Course: Comp 1640

Enterprise Web Software Development

Submitted To	Submitted By
Name: Dr Ray Stoneham	**Name:** Mhm Munem Maruf-ur-Rahman
Designation: Course Leader	
University of Greenwich, UK	**ID: 000989724**

Date of submission: 12th April, 2017

Team Members and Roles

Team Name: SCRUM SQUAD	
Member Name	**Role**
Shahnewaj Muhammad Shakil	Database Designer
Abdullah Al Masum	Site Designer, Programmer
Abdullah Al Zubair	Tester
Mhm Munem Maruf-Ur-Rahman	**System Analyst, Scrum Master**

System Credentials: -

Role	Email	Username	Password
Administrator			
Course Coordinator			
Manager			
Student			

Note- all users can be created dynamically; it is not necessary to use these credentials for checking purposes.

Note 2- to create a student, complete student admission form and approve as admin.

Contents

Introduction

ECC Processing System is a role-based application for a large university that collects Extenuating Circumstances claims or EC claims made by the students and processes them. It has been assumed that the university already has a system and this particular application is a subsystem. Since this is a group assignment, SCRUM SQUAD followed scrum framework to help divide the tasks according to the roles to be played by each team member in the development process-

- A system analyst to breakdown the system's requirements and outline its structure.
- A database designer for developing the database and related tasks.
- A programmer to implement the system according to the provided architecture and database.
- A tester to test whether system meets all the requirements.

Note: - throughout the development, some assumptions needed to be made by each member in order to make a more robust and efficient system.

This report consists of brief details about my contribution to the team as a system architect which includes diagrams, charts, assumptions, evaluation of the system as well as the group, probable future developments for the system, etc. (The contents of the report are represented as simply as possible for easier understanding).

Evaluation of product and process

ECC Processing System is a role based system for processing extenuating circumstances (EC) claims made by students at a university. Among all the key features of the system the two most important functionalities include the system to be role-based and system to be able to process EC claims. Before commencing development, user requirements were defined and listed out with given level of priorities so that the key features are developed first because Agile methodology encourages the most important features to be developed first without having to fulfill every single requirements, even the less important ones, so that a fully functional system that is able to perform the needed tasks can be delivered in a short period of time. The less important features are left for further development of the system.

Brief descriptions and evidences of the key features are given below-

Role-based system

Administrator:

Admin's role is to maintain system data that includes, adding users, editing information, enrolling students, creating- faculties, courses, assessments and related academic dates, type of EC claims and assigning Coordinators, Manager and Administrators.

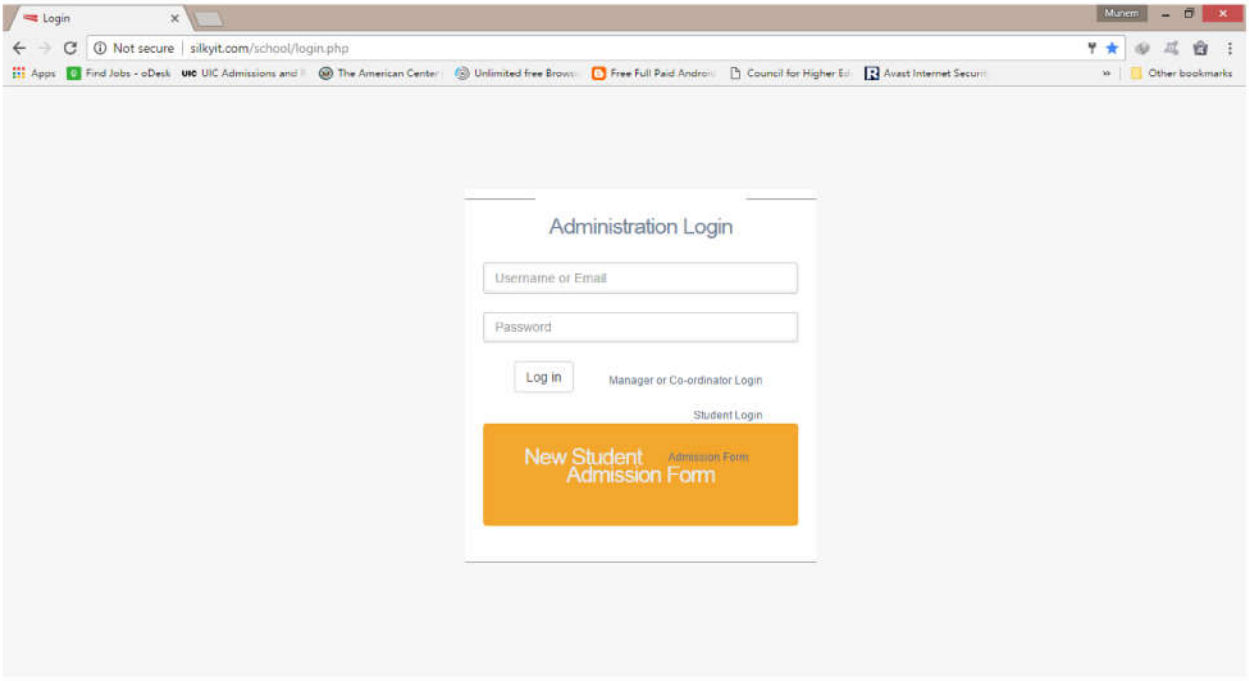

Figure 1: Admin Login

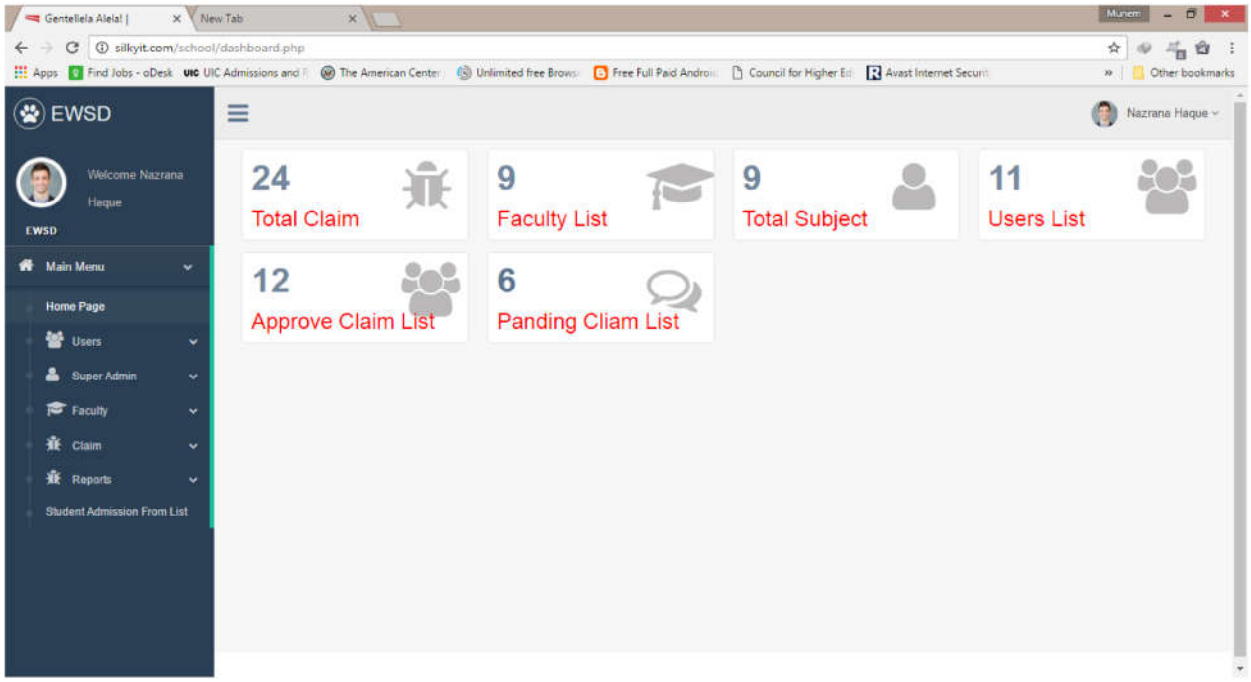

Figure 2: Admin Dashboard

Manager:

Manager's role is to oversee all the processes carried out in the system and to view reports.

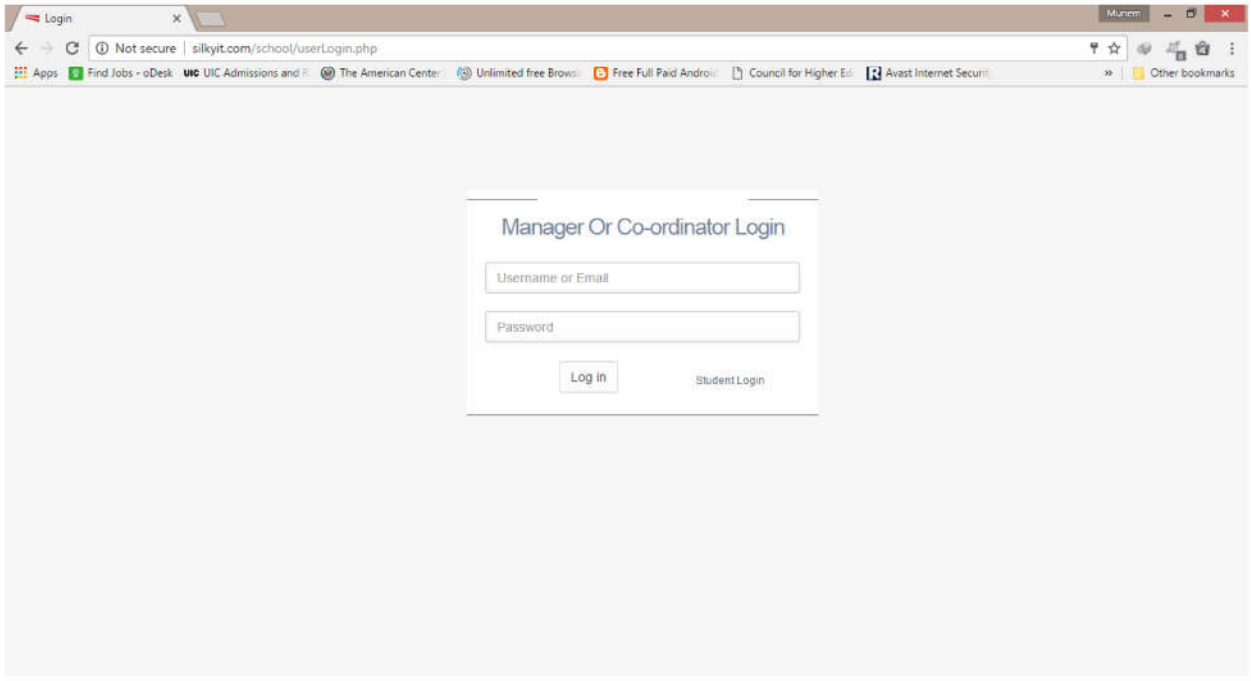

Figure 3: Manager Login

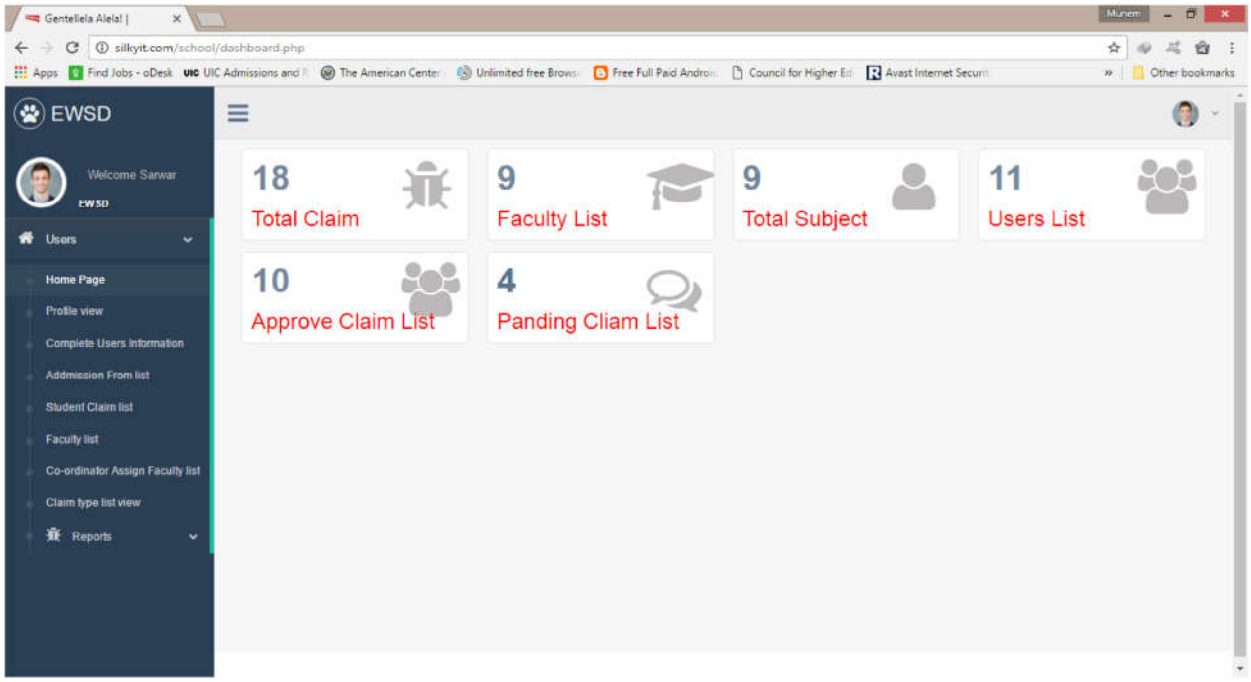

Figure 4: Manager Dashboard

EC Coordinator:

The Coordinator's role is to view the claims in their respective faculties and approve or reject the claims made by students from the respective faculties.

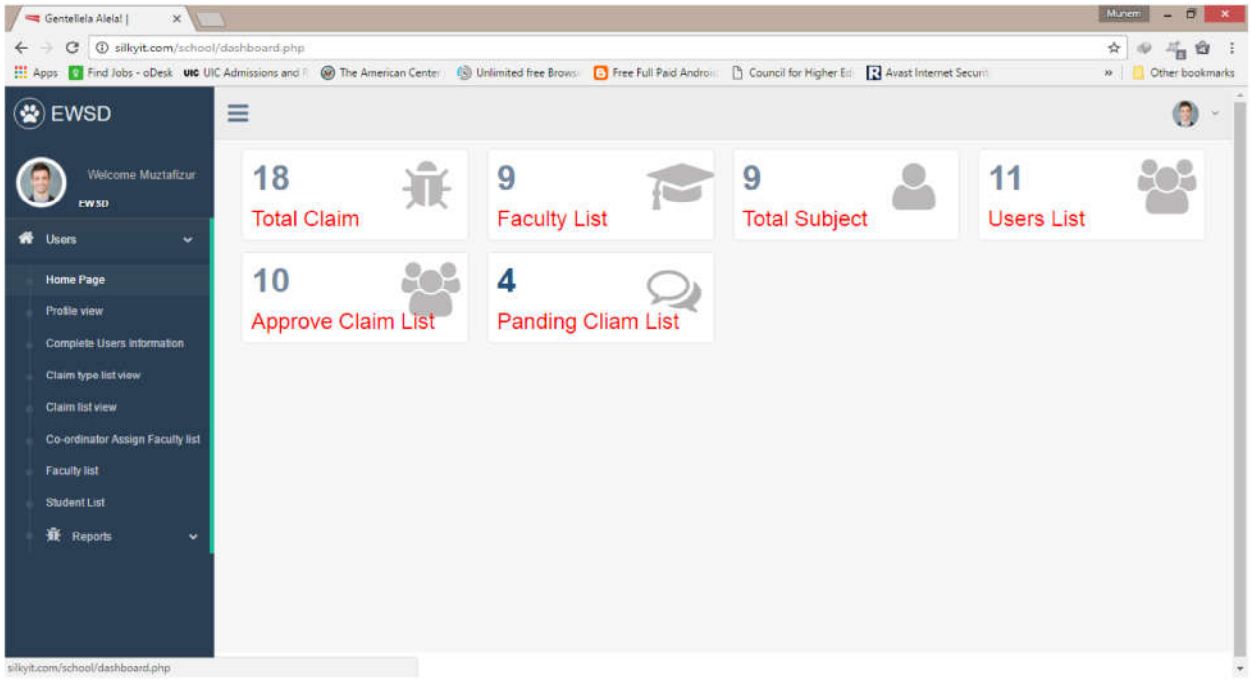

Figure 5: Coordinator Dashboard

Student:

A student's role for the system is to submit EC claims and view their claims.

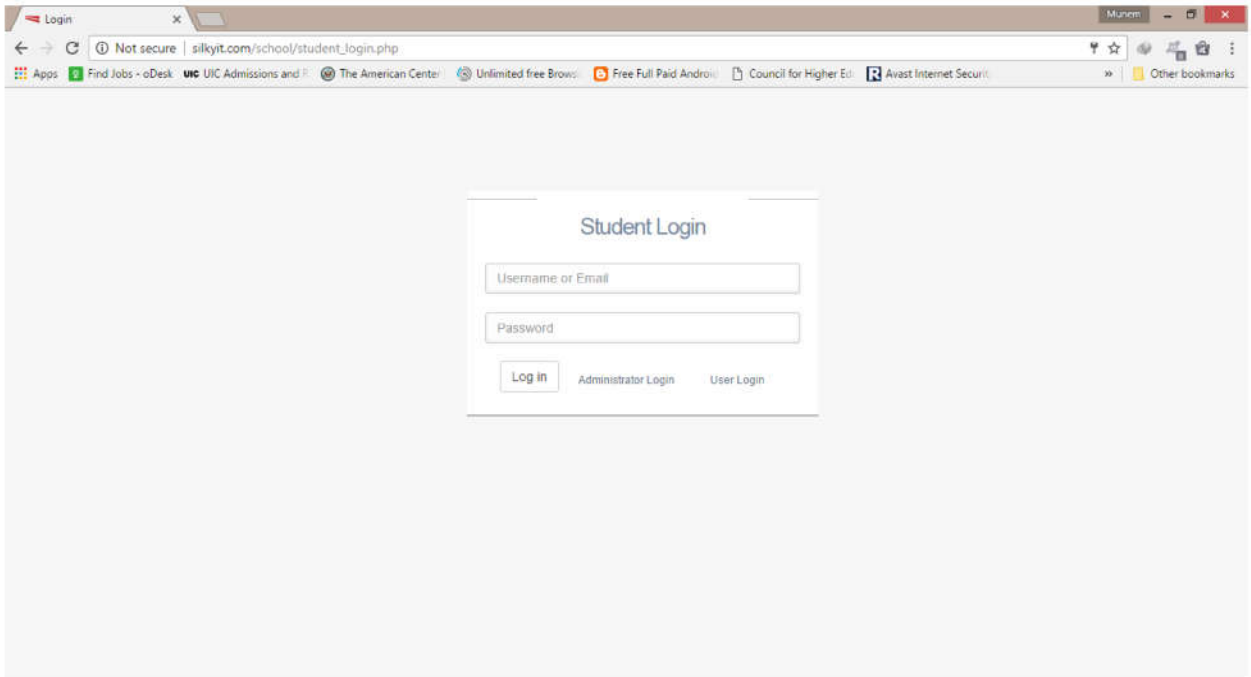

Figure 6: Student Login

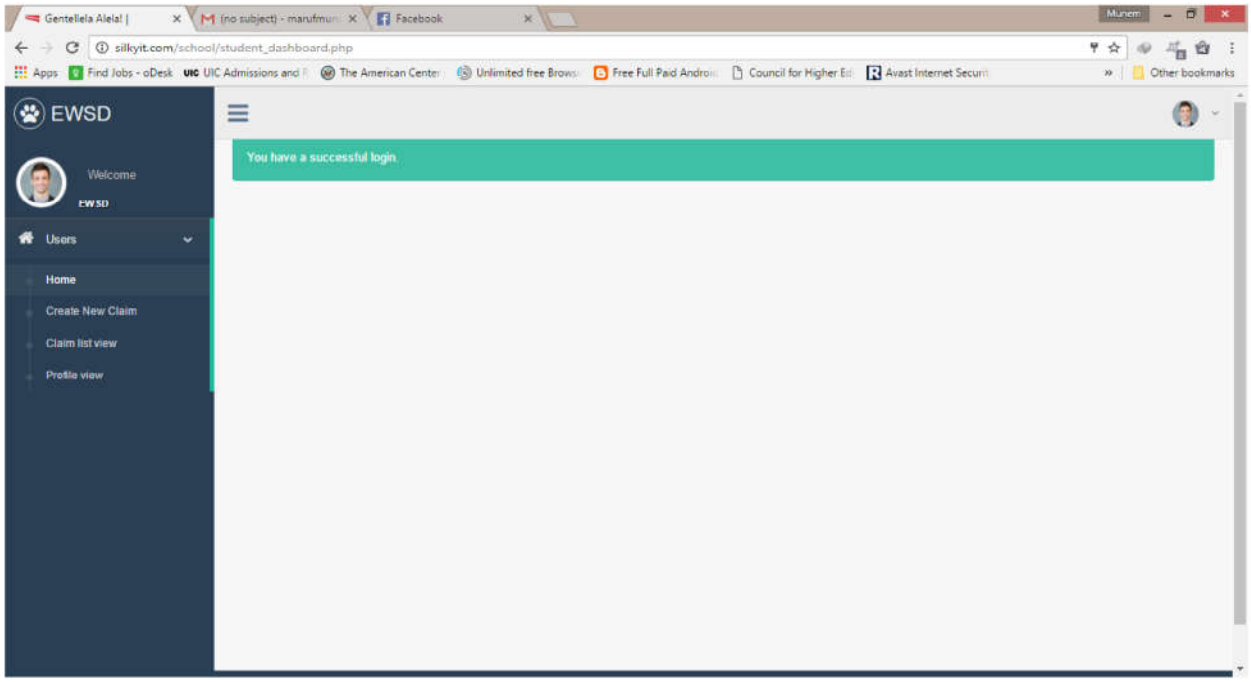

Figure 7: Student Dashboard

EC Coordinator's claim processing

Coordinator can either 'Confirm' an EC claim to approve it or 'Reject' an EC claim, both of which sends email to the student that submitted the claim.

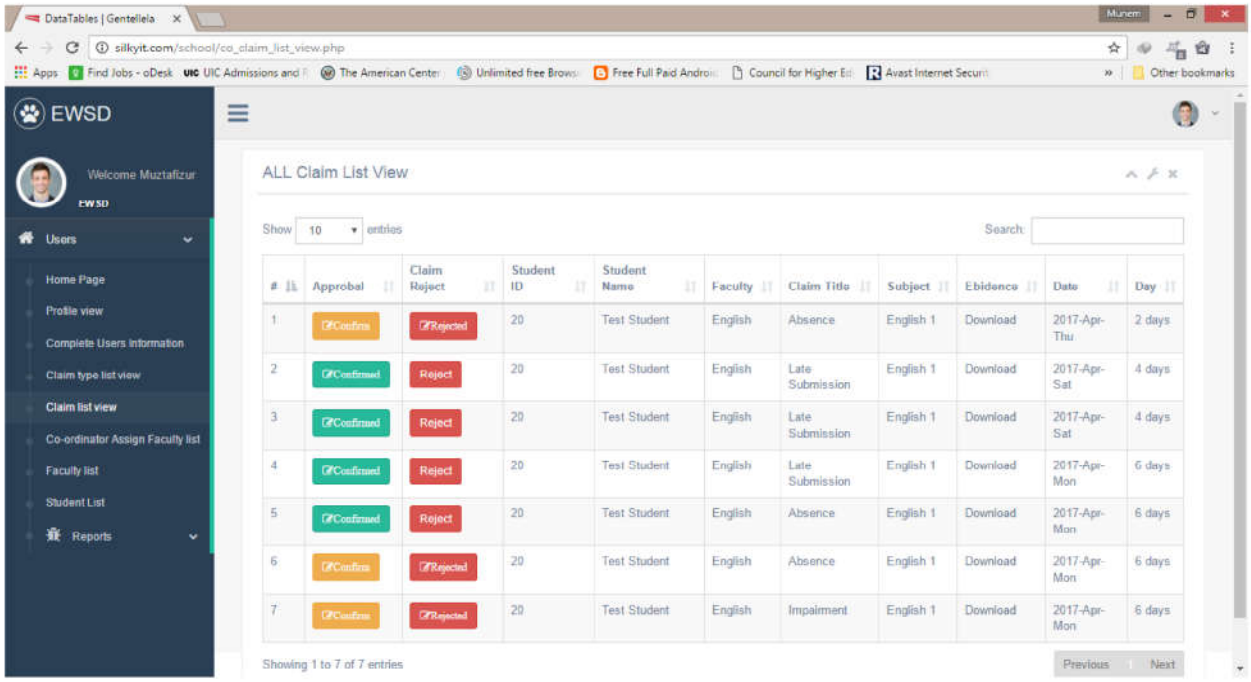

Figure 8: Coordinator's view of the claims

Student email notification

Student is notified via email after Coordinator processes their claims. The email notification is an automated function of the system.

Note- the email message here is just for test purposes, the message will either show that the Coordinator accepted the claim or show that the claim has been rejected.

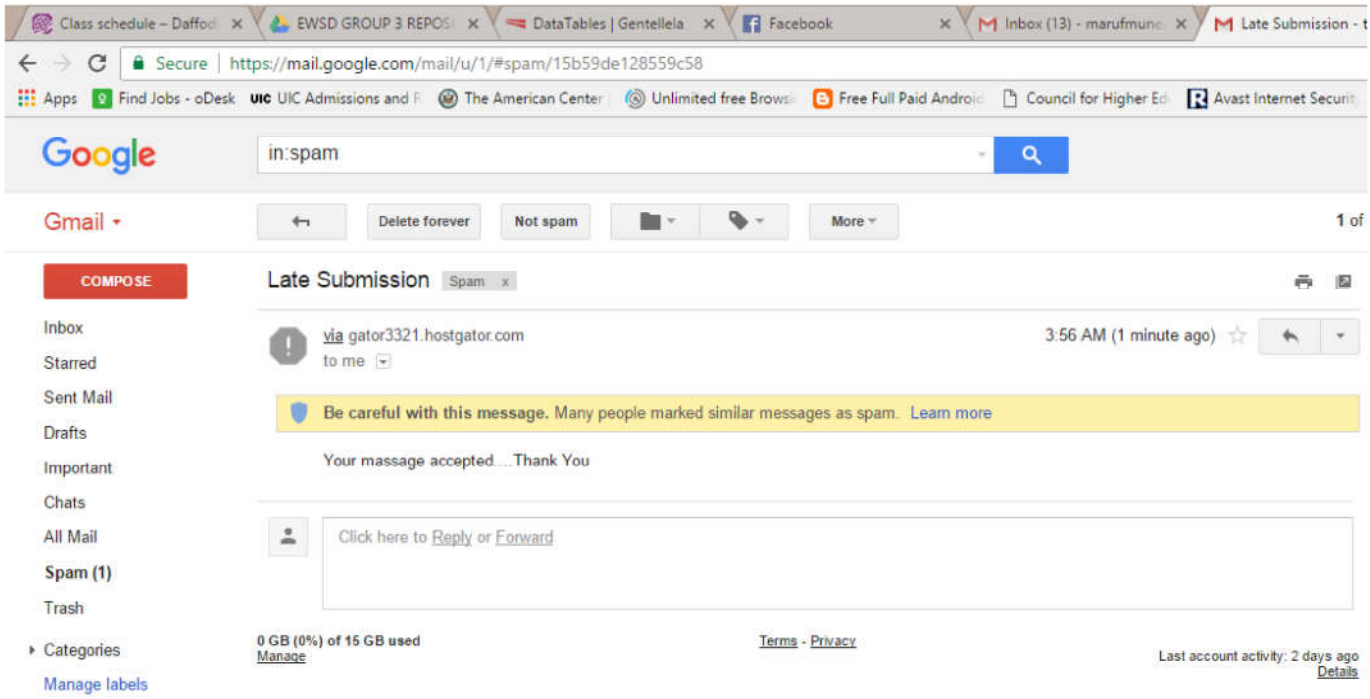

Figure 9: Email sent to Student from Coordinator

Reports

Reports are already available on Admin's, Manger's and Coordinator's dashboards but they can also generate reports, both Statistical and Exceptional, according to their demands. Below are examples from the Manager's dashboard.

Statistical Report

This report will show the number of claims and the percentage of claims made in a particular year and the number of students that made the claims.

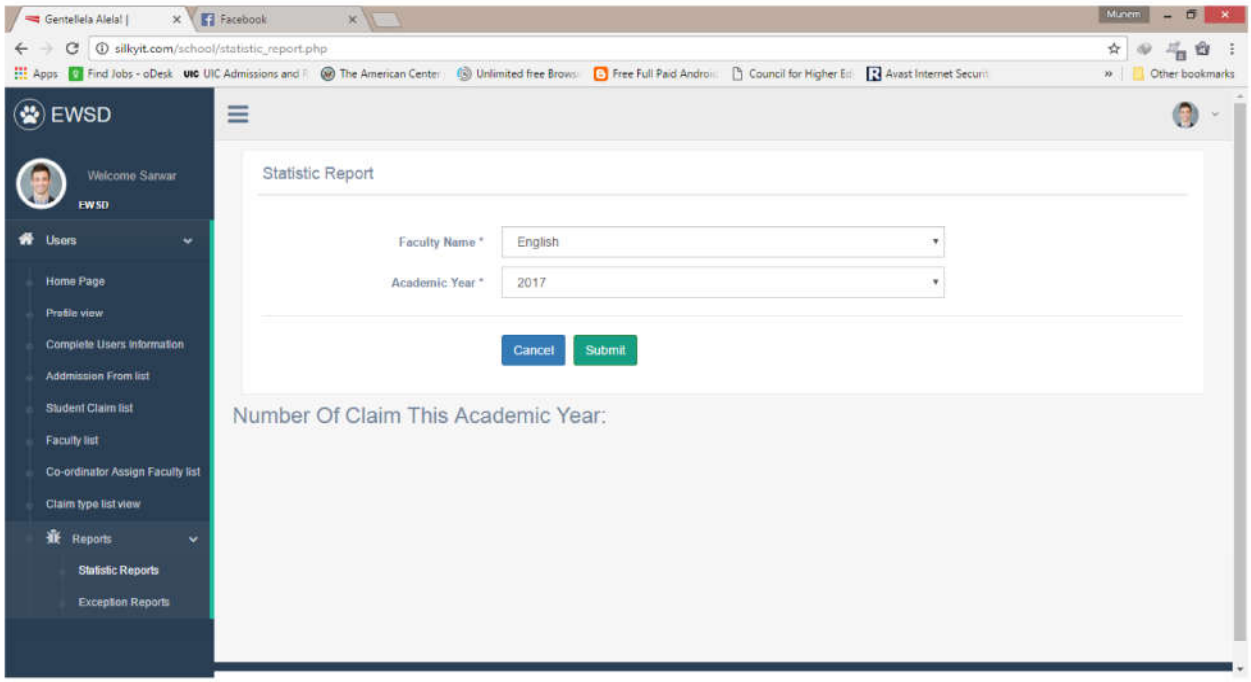

Figure 10: Manager selects condition for the Statistical report to be generated

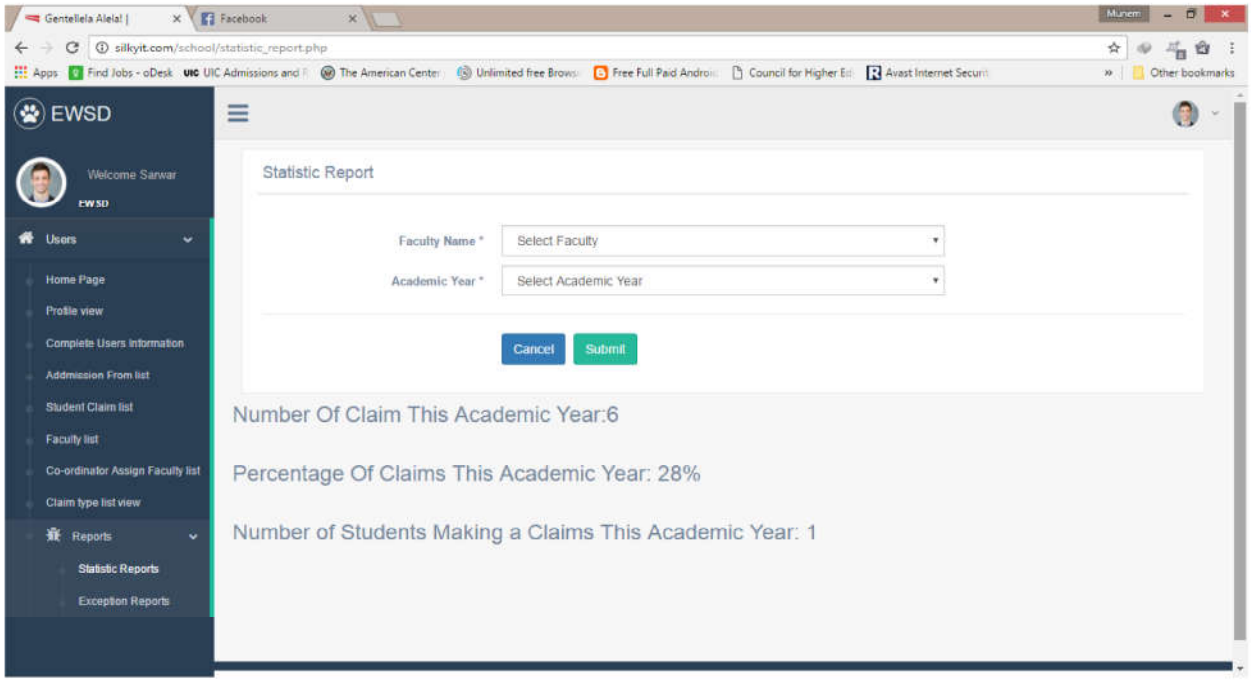

Figure 11: Statistical report

Exceptional Report

This report will show how many EC claims were made where students did not upload their evidences and how many claims remained unprocessed after 14 days of claim submission.

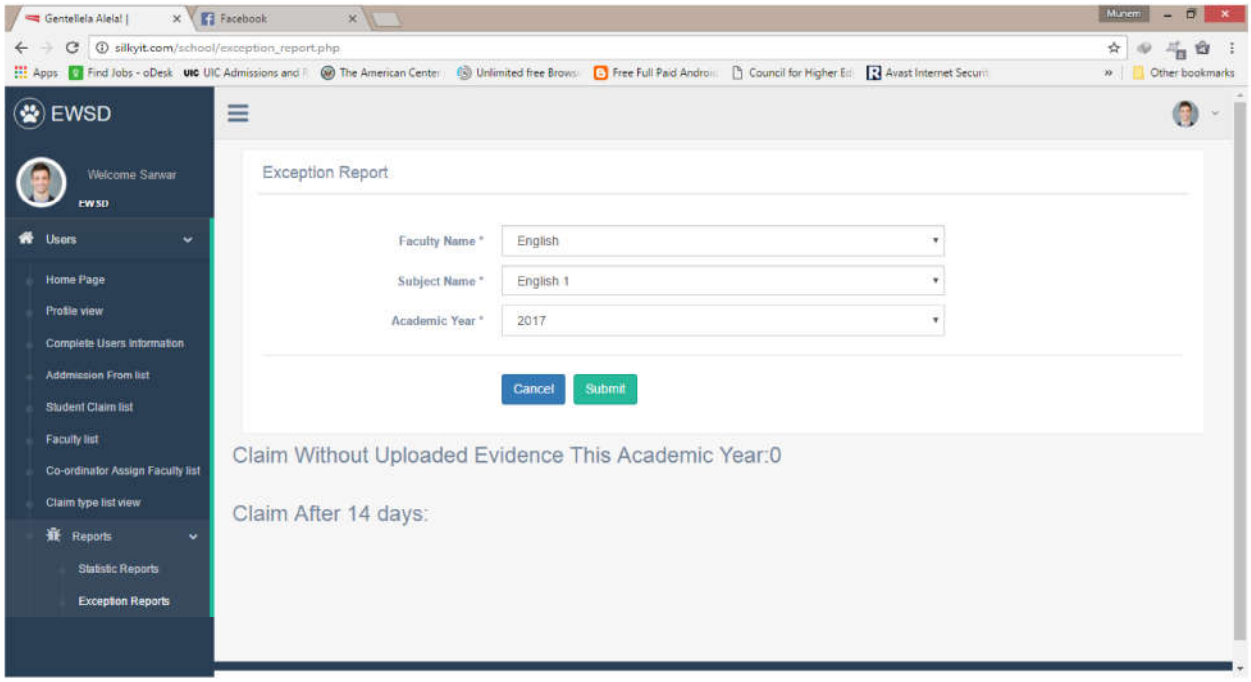

Figure 12: Manager selects condition for the Exception report to be generated

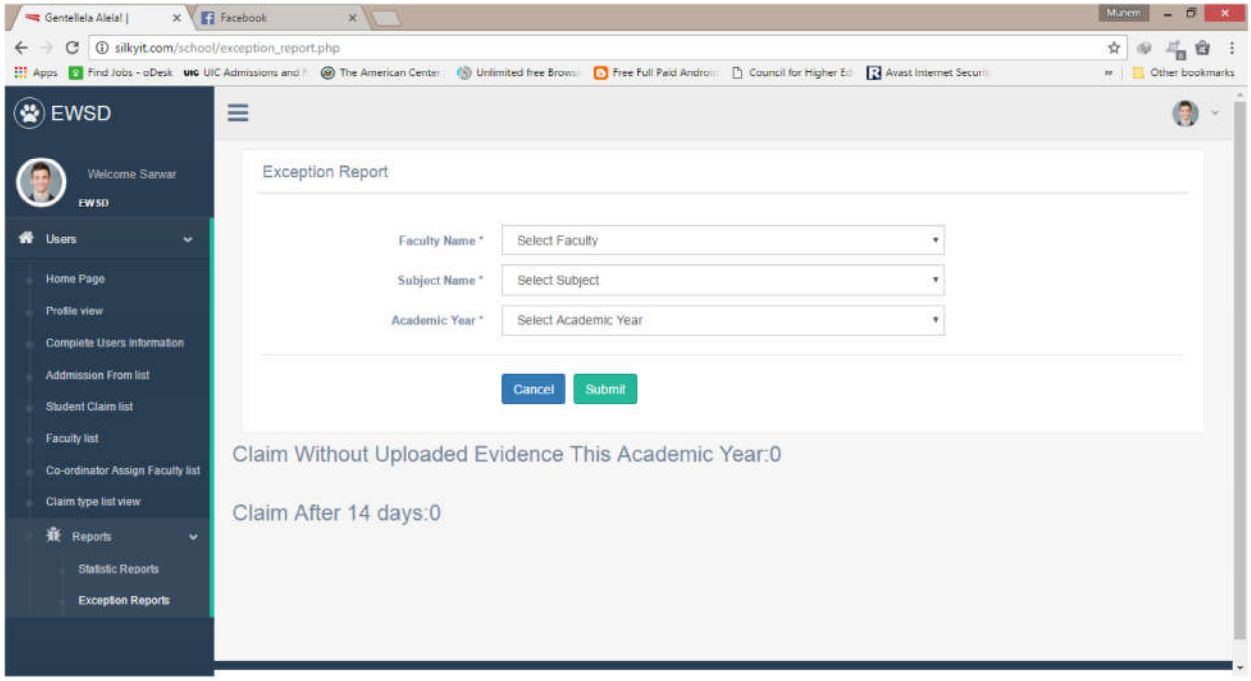

Figure 13: Report showing no exceptional cases made

Responsiveness

The system is made to run on all types of devices including mobile phones and tablets which makes it highly usable. Here, is an example of it running on a phone.

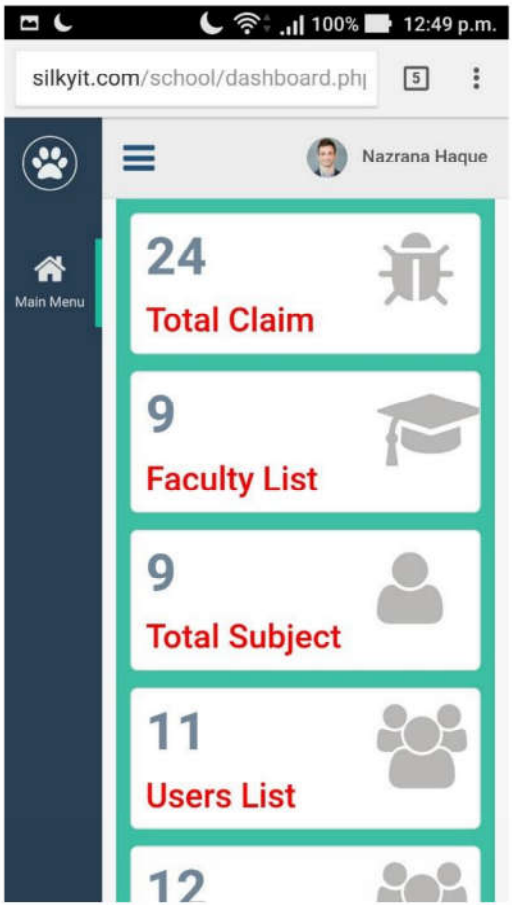

Figure 14: Administrator dashboard logged in from a smartphone

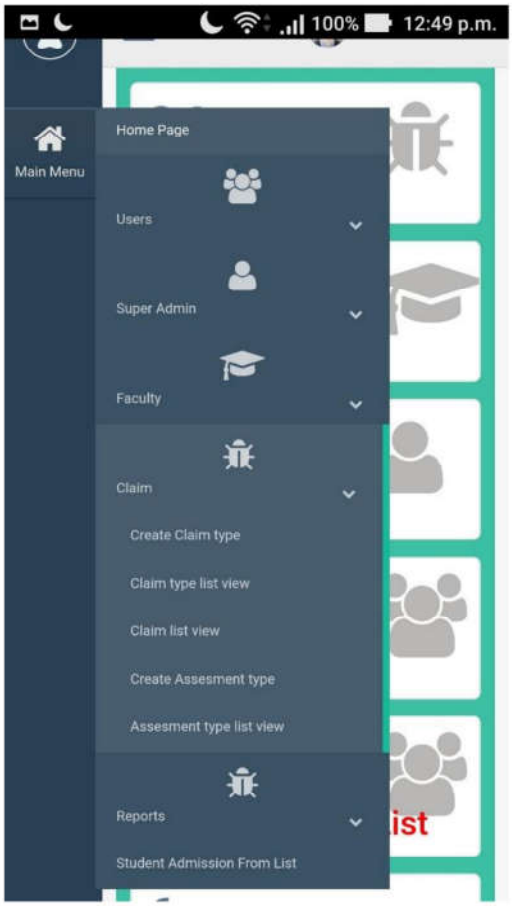

Figure 15: Administrator dashboard logged in from a smartphone

System Strengths

These are the features that makes the system robust and stand out from the other applications-

- Secure role-based system
- Password encryption for higher security
- Fully dynamic (even students have to be enrolled before they can be involved with the system)
- Organized navigation
- Highly responsive
- Notification via email so that users don't have to login to the system every time to check relevant statuses
- Claim evidences can be downloaded
- Dashboards show user-relevant information (e.g. number of pending claims for the EC Coordinator to be processed)

System Weakness

These are the features that need further improvement and features that could make the system better if implemented-

- Users other than students don't get any email notification
- UI could be made more aesthetically pleasing
- Within system notification should also be integrated

Further Development

From Agile perspective, the system can be assessed as quite robust since it fulfills all the main functionalities required. However, further improvements can make the system even more enhanced-

- Email notification for all users
- 2-factor authentication for higher security
- Reminder alarm for the Coordinator to process claims within 14 days
- Live messaging/chat box features
- Report printing facilities
- Personalization facilities

Development Tools

- PHP
- Java Script
- MySQL
- JQuery
- Ajax
- CSS
- HTML
- Bootstrap

Evaluation of team

The name of our team is SCRUM SQUAD made up of four members. The best part of the team is that the mindset of each member matched with the rest. The team wanted the task to be done efficiently and strictly on time and provide the best everybody has got and thankfully, the team managed to pull through as planned.

As the system analyst of the team, I managed to breakdown the requirements as needed by the system and shared them with the rest of the team, followed by the site designer's role who designed the system so that it appears as user friendly and as eye-catching as possible. The designer also played the role of the team's programmer. Database designer played a vital role as his job was to design a fully normalized, well thought-out database which ultimately helped the programmer to develop a system that is secured, dynamic and responsive. Finally, it was the tester's role to test the system throughout the development, at the end of each sprint, to assure quality of the system. Tester tested the system according to prepared test cases and provided feedback consistently.

As mentioned earlier, mindset is a key factor in a group project and so is communication. There were regular scrum meetings which were solely necessary for this assignment to be a success involving discussions on previously given tasks and upcoming tasks. The contribution of each member, however, can be assessed against a set of criteria-

- Availability
- Communication
- Role-based skills
- Technical skills
- Collaboration
- Confidence

To assess all the members of the team, I have made an estimation within a scale of 5. The following is a tabular representation of that estimation-

Assessment Criteria		Names			
		Maruf	Shakil	Masum	Zubair
	Availability	5	3	3	5
	Communication	5	4	4	4
	Role-based skills	4	5	4	5
	Technical skills	3	4	4	4
	Collaboration	5	3	4	4
	Confidence	5	5	5	4

Self-evaluation

As the analyst of the group, I had to go over the assignment scenario several times until I managed to breakdown the requirements and shared them with the team members. I made a number of assumptions where needed and contributed my ideas to the team. I have identified the use cases and modelled the UML diagrams and site maps for a more robust understanding of the system.

I did not limit myself to the required tasks as an analyst. I encouraged my team every moment, whether it is on a meeting, in class, on the phone or on social media, so that they don't fall out of interest or fall behind schedule, as this tends to happen a lot in group tasks. I have also remained involved with their individual tasks that included providing a lot of feedback to the database designer and the programmer and collaborating with the tester, also participating in the testing role myself. I have also volunteered to develop the artefacts of the scrum documentation and completed them collaborating with the team; it included user stories, product and sprint backlogs, burndown chart and scrum meeting reports. I do lack quite on the technical side but I feel that collaborating with the group throughout this assignment gave me more confidence with my skills.

Followed Methodology

For this particular project, Scrum framework, which is one of the Agile frameworks, was recommended to be followed. This framework was best suited for this project for a number of reasons-

- Scrum is a popular framework for team-based development.
- Team consists of members that can perform individual roles in the development process.
- No member was put under pressure or forced to take on a role they are not comfortable with.
- Scrum allowed us to break down tasks and perform them as in 'sprints'.
- It allowed us an iterative development of the system because requirements are interrelated and delivery of the system can be made incrementally.
- Involves daily scrum meetings which helped us collaborate more often and be more progressive.

Lesson Learned

Teamwork is not an easy task. What I have learned from this project is that the first step to a successful teamwork is a mindset. It is very important that the entire team is on the same page and have the same ultimate goal. As long as this is not established, the road ahead is bound to be rough. Another important thing I believe I have learned is to grow more patient. This was a contributed work from four different individuals and things did not always go as planned but not having to fall out, and consistently improvising and moving is what I believe helped the team members click at one point and ultimately complete the assignment.

Working as a team also helped learn about how an actual project in the professional field may work and the various tools and techniques used. This was probably just a glimpse but it's a start. I am privileged to be a part of this assignment and the team.

Conclusion

SCRUM SQUAD did its best to accomplish its goal which was to deliver a system that processes Extenuating Circumstances claims made my students on academic assessments. It took more than a month and a lot of determination and hard work to develop a fully functional system. I believe the system could still be further developed and could undergo a lot of improvements. However, the biggest challenge probably was the teamwork, because I believe a system like this needs a group of members that are well-accustomed to Agile practices, but for a group of newbies like us, this was a great start.

Appendices

Appendix A: Requirements Specification

Functional Requirements

1) Users can login to the system that takes them to their individual dashboards.
2) Admin can add, update and delete user.
3) Admin assign roles i.e., assign EC Coordinator, Manager, Administrator.
4) Admin assigns closure dates of courses and items of assessment.
5) EC Manager can view all the processed and unprocessed EC claims.
6) EC Coordinator can approve or reject an EC claim.
7) EC Coordinator must process an EC Claim within 14 days after the claim is made.
8) Students can submit more than one EC claims on any type of assessment.
9) Students can submit evidence until the closure date is reached.
10) EC Coordinator can only process a claim if evidence is provided.
11) Each faculty member can view processed (or unprocessed) claims in their faculty only.
12) Each student can only view the claims they make.
13) Notification via Email.
14) All faculty members can generate and view statistical and exception reports.

Non-functional Requirements

1) All faculties have an EC Coordinator.
2) Number of processed (and unprocessed) EC claims.
3) Percentage of processed (and unprocessed) EC claims.
4) Users can change their password.
5) Admin can maintain system data and user data.
6) Users can view reports.

Assumptions

1) EC claim processing system is a subsystem of the university.

2) Admin maintains the entire system and all the tasks.
3) Admin sets the boundaries for each role.
4) Admin can allow or withdraw services for a specific role.
5) Only users with authorized IP addresses can access the system.
6) Admin does not have the right to change or view user password after providing it.
7) Passwords should be encrypted.
8) Unique username and password must be a minimum of 8 characters with one upper case letter, one lower case letter, numbers and symbols.

Use Case Diagrams

Admin Use case diagram

Figure 16: Admin Use Case Diagram

EC Coordinator Use case diagram

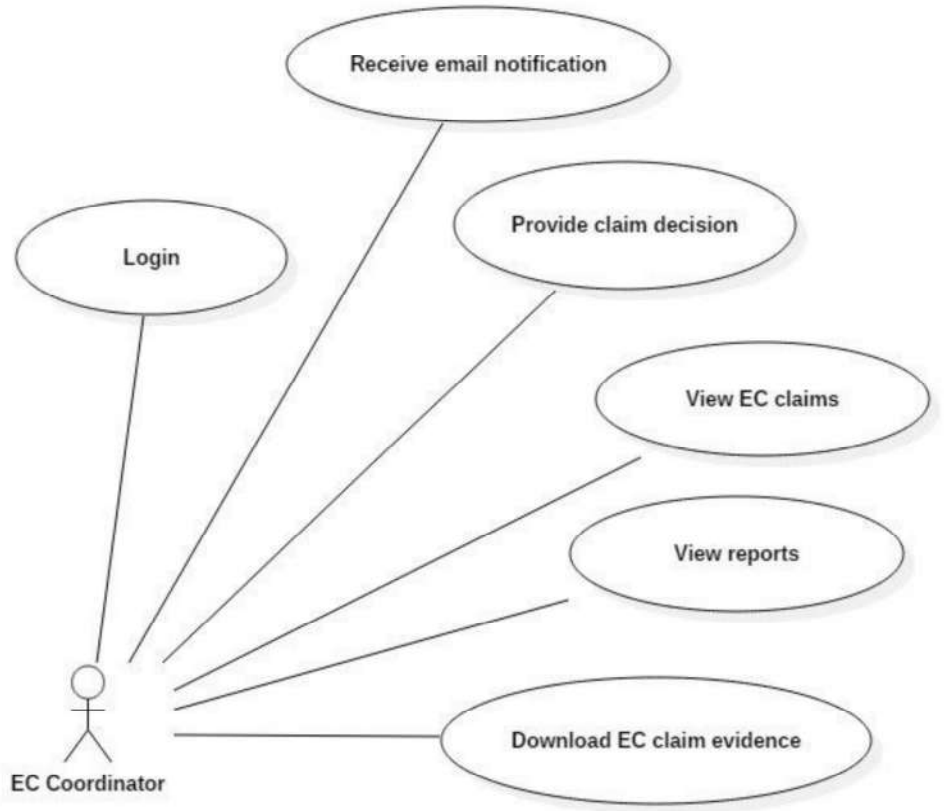

Figure 17: EC Coordinator Use Case Diagram

EC Manager Use case diagram

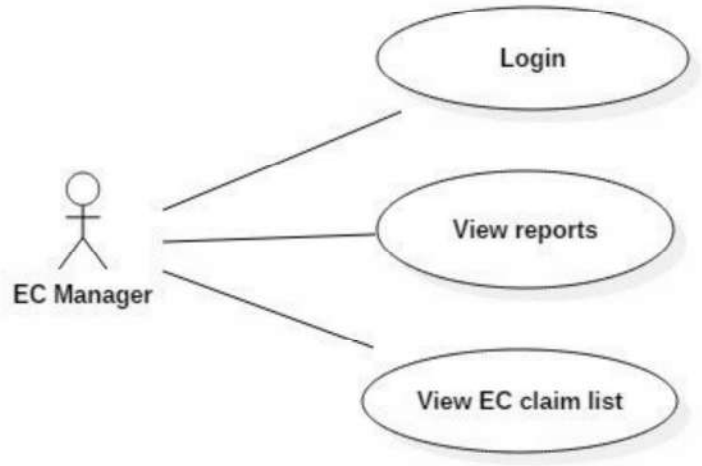

Figure 18: EC Manager Use Case Diagram

Student Use case diagram

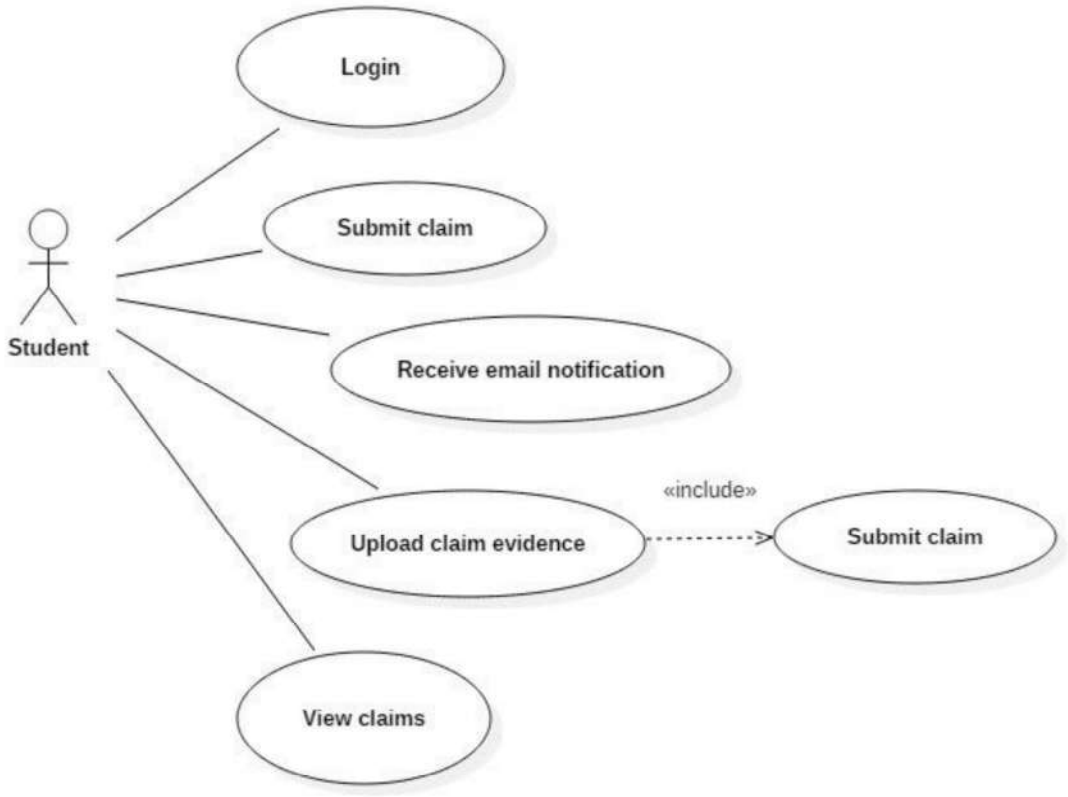

Figure 19: Student Use Case Diagram

Use Case Descriptions

Use case id	**001**
Use case title	Create Faculty
Actor	Admin
Pre-condition	Admin must be logged in to the system
Primary scenario	1. Admin goes to dashboard and opens Main Menu 2. Select Create Faculty from Faculty 3. Input faculty details 4. Creates faculty

Secondary scenario	Faculty cannot be created because a faculty with a similar name already exists.
Post-condition	Faculty is created

Use case id	**002**
Use case title	Assign Coordinator
Actor	Admin
Pre-condition	Admin must be logged in to the system
Include	Complete user information
Primary scenario	1. Admin goes to dashboard and opens Main Menu 2. Admin opens Faculty 3. Admin selects Assign Coordinator 4. Input Coordinator name and Faculty name 5. Coordinator is assigned to the faculty
Secondary scenario	Coordinator cannot be assigned to the faculty because a Coordinator already exists for that particular faculty.
Post-condition	Coordinator is assigned for a particular Faculty

Use case id	**003**
Use case title	Create Manager
Actor	Admin
Pre-condition	Admin must be logged in to the system
Primary scenario	1. Admin goes to dashboard and opens Main Menu

	2. Admin opens Faculty 3. Admin selects Create new user 4. Input user details and select Manager from User category 5. Manager is created
Secondary scenario	Manager is not created because a Manager with the same email already exists.
Post-condition	Coordinator is assigned for a particular Faculty

Use case id	**004**
Use case title	Create other Admin
Actor	Admin
Pre-condition	Admin must be logged in to the system
Include	Complete user information
Primary scenario	1. Admin goes to dashboard and opens Main Menu 2. Admin opens Super Admin 3. Admin selects Create New Super Admin 4. Input user details 5. A new Administrator is created
Secondary scenario	Administrator cannot be created because an Admin with the same email exists.
Post-condition	Administrator is created.

Use case id	**005**
Use case title	Enroll Student
Actor	Admin

Pre-condition	Admin must be logged in to the system
Include	Student submits admission form
Primary scenario	1. Admin goes to dashboard and opens Main Menu 2. Admin opens Student Admission Form list 3. Admin presses Confirmed button to accept student admission 4. New Student is enrolled into the system
Secondary scenario	Admin does not Confirm student admission thus not enrolling a student because the student may have not submitted their photo.
Post-condition	Coordinator is assigned for a particular Faculty

Use case id	**006**
Use case title	Create subject
Actor	Admin
Pre-condition	Admin must be logged in to the system
Include	Create Faculty
Primary scenario	1. Admin goes to dashboard and opens Main Menu 2. Admin opens Faculty 3. Admin selects Create Faculty Subject 4. Input details about faculty, subject name and all associated dates 5. Subject is created

Secondary scenario	Subject is not created because same a subject with the same name already exists in that particular Faculty.
Post-condition	Subject created for the Faculty

Use case id	**007**
Use case title	Assign dates
Actor	Admin
Pre-condition	Admin must be logged in to the system
Include	Create subject
Primary scenario	1. Admin goes to dashboard and opens Main Menu 2. Admin opens Faculty 3. Admin selects Create Faculty Subject 4. Input dates including closure date for the subject for a particular session- the dates for claim and evidence submission.
Secondary scenario	Dates are not assigned because a subject with the same name is created because dates depend on subject created.
Post-condition	Dates are assigned for a subject.

Use case id	**008**
Use case title	Create assessment
Actor	Admin
Pre-condition	Admin must be logged in to the system
Include	Create subject

Primary scenario	1. Admin goes to dashboard and opens Main Menu 2. Admin opens Claim 3. Admin selects Create Assessment Type 4. Input Assessment details 5. A new assessment is created
Secondary scenario	Assessment is not created because same assessment type already exists.
Post-condition	Assessment is created

Use case id	**009**
Use case title	Create EC claim type
Actor	Admin
Pre-condition	Admin must be logged in to the system
Primary scenario	1. Admin goes to dashboard and opens Main Menu 2. Admin opens Claim 3. Admin selects Create Claim Type 4. Input claim type details 5. A new claim type is created
Secondary scenario	Claim type is not created because same claim type already exists.
Post-condition	Claim type is created.

Use case id	**010**
Use case title	View all subjects
Actor	Admin
Pre-condition	Admin must be logged in to the system

Include	Create subject, Create faculty
Primary scenario	1. Admin goes to their dashboard and opens Main Menu 2. Admin opens Faculty 3. Admin selects Faculty Subject list 4. Subjects are displayed along with their respective faculties
Secondary scenario	Subjects cannot be shown if there are no subjects and/or faculties created.
Post-condition	All Faculty-wise subjects are shown

Use case id	**011**
Use case title	View all Faculties
Actor	Admin, Coordinator
Pre-condition	Actors must be logged in to the system
Include	Create faculty
Primary scenario	1. Actor goes to their dashboard and opens Main Menu 2. Actor opens Faculty 3. Actor selects Faculty list 4. Faculties are displayed
Secondary scenario	Faculties cannot be shown if there are no faculties created.
Post-condition	All Faculties are shown

Use case id	**012**
Use case title	View EC Claims
Actor	Admin, Coordinator, Manager

Pre-condition	Actors must be logged in to the system
Include	EC claims made by students
Primary scenario	1. Actors goes to their dashboard and opens Main Menu 2. Actors opens Claim 3. Actors selects Claim list view 4. Admin can view all EC claims from all Faculties 5. Manager can view all EC claims from all Faculties 6. Coordinator can view claims made only in their respective faculty
Secondary scenario	Claims cannot be displayed if no claims are made
Post-condition	All claims are shown along with the students that submitted the claims and the dates of submission.

Use case id	**013**
Use case title	View all users
Actor	Admin
Pre-condition	Admin must be logged in to the system
Primary scenario	1. Admin goes to their dashboard and opens Main Menu 2. Admin opens Users 3. Admin selects All Users List to view all users associated with the system

	4. Admin selects Coordinator List to view all Coordinators associated with the system 5. Admin selects Manager List to view all Managers associated with the system 6. Users are displayed
Secondary scenario	Users cannot be shown if there are no users in the system.
Post-condition	Users are shown according to their names and their categories.

Use case id	**014**
Use case title	Receive email notification
Actor	EC Coordinator, Student
Pre-condition	Actors must be logged in to their email accounts
Include	1. EC claims made by students 2. Claim decision provided
Primary scenario	1. Actor goes to their email account 2. Actor opens mail about a claim 3. Coordinator opens email about a submitted claim 4. Student opens email about a decision made on their submitted claim
Post-condition	Claim email is received.

Use case id	**015**
Use case title	Provide claim decision
Actor	Coordinator

Pre-condition	Coordinator must be logged in to their email accounts
Include	EC claims made by students
Primary scenario	1. Coordinator goes to dashboard and opens Main Menu 2. Coordinator opens Claim list view 3. Approve or reject claim
Post-condition	Claim decision is provided and email is sent to student.

Use case id	**016**
Use case title	Provide claim decision
Actor	Coordinator
Pre-condition	Coordinator must be logged in to the system
Include	EC claims made by students
Primary scenario	1. Coordinator goes to dashboard and opens Main Menu 2. Coordinator opens Claim list view 3. Approve or reject claim
Post-condition	Claim decision is provided and email is sent to student.

Use case id	017
Use case title	Download claim evidence
Actor	Coordinator
Pre-condition	Coordinator must be logged in to their email accounts
Include	Claim evidence submitted by student
Primary scenario	1. Coordinator goes to dashboard and opens Main Menu 2. Coordinator opens Claim list view 3. Download evidence
Post-condition	Claim evidence is downloaded

Use case id	018
Use case title	View reports
Actor	Admin, Manager, Coordinator
Pre-condition	Actors must be logged in to the system
Include	None
Primary scenario	1. Actors goes to their dashboard and opens Main Menu 2. Actors open Reports 3. Actors input details about faculty, subject and year to generate report 4. Claim reports are created
Secondary scenario	If no claim is made in a particular year for a particular faculty and a particular subject, there will be no report to show.
Post-condition	Report is created

Use case id	019
Use case title	Submit EC claims
Actor	Student
Pre-condition	Student must be logged into the system
Primary scenario	1. Student goes to dashboard and opens Main Menu 2. Student selects Create new claim 3. Input all claim details 4. Upload evidence 5. Submit claim
Secondary scenario	1. If the last date for evidence submission is passed, evidence can no longer be submitted. 2. If the closure date for the assessment is passed claim cannot be submitted.
Post-condition	Claim is submitted

Use case id	020
Use case title	Upload EC claim evidence
Actor	Student
Pre-condition	Student must be logged into the system
Include	Submit claim
Primary scenario	1. Student goes to dashboard and opens Main Menu 2. Student opens Claim list view 3. Student clicks on edit 4. Upload evidence

Secondary scenario	1. If the last date for evidence submission is passed, evidence can no longer be submitted. 2. If the closure date for the assessment is passed claim cannot be submitted.
Post-condition	Evidence is uploaded

Initial Class Diagram

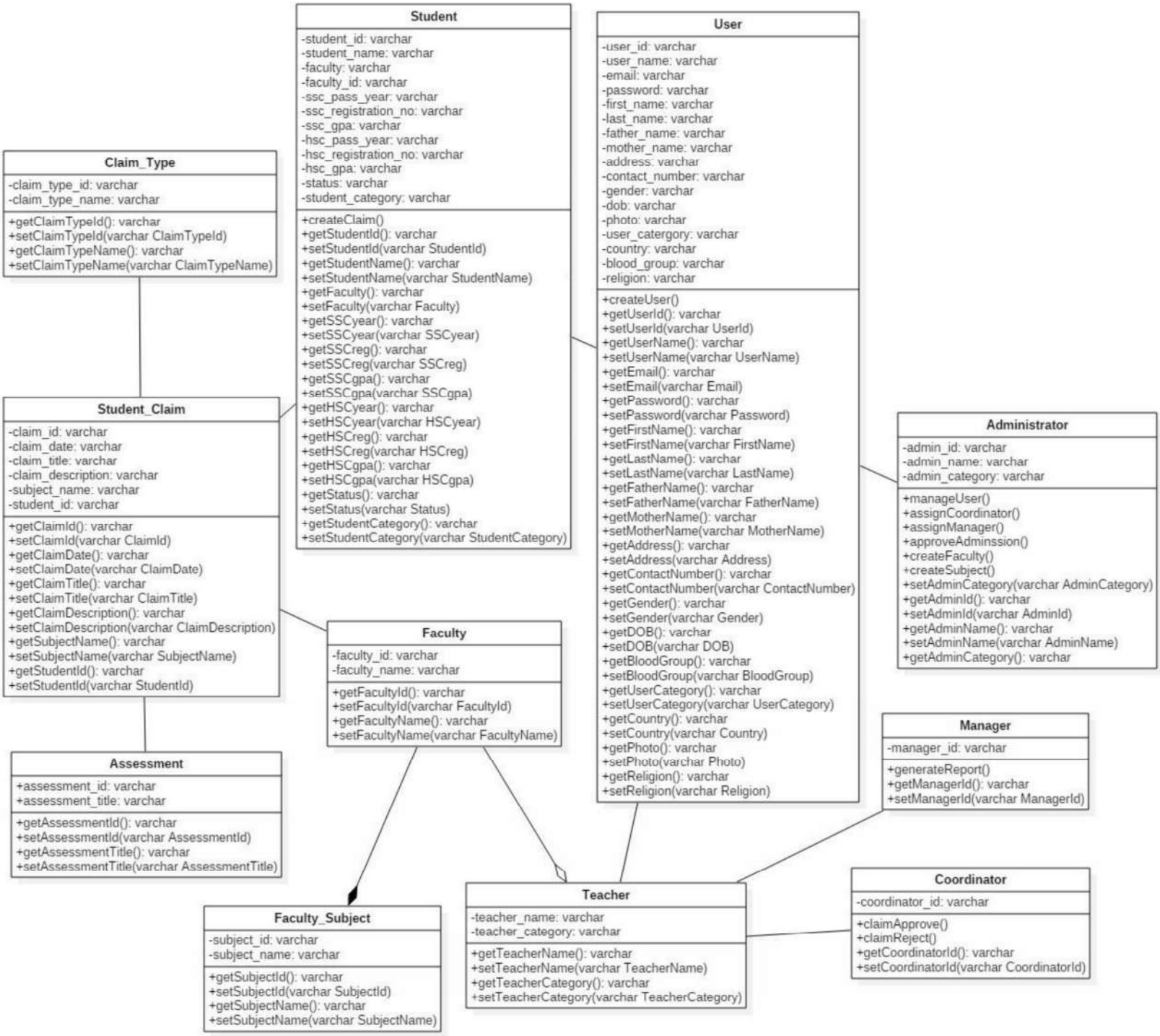

Figure 20: Initial Class Diagram of the System

Appendix B: Workflow Implementation

Activity Diagram

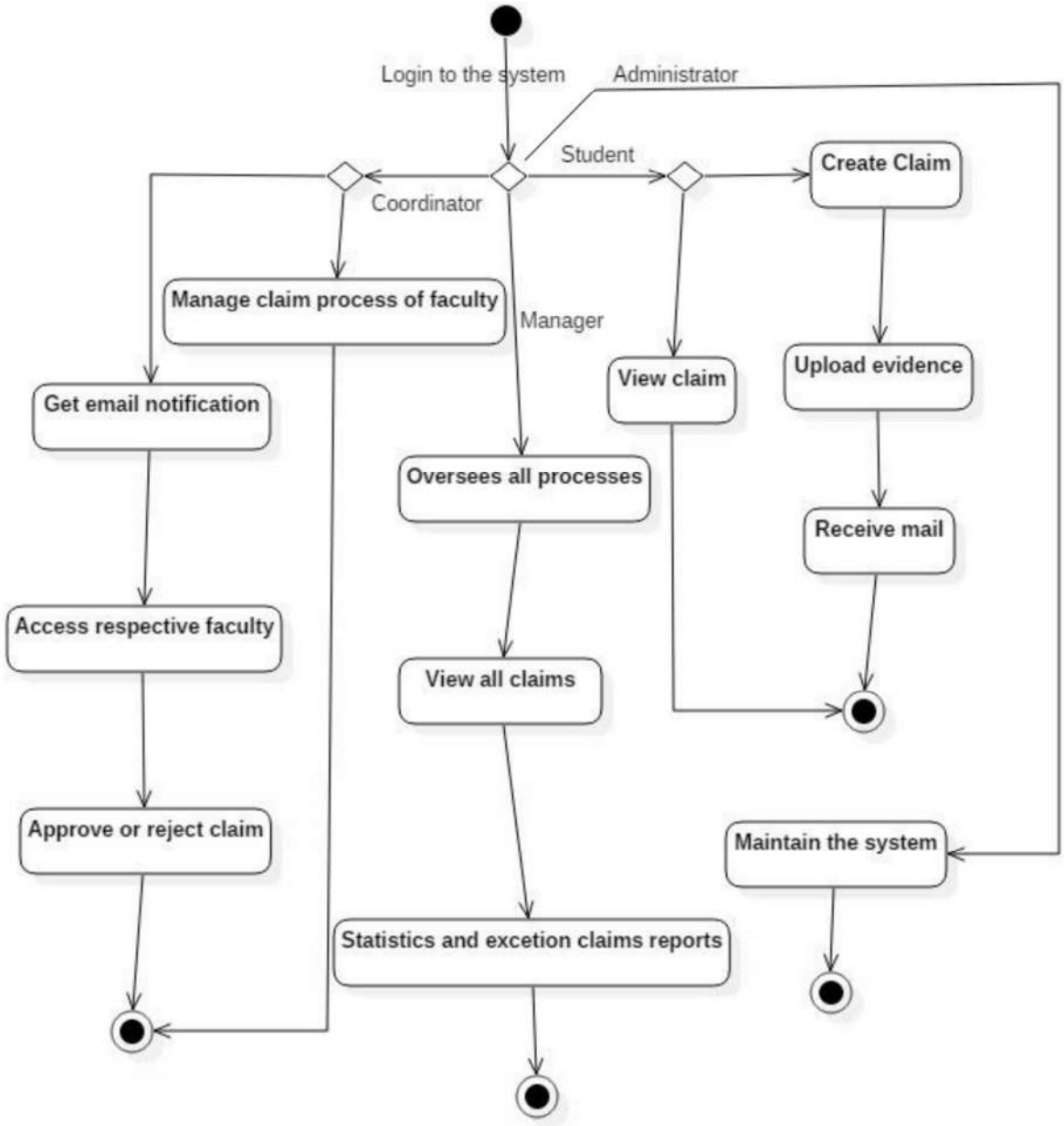

Figure 21: Activity Diagram of the System

Site Maps

Admin site map

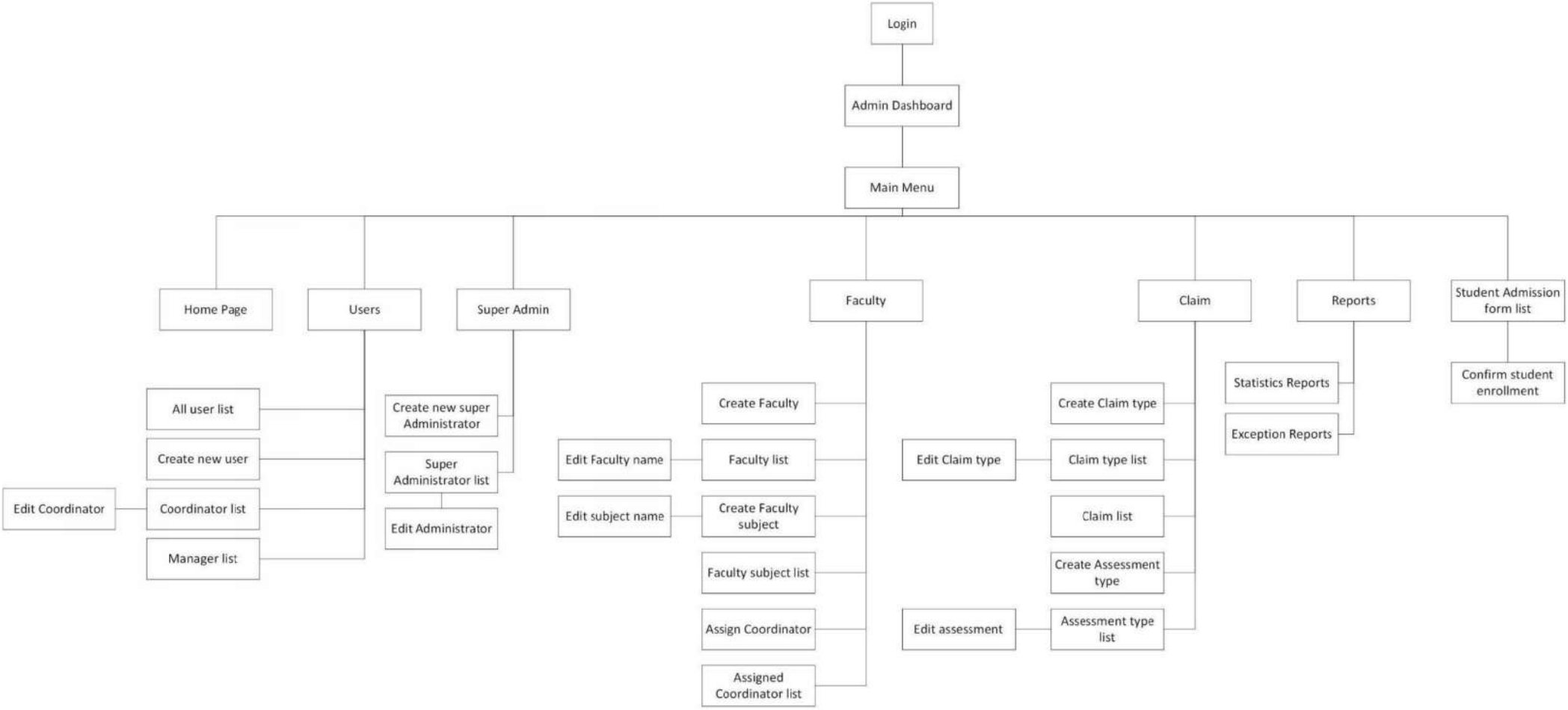

Figure 22: Admin Site Map

EC Coordinator site map

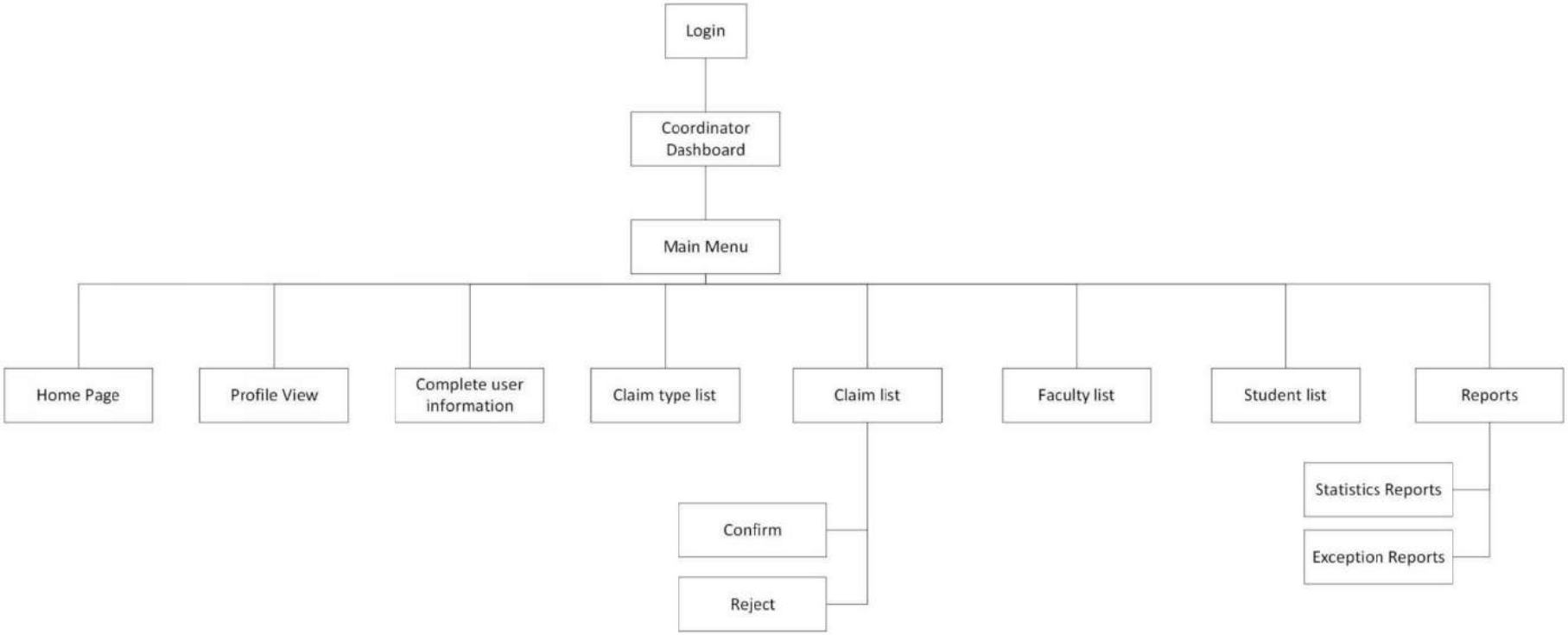

Figure 23: EC Coordinator Site Map

Manager site map

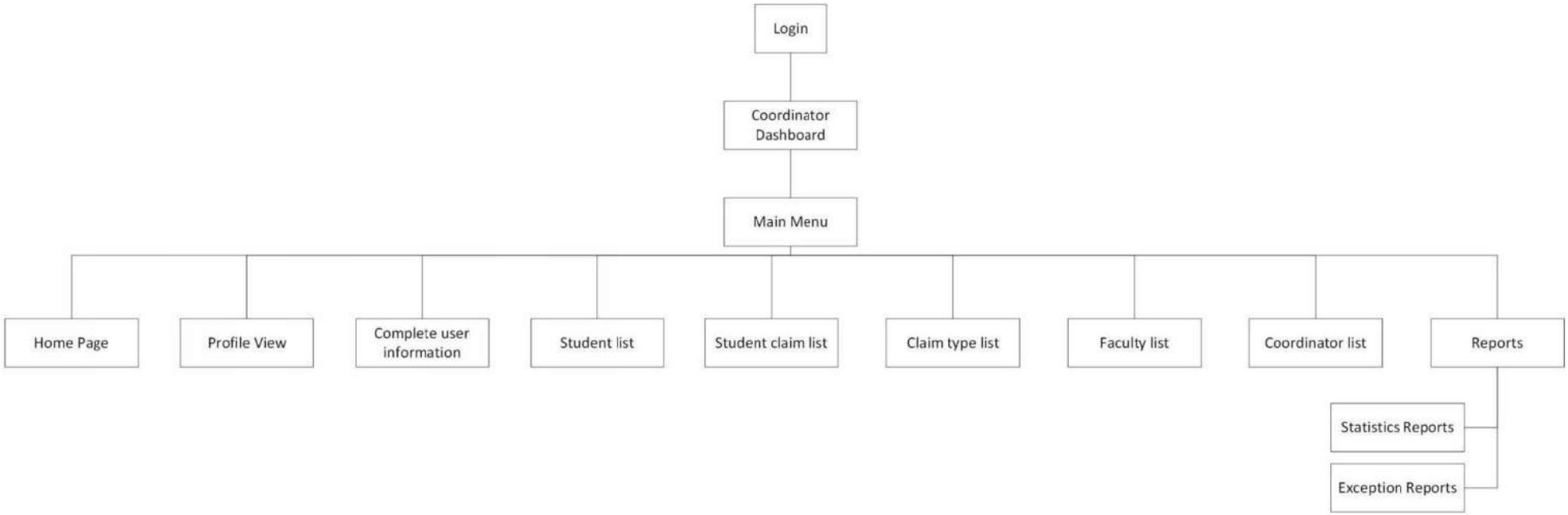

Figure 24: EC Manager Site Map

Student site map

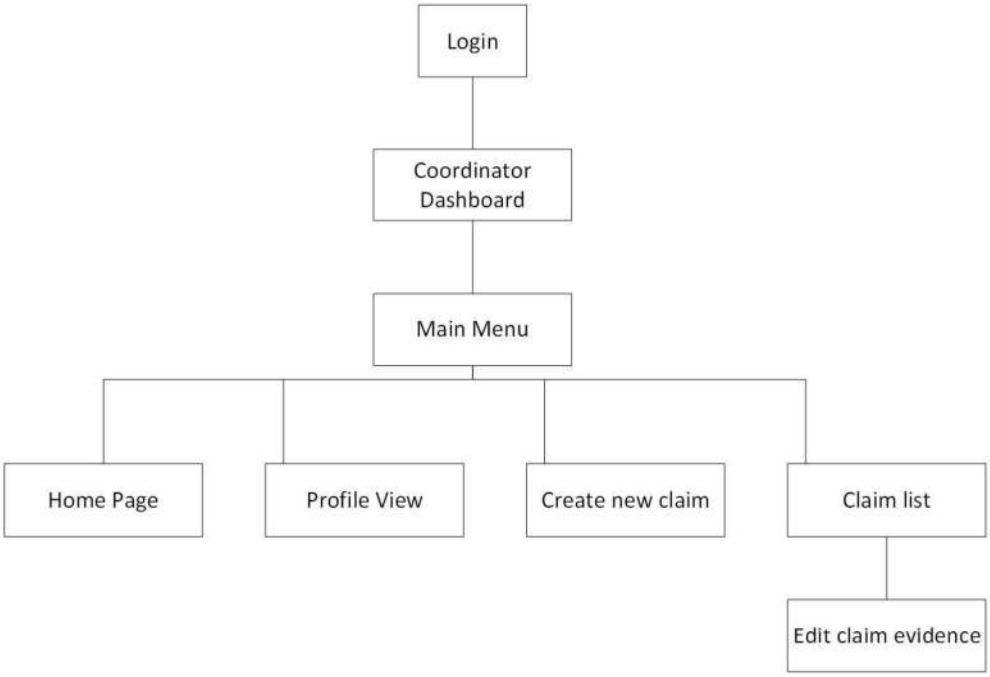

Figure 25: Student Site Map

Appendix C: Minutes of Meetings

Meeting Objectives	Duration (in minutes)	Frequency (in days)
Defining and distributing the roles among team members	20	1
Analyzing and specifying the requirements of the system	30	3
Designing UML diagrams	30	7
Designing the database, includes- EERD, data dictionary, mapping and normalized form	20	3
Interface design respective to the requirements of the system	20	7
Development of the application	60	21
Determining user limitations for a role-based security system	20	3
Determining test plan and developing test log	30	7

Below are details of some of the meetings-

Date of meeting:	Time:	**4:30PM – 5:30PM**
February,13,2017	**Location:**	DIA
Meeting Objective:		
Discussion on team roles		
Attendees:		
1. Shahnewaj Muhammad Shakil 2. Abdullah Al Masum 3. Zubair Jony 4. Mhm Munem Maruf-ur-Rahman		
Meeting Outcome:		
Roles were distributed among the team		

Date of meeting:	Time:	**4:30PM – 5:30PM**
February,18,2017	**Location:**	DIA
Meeting Objective:		
Analyze the system		
Attendees:		
1. Shahnewaj Muhammad Shakil 2. Abdullah Al Zubair 3. Mhm Munem Maruf-ur-Rahman		
Meeting Outcome:		
System Analysis and login interface have been decided upon		

Date of meeting:	Time:	**4:30PM – 5:30PM**
February,22,2017	**Location:**	DIA
Meeting Objective:		
Database architecture		

Attendees:
1. Shahnewaj Muhammad Shakil 2. Abdullah Al Zubair 3. Abdullah Al Masum 4. Mhm Munem Maruf-ur-Rahman
Meeting Outcome:
Database structure and related components have been decided upon

Date of meeting:	Time:	**4:30PM – 5:30PM**
March,15,2017	**Location:**	DIA
Meeting Objective:		
Discuss Test plans		
Attendees:		
1. Shahnewaj Muhammad Shakil 2. Abdullah Al Zubair 3. Abdullah Al Masum 4. Mhm Munem Maruf-ur-Rahman		
Meeting Outcome:		
Test plan and test log has been decided upon		

Date of meeting:	Time:	**4:30PM – 5:30PM**
March,22,2017	**Location:**	DIA
Meeting Objective:		
Discuss changes to be made in the EC Coordinator's role		
Attendees:		
1. Shahnewaj Muhammad Shakil 2. Abdullah Al Zubair 3. Abdullah Al Masum 4. Mhm Munem Maruf-ur-Rahman		

Meeting Outcome:
Decided that EC Coordinator's role must include some exceptional EC claims to be auto-processed

Date of meeting:	Time:	4:30PM – 5:30PM
March,25,2017	**Location:**	DIA
Meeting Objective:		
Review role based limitations for the users		
Attendees:		
1. Shahnewaj Muhammad Shakil 2. Abdullah Al Zubair 3. Abdullah Al Masum 4. Mhm Munem Maruf-ur-Rahman		
Meeting Outcome:		
Changes in the EC Coordinator and Admin access limitations have been decided upon		

Date of meeting:	Time:	4:30PM – 5:30PM
April,01,2017	**Location:**	DIA
Meeting Objective:		
Review the system to check if it completed		
Attendees:		
1. Shahnewaj Muhammad Shakil 2. Abdullah Al Zubair 3. Abdullah Al Masum 4. Mhm Munem Maruf-ur-Rahman		
Meeting Outcome:		
The quality of the system was discussed and probable further development was decided upon		

Appendix D: Scrum Documentation

User Story

Serial No.	User Type	User Story	Story Type
1	All Users	As a User, I want to login to the system securely.	Story
2	EC Manager	As an EC manager, I want to generate reports, so that I can check the statistics and exceptions on EC claims made annually.	Story
3	EC Coordinator	As an EC Coordinator, I want view all EC claims made by students in my Faculty.	Story
		As an EC Coordinator, I want to provide a decision for a claim.	
4	EC Coordinator	As EC Coordinator, I want email notification facilities so that I can be	Story

		notified of the claims made in my Faculty.	
5	Student	As a student, I want to submit EC claims with evidence for any item of assessment.	Epic
6	Student	As a student, I want notification facilities so that I can be notified of the decision made against my claim.	Story
7	Student	As a student, I want to view all my past and present EC claims and the decision made for the claims.	Story
8	Administrator	As an Administrator, I want to manage the system, so that I can add and/or remove users based on their current status at the university.	Epic

9	Administrator	As an Administrator, I want to manage Faculty information and Assessment information.	Epic
10	Administrator	As an Administrator, I want to assign EC Coordinators to Faculties.	Story

Product Backlog

Number	As a/an	I want to…	so that…	Priority
1	User	login to the system securely	I can access my account.	High
2	EC Manager	generate reports	I can check the statistics and exceptions on EC claims made annually.	Medium
3	EC Coordinator	view all the claims including processed and new (unprocessed claims)		High

4	EC Coordinator	be able to approve or reject a claim		High
5	Student	be able to submit EC claims for any item of assessment.		High
6	Student	upload image file or pdf document	so that it will act as the supporting evidence for my EC claim.	High
7	Student	view all my claims	I can check the processed statuses of the claims I made	Low
8	Administrator	be able to add, update and delete user and user information		Medium
9	Administrator	be able to create faculty		Low
10	Administrator	assign EC Coordinators for faculties	they can process student EC claims	Medium
11	Administrator	create and manage closure dates of assessment	users are aware of the last date of claim submission, last date of evidence submission for the claims and last	High

			date of processing a claim.	
12	Administrator	be able to confirm student registration	enroll a student into the system.	Low

Sprint Backlog

Key note- test cases were written prior to development

Sprint No.	Tasks	Volunteer	Timeline	Hours	Status
1	Analysis of **login** system	Maruf	Day 1	1	Completed
	Create database and **Users** table	Shakil	Day 3	4	Completed
	Create **login** interface	Masum	Day 3	6	Completed
	Test and review	Zubair	Day 1	2	Completed

Sprint No.	Tasks	Volunteer	Timeline	Hours	Status
2	Analysis of **roles** in the system	Maruf	Day 2	4	Completed
	Create **Admin** table in the database	Shakil	Day 3	1	Completed
	Create dashboard for **Admin**	Masum	Day 4	9	Completed
	Test and Review	Zubair	Day 2	2	Completed

Sprint No.	Tasks	Volunteer	Timeline	Hours	Status

3	Analysis of **Student enrollment** into the system	Maruf	Day 2	3	Completed
	Create **Student** table in the database	Shakil	Day 3	1	Completed
	Create Student **Admission Form** interface	Masum	Day 4	8	Completed
	Test and Review	Zubair	Day 2	4	Completed

Sprint No.	Tasks	Volunteer	Timeline	Hours	Status
4	Analysis of **Coordinator's** role and **Manager's** role in the system	Maruf	Day 2	4	Completed
	Create Coordinator table and Manager table in the database	Shakil	Day 3	0.5	Completed
	Create **dashboard** for Coordinator and Manager	Masum	Day 4	10	Completed
	Test and Review	Zubair	Day 2	4	Completed

Sprint No.	Tasks	Volunteer	Timeline	Hours	Status
5	Analysis of **EC claims** as a system	Maruf	Day 2	3	Completed
	Create **Student Claim** table and	Shakil	Day 3	4	Completed

	Claim Type table in the database				
	Create EC claim form interface	Masum	Day 4	9	Completed
	Test and Review	Zubair	Day 2	2	Completed

Sprint No.	Tasks	Volunteer	Timeline	Hours	Status
6	Analysis of **Faculty** and **Assessment**	Maruf	Day 2	4	Completed
	Create **Assessment Type** table, **Faculty** table and **Faculty Subject** table in the database	Shakil	Day 3	4	Completed
	Create forms to **Assign** Faculty, Subject and Assessment in the **Admin's** interface	Masum	Day 4	9	Completed
	Test and Review	Zubair	Day 2	1	Completed

Sprint No.	Tasks	Volunteer	Timeline	Hours	Status
7	Analysis of **Claim Reports** and **Exception** Reports	Maruf	Day 2	4	Completed
	Create **Report forms** for Admin, Manager and Coordinator	Masum	Day 3	10	Completed
	Test and Review	Zubair	Day 2	4	Completed

Sprint burndown Chart

Key note- the highlighted blue circles represent tasks remaining after every sprint week