CERTIFICATION OF APPROVAL

HOSPITAL MANAGEMENT SYSTEM

By
Ismaeel Hindawi
Mohammad Alafandi

Degree Bachelor of Engineering in
[Department of communication and networking]

ABSTRACT

الغرض:

تستخدم المستشفيات حاليا نظام يدوي لإدارة وصيانة المعلومات الهامة .الأنظمة الحالية في المستشفيات تتطلب العديد من النماذج الورقية مع مخازن البيانات المنتشرة في جميع أنحاء البنية التحتية لإدارة المستشفى.-[1]

وكما نلاحظ أنه في كثير من الأحيان فإن المعلومات في النماذج الورقية تكون غير مكتملة أو لا تتبع معايير الإدارة .وغالبا ما تضيع في أشكال العبور بين الإدارات التي تتطلب عملية المراجعة الشاملة لضمان أن لا تضيع المعلومات الحيوية.فوجود نشخ متعددة لنفس المعلومات في المستشفى ربما يؤدي لتضارب البيانات في مخازن البيانات المختلفة.

وإن جزءا كبيرا من القدرة على تشغيل أي مستشفى ينطوي على اكتساب وإدارة واسترجاع كميات كبيرة من المعلومات في الوقت المناسب .وهذه المعلومات تشتمل على معلومات المريض الشخصية والتاريخ الطبي ومعلومات الموظفين ,جدولة الغرف والأجنحة ,جدولة غرف العمليات ومختلف المرافق الأخرى .

كل هذه المعلومات يجب أن تدار بطريقة حكيمة وفعال ويتم ذلك باستخدام أنظمة إدارة المشافي (HMS) مما يجعلها أكثر كفاءة وخالية من الأخطاء .

لمحة عامة عن المشروع:

تم تصميم نظام إدارة المستشفى (HMS) ليحل مكان النماذج الورقية في المشافي .وتم تحديد وظائف رئيسية ليتم التحكم والسيطرة عليها وهي :

معلومات المرضى , معلومات تتعلق بالكادر الطبي سواء كانوا أطباء أو ممرضين , معلومات تتعلق بالموظفين في المشفى , بيانات غرف العمليات وبيانات الأدوية المتاحة في المشفى ,والفواتير المرتبطة بالمرضى. طبعا كل ذلك يهدف إلى الحد من الوقت والموارد المطلوبة لتنفيذ مثل هذه المهام , لذلك فإن جزءا كبيرا من عملية تشغيل أي مستشفى بشكل صحيح يعتمد على اكتساب وإدارة واسترجاع كميات كبيرة من المعلومات خلال فترة زمنية قصيرة .

تصميم النظام

في هذا البرنامج قمنا بتطوير بعض النماذج , وسنسرد توصيفا مختصرا عن هذه النماذج كالآتي:

-الاستقبال:

وحدة استقبال تعالج استفسارات مختلفة حول قبول المريض وإعداد السرير وتنقلات المريض داخل المستشفى . يمكن للنظام أيضا التعامل مع تكاليف المترتبة على المرضى من تكاليف العمليات الجراحية والمراجعات والفحوص السريرية وتتضمن جداول أخرى ترتبط بالأطباء ومنها:

- جدول زيارات الطبيب.
- جول تعيين الأطباء المختصين.
- الاستفسار عن المريض .
- البحث عن التاريخ المرضي للمرضى.

-الادارة:

هذه الوحدة تعالج كافة التفاصيل الرئيسية لمتطلبات المستشفى مثل الطبيب الاستشاري والطبيب المختص ورسوم الاستشارات ورسوم الخدمات وتعالج أمور أخرى مثل :

- الموظفين المسؤولين عن تسجيل التفاصيل.
- الأطباء الاختصاصيين.
- الاطباء المعيدين .
- إحالة الأطباء.

-الصيدلة:

تتعامل هذه الوحدة مع جميع أنواع المواد الطبية .هذه الوحدة تساعد في الحفاظ على تاريخ الأدوية ومواد التخدير والمواد الاستهلاكية وتتعامل مع المردود المادي وتوليد فواتير الدفع كما أنه يساعد على تأمين احتياجات العيادات الداخلية والخارجية للمستشفى .

-المختبر:

هـذه الوحـدة تمكـن مـن الحفـاظ علـى طلبـات الاختبـار المطلوبـة مـن المـريض وتوليـد نتـائج الاختبـارات لمختلـف الخـدمات المتاحـة مثـل علـم الأمـراض السـريرية والأشـعة السـينية واختبـارات الموجـات فـوق الصـوتية .وهـذه الطلبـات يمكـن أن تكـون قادمـة مـن نقـاط مختلفـة فـي المستشفى بما في ذلك الأجنحة والفواتير وجمع العينات .

