**Sign in (UC-B1)**

* **Actors**: Student, Admin, Teacher.
* **Brief description**: This use case describes How to sign in.

redirect the user to their view that Available to them.

* **Pre-conditions**: no Pre-conditions.
* **Post-condition**: redirect the user to view.

.

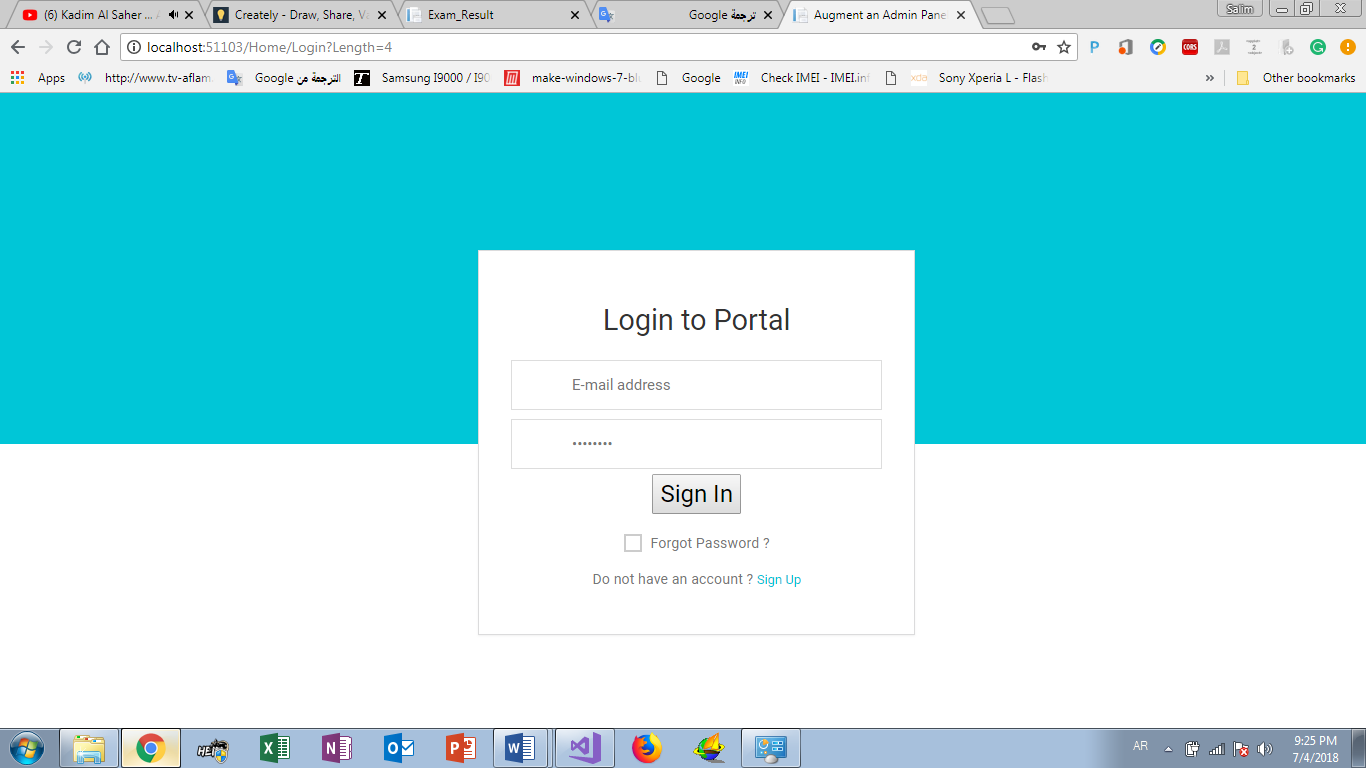
* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Enter login information. |  |
| 2.Click Button “Sign in” |  |
|  | 2.Request for view that Available to this user. |
|  | 3.Rediret to the view. |

Figure 28: Sign in flow of events

**Critical scenario**:

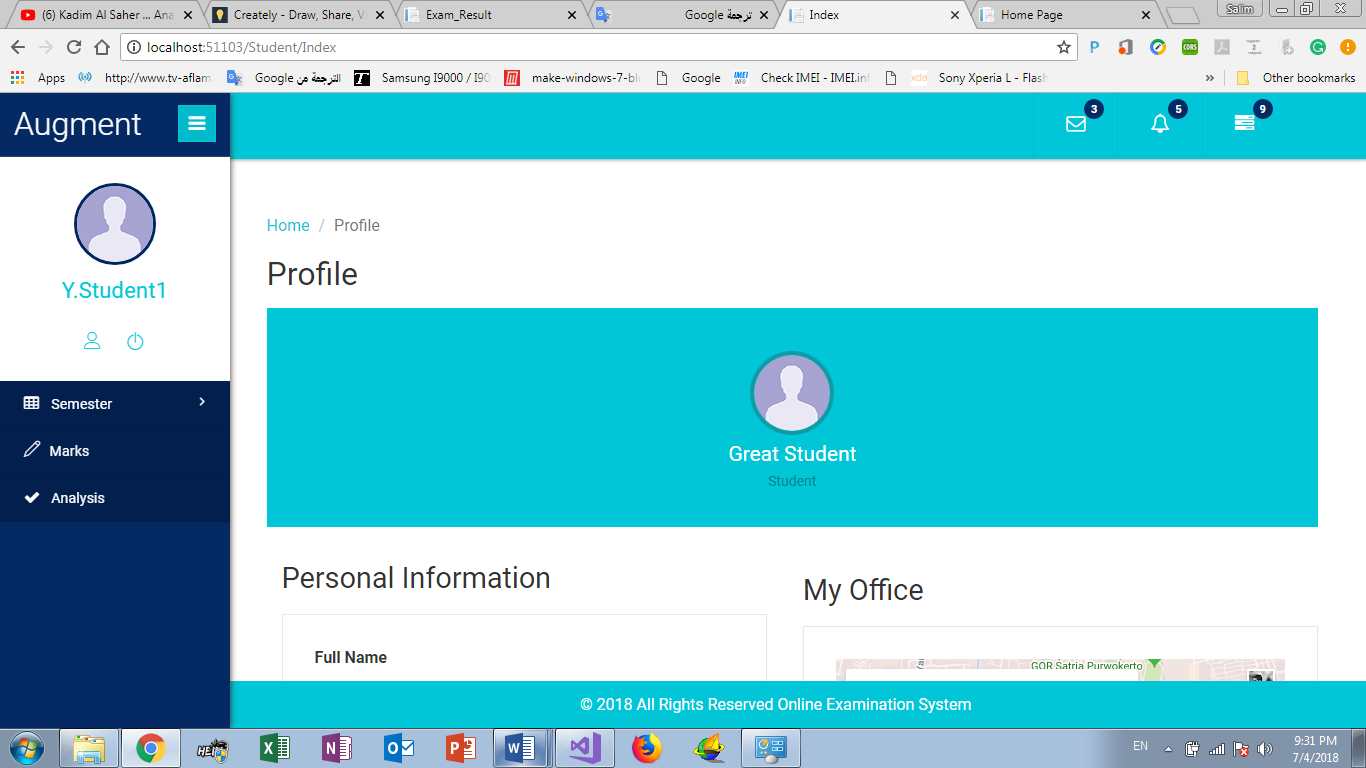
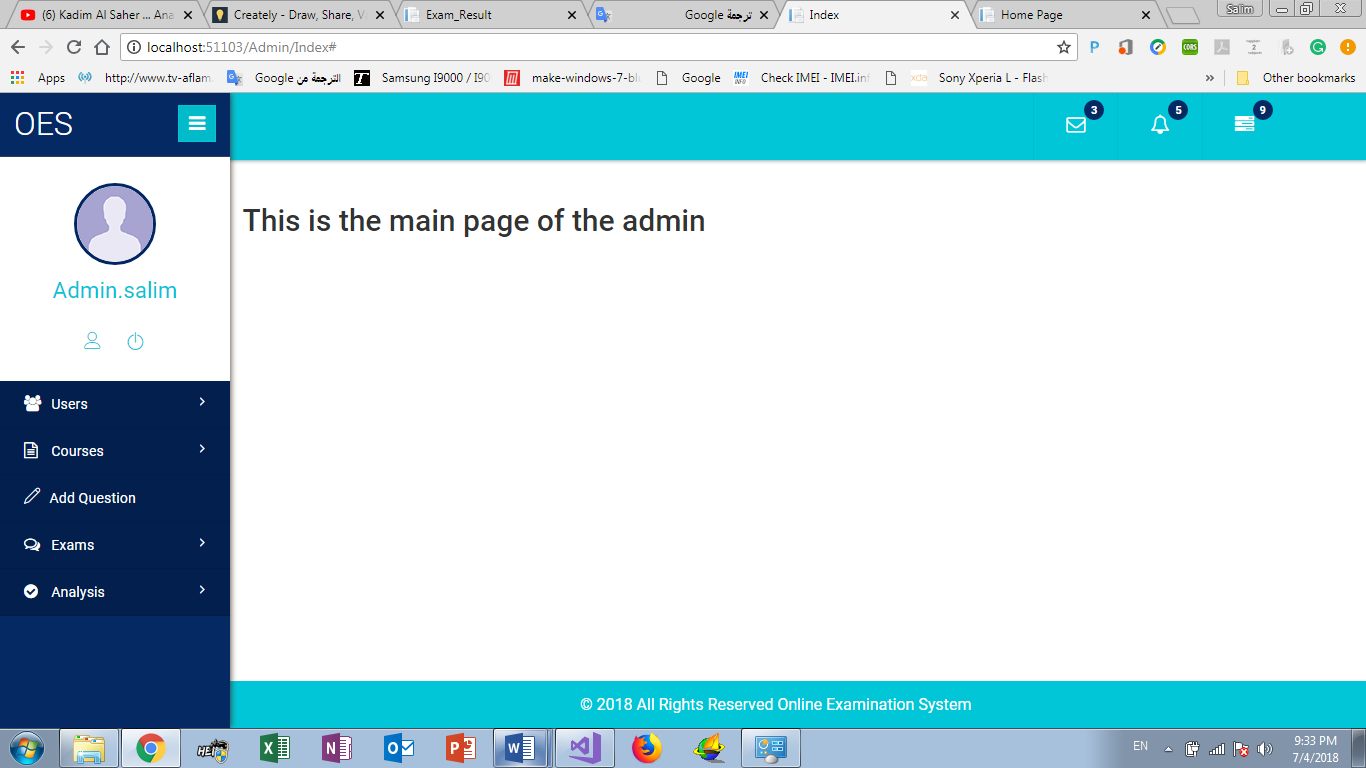
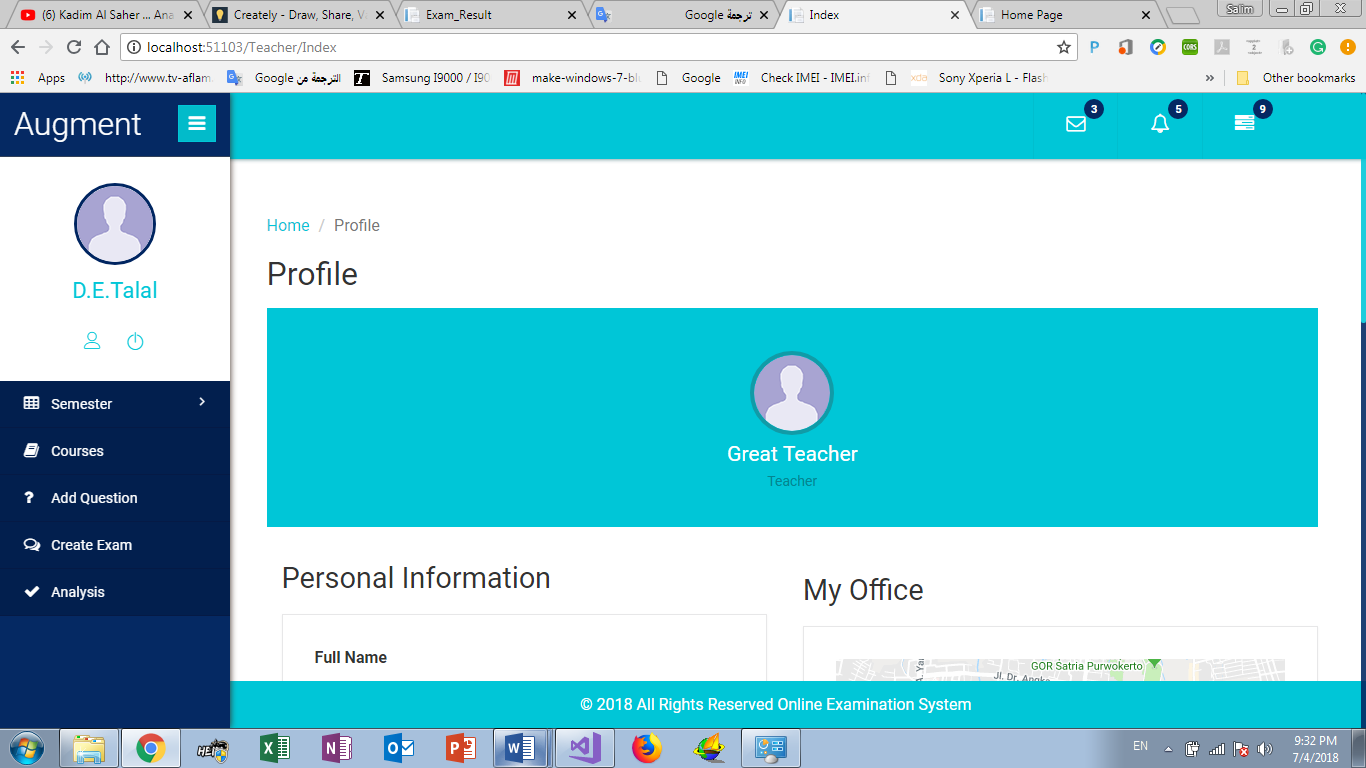
1. Error with login information

**User interface prototype:**

student

Teacher

Admin



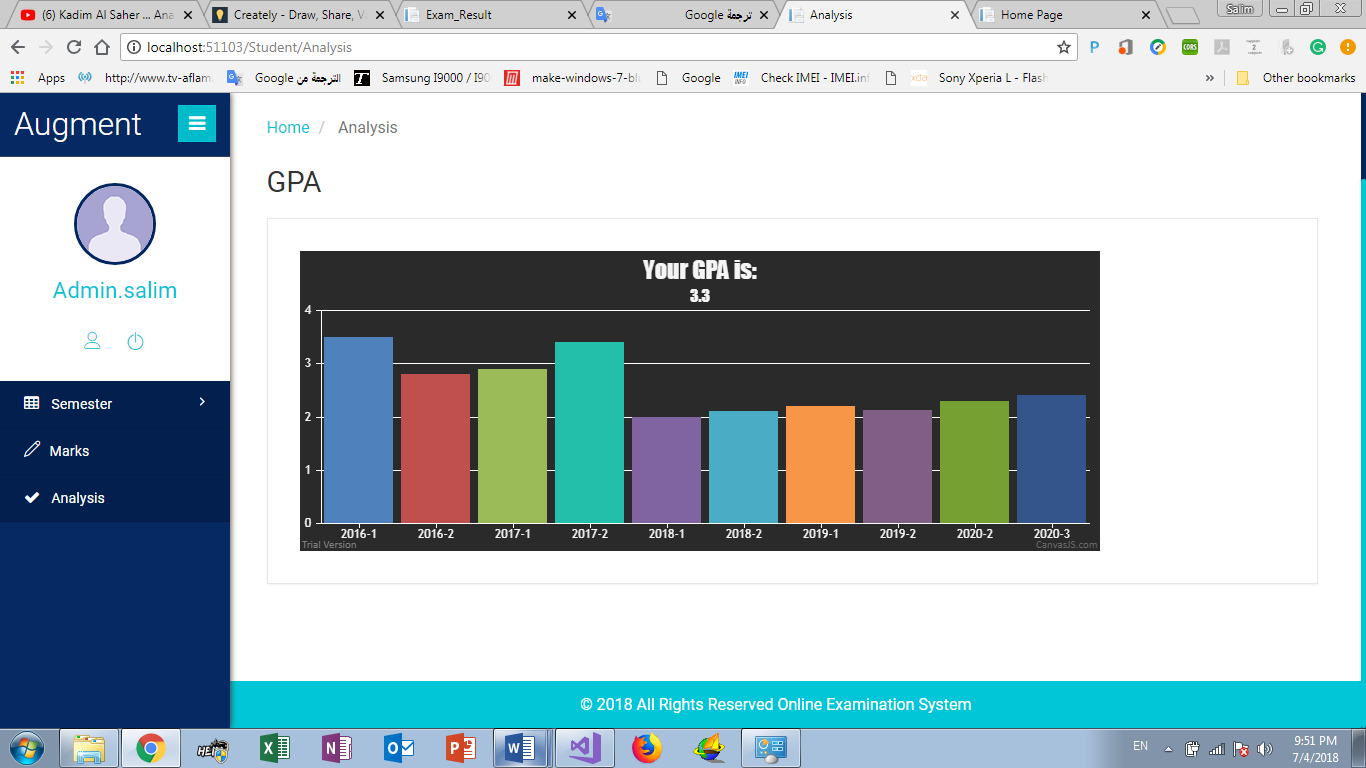
**GPA (UC-B2)**

* **Actors**: Student.
* **Brief description**: This use case describes how student can get his GPA
* **Pre-conditions**: Log in as Student.
* **Post-condition**: show student rate in each semester than calculate student GPA
* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Click”GPA ” |  |
|  | 2.Request for get student rate in each semester. |
|  | 3.show student rate and his GPA. |

**Critical scenario**:

Begin of semester

**User interface prototype:**

**Marks (UC-B3)**

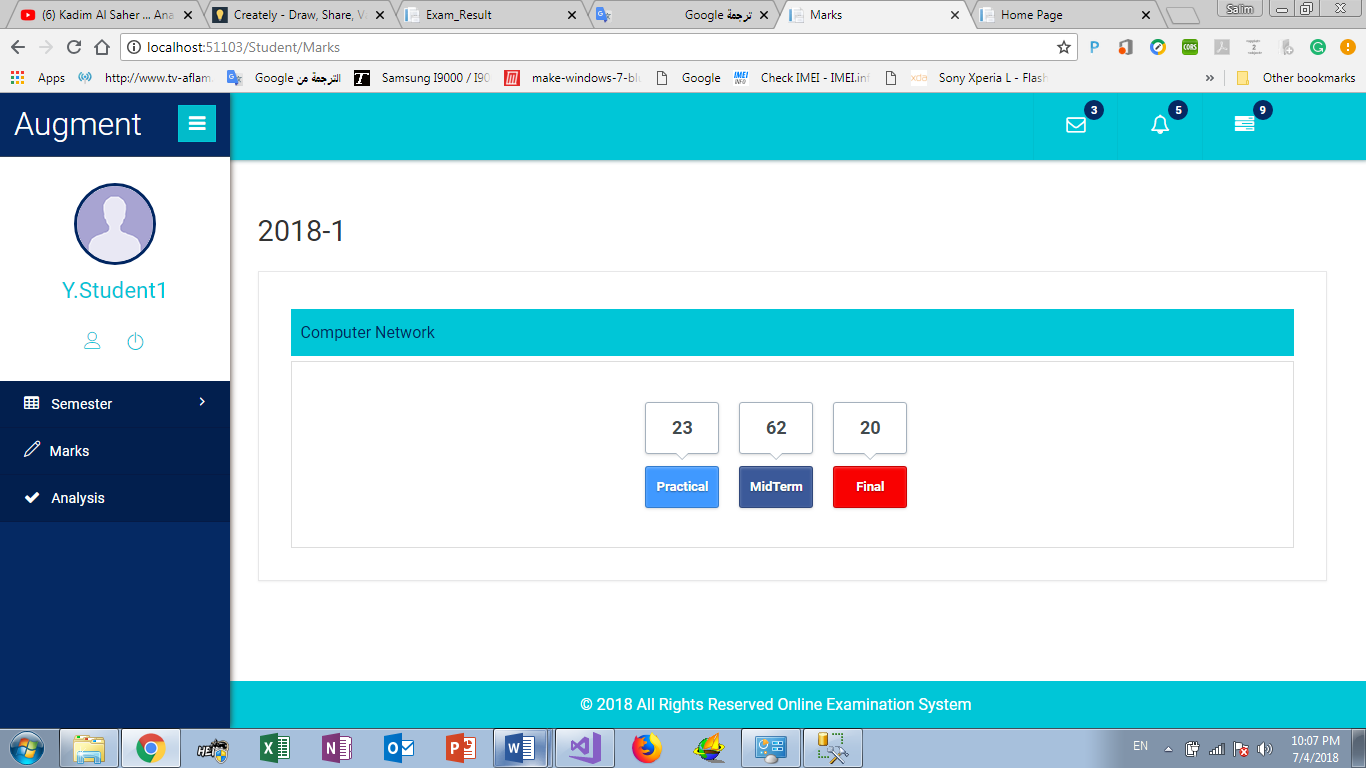
* **Actors**: Student.
* **Brief description**: This use case describes how student can show his marks in each course
* **Pre-conditions**: Log in as Student.
* **Post-condition**: show student’s marks in each course in the end of semester
* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Click”Marks ” |  |
|  | 2.Request for get student Marks in each Course. |
|  | 3.show student Marks |

