

**CIE203**

**Software Engineering**

Project Name: BookIt

Software Requirements Specifications

Team Name: BugBusters

Team Leader: Mahmoud ElSayed

Month & Year

**Contents**

**Contents 2**

**Team 3**

**Document Purpose and Audience 3**

**Introduction 3**

**Software Purpose 3**

**Software Scope 3**

**Definitions, acronyms, and abbreviations 4**

**Requirements 5**

**Functional Requirements 5**

**Non Functional Requirements 8**

**System Models 9**

**Use Case Model 9**

**Use Case Tables 10**

**Ownership Report 34**

**Policy Regarding Plagiarism: 30**

# 

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 201304903 | Mahmoud ElSayed Abdelhameed | s-mahmoud.abdelhameed@zewailcity.edu.eg | 01285301109 |
| 201304766 | Omar Ibrahim Hagrass | s-omar.hagrass@zewailcity.edu.eg | 01095541958 |
| 201304126 | Rohanda ElSayed Abdallah | s-rohanda.abdallah@zewailcity.edu.eg | 01009644655 |

# Document Purpose and Audience

-This document describes Software requirements specifications for a class reservation software. It includes functional, non-functional requirements and full use case model of the system and all its features. It is intended to help developers to organize, design and test the system. it also serves as a contract between the clients and the developers. The document is intended for client, system designers and users.

# Introduction

## Software Purpose

The Software aims mainly to manage class reservation process in Zewail city for science and technology. It makes things easier for the registrar employees to handle reservation requests more efficiently and it gives the best possible class room list according to the user's requirements and the current available classes.

## Software Scope

BookIt contain a calendar that keeps track of the available and occupied class rooms at any time, it also holds a database of all the classes with their specifications (equipment and the number of students the class can hold)

It allow all users to view the list of available class rooms at any time. It also allows premium users to request a room with certain requirements at a determined time and returns a list of the best available rooms that meets his requirements.

It can also handle conflicts between reservation according to a predefined priority algorithm based on the type of the user and the reservation purpose.

## 

## Definitions, acronyms, and abbreviations

**Schedule shifting**: shift all lectures and tutorials by a specified time in a certain day.

**Priority algorithm**: It is the algorithm used to prioritize the reservation requests by using the user university position and the request purpose.

**Class room data base:** it is the data base which hold all the reservation through the whole semester , for Each day each time slot holds the IDs of the reservation that were done on that time slot and the IDs of the waiting reservation requests (waiting list).

**Reservation data base :** it is the data base which hold all the reservation it contains the reservation IDs, priority of the request , the user ID who did the request,status whether its reserved or waiting.

**Users data base :**  it holds the user IDs, university position ,email , phone number

and also the reservation IDs for the request done by the user.

**Administrators:**  are the registrar employees and are responsible for managing class reservation system.

**Premium users :** Doctors , Teaching assistance and Major representatives

**User (Zc member):**  students

**Emergency request :** Is a request sent by the premium user when no rooms are available and he wants to replace an existing reservation.

# Requirements

## Functional Requirements

NOTE : the number between the brackets is the expected complexity level of the function

**1. recurring booking request: (2)**

Administrators can request a recurring booking request which always to add ,replace or cancel requests through the whole semester.( reserve the same room for the same time slot each week )

Booklet then stores the reservation through the whole semester.

This feature can be used for example to upload the semester schedule or updating the the current schedule.

**2. Registration: (1)**

BookIt allows users to create an account using their Zewailcity email, password and university position.

BookIt store user ID, name, password , user university position , email and phone number in the users' data base upon registration.

**4. Log in: (2)**

BookIt will allow registered users to log in using their user name and password.

**5. Classes List: (3)**

BookIt allow all users who log in to view the list of free classes at any specified time, even non-premium users can benefit from this list by for example searching for an empty room to study.

**6. Class reservation: (1)**

BookIt allows premium users to request a reservation of a room after specifying the time slot, equipment needed, number of students and the purpose of the reservation. After that a list of available class rooms with the specifications chosen appear for the premium user to choose from.