-التسجيل:

هـذه الوحـدة تسـاعد فـي تسـجيل المعلومـات المختلفـة حـول المرضـى والتعامـل مـع كـل المرضـى فـي العيـادات الداخليـة والخارجيـة للمستشـفى.ويتم إنشـاء رقـم تعريفـي وحيـد لكـل مـريض بعـد التسـجيل .وهـذا يسـاعد فـي إدارة علاقـات المرضـى داخـل المستشـفى ويحـافظ علـى التاريخ الطبي للمريض.

ACKNOWLEDGEMENTS

الشـــكر والتقـــدير : بـــدأنا بـــأكثر مـــن يـــد وقاســـينا أكثـــر مـــن هـــم وعانينـــا الكثيـــر مـــن الصـــعوبات وهـــانحن اليـــوم والحمـــد لله نطـــوي ســـهر الليـــالي وتعـــب الأيـــام وخلاصـــة مشـــوارنا بـــين دفتـــي هـــذا العمـــل المتواضـــع ففـــي فتـــرة تهيئـــة هـــذا المشـــروع , نحـــن تلقينـــا دعمـــا ثمينـــا مـــن الأصـــدقاء و الأســـاتذة ونخـــص بالـــذكر هنـــا الأســـتاذ الـــدكتور علـــي عواضـــة والأســـتاذ المهنـــدس حســـام بوفـــاعور ونهـــدي هـــذا العمـــل لكـــل الأصـــدقاء ممـــن ســـاندونا وأظهـــروا اهتمامـــا خـــلال فتـــرة تحضيرنا لهذا العمل .

وطبعـــا الشـــكر الأكبـــر لضـــياء الحيـــاة وأثمـــن مـــا فـــي الوجـــود (عائلاتنـــا) الآبـــاء والأمهـــات فهـــم مـــن ســـخروا حيـــاتهم كرمـــى لنـــا وأنـــاروا درب علمنـــا بحنـــانهم وعطفهـــم لنكـــون فـــي مقدمـــة الركب دائما فهم بلسم الوجود .....

متطلبات البرنامج

- تقنيات الويب المستخدمة: ASP.NET.
- اللغة البرمجية المستخدمة في البرنامج:C#.
- مخدم الويب: IIS.
- نظام التشغيل :Windows 8/7/vista.

TABLE OF CONTENT

الغرض..... 3
تصميم النظام 4
الشكر والتقدير..... 6
متطلبات البرنامج..... 7
جدول المحتويات..... 8
جدول الأشكال..... 10
الإختصارات..... 11
1. المقدمة 12
2. التحليل و التصميم..... 13
2.1 التحليل..... 13
2.2 التصميم..... 13
3. وحدات المشروع 15
4. التطبيق 18
4.1 لمحة عامة عن التطبيقات المستعملة 18
4.2 مكتبة الصف الخاصة بال NET FRAMEWORK..... 22
4.3 ACTIVE SERVER PAGE (asp.net)..... 25
5. تكنولوجيا التطبيق 26
5.1 لمحة تاريخية عن التكنولوجيات المتبعة في هذا العمل 28
5.2 تكامل النظام الحالي مع برامج المخدمات الأخرى..... 30
5.3 الوظائف 34
5.4 أنماط قواعد البيانات 38
6. لغة البرمجة C# 40
12.1 ميزات لغة البرمجة C#..... 40
12.2 نظام النمط المشترك 41
7. دراسة الإمكانية ومنهجية البرمجيات 42
8.الإختبار 49

9. التنفيذ العملي.....53
9.1 شاشات الإخراج 53
9.2 الشبكة 63
10. الخلاصة والأفكار المستقبلية.....67
المراجع 69