**Critical scenario**:

Begin of semester

**User interface prototype:**



**Add Question (UC-B4)**

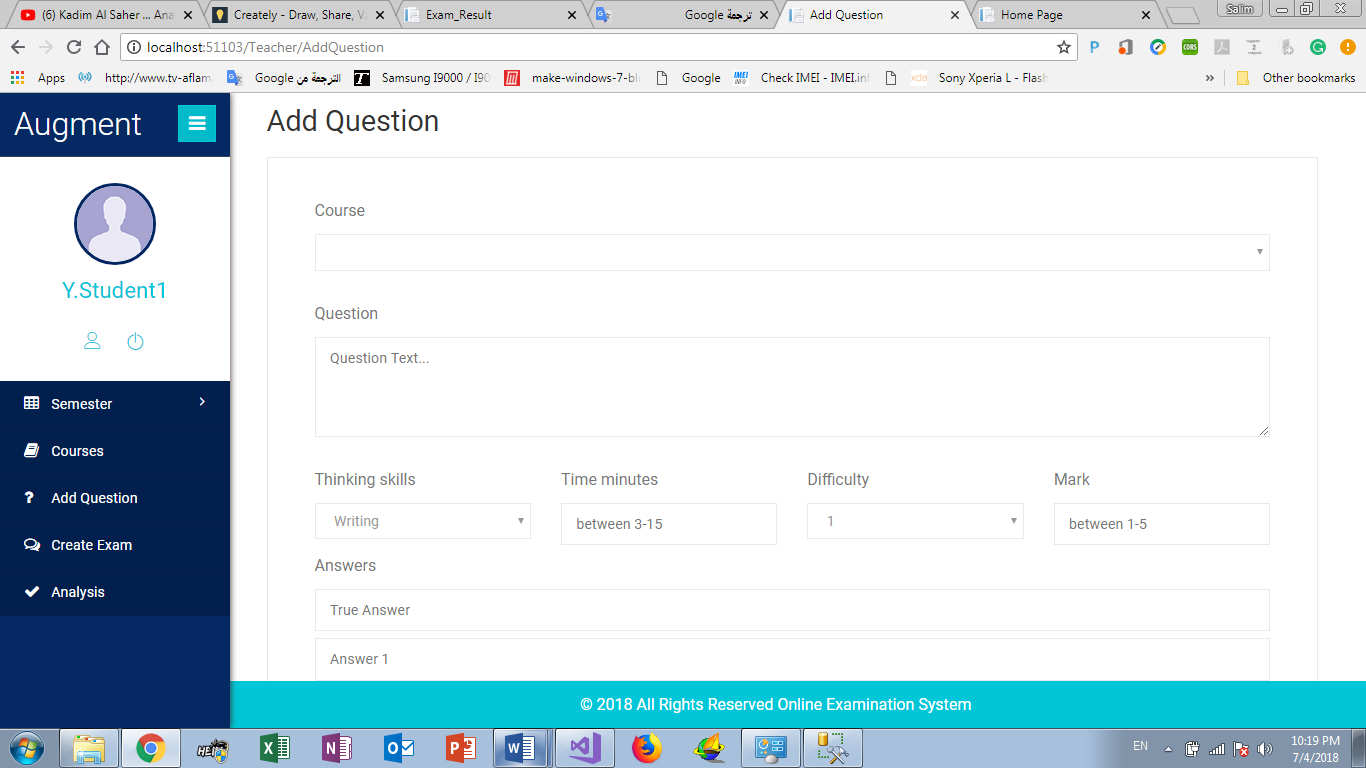
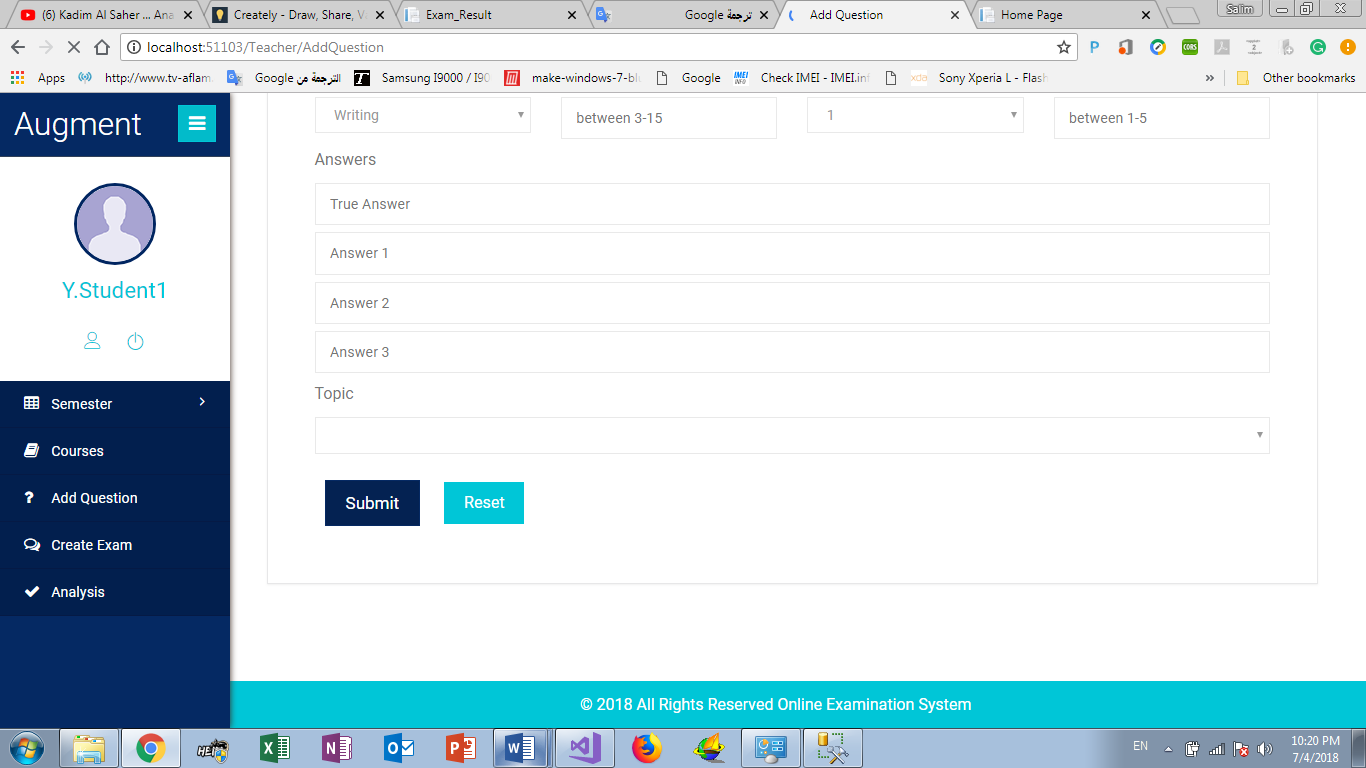
* **Actors**: Teacher.
* **Brief description**: This use case describes how Teacher can Add new Question
* **Pre-conditions**: Log in as Teacher.
* **Post-condition**: Teacher add new question in course
* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Click”Add Question ” |  |
| 1. select course |  |
| 1. insert question text |  |
| 1. Insert question 4 Answers |  |
| 1. Insert question thinking skills |  |
| 1. Insert question time |  |
| 1. Insert question difficulty |  |
| 1. Insert question marks |  |
| 1. Select topic |  |
| 1. Click”Add” |  |
|  | 1. Request for add new question |
|  | 1. Show “Success” |

**Critical scenario**:

Error in question information

**User interface prototype:**



**Create Exam (UC-B5)**

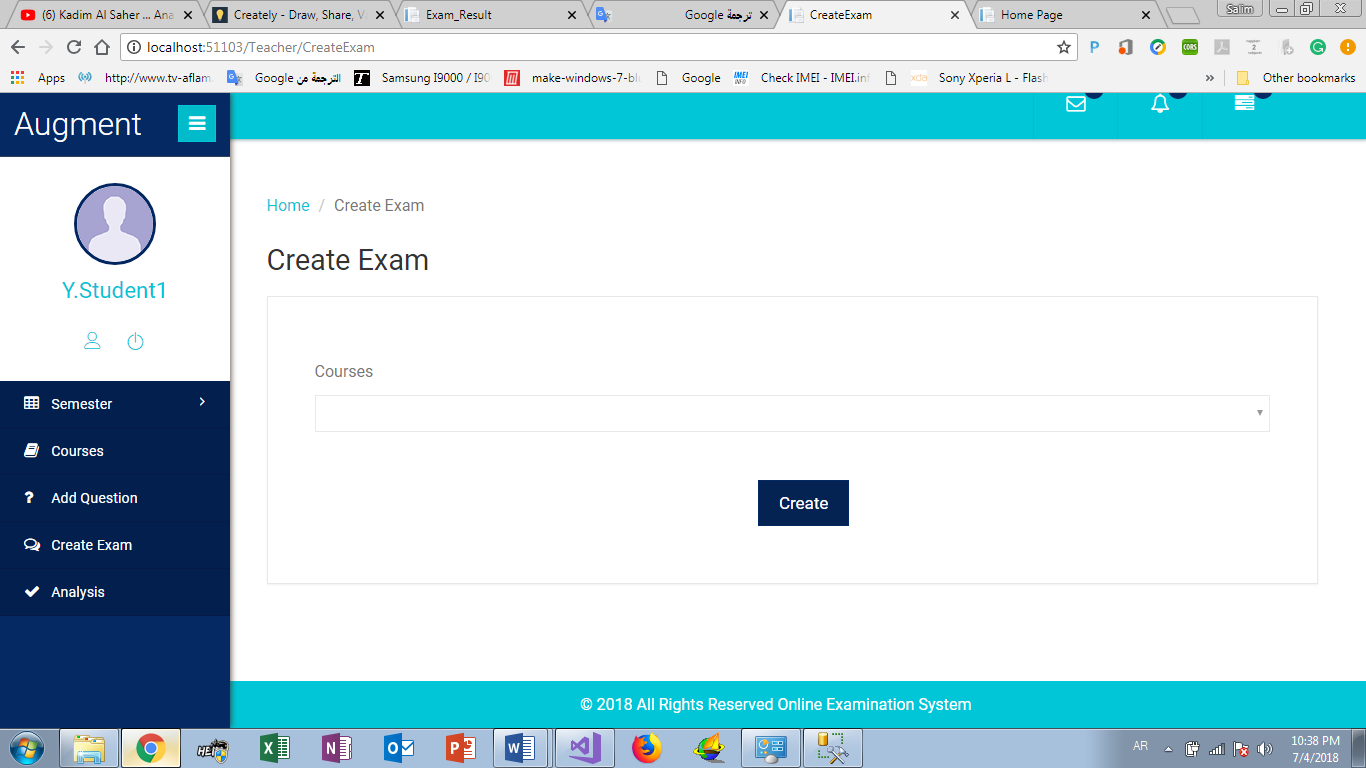
* **Actors**: Teacher.
* **Brief description**: This use case describes how Teacher can create Exam
* **Pre-conditions**: Log in as Teacher.
* **Post-condition**: Teacher create exam and select exam topic and information
* **Flow of event**:

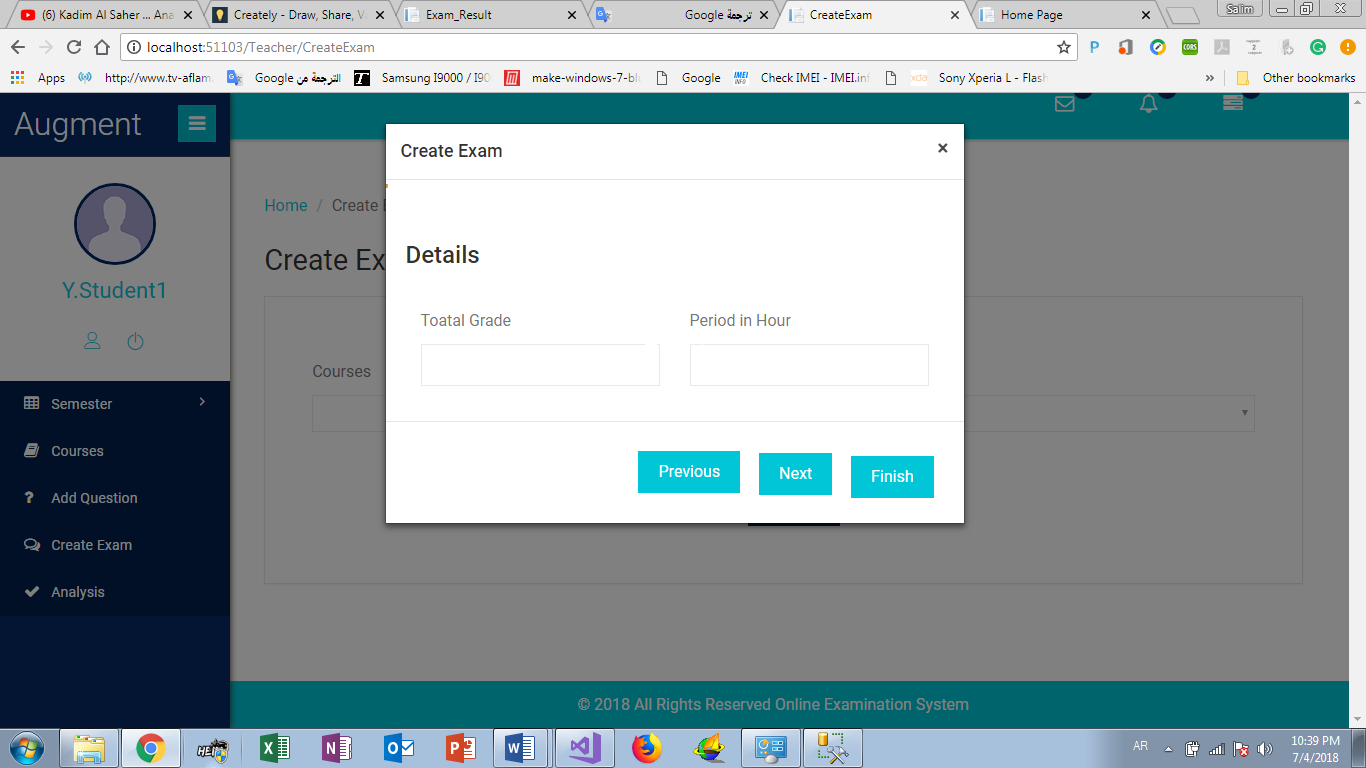
|  |  |
| --- | --- |
| Actor | System |
| 1. Click”Create Exam” |  |
| 1. select course |  |
| 1. Insert Exam total Grade |  |
| 1. Insert Exam period |  |
| 1. Select Exam topic |  |
| 1. Select exam thinking skills |  |
| 1. Select Exam difficulty |  |
| 1. Click”finash” |  |
| 1. making sure the information is good |  |
| 1. Click”Generate” |  |
|  | 1. Request for create exam |
|  | 1. Show “success” |

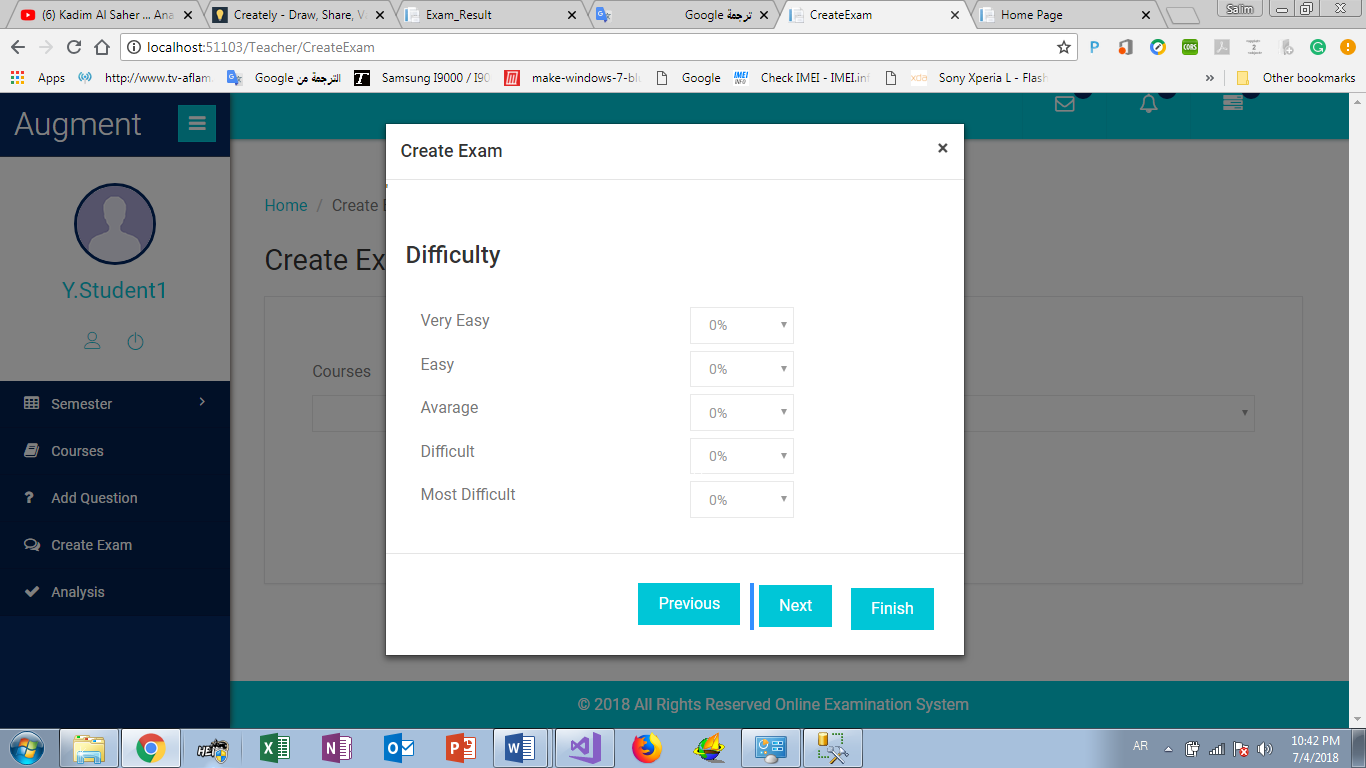
**Critical scenario**:

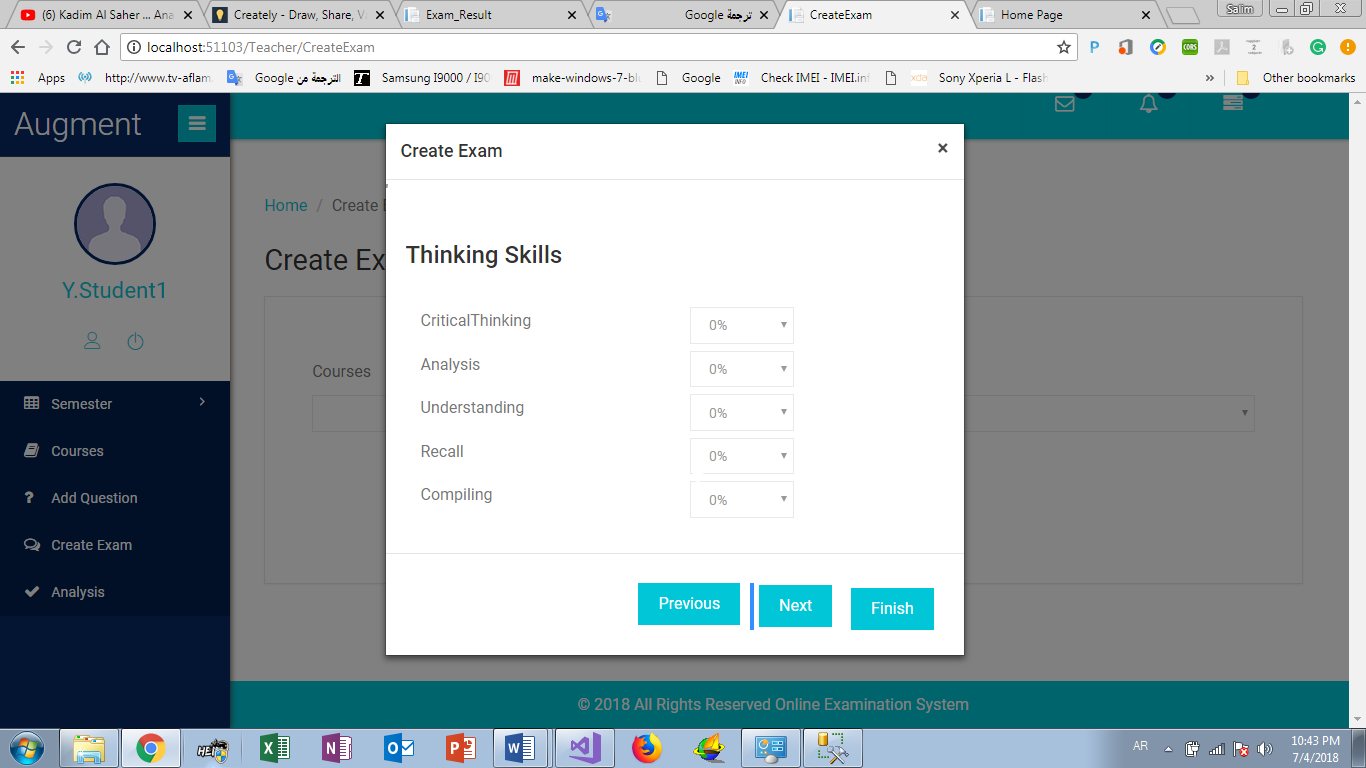
Error in Exam information

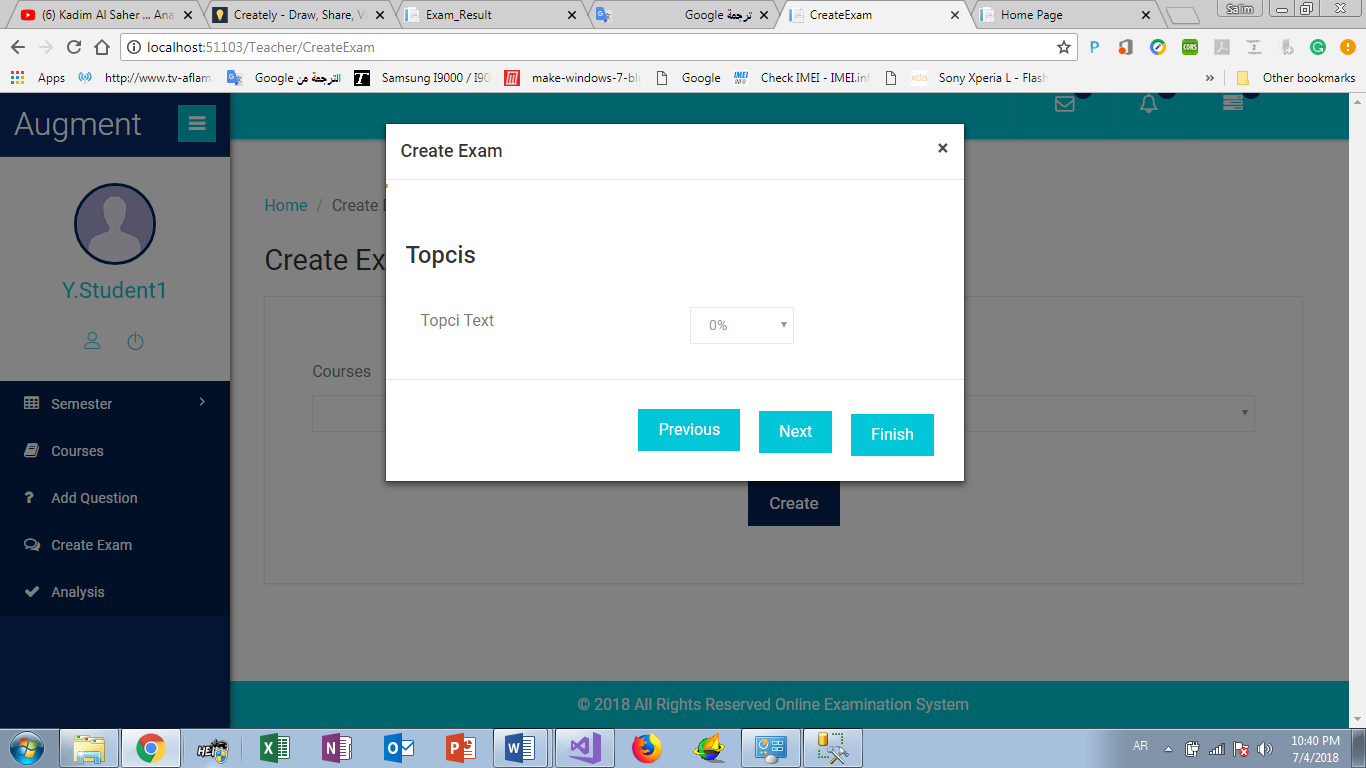
**User interface prototype:**











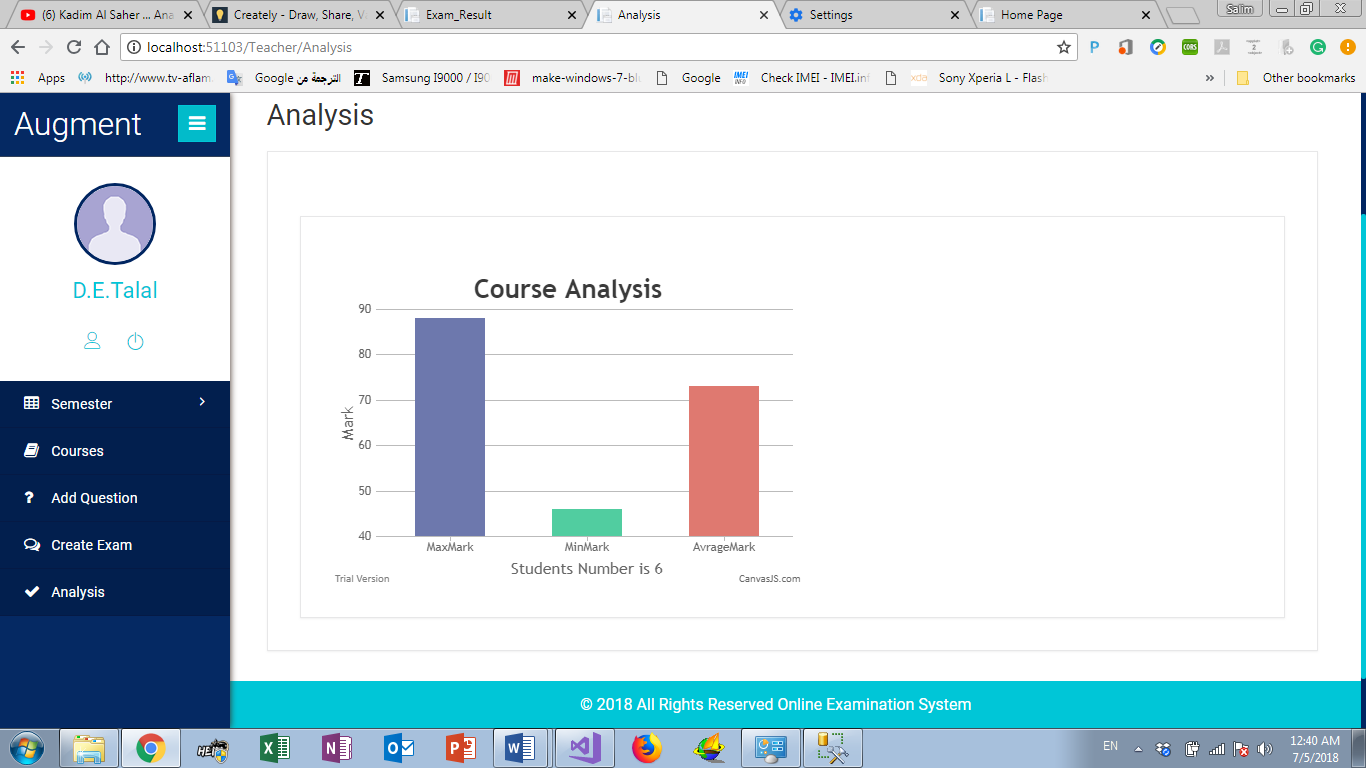
**Course Analysis (UC-B6)**

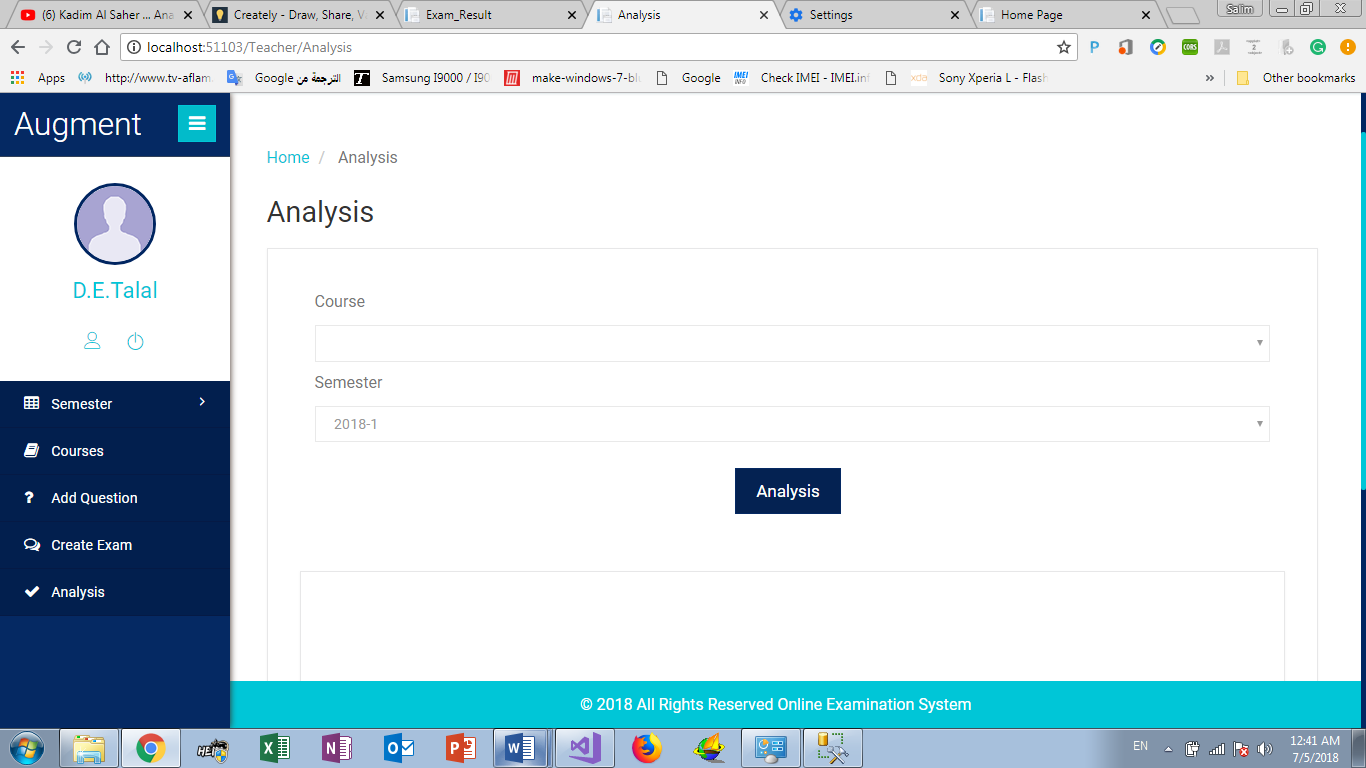
* **Actors**: Teacher.
* **Brief description**: This use case describes how Teacher can get information about his courses
* **Pre-conditions**: Log in as Teacher.
* **Post-condition**: Teacher get information about his courses like max mark, min mark and average mark
* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Click “Analysis” |  |
| 1. Select Semester |  |
| 3. Select Course | 2.Request for get Course Information. |
|  | 3.show chart about course information. |

**Critical scenario**:

No information about course

 **User interface prototype:**



**Add User (UC-B7)**

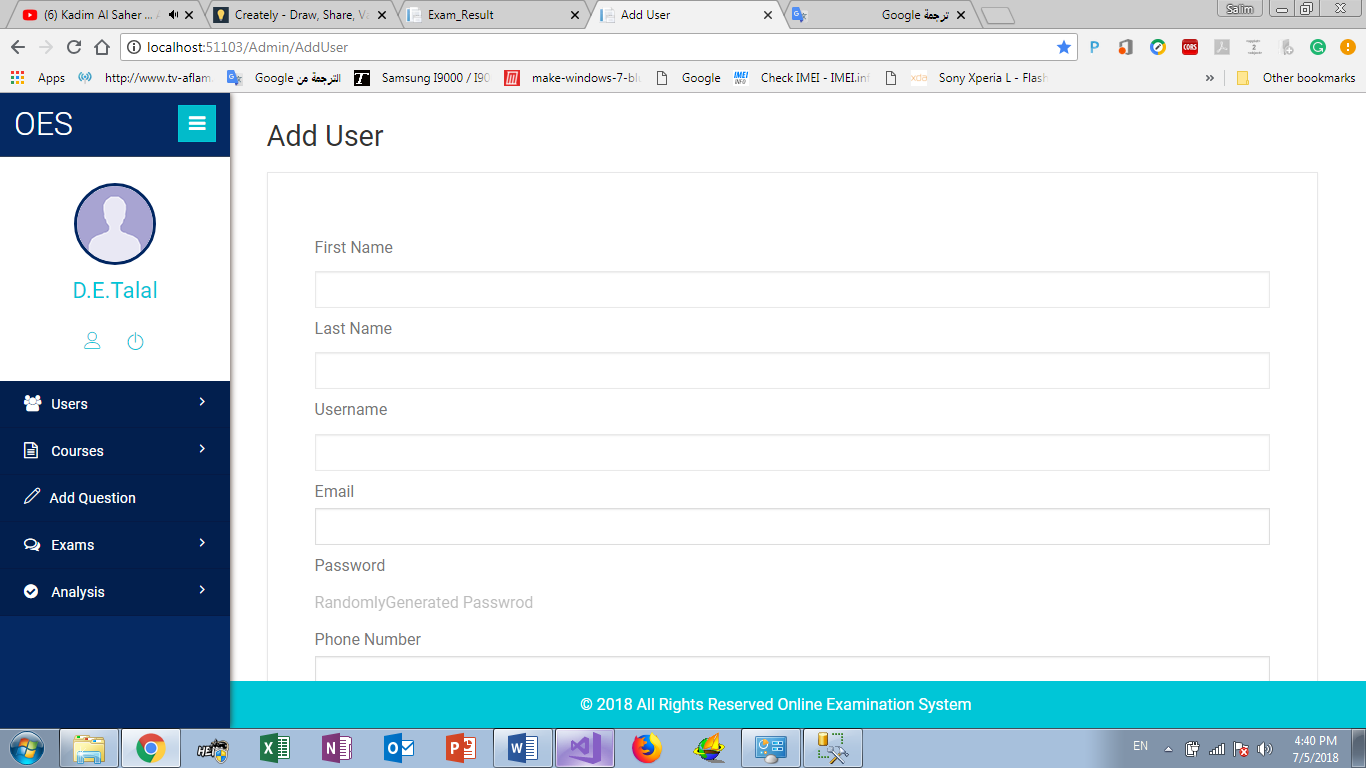
* **Actors**: Admin.
* **Brief description**: This use case describes how Admin can add new user (Teacher or Student or Admin)
* **Pre-conditions**: Log in as Admin.
* **Post-condition**: add user information’s, and send username and password in email to user, after that he can login the system
* **Flow of event**:

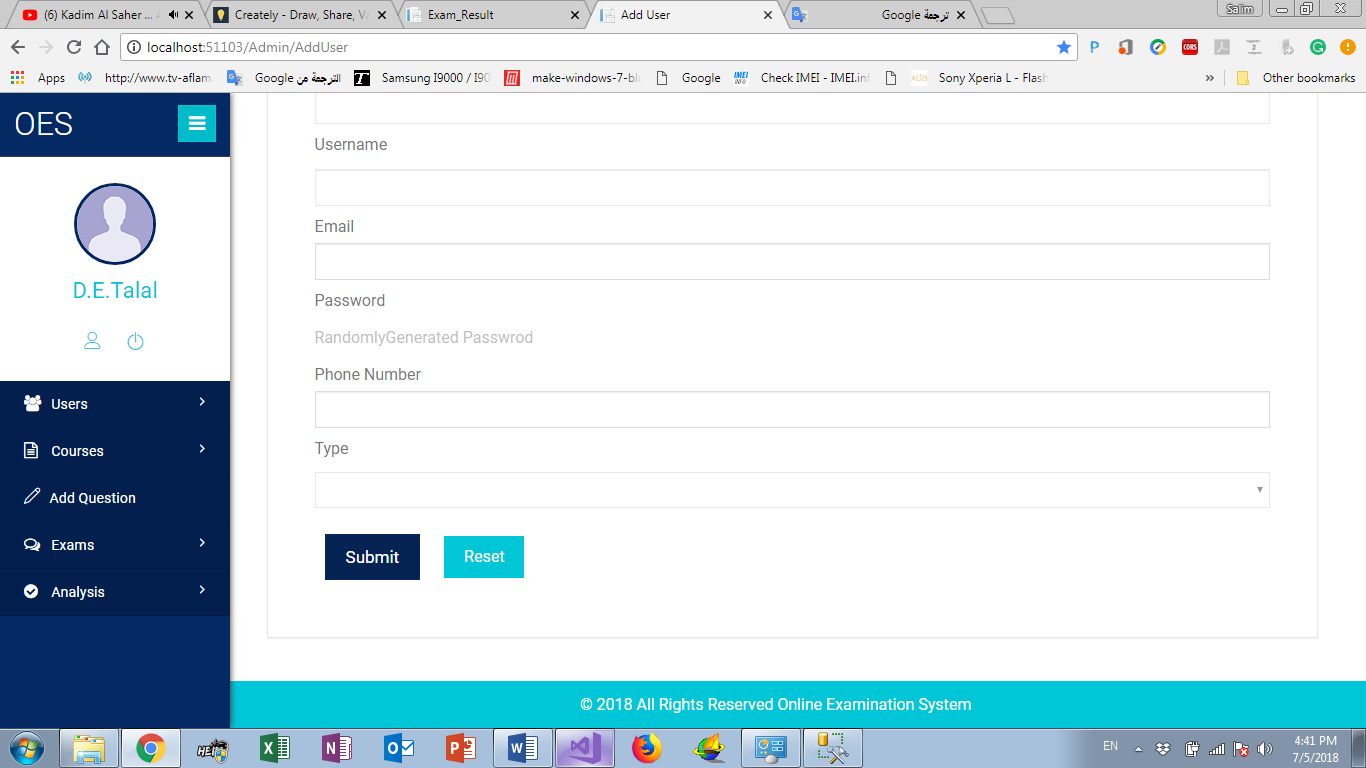
|  |  |
| --- | --- |
| Actor | System |
| 1. Click “Add User” |  |
| 1. Add user information |  |
| 1. Click” Add” | 1. Generate random password |
|  | 1. check the information and request for add new user . |
|  | 1. show “success” |

**Critical scenario**:

error in user information

**User interface prototype:**





**Add New Course (UC-B8)**

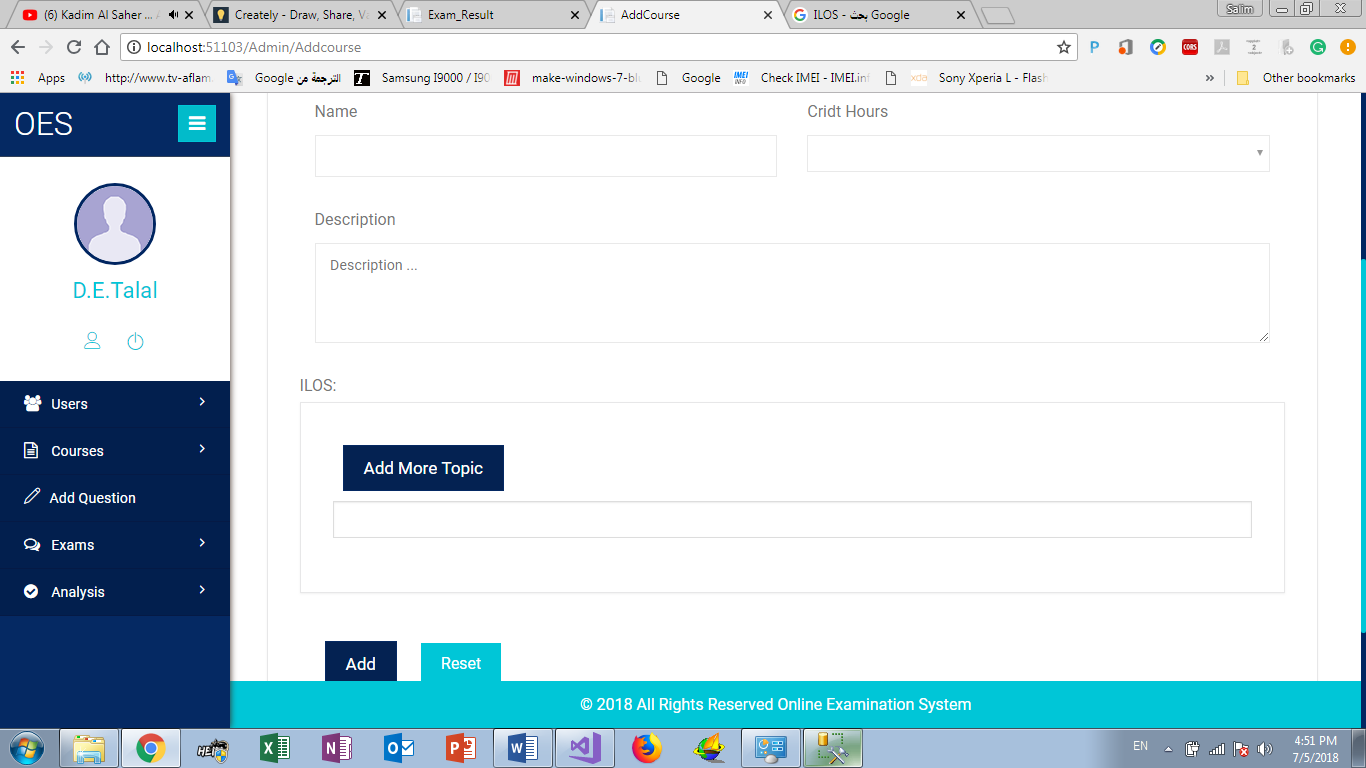
* **Actors**: Admin.
* **Brief description**: This use case describes how Admin can add new course with his ILOS
* **Pre-conditions**: Login as Admin.
* **Post-condition**: add new course to university courses with his Intended Learning Outcomes (ILOS)
* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Click “Add Course” |  |
| 1. Add Course information and ILOS |  |
| 1. Click” Add” | 1. check the information and request for add new user. |
|  | 1. show “success” |
|  |  |

**Critical scenario**:

error in Course information’s

**User interface prototype:**



**Course statistics (UC-B9)**

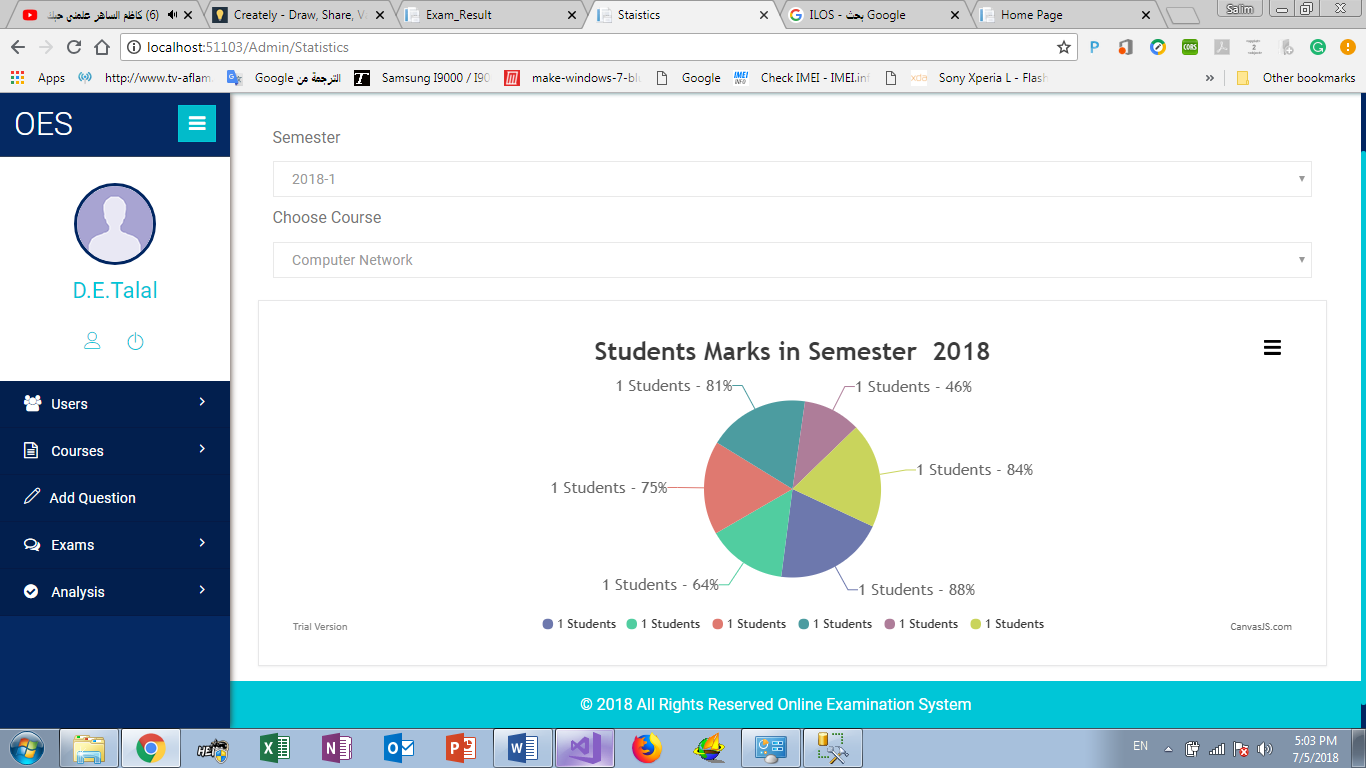
* **Actors**: Admin.
* **Brief description**: This use case describes how Admin can get students marks in course
* **Pre-conditions**: Login as Admin.
* **Post-condition**: get students marks and show the mark in chart
* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Click “course statistics” |  |
| 1. Select Course |  |
| 1. Select semester |  |
|  | 1. check the information and request for get students mark |
|  | 1. show “Student mark in chart” |

**Critical scenario**:

no students mark.

**User interface prototype:**



**Add Exam (UC-B10)**

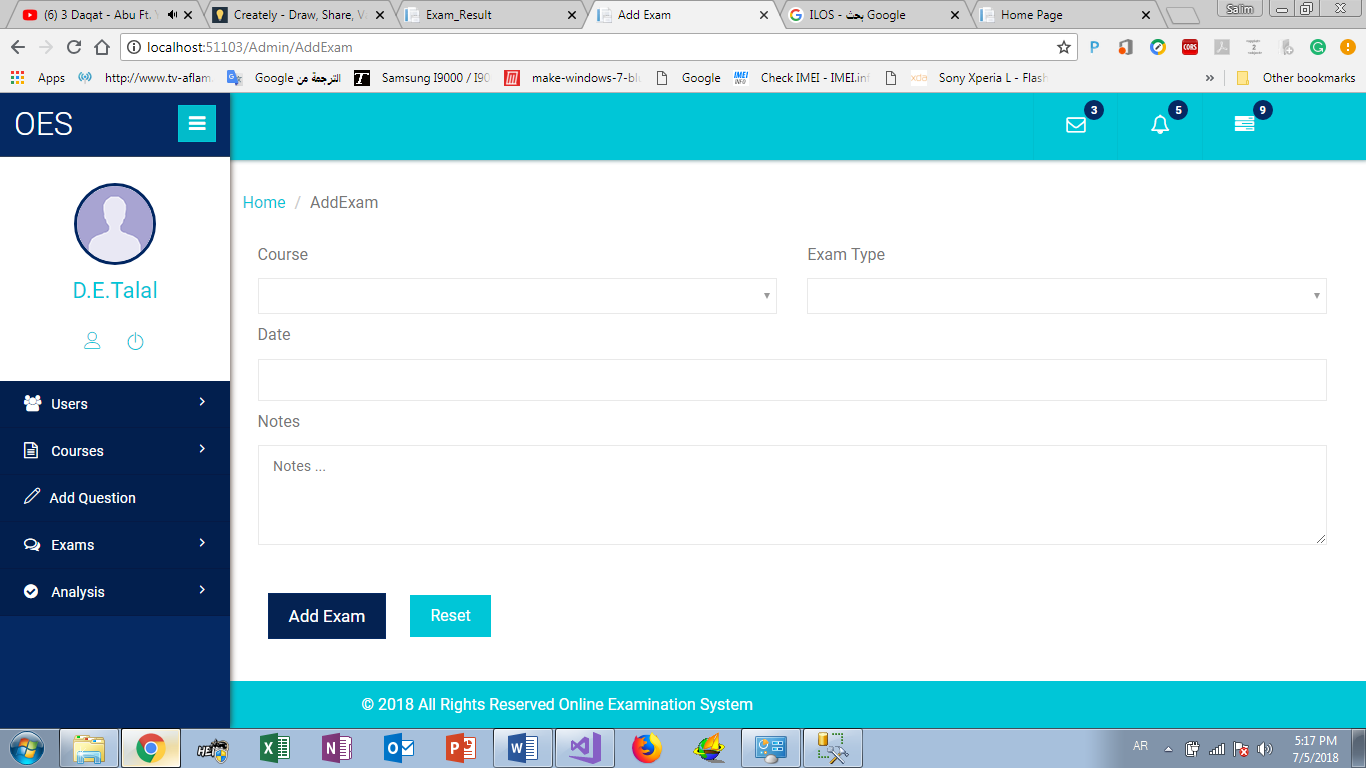
* **Actors**: Admin.
* **Brief description**: This use case describes how Admin can add new Exam
* **Pre-conditions**: Login as Admin.
* **Post-condition**: add new exam
* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Click “Add Exam” |  |
| 1. Select Course |  |
| 1. Select Exam Type |  |
| 1. Add Exam information | 1. check the information and request for get add new exam |
|  | 1. show “Success” |

**Critical scenario**:

Exam added by another Admin

**User interface prototype:**



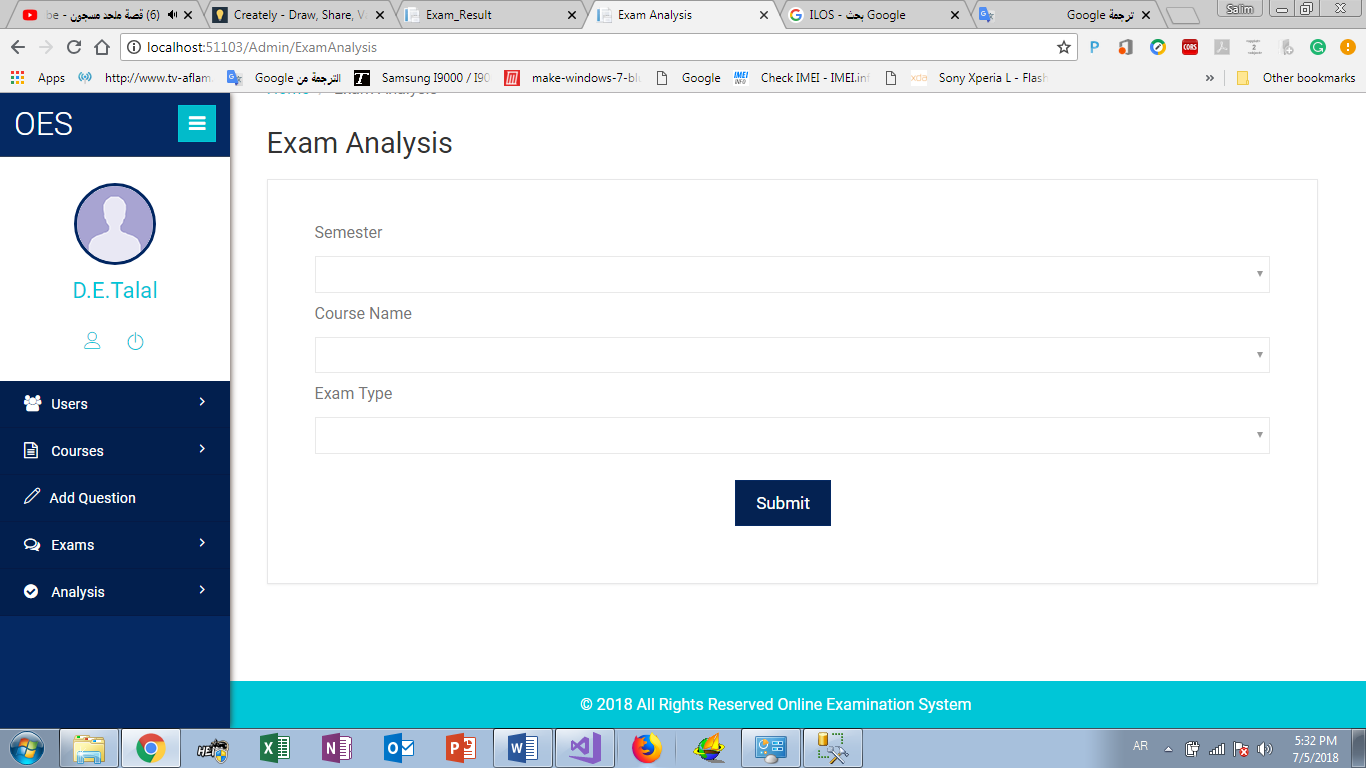
**Exam Analysis (UC-B11)**

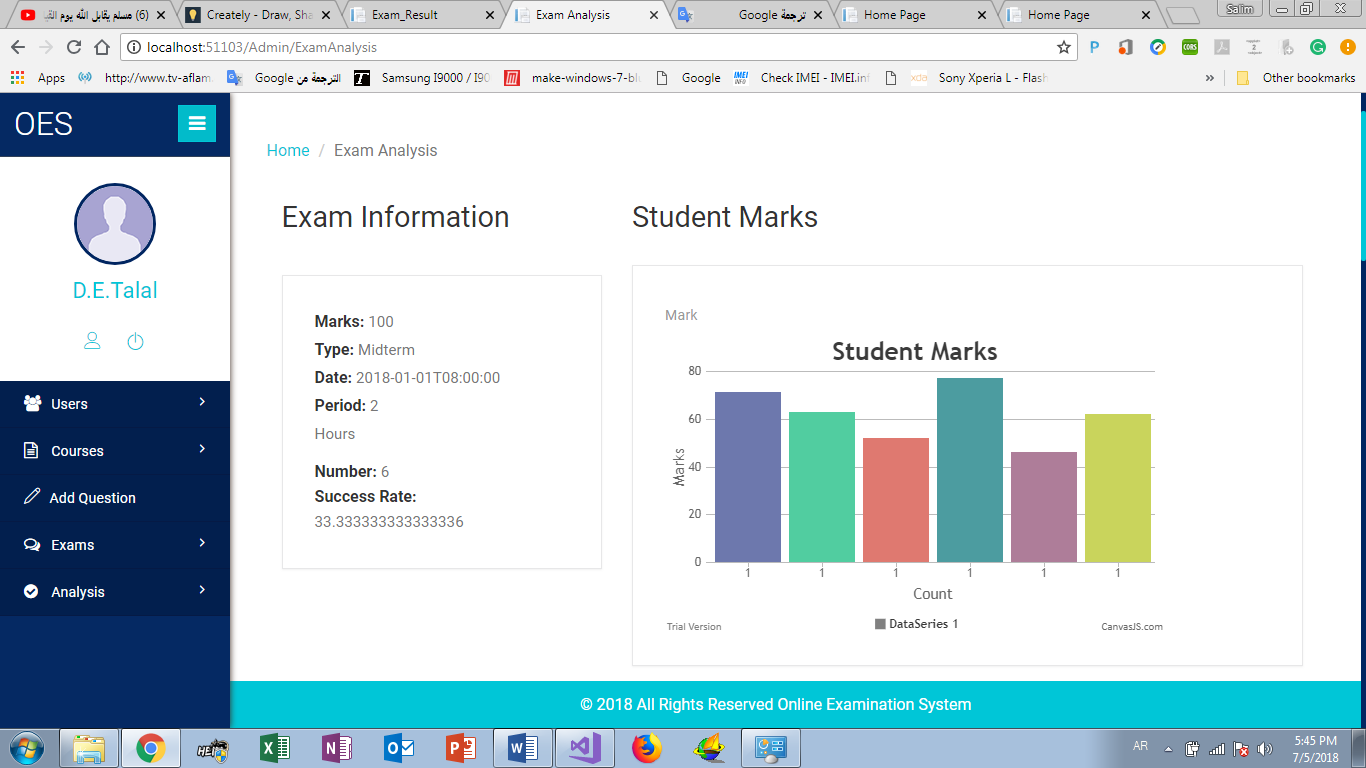
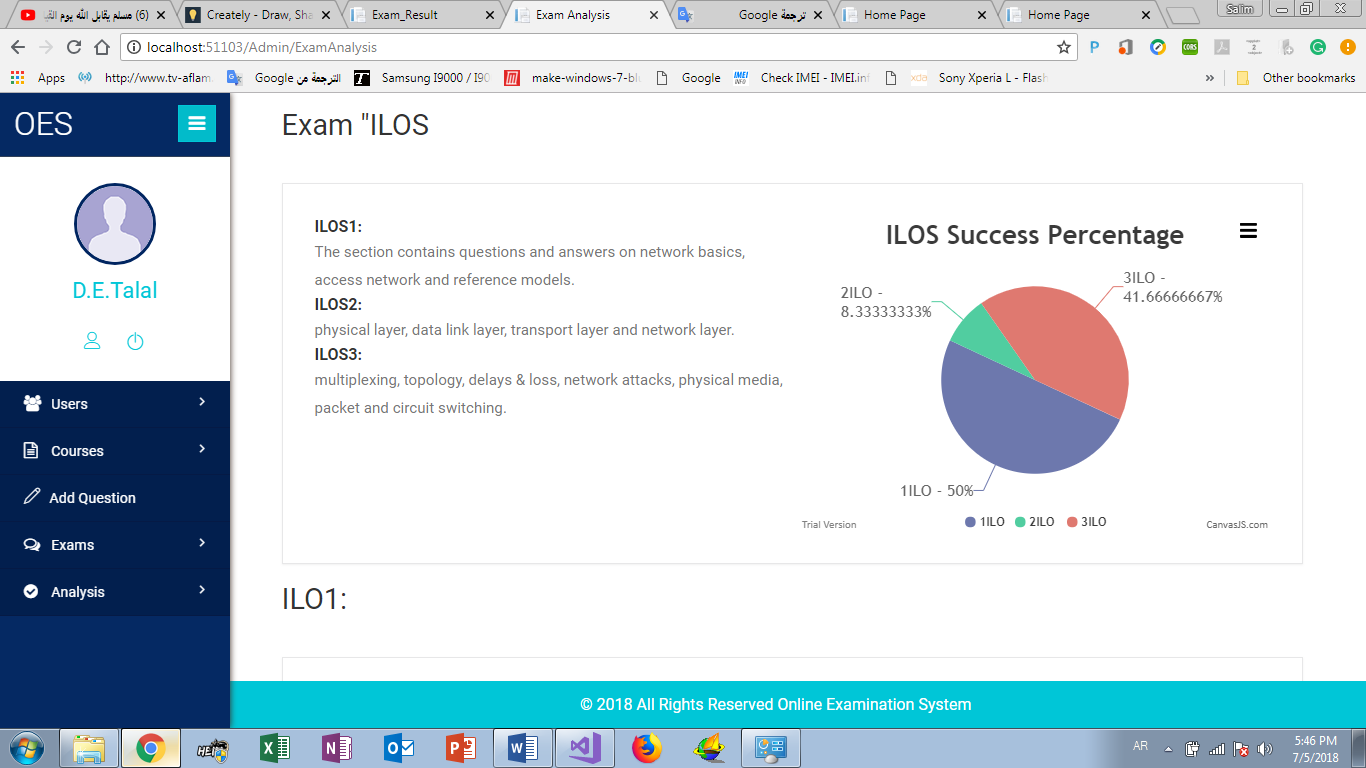
* **Actors**: Admin.
* **Brief description**: This use case describes how Admin can Analysis any Exam
* **Pre-conditions**: Login as Admin.
* **Post-condition**: get information about Exam like ILOS,Topic, success percentage ….
* **Flow of event**:

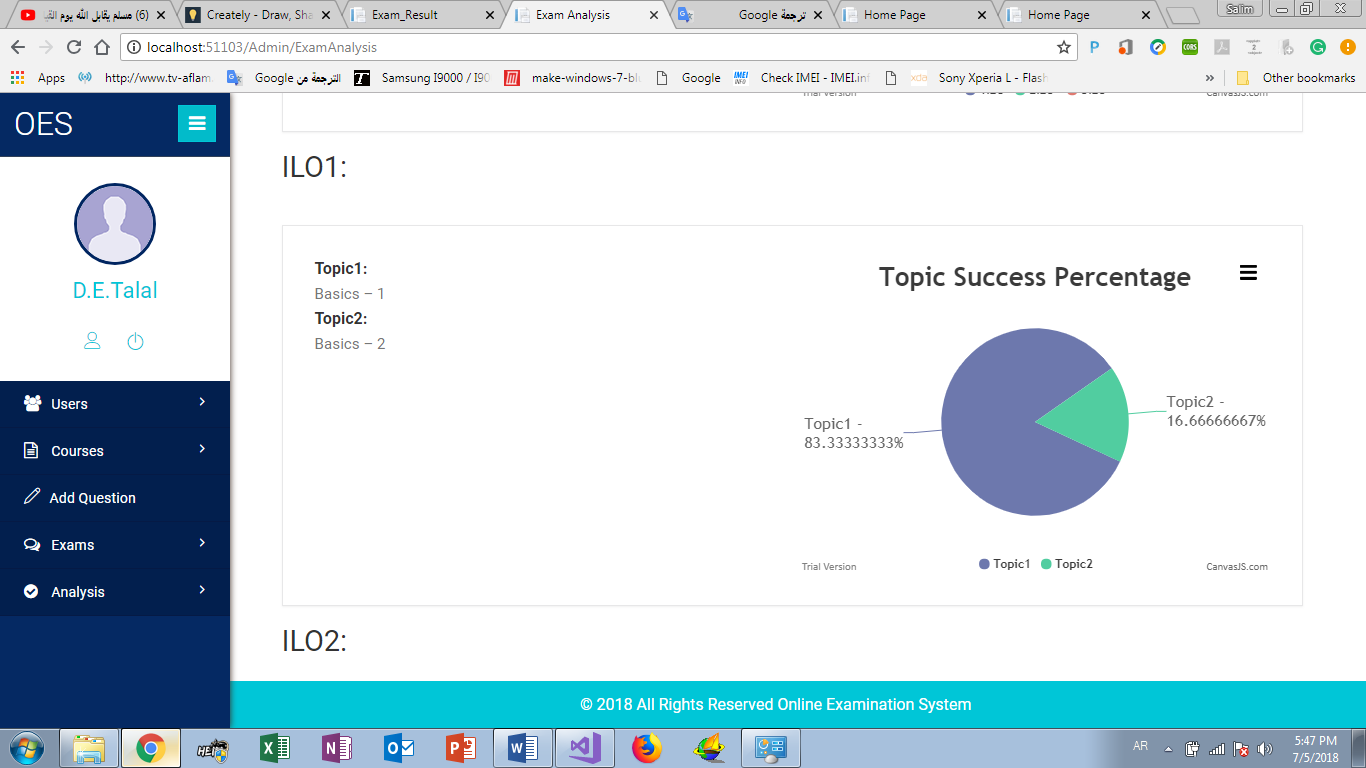
|  |  |
| --- | --- |
| Actor | System |
| 1. Click “Exam Analysis” |  |
| 1. Select Course and semester |  |
| 1. Select Exam Type |  |
| 1. Click “Analysis” |  |
|  | 1. check the information and request for get Exam information’s |
|  | 1. show “Exam information’s” |

**Critical scenario**:

Not the exam time

 **User interface prototype:**





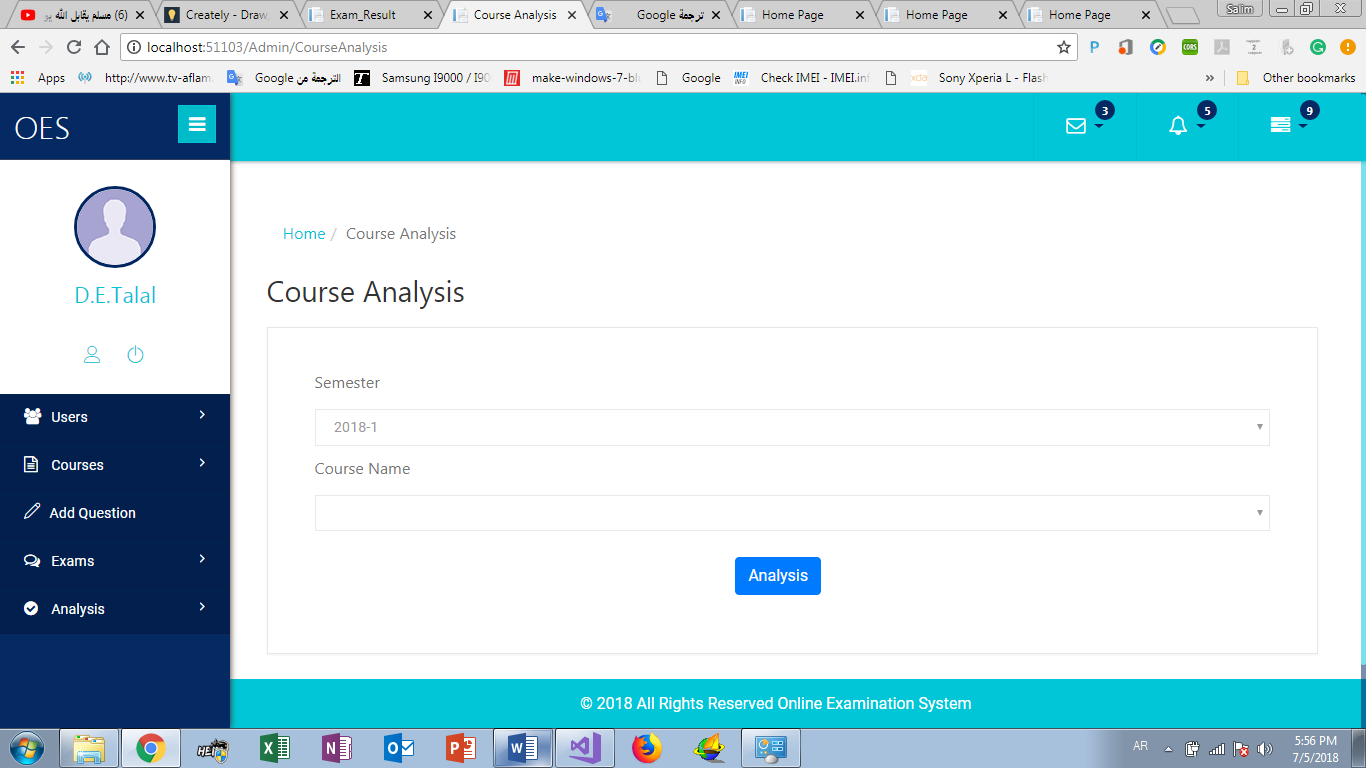
**Course Analysis (UC-B12)**

* **Actors**: Admin.
* **Brief description**: This use case describes how Admin can Analysis any Course
* **Pre-conditions**: Login as Admin.
* **Post-condition**: get information about Exam like Course like Marks, percentage ….
* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Click “Course Analysis” |  |
| 1. Select semester |  |
| 1. Select Course |  |
| 1. Click “Analysis” |  |
|  | 1. check the information and request for get Course information’s |
|  | 1. show “Course information’s” |

**Critical scenario**:

No information about Course

 **User interface prototype:**

