Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering II**

E-Comm

Software Requirements Specifications

Contents

[Team 3](#_Toc402452669)

[Document Purpose and Audience 3](#_Toc402452670)

[Introduction 3](#_Toc402452671)

[Software Purpose 3](#_Toc402452672)

[Software Scope 4](#_Toc402452673)

[Definitions, acronyms, and abbreviations 4](#_Toc402452674)

[Requirements 4](#_Toc402452675)

[Functional Requirements 4](#_Toc402452676)

[Non Functional Requirements 4](#_Toc402452677)

[System Models 5](#_Toc402452678)

[Use Case Model 5](#_Toc402452679)

[Use Case Tables 5](#_Toc402452680)

[Ownership Report 6](#_Toc402452681)

[Policy Regarding Plagiarism: 6](#_Toc402452682)

# 

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20146006 | Hussein Khaled | [husseinkk96@gmail.com](mailto:husseinkk96@gmail.com) | 01122555215 |
| 20146019 | Abdallah Ahmed | [abdallah.se95@gmail.com](mailto:abdallah.se95@gmail.com) | 01096276749 |
| 20146007 | Rana Mohamed | rana.mohamed.97@hotmail.com | 01111273318 |
| 20146016 | Ismail Sherif | [moly\_meme@hotmail.com](mailto:moly_meme@hotmail.com) | 01156596754 |

# 

# Document Purpose and Audience

* **Expected Audience : Faculty Administration, Faculty Staff and Students**

# Introduction

## Software Purpose

* **It is a web application which provides a reliable environment for students and faculty staff (Doctors & Teacher assistants) and allows them to communicate.**

## Software Scope

* **Registration System**
* **Notification System**
* **Rating System**
* **Attendance System**
* **Grading System**
* **Offering System**
* **Hub Menu**

## Definitions, acronyms, and abbreviations

* **X 🡪 Companies and organizations**

# 

# Requirements

## Functional Requirements

* Users (Admin, Students, Teachers and X) could sign in the web application
* Users (Admin, Students, Teachers and X) could sign out the web application
* Users (Admin, Students, Teachers and X) could manage notifications by viewing or clearing notifications
* Admin user could add any type of users (Admin, Students, Teachers and X)
* Admin user could remove any type of users (Admin, Students, Teachers and X)
* Admin user could manage courses ( View courses, Register courses, Drop courses, Assign courses, Add courses, Delete courses)
* Admin user could manage users (Admin, Students, Teachers and X) by adding or deleting them
* Admin user and Teachers could manage grades by viewing, editing and adding grades
* Admin user should add final and Teachers should add other grades
* System should notify students with exams, assignments and schedule timing
* System should notify doctors and teacher assistants with their lectures and sections schedule
* System should notify any user (Admin, Students, Teachers and X) with incoming messages
* Students would rate teachers
* System should calculate grades and GPA
* X would offer students with courses and internships
* Students could manage courses by registering, viewing and dropping courses

## Non Functional Requirements

* **Security**
  + **Access permissions for application data**
  + **Password requirements: length (8 characters), characters (Upper case, Lower case and Numbers)**
* **Performance**
  + **Access time: 5 secs**
  + **Processing time: 5 secs**
  + **Withdraw operation will be done within 20 second**
* **Capacity** 
  + **Year-on-year growth 50% +**
  + **Storage: Hostage size -> 5 GB**
* **Maintenance**
  + **Holidays: Summer Time**
* **Scalability**