-Equipment are chosen from an optional check list that contains: projector, computer, microphone and smart board.

-The user can choose one of the reservation purposes from the following:

Lecture – Tutorials – Exams – Seminars – Meetings – Student activities – Others.

The reservation purposes list will appear according to the user’s position in the university.( for example only professors will have the option of reserving a room for lecture purpose.)

-The time slot can be chosen from either a calendar or by entering the date.

**7. Room suggestions: (4)**

BookIt should compare the user requirements and shows a list of the classes that best meets his requirements, then the user chooses from that list the appropriate class room and then user confirm the reservation , BookIt should confirms the room reservation from the user and mark the room as reserved in the calendar data base. It also gives this reservation request an ID and saves it in the reservation data bases .

Note: request priority is calculated using the reservation purposes and the user university position.

**8. Cancel reservation: (3)**

Users should be able to cancel reservation that they made. And the system then update the class room database after the cancellation.

**10. Conflict handling: (5)**

BookIt should handle when no rooms are available to meet the user requirement by giving the user a list of available room in the nearest time slot.

The user can either confirm, cancel or request an emergency request .

The user can also choose to put the request in the waiting list of this time slot.

If the user chooses to send an emergency request the system then should use the priority algorithm and show a list of the rooms with the least priority reservation that meets the user requirements as a replacement for the current request (however this room reservation priority should be less that or equal to the current request priority).

-The user chooses from this list a certain room .

- BookIt then send this emergency replacement request to administrators to confirm replacement of the room.

-if replacement was confirmed , the new reservation is saved and a notification is sent to both users, then the software offers the replaced party suggestions that best meet his requirement , but in another time slot (the nearest time slot) along with an apology .and also the option to put the request in the waiting list.

**11. Schedule shift handling: (5)**

The administrators can request “schedule shifting” in a specific day by certain number of hours.

A shift will be done in the specified date by the specified hours( I.e all reservations that were done by recurring booking (permanent reservations) will be shifted by certain hours and then all non permanent reservations that were done on that day will be removed and then reinstalled after the shift and also users with reservations on the waiting list will be notified if a room became available for them.

All requests which cannot be reinserted will go to the waiting list and given the highest priority.

**12. Notification system: (4)**

Premium users can choose to send a notification about a new reservation to a given list of other users. The system should then broadcast the notification to the selected list through the local network that connect the users together.

Also BookIt can notify the users if a schedule related to them was updated or if any of their reservation requests was canceled or approved.

**13.System clean up: (3)**

Every Day BookIt checks which requests have timed out and remove them form the data bases .

**14- add new class rooms to the class room database(1)**

## Non Functional Requirements

1. Usability:

Simple to use: the software has a user friendly interface that makes it easy for almost anyone to use.

1. Reliability:

their should be no crash for 90% of the runtime.

1. Performance:

Fast response: This software application shouldn’t exceed 5 seconds as a response time.

1. Supportability:

Market customization: This software application will be built to work on the ZewailCity regime only.

1. Implementation:

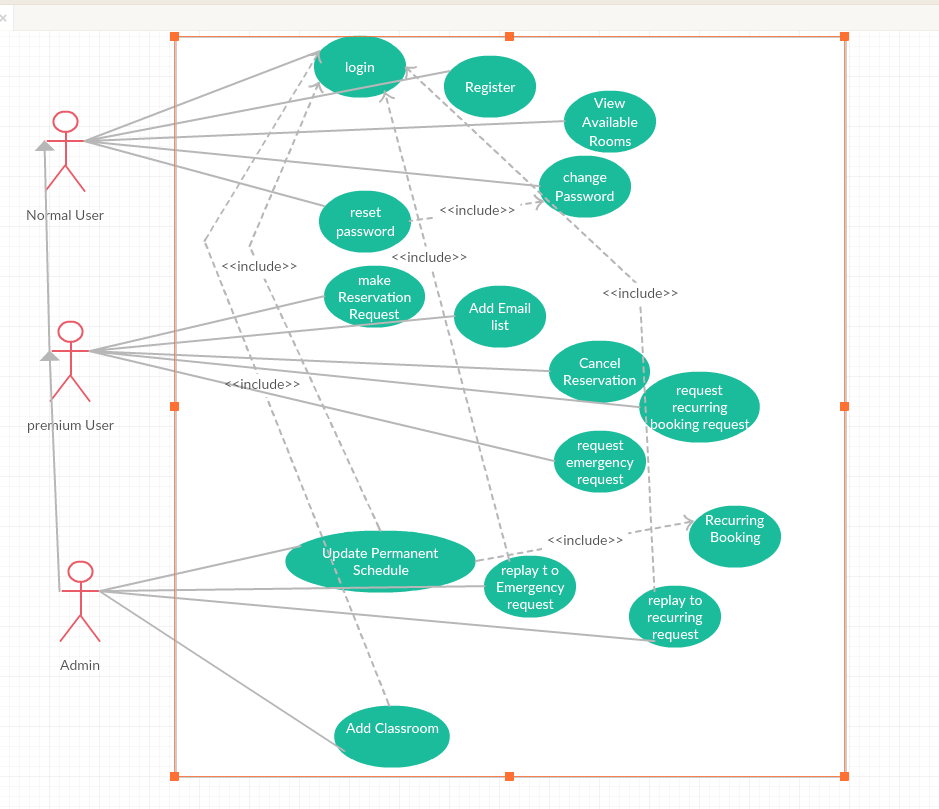
Programming Language: This software application will be implemented using Java.

6- Data Base Backup : all Data bases will have a backup if any error happened

we can retrieve the data from the backup.

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Register | |
| Actors: | Normal user – premium user – Admin | |
| Pre-conditions: | The BookIt App has started. | |
| Post-conditions: | The actors can log in into the app. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User enters full name, ID,email address and faculty position. |  |
|  | 2- System sends a confirmation email to the actor. |
| 3- User read the email and confirm the registration process. |  |
| Exceptions: | **User Action** | **System Action** |
| 1- Entered wrong ID or email format. |  |
|  | 2- ID or email is not identifiable. |
|  | 3- The email address, ID and name don’t belong to the same person on the data base. |  |
|  |  | 4- Shows error message to notify the user that The email address, ID and name don’t belong to the same person on the data base and to fix the problem please contact the admin |
| Includes: |  | |
| Notes and Issues: | Every email can only have one account on BookiT. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Log in | |
| Actors: | Normal user – premium user – Admin | |
| Pre-conditions: | The BookIt App has started and the user is a registered user. | |
| Post-conditions: | The actors are ready to use the features of the BookIt application, also actors are recorded in the system as active at the moment. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User enters user name ( or email Address) and Password. |  |
|  | 2- System Verifies user data and allows logging in. |
| Exceptions: | **User Action** | **System Action** |
| 1- User enters wrong user name. |  |
|  | 2-System Shows a message “This User name doesn’t exist” |
|  | 3- User enters right user name , but wrong password. |  |
|  |  | 4-System shows a message “Wrong password” |
| Includes: |  | |
| Notes and Issues: | User can’t log in from two different devices with the same account at the same time. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Change Password | |
| Actors: | Normal user – premium user – Admin | |
| Pre-conditions: | The BookIt App has started and the user is registered on the system. | |
| Post-conditions: | The user Password changed. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User chooses the “change password” command and enters his old password and his new password. |  |
|  | 2- System verifies his old password and change the password to the new password |
| Exceptions | **User Action** | **System Action** |
|  | 1-user enter the same old password |  |
|  |  | 2-System shows a message to tell him to enter a new password |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Reset Password | |
| Actors: | Normal user – premium user – Admin | |
| Pre-conditions: | The BookIt App has started and the user is registered on the system. | |
| Post-conditions: | The user is ready to log in. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User chooses the “reset password” command and enters his user name. |  |
|  | 2- System sends an email to this user with a random password. |
|  | 3- User uses this password to login . |  |
|  | 4-User calls change password |  |
|  |  | 5- System updates the user’s new password in the database. |
| Exceptions: | **User Action** | **System Action** |
| 1- User enters wrong user name. |  |
|  |  | 2-System Shows a message “This User name doesn’t exist” |
|  | 3- User chooses the new password the same as the old password. |  |
|  |  | 4- System rejects request and shows a message “The new password is the same as the old password”. |
| Includes: | Change password | |
| Notes and Issues: | If the user requests many resets in a short period of time admins should be alerted because it might be an attack or a virus. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | View available rooms | |
| Actors: | Normal user – premium user – Admin | |
| Pre-conditions: | The BookIt App has started and the user has logged in. | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1- User chooses the command to view the available class rooms list at any time specified by the user. |  |
|  | 2- System shows the user the list of the available rooms at the specified time entered by the user. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: | If there are no available room the system returns a message saying “There are no available rooms at that time”.  Normal users can benefit from this feature because they can use it to search for an empty room to study or whatever. | |

|  |  |  |
| --- | --- | --- |
| * Case ID | 6 | |
| Use Case Name: | Make class reservation request | |
| Actors: | premium user – Admin | |
| Pre-conditions: | The BookIt App has started , the user has logged in and the calendar is displayed | |
| Post-conditions: | A reservation notification is sent to the user | |
| Flow of events: | **User Action** | **System Action** |
| 1-User chooses the date and time for his reservation from the displayed calendar |  |
|  | 2- System displays the available options of classroom capacity and the equipment in a checklist form and asks the user to mark his requirements. |
|  | 3-Users chooses the number of students ,purpose and needed equipment from the list if the user has no specific requirements he can choose “open” |  |
|  |  | 4-System displays (suggests)the classrooms that are available at the specified date and that meet the user requirements .If the user has chosen “open “ the system displays all the available classes the meet the date and time specified . |
|  | 5-The user chooses one of the suggested classes and thus makes a reservation request |  |
|  |  | 6-The system checks that the user has made only one request at a specified time  and apply the reservation request . And message appears:  “Thank you for using BookIt ,your request is done” |
|  |  |  |
|  |  | 7-The system asks the user if he wants to notify the audience with the reservation and asks him to choose the mail list entered previously or to enter a new mail list. |
|  |  | 10- The system sends an update to all mails specified by the users |
| Exceptions: | **User Action** | **System Action** |
|  | 1-If all classes are booked at a certain date and time ,the system asks the user if wants to be on the waiting list of on this time slot or send emergency request while showing him the list of rooms that fits his request in the nearest time slot |
|  | 2a. if user chooses waiting list |  |
|  |  | 3a.System puts the user on a waiting list and notifies him if cancellation or shift happened that makes his request available an this the specified slot before the desired date (see notes) and that his request was done. |
|  | 2b.if user chooses the emergency request |  |
|  |  | 3b. the system then uses the priority algorithm and show a list of the rooms with the least priority reservation that meets the user requirements as a replacement for the current request (however this room reservation priority should be less that or equal to the current request priority). |
|  | 4.-The User chooses from this list a certain room. |  |
|  |  | 5.Request is sent to Admins |
|  |  | 6. Administrator reply to emergency request(included) |
| included | Reply to emergency request (use case ID 12) | |
| Notes and Issues: | If the admin rejected the registration request the system gives him the option to suggest another classroom for the user .  The user can cancel the reservation after the notification if he doesn't want it any more. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | Request recurring booking | |
| Actors: | Administrators | |
| Pre-conditions: |  | |
| Post-conditions: | Send a recurring booking request to the admins | |
| Flow of events: | **User Action** | **System Action** |
| 1- User request log in (included) |  |
|  | 2-User specifies the needed reservation parameters (data , time , purpose,number of students) and click on request recurring booking |  |
|  |  | 3-System asks for confirmation |
|  | 4-User chooses confirm |  |
| Includes: | Log in(ID 2) | |

|  |  |  |
| --- | --- | --- |
| Case ID | 8 | |
| Use Case Name: | Cancel reservation | |
| Actors: | premium user – Admin | |
| Pre-conditions: | The BookIt App has started , the user has logged in and previous registration was made and displayed on the user profile . | |
| Post-conditions: | The classroom is now available | |
| Flow of events: | **User Action** | **System Action** |
| 1-User enters to his profile and click on the registration he made |  |
|  | 2- System displays the actions that the user could do : cancel reservation ,see details |
|  | 3-User chooses cancel registration |  |
|  |  | 4-System cancels the registration, notify the admin and displays a message to users “registration is cancelled”  And the registration disappears from the profile |
|  |  | 5-The system asks the user if he wants to update the attendees with the registration cancellation |
|  |  | 6-If yes the system sends a notification to all mails registered by the user |
|  |  | 7-The system checks if there are users on the waiting list who can take this class and sends a notification that their request is now done . |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |

|  |  |  |
| --- | --- | --- |
| Case ID | 9 | |
| Use Case Name: | Email list entry | |
| Actors: | premium user – Admin | |
| Pre-conditions: | The BookIt App has started , the user has logged in | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-The user log in to his profile and chooses to update email list |  |
|  | 2- System asks the user if he wants to enter the mails manually or to ask the registrar to send the registered student mailing list |
|  | 3-User chooses one of the available options |  |
|  |  | 4a-If the user chooses to enter the mails manually, the system provides an entry field to enter all mails  4b-if the user chooses to request the mail list from the registrar the system sends a request to the registrar that includes the user name and purpose  And display a message to the user that his request was sent . |
|  | 5a-The user enters the mail list or the registrar sends it |  |
|  |  | 6-The system asks the user if he wants to create a group with the provided mail |
|  |  | 7-The system stores the mails in its database |
| Exceptions: | **User Action** | **System Action** |
| 1-If the user teaches the same course as a previously registered one he could ask the software to create the same mail list |  |
|  |  | 2-The system checks that the user is eligible to access the mailing list of the course and copy the mail list on the user profile |
| Includes: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 10 | |
| Use Case Name: | Recurring booking (inclusion) | |
| Actors: | Administrators | |
| Pre-conditions: |  | |
| Post-conditions: | A reservation is done or removed in certain time slot of the week for a certain class room through the whole semester. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User will request a semester recurring booking. |  |
|  | 2- System shows a one week calendar with days as columns and time slots as rows. |
| 3- User select a cell in the calendar |  |
|  |  | 4- The system shows a dialogue to choose both the room number and the purpose (lecture -tutorial-student activity-others) or whether to remove any reservation in this room at that time slot. |
|  | 7-User chooses the room number and the purpose. |  |
|  |  | 8- Dialogue box disappears and System apply the request. |
| Exceptions: | **User Action** | **System Action** |
| 7-User chooses the room number and the purpose |  |
|  | 8-System display a message that the room is already reserved and shows a dialogue box whether to confirm replacement or cancel. |
|  | 9.a If user chooses to cancel. |  |
|  |  | 10.a-Dialogue box disappears and nothing happens. |
|  | 9.b if user chooses to confirm. |  |
|  |  | 10.b System apply the replacement and Dialogue disappear |
|  |  | 11-System send notification about the cancellation to the affected users. |
| Exceptions: | **User Action** | **System Action** |
|  | 7- User select remove reservation from a class |  |
|  |  | 8- Dialogue box disappears and system applies the request |
|  |  | 9-The system checks if any reservation requests in the reservation waiting list can be inserted in this room and send a notification to the request owner that a room is available for his request now to let the user chooses to confirm or remove his request from waiting list. |
| Includes: |  | |
| Notes and Issues: | Affected users are the users who were in the replaced class | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 11 | |
| Use Case Name: | Up date Semester Schedule | |
| Actors: | Administrators | |
| Pre-conditions: | The Semester schedule should be already done and ready to be entered to the system and at least one administrator should be registered. | |
| Post-conditions: | The Class room Calendar data base will add,remove a new permanent class to the semester schedule ( i.e. scheduled classes as reserved through the whole semester). | |
| Flow of events: | **User Action** | **System Action** |
| 1- User request log in (included) |  |
| 2- User call Recurring Booking (included) for the needed class room with the needed request either to add or remove recurring reservation |  |
|  |  | 3- Recurring Booking ends and the system will still be showing the Week Calender |
|  | 5-User call Recurring Booking (included) until all classes are entered to the system. |  |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: | Recurring Booking(ID 8) , Log in(ID 2) | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 12 | |
| Use Case Name: | Schedule Shift request | |
| Actors: | Administrators | |
| Pre-conditions: | A Semester schedule should be uploaded | |
| Post-conditions: | One day will be shifted by a specified hours | |
| Flow of events: | **User Action** | **System Action** |
| 1- User request log in (included) |  |
| 2- User request schedule shift . |  |
|  |  | 3-The system will show a dialogue box to choose the date of shift and number of slots to shift ( i.e 2 slots means shift down two cells in this date ) |
|  | 4-User choose the date and the number shifting slots . |  |
|  |  | 5-System show a dialogue to confirm or cancel the shift. |
|  | 6- User confirms the request. |  |
|  |  | 7-System apply the changes by shifting the time slot of all the reservations |
|  |  | 8-System Notifies all affected users with the shift . |
| Exceptions: | **User Action** | **System Action** |
| 6- User Cancel the request |  |
|  | 7-Dialogue box disappears. |
| Includes: | Log in (ID 2) | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 13 | |
| Use Case Name: | Reply to Emergency request | |
| Actors: | Administrators | |
| Pre-conditions: | An Emergency should have been received | |
| Post-conditions: | Emergency Request either approved or canceled | |
| Flow of events: | **User Action** | **System Action** |
| 1- User request log in (included) |  |
|  | 2-System shows a notification about the emergency request. |
|  | 3-User chooses to answer the request |  |
|  |  | 4-System show the request body in a dialogue box to choose whether to confirm or cancel. |
|  | 5-User chooses confirm |  |
|  |  | 7-System apply the request and send notification to the affected user ( the user whose reservation was canceled) and also send an approval notification to the user who sends the emergency request . |
|  |  | 8- Dialogue box close. |
| Exceptions: | **User Action** | **System Action** |
| 5- User chooses to cancel |  |
|  | 7-Dialogue box disappears and refuse notification is sent to the user who sends the emergency request. |
| Includes: | Log in(ID 2) | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 14 | |
| Use Case Name: | Reply to recurring request | |
| Actors: | Administrators | |
| Pre-conditions: | An recurring request should have been received | |
| Post-conditions: | Recurring Request either approved or canceled | |
| Flow of events: | **User Action** | **System Action** |
| 1- User request log in (included) |  |
|  | 2-System shows a notification about the recurring request. |
|  | 3-User chooses to answer the request |  |
|  |  | 4-System show the request body in a dialogue box to choose whether to confirm or cancel. |
|  | 5-User chooses confirm |  |
|  |  | 7-System apply the request and send notification to the affected users |
|  |  | 8- Dialogue box close. |
| Exceptions: | **User Action** | **System Action** |
| 5- User chooses to cancel |  |
|  | 6-Dialogue box disappears and refuse notification is sent to the user who sends the recurring request. |
| Includes: | Log in(ID 2) | |
| Notes and Issues: | Cancel | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 15 | |
| Use Case Name: | Add Class room | |
| Actors: | Administrators | |
| Pre-conditions: | An Admin should log in | |
| Post-conditions: | An new class room is added to the data base | |
| Flow of events: | **User Action** | **System Action** |
| 1-press add new class room |  |
|  | 2-system show dialogue to enter the properties of the new class(its capacity and equipments) |
|  | 3-User fill the text boxes |  |
|  |  | 4- system shows confirmation request |
|  | 5-User chooses confirm |  |
| Includes: |  | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Software Purpose-Software Scope-Functional requirements | *All team members* |
| Non functional requirement | *Mahmoud Elsayed* |
| Use cases 1-2-3 | *Mahmoud Elsayed* |
| Use cases 3-4-5 | *Rohanda Hamed* |
| Use case 8 to 12 | *Omar Hagrass* |

# Policy Regarding Plagiarism:

**Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.**

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
5. إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر.