

# **University Project Management**

BY

Sidharth Kumar

K. Suneeth

Siddam Manish Sagar

E. Vinay Kumar

N. Sharat Chandra

A PROJECT REPORT

SOFTWARE ENGINEERING LABORATORY

FOR B.TECH. IN COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

NARAYANGUDA, HYDERABAD, 5000029

(AFFILIATED TO JNTUH)

NOVEMBER 2019

# Table of Contents

1.INTRODUCTION.....	1
1.1 Purpose.....	1
1.2 Scope.....	1
1.3 Overview.....	1
1.4 Process Model.....	2
2.0 REQUIREMENT SPECIFICATION.....	3
2.1 High-level Functional Requirements.....	3
2.1.1 REQCF1: Login.....	3
2.1.2 REQCF2: Projects.....	3
2.1.3 REQCF3: Groups.....	3
2.1.4 REQCF4: Add/Delete.....	3
2.1.5 REQCF5: Upload.....	3
2.2 Requirements Specification.....	4
2.2.1 Use Case Diagrams.....	4
2.2.1.1 University Project Management.....	4
2.2.1.2 UID1 Login.....	5
2.2.1.3 UID2 Projects.....	6
2.2.1.4 UID3 Groups.....	7
2.2.1.5 UID4 Add/Delete.....	8
2.2.1.6 UID5 Upload.....	9
2.2.3 Use Case Specification.....	10
2.2.3.1: UC501:Login Use Case.....	10
2.2.3.2: UC517:Project Use Case.....	10
2.2.3.3: UC51J:Groups Use Case.....	11
2.2.3.4: UC51K:Add/Delete Use Case.....	11
2.2.3.5: UC504LE:Upload Use Case.....	12
2.3 Non-functional Requirements.....	13
3. PRELIMINARY DESIGN.....	14
3.1 Class Diagrams.....	14
3.1.1 University Project Management Class Diagram.....	14
3.1.2 Login and Registration Class Diagram.....	15
3.1.3 Project Class Diagram.....	17
3.1.4 Groups Class Diagram.....	18
3.1.5 Add/Delete Class Diagram.....	18

3.1.6 Upload Class Diagram.....	20
4. DETAILED DESIGN.....	21
4.1 GUI Design.....	21
4.1.1 User Login And Registration.....	21
4.1.2 Projects.....	23
4.1.3 Groups.....	24
4.1.4 Add/Delete Student.....	25
4.1.5 Uploading Documents.....	26
5. IMPLEMENTATION AND TESTING.....	27
5.1 Test Case Design.....	27
5.1.1 Login Test Case Design.....	27
5.1.2 Create and Select Projects Test Case Design.....	28
5.1.3 Grouping Test Case Design.....	28
5.1.4 Add/Delete Student Test Case Design.....	29
5.1.5 Upload Documents Test Case Design.....	29
5.2 Implementation.....	31
6. CONCLUSION AND FUTURE WORK.....	36
Appendix A: Requirement Traceability.....	37
Appendix B: Design Traceability.....	38
Appendix C: Data Dictionary.....	39
Appendix D: Test Log.....	40

# **1.INTRODUCTION**

## **1.1 Purpose**

This project should help faculty/guides to manage and track their final year students' projects. The application should have the following requirements:

- The faculty should be able to login and create projects for the students.
- The students should be able to login and select the projects of their interest.
- The web-based application should automatically form teams (3-4) based on the student preference of projects.
- It should provide administrative functions to faculty like adding and deleting students from the project.
- The students should be able to check the deadlines and submit documents and code related to the project.

## **1.2 Scope**

This project will only implement the software necessary to implement the services for the faculty and student. The university server is assumed to be already implemented and this system will interface with the university server to achieve its objectives. Only the user interface prototypes of the systems will be implemented in this project to understand the requirements. Prototypes will be thrown away when the actual system is implemented.

## **1.3 Overview**

Section 2. lists all the high-level functional requirements of the system. These requirements were elicited from the description provided for the projects and discussions with the customer/instructor of this course. Section 3. provides the preliminary design in terms of identifying the classes using Object-Oriented Analysis and Design approach. Class Responsibility and Collaboration (CRC) technique was used for this purpose and class diagrams. Section 4. provides the activity diagrams and GUI prototype designs and activity diagrams for work flow analysis. Section 5. provides test case design tables along with implementation details and any screen shots of the actual user interface prototype implementation of the system. Section 6. will contain the conclusions and future work of the project.

## 1.4 Process Model -

### SCRUM:

Scrum is the type of Agile framework. It is a framework within which people can address complex adaptive problem while productivity and creativity of delivering product is at highest possible values.

Scrum uses Iterative process.

#### Silent features of Scrum are:

- Scrum is light-weighted framework.
- Scrum emphasizes self-organization.
- Scrum is simple to understand.
- Scrum framework help the team to work together

#### Lifecycle of Scrum:

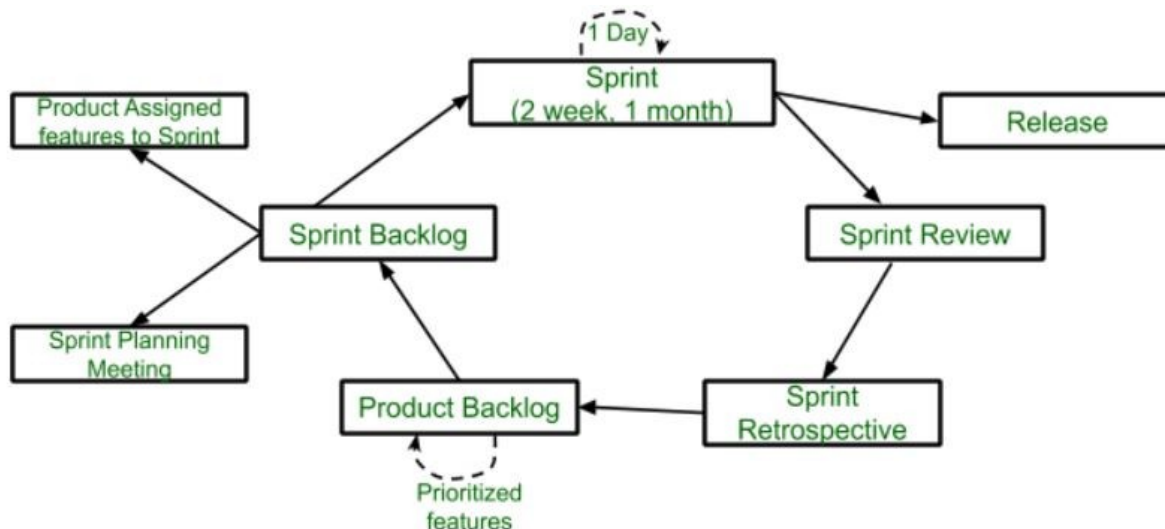


Figure 1.4 Lifecycle of Scrum

#### Advantage of using Scrum framework:

- Scrum framework is fast moving and money efficient.
- In Scrum customer satisfaction is very important.

#### Disadvantage of using Scrum framework:

- Scrum framework do not allow changes into their sprint.
- It can be difficult for the Scrum to plan, structure and organize a project that lacks a clear definition.

## **2.0 REQUIREMENT SPECIFICATION**

### **2.1 High-level Functional Requirements**

The function requirements of the system specify the behaviour of the system. Those functional requirements are the responsible for specifying the methods and operations and these operations describe your project.

#### **2.1.1 REQCF1: Login**

A student and faculty has login to their respective pages of their University web application. In case don't have any login, they need to sign up for new registration.

#### **2.1.2 REQCF2: Projects**

Faculty has a right to create projects. Students who are interested in can select the projects which are created by faculty and every student has right to select any project.

#### **2.1.3 REQCF3: Groups**

When student selects the project that they are interested in, they were grouped based on their respective projects chosen. A group of two is formed based on the order of chosen. In case if there are odd number of students chosen this project the last person is added to any random group.

#### **2.1.4 REQCF4: Add/Delete**

Faculty can add or delete students based on their performance. He maintains all the record of students of each project. He manages whether the work is done by student based on it faculty either add or delete students.

#### **2.1.5 REQCF5: Upload**

Faculty puts a deadline to the submission of project, students need to finish it on time and submit or upload files. Once deadline goes off no student cannot submit or upload file.

## 2.2 Requirements Specification

### 2.2.1 Use Case Diagrams

To run a web application, we need proper use of all functional requirements. These use case diagrams are used to simplify how we are making those functions work. This diagram is used to represent the functions to be performed on the university project management web application. There is a rectangular box it is called a system and the variables or methods defined in it gives the link and the functional work of it.

#### 2.2.1.1 University Project Management

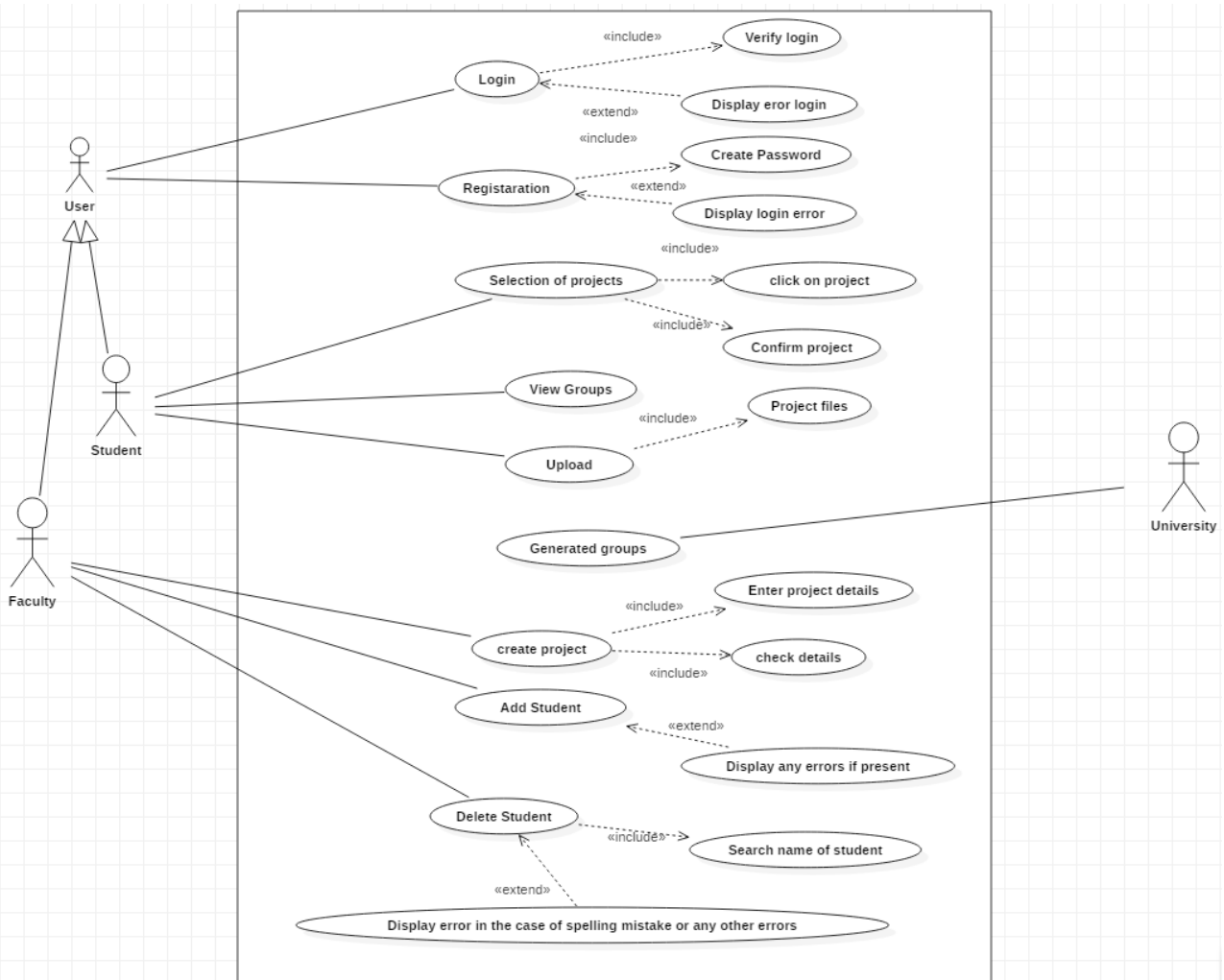


Figure 2.2.1.1 University Project Management Use Case Diagram

Here we have two user student and faculty. Both have separate logins if there is no login, sign up to register. Enter your details user name and password now you will get into your respective pages of your website application. Faculty creates project and students who are interested can select their respective projects. Students, who choose same project are grouped together. A group of 2 is made, if there are odd number of students chosen the project the last person who choose the same project is added to any random group of their respective project. The groups formed are viewed faculty and student. Faculty has a right to add or delete students depending on their work done. They maintain a record of each student according to their performance. Faculty puts a deadline to all projects, students who has completed can submit their projects. If they miss a dead line they are not allowed to submit the project.

### 2.2.1.2 UID1 Login

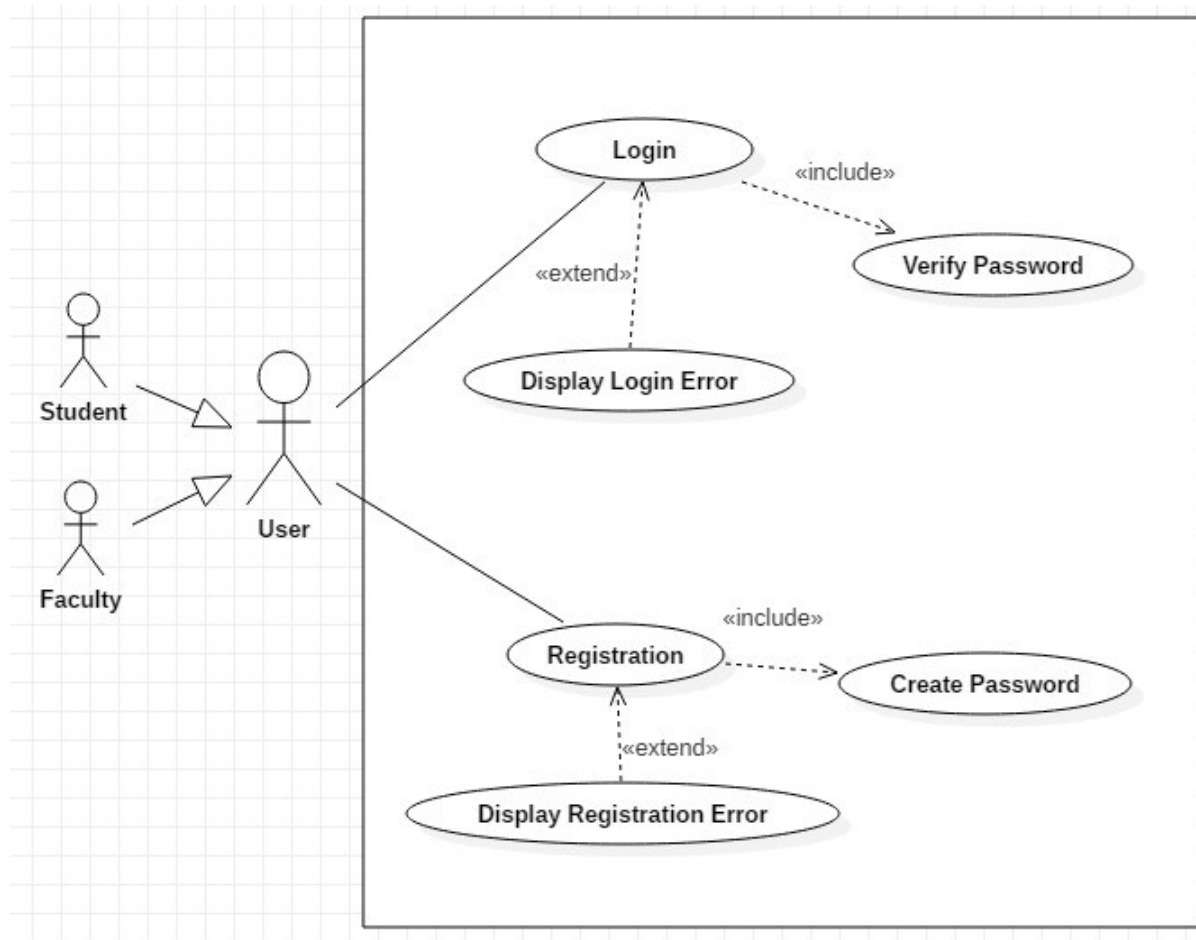


Figure 2.2.1.2 Login Use Case Diagram

Student and Faculty can register with their Email and can create password to their account. Student and Faculty can login to their respective pages by entering valid email and password. System will display login and registration errors to user(Student or Faculty).



### 2.2.1.3 UID2 Projects

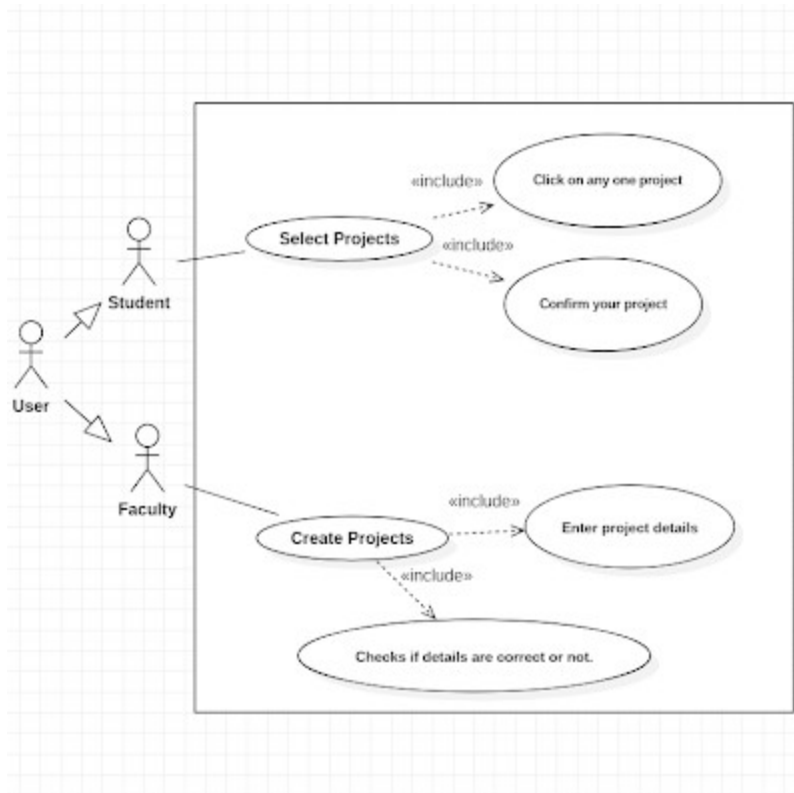


Figure 2.2.1.3 Projects Use Case Diagram

- Faculty can create projects by entering project details in the required fields .
- Students can see and select any project created by the faculty and confirm them.

#### 2.2.1.4 UID3 Groups

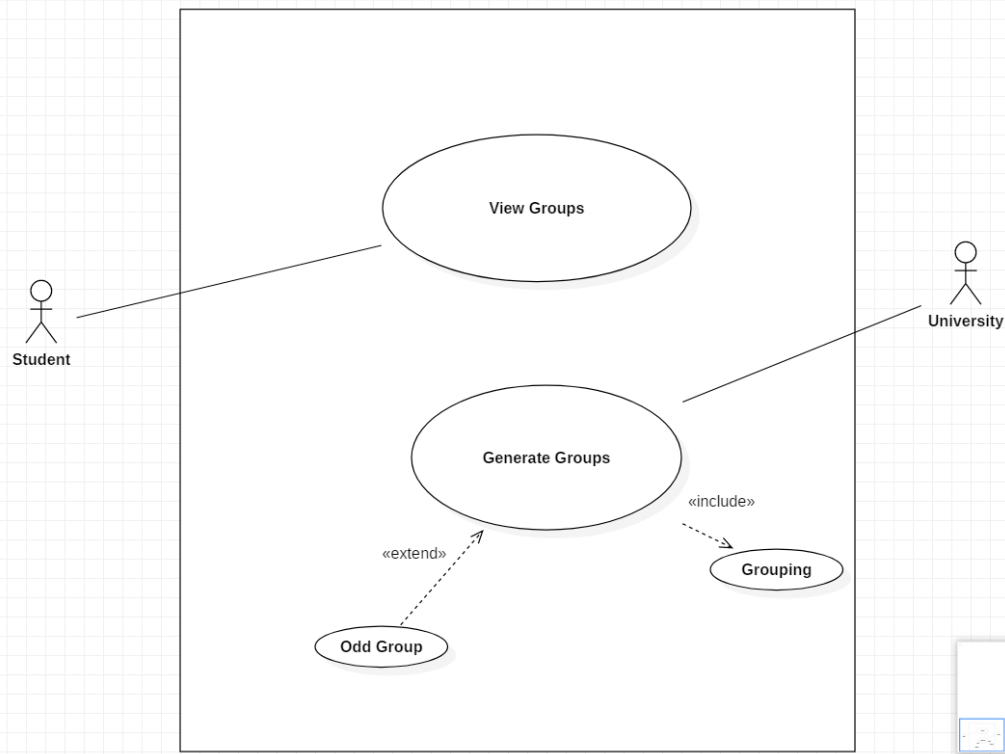


Figure 2.2.1.3 Groups Use Case Diagram

Students of same groups are generated base on their respective projects and they can viewed by a student of same group or a faculty.

### 2.2.1.5 UID4 Add/Delete

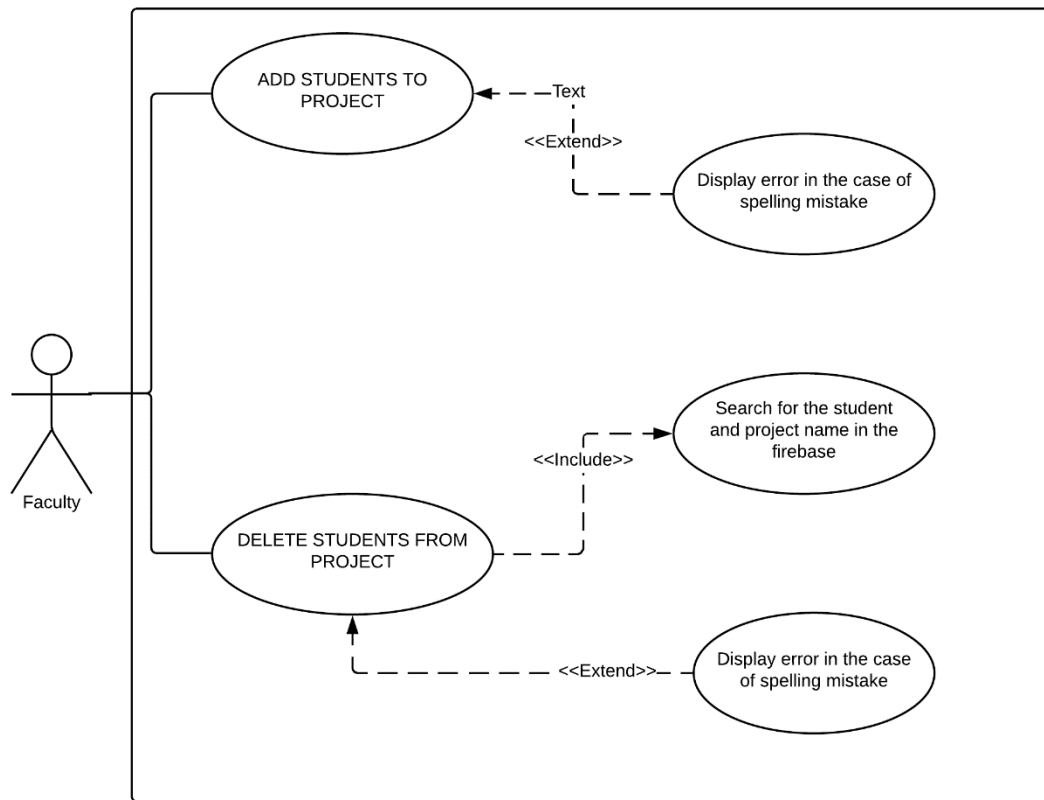


Figure 2.2.1.5 Add/Delete Use Case Diagram

- Faculty can add students to a project by clicking on add and if requirements are not correct it will display the error message.
- Faculty can delete students from a project by choosing the delete option and if requirements are not correct it will display the error message.

### 2.2.1.6 UID5 Upload

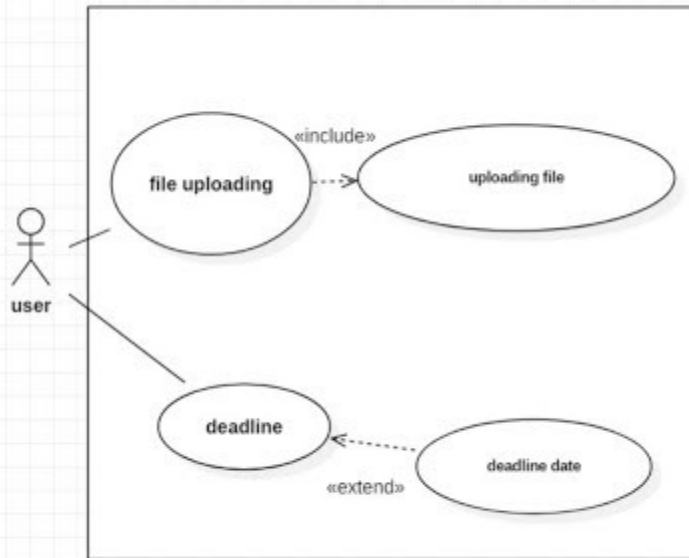


Figure 2.2.1.6 Upload Use Case Diagram

Faculty set a deadline to project. Students should submit their project on or before deadline. Students who fail to submit on or before dead line cannot able to submit their files.

### 2.2.3 Use Case Specification

The use case specifications defines in brief about the use case diagrams. It describes in brief about all functions and operations.

#### 2.2.3.1: UC501:Login Use Case

Table 2.2.3.1: Login Use case

Use Case ID:	UC 501		
Use Case Name:	Student And Faculty Login		
End Objective:	Student and Faculty should be able to login to there respective pages		
Created by:	E.Vinay Kumar	On (date):	17 August 2019
User/Actor:	Student and Faculty		
Trigger:	Student/Faculty initiates by clicking on login button.		
Basic Flow			
User Actions		System Actions	
1.Student/Faculty clicks on Login button.		1.Application will ask to enter login credentials.	
2.Student/Faculty will Enter login credentials and Clicks on Login Button.		2.Applicatin will show Respective Page/Activity to Student/Faculty	
Exception Flow			
User Actions		System Actions	
Student/Faculty enters wrong login credentials.But Student/Faculty is not aware of this.		Application checks and gives message saying Please check your Login credentials	

#### 2.2.3.2: UC517:Project Use Case

Table 2.2.3.2: Project Use case

Use Case ID:	UC 517		
Use Case Name:	Project Creation and Selection.		
End Objective:	Faculty can create projects and students can select projects created by projects.		
Created by:	N.Sharat Chandra	On (date):	18 August 2019
User/Actor:	Student and Faculty		
Trigger:	When student clicks on one of project from project list and when faculty clicks on create button.		
Basic Flow <The optimal or normal ("good day") flow of events. The basic flow of events should describe the events that walk through a successful scenario. The basic flow should not include "and/if scenarios">			
User Actions		System Actions	
1.Faculty clicks on create button.		1.Faculty will have to enter project name and deadline.	
2.Student clicks on one of the project.		2.He will info about the project then he can click on submit.	
Exception Flow <identify system and data error conditions that could occur for each step in the normal flow>			
User Actions		System Actions	
1.When faculty does not enter project name or deadline.		1.A message will pop up saying to " Enter project name or Deadline".	

### 2.2.3.3: UC51J:Groups Use Case

Table 2.2.3.3: Groups Use case

Use Case ID:	UC 51J		
Use Case Name:	Students and their Projects		
End Objective:	Grouping of students having similar Projects		
Created by:	Siddam Manish Sagar	On (date):	18 August 2019
User/Actor:	Faculty and Student		
Trigger:	When student selects a project it automatically triggers function		
Basic Flow <The optimal or normal ("good day") flow of events. The basic flow of events should describe the events that walk through a successful scenario. The basic flow should not include "and/if scenarios">			
User Actions		System Actions	
When student selects a Project.		It groups the students of similar project by limiting the size of group(2 or 3).	
Exception Flow <identify system and data error conditions that could occur for each step in the normal flow>			
User Actions		System Actions	
we have grouped even of students incase if it is odd i.e only single student left.		It allots him in any random group.	

### 2.2.3.4: UC51K:Add/Delete Use Case

Table 2.2.3.4: Add/Delete Use case

Use Case ID:	UC 51K		
Use Case Name:	Addition and deletion of students entry from the projects		
End Objective:	Faculty will able to add and delete students from projects.		
Created by:	Sidharth Kumar	On (date):	18 August 2019
User/Actor:	Faculty		
Trigger:	Students can be added and deleted from the database by click on add and delete button respectively.		
Basic Flow <The optimal or normal ("good day") flow of events. The basic flow of events should describe the events that walk through a successful scenario. The basic flow should not include "and/if scenarios">			
User Actions		System Actions	
Faculty enters the name of the student and project in the label provided. Then he can click on delete or add button to do deletion or addition operations respectively.		When the name of the student and project is entered in the label and add button is clicked, a new entry is created in the database under the project mentioned.  When the name of the student and project is entered in the label and delete button is clicked, the system searches for an entry in the database with that name and deletes that entry in that project.	
Exception Flow <identify system and data error conditions that could occur for each step in the normal flow>			
User Actions		System Actions	
If faculty enters the wrong spelling of a student or project and then click on the delete button.		The system searches for an entry with that name if that name is not present it popup an error message that "Student or Project with this name is not present".	

### 2.2.3.5: UC504LE:Upload Use Case

Table 2.2.3.5: Upload Use case

Use Case ID:	UC 504LE		
Use Case Name:	STUDENT DEADLINE AND SUBMISSION BUTTON		
End Objective:	DEADLINE AND SUBMISSION		
Created by:	K.SUNEETH	On (date):	18 AUGUST 19
User/Actor:	FACULTY AND STUDENT		
Trigger:	STUDENT SUBMISSION		
Basic Flow <The optimal or normal ("good day") flow of events. The basic flow of events should describe the events that walk through a successful scenario. The basic flow should not include “and/if scenarios”>			
User Actions		System Actions	
CHECKS FOR DEADLINE		PROVIDING EXACT DEADLINE AND SUBMISSION	
Exception Flow <identify system and data error conditions that could occur for each step in the normal flow>			
User Actions		System Actions	
CLICK ON SUBMIT BUTTON		IT POP UP SUBMISSION SUCCESSFULL	

## **2.3 Non-functional Requirements**

### **2.3.1 Login:**

Student and Faculty Login credentials will be linked to the Email that they provide.

### **2.3.2 Project:**

Students can multiple projects and faculty can create multiple projects.

### **2.3.3 Groups:**

Students could be able to see their teammates who are randomly grouped.

### **2.3.4 Add/Delete:**

Faculty can easily form groups of the students and can add and delete students from a project at any time.

Faculty can easily manage and maintain the records of the projects done by each student.

### **2.3.5 Upload:**

Students can also see brief details about the project.



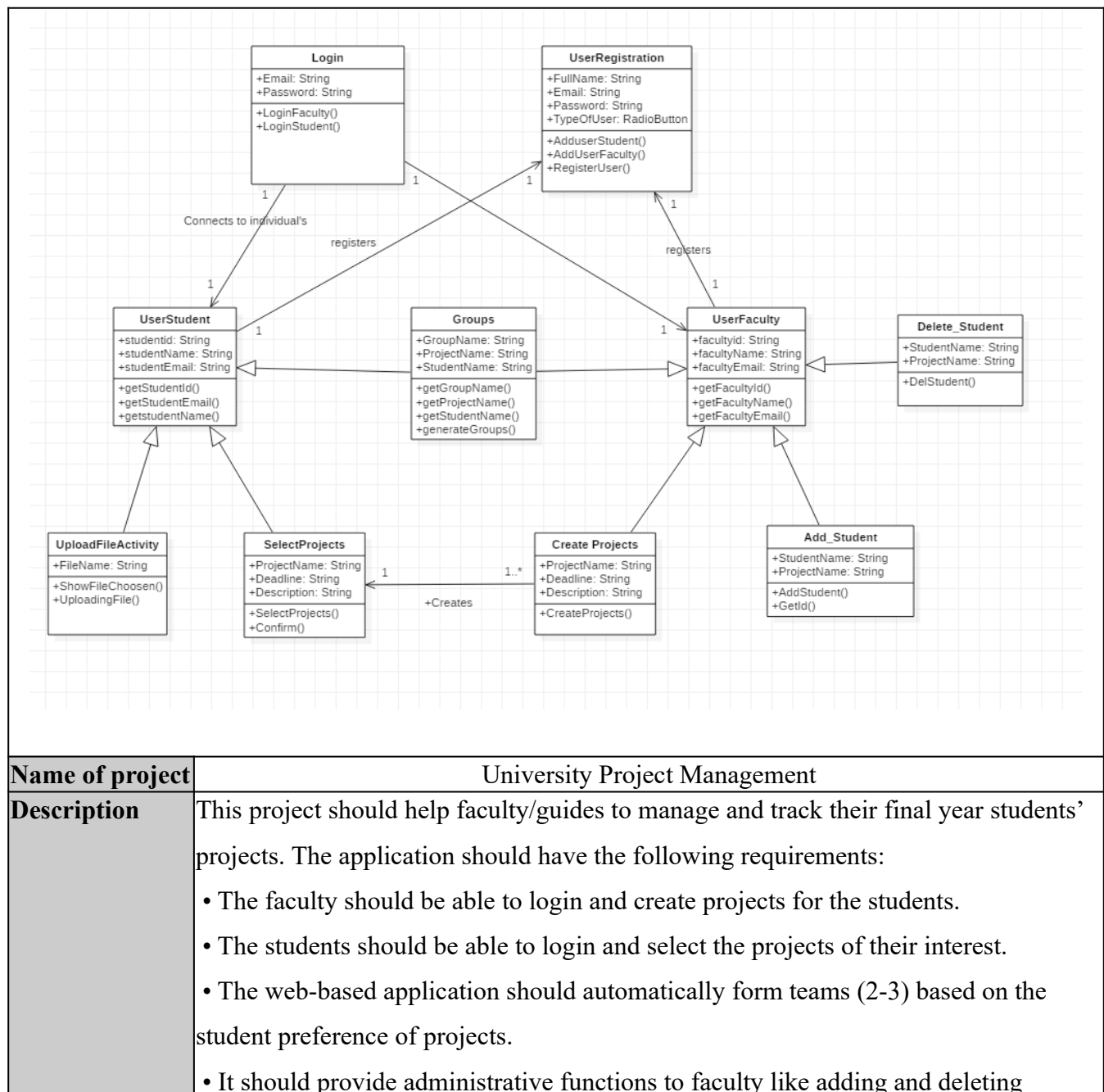
### 3. PRELIMINARY DESIGN

#### 3.1 Class Diagrams

##### 3.1.1 University Project Management Class Diagram

This class diagram is the complete diagram of the entire team.

Table 3.1.1: University Project Management Class Diagram

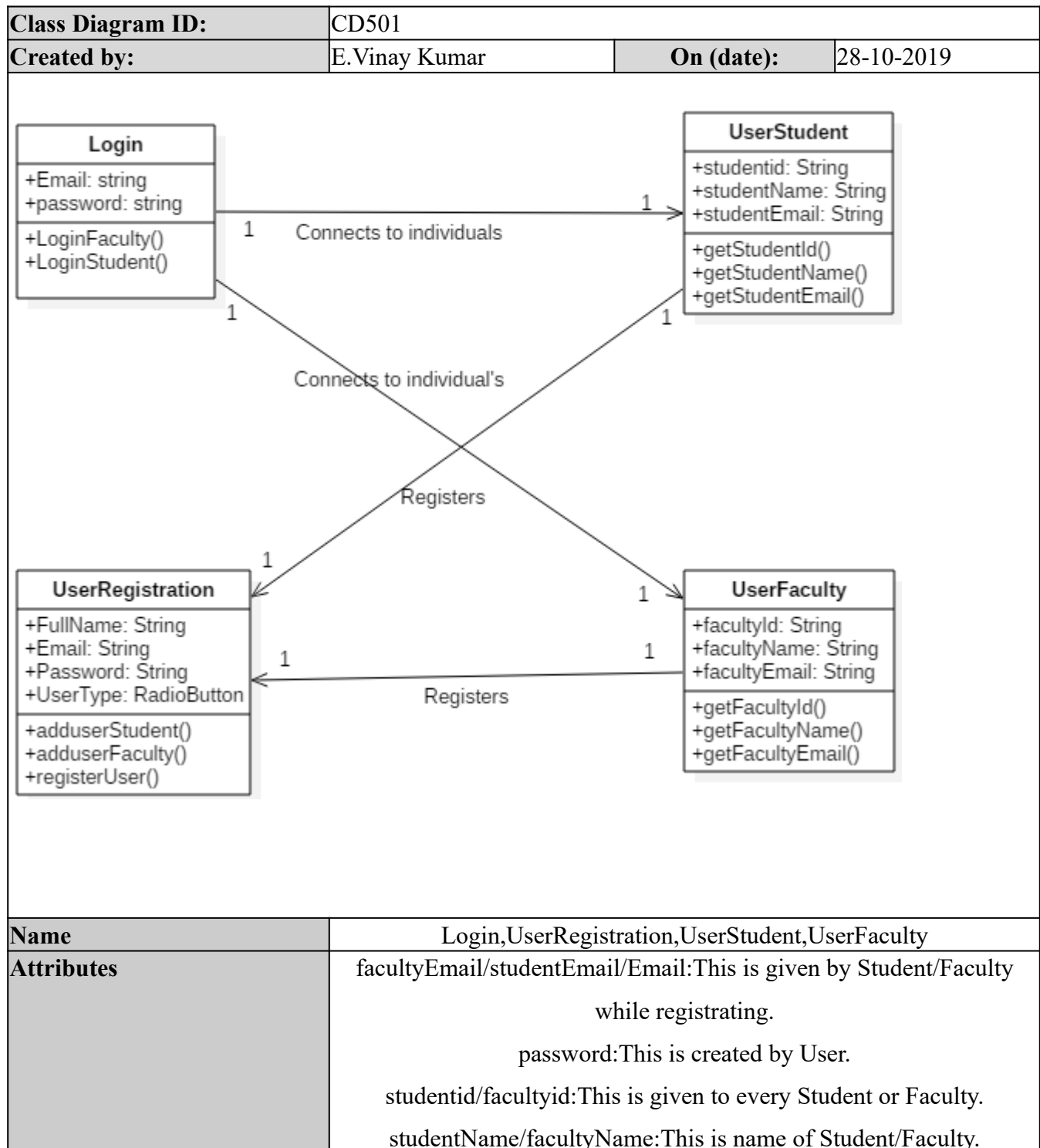


students from the project.

- The students should be able to check the deadlines and submit documents and code related to the project.

### 3.1.2 Login and Registration Class Diagram

Table 3.1.2: Login and Resgistration Class Diagram



	FullName:This is Fullname of Student/Faculty
<b>Methods</b>	<p>LoginFaculty()/LoginStudent():This method is used to Login as Student/Faculty</p> <p>getStudentid()/getfacultyid():This method is used to get Student/Faculty id</p> <p>getStudentName()/getFacultyName():This method is used to get Student/Faculty names</p> <p>getStudentEmail()/getFacultyEmail():This method is used to get Student/Faculty Emails</p> <p>registerUser():This is used to register user as Student/Faculty</p> <p>adduserStudent()/adduserFaculty():This is used to add user to databases</p>
<b>Relationship</b>	<p>UserStudent and UserFaculty are related to Login by Simple Association :Login class connects individually to UserStudent/UserFaculty i.e one to one.</p> <p>UserStudent and UserFaculty are related to UserRegistration by simple association: UserRegistration registers one account to one user i.e one to one.</p>

### 3.1.3 Project Class Diagram

Table 3.1.3: Project Class Diagram

<b>Class Diagram ID:</b>	CD517		
<b>Created by:</b>	N.Sharat Chandra	<b>On (date):</b>	28 - 10 – 2019
<pre> classDiagram     class CreateProjects {         +Project Name: String         +Deadline: String         +Description: String         +createProjects()     }     class SelectProjects {         +Project Name: String         +Deadline: String         +Description: String         +selectProject()         +confirm()     }     CreateProjects "1..*" --&gt; "1" SelectProjects : creates         </pre>			
<b>Name</b>	Select Project , Create Project.		
<b>Attributes</b>	<p>Project Name: Name of project to be created.</p> <p>Deadline: Date on which project should be submitted</p> <p>Description: Project description telling what one must do in the project.</p>		
<b>Methods</b>	<p>selectProject(): This method is used for storing the project selected by student in database.</p> <p>createProject(): This method is used for creating project by faculty in database.</p>		
<b>Relationship</b>	Select Project is related to Create Project by association : Create Project class can create multiple projects.		

### 3.1.4 Groups Class Diagram

Table 3.1.4: Groups Class Diagram

<b>Class Diagram ID:</b>	CD51J		
<b>Created by:</b>	Siddam Manish Sagar	<b>On (date):</b>	28-10-2019
<pre> classDiagram     class Groups {         +GroupName: String         +ProjectName: String         +StudentName: String         +getGroupName()         +getStudentName()         +getProjectName()         +generateGroups()     } </pre>			
<b>Name</b>	Groups		
<b>Attributes</b>	<p>GroupName: Name of the group according to their order of respective project.</p> <p>ProjectName: Name of the project that they are selected.</p> <p>StudentName: Name of the student of a particular group.</p>		
<b>Methods</b>	<p>GetGroupName(): This method is used to get group name of a particular project.</p> <p>GetProjectName(): Name of a particular project that has chosen.</p> <p>GetStudentName(): Name of the student who was in a project.</p> <p>GenerateGroups: A group of 2 people selected as team according to the order they have chosen. Max limit of the group is 3 (if odd number of students have chosen project)/</p>		
<b>Relationship</b>	On clicking view groups, groups are generated based on their respective projects and the order they have chosen.		

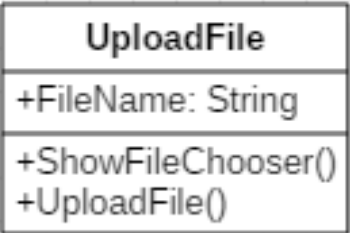
### 3.1.5 Add/Delete Class Diagram

Table 3.1.5: Add/Delete Class Diagram

<b>Class Diagram ID:</b> CD51K											
<b>Created by:</b>	Sidharth Kumar	<b>On (date):</b>	28-10-2019								
<div><div><div><div>Delete_Student</div><div><div>-StudentName: String</div><div>-ProjectName:String</div></div><div><div>+DelStudent()</div></div></div><div><div>Add_Student</div><div><div>-StudentName: String</div><div>-ProjectName:String</div></div><div><div>+AddStudent()</div><div>+GetID()</div></div></div><div><div>Firestore</div><div><div>FireBase_Name</div><div><div>-StudentName: String</div><div>-ProjectName: String</div><div>-UniqueID: String</div></div></div></div><div><div>Delete Student</div></div><div><div>Add Student</div></div></div></div> <table><tr><td><b>Name</b></td><td>Insert and Delete students from the project.</td></tr><tr><td><b>Attributes</b></td><td><p>Student and project name: This is given by the faculty in the label provided.</p><p>Unique ID: It is generated by the firebase when a student is added to a project.</p></td></tr><tr><td><b>Methods</b></td><td><p>addStudent(): This is used to add a particular student to a project.</p><p>delstudent(): This is used to delete a student from a particular project.</p><p>getID(): This is used to get the id generated by the firebase after the addition of a student into a project.</p></td></tr><tr><td><b>Relationship</b></td><td>addstudent and delstudent are related by association: When a student is added to a project it generates a unique id in the database which is stored in a list. When deleted is called then a student is deleted from firebase and its ID is deleted from the list.</td></tr></table>				<b>Name</b>	Insert and Delete students from the project.	<b>Attributes</b>	<p>Student and project name: This is given by the faculty in the label provided.</p> <p>Unique ID: It is generated by the firebase when a student is added to a project.</p>	<b>Methods</b>	<p>addStudent(): This is used to add a particular student to a project.</p> <p>delstudent(): This is used to delete a student from a particular project.</p> <p>getID(): This is used to get the id generated by the firebase after the addition of a student into a project.</p>	<b>Relationship</b>	addstudent and delstudent are related by association: When a student is added to a project it generates a unique id in the database which is stored in a list. When deleted is called then a student is deleted from firebase and its ID is deleted from the list.
<b>Name</b>	Insert and Delete students from the project.										
<b>Attributes</b>	<p>Student and project name: This is given by the faculty in the label provided.</p> <p>Unique ID: It is generated by the firebase when a student is added to a project.</p>										
<b>Methods</b>	<p>addStudent(): This is used to add a particular student to a project.</p> <p>delstudent(): This is used to delete a student from a particular project.</p> <p>getID(): This is used to get the id generated by the firebase after the addition of a student into a project.</p>										
<b>Relationship</b>	addstudent and delstudent are related by association: When a student is added to a project it generates a unique id in the database which is stored in a list. When deleted is called then a student is deleted from firebase and its ID is deleted from the list.										

### 3.1.6 Upload Class Diagram

Table 3.1.6: Upload Class Diagram

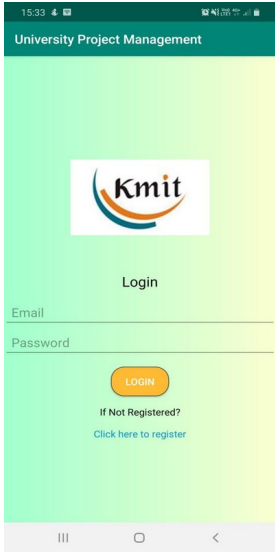
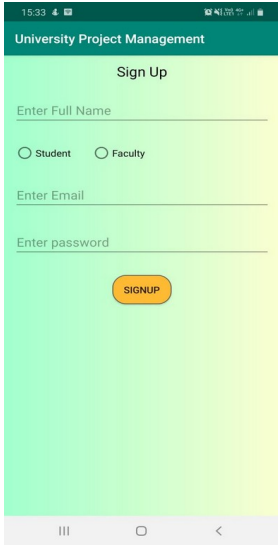
<b>Class Diagram</b>	CD504		
<b>Id:</b>			
<b>Created By:</b>	K.Suneeth	<b>On (Date):</b>	28-10-2109
 <pre> classDiagram     class UploadFile {         +FileName: String         +ShowFileChooser()         +UploadFile()     }         </pre>			
<b>Name</b>	Selecting, choosing and uploading file.		
<b>Attributes</b>	File name. Choosing file. Uploading file.		
<b>Methods</b>	Submit(): to upload file in fire base.		
<b>Relationship</b>	Adding file name is related to choose the file by associated: check the file in Firebase and choosing.		

## 4. DETAILED DESIGN

### 4.1 GUI Design

#### 4.1.1 User Login And Registration

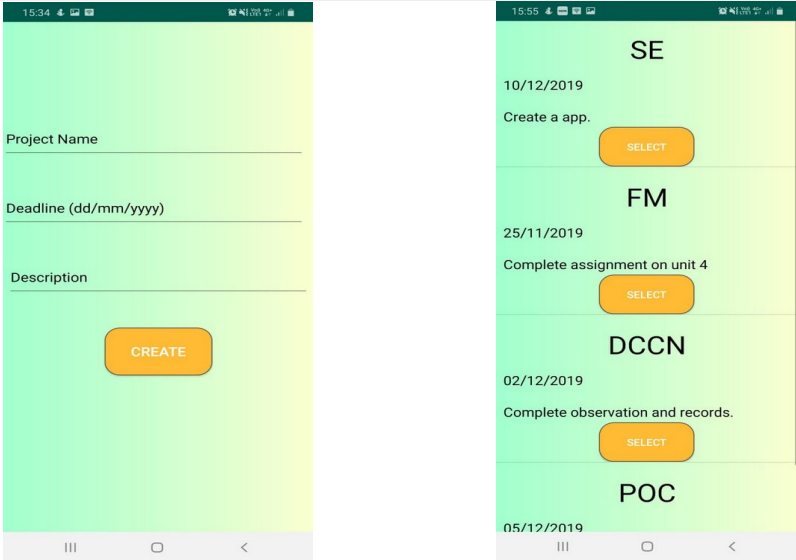
Table 4.1.1 : User Login And Registration

<b>ID</b>	UI501		
<b>Author:</b>	E. VINAY KUMAR	<b>Date</b>	25-08-2019
<b>Description</b>	<p>Sign Up Interface contain two radio buttons and three edit texts(Text boxes).User have to enter his name and has select appropriate radio button to sign up either as student or faculty and have to enter valid email and password.</p> <p>Login interface contains two edit texts(Text boxes). User have to enter valid email and password which were entered during Sign up.</p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div>		
<b>Exception Handling</b>	<p>If user misses to enter details in any one of the edit text(Text box), then system will stop process and will give a message to check entered details.</p> <p>If user misses to select radio button then system will stop process and will give a message to select appropriate radio button.</p>		




## 4.1.2 Projects

Table 4.1.2 : Projects

<b>ID</b>	UI517		
<b>Author:</b>	N. Sharat Chandra	<b>Date</b>	25-08-2019
<b>Description</b>	<p>Staff can create projects by selecting create project button and he has to enter project name, deadline for the project and project description and click on create button to create it.</p> <p>Student can select the projects by selecting select project button and he can see the different projects available. He can then select any one project by clicking select button.</p>		
			
<b>Exception Handling</b>	If the staff doesn't enter project name, deadline or description correctly then he will prompted with a error message to enter details correctly.		

### 4.1.3 Groups

Table 4.1.3 Groups

ID	UI51J		
Author:	Siddam Manish Sagar	Date	25-08-2019
Description	<p>Interface consists of one button and an image over button. Button are named as View Groups. On clicking on generate group button, groups are created.</p> <p>On clicking on View Groups button the groups that are formed based on Project or course selected. Groups are viewed on screen/user interface.</p>		
<div></div>			
Exception Handling	If after grouping of students one student is left out, then that student will added to some other group doing the same project.		

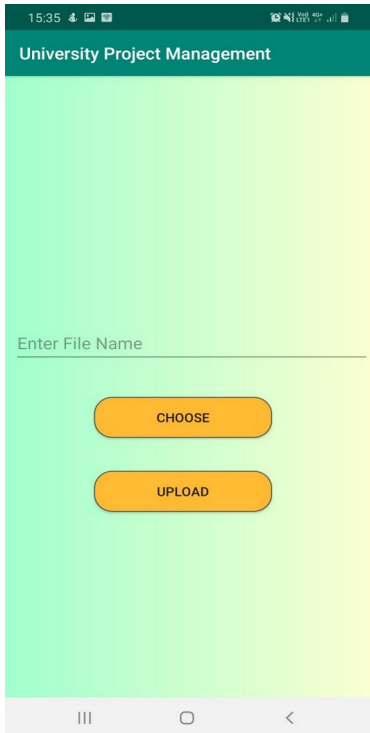
#### 4.1.4 Add/Delete Student

Table 4.1.4:Add/Delete Student

ID	UI51K		
Author:	Sidharth Kumar	Date	25-08-2019
Description	<p>After opening the app the faculty will have two buttons either add or delete students from their respective projects.</p> <p>After clicking on the add button the faculty will have to enter the project and student names and then click on the add button to add the students to respective project.</p> <p>After clicking on the delete button the faculty will have to enter the project and student names and then click on the delete button to delete the students from their respective project.</p>		
<div><div><div>15:34</div><div>University Project Management</div><div>Enter student name</div><div>Enter project</div><div>ADD</div></div><div><div>15:35</div><div>University Project Management</div><div>Enter student name</div><div>Enter project name</div><div>DELETE</div></div></div>			
Exception Handling	<p>If partial data is provided then the system asks user to enter the details.</p> <p>If the wrong spelling of the project or student is provided then the system gives an alert message.</p>		

#### 4.1.5 Uploading Documents

Table 4.1.5 : Uploading Documents

ID	UI504		
Author:	K. Suneeth	Date	25-08-2019
Description	<p>Activity contains a choose and upload buttons in it, which is connected to database(Firebase).</p> <p>Choose button makes user to choose file from his/her device file manager.</p> <p>Upload button uploads selected file to Firebase.</p>		
<div></div>			
Exception Handling	If the Firebase file storage is exceeded then a error message is displayed saying unable to upload selected file.		

## 5. IMPLEMENTATION AND TESTING

### 5.1 Test Case Design

#### 5.1.1 Login Test Case Design

Table 5.1.1: Login Test Case Design

<b>Name of Student:</b>	<b>E. VINAY KUMAR</b>		<b>Date Completed:</b>	13 OCTOBER 2019
<b>Use Case ID#</b>	<b>Test Case ID#</b>	<b>Preconditions</b>	<b>Input</b>	<b>Expected output</b>
UC501-User Registration	TID1	User must have Valid Email and other credentials.	User enters required Information and Presses sign up button.	The System should display a dialog box saying "Registration successful" and user is redirected to Login Page.
UC501-User Registration	TID2	User must have Valid Email and other credentials.	User enters any invalid Data while registration.	The System should give Message Saying "Registration Error".
UC501-User Login	TID3	The user is already authenticated.	User enters email, password and clicks on login button.	The system should take the user to correct page/respective activity.
UC501-User Login	TID4	The user is already authenticated.	User enters invalid email/password and clicks on login button.	The system should give message Saying "Login Error".
UC501-Login	TID5	The user is already authenticated and is on login activity.	User enters valid email, password and clicks on login button. But there is no internet connection.	The System gives login error.

### 5.1.2 Create and Select Projects Test Case Design

Table 5.1.2 : Create and Select Projects Test Case Design

<b>Name of Student:</b>	<b>N. Sharat Chandra</b>		<b>Date Completed:</b>	13-10-2019
<b>Use Case ID#</b>	<b>Test Case ID#</b>	<b>Preconditions</b>	<b>Input</b>	<b>Expected output</b>
UC517 - Select Projects	TID1	The Student can select a project from the given projects list.	Student clicks on select button to select a project.	Student is given a confirmation box asking if he wants to select the project.
UC517 - Create Projects	TID2	The faculty can create a project.	Faculty has to enter the details of the project.	Project details is added to database and the user redirected to faculty main activity.
UC517 - Create Projects	TID3	The faculty has to enter the details in the correct format.	Faculty has to enter the details of the project.	An error message "Check the project details " is displayed.

### 5.1.3 Grouping Test Case Design

Table 5.1.3 : Grouping Test Case Design

<b>Name of Student:</b>	<b>Siddam Manish Sagar</b>		<b>Date Completed:</b>	13-10-2019
<b>Use Case ID#</b>	<b>Test Case ID#</b>	<b>Preconditions</b>	<b>Input</b>	<b>Expected output</b>
UC51J-Generate groups	TID1	Students who have selected same projects are grouped. A group of two are made.	Automatically generated.	Groups have been generated.
UC51J-View Groups	TID2	Groups have been done in order of selecting groups.	On clicking View groups button.	Group details like group number, members of a group, project name are displayed.
UC51J-Odd Groups	TID3	If one student is left out after creating groups.	On clicking generate groups.	The student is added to any group, an exceptional case of generation of groups.

#### 5.1.4 Add/Delete Student Test Case Design

Table: Add/Delete Student Test Case Design

<b>Name of Student:</b>	<b>Sidharth Kumar</b>		<b>Date Completed:</b>	14/10/2019
<b>Use Case ID#</b>	<b>Test Case ID#</b>	<b>Preconditions</b>	<b>Input</b>	<b>Expected output</b>
UC51K-Addition of students into a project	TID1	Users must enter the correct details of student and project name.	The user enters the required information and click on the add button.	The data should be entered into the database and toast is displayed saying "Student Added".
UC51K-Deletion of students from a project	TID2	User must enter the correct details of the student and project name.	The user enters the required information and clicks on the delete button.	The data should be removed from the database and a toast is displayed saying "Student Deleted".
UC51K-Deletion of students from a project	TID3	The name of the project and student entered must be present in the database.	The user enters the required information and clicks on the delete button.	A dialog box must be displayed saying that "Invalid Data Entered".

#### 5.1.5 Upload Documents Test Case Design

Table 5.1.5 : Upload Documents Test Case Design

<b>Name of Student:</b>	<b>K. Suneeth</b>		<b>Date Completed:</b>	13-10-19
<b>Use Case ID#</b>	<b>Test Case ID#</b>	<b>Preconditions</b>	<b>Input</b>	<b>Expected output</b>
UC504	TID1	Student must enter group number as there file name.	Student enter the file name in edit text(Text box).	File name is entered.
UC504	TID2	Proper internet connection is required.	Student clicks on choose button.	Device file manager is opened and required file can be selected.
UC504	TID3	Proper internet connection is required.	Student clicks on choose button.	The selected file is uploaded to database.

## 5.2 Implementation

- The University app is designed for the project management for faculty and student.
- It helps the faculties and students to keep record of there work and progress of their respective projects.
- This app can record the progress and information of both students and faculty at the same time in an integrated way.
- When app is opened initially the user as to login by entering his/her email and password. In case if the user is not registered he should click on the “Click here to Register” link. After this the user will redirected to user registration activity where he has to enter his/her details.

This app contains two type of login as this app handle two types of use

- i. Faculty
- ii. Student

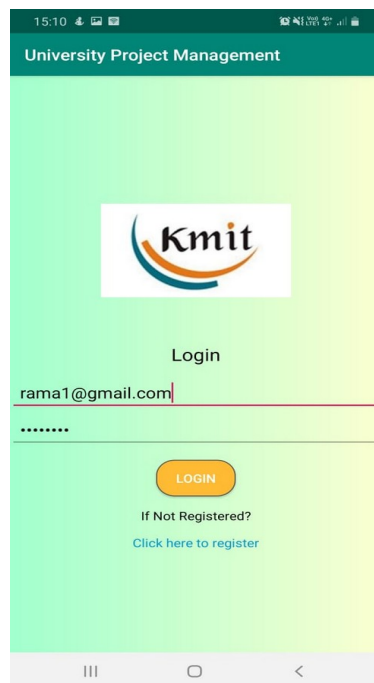


Figure 5.2.1 Login

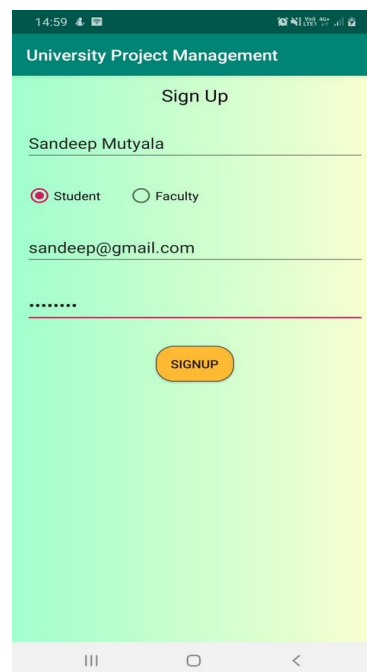


Figure 5.2.2 User Registration



### **Faculty :**

Faculty has to provide the required information to login into the account. Then faculty may click on any of the following four buttons

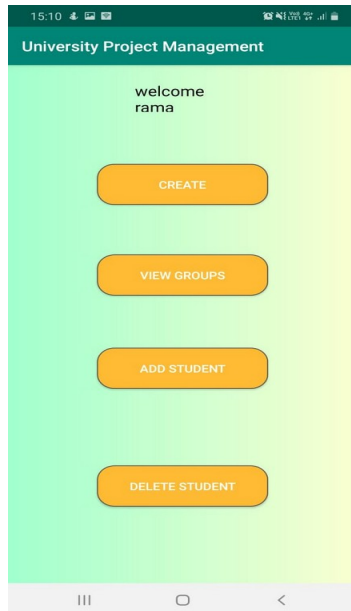


Figure 5.2.3 Faculty Main Activity

**1.Create:** By clicking on create, the faculty will be directed to page where he can create projects by providing the information required(refer the below figure).

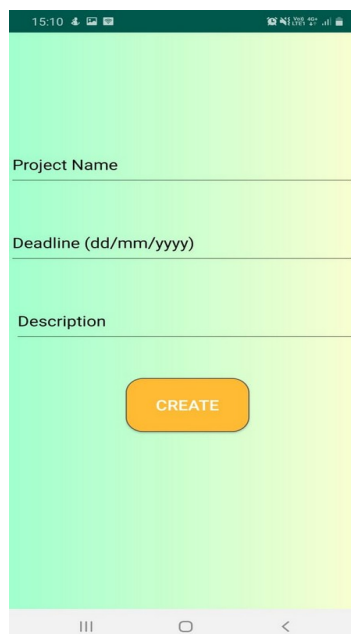


Figure 5.2.4 Create Projects Activity

**2. View group:** Faculty can view the automatically generated groups along with their details.



Figure 5.2.5 View Groups Activity

**3. Add Student:** Faculty can add student into any database. By providing the name of the student and the project name.

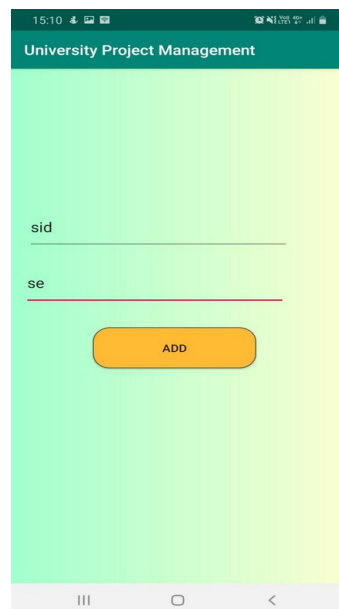
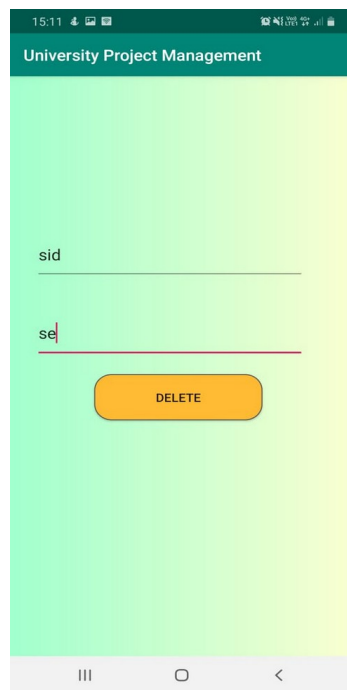


Figure 5.2.6 Add Student Activity

**4. Delete Student:** Faculty can delete students from the database. By providing the name of the student and the project.



The screenshot displays a mobile application interface for 'University Project Management'. At the top, a dark green header bar contains the app's name. The main area has a light green-to-yellow gradient background. Two text input fields are present: the first is labeled 'sid' and the second is labeled 'se'. Both fields have a red horizontal line underneath them, indicating a validation error. Below these fields is a prominent orange button with the text 'DELETE' in black capital letters. The bottom of the screen shows the standard Android navigation bar with icons for the app drawer, home, and back.

Figure 5.2.7 Delete Student Activity

### **Student:**

Student has to provide the required information to login into the account. Then student will have three options to choose from

**1. Select:** Students can select any projects from the project list created by the faculty.

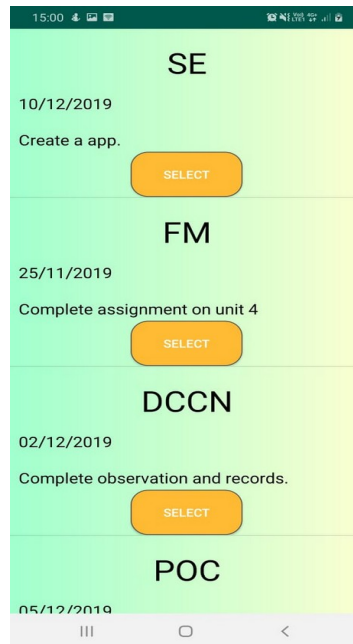


Figure 5.2.8 Select Project Activity

**2. View Groups:** By clicking on the view group button a student can view the automatically generated groups.



Figure 5.2.9 View Groups Activity

**3. Uploading files:** By clicking on the upload button the student can upload the files or documents which he has done for the project.



Figure 5.2.10 Upload File Activity

## 6. CONCLUSION AND FUTURE WORK

We have developed an application for University Project Management using agile software development process. Our technology stack includes of Android Studio and Firebase. The application consists of a the different features for different types of users (faculty and students)

### **Faculty:**

- Login/Registration.
- Creation of projects.
- List of groups generated along with student names and group numbers.
- Adding / Deletion of students from any project.

### **Student:**

- Login/Registration.
- Selection of projects.
- List of groups generated along with student names and group numbers.
- Uploading of files to database.

In the future we intend to make following changes to our application

- Use a more secure database to store user info and files.
- A feature for students to know about any updates on the project they have selected.
- A feature for faculty to know about the progress of project of each group.

## Appendix A: Requirement Traceability

User Requirement IDs		REQCF1	REQCF2	REQCF3	REQCF4	REQCF5
User Requirement Description		Login	Projects	Groups	Add/delete	Upload
UC501	Login	Yes				
UC517	Projects		Yes			
UC51J	Groups			Yes		
UC51K	Add/delete				Yes	
UC504LE	Upload					Yes

## Appendix B: Design Traceability

CRC IDs		CRC501	CRC517	CRC51J	CRC51K	CRC504LE
Class Name		Login	Project Create	View Groups	Add Student	Upload File
UC501	Login	Yes				
UC517	Projects		yes			
UC51J	Groups			yes		
UC51K	Add/Delete				yes	
UC504LE	Upload					yes



## Appendix C: Data Dictionary

- Email: This attribute is used to Login as Student/Faculty
- Password: This is created by User.
- studentid/facultyid: This is given to every Student or Faculty.
- studentName/facultyName: This is name of Student/Faculty.
- FullName: This is Full name of Student/Faculty
- Project Name: Name of project to be created.
- Deadline: Date on which project should be submitted.
- Description: Project description telling what one must do in the project.
- GroupName: Name of the group according to their order of respective project.
- ProjectName: Name of the project that they are selected.
- StudentName: Name of the student of a particular group.
- Student and project name: This is given by the faculty in the label provided.
- Unique ID: It is generated by the Firebase when a student is added to a project.
- File name: Naming the file of the project you have done.

## Appendix D: Test Log

### Login Test Case Design:

Table 1: Login Test Case Design

<b>Name of Student:</b>	<b>E. Vinay Kumar</b>		<b>Date of Testing:</b>			
<b>Use Case ID#</b>	<b>Test Case ID#</b>	<b>Preconditions</b>	<b>Input</b>	<b>Expected output</b>	<b>Actual Output</b>	<b>Pass/Fail</b>
UC501-User Registration	TID1	User must have Valid Email and other credentials.	User enters required Information and Presses sign up button.	The System should display a dialog box saying "Registration successful" and user is redirected to Login Page.	The System should display a dialog box saying "Registration successful" and user is redirected to Login Page.	Pass
UC501-User Registration	TID2	User must have Valid Email and other credentials.	User enters any invalid Data while registration.	The System should give Message Saying "Registration Error".	The System should give Message Saying "Registration Error".	Pass
UC501-User Login	TID3	The user is already authenticated.	User enters email, password and clicks on login button.	The system should take the user to correct page/respective activity.	The system should take the user to correct page/respective activity.	Pass
UC501-User Login	TID4	The user is already authenticated.	User enters invalid email/password and clicks on login button.	The system should give message Saying "Login Error".	The system should give message Saying "Login Error".	Pass
UC501-Login	TID5	The user is already authenticated and is on login activity.	User enters valid email, password and clicks on login button. But there is no	The System gives login error.	The System gives login error.	Pass

			internet connection.			
--	--	--	----------------------	--	--	--

### Create and Select Projects Test Case Design:

Table 2: Create and Select Projects Test Case Design

<b>Name of Student:</b>	<b>N. Sharat Chandra</b>		<b>Date of Testing:</b>			
<b>Use Case ID#</b>	<b>Test Case ID#</b>	<b>Preconditions</b>	<b>Input</b>	<b>Expected output</b>	<b>Actual Output</b>	<b>Pass/Fail</b>
UC517 - Select Projects	TID1	The Student can select a project from the given projects list.	Student clicks on select button to select a project.	Student is given a confirmation box asking if he wants to select the project.	Student is given a confirmation box asking if he wants to select the project.	Pass
UC517 - Create Projects	TID2	The faculty can create a project.	Faculty has to enter the details of the project.	Project details is added to database and the user is redirected to faculty main activity.	Project details is added to database and the user is redirected to faculty main activity.	Pass
UC517 - Create Projects	TID3	The faculty has to enter the details in the correct format.	Faculty has to enter the details of the project.	An error message "Check the project details " is displayed.	An error message "Check the project details " is displayed.	Pass

### Grouping Test Case Design:

Table 3: Grouping Test Case Design

<b>Name of Student:</b>	<b>Siddam Manish Sagar</b>		<b>Date of Testing:</b>			
<b>Use Case ID#</b>	<b>Test Case ID#</b>	<b>Preconditions</b>	<b>Input</b>	<b>Expected output</b>	<b>Actual Output</b>	<b>Pass/Fail</b>
UC51J-Generate groups	TID1	Students who have selected same projects are grouped. A group of two are made.	Automatically generated.	Groups have been generated.	Groups have been generated.	Pass
UC51J-View Groups	TID2	Groups have been done in order of selecting groups.	On clicking View groups button.	Group details like group number, members of a group, project name are displayed.	Group details like group number, members of a group, project name are displayed.	Pass
UC51J-Odd Groups	TID3	If one student is left out after creating groups.	On clicking generate groups.	The student is added to any group, an exceptional case of generation of groups.	The student is added to any group, an exceptional case of generation of groups.	Pass

**Add/Delete Student Test Case Design:**

Table 4: Add/Delete Student Test Case Design

<b>Name of Student:</b>	<b>Sidharth Kumar</b>		<b>Date of Testing:</b>			
<b>Use Case ID#</b>	<b>Test Case ID#</b>	<b>Preconditions</b>	<b>Input</b>	<b>Expected output</b>	<b>Actual Output</b>	<b>Pass/Fail</b>
UC51K-Addition of students into a project	TID1	Users must enter the correct details of student and project name.	The user enters the required information and click on the add button.	The data should be entered into the database and toast is displayed saying "Student Added".	The data should be entered into the database and toast is displayed saying "Student Added".	Pass
UC51K-Deletion of students from a project	TID2	User must enter the correct details of the student and project name.	The user enters the required information and clicks on the delete button.	The data should be removed from the database and a toast is displayed saying "Student Deleted".	The data should be removed from the database and a toast is displayed saying "Student Deleted".	Pass
UC51K-Deletion of students from a project	TID3	The name of the project and student entered must be present in the database.	The user enters the required information and clicks on the delete button.	A dialog box must be displayed saying that "Invalid Data Entered".	A dialog box must be displayed saying that "Invalid Data Entered".	Pass

### Upload Documents Test Case Design:

Table 5: Upload Documents Test Case Design

<b>Name of Student:</b>	<b>K. Suneeth</b>		<b>Date of Testing:</b>			
<b>Use Case ID#</b>	<b>Test Case ID#</b>	<b>Preconditions</b>	<b>Input</b>	<b>Expected output</b>	<b>Actual Output</b>	<b>Pass/Fail</b>
UC504	TID1	Student must enter group number as there file name.	Student enter the file name in edit text(Text box).	File name is entered.	File name is entered.	Pass
UC504	TID2	Proper internet connection is required.	Student clicks on choose button.	Device file manager is opened and required file can be selected.	Device file manager is opened and required file can be selected.	Pass
UC504	TID3	Proper internet connection is required.	Student clicks on choose button.	The selected file is uploaded to database.	The selected file is uploaded to database.	Pass