# 

# System Models

## Use Case Model

## 

## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Sign in | |
| Actors: | Students – Teachers – Administrators – X | |
| Pre-conditions: | User Starts app | |
| Post-conditions: | User directed to his page | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enter Username and password |  |
|  | 2- System Verifies user data  3- System checks for username and password in DB  4- DB confirms username and password  5- System directs the user to his hub menu |
| Exceptions: | **User Action** | **System Action** |
| 1- User enters invalid username or password |  |
|  | 2- System check username and password  3- DB system searches for username and password  4- DB confirms that username or password is invalid  5- System sends a message for the user that username or password is invalid |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Sign out | |
| Actors: | Students –Administrators – Teachers – X | |
| Pre-conditions: | User is logged in | |
| Post-conditions: | User directed to startup page | |
| Flow of events: | User Action | System Action |
| 1- User press sign out button |  |
|  | 2- System direct user to startup page |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | View courses | |
| Actors: | Students – Administrators – Teachers | |
| Pre-conditions: | User is logged in | |
| Post-conditions: | Courses viewed to the user | |
| Flow of events: | User Action | System Action |
| 1-User press view courses button |  |
|  | 2- System sends user ID to DB system  3- DB system retrieves only courses accessed by user ID  4- DB system sends all courses to main system  5- System shows courses to the user |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Assign courses | |
| Actors: | Administrators | |
| Pre-conditions: | Admin is logged in | |
| Post-conditions: | Admin assigns the courses to Teachers | |
| Flow of events: | User Action | System Action |
| 1- Admin press assign courses button |  |
|  | 2- System view courses |
| 3- Admin user chooses the course  4- Admin user enters Teacher ID |  |
|  | 5- DB system searches for the teacher ID  6- DB system updates teacher ID of the course with the new ID |
| Exceptions: | User Action | System Action |
| 1- Admin user enters an unexciting teacher ID |  |
|  | 2- DB system searches for the teacher ID  3- DB system doesn’t find the teacher ID  4- DB system confirms that the teacher ID doesn’t exist  5- System sends a message to the admin user that the teacher is invalid |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Add courses | |
| Actors: | Admin user | |
| Pre-conditions: | Admin is logged in | |
| Post-conditions: | Admin add course(s) to the faculty | |
| Flow of events: | User Action | System Action |
| 1- Admin press add course button |  |
|  | 1. System allow admin to enter the course name and ID 2. DB system add the new course name and ID in the database |
| Exceptions: | User Action | System Action |
| 1. Admin enters an exist course name or ID |  |
|  | 1. DB system founds that the course name or ID is already exist. 2. System sends a message to the admin user that the course is already exist. |
|  | 1. Admin user enters course name in the course ID slot |  |
|  | 1. System checks course name and ID data types 2. System sends a message to the admin user that data entered incorrectly |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Register course | |
| Actors: | Students – Admin user | |
| Pre-conditions: | User is logged in | |
| Post-conditions: | User register course | |
| Flow of events: | User Action | System Action |
| 1- User press register course button |  |
|  | 2- System view available courses |
| 3- User choose the course and the student |  |
|  | 1. System checks type of user 2. if user is admin system asks for student id |
| 1. Admin enters student id |  |
| Exceptions: | User Action | System Action |
| 1. User exceeds number of hours (18 hour) |  |
|  | 1. System send warning message |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | Drop course | |
| Actors: | Administrators - Student | |
| Pre-conditions: | User pressed on manage courses | |
| Post-conditions: | User Drops course(s) | |
| Flow of events: | User Action | System Action |
| 1- User press drop course button |  |
|  | 2- System view courses registered by the student that can be dropped |
| 1. User chooses courses to be dropped |  |
|  | 1. System sends the course(s) ID(s) to the DB system 2. DB system update the database 3. DB system confirms that the course(s) are canceled 4. System sends a message to the user that the course(s) are dropped successfully. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | Add user | |
| Actors: | Administrators | |
| Pre-conditions: | Admin user pressed manage users | |
| Post-conditions: | Admin added user(s) | |
| Flow of events: | User Action | System Action |
| 1- Admin press add user button |  |
|  | 2- System direct admin to add user page |
| 3- Admin user add user’s details |  |
|  | 1. System validates the user’s data 2. System sends user’s data to DB system 3. DB system add user’s data to the database and confirms to the system that the user has been added 4. System sends a message to the admin user that the user is added correctly |
| Exceptions: | User Action | System Action |
| 1. Admin entered an existing user ID |  |
|  | 1. System sends the data to DB system 2. DB system checks user ID and found that the ID is already exist 3. DB systems send a confirmation to the system that the ID is already taken 4. System sends a message to the admin user that the ID is already taken. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Remove user | |
| Actors: | Administrators | |
| Pre-conditions: | Admin pressed manage users | |
| Post-conditions: | Admin remove user(s) | |
| Flow of events: | User Action | System Action |
| 1. Admin user press remove user button 2. Admin user enters user’s ID |  |
|  | 1. System checks the user’s ID 2. System sends the user’s ID to DB system 3. DB system searches for the user’s ID 4. DB system removes all data of the user with this ID 5. DB system sends a confirmation to system 6. System sends a message to admin user that the user is removed |
| Exceptions: | User Action | System Action |
| 1. Admin user enters username |  |
|  | 1. System checks if the user name is valid 2. System found the username is invalid 3. System will redirect to manage users interface |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 10 | |
| Use Case Name: | View grades | |
| Actors: | Administrators – Student- Teachers | |
| Pre-conditions: | User is logged in | |
| Post-conditions: | Grades are viewed | |
| Flow of events: | User Action | System Action |
| 1- User press view grades button |  |
|  | 1. System sends the user’s ID to DB system 2. DB system retrieves the grades of the user from the database 3. DB system sends the user’s grades to the system 4. System views the grades to the user |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 11 | |
| Use Case Name: | Edit grades | |
| Actors: | Administrators – Teachers | |
| Pre-conditions: | User presses manage grades | |
| Post-conditions: | User edited the grades | |
| Flow of events: | User Action | System Action |
| 1. User press edit grade button 2. User enters Student’s ID |  |
|  | 1. System sends student’s ID to DB system 2. DB system checks for student’s ID 3. DB system retrieves student’s grades and sends them to system 4. System views student’s grades to the user |
| 1. User edit student’s grades |  |
|  | 1. System sends the edited grades to DB system 2. DB system updates student’s grades |
| Exceptions: | User Action | System Action |
| 1. User enters an unexciting student’s ID |  |
|  | 1. System sends the student’s ID to DB system 2. DB system checks for student’s ID 3. DB sends a message to system that student’s ID doesn’t exist 4. System sends a message to the user that the Student doesn’t exist |
|  | 1. User enters a grade outside the range |  |
|  | 1. System validates the data using range check 2. System sends a message to the user that the grade is outside the range |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 12 | |
| Use Case Name: | Add grades | |
| Actors: | Administrators – Teachers | |
| Pre-conditions: | User pressed manage grades | |
| Post-conditions: | User added the grades | |
| Flow of events: | User Action | System Action |
| 1. User press add grade button |  |
|  | 1. System view the courses that can accessed by the user |
| 1. User choose the course to add grades in 2. User adds a new grade |  |
|  | 1. System checks for the new grade to be within the range 2. System sends the new grade to DB system 3. DB system update the database with the new grade 4. DB system sends a confirmation message to System that the grade is added successfully 5. System sends to the user that the grades has been added successfully |
| Exceptions: | User Action | System Action |
| 1. User enters a new grade outside the range |  |
|  | 1. System validates the data using range check validation 2. System sends a message to the user that the grade is outside the range |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 13 | |
| Use Case Name: | View notification | |
| Actors: | Administrators – Teachers – Student – X | |
| Pre-conditions: | User pressed manage notifications | |
| Post-conditions: | Notifications are viewed | |
| Flow of events: | User Action | System Action |
| 1- User press view notification button |  |
|  | 1. System requests the notifications of user’ ID from notifications system 2. Notifications system sends the user’s ID to DB system 3. DB system retrieves notifications of the user and send them to notifications system 4. Notifications system sends them to System 5. System views notifications to the user |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 14 | |
| Use Case Name: | Clear Notifications | |
| Actors: | Administrators – Teachers – Student – X | |
| Pre-conditions: | View Notifications | |
| Post-conditions: | Notifications cleared | |
| Flow of events: | User Action | System Action |
| 1- User press clear notifications |  |
|  | 1. System requests the notifications system to clear notifications 2. Notifications system sends user’s ID to DB system 3. DB system delete notifications from the database |

|  |  |  |
| --- | --- | --- |
|  | 15 | |
| Use Case Name: | Rate teacher | |
| Actors: | Student | |
| Pre-conditions: | User pressed rate teacher | |
| Post-conditions: | Teacher is rated | |
| Flow of events: | User Action | System Action |
| 1. User presses rate teacher 2. User selects teacher 3. User enters rate |  |
|  | 1. System validates the entry to be in the range 2. System calculates total rate for the teacher 3. System sends total rate to DB system 4. DB system update teacher’s rate |

|  |  |  |
| --- | --- | --- |
|  | 16 | |
| Use Case Name: | Delete course | |
| Actors: | Admin user | |
| Pre-conditions: | Admin user pressed manage courses | |
| Post-conditions: | Course is deleted | |
| Flow of events: | User Action | System Action |
| 1. User presses delete course |  |
|  | 1. System views course(s) that can be accessed by the user |
| 1. User selects the course |  |
|  | 1. System sends course’s ID to DB system 2. DB system deletes course’s data from the database |
| Exceptions: | User Action | System Action |
| 1. Users choose a course has existing students |  |
|  | 1. System checks that there is no students attend the course 2. System sends a message to the admin user that the course is already taken by a student and it cannot be deleted |

|  |  |  |
| --- | --- | --- |
|  | 17 | |
| Use Case Name: | Add offer | |
| Actors: | X | |
| Pre-conditions: | User pressed add offer | |
| Post-conditions: | Offer is sent to student | |
| Flow of events: | User Action | System Action |
| 1. User presses add offer 2. User enters offer details 3. User enters preferred department and grades |  |
|  | 1. System send offer to notifications system 2. Notifications system sends the offer to specific students |

|  |  |  |
| --- | --- | --- |
|  | 18 | |
| Use Case Name: | Manage users | |
| Actors: | Admin user | |
| Pre-conditions: | Admin user pressed manage users | |
| Post-conditions: | Manage users menu shows up to the admin user | |
| Flow of events: | User Action | System Action |
| 1. Admin user presses manage users |  |
|  | 1. System shows up manage users menu ( add and remove user ) |

|  |  |  |
| --- | --- | --- |
|  | 19 | |
| Use Case Name: | Manage courses | |
| Actors: | Admin user | |
| Pre-conditions: | Admin user pressed manage users | |
| Post-conditions: | Manage courses menu shows up to the admin user | |
| Flow of events: | User Action | System Action |
| 1. Admin user presses manage users |  |
|  | 1. System shows up manage courses menu ( view, add, assign, delete, drop and register course ) |

|  |  |  |
| --- | --- | --- |
|  | 20 | |
| Use Case Name: | Manage student courses | |
| Actors: | Student | |
| Pre-conditions: | Student pressed manage student courses | |
| Post-conditions: | Manage users menu shows up to the admin user | |
| Flow of events: | User Action | System Action |
| 1. Admin user presses manage student courses |  |
|  | 1. System shows up manage student courses menu ( register, view and drop course ) |

|  |  |  |
| --- | --- | --- |
|  | 21 | |
| Use Case Name: | Manage notifications | |
| Actors: | Admin user – Student – Teacher – X | |
| Pre-conditions: | User pressed manage notifications | |
| Post-conditions: | Manage notifications menu shows up to the user | |
| Flow of events: | User Action | System Action |
| 1. User presses manage notifications |  |
|  | 1. System shows up manage notifications menu ( view and clear notification ) |

Ownership Report

Remove the following notes and any red notes

For every item in this document, write the owners. If someone is owner of something, s/he understands it 100%

Team leader must verify the table with the team members.

|  |  |
| --- | --- |
| Item | Owners |
|  |  |
|  |  |