



---

000959097 \_SAKAOATH HOSSIN  
TANVIR\_EWSD\_APRIl 2017

---

COMP 1640



SUBMITTED TO:

Dr. Ray Stoneham  
Greenwich Coordinator

SUBMITTED BY:

Sakaoath Hossain Tanvir  
ID: 000959097

SUBMISSION DATE: 13<sup>th</sup> April, 2017

Team Name: Ironclad

Team Members:

Greenwich ID	Name	Role
000990161	Ariful Islam	System Analyst
000959097	Sakaoath Hossain Tanvir	Database Designer
000959089	Jannatul Bakee	System Designer
000959088	Moshiur Rahman	Programmer
000959095	Rayhan Kabir	Tester

Click here to add more team members.

Credentials:

Role	User (email address)	Password
Admin		
Manager		
Teacher		
Student		

## Contents

<b>Introduction</b> .....	3
<b>Objectives</b> .....	3
<b>Assumption</b> .....	4
<b>Evaluation of product and process</b> .....	5
<b>Further improvement</b> .....	6
Evaluation of team.....	7
Self-Evaluation .....	8
Lesson Learnt .....	9
Conclusion.....	10
<b>Appendix B: Database Design</b> .....	12
<b>Conceptual Data Model</b> .....	12
<b>Mapping</b> .....	13
<b>Extended Entity Relationship Diagram</b> .....	14
<b>Data dictionary</b> .....	15
<b>Implementation of data integrity constraints</b> .....	18
Creating the tables.....	18
Creating index .....	21
Creating Triggers.....	21
Making View.....	21
<b>Project Planning</b> .....	22
<b>User Stories</b> .....	22
<b>Minutes of meeting</b> .....	31
<b>Sprint Backlog</b> .....	38
<b>Product Backlog</b> .....	40
<b>Burn down chart</b> .....	42

## Introduction

The whole project is set of tasks done by the whole group. The system will be developed as web based solution. There will be sets of roles in the system to monitor and to keep the system functional. So all the features will be there but based on roll users will participate on them.

## Objectives

Our main objective for the project is to develop web based EC claim system and the objectives to achieve it follows:

- Creating role based login system.
- Administrator will maintain all the assessments. Adding, updating, deleting etc.
- Student will be able to claim for their extenuating circumstances. They will be able to submit their proof/s.
- Once submitted student won't be able to modify anything.
- Notification will be sent to respective EC coordinator upon claiming an EC.
- EC coordinator will be able to accept or deny upon receiving an EC claim.
- Notification will be sent to the student upon receiving the result of claimed EC.
- EC manager can monitor all the previous and running, closed and opened EC claims. Admin can do these activities too.
- Multiple types of reports will be available to the users based on their roles.
- The user interface must be platform independent.

Some of these features will be implemented by the followed assumption.

## Assumption

- ❖ The message notification will alert the EC coordinators, students to check their EC claims.
- ❖ Students will be able to upload more than one evidence to support their EC claim.
- ❖ Once an EC claim has been sent it has to be processed within 14 days.
- ❖ All types of users are already in main database. So EC administrator won't have access to those data so can't update, delete existing user nor can delete them.
- ❖ All the faculties, departments and subjects are in university central database. So data will be fetched for information but EC administrator won't be able to add or update or delete them.
- ❖ All the assessments will be added by administrator but administrator won't be able to delete assessment once it's there. Because deletion of existing assessments will cause integrity of EC claims and their results.
- ❖ EC claim can be printed out.

## Evaluation of product and process

Our goal is to make a web based system for claiming extenuating circumstances. The goal is to take data from students in case he/she has to apply for extenuating circumstances. After thorough team meeting and analysis, we have found all the requirements that has been asked by the university and to fulfill the system completeness. It took several meetings with group to assign individual role and their tasks and later analyzing the tasks within group. The tasks were also prioritized into a list to sort out which functionalities are to deliver first to meet customer's priority. Also apart from customer's demand a MoSCoW prioritization list was made. After several meetings and thorough analysis, a list was made evaluating from both point of view.

These are the requirements that the team has found out to be done immediately in order make the system up and running completely:

- Administrator can add assessments for the existing subjects.
- Student will claim for extenuating circumstances of their eligible assessments. Student will also be able to submit proof for their EC.
- EC coordinator will be notified upon receiving an EC.
- EC coordinator can check the whole claim clearly, give decision and make remarks if willing.
- Alert for processing an EC within 14 days.
- Student will be notified upon having a decision of his/her EC claim.
- EC manager will be able to view the EC processing and processed claims.

After meeting all the "must have" requirements, others requirements were implemented on the system. Not all the requirements that have been implemented was asked in the customer requirement. There were few requirements that were brought meetings of analysis of future system. Assumed features were added to the requirement list to make the system user friendly and futuristic. The delivery of the system is rather soon so some of the less important features were not implemented but within more time will be brought on to the system in near future. At the end, it is quiet easy to say our system has met more than customer has demanded.

In the appendix area of project respective group member will mention the functional requirements and non-functional requirements.

## **System Strength**

- The system is robust.
- User access level is strictly maintained due to use of latest framework.
- Any kind of hacking, malpractice, threats is impossible. Encryption were brought to prevent these while making the system friendlier to the users in terms of password and URLs.
- Implementation of latest framework of PHP Laravel 5.2.
- The system is platform independent.
- Internal notification between EC coordinator and students.
- Simple reporting system for the users. Very easy to understand, no complex chart or graph were brought to mess it up.
- The database design is very futuristic.

## **System Weakness**

- No password retrieval system.
- Email for notifying the users.
- Administrator dependency on adding assessments.

## **Further improvement**

With the iterative approach of development there will be more development of the system by which few more functionalities will be introduced. With that the system would be more resourceful and more robust.

- Two step login could be introduced to make system more secured.
- A repetitive reminder for the EC coordinator to make him/her more efficient to process the EC claim more effectively.
- Password retrieval system through phone or via email.
- Live chatting can be introduced.
- Designing the system more.

# Evaluation of team

The team “Ironclad” was consisted a number of five members at the beginning. The assigned role was a system analyst, a database designer, a system designer, a developer and a tester. As the goal was to have team product so scrum master or product owner were not brought up or assigned. I myself had to take initiation due to inconsistence presence of team members. I had to allocate members their role upon their agreement. All of them were happy with given role and most the team mates did a great job to make the system complete and successful.

The first task was to analyzing the system, finding out customer demands, product demands, ways to implement the demands. It seems to be the task of system analyst but most of the members took part on this phase. Not all of them were every meeting but I personally made sure that the minimum right participants were presented in the meeting to get the task done. After a thorough series of meeting the user stories, the draft of functional requirements and nonfunctional requirements were handed out.

After the whole system is analyzed the part of database designing and site designing begins. For the sake of delivering the project in time it is imperative to do these both tasks simultaneously. This phase mostly involves database designer, system designer and developer. Also partially tester would be needed. But due to continuous absence of web designer, team had to waste time waiting for design. So me and developer had to take steps to make the system done and deliver in given time. The design is quiet ok but not eye catchy. It fulfills all the tasks customer demanded. The database design was rather tricky, we had to change many approaches and after 2-3 conceptual designs the final and logical design was implemented into physical database design.

The development was best. Due to thorough analysis before development the goal was quiet clear about what to make and how to achieve. Also our developer had vast experience in web developing. We have implemented latest and strongest PHP framework. It didn't take too long to make the system due to team effort of design and thorough testing during development. And after development most testing was done to make sure the system is secured and fool proof.

In order to achieve a successful team work, team must comprehend with right skills. There are certain criteria to evaluate a team. They are following:

- ✓ Availability
- ✓ Contribution on key requirements
- ✓ Technical skills
- ✓ Expertise in respective area
- ✓ Communication skills
- ✓ Confidence
- ✓ Team collaboration

I have evaluated my team members based on above criteria in scale of 10. The chart is drawn below.

Name	Tanvir	Ariful	Moshiur	Rayhan	Jannatul
Availability	9	8	8	8	0
Contribution on key requirements	10	7	10	8	0
Technical skills	8	6	9	7	2
Expertise in respective area	9	7	10	9	2
Communication skills	10	7	8	7	3
Confidence	8	8	8	8	10
Team Collaboration	8	8	8	8	0
Total (out of 70)	62	51	61	55	17

## Self-Evaluation

Although my role was to design the system but I had to also take the responsibility of SCRUM master. Therefore, I had to take partial or major part in most of the areas except for coding. I myself had to organize all of the meetings, assign tasks to the team mates. Through these meetings I had to motivate all the team mates about developing the best solution with given time budget. Though the system analyst was needed in the life cycle of development but I had to do most of his tasks not because of his availability but because of his lack expertise in respective area. Also I had to explain the outcome and expectation to the team members as they were not present in primary classes of user requirement.

I have tried several times to contact with the UI designer but due to his “personal issues” he wasn’t there. I myself had to hand draw and say orally which to be implemented and how to be implemented.

As for my part, it was database design. Little as it sounds but it is the most crucial task of anything. Because the data is the only part that make the system running, make the system most resourceful. So a well-established database design is mandatory to make the most of the system in future too. There might be only one final outcome of database, but it was gained through several conceptual database design, mapping and relational schemas. The end design AKA physical database design meets all the requirements that we have hoped for and will be helpful in future development.

Though I gave most time in the project, I cannot deny their participation in the project and meetings. I may have fueled the engine but they were there to run the whole organism run. I can say I have tried my best not only to make myself best but to make my team best and gain best solution.

In the demonstration part, I have presented the system to the customers and other people. I have presented all the functions in the presentation part and video. After presentation, we as a team answered all the questions asked by other team members and customers. The presentation was fruitful and no weakness or lacking were to be found.

## Lesson Learnt

Lifetime of this project has taught me several things from my team members and my course teacher. I have drawn a list below with only the significant subjects that had helped me throughout the project and I’ll be in great debt in future for this lesson that I’ve been taught.

- ✓ Scrum methodology and its effectiveness. Daily online meeting, weekly meetings, feedback on the solution, update of the solution has helped all of us to detect project status and become aware of all types of threats.
- ✓ Product backlog, sprint backlog, burn down chart has helped up use resource and time efficiently and plan accordingly.

- ✓ Working in a collaborative manner has speeded up all of our work and helped us to each other's knowledge by sharing.
- ✓ Sharing resource amongst each other.

## Conclusion

The development of the system was challenging as the scenario has been taken from real life scenario. Steering through the system has helped us the knowledge about this new types of problems. Different types of tools as SCRUM methodology, analyzing tools, development framework, designing tools was implemented. For us to sort out all the key requirement from the customer was a great learning experience.

The task was to make a system for a university. It was not a full system rather partial system that would be added with the existing system making that system more useful and resourceful. Students would be able to submit their claim for extenuating circumstances and course coordinator will simply say yes or no. Simple as it sounds but there were quite a lot of tasks that needed to be completed to make the system healthy and robust.

List below show all the requirements that has been done in order to establish the system successful:

- Four types of users were implemented.
- Admin can add assessments.
- Students belong to a faculty.
- Each faculty has a course coordinator.
- Course coordinator accepts or rejects EC claim.
- Students claims for EC against an assessment.
- Students can upload one or more than one evidence to support their claim.
- Course coordinator has to finish a claim within 14 days.
- A mail system for course coordinator and students to interact.
- Downloading system for course coordinator.
- All existing EC claim to be viewed.
- EC manager can view all the EC claims.

- Admin can view all the EC claims.
- Custom report generating system based on existing EC.
- Custom report generating system based on user role.
- A platform independent system.

The list above shows that the system built was a considerable successful system due to fulfilling all the key requirements and more than that. We won't be arrogant enough to say that this system has no further development but within given time we have done our best and with more time we will be able to make the system more fulfilling and complete.

# Appendix B: Database Design

## Conceptual Data Model

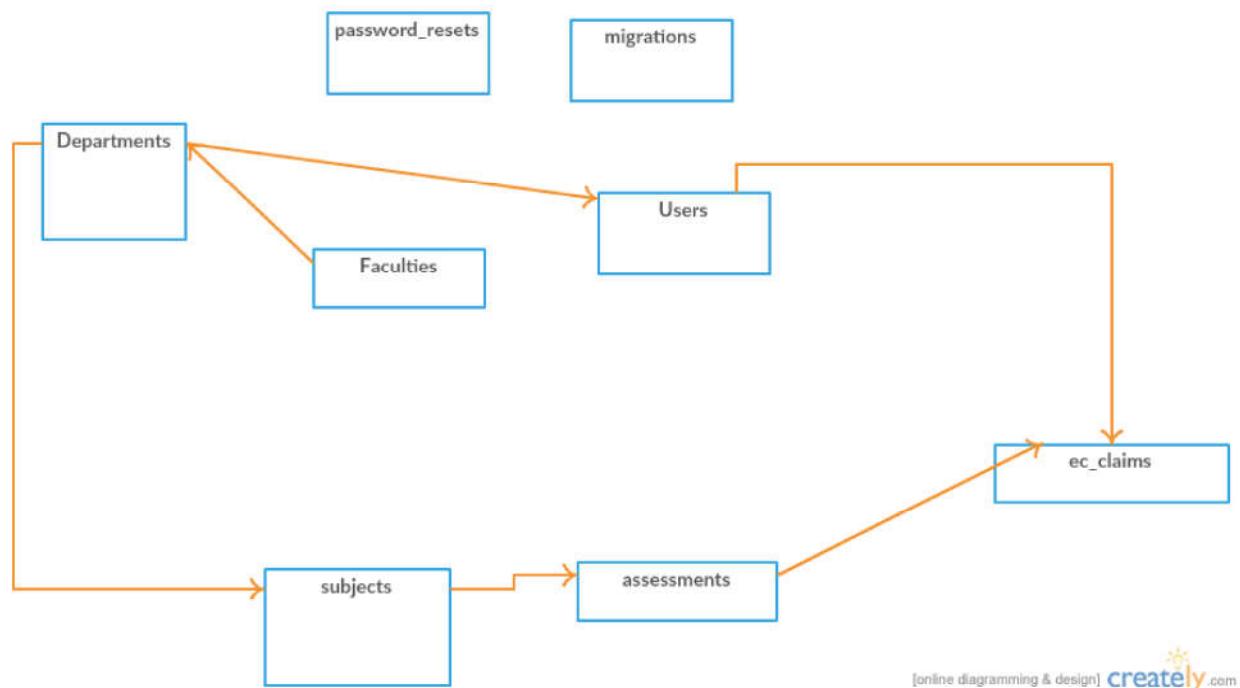


Fig: Conceptual data model

## Mapping

a. faculties

<u>id</u>	<u>name</u>	<u>created_at</u>	<u>updated_at</u>
-----------	-------------	-------------------	-------------------

b. departments

<u>id</u>	<u>name</u>	<u>created_at</u>	<u>updated_at</u>	<u>faculty_id</u>
-----------	-------------	-------------------	-------------------	-------------------

c. subjects

<u>id</u>	<u>name</u>	<u>created_at</u>	<u>updated_at</u>	<u>department_id</u>
-----------	-------------	-------------------	-------------------	----------------------

d. assessments

<u>id</u>	<u>name</u>	<u>deadline</u>	<u>created_at</u>	<u>updated_at</u>	<u>subject_id</u>
-----------	-------------	-----------------	-------------------	-------------------	-------------------

e. users

<u>I</u>	<u>Name</u>	<u>Gender</u>	<u>Email</u>	<u>Password</u>	<u>Date of Birth</u>	<u>Role</u>	<u>Faculty_id</u>	<u>Department_id</u>	<u>Remember token</u>	<u>created_at</u>	<u>updated_at</u>
----------	-------------	---------------	--------------	-----------------	----------------------	-------------	-------------------	----------------------	-----------------------	-------------------	-------------------

f. ec\_claims

<u>id</u>	<u>name</u>	<u>evidence_01</u>	<u>evidence_02</u>	<u>evidence_03</u>	<u>claim_criteria</u>	<u>status</u>	<u>assessment_id</u>	<u>user_id</u>
-----------	-------------	--------------------	--------------------	--------------------	-----------------------	---------------	----------------------	----------------

g. migrations

<u>id</u>	<u>migration</u>	<u>batch</u>
-----------	------------------	--------------

h. password\_resets

<u>id</u>	<u>created_at</u>
-----------	-------------------

Fig: ERD to mapping

## Extended Entity Relationship Diagram

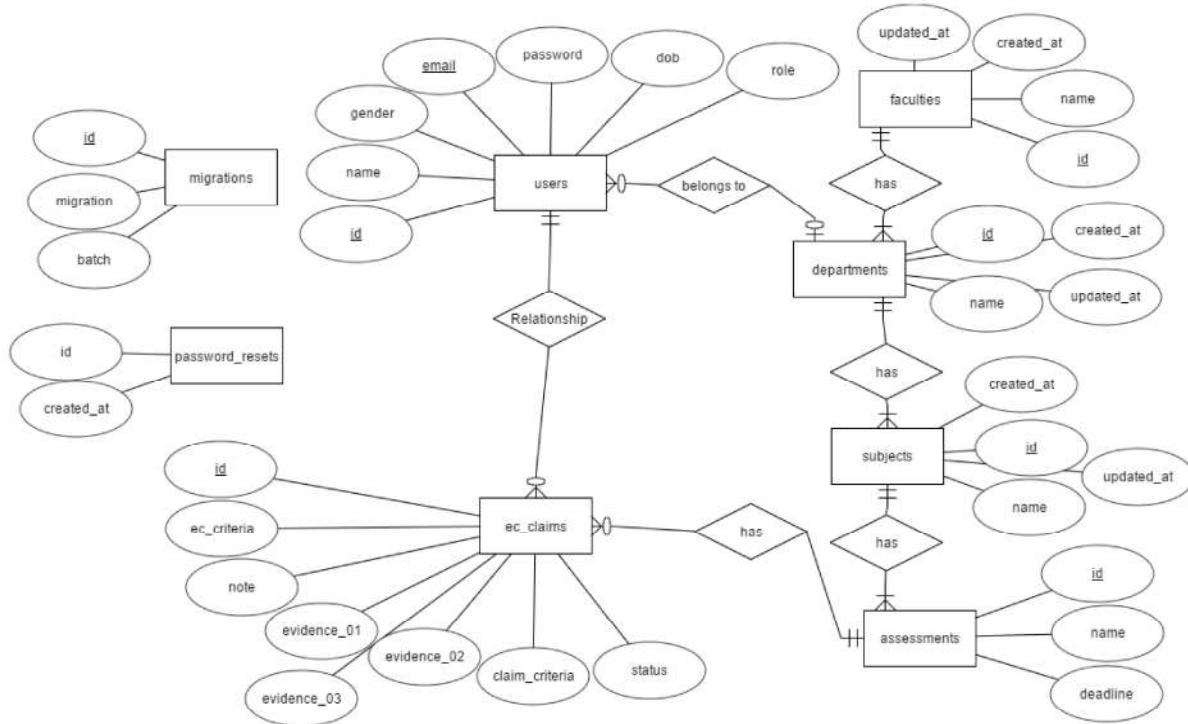


Fig: Extended Entity Relationship Diagram

## Data dictionary

### ❖ Faculties

Attribute Name	Data type	Data length	Primary key	Foreign key	Reference table and column	Extra
<b>id</b>	int	10	yes			auto_increment
<b>name</b>	varchar	255				

### ❖ Departments

Attribute Name	Data type	Data length	Primary key	Foreign key	Reference table and column	Extra
<b>id</b>	int	10	yes			auto_increment
<b>name</b>	varchar	255				
<b>faculty_id</b>	int	10		yes	faculty id	
<b>created_at</b>	timestamp					
<b>updated_at</b>	timestamp					

### ❖ Subjects

Attribute Name	Data type	Data length	Primary key	Foreign key	Reference table and column	Extra
<b>id</b>	int	10	yes			auto_increments
<b>name</b>	varchar	255				
<b>department_id</b>	int	10		yes	department id	
<b>created_at</b>	timestamp					
<b>updated_at</b>	timestamp					

## -Assessments

Attribute Name	Data type	Data length	Primary key	Foreign key	Reference table and column	Extra
<b>id</b>	int	10	yes			auto_increment
<b>name</b>	varchar	255				
<b>deadline</b>	date					
<b>subject_id</b>	int	10		yes	subject id	
<b>created_at</b>	timestamp					
<b>updated_at</b>	timestamp					

## -Users

Attribute Name	Data type	Data length	Primary key	Foreign key	Reference table and column	Extra
<b>id</b>	int	10	yes			auto_increment
<b>name</b>	varchar	255				
<b>email</b>	varchar	255				
<b>password</b>	varchar	255				
<b>role</b>	varchar	255				
<b>faculty_id</b>	int	10		yes	faculty id	
<b>department_id</b>	int	10		yes	department id	
<b>dob</b>	date					
<b>gender</b>	varchar	255				
<b>remember_token</b>	varchar	100				
<b>created_at</b>	timestamp					
<b>updated_at</b>	timestamp					

## Ec\_claims

Attribute Name	Data type	Data length	Primary key	Foreign key	Reference table and column	Extra
<b>id</b>	int	10	yes			auto_increment
<b>assessment_id</b>	int	10		yes	assessment id	
<b>user_id</b>	int	10		yes	user id	
<b>claim_criteria</b>	varchar	255				
<b>note</b>	varchar	255				
<b>evidence_01</b>	varchar	255				
<b>evidence_02</b>	varchar	255				
<b>evidence_03</b>	varchar	255				
<b>status</b>	varchar	255				
<b>created_at</b>	timestamp					
<b>updated_at</b>	timestamp					

## Migrations

Attribute Name	Data type	Data length	Primary key	Foreign key	Reference table and column	Extra
<b>id</b>	int	10	yes			auto_increment
<b>migration</b>	varchar	255				
<b>batch</b>	int	11				

## Password\_resets

Attribute Name	Data type	Data length	Primary key	Foreign key	Reference table and column	Extra
<b>id</b>	int	10	yes			auto_increment
<b>created_at</b>	timestamp					
<b>updated_at</b>	timestamp					

## Implementation of data integrity constraints

Creating the tables

```
CREATE TABLE `faculties` (
  `id` int(10) UNSIGNED NOT NULL,
  `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `departments` (
  `id` int(10) UNSIGNED NOT NULL,
  `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `faculty_id` int(11) NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `subjects` (
  `id` int(10) UNSIGNED NOT NULL,
  `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `department_id` int(11) NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `assessments` (
  `id` int(10) UNSIGNED NOT NULL,
  `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `deadline` date NOT NULL,
  `subject_id` int(11) NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `users` (
  `id` int(10) UNSIGNED NOT NULL,
  `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `password` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `role` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `faculty_id` int(11) DEFAULT NULL,
  `department_id` int(11) DEFAULT NULL,
  `dob` date NOT NULL,
  `gender` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `remember_token` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```

CREATE TABLE `ec_claims` (
  `id` int(10) UNSIGNED NOT NULL,
  `assessment_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `claim_criteria` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `note` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `evidence_01` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `evidence_02` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `evidence_03` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `status` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT
  'pending',
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```

CREATE TABLE `migrations` (
  `id` int(10) UNSIGNED NOT NULL,
  `migration` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `batch` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```

CREATE TABLE `password_resets` (
  `email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `token` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

## Creating index

```
CREATE INDEX tIndex ON faculties (id);
```

Now we can view the output anytime by simply writing the code below:

```
Select * FROM id;
```

## Creating Triggers

```
CREATE TRIGGER `dont_update_after_insertion` AFTER UPDATE ON
`departments` FOR EACH ROW BEGIN
    INSERT INTO departments
    SET ACTION="UPDATE",
    name = OLD.name,
    id = OLD.id,
    changedon = NOW();
END
;;
DELIMITER ;
```

## Making View

```
CREATE VIEW extanuating_circumstances_claims
AS
SELECT assessment_id as 'Assessment Id', status, note, user_id as
'Student ID'
FROM
ec_claims;
```

The SQL file can be found in group repository's directory.  
Tanvir(database) folder contains .sql file inside Zipped file.

# Project Planning

## User Stories

User Story	User Login
As a	<i>Registered Existing User</i>
I want	<i>Log in to the system</i>
So that	<i>I can use the system</i>
<b>I N V E S T</b> 	Size: 7 Priority: 10
Acceptance Criteria	
<i>Upon successful login users will be directed to disassociated interface according the role.</i>  <i>In case of bad login, there will be clear indication of where it went wrong.</i>	

Fig: User story 1

User Story	<i>Submission of an EC</i>
As a	<i>Student</i>
I want	<i>to submit an EC</i>
So that	<i>I can claim my EC</i>
<b>I N V E S T</b> 	<p>Size: 8</p> <p>Priority: 10</p>
Acceptance Criteria	
<i>Upon successful login students will be directed to able to claim for an extenuating circumstances</i>	

Fig: User story 2

User Story	Getting notification/mail
As a	<i>Course coordinator and student</i>
I want	<i>to get mail notification</i>
So that	<i>I can be alerted about my pending EC.</i>
<b>I N V E S T</b> 	<p>Size: 9</p> <p>Priority: 10</p>
Acceptance Criteria	
<p><i>Upon successful submission course coordinator will be notified to check the pending EC.</i></p> <p><i>Upon successful result from an EC claim, student will be notified to check his/her pending EC.</i></p>	

Fig: User story 3

<b>User Story</b>	<i>Procession of an Extenuating circumstances.</i>
As a	<i>Course coordinator</i>
I want	<i>to accept or deny pending EC.</i>
So that	<i>I can feedback on submitted EC.</i>
<b>I N V E S T</b> 	<p>Size: 7</p> <p>Priority: 10</p>
<b>Acceptance Criteria</b>	
<p><i>Upon successful login course coordinator will be able see an EC, read it, download the evidence, accept or deny the claim and also make notes upon the claim.</i></p>	

Fig: User story 4

User Story	Managing Users
As an	Administrator
I want	Manage the users
So that	View the people who are using it.
<b>I N V E S T</b>  <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Size: 8  Priority: 6
Acceptance Criteria	
<i>Upon successful login administrator can view the users who have access to the EC claim solution. He/she cannot modify them but only to see.</i>	

Fig: User story 5

User Story	<i>Viewing all existing EC claims</i>
As an	<i>Administrator, EC manager, student and course coordinator</i>
I want	<i>to view existing EC claims.</i>
So that	<i>I can view the EC claims, assessments of those and their results.</i>
<b>I N V E S T</b> 	<p>Size: 9</p> <p>Priority: 8</p>
Acceptance Criteria	
<p><i>Upon successful login administrator, EC manager can view the existing EC claims and important data related to it but not any modification.</i></p> <p><i>Students and course coordinator will only see the claims only related to themselves.</i></p>	

Fig: User story 6

User Story	Dynamic Reports
As an	<i>Administrator, EC manager and course coordinator</i>
I want	<i>to see reports dynamically.</i>
So that	<i>I can see filtered results.</i>
<b>I N V E S T</b> 	<p>Size: 10</p> <p>Priority: 10</p>
Acceptance Criteria	
<p><i>Upon successful login administrator and EC manager can view the existing EC claims based on years, subjects, departments.</i></p>	

Fig: User story 7

<b>User Story</b>	<i>Adding assessments</i>
As an	<i>Administrator</i>
I want	<i>to add assessments</i>
So that	<i>Students can claim EC for the assessments.</i>
<b>I N V E S T</b>  	Size: 8  Priority: 9
<b>Acceptance Criteria</b>	
<p><i>Upon successful login administrator can add assessments only for the existing subjects. Admin also have to put deadline for the assessments.</i></p>	

Fig: User story 8

<b>User Story</b>	<i>14 days notification</i>
As an	<i>Course coordinator</i>
I want	<i>to get notification.</i>
So that	<i>I can finish an EC claim within 14 days.</i>
<b>I N V E S T</b>  <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Size: 6  Priority: 9
<b>Acceptance Criteria</b>	
<i>Course coordinator will get notification again if the time near to process within 14 days.</i>	

Fig: User story 9

## Minutes of meeting

<b>Meeting No: 1</b>	Wrapping up members' role.		
<b>Date of Meeting</b>	February 9 <sup>th</sup> , 2017	<b>Time</b>	04:00 PM-05:30 PM
<b>Location</b>			
<b>Meeting Objectives</b>			
Deliberation of different roles in the amongst the members			
<b>Attendance</b>			
<ul style="list-style-type: none"><li>➤ Sakaoath Hossain Tanvir</li><li>➤ Ariful Islam</li><li>➤ Rayhan Kabir</li><li>➤ Moshiur Rahman</li></ul>			
<b>Meeting Outcome</b>			
In order to finish the system through agile development framework, the role has been handed out to the members of group two.			

<b>Meeting No: 2</b>	Determining the functionality for the system		
<b>Date of Meeting</b>	February 14 <sup>th</sup> , 2017	<b>Time</b>	11:00 AM-12:30 PM
<b>Location</b>			
<b>Meeting Objectives</b>			
Acquiring the number of functionality from the scenario.			
<b>Attendance</b>			

- Sakaoath Hossain Tanvir
- Ariful Islam
- Rayhan Kabir
- Moshiur Rahman

### **Meeting Outcome**

Through a detail depiction of scenario, number of requirement functionality has been determined.

<b>Meeting No: 3</b>	Determining the functionality for the system		
<b>Date of Meeting</b>	February 21 <sup>st</sup> , 2017	<b>Time</b>	03:00 PM-05:30 PM

### **Meeting Objectives**

Acquiring the number of functionality from the scenario.

### **Attendance**

- Sakaoath Hossain Tanvir
- Ariful Islam
- Rayhan Kabir
- Moshiur Rahman

### **Meeting Outcome**

Through a detail depiction of scenario, number of requirement functionality has been determined.

<b>Meeting No: 4</b>	Database design		
<b>Date of Meeting</b>	February 28 <sup>th</sup> , 2017	<b>Time</b>	08:00 AM-02:30 PM
		<b>Location</b>	Tanvir's Home
<b>Meeting Objectives</b>			
Acquiring the database desing			
<b>Attendance</b>			
<ul style="list-style-type: none"> <li>➤ Sakaoath Hossain Tanvir</li> <li>➤ Moshiur Rahman</li> </ul>			
<b>Meeting Outcome</b>			
Through a detail depiction of scenario, database has been found out.			

<b>Meeting No: 5</b>	Determining the website design		
<b>Date of Meeting</b>	March 7 <sup>th</sup> , 2017	<b>Time</b>	04:00 PM-08:30 PM
		<b>Location</b>	DIA
<b>Meeting Objectives</b>			
Acquiring the final design.			
<b>Attendance</b>			
<ul style="list-style-type: none"> <li>➤ Sakaoath Hossain Tanvir</li> <li>➤ Moshiur Rahman</li> </ul>			

**Meeting Outcome**

Through a detail depiction of scenario, number of interfaces required and color of the design has been acquired.

<b>Meeting No: 6</b>	Determining the final database and design		
<b>Date of Meeting</b>	March 10 <sup>th</sup> , 2017	<b>Time</b>	11:00 AM-12:30 PM
<b>Location</b>			
<b>Meeting Objectives</b>			
Acquiring the final database design and website design to be implemented			
<b>Attendance</b>			
<ul style="list-style-type: none"><li>➤ Sakaoath Hossain Tanvir</li><li>➤ Ariful Islam</li><li>➤ Rayhan Kabir</li><li>➤ Moshiur Rahman</li></ul>			
<b>Meeting Outcome</b>			
Through a detail depiction of scenario and multiple scratch of database design and website design, the final design and website layout has been chosen.			

<b>Meeting No: 7</b>	Update of coding		
<b>Date of Meeting</b>	March 14 <sup>th</sup> , 2017	<b>Time</b>	11:00 AM-12:30 PM
		<b>Location</b>	DIA
<b>Meeting Objectives</b>			
Getting update of how the development is going on.			
<b>Attendance</b>			
<ul style="list-style-type: none"> <li>➤ Sakaoath Hossain Tanvir</li> <li>➤ Ariful Islam</li> <li>➤ Rayhan Kabir</li> <li>➤ Moshiur Rahman</li> </ul>			
<b>Meeting Outcome</b>			
The development has been going well, few misconceptions were cleared out. And developer assured that within next 10 days it will be finished.			

<b>Meeting No: 8</b>	Update of development and planning for testing.		
<b>Date of Meeting</b>	March 24 <sup>th</sup> , 2017	<b>Time</b>	10:00 AM-01:30 PM
		<b>Location</b>	Rayhan's Home
<b>Meeting Objectives</b>			
Acquiring the development and planning for testing.			
<b>Attendance</b>			
<ul style="list-style-type: none"> <li>➤ Sakaoath Hossain Tanvir</li> <li>➤ Rayhan Kabir</li> <li>➤ Moshiur Rahman</li> </ul>			
<b>Meeting Outcome</b>			

Most of the development has been done, simultaneous testing has been initiated.

<b>Meeting No: 9</b>	Acquiring final system and update of testing		
<b>Date of Meeting</b>	March 30 <sup>th</sup> , 2017	<b>Time</b>	04:00 PM-06:30 PM
		<b>Location</b>	DIA
<b>Meeting Objectives</b>			
Acquiring the final system and getting update of testing.			
<b>Attendance</b>			
<ul style="list-style-type: none"> <li>➤ Sakaoath Hossain Tanvir</li> <li>➤ Ariful Islam</li> <li>➤ Rayhan Kabir</li> <li>➤ Moshiur Rahman</li> </ul>			
<b>Meeting Outcome</b>			
The final system has been produced. Though simultaneous testing was going on but final testing plan has been taken on.			

<b>Meeting No: 10</b>	Getting final report		
<b>Date of Meeting</b>	April 7 <sup>th</sup> , 2017	<b>Time</b>	02:00 PM-04:30 PM
		<b>Location</b>	DIA
<b>Meeting Objectives</b>			
Acquiring confirmation on final testing.			
<b>Attendance</b>			

- Sakaoath Hossain Tanvir
- Ariful Islam
- Rayhan Kabir
- Moshiur Rahman

### **Meeting Outcome**

Between last meeting, tester and programmer has worked on few minor glitches and fixed it. So the end product is OK.

<b>Meeting No: 11</b>	Evaluating the final build system.		
<b>Date of Meeting</b>	April 10 <sup>th</sup> , 2017	<b>Time</b>	11:00 AM-12:30 PM
		<b>Location</b>	DIA

### **Meeting Objectives**

Evaluating the final built system against expectation.

### **Attendance**

- Sakaoath Hossain Tanvir
- Ariful Islam
- Rayhan Kabir
- Moshiur Rahman

### **Meeting Outcome**

It all seems good now we will wait for the demonstration.

## Sprint Backlog

User Story	Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Role Based Login	Meeting with DB designer	⚡	⚡					
	Meeting with UI designer		⚡					
	Code the interface			⚡	⚡	⚡		
	Testing with tester				⚡			
	Security Test					⚡	⚡	
Adding Assessments	Meeting with Tanvir	⚡	⚡					
	Meeting with UX design		⚡					
	Code the system			⚡	⚡			
Management of users	Meeting with team	⚡						
	Interface design		⚡					
	Code the design			⚡	⚡			
Submission of Extenuating circumstances	Meeting with Tanvir	⚡	⚡	⚡	⚡			
	Meeting for design		⚡					
	Code the design			⚡	⚡	⚡		
	Security test			⚡		⚡		
	Update security test				⚡			
	Code the final outcome					⚡	⚡	⚡

Getting email	Meeting about design	⚡						
	code the design		⚡	⚡	⚡			
	Testing the part			⚡	⚡	⚡		
Procession of Extenuating circumstances	Meeting with Tanvir	⚡	⚡	⚡				
	Meeting for design		⚡				⚡	
	Code the design		⚡	⚡	⚡	⚡		
	Security test			⚡	⚡		⚡	
	Update security test					⚡		⚡
Viewing of Extenuating circumstances	Design the interface	⚡						
	Meeting with team mates		⚡	⚡				
	Write test plan			⚡				
	Automate tests				⚡			
	Update security tests					⚡		
	Testing...						⚡	⚡
Viewing reports based on filters	Design the interface	⚡						
	Meeting with team mates		⚡	⚡				
	Write test plan			⚡				
	Automate tests				⚡	⚡		

	Update security tests							
	Testing...							

## Product Backlog

ID	As a/an	I want to ....	So that ....	Notes	Priority	Estimation	Status
1	User	log in the system.	I can access the system.	Role based login	1	7	<b>Done</b>
2	Student	Submit an Extenuating circumstances	I can claim for my extenuating circumstances.	Multiple file can be attached	4	7	<b>Done</b>
3	Course Co-ordinator	Processing of Extenuating Circumstances	I can either accept or reject the extenuating circumstances that has been applied for.	Notes can be added	6	7	<b>Done</b>
4	Course Co-ordinator	Get email	I want to receive email on submission of new EC in my faculty		5	4	<b>Done</b>
4	Student	Get email	I want to receive email upon getting the result of submitted EC		7	3	<b>Done</b>
5	Administrator	Manage users	I can see existing users	Musn't have the access to modify or delete the users	3	7	<b>Done</b>

<b>6</b>	Manager	view existing extenuating circumstances	I can view existing EC and their result.	Can't modify	8	2	<b>Done</b>
<b>6</b>	Administrator	view existing extenuating circumstances	I can view existing EC and their result.	Can't modify	8	2	<b>Done</b>
<b>6</b>	Course Co-ordinator	view all extenuating on my faculty	I can view all the EC and their result - submitted on my department.	Can't modify	8	2	<b>Done</b>
<b>6</b>	Student	view all my extenuating circumstances	I can view all my EC that I have applied so far and their result.	Only view	8	1	<b>Done</b>
<b>7</b>	Administrator	view the report	will be able to see claimed EC through multiple filter.	Only view	9	3	<b>Done</b>
<b>7</b>	Manager	view the report	will be able to see claimed EC through multiple filter.	Only view	9	2	<b>Done</b>
<b>7</b>	Course Co-ordinator	view the report	will be able to view claimed on my faculty and can filter them.	Only view	9	2	<b>Done</b>
<b>8</b>	Administrator	add assesments	I can add new assesments for the courses and deadline for them.	Can't update or delete once submitted	2	7	<b>Done</b>
					Total =	56	

## Burn down chart



Actual Progress

Projected Progress

Key Points