The Syrian Private University

Faculty of Computer

and Informatics Engineering

الجامعـة الســـورية الخاصة

**كلية هندسة الحاسوب والمعلوماتية**

# Automation and maintenance of university laboratories

# أتمته وصيانة مخابر الجامعة

a report submitted to faculty of computer and information engineering SPU

***Supervised By:***

***Dr. Samir Jafar***

***Prepared By:***

***Mohamad Aljoumaa***

***Eman Merra***

***Tarek Yagi***

***Fadia Esmail***

***Wlaat Joumaa***

*ALL rights reserved © SPU 2015*

## ملخص:

قمنا في هذا المشروع ببناء تطبيق لإدارة مخابر الحاسوب في مختلف كليات الجامعة والحواسب الموجودة في كل منها ، حيث يمكن لمستخدم النظام أن يقوم بكافة عمليات الإدارة على مخابر كليات الجامعة من إضافة وتعديل وحذف، وكذلك يمكن التحكم بكافة عمليات الإدارة على مخابر الجامعة من إضافة وحذف وتعديل وداخل هذه المخابر توجد مجموعة من الحواسب التي يمكن لمستخدم النظام التحكم بكافة عمليات إدارتها من إضافة وتعديل وحذف وغيره.

إضافة إلى ما سبق فإن التطبيق يقدم مجموعة من الوظائف الأخرى مثل إضافة الأعطال إلى الحواسب الموجودة لدينا وهذه الأعطال تشمل جميع أنواع الأعطال على الحواسب الموجودة في المخابر، ويتيح التطبيق البحث عن مجموعة التقارير بالأخطاء التي تم إصلاحها والأخطاء الأخرى التي لم يتم إصلاحها بعد وطباعة هذه التقارير عبر أي طابعة موصولة مع الحاسب.



## Abstract :

We have in this project to build an application for managing the labs in the university and computers in them, where a user can system that does all the management processes on the faculties of the university to add, modify and delete, as can all management processes control the laboratories of the University of add, delete and modify, and within these laboratories There are a range of computers that a user can manage all system operations control to add, modify and delete the others.

In addition to the above, the application offers a range of other functions such as adding faults to computers available to us and these faults include all types of faults on the computers in the labs, and allows the application Find reports a group of bugs that have been closed and other errors that have not been closed and print these reports through any printer connected with the computer.

# فهرس المحتويات


| | |
|---|---|
| **أتمته وصيانة مخابر الجامعة** | **1** |
| ملخص عربي | **2** |
| ملخص انكليزي | **2** |
| إهداء | **5** |
| شكر وتقدير | **6** |
| مقدمة عن تاريخ البرمجة | **7+8** |
| الفصل الأول | **9** |
| 1.1تحليل النظام | **9** |
| 1.1.1طبقة قاعدة المعطيات | **9** |
| 1.1.2طبقة التعامل مع قاعدة المعطيات | **9** |
| 1.13طبقة العرض | **9** |
| 1.1.4طبقة التعامل مع قاعدة المعطيات | **9** |
| 1.1.5طبقة الواجهات | **9** |
| 1.2تحليل النظام | **10** |
| 1.2.1المتطلبات غير الوظيفية | **10** |
| 1.2.2المتطلبات الوظيفية | **11** |
| 1.2.3حالات الاستخدام | **12** |
| 1.2.4مخطط حالات الاستخدام | **13** |
| 1.2.5شرح حالات الاستخدام | **15** |
| الفصل الثاني | **33** |
| 2تصميم النظام | **33** |
| 2.1واجهات النظام | **33** |
| 2.2مخطط صفحات النظام | **41** |
| 2.3طبقة قاعدة المعطيات | **43** |
| الفصل الثالث | **46** |
| 3اختبار النظام | **46** |
| 3.1الدخول كمستخدم عادي | **46** |
| 3.2.الدخول كمسؤول النظام | **50** |
| الفصل الرابع | **52** |
| 4الخاتمة | **52** |
| | |
| 5.1ملحقات | **53** |
| 5.1ملحق1 | **53** |
| 5.2ملحق 2 | **55** |
| 5.3شرح الكود | **60** |
| المراجع والأدوات المستخدمة | **61** |

# فهرس الجداول




| | |
|---|---|
| مخطط 1.1 | 13 |
| مخطط 1.2 | 14 |
| جدول 1.1 | 15 |
| جدول 1.2 | 17 |
| جدول 1.3 | 18 |
| جدول 1.4 | 29 |
| جدول 1.5 | 20 |
| جدول 1.6 | 21 |
| جدول 1.7 | 23 |
| جدول 1.8 | 24 |
| جدول 1.9 | 25 |
| جدول 1.10 | 27 |
| جدول 1.11 | 28 |
| جدول 1.12 | 30 |
| جدول 1.13 | 31 |
| صورة 2.1 | 33 |
| صورة 2.2 | 34 |
| صورة 2.3 | 35 |
| صورة 2.4 | 35 |
| صورة 2.5 | 36 |
| صورة 2.6 | 37 |
| صورة 2.7 | 38 |
| صورة 2.8 | 38 |
| صورة 2.9 | 39 |
| صورة 2.10 | 39 |
| صورة 2.11 | 40 |
| صورة 2.12 | 41 |
| صورة 2.13 | 41 |
| مخطط 2.1 | 42 |
| صورة 2.14 | 43 |
| صورة 3.1 | 47 |
| صورة 3.2 | 48 |
| صورة 3.3 | 48 |
| صورة 3.4 | 49 |
| صورة 3.5 | 49 |
| صورة 3.6 | 50 |
| صورة 3.7 | 50 |
| صورة 3.7.1 | 51 |

# الإهداء

نهدي هذا المشروع المتواضع لأمنا وامتنا التي تربينا في حضنها وعشنا اجمل أيامنا طفولتنا وشبابنا حفظتنا وأنشأتنا لأمنا سوريا
ولأهلنا الذين عانوا ما عانوا لنكون يومنا مانحن عليه اليوم
ولكل معلم اهدانا علماً وسقانا من كرمه ولم يبخل علينا بتعبه وجهده
لينبتنا النبت الصالح لنكون على الطريق الصحيح في الحق والخير .

طلاب المشروع

أهديك هذا وانا خجل فكيف اهديك ما هو أقل فالعمر يوماً ودوما يرخص لك فأنا منك خلقت تكونت ثم أصبحت لأمي ....

محمد الجمعة

لمعلمي الاول في الحياة لشمعة حياتي ونور عيني اهديك هذا العمل المتواضع الذي لا يليق بمقامك لأبي

إيمان ميرة

# شكر تقدير

We like to thank All who contributed in the completion of this project and special thanks to:

Dr.SamIr Jafar

# Introduction(history of programing)

## Early history

During a nine-month period in 1840-1843, Ada Lovelace translated the memoir of Italian mathematician Luigi Menabrea about Charles Babbage's newest proposed machine, the Analytical Engine. With the article she appended a set of notes which specified in complete detail a method for calculating Bernoulli numbers with the Analytical Engine, recognized by some historians as the world's first computer program.

Herman Hollerith realized that he could encode information on punch cards when he observed that train conductors encode the appearance of the ticket holders on the train tickets using the position of punched holes on the tickets. Hollerith then encoded the 1890 census data on punch cards.

The first computer codes were specialized for their applications. In the first decades of the 20th century, numerical calculations were based on decimal numbers. Eventually it was realized that logic could be represented with numbers, not only with words. For example, Alonzo Church was able to express the lambda calculus in a formulaic way. The Turing machine was an abstraction of the operation of a tape-marking machine, for example, in use at the telephone companies. Turing machines set the basis for storage of programs as data in the von Neumann architecture of computers by representing a machine through a finite number. However, unlike the lambda calculus, Turing's code does not serve well as a basis for higher-level languages—its principal use is in rigorous analyses of algorithmic complexity.

Like many "firsts" in history, the first modern programming language is hard to identify. From the start, the restrictions of the hardware defined the language. Punch cards allowed 80 columns, but some of the columns had to be used for a sorting number on each card. FORTRAN included some keywords which were the same as English words, such as "IF", "GOTO" (go to) and "CONTINUE". The use of a magnetic drum for memory meant that computer programs also had to be interleaved with the rotations of the drum. Thus the programs were more hardware-dependent.

To some people, what was the first modern programming language depends on how much power and human-readability is required before the status of "programming language" is granted. Jacquard looms and Charles Babbage's Difference Engine both had simple, extremely limited languages for describing the actions that these machines should perform. One can even regard the punch holes on a player piano scroll as a limited domain-specific language, albeit not designed for human consumption.

### 1990s: the Internet age

The rapid growth of the Internet in the mid-1990s was the next major historic event in programming languages. By opening up a radically new platform for computer systems, the Internet created an opportunity for new languages to be adopted. In particular, the JavaScript programming language rose to popularity because of its early integration with the Netscape Navigator web browser. Various other scripting languages achieved widespread use in developing customized application for web servers such as PHP. The 1990s saw no fundamental novelty in imperative languages, but much recombination and maturation of old ideas. This era began the spread of functional languages. A big driving philosophy was programmer productivity. Many "rapid application development" (RAD) languages emerged, which usually came with an IDE, garbage collection, and were descendants of older languages. All such languages were object-oriented. These included Object Pascal, Visual Basic, and Java. Java in particular received much attention. More radical and innovative than the RAD languages were the new scripting languages. These did not directly descend from other languages and featured new syntaxes and more liberal incorporation of features. Many consider these scripting languages to be more productive than even the RAD languages, but often because of choices that make small programs simpler but large programs more difficult to write and maintain. Nevertheless, scripting languages came to be the most prominent ones used in connection with the Web

# الفصل الأول

## 1- تحليل النظام

### 1.1 بناء المشروع بطريقة الطبقات الثلاث:

قمنا في بناء هذا المشروع بتقسيم التنفيذ العملي وإنجاز الوظائف المطلوبة عن طريق إعتماد نموذج three tier (الطبقات الثلاث) وتم تنجيز هذه الطبقات بدءاً من الأسفل إلى الأعلى كما يلي:

#### 1.1.1 طبقة قاعدة المعطيات:

تمثل هذه الطبقة جداول قاعدة المعطيات الموجودة في النظام والمتمثلة بمجموعة من الجداول الموجودة لدينا.

#### 1.1.2 طبقة التعامل مع قاعدة المعطيات:

تمثل هذه الطبقة التوابع التي تقوم بعمليات الإستعلام والتعديل على جداول قاعدة المعطيات لدينا حيث يقوم كل تابع من التوابع بعملية محددة على جداول قاعدة المعطيات حسب الواجهة المطلوبة.

#### 1.1.3 طبقة presentation:

قمنا في هذه الطبقة بتنجيز جميع الواجهات الخاصة بالتطبيق لدينا انطلاقاً من الواجهات الخاصة بإدارة الكليات والمخابر الموجودة ثم تنجيز الواجهات الخاصة بإضافة الأعطال في الحواسب والمخابر ثم بعد ذلك وضع الحلول لمعالجة هذه الأخطاء الموجودة.

#### 1.1.4 طبقة التعامل مع قاعدة المعطيات:

تمثل هذه الطبقة جميع التوابع المستخدمة للتعامل مع قاعدة المعطيات، وهذه التوابع هي توابع الإستعلام عن قواعد المعطيات والتعديل عليها وجميع التوابع الأخرى من عمليات البحث وغيرها من العمليات الأخرى.

#### 1.1.5 طبقة الواجهات:

تمثل هذه الطبقة جميع واجهات تعامل المستخدم مع النظام، وهذه الواجهات هي جميع الواجهات الخاصة بالإضافة والتعديل والحذف على قواعد المعطيات والتي سنأتي على شرحها لاحقاً.

## 1.2 تحليل النظام

في هذه المرحلة سنقوم بتحليل النظام وذلك بوضع مخطط تفصيلي لحالات الاستخدام ، ومن ثم سنورد وصفا تفصيليا لكل حالة من حالات استخدام النظام.

### 1.2.1 المتطلبات غير الوظيفية العامة:

في الحالة العامة لدينا المتطلبات غير الوظيفية التالية:

1. يجب أن تحقق واجهات الاستخدام واجهات خاصة بالمستخدم العادي وواجهات خاصة بمدير النظام أي يجب أن تكون الواجهات التي تتيح للمستخدم إمكانية التعديل على المستخدمين متاحة لمدير النظام ولا يمكن للمستخدم العادي الوصول إليها. يمكن تحقيق ذلك مثلا بطلب إسم مستخدم وكلمة مرور للوصول إلى هذه الواجهات خاص بمدير النظام.

2. سهولة الاستخدام: يجب أن تكون الواجهات سهلة وواضحة بالنسبة لمستخدم النظام وتكون جميع الأوامر موضحة للمستخدم في واجهات البرنامج المتتالية.

3. استجابة النظام: يجب أن يحقق النظام سرعة استجابة للزبون لا تتجاوز الثانية الواحدة، حيث لا يمكن ان يقوم المستخدم بطلب أمر ما و الانتظار مدة طويلة لتنفيذه لأن معظم الأوامر في نظامنا هي أوامر بسيطة تتعلق بعملية إدارة المختبرات.

4. يجب أن يكون النظام صحيحاً (correctness)، أي أن يؤدي جميع وظائفه بالصورة الصحيحة

5. يجب توفير دليل للمستخدم يحوي كيفية التعامل مع النظام واستخدامه.

6. قابلية ومرونة تطوير التطبيق (flexibility)، أي يجب أن يكون النظام مبنياً بطريقة البرمجة غرضية التوجه وأن تكون صفوفه واضحة مما يتيح إمكانية تطويره بسهولة.

### 1.2.2 تحديد المتطلبات الوظيفية:

يجب للنظام أن يحقق تطبيق يؤمن العمليات التالية:

1. تتيح للمستخدم تسجيل الدخول إلى النظام، حيث لايمكن إلا لمستخدمين مسجلين في الموقع بالدخول إليه.
2. تتيح للمستخدم كافة عمليات الإدارة على الكليات الموجودة من إضافة وتعديل وحذف.
3. تتيح للمستخدم كافة عمليات الإدارة على المخابر من إضافة وتعديل وحذف.
4. تتيح للمستخدم إضافة حواسب جديدة إلى المخابر الموجودة في الكليات.
5. تتيح للمستخدم التعديل على الحواسب الموجودة أصلا في مخابر الكليات.
6. تتيح للمستخدم حذف الحواسب الموجودة أصلا في مخابر الكليات.
7. تتيح للمستخدم إدخال المشاكل والأخطاء الموجودة في المخابر.
8. تتيح للمدير حل المشاكل الموجودة في النظام والتي قام مستخدمون سابقون بإضافتها إلى التطبيق الموجود لدينا .
9. تتيح للمستخدم توليد التقارير الخاصة بالإصلاحات الموجودة في النظام حيث يمكن للمستخدم البحث واستعراض كافة التقارير المدخلة في النظام.

### 1.2.3 حالات الاستخدام:

سنقوم في هذه الخطوة بذكر جميع حالات الاستخدام بين النظام والمستخدم وفي الخطوة التالية سنقوم برسم مخطط حالات استخدام النظام ( Use Case Diagram) فتساعد هاتان الخطوتان في بناء الواجهات التي ستربط النظام مع المستخدم

يمكن تقسيم حالات الاستخدام السابقة إلى مايلي:

1. تسجيل الدخول:

يحدد زائر الموقع إسم المستخدم وكلمة السر الخاصة به من أجل الدخول إلى الموقع.

2. إدارة الكليات:

يقوم المستخدم في هذه الحالة بكافة عمليات الإدارة على الكليات من إضافة وحذف وتعديل حيث يمكن للمستخدم إضافة كلية وتعديل كلية وحذف كلية.

3. إدارة المخابر:

يقوم المستخدم في هذه الحالة بكافة عمليات الإدارة على المخابر الموجودة حيث تشمل عمليات الإدارة إضافة مخبر أو تعديل معلومات مخبر مضاف أو حذف مخبر موجود سابقاً .

4. إضافة حاسب جديد إلى مخبر:

يمكن للمستخدم إضافة مجموعة من الحواسب إلى المخابر الموجودة في الكليات حيث يحدد المستخدم كافة المعلومات اللازمة لإضافة الحاسب (مواصفات الحاسب من معالج وذاكرة وغيرها من المواصفات).

5. تعديل معلومات حاسب في مخبر:

يمكن للمستخدم تعديل معلومة حاسب ما مضاف إلى مخبر معين حيث يمكنه تعديل أي معلومة من المعلومات الخاصة بالحاسب الموجود في المخبر.

6. حذف حاسب في مخبر:

يمكن للمستخدم حذف أي حاسب من الحواسب الموجدة في المخابر.

7. إدخال المشاكل والأخطاء في المخابر:

يمكن للمستخدم إضافة مجموعة المشاكل والأخطاء الموجودة في المخابر، حيث يقوم النظام بتخزينها ضمن قاعدة المعطيات الخاصة به.

8. حل مشاكل المخابر:

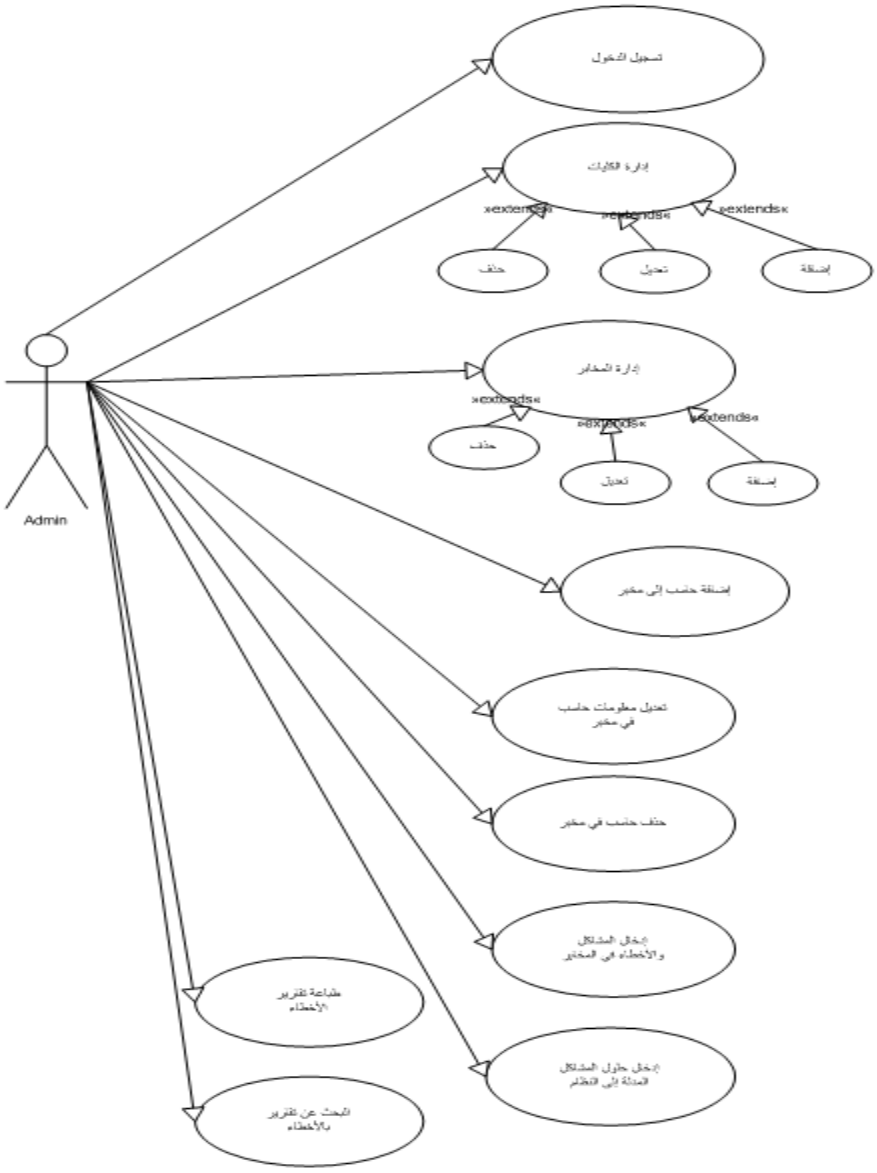
يمكن للمستخدم إقتراح حلول للمشاكل الموجودة في النظام، حيث يقوم النظام بتخزين هذه الحلول في قاعدة معطيات النظام.

9. توليد التقارير الخاصة بالنظام:

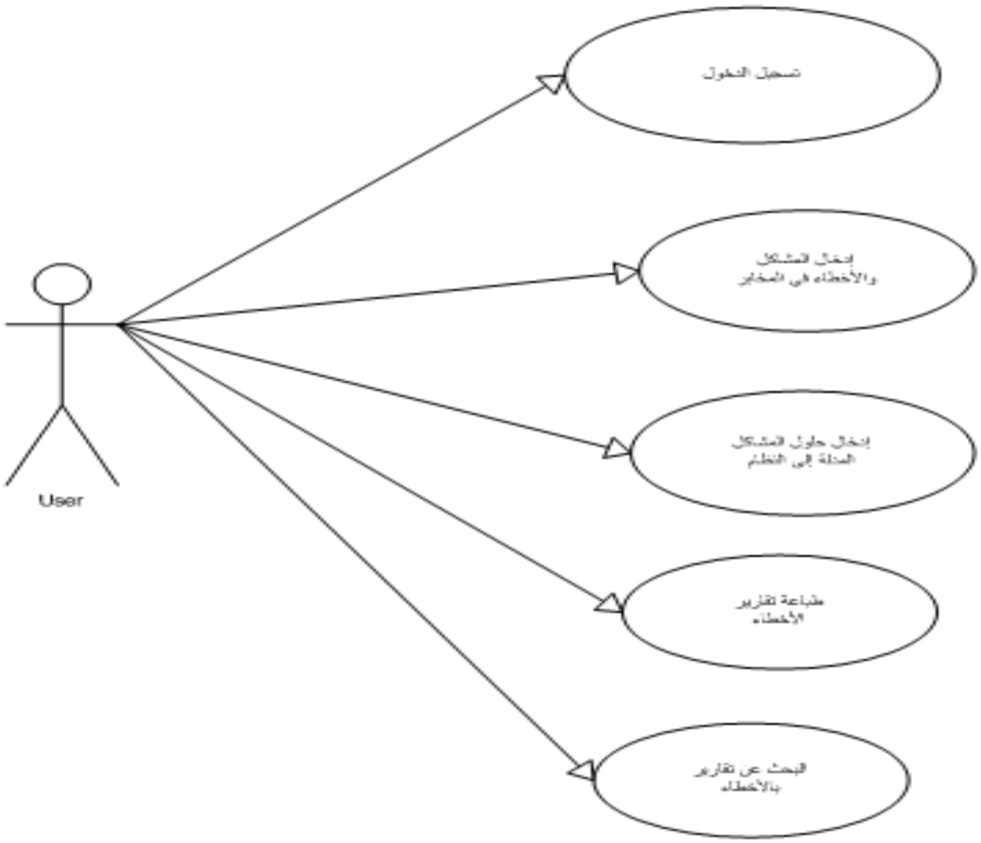
يمكن للمستخدم البحث في مجموعة التقارير الموجودة في قاعدة المعطيات لدى النظام، حيث يحدد المستخدم تاريخ البداية والنهاية الذي يريد البحث ضمنه.

### 1.2.4 مخطط حالات الاستخدام. (use case diagram)

يبين الشكل التالي حالات الاستخدام الخاصة بالنظام مع الفاعلين الأوليين والثانويين المشاركين بكل حالة. ملاحظة: قمنا بفصل حالات الإستخدام الخاصة بالمدير عن حالات الإستخدام الخاصة بالمستخدم العادي من أجل توضيح المخطط من أجل كلتا الحالتين.




مخطط1.1 حالات الاستخدام الخاصة بمدير النظام




مخطط 1.2 حالات الاستخدام الخاصة بالمستخدم العادي

### 1.2.5 شرح حالات الاستخدام (Expanded use case narrative)

1. **اسم حالة الاستخدام:** (تسجيل دخول)

الفاعلون الأوليون: المستخدم ومدير النظام.

الفاعلون الثانويون: لا يوجد

**ملخص :** يقوم المستخدم العادي أو مدير النظام بتشغيل النظام، ومن ثم يدخل اسم المستخدم وكلمة المرور فإذا كان اسم المستخدم وكلمة المرور صحيح فاني دخل إلى الصفحة الرئيسية الخاصة بالموقع والتي تسمح بكافة عمليات الإدارة.

**سير الأحداث:**

الشروط المسبقة:

قام المستخدم أو المدير بطلب الموقع.

تمّ إدخال إسم المستخدم وكلمة المرور.

تمّ التحقق من اسم المستخدم و كلمة المرور.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
|---|---|
| 1. تبدأ هذه الحالة عندما يطلب المستخدم أو المدي الأمر الدخول إلى الموقع. | |
| | 2. يطلب النظام من المستخدم أو المدير إدخال اسم المستخدم. |
| 3. يدخل المستخدم أو المدير اسم المستخدم. | |
| | 4. يطلب النظام من المستخدم أو المدير إدخال كلمة المرور. |
| 5. يدخل المدير كلمة المرور. | |
| | 6. يعرض النظام للمستخدم الصفحة الرئيسية للموقع والتي يمكن من خلالها إدارة كافة وظائف الموقع. |

جدول 1.1 سير احداث تسجيل الدخول

المسارات البديلة:

- اسم المستخدم المدخل غير صحيح

يبدأ هذا المسار عند النقطة 3 من المسار الأساسي الناجح.

4. يبقى النظام في نفس الحالة ويطلب منه إعادة إدخال اسم المستخدم وكلمة المرور.

كلمة المرور المدخلة غير صحيحة

يبدأ هذا المسار عند النقطة 5 من المسار الأساسي

6. يبقى النظام في نفس الحالة ويطلب منه إعادة إدخال اسم المستخدم وكلمة المرور.

الشروط اللاحقة:

1. تم تشغيل النظام

2. تم الدخول إلى الصفحة الرئيسية الخاصة بالموقع.

2. **اسم حالة الاستخدام:** (إضافة كلية.)

الفاعلون الأوليون: مدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص :** يقوم مدير النظام بتشغيل النظام ويطلب من النظام الأمر إضافة كلية.

**سير الأحداث:**

الشروط المسبقة:

قام المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر إضافة الكلية.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
| --- | --- |
| 1. تبدأ هذه الحالة عندما يريد مدير النظام إضافة كلية.<br>3. يقوم المدير بتحديد الاسم. | 2. يطلب النظام من المدير تحديد اسم الكلية التي يريد إضافتها.<br>4. يقوم النظام بإضافة الكلية السابقة إلى قاعدة المعطيات لديه. |

جدول 1.2 سير احداث إضافة كلية

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ إضافة كلية إلى النظام.

### 3. **اسم حالة الاستخدام**: (تعديل كلية.)

الفاعلون الأوليون: مدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص** : يقوم مدير النظام بتشغيل النظام ويطلب من النظام الأمر تعديل كلية.

**سير الأحداث:**

الشروط المسبقة:

قام المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر تعديل الكلية.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
|---|---|
| 1. تبدأ هذه الحالة عندما يريد مدير النظام تعديل معلومات كلية. | |
| | 2. يطلب النظام من المدير تحديد الكلية التي يريد تعديلها. |
| 3. يقوم المدير بتحديد الكلية. | |
| | 4. يطلب النظام من المستخدم إدخال المعلومات الجديدة الخاصة بالكلية. |
| 5. يدخل المستخدم المعلومات الجديدة ويطلب أمر التعديل. | |
| | 6.يقوم النظام بتعديل المعلومات القديمة للكلية وتحديثها. |

جدول 1.3 سير احداث تعديل كلية

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ تعديل معلومات كلية محددة في النظام.

4. **اسم حالة الاستخدام:** (حذف كلية.)

الفاعلون الأوليون: المستخدم ومدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص :** يقوم مدير النظام بتشغيل النظام ويطلب من النظام الأمر حذف كلية.

**سير الأحداث:**

الشروط المسبقة:

قام المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر حذف الكلية.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
| --- | --- |
| 1. تبدأ هذه الحالة عندما يريد مدير النظام حذف كلية. | |
| | 2. يطلب النظام من المدير تحديد الكلية التي يريد حذفها. |
| 3. يقوم المدير بتحديد الكلية. | |
| | 4. يقوم النظام بحذف الكلية المحددة. |

جدول 1.4 سير احداث حذف كلية

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ حذف كلية محددة في النظام.

5. **اسم حالة الاستخدام:** (إضافة مخبر.)

الفاعلون الأوليون: مدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص :** يقوم مدير النظام بتشغيل النظام ويطلب من النظام الأمر إضافة مخبر.

**سير الأحداث:**

الشروط المسبقة:

قام المستخدم أو المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر إضافة مخبر إلى كلية محددة.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
|---|---|
| 1. تبدأ هذه الحالة عندما يريد مدير النظام إضافة مخبر. | |
| | 2. يطلب النظام من المدير تحديد اسم الكلية التي يريد إضافة المخبر إليها. |
| 3. يقوم المدير بتحديد الكلية. | |
| | 4. يطلب النظام من المستخدم إدخال معلومات المخبر الذي يريد إضافته. |
| 5. يدخل المستخدم معلومات المخبر المراد إضافته. | |
| | 6. يقوم النظام بإضافة المخبر إلى الكلية المحددة في النظام. |

جدول 1.5 سير احداث إضافة مخبر

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ إضافة مخبر إلى كلية محددة في النظام.

6. **اسم حالة الاستخدام:** (تعديل مخبر.)

الفاعلون الأوليون: مدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص :** يقوم مدير النظام بتشغيل النظام ويطلب من النظام الأمر تعديل مخبر.

**سير الأحداث:**

الشروط المسبقة:

قام المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر تعديل مخبر محدد.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
|---|---|
| 1. تبدأ هذه الحالة عندما يريد مدير النظام تعديل معلومات مخبر. | |
| | 2. يطلب النظام من المدير تحديد الكلية التي تحوي المخبر. |
| 3. يقوم المدير بتحديد الكلية. | |
| | 4. يطلب النظام من المدير تحديد المخبر المراد تعديل معلوماته. |
| 5. يدخل المدير المعلومات الجديدة الخاصة بالمخبر المراد تعديله. | 6.يقوم النظام بتعديل معلومات المخبر |

جدول 1.6 سير احداث تعديل مخبر

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ تعديل معلومات مخبر ما في النظام.

7. **اسم حالة الاستخدام:** (حذف مخبر.)

الفاعلون الأوليون: مدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص** : يقوم مدير النظام بتشغيل النظام ويطلب من النظام الأمر حذف مخبر.

**سير الأحداث:**

الشروط المسبقة:

قام المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر حذف مخبر.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
| --- | --- |
| 1. تبدأ هذه الحالة عندما يريد مدير النظام حذف مخبر. | |
| | 2. يطلب النظام من المدير تحديد المخبرالذي يريد حذفه. |
| 3. يقوم المدير بتحديد المخبر. | |
| | 4. يقوم النظام بحذف المخبر المحدد. |

جدول 1.7 سير احداث حذف مخبر

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ حذف مخبر محدد في النظام.

8. **اسم حالة الاستخدام**: (إضافة حاسب إلى مخبر.)

الفاعلون الأوليون: مدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص** : يقوم مدير النظام بتشغيل النظام ويطلب من النظام الأمر إضافة حاسب إلى مخبر.

**سير الأحداث**:

الشروط المسبقة:

قام المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر إضافة حاسب إلى مخبر في كلية محددة.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
|---|---|
| 1. تبدأ هذه الحالة عندما يريد مدير النظام إضافة حاسب إلى مخبر. | |
| | 2. يطلب النظام من المدير تحديد اسم الكلية. |
| 3. يقوم المدير بتحديد الكلية. | |
| | 4. يطلب النظام من المدير تحديد المخبر. |
| 5. يقوم المدير بتحديد المخبر. | |
| | 6. يطلب النظام من المدير إدخال معلومات الحاسب المراد إضافته إلى المخبر. |
| 7. يدخل المدير معلومات الحاسب. | |
| | 8. يقوم النظام بإضافة الحاسب الجديد إلى المخبر. |

جدول 1.8 سير احداث إضافة حاسب الى مخبر

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ إضافة حاسب إلى مخبر في كلية محددة

9. **اسم حالة الاستخدام**: (تعديل حاسب في مخبر.)

الفاعلون الأوليون: مدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص** : يقوم مدير النظام بتشغيل النظام ويطلب من النظام الأمر تعديل حاسب في مخبر.

**سير الأحداث**:

الشروط المسبقة:

قام المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر تعديل حاسب في مخبر.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
|---|---|
| 1. تبدأ هذه الحالة عندما يريد مدير النظام تعديل حاسب في مخبر. | |
| | 2. يطلب النظام من المدير تحديد اسم الكلية. |
| 3. يقوم المدير بتحديد الكلية. | |
| | 4. يطلب النظام من المدير تحديد المخبر. |
| 5. يقوم المدير بتحديد المخبر. | |
| | 6. يطلب النظام من المدير تحديد الحاسب المراد تعديله. |
| 7. يحدد المدير معلومات الحاسب الجديدة. | |
| | 8. يقوم المدير بتعديل معلومات الحاسب. |

جدول 1.9 سير احداث تعديل حاسب في مخبر

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ تعديل معلومات الحاسب المحدد في المخبر.

10. **اسم حالة الاستخدام:** (حذف حاسب في مخبر.)

الفاعلون الأوليون: مدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص** : يقوم مدير النظام بتشغيل النظام ويطلب من النظام الأمر حذف حاسب في مخبر.

**سير الأحداث:**

الشروط المسبقة:

قام المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر حذف حاسب في مخبر.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
| --- | --- |
| 1. تبدأ هذه الحالة عندما يريد مدير النظام حذف حاسب في مخبر. | |
| | 2. يطلب النظام من المدير تحديد الكلية. |
| 3. يقوم المدير بتحديد الكلية. | |
| | 4. يطلب النظام من المدير تحديد المخبر. |
| 5. يقوم المدير بتحديد المخبر. | |
| | 6. يطلب النظام من المدير تحديد الحاسب المراد حذفه. |
| 7. يحدد المدير الحاسب المراد حذفه. | |
| | 8. يقوم النظام بحذف الحاسب المحدد . |

جدول 1.10 سير احداث حذف حاسب في مخبر

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ حذف الحاسب المحدد في المخبر.

11. **اسم حالة الاستخدام**: (إدخال المشاكل والأخطاء في المخابر.)

الفاعلون الأوليون: المستخدم ومدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص** : يقوم المستخدم العادي أو مدير النظام بتشغيل النظام ويطلب من النظام الأمر إضافة مشكلة معينةفي حاسب موجود في مخبر ما في كلية ما.

**سير الأحداث:**

الشروط المسبقة:

قام المستخدم أو المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر إدخال مشكلة في حاسب ما في مخبر معين

السيناريو الأساسي الناجح:

| الفاعل | النظام |
|---|---|
| 1. تبدأ هذه الحالة عندما يريد المستخدم أو مدير النظام إضافة مشكلة في حاسب. | |
| | 2. يطلب النظام من المستخدم أوالمدير تحديد اسم الكلية. |
| 3. يقوم المستخدم أو المديرتحديد الكلية. | |
| | 4. يطلب النظام من المستخدم تحديد المخبر. |
| 5. يقوم المستخدم بتحديد المخبر. | |
| | 6. يطلب النظام من المستخدم تحديد الحاسب |
| 7. يقوم المستخدم بتحديد الحاسب. | |
| | 8. يطلب النظام من المستخدم إدخال المشكلة في الحاسب المحدد. |
| 9. يقوم المستخدم بتحديد المشكلة، ويوصف كافة المعلومات الخاصة بها. | |
| | 10. يقوم بالنظام بإدخال المشكلة ضمن قاعدة المعطيات الخاصة به. |

جدول 1.11 سير احداث إدخال المشاكل والأخطاء في مخبر

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ إضافة مشكلة في حاسب موجود في مخبر في كلية محددة.

12. **اسم حالة الاستخدام:** (إدخال حل لمشكلة معينة في حاسب معين في أحد المخابر.)

الفاعلون الأوليون: المستخدم ومدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص** : يقوم المستخدم العادي أو مدير النظام بتشغيل النظام ويطلب من النظام الأمر إضافة حل لمشكلة معينة في حاسب مدخلة سابقاً.

**سير الأحداث:**

الشروط المسبقة:

قام المستخدم أو المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر إدخال حل مشكلة في حاسب ما في مخبر معين.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
|---|---|
| 1. تبدأ هذه الحالة عندما يريد المستخدم أو مدير النظام إضافة حل لمشكلة في مخبر. | |
| | 2. يطلب النظام من المستخدم أوالمدير تحديد اسم الكلية. |
| 3. يقوم المستخدم أو المديرتحديد الكلية. | |
| | 4. يطلب النظام من المستخدم تحديد المخبر. |
| 5. يقوم المستخدم بتحديد المخبر. | |
| | 6. يطلب النظام من المستخدم تحديد الخطأ الذي يريد حله في حال وجود أكثر من خطأ. |
| 7. يقوم المستخدم بتحديد الخطأ ومن ثم يحدد الحل له ويغلق الخطأ نهائياً. | |

جدول 1.12 سير احداث إدخال حل لمشكلة معينة في حاسب معين في احد المخابر

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّ إضافة حل لمشكلة معينة موجودة في حاسب في مخبر ما.

13. **اسم حالة الاستخدام:** (توليد التقارير الخاصة بالنظام.)

الفاعلون الأوليون: المستخدم ومدير النظام.

الفاعلون الثانويون: لايوجد

**ملخص** : يقوم المستخدم العادي أو مدير النظام بتشغيل النظام ويطلب من النظام توليد تقرير خاص بالمشاكل أو الأخطاء التي تم أو لم يتم معالجتها في النظام.

**سير الأحداث:**

الشروط المسبقة:

قام المستخدم أو المدير بتشغيل النظام.

تم إدخال إسم المستخدم وكلمة المرور الخاصتين به وهما صحيحين.

تم طلب الأمر توليد التقارير الخاصة بالنظام.

السيناريو الأساسي الناجح:

| الفاعل | النظام |
|---|---|
| 1. تبدأ هذه الحالة عندما يريد المستخدم أو مدير النظام البحث عن تقرير بخطأ معين. | |
| | 2. يطلب النظام من المستخدم أوالمدير تحديد اسم الكلية. |
| 3. يقوم المستخدم أو المديربتحديد الكلية. | |
| | 4. يطلب النظام من المستخدم تحديد زمن بداية للتقارير التي يريد البحث عنها. |
| 5. يقوم المستخدم بتحديد تاريخ البداية. | |
| | 6. يطلب النظام من المستخدم تحديد تاريخ نهاية البحث عن التقارير. |
| 7. يقوم المستخدم بتحديد تاريخ نهاية تاريخ البحث عن التقارير. | |
| | 8. يطلب النظام من المستخدم تحديد الأخطاء التي البحث عنها فيما إذا كانت منتهية أو لا. |
| 9. يقوم المستخدم بتحديد طبيعة الأخطاء التي يريد البحث عنها (منتهية أو غير منتهية)، ويطلب أمر البحث. | |
| | 10. يقوم النظام بعرض التقارير الخاصة بالأخطاء المخزنة في قاعدة المعطيات وفق القيود المحددة في خيارات البحث. |

جدول 1.13 سير احداث توليد التقارير الخاصة بالنظام

المسارات البديلة:

لايوجد

الشروط اللاحقة:

1. تم تشغيل النظام.

2. تم إدخال إسم المستخدم وكلمة المرور الصحيحتين وأصبح المستخدم أو مدير النظام جاهزا لتنفيذ العمليات المتاحة.

3. تمّت عملية البحث عن تقارير بمجموعة الأخطاء المخزنة في النظام وفق القيود والشروط المحددة من قبل المستخدم.

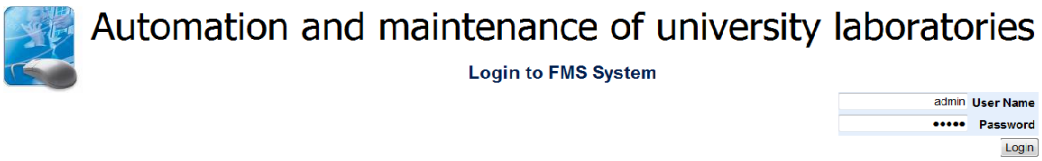
# الفصل الثاني

## 2-تصميم النظام

### 2.1 واجهات النظام:

قمنا بتنجيز واجهات النظام باللغتين العربية والإنكليزية من أجل سهولة الإستخدام حسب متطلبات المستخدم.

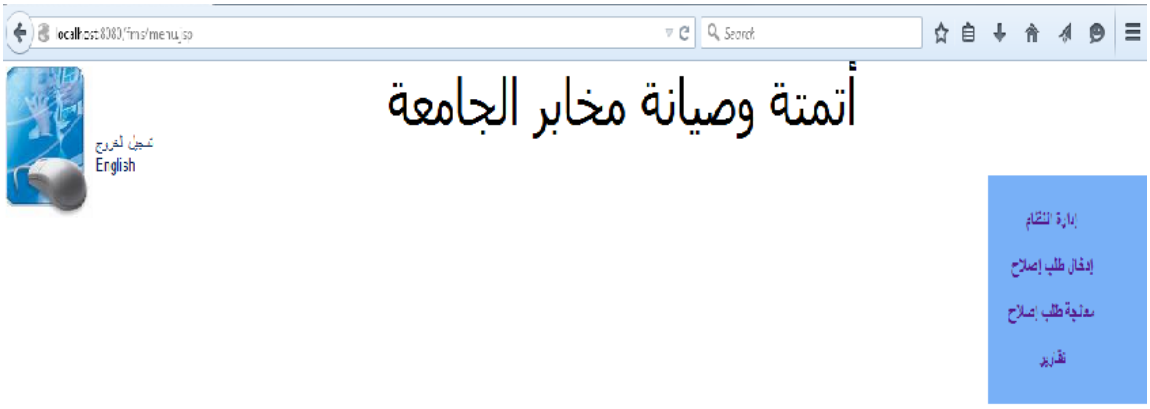
سنقوم فيما يلي بعرض كل واجهات النظام باللغتين العربية والإنكليزية مع شرح لوظيفة كل واجهة من الواجهات التالية.



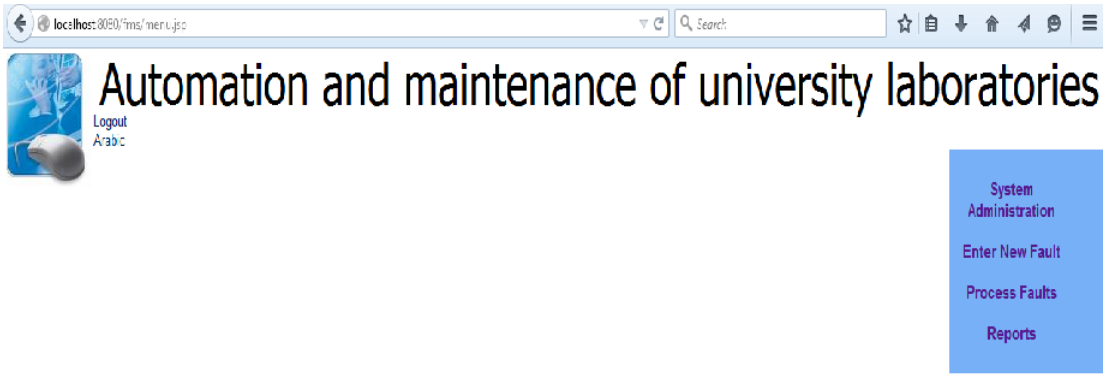

صورة 2.1 - واجهة الدخول باللغة الإنكليزية

صورة 2.2 - واجهة الدخول باللغة العربية

يقوم المستخدم في واجهة الدخول سواء باللغة العربية أو الإنكليزية بإدخال إسم المستخدم الخاص به، فإذا كان اسم المستخدم وكلمة السر صحيحتين فإن المستخدم يمكنه الدخول إلى واجهة النظام الرئيسية، والقيام بكافة عمليات الإدارة والتي سنأتي على ذكرها لاحقاً.




صورة 2.3 - واجهة النظام الرئيسية باللغة العربية

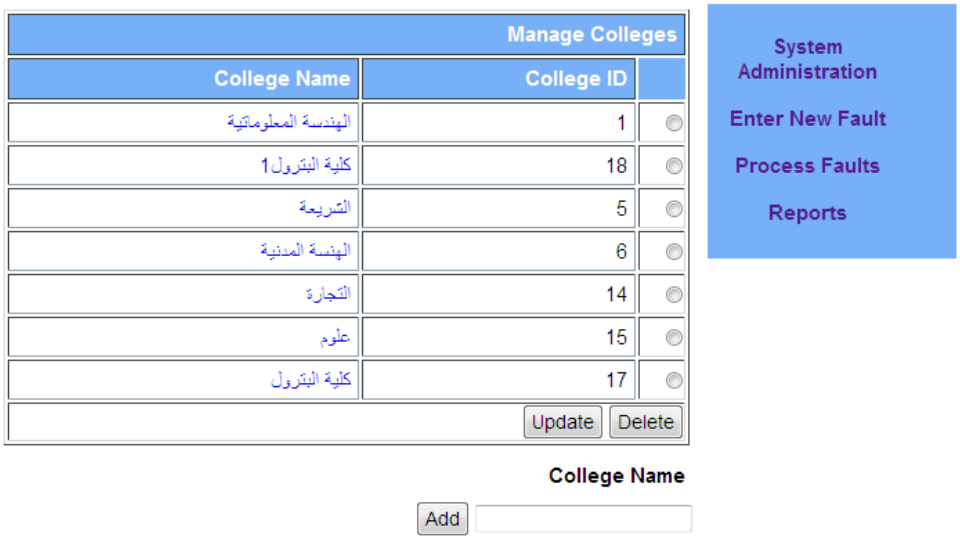



صورة 2.4 - واجهة النظام الرئيسية باللغة الإنكليزية

تمثل الواجهتين السابقتين الواجهة الرئيسية للنظام باللغتين العربية والإنكليزية حيث يمكن لمستخدم النظام من خلال الواجهتين السابقتين القيام بكافة عمليات الإدارة على النظام السابق من إدارة للكليات والمخابر وللأخطاء وغيرها من الروابط التي نلاحظها ضمن الواجهة الرئيسية للنظام.

# آتمتة وصيانة مخابر الجامعة

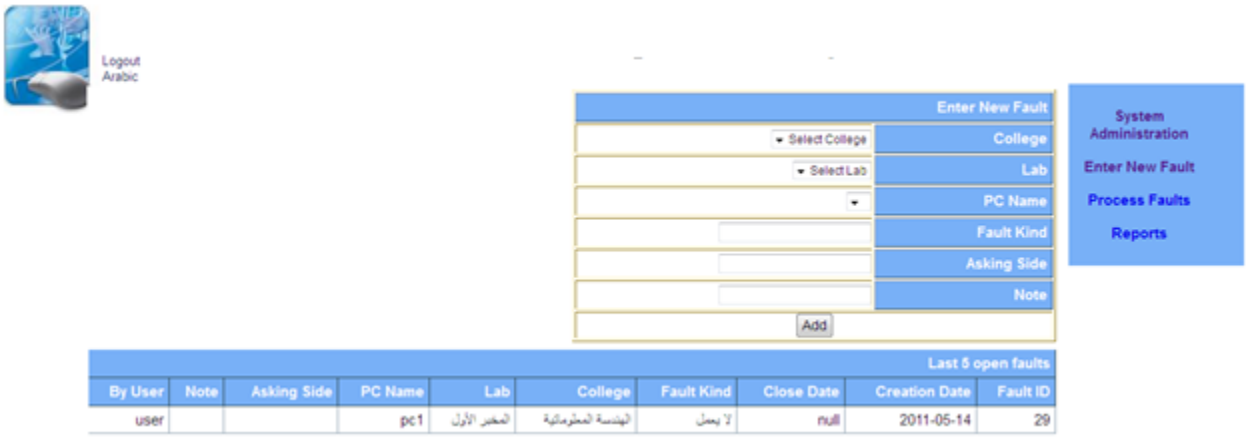
صورة 2.5 - واجهة إدارة الكليات باللغة العربية




صورة 2.6 - واجهة إدارة الكليات باللغة الإنكليزية

يمكننا من خلال الواجهتين السابقتين الخاصتين بإدارة الكليات إدارة كافة العمليات على الكليات الموجودة في قاعدة المعطيات لدينا من إضافة وتعديل وحذف.

صورة 2.7 - واجهة إدخال طلب إصلاح باللغة العربية




صورة 2.8 - واجهة إدخال طلب إصلاح باللغة الإنكليزية

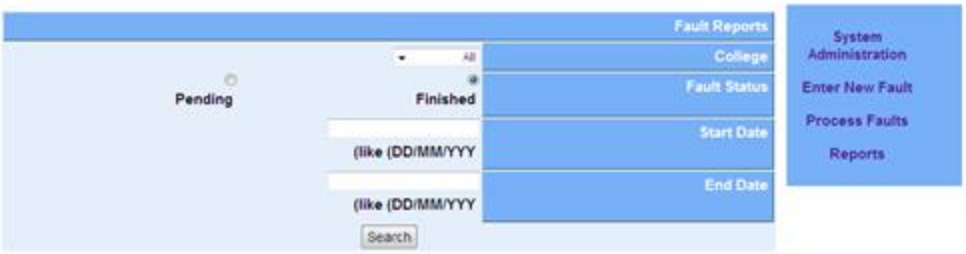
يمكننا من خلال الواجهتين السابقتين إدخال طلب إصلاح خاص بحاسب معين موجود في مخبر ما حيث نقوم بتحديد كافة المعلومات التي تحدد طلب الإصلاح من توصيف للعطل والجهة الطالبة له ويمكننا كذلك إضافة ملاحظات خاصة بالعطل المحدد.

صورة 2.9 - واجهة إغلاق الأخطاء في النظام بالواجهة العربية

صورة 2.10 - واجهة إغلاق الأخطاء باللغة الإنكليزية

يمكننا من خلال الواجهتين السابقتين إغلاق خطأ ما حيث نقوم بتحديد أولاً بالبحث عن الأخطاء الموجودة لدينا عن طريق تحديد الكلية، ومن ثم نقوم بالبحث عن قائمة الأخطاء الموجودة في قاعدة المعطيات لدينا. إذا أردنا إغلاق خطأ ما نقوم بتحديد الخطأ المراد إغلاقه ومن ثم ندخل ملاحظات الإصلاح الخاصة بالخطأ ونطلب أمر الإصلاح ليتم تخزين كافة معلومات الإصلاح في قاعدة المعطيات.

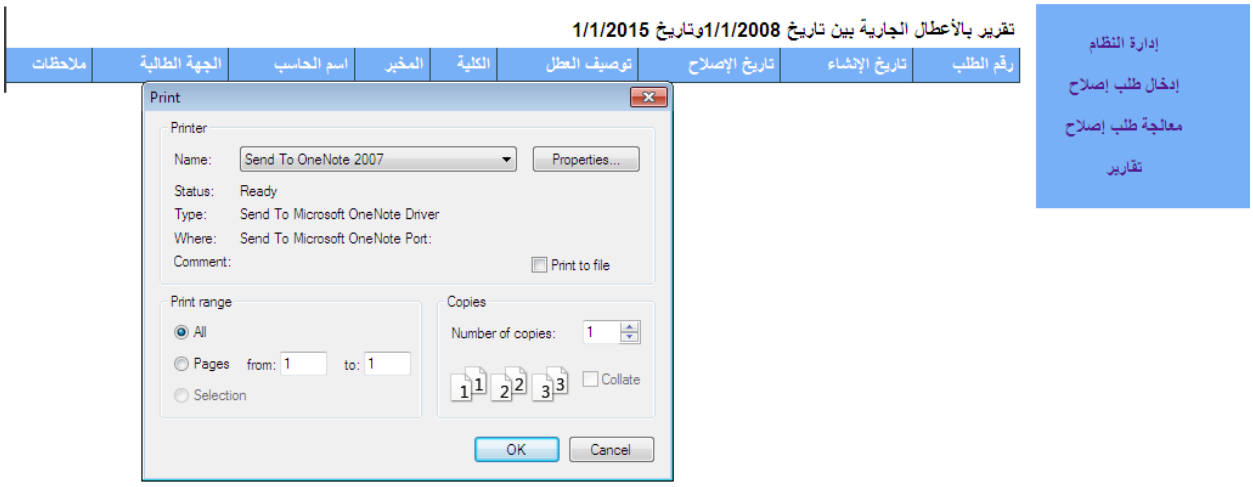
صورة 2.11 - واجهة البحث عن التقارير باللغة العربية




صورة 2.12 - واجهة البحث عن التقارير باللغة الإنكليزية

يمكننا من خلال الواجهتين السابقتين البحث عن التقارير الممخزنة لدينا حيث نقوم بتحديد الكلية وطبيعة الخطأ (الخطأ معالج أو لا) وتاريخ البداية وتاريخ النهاية للتقارير المراد البحث عنها.

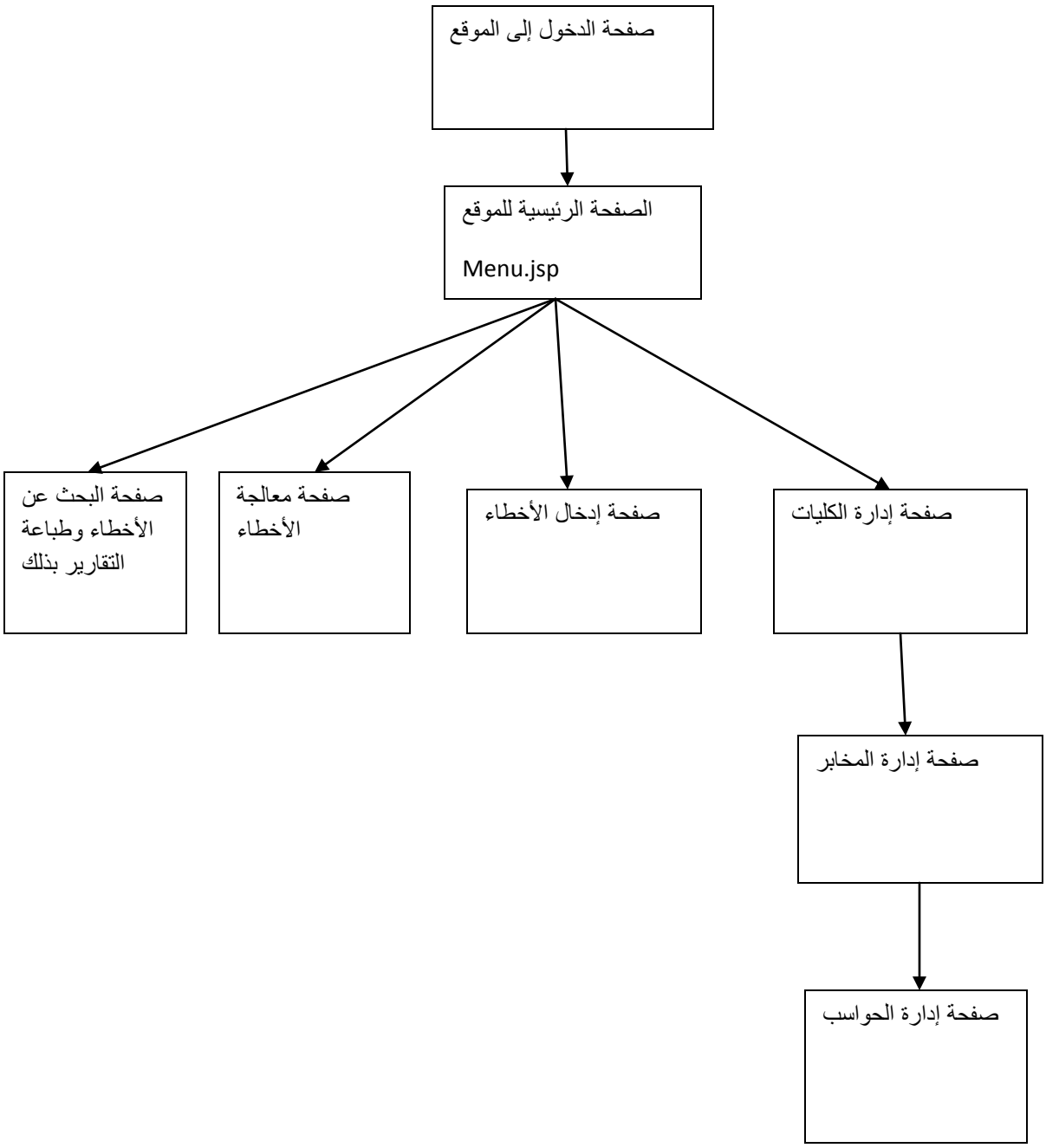
بعد تحديد المعاملات السابقة نقوم بطلب أمر البحث ليتم عرض التقارير التي تحقق القيود السابقة في واجهة خاصة بعرض التقارير كما سيلي لاحقاً.




صورة 2.13 - واجهة عرض التقارير باللغة العربية

يتم من خلال الواجهتين السابقتين عرض مجموعة التقارير التي تحقق شروط البحث والتي تم البحث عنها حسب الواجهة السابقة، وكما نلاحظ أن الواجهة السابقة تتيح عملية الطباعة لمجموعة التقارير الناتجة عن عملية البحث عبر اختيار أي طابعة متصلة مع الحاسب وطلب أمر الطباعة.

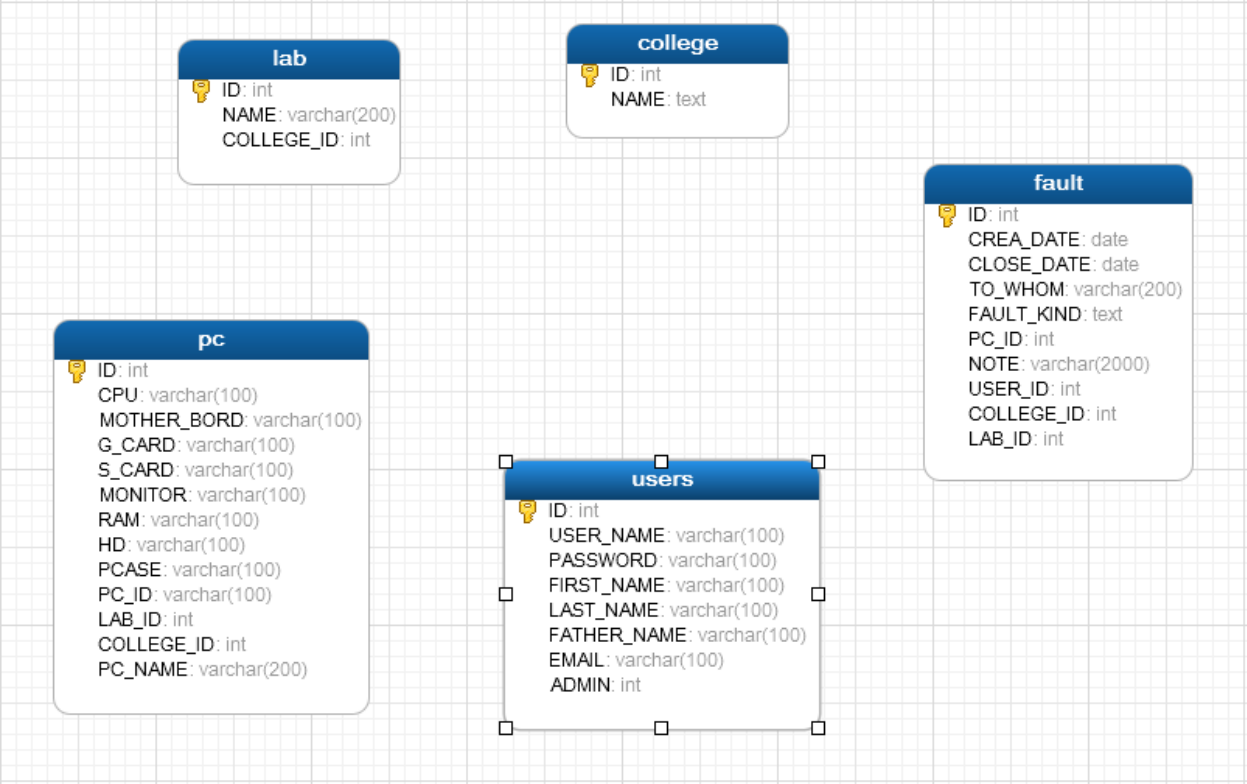
## 2.2 (مخطط صفحات النظام):




مخطط 2.1 مخطط صفحات النظام

## 2.3طبقة قواعد المعطيات:

سنقوم هنا بوضع مخطط لقاعدة المعطيات لدينا مع شرح للجداول الموجودة فيها.




صورة 2.14 - قاعدة البيانات المستخدمة

# 1.2.9جداول قاعدة المعطيات:

### الجدول college:

يمثل هذه الجدول كليات النظام (كلية الهندسة المعلوماتية، كلية البترول وغيرها من الكليات الأخرى)، ويحوي المتحولات التالية:

المتحول Id:

يمثل الرقم المعرف للكلية.

المتحول NAME:

يمثل إسم الكلية.

## الجدول lab:

يمثل هذا الجدول المخابر الموجودة لدينا في قاعدة المعطيات، حيث أن كل مخبر من هذه المخابر تابع لكلية معينة.

يحوي هذه الجدول المتحولات التالية:

المتحول ID:

يمثل الرقم المعرف الخاص بالمخبر.

المتحول NAME:

يمثل اسم المخبر.

المتحول COLLEGE_ID:

يمثل الكلية التي يتبع لها المخبر.

## الجدول pc:

يمثل هذه الجدول جميع الحواسب الموجودة لدينا في المخابر، حيث أن كل حاسب من هذه الحواسب تابع لمخبر معين موجود في كلية معينة.

يحوي هذه الجدول المتحولات التالية:

المتحول ID:

يمثل الرقم المعرف الخاص بالحاسب.

المتحول cpu:

يمثل مواصفات الحاسب العامة.

المتحول MOTHER_BORD:

يمثل اللوحة الأم الخاصة بالحاسب.

المتحول G_CARD:

يمثل نوع كرت الشاشة الموضوع في الحاسب.

المتحول S_CARD:

يمثل نوع كرت الشاشة (مدمج أو منفصل)

المتحول MONITOR:

يمثل نوع شاشة الحاسب.

المتحول RAM:

يمثل ذاكرة الحاسب.

المتحول HD:

يمثل الهارد الخاص بالحاسب.

المتحول PCASE:

يمثل نوع ال case للحاسب.

المتحول LAB_ID:

يمثل المخبر الذي يتبع له الحاسب.

المتحول COLLEGE_ID:

يمثل الكلية التي يتبع لها الحاسب.

المتحول PC_NAME:

يمثل الاسم الخاص بالحاسب (مثلاً الحاسب رقم 6 او 7 وهكذا).

## الجدول users:

يمثل هذه الجدول جميع مستخدمي النظام.

يحوي هذه الجدول المتحولات التالية:

المتحول ID:

يمثل الرقم المعرف الخاص بالمستخدم.

المتحول USERNAME:

يمثل اسم المستخدم.

المتحول PASSWORD:

يمثل كلمة السر الخاصة بالمستخدم.

المتحول FIRSTNAME:

يمثل الاسم الأول الخاص بالمستخدم.

المتحول LASTNAME:

يمثل الاسم الثاني الخاص بالمستخدم.

المتحولFATHERNAME:

يمثل اسم الاب الخاص بالمستخدم.

المتحول EMAIL:

يمثل البريد الإلكتروني للمستخدم.

المتحول ADMIN:

يمثل فيما إذا كان المستخدم الحالي مستخدم عادي للنظام أو مدير له وذلك من أجل إعطاء الصلاحيات المناسبة لهُ.

## الجدول fault:

يمثل هذه الجدول جميع الأخطاء المدخلة في النظام والمقصود بالأخطاء هنا الأعطال في الحواسب أو اي خطأ أخر موجود في مخبر من المخابر المحددة لديه.

يحوي هذه الجدول المتحولات التالية:

المتحول ID:

يمثل الرقم المعرف الخاص بالمستخدم.

المتحول CREA_DATE:

يمثل تاريخ إضافة الخطأ أو العطل إلى النظام.

المتحولCLOSE_DATE:

يمثل تاريخ إقفال الخطأ.

المتحول TO_WHOM:

يمثل الكلية التي يتبع لها الخطأ.

المتحول FAULT_KIND:

يمثل نوع الخطأ (عطل في الهارد أو الرام).

المتحولPC_ID:

يمثل الرقم المعرف الخاص بالحاسب المعطل.

المتحول NOTE:

يمثل ملاحظات حول الخطأ الموجود.

المتحول USER_ID:

يمثل الرقم المعرف للمستخدم الذي أضاف الخطأ.

المتحول COLLEGE_ID:

يمثل الكلية التي يتبع لها الحاسب المعطل.

المتحول LAB_ID:

يمثل المخبر الذي يتبع له الحاسب المعطل.

# الفصل الثالث

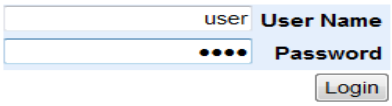
## 3-اختبار النظام:

سنقوم في هذه المرحلة بوضع نموذج اختبار كامل لعملية إضافة خطأ إلى النظام من قبل مستخدم عادي، ومن ثم سيقوم المدير بالدخول ومعالجة الخطأ السابق المدخل من قبل المستخدم العادي.

المرحلة الأولى من الاختبار:

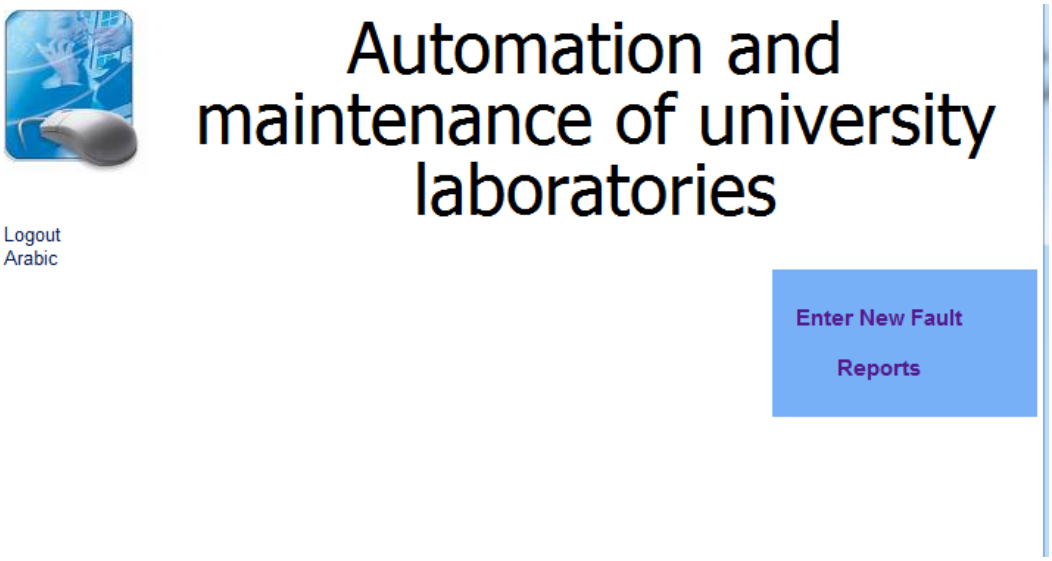
الدخول كمستخدم عادي للنظام وإدخال مشكلة في حاسب موجود في مخبر معين.

### 3.1الدخول إلى النظام كمستخدم عادي للنظام:




صورة 3.1 - تسجيل الدخول كمستخدم عادي

1. إدخال خطأ من قبل المستخدم، ويضاف إلى أحد الحواسب الموجودة في كلية معينة في مخبر محدد.



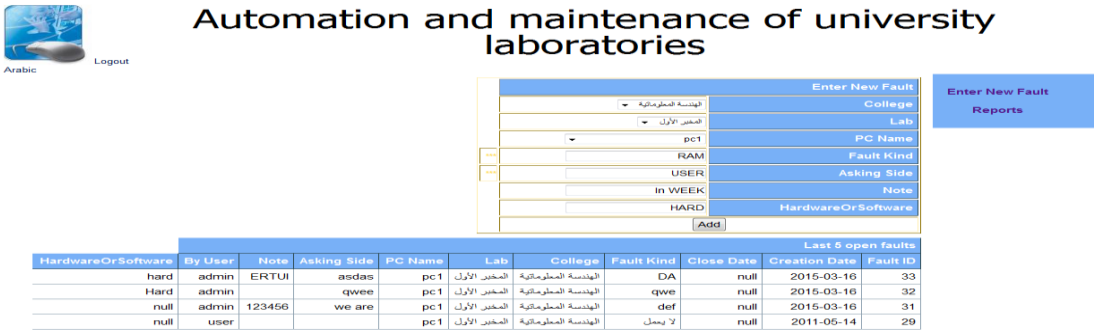

صورة 3.2 - واجهة المستخدم العادي باللغة الانكليزية

نلاحظ ضمن الواجهة السابقة الخيارات المتاحة للمستخدم العادي، وهي:

إدخال خطأ أو مشكلة معينة.

البحث عن التقارير أو المشاكل والأخطاء المدخلة وطباعتها.

سنقوم في هذه المرحلة بإختبار إدخال خطأ ومعالجته، لذلك سنقوم في المرحلة اللاحقة بالدخول إلى النظام وإدخال مشكلة في احد الحواسب حيث سنقوم بإدخال المشكلة عن طريق المستخدم الحالي user.




صورة 3.3 – واجهة ادخال الأخطاء والمشاكل

نلاحظ هنا في الصورة التالية إضافة الخطأ بنجاح إلى قاعدة المعطيات الموجودة (أول سطر من الجدول التالي):

| | Last 5 open faults | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| HardwareOrSoftware | By User | Note | Asking Side | PC Name | Lab | College | Fault Kind | Close Date | Creation Date | Fault ID |
| HARD | user | In WEEK | USER | pc1 | المخبر الأول | الهندسة المعلوماتية | RAM | null | 2015-03-23 | 34 |
| hard | admin | ERTUI | asdas | pc1 | المخبر الأول | الهندسة المعلوماتية | DA | null | 2015-03-16 | 33 |
| Hard | admin | | qwee | pc1 | المخبر الأول | الهندسة المعلوماتية | qwe | null | 2015-03-16 | 32 |
| null | admin | 123456 | we are | pc1 | المخبر الأول | الهندسة المعلوماتية | def | null | 2015-03-16 | 31 |
| null | user | | | pc1 | المخبر الأول | الهندسة المعلوماتية | لا يعمل | null | 2011-05-14 | 29 |

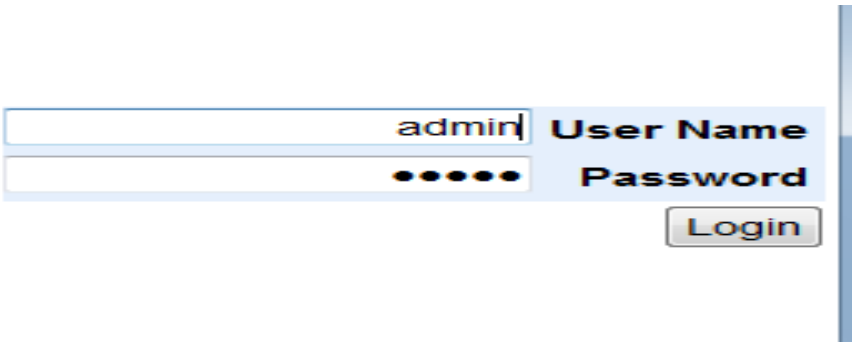
صورة 3.4 – جدول يبين اخر 5 اضافات الى قاعدة البيانات

في نهاية هذه الخطوات يكون المستخدم (user) قد أضاف مشكلة رام في الحاسب pc1 الموجود في كلية الهندسة المعلوماتية في المخبر الأول.

## المرحلة الثانية من الإختبار:

الدخول كمدير للنظام ومعالجة الخطأ المدخل من قبل المستخدم العادي.

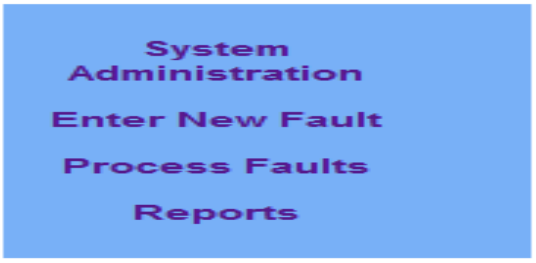
1. الدخول كمدير للنظام:




صورة 3.5 – تسجيل الدخول كمدير للنظام

## 3.2الواجهة الأساسية الخاصة بالمدير:

في هذه الواجهة نلاحظ صلاحيات المدير والتي تختلف عن صلاحيات المستخدم العادي حيث يمكن للمدير القيام بكافة عمليات الإدارة على الكليات والمخابر والحواسب الموجودة في قاعدة المعطيات لدينا من إضافة وتعديل وحذف، كما يمكن للمدير أيضا معالجة كافة الأخطاء المدخلة في قاعدة المعطيات وإدخال حل لها.

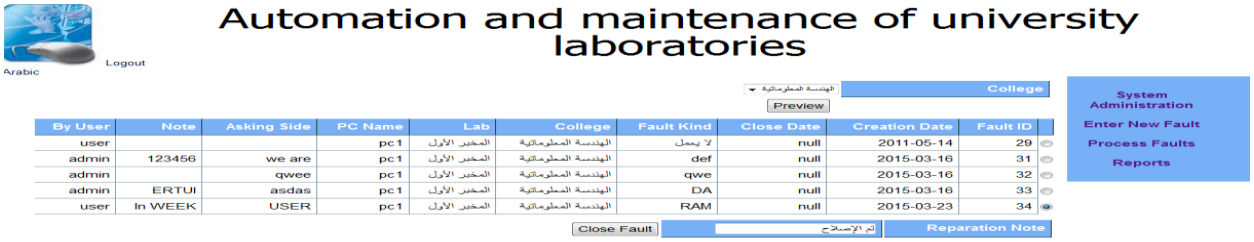



صورة 3.6 - الواجهة الاساسية الخاصة بالمدير

2. نقوم بالدخول إلى الواجهة الخاصة بمعالجة الأخطاء المدخلة:

صورة 3.7 - واجهة عرض التقارير

3. نقوم بتحديد الكلية لعرض الأخطاء ضمنها، وفي حالتنا الخطأ المدخل ضمن كلية المعلوماتية في المخبر الأول.




صورة 3.7.1 - واجهة عرض التقارير

نقوم بتحديد الخطأ الذي نريد وضع معلومات الإصلاح الخاصة به، ومن ثم نقوم بإضافة معلومة الإصلاح المرادة (ملاحظات الإصلاح).

في نهاية هذه المرحلة نقوم بطلب الأمر إغلاق الخطأ، ليقوم النظام بإضافة الخطأ إلى قاعدة المعطيات كخطأ مغلق.

# الفصل الرابع

## 4-الخاتمة:

قمنا في هذا المشروع بتصميم تطبيق كامل لإدارة كليات الجامعة، حيث يمكننا التطبيق المنشأ من القيام بكافة عمليات الإدارة على الكليات الموجودة في النظام(كلية المعلوماتية، كلية الهندسة البترولية وغيرها من الكليات) وهذه العمليات تشمل عمليات الإضافة والتعديل والحذف ومن ثم يمكننا التطبيق من إدارة عمليات الإضافة والحذف على المخابر الموجودة في الكليات وضمن كل مخبر من المخابر السابقة يوجد مجموعة من الحواسب المضافة وكذلك يؤمن التطبيق عمليات الإدارة على الحواسب الموجودة (إضافة ، تعديل وحذف).

في المرحلة التالية من العمل قمنا بتوزيع صلاحيات التطبيق بين مستخدم عادي ومدير للنظام حيث يؤمن التطبيق للمستخدم العادي إمكانية إضافة خطأ إلى قاعدة المعطيات حيث يقوم المستخدم بتحديد بارامترات الخطأ من نوع الخطأ (المشكلة) والجهة الطالبة للمشكلة وغيرها من المعلومات الأخرى المتعلقة بالمشكلة، وكما يؤمن التطبيق إمكانية البحث عن الأخطاء أو المشاكل المدخلة في قاعدة المعطيات لدينا وطباعتها.

يؤمن التطبيق بالنسبة لمدير النظام إضافة إلى الصلاحيات السابقة عمليات الإدارة على الكليات والمخابر والحواسب الموجودة في قاعدة المعطيات (إضافة، تعديل وحذف).

حيث يمكننا مستقبلاً تطوير هذا المشروع ليمكننا من القيام بعمليات شراء داخله الكترونياً من مواقع نربطه بها حيث تقدم لنا هذه المواقع عروض على القطع والبرمجيات لاتي تكون مرفوعة بالتقارير التي لدينا وطلب منها توصيلها وذلك سيعجل من العملية

## 5ملحقات:

### 5.1 ملحق1:

كود الاتصال مع قاعدة المعطيات:

```
Class.forName("com.mysql.jdbc.Driver");
String URL = "jdbc:mysql://localhost:3306/fms";

//Connection And The State Of Connection
Connection conn = DriverManager.getConnection(URL,"root","1234");
   Statement st = conn.createStatement();

//Query For Database
   String Query = "SELECT id, admin FROM users where
user_name='"+request.getParameter("user_name")+"' and
password='"+
   request.getParameter("pass")+"'";
  //Get The Results
   ResultSet rec = st.executeQuery(Query);

//Navigate The Result
      while(rec.next())
   {

       LoginIfnoMsg="User:  <b>
"+request.getParameter("user_name")+"</b><br>";
      session.setAttribute("logged","1");
      session.setAttribute("IsAdmin", new
String(rec.getInt(2)+""));
      session.setAttribute("user_id",new
String(rec.getInt(1)+""));
   }
```

الكود الخاص بالاتصال مع قاعدة البيانات

الكود السابق يمثل الإتصال مع فاعدة المعطيات ضمن لغة Java حيث تتم العملية كما يلي:

1. نقوم بتحميل الدرايفر الخاصة بالتعامل مع قاعدة المعطيات في لغة البرمجة java.

2. نقوم بتحديد مسار قاعدة المعطيات التي سنتعامل معها.

3. نقوم بفتح إتصال مع قاعدة المعطيات السابقة.

4. نقوم بإنشاء الإجرائية التي ستتعامل مع قاعدة المعطيات، والتي ستقوم بجلب النتائج التي نحتاجها في موقعنا الموجود، وهذه الإجرائية هي عبارة عن متحول نصي يمثل نص الإجرائية المقصودة.

5. نقوم بتنفيذ الإجرائية على قاعدة المعطيات لدينا.

6. نقوم بتخزين النتائج ضمن متحول من النمط ResultSet.

7. نقوم بالمرور على النتائج الموجودة، ونقوم بإجراء المطلوب منها (عرضها في جدول، التحقق من غسم المستخدم وكلمة المرور، وغيرها من العمليات الأخرى).

### 5.1.2 ملحق 2:

شرح تصميم صفحة مع كود بداخلها وسنشرح صفحة Menu.jsp:

```
<%@ page language="java" import="java.sql.*"%>
<%
if(session.getAttribute("lang")==null)
{
if(request.getLocale().toString().equals("en_US"))
{
	//out.println("Your Browser Language is English!!!!");
%>
<%@include file="fmsResources.jsp" %>
<%}else
{
	//out.println("Your Browser Language is Arabic!!!!");
%>
<%@include file="fmsResources_ar.jsp" %>
<%
}
}
%>
<%@include file="header.jsp" %>
<link rel="stylesheet" type="text/css" href="./css/fms.css"/>
<%
session.setAttribute("logged","0");//invlidate session
String LoginIfnoMsg="";
```

```
if(request.getParameter("sent")!=null)
{
    Class.forName("com.mysql.jdbc.Driver");
    String URL = "jdbc:mysql://localhost:3306/fms";
    Connection conn = DriverManager.getConnection(URL,"root","1234");
   Statement st = conn.createStatement();
   String Query = "SELECT id, admin FROM users where user_name='"+request.getParameter("user_name")+"'
and password='"+
   request.getParameter("pass")+"'";
   ResultSet rec = st.executeQuery(Query);

     while(rec.next())
   {
            LoginIfnoMsg="User:  <b> "+request.getParameter("user_name")+"</b><br>";
     session.setAttribute("logged","1");
     session.setAttribute("IsAdmin", new String(rec.getInt(2)+""));
     session.setAttribute("user_id",new String(rec.getInt(1)+""));
   }

            if(session.getAttribute("logged")=="0")
            {
                    LoginIfnoMsg="<b class=\"ErrInfoTxt\">"+ session.getAttribute("errLogin")+"</b><br>";
            }
     rec.close();
     conn.close();
}
```

```
if(session.getAttribute("logged")!=null){
if(session.getAttribute("logged").toString()=="1"){ %>
<div align="left">
<a class="LogoutLink" href="logout.jsp" ><%=session.getAttribute("linkLogout")%></a>
</div>
<div align="left">
<a class="LogoutLink" href="changeLang.jsp" ><%=session.getAttribute("linkClang")%></a>
</div>
<%}}%>
<script language="JavaScript">
<!--
function calcHeight()
{
  //find the height of the internal page
  var the_height=
    document.getElementById('main').contentWindow.
      document.body.scrollHeight;

  //change the height of the iframe
  document.getElementById('main').height=
      the_height;
}
```

```
</script>
<body dir="rtl" bgcolor="#990000" background="images\mm_bg_red.gif">
<%
if(session.getAttribute("logged")!=null){
if(session.getAttribute("logged").toString()=="1"){ %>
<div class="Wrapper">
<div class="LeftMenu">
<ul class="LeftMenuList">
   <%
   if(session.getAttribute("IsAdmin").toString().compareTo("1") == 0){
   %>
<li> </li>
<li> </li>
<li class="LeftMenuLink"><a href="manageCollege.jsp"
target="main"><%=session.getAttribute("menuSystemAdmin")%></a></li>
<% } %>
<li> </li>
<li> </li>
<li class="LeftMenuLink"><a href="fault.jsp"
target="main"><%=session.getAttribute("menuNewFault")%></a></li>
  <%
   if(session.getAttribute("IsAdmin").toString().compareTo("1") == 0){
  %>
<li> </li>
<li> </li>
<li class="LeftMenuLink"><a href="processFault.jsp"
target="main"><%=session.getAttribute("menuProcessFault")%></a></li>
```

```
<% } %>

<li> </li>

<li> </li>

<li class="LeftMenuLink"><a href="fault.jsp"
target="main"><%=session.getAttribute("menuNewFault")%></a></li>

  <%

    if(session.getAttribute("IsAdmin").toString().compareTo("1") == 0){

  %>

<li> </li>

<li> </li>

<li class="LeftMenuLink"><a href="processFault.jsp"
target="main"><%=session.getAttribute("menuProcessFault")%></a></li>

<% } %>

<li> </li>

<li> </li>

<li class="LeftMenuLink"><a href="reports.jsp"
target="main"><%=session.getAttribute("menuReports")%></a></li>

<li> </li>

<li> </li>

</ul>

</div>
```

### 5.1.3 شرح كود الصفحة:

1. الجزء الأول من الكود يمثل الإتصال مع قاعدة المعطيات كما تم شرحه في الملحق الأول.
2. كل مابين قوسين <% %> يمثل كود بلغة جافا، يتم عن طريقة التعامل مع المتحولات المخزنة ضمن session وغيرها من العمليات الأخرى المتعلقة بهذه اللغة. بحيث أنه لايتم عرض مابعدها بدون تحقيق شروط المحققة لهذا العرض مثال:
نلاحظ أنه ضمن الشرط

```
if(session.getAttribute("IsAdmin").toString().compareTo("1") == 0){
```

يتم عرض مجموعة من الروابط الخاصة بصلاحيات الأدمن فقط وهذه الروابط هي الروابط المسؤلة عن إدارة الكليات والمخابر والحواسب وكذلك الرابط المسؤول عن إغلاق الأخطاء التي قام المستخدمون العاديون بإضافتها إلى النظام أو قاعدة المعطيات.
بالطريقة السابقة يتم الدمج بين لغتي جافا وhtml ضمن صفحات jsp. لتحقيق الشروط وعرض المطلوب من كل صفحة.

## المراجع والأدوات المستخدمة :

### - المراجع :

| اسم المرجع | المؤلف | |
|---|---|---|
| MySQL® 5.0 Certification Study Guide | By Paul DuBois, Stefan Hinz, Carsten Pedersen<br>ISBN: 0672328127 Publisher: MySQL Press<br>Prepared for Francisco Leon Nieto,<br>Safari ID: Francisco.Leon@Sun.COM<br>Licensed by Francisco Leon Nieto<br>Print Publication Date: 2005/08/24 | **Web site (http://dev.mysql.com).** |
| JAVASCRIPT LERNNING | By :<br>Eng . SAMI AL-RABA'AH | |
| (Java Server Pages (JSP Basics | version 1 created by :2002/24/12 Sameer Tyagi<br>version 2 revised by Sang :2003/12/01 Shin with speaker notes<br>version 3 revised (Sang :2003/13/05 (Shin | |
| | | |
| | | |