TABLE OF FIGURES

الشكل4.1 ...العلاقة بين وقت تشغيل اللغة العامة ومكتبة الصفوف 20
الشكل 5.1 .....قاعدة بيانات وحيدة الصف.....38
الشكل 5.2....قاعدة بيانات ثنائية الصف 39
الشكل 9.1....الواجهة الرئيسية 53
الشكل 9.2.....الواجهة الرئيسية(2).....54
الشكل 9.3.....واجهة الإدارة 54
الشكل 9.4.....واجهة تسجيل دخول الأطباء.....55
الشكل 9.5.....واجهة تسجيل طبيب جديد.....55
الشكل 9.6.....واجهة تسجيل دخول الموظفين.....56
الشكل 9.7.....واجهة تسجيل موظف جديد.....56
الشكل 9.8.....واجهة التذكير بكلمة السر.....57
الشكل 9.9.....واجهة المختبرات.....57
الشكل 9.10.....واجهة مسرح العمليات.....58
الشكل 9.11...واجهة الإختبارات الطبية (فحص الدم).....58
الشكل 9.12...واجهة الإختبارات الطبية(فحص البول).....59
الشكل 9.13...واجهة تسجيل المرضى.....59
الشكل 9.14...واجهة تسجيل المرضى الداخليين في المستشفى.....60
الشكل 9.15...واجهة تسجيل المرضى الخارجيين في المستشفى.....61
الشكل 9.16...واجهة بيانات تخريج المرضى من المستشفى.....61
الشكل 9.17...واجهة مخزن الأدوية (الصيدلية).....62
الشكل 9.18.....بيئة الطابق الأول من المستشفى 65
الشكل 9.19.....بيئة الطابق الثاني والثالث من المستشفى 66

GLOSSARY

HMS	HOSPITAL MANAGEMENT SYSTEM.
IIS	INTERNET INTERFACE SERVICE.
UML	UNIFIED MODOLING LANGUAGE.
E-R	ENTITY RELATIONSHIP DIAGRAM.
IPD	INPATIENT DEPARTMENT.
OPD	OUTPATIENT DEPARTMENT.
ID	IDENTIFICTION NUMBER.
GUI	GRAPHICAL USER INTERFACE.
XML	EXTENSIBLE MARKUP LANGUAGE.
MIME	MULTPURPOSE INTERNET MAIL EXTENSION.
HTML	HYPERTEXT MARKUP LANGUAGE.
CTS	COMMON TYPE SYSTEM.
JIT	JUST IN TIME.
SOAP	SIMPLE OBJECT ACCESS PROTOCOL.
WDSL	WEB DESCRIPTION SERVICES LANGUAGE.
ASP	ACTIVE SERVER PAGE.
SQL	STRUCT QUERY LANGUAGE.
TCP	TRANSMISSION CONTROL PROTOCOL.
IP	INTERNET PROTOCOL.

Chapter 1

المقدمة

واحدة من القضايا المهمة في المشافي هي الخدمات الصحية مثل تزويد المستشفيات بالمعدات والمواد المطلوبة وتأمين المساعدة والرعاية الصحية المناسبة واللازمة للناس , وتأمين أنظمة خاصة تعمل داخل المشافي وتعمل على توفير الوقت والجهد خلال عملية التواصل داخل أنحاء المستشفيات .لذلك تم تطوير أنظمة إدارة المشافي (HMS) وهذه الأنظمة يمكن تعريفها بأنها أنظمة ضخمة ,وهي نظم دعم متكاملة لمتطلبات المستشفيات , بما في ذلك تأمين معلومات شاملة عن المريض , العلاج السريري , والتعاملات المالية داخل المستشفى . المستشفيات هي مؤسسات معقدة وضخمة وأصبحت في أيامنا هذه أكثر اعتمادا على أنظمة إدارة المستشفى (HMS) للمساعدة في حالات التشخيص والإدارة والمعلومات من أجل الخدمات وتحسينها. [3]

لذلك ومما سبق نرى بأن نظام إدارة المستشفى هو مبادرة تطبق على نطاق المنشأة ككل والتي هي المستشفى وتهدف إلى تحسين الوصول إلى المعلومات المرتبطة بالمرضى من خلال المعلومات المخزنة داخل هذا النظام . وتهدف أيضا إلى تبسيط تدفق المعلومات المرتبطة بالمرضى وسهولة الوصول إليها من قبل الأطباء والممرضين وغيرهم من الأشخاص اللذين يقدمون الرعاية الصحية .

ومن الفوائد الأخرى لنظام إدارة المستشفى بأنه يتيح لكل مريض أن يمتلك سجل خاص به فيتم تخزين معلومات خاصة بالمريض داخله وتتضمن هذه المعلومات التاريخ المرضي للمريض ومعلومات شخصية تتعلق به وتواريخ إجراء العمليات والفحوص السريرية والنتائج القادمة من المختبر والوصفات الدوائية السابقة والحالية ومعلومات أخرى إذا وجدت .

Chapter 2

التحليل و التصميم

أولا: التحليل

1.نظـــــام القائمـــــة

أغلـــب المستشـــفيات كانـــت ولا زالـــت تســـتخدم نظـــام يـــدوي لإدارة وصـــيانة المعلومـــات الهامـــة , ويـــتم تخـــزين البيانـــات فـــي كامـــل أرجـــاء البنيـــة التحتيـــة للمستشـــفى . وفـــي كثيـــر مـــن الأحيـــان المعلومـــات فـــي النمـــاذج الورقيـــة تكـــون غيـــر مكتملـــة أو لا تتبـــع لمعـــايير الإدارة .وغالبـــا مـــا يضـــيع جـــزء مـــن المعلومـــات أثنـــاء عبورهـــا بـــين الإدارات .ووجـــود نســـخ متعـــددة مـــن نفـــس المعلومـــات فـــي المستشفى ربما يؤدي لتضارب في البيانات في مخازن البيانات المختلفة .

2.النظـــــام المقتـــــرح

تـــم تصـــميم نظـــام إدارة المستشـــفى ليـــتم وضـــعه كـــدليل عوضـــا عـــن القـــوائم الورقيـــة .وهـــذا النظـــام الجديـــد تـــم وضـــعه للســـيطرة والـــتحكم بالمعلومـــات التاليـــة : معلومـــات المـــريض , تـــوافر الغـــرف والمـــوظفين وغـــرف العمليـــات والجـــداول الزمنيـــة المرتبطـــة بهـــا والفـــواتير المرتبطـــة بالمرضـــى .

3.الهـــدف مـــن النظـــام

وإن جـــزءا كبيـــرا مـــن القـــدرة علـــى تشـــغيل أي مستشـــفى ينطـــوي علـــى اكتســـاب وإدارة واســـترجاع كميـــات كبيـــرة مـــن المعلومـــات فـــي الوقـــت المناســـب .وهـــذه المعلومـــات تشـــتمل علـــى معلومـــات المـــريض الشخصـــية والتـــاريخ الطبـــي ومعلومـــات المـــوظفين ,جدولـــة الغـــرف والأجنحـــة ,جدولـــة غـــرف العمليـــات ومختلـــف المرافـــق الأخـــرى . كـــل هـــذه المعلومـــات يجـــب أن تـــدار بطريقـــة حكيمـــة وفعالـــة ويـــتم ذلـــك باســـتخدام أنظمـــة إدارة المشـــافي (HMS) ممـــا يجعلهـــا أكثـــر كفـــاءة وخالية من الأخطاء .

ثانيا: التصميم

التصـميم هـو الخطـوة الأولـى فـي مرحلـة التطـوير لتقنيـة مـا وعمليـة تحديـد مبـادئ هـذه التقنيـة بتفصـيل كـاف للسـماح بتحقيـق الاهـداف المرجـوة مـن هـذا التطـوير[4] .ويـتم تحليـل متطلبـات البرمجيـات لمـرة واحـدة , كـذلك يـتم تحديـد تصـميم البرمجيـات ويشمـل ثلاثـة أنشـطة فنيـة وهي :

- التصميم والترميز .
- تنفيذ الاختبارات المطلوبة للتصميم .
- التحقق من البرمجـيات .

إن الأنشـطة المرتبطـة بالتصـميم ذات أهميـة رئيسـية فـي هـذه المرحلـة لأنـه فـي هـذا النشـاط يـتم اتخـاذ القـرارات التـي تـؤثر فـي نهايـة المطـاف علـى نجـاح تنفيـذ البرنـامج وسـهولة الصـيانة .وهـذه القـرارات ذات تـأثير نهـائي علـى اعتماديـة وصـيانة النظـام فـي أوقـات لاحقـة . وتصـميم البرمجيات هي عملية يتـم من خلالها ترجمة المتطلبات إلى تمثيـل من البرمجيات .

Chapter 3

وحدات المشروع

- الاستقبال .
- الإدارة .
- الصيدلية .
- المختبر .
- التسجيل .
- تخريج المريض .

1-اسم الوحدة : الإستقبال .

تعالج وحدة الاستقبال أمور مختلفة حول دخول المريض وتفاصيل الإطلاق , إحصاء النزلاء من المرضى في الأسرة ,وحركات المريض ضمن المستشفى. النظام يمكن أن يعالج أيضا صفقات التكلفة الثابتة الشاملة أيضا للمرضى .بالإضافة إلى طبيب الاستشارة وأجور اطباء الاستشارة [5].

وحدات جزئية :

- جدول زيارات الطبيب .
- جدول مواعيد الطبيب .
- تحقيق المريض .
- وضع تاريخ عن تحقيقات المريض .

2 - اسم الوحدة : الإدارة .

تعالج هذه الوحدة كل تفاصيل الدخول الرئيسي لمتطلبات المستشفى مثل تفاصيل الاستشارة , تخصص الطبيب , أجور الاستشارة , ورسوم الخدمة .

وحدات جزئية :

- تسجيل تفاصيل الموظف .
- نوع الطبيب .
- الطبيب المختص .
- أطباء الإحالة .

3 - اسم الوحدة : الصيدلية .

هذه الوحدة تتعامل مع كل المواد الطبية .تساعد هذه الوحدة في إيصال المواد المخدرة لكافة أنحاء و أقسام المستشفى وتساعد أيضا في معالجة وحصر العائدات المادية وتولد فواتير البيع بالمفرد .وتساعد أيضا في تأمين متطلبات صيدلية ال IPD وال OPD أي الصيدليات التي تؤمن الدواء للعيادات الداخلية والخارجية في المستشفى .

4 - اسم الوحدة : المختبر .

تمكننا هذه الوحدة من صيانة طلبات التحقق المأخوذة من المريض , بالإضافة لتوليد نتائج الاختبارات للخدمات المختلفة المتوفرة , مثل علم الأمراض السريري ,الأشعة السينية والاختبارات فوق السمعية .الطلبات يمكن أن يتم تقديمها من نقاط مختلفة في المستشفى ويتضمن ذلك : الردهات ,المحاسبة ,مجموعة العينية . إن وحدة المختبر هي عبارة عن وحدة متكاملة مع المرضى داخل المستشفى والمرضى خارج المستشفى بالإضافة لوحدات المحاسبة .

5 - اسم الوحدة : التسجيل .

تساعد هذه الوحدة في تسجيل المعلومات حول المرضى وهناك هوية فريدة او رقم ID وحيد لكل مريض على حدى ويتم توليد هذا الرقم مباشرة بعد التسجيل . وهذا يتضمن الإبقاء على سجل طبي للمريض لتسجيل بياناته .

6 – تخريج المريض.

تساعد هذه الوحدة في توليد خلاصة تتضمن تخريج المريض من المستشفى بعد امتثاله للشفاء والتي تتضمن أيضا صحة المريض في وقت خروجه من المستشفى والتاريخ الطبي وتشخيصات مختلفة بالإضافة لوصفات المهدئات والمخدرات .

Chapter 4

التطبيق

أولا: نظرة عامة عن التقنيات المستعملة

تقنية الواجهة الأمامية .(Microsoft .NET Framework)

إن هذه التقنية هي رصيف حسابات جديد الذي يبسط تطوير التطبيقات في البيئة الموزعة للإنترنت . إن ال NET Framework. يتم تصميمه لإنجاز المهام التالية :

- لتزويد بيئة برمجية موجهة سواء كان الكود الغرضي مخزن ومنفذ محليا , أو منفذ محليا وموزع على الانترنت , أو ينفذ عن بعد .
- لتزويد بيئة تنفيذ الكود البرمجي التي تقلل قدر الإمكان من انتشار البرمجيات .
- لتزويد بيئة تنفيذ الكود البرمجي التي تتضمن تنفيذا آمنا للرمز .ويضمن ذلك الكود خلق طرف ثالث مجهول أو نصف مؤتمن .
- لتزويد بيئة تنفيذ الكود البرمجي التي تزيل مشاكل الأداء التي كتبت أو ترجمت البيانات .
- لجعل تجربة المطور ثابتة عبر أنواع مختلفة جدا من التطبيقات ,مثل التطبيقات التي أساسها النوافذ والتطبيقات المطبقة على الانترنت .
- لبناء كل الاتصالات الممكنة بالاعتماد على أساسات صناعية لضمان أن يكون الكود البرمجي المعتمد على .NETFramework. بإمكاننا أن نجعله متكاملا باستخدام أي كود برمجي آخر .

NET Framework. يتكون من مكونان أساسيان :

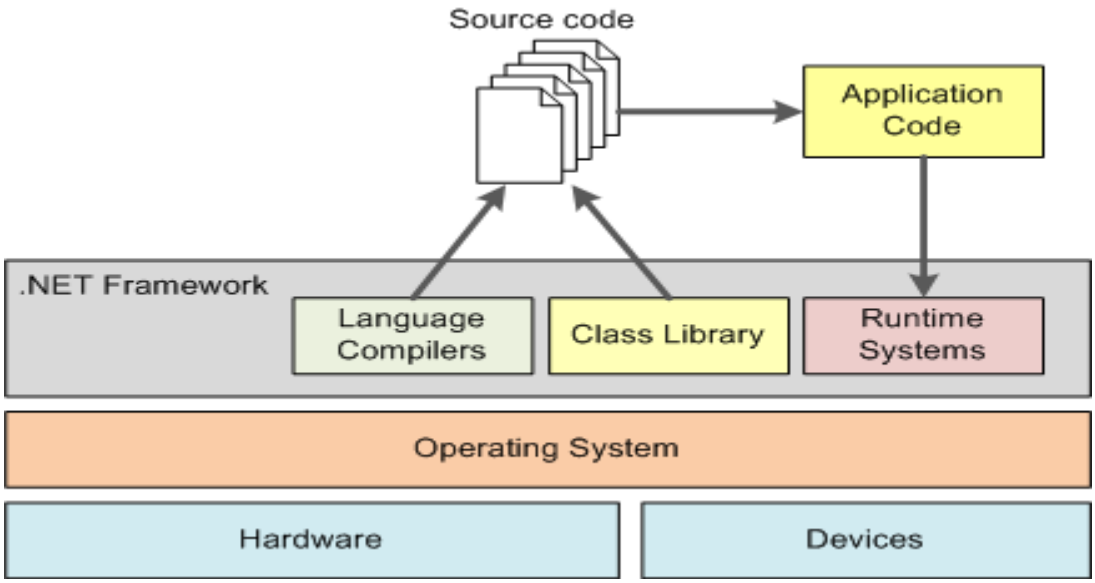
وقت تشغيل اللغة العامة ومكتبة الصف المتبعة في ال NETFramework. ووقت تشغيل اللغة المشتركة تعتبر أساسا لل NETFramework. كما ويمكنك التفكير في وقت التشغيل كأنه وكيل وهذا الوقت يديره كود برمجي معين أثناء عملية التنفيذ بالاضافة لتوفير الخدمات الأساسية مثل إدارة الذاكرة وإدارة موضوع معين والاتصال عن بعد ومن ناحية أخرى فإن وقت تشغيل لغة مشتركة قد فرض أيضا شروط لضمان السلامة الصارمة ونوع من الدقة على شكل أكواد أخرى وهذه الأكواد هي التي تضمن الأمن والمتانة .

في الواقع فإن مفهوم كود الإدارة هو عبارة عن مبدأ أساسي من مبادئ وقت التشغيل .ومن المعروف فإن التعليمات البرمجية التي تستهدف وقت التشغيل تكون تعليمات برمجية مدارة أما التعليمات التي لا تستهدف وقت التشغيل تعرف بإسم التعليمات البرمجية غير المدارة .

مكتبة الصفوف وهي المكون الرئيسي الآخر لل NETFramework. وهو يقوم بجمع وجوه شاملة من انواع قابلة لاعادة الاستخدام والتي يمكن استخدامها لتطوير تطبيقات تعتمد على تطبيقات واجهة المستخدم الرسومية (GUI) على أساس أحدث الابتكارات التي تقدمها بيئة ال ASP.NET مثل نماذج الويب وخدمات الويب XML. كما ويمكن استضافة ال NET Framework. بواسطة مكونات التعليمات البرمجية غير المدارة والتي يتم تحميلها في وقت تشغيل اللغة العامة والشروع في تنفيذ التعليمات البرمجبة المدارة .وبالتالي خلق بيئة برمجيات تمكننا من استغلال الميزات في التعليمات البرمجية المدارة وغير المدارة .

NETFramework. لا يوفر فقط عدة مضيفين في وقت التشغيل ولكنه ايضا يدعم تطوير المضيفين أثناء التشغيل فعلى سبيل المثال يقوم ASP.NET بتوفير بيئة عمل للتعليمات البرمجية المدارة من جانب المخدم .ويعمل مباشرة مع وقت التشغيل لتمكين التطبيقات المعتمدة على نماذج الويب وخدمات الويب XML. أما بالنسبة لمستعرض الانترنت (انترنت اكسبلورر) فهو مثال لتطبيق من التعليمات البرمجية غير المدارة والتي تعمل في وقت التشغيل كملحق من نوع MIME.واستخدام المستعرض لاستضافة وقت التشغيل تمكنك من تضمين المكونات المدارة أو عناصر تحكم WindowsForm في مستندات HTML.فإستضافة وقت التشغيل بهذه الطريقة يجعل التعليمات البرمجية المدارة ممكنة ولكن مع التحسينات الهامة للتعليمات البرمجية المدارة فقط فإنها يمكن ان تكون قابلة للاستخدام مثل تأمين تخزين الملفات المعزولة .

ويبين الرسم التوضيحي ادناه العلاقة ما بين وقت تشغيل اللغة العامة ومكتبة الصفوف للتطبيقات الخاصة بك مع النظام العام ,كما ويبين كيف تعمل التعليمات البرمجية المدارة داخل بنية أكبر .



الشكل 4.1 – العلاقة بين وقت التشغيل ومكتبة الصفوف

1- صفات اللغة المشتركة لوقت التشغيل :

وقت تشغيل اللغة العامة بإدارة الذاكرة وموضوع التنفيذ يتم بتنفيذ التعليمات البرمجية والتحقق من سلامة الرمز والتجميع ونظام الخدمات الأخرى.هذه الميزات الجوهرية للتعليمات البرمجية المدارة التي يتم تشغيلها في وقت تشغيل اللغة العامة .فيما يتعلق بالأمن فيتم منح المكونات المدارة بدرجات متفاوتة من الثقة وهذا يتوقف على عدد العوامل التي تشمل الأصل مثل الانترنت وشبكة المؤسسة والكمبيوتر المحلي أو وظائف أخرى حساسة حتى اذا كان يتم استخدامه في نفس التطبيق النشط.وفي الوقت الذي يفرض وصول رمز الحماية مثلا يمكنك الوصول لصفحة الويب الخاصة بالمستخدمين الذين هم على ثقة فيمكنك مثلا من اللعب بالرسوم المتحركة على الشاشة .ولكن لا يمكنك الوصول للبيانات الشخصية الخاصة بهم او الوصول لنظام الملفات .

ومن ناحية أخرى يفرض ايضا رمز متانة من خلال تنفيذ رمز التحقق من البنية التحتية اذا كانت صارمة وعادة يعرف باسم (CTS) وهذا الرمز يضمن بان تكون كافة المكونات البرمجية المدارة تصف خصوصية الصفحة .ولكن المجميعين في مايكروسوفت اختلفو في توليد تعليمات برمجية مدارة تتوافق مع ال (CTS) وهذا يعني ان التعليمات البرمجية المدارة يمكن أن تستهلك أنواع التعليمات المدارة الأخرى أثناء فرض رمز المتانة والأمن والسلامة .بالإضافة الى ذلك فإن أي بيئة مدارة من وقت التشغيل تقوم بإلغاء العديد من مزايا البرامج المشتركة فعلى سبيل المثال فان وقت التشغيل يعالج تلقائيا تخطيط الكائن ويدير مراجع الكائنات ويحررهم عندما ينتهي وقت استخدامهم .وهذه تدعى ادارة الذاكرة التلقائية وهي تحل اثنين من أخطاء التطبيق الاكثرشيوعا :تسرب الذاكرة واهدارها ومراجع الذاكرة غير الصالحة .

وقت التشغيل ايضا يسرع من انتاجية المطورين فعلى سبيل المثال :

يمكن للمبرمجين من كتابة التطبيقات في لغة التنمية التي يختارونها ولكن للاستفادة الكاملة من وقت التشغيل ومكتبة الصفوف والمكونات المكتوبة بلغات اخرى من قبل مطورين آخرين لذلك يؤخذ وقت التشغيل في عين الاعتبار .أما بالنسبة للمستقبل فقد تم تصميم وقت تشغيل للبر امج كما أنها تدعم البرامج الحالية والقديمة . والتوافق بين التعليمات البرمجية المدارة وغير المدارة يتيح للمطورين مواصلة استخدام مكونات كوم اللازمة .تم تصميم وقت التشغيل لتحسين الأداء على الرغم من أن وقت تشغيل اللغة المشتركة يوفر العديد من الخدمات إلا أنه لم يتم تفسير التعليمات البرمجية ابدا وهناك ميزة اخرى تسمى(JIT) تمكن تشغيل جميع التعليمات البرمجية المدارة بلغة الجهاز الاصلي من النظام الذي يتم تنفيذ التطبيق عليه .وفي الوقت نفسه فإن إدارة الذاكرة يزيل الامكانيات المتاحة للذاكرة المجزئة ويزيد من امكانيات الذاكرة المتاحة وهذا يزيد من مرجعية الأداء .

وأخيرا فإن وقت التشغيل يمكن تعيينه بأداء عالي وذلك بواسطة تطبيقات يتم تحديدها من قبل الملقم .مثل :مخدم ال MicrosoftSQLوخدمات معلومات الانترنت (IIS) وتوفر هذه البنية التحتية لك استخدام التعليمات البرمجية المدارة لكتابة المنطق الخاص بالعمل الذي تديره .

ثانيا: مكتبة الصف الخاصة بال .NET Framework

وهي عبارة عن مجموعة من أنواع قابلة لإعادة الاستخدام وتتكامل بشكل وثيق مع وقت التشغيل للغة العامة .وهذه المكتبة توفر انواع من التعليمات البرمجية المدارة التي وظيفتك الأساسية كمبرمج أن تحصل عليها .ولكن هذا لا يجعل فقط انواع NETFramework.سهلة الاستخدام بل ويقلل أيضا من الوقت اللازم لتعلم الميزات الجديدة لل NETFramework. بالإضافة إلى ذلك يمكن لمكونات الطرف الثالث ان تتكامل مع الصفوف الموجودة في .NETFramework فعلى سبيل المثال مجموعة الصفوف في ال .NETFramework تقوم بتنفيذ مجموعة من الواجهات التي يمكنك استخدامها لتطوير مجموعة الصفوف الخاصة بك والطبقات في مجموعتك ستندمج بسلاسة مع الصفوف في ال .NETFramework بالاضافة لذلك فان انواع .NETFramework تمكنك من انجاز مجموعة من المهام البرمجية المشتركة مثل : إدارة سلسلة , جمع البيانات , الربط مع قاعدة البيانات , والوصول الى الملفات . بالاضافة الى كل تلك المهام الشائعة سيظل وجود أنواع من الصفوف التي تدعم مجموعة من السيناريوهات للتنمية المتخصصة أمر وارد فعلى سبيل المثال يمكنك استخدام.NETFramework.لتطوير الأنواع التالية من التطبيقات والخدمات:

- تطبيقات وحدة التحكم .
- التطبيقات النصية .
- تطبيقات واجهة المستخدم الرسومية في ويندوز و تطبيقات ال ASP.NET.

1 - تطوير تطبيقات العميل :

تطبيقات العميل هي أقرب الى النمط التقليدي المتبع في تطبيقات البرمجة المستندة إلى ويندوز .وهذه هي التطبيقات التي تعرض الأشكال والأيقونات على سطح المكتب .وتشمل تطبيقات العميل التطبيقات مثل معالجات النصوص وجداول البيانات وكذلك تطبيقات الاعمال المتخصصة مثل ادوات ادخال البيانات وادوات اعداد التقارير .وتطبيقات العميل عادة تستخدم النوافذ والقوائم والازرار وعناصر واجهة المستخدم الرسومية الاخرى . نوع آخر من انواع تطبيقات العميل هو عنصر تحكم ال ActiveX التقليدية والمنتشرة عبر الانترنت لصفحات الويب وهذا التطبيق هو مثل الكثير من تطبيقات العميل الاخرى يتم تنفيذه ولديه الحق بالوصول إلى الموارد وشيتمل على العناصر الرسومية .

2 - تطوير تطبيقات المخدم :

يتم تنفيذ تطبيقات المخدم في العالم من خلال مضيفين وقت التشغيل والتطبيقات غير المدارة تضيف وقت تشغيل اللغة المشتركة .والذي يسمح لكود البرمجة الخاص بك من التحكم بسلوك المخدم .وهذا النموذج يوفر لك جميع الميزات من وقت التشغيل ومكتبة اللغة المشتركة بينما اكتساب اللأداء وتطويره يتم من خلال المخدم المضيف .

3 - الأكواد البرمجية المدارة من قبل المخدم :

ASP.NET هو بيئة تتيح للمطورين استخدام .NETFramework. لتنفيذ التطبيقات المستندة الى الويب ومع ذلك فإن ASP.NET هو أكثر من مجرد مضيف لوقت التشغيل بل هو بنية متكاملة لتطوير مواقع الويب وكل من نماذج الويب وخدمات الويب من نوع XML تستخدم ال IIS و ال ASP.NET كآلية لنشر التطبيقات وكلاهما عبارة عن مجموعة من الصفوف لدعم ال .NETFramework. وتوزيع خدمات الويب XML أحدثت تطورا هاما في مجال التكنولوجيا على شبكة الانترنت فمكونات التطبيقات من جانب المخدم مماثلة لتلك المتبعة في مواقع الويب المشتركة ولكن خلافا للتطبيقات المستندة الى الويب فمكونات خدمات الويب من نوع XML ليس لها واجهة للمستخدم وليست موجهة للمستعرضات مثل الانترنت اكسبلورر وبدلا من ذلك تتكون خدمات الويب من نوع XML من مكونات برامج قابلة لاعادة الاستخدام ومصممة ليتم استهلاكها من قبل التطبيقات الاخرى مثل تطبيقات العميل التقليدية ,التطبيقات المستندة الى الويب او حتى غيرها من خدمات XML وتكنولوجيا خدمات الويب تتحرك بسرعة لتطوير التطبيقات ونشرها في البيئة الموزعة للانترنت فمثلا اذا كنت من المبرمجين الذين استخدموا النسخة الاولية من ال ASP فإنك ستلاحظ التحسينات التي طرأت على النسخة المحدثة منها وهي ال ASP.NