in the database and allowed modifications accordingly. This test proved successful because there was access to the course page And other actions like search for a course was possible

Student page: during the testing: this page gave a detail description of each student in the database. It displays detail information about students. This test was successful because the supervisor had access to all this information on the student page.

6.1.3 System Testing

This test was done to confirm and ensure that the system is doing what it is expected to do. The system testing became easier once all units have been properly tested and the different components are tested it was easy to go ahead with the testing of the entire system. About twenty students were enrolled in a course to help form the group. The form group button was clicked the groups size was selected, the course and the formation criteria was random since random was the most common way groups are formed in class. In very little time random groups of size 5 were formed efficiently.

6.2 User Testing

One of the most necessary testing approaches is the user testing. It is relevant that the user test the application and give feedback accordingly. During the testing process three mock testers were used. The test process was as follows:

The user was given a token with which he accessed the system that is a username and password. Once the login was successful, he accessed the home page. The page presented him with lot of options and he knew he could do a number of things but was not sure what to do. He was given the go ahead to explore.

First he decided to add a course, he clicked on the add course button it display the fields he filled them accordingly and submitted. At that point the course was already added to the database but since there is no message or alert to indicate this to the user he was a little lost.

He got the assurance and to confirm this he clicked view and saw the details of course that was provided. He went ahead to enroll a number of student to the course and decided to form group based on the list the first mock tester chose the group formation criteria already

implemented hence he was able to form groups. The second however picked the GPA criteria which is yet to be implemented.

Generally the users gave a positive feedback especially with navigation on the interface i.e. it was easy to use the system.

Chapter 7: Conclusion & Recommendation

7.1 Conclusion

The purpose of this project is to increase efficiency in the formation and evaluation of teams especially in institutions of education. The application permits login for both students and supervisors the main users of the system in order to have access. Supervisors can use the application to add a course and enroll students to that course. Once students are successfully added, teams can be formed from a criteria chosen by the supervisor. Upon successful login students are able to see courses they are enrolled in and groups they are in as well.

Supervisors can also post projects for teams to work on. Upon completion, the group members do evaluations for each member which illustrates the level of commitment of members to the team success. Also the supervisor or supervisors can also do an evaluation of the team's performance.

The systems is developed to be flexible and user friendly as such it runs on both mobile and web. The mobile version also supports the major operating systems. The system fosters interaction among students and supervisors as it contribute to give both students and evaluators the best and efficient experience with regards to team formation and experience.

7.2 Limitation

The smart team maker is limited in a way because it still makes the team formation solely the supervisor's task. The student has no voice or influence over what criteria is used to form the groups or who ends up going to what group. There is no feature on the app that gives the student some control over what group they are in hence except the supervisors moves a

student it is impossible to leave a group. Once groups are formed even the supervisor can't reshuffle.

Also the fact that the supervisor have to manually enroll student can be a limitation when there is a large number of student to be enrolled in a particular class. Some users may forget to complete registration making it had to get them in groups when other criteria apart from random is used to for teams.

Also the student personal data is not encrypted and hidden from other users of the system making them a security threat to the student.

7.3 Recommendation

The first version of the application has implemented and employed most of the design and functionalities. However there is lot of work to be done. The updated version of the application can implement a system which can form better groups with the consent of both supervisor and student. Students should be free to reject or accept a group or ask to be reassigned.

Also student enrollment should be done in a way that students can access the app and after registration and login student should be able to enroll in a class, add a course and drop a course based on preference. It would be nice for the application to send a push notification to allow students received notifications whenever there is an action taken that affects them. In addition instead of posting titles of projects as project assignment, the project assignment for the group can actually be uploaded and the teams can do a download to access the material. It could also be designed to add time constraints for submissions so that if a particular teams submits late it attracts a penalty on acquired grade.

Finally to enhance the security features, every data from the student can be encrypted this will increase data integrity and confidentiality.

Bibliography

- Barbara Oakley, R. B., & Imad, E. (2004). Turning Student Groups into Effective. *Journal of student centered Learning*.
- Chiriac, E. H. (2014). Group work as an incentive for learning – students’ experiences of group work. *Frontiers in psychology*.
- Coacher, V. (2011). *Team building days*. Retrieved from voice coaching London:
<http://www.voicecoachinglondon.co.uk/why-is-team-building-important.html>
- Gallies, R. M. (2003). The behaviours, interactions, and perceptions of students during small group learning. *Journal of Education*.
- Hansen, R. S. (2006). Benefits and problems with student teams. *J education*.
- Konak, S., Konak, A., & Esparangoza, I. (2014). Peer Evaluation and Assessment Resource to Access Student Professional Skills.
- Konnikova, M. (2010). *Teamwork? Good until it’s not*. Retrieved from artful choices:
<http://bigthink.com/artful-choice/teamwork-good-until-its-not-when-working-alone-worksbest>
- Loughry, M. L., & Ohland, M. W. (2014). Assessing team work skills for assurance of learning *journal of marketing education*, 5-19.
- O’Connel, J. (2010). *Team maker an easy random team generator*. Retrieved from make use of groups: <http://www.makeuseof.com/tag/team-maker-random-team-generator/>
- Potosky, D., & Duke, J. (2007). Forming Teams for Classroom Projects. *Developments in Business Simulation and Experiential Learning*.

Queensland, U. (2016). *Problems associated with group work*. Retrieved from student services:

<http://www.uq.edu.au/student-services/learning/problems-associated-group-work>

Simmons, C. S. (2015). Using CATME Team-Maker to Form Student Groups in a *journal of mathematics and aerospace engineering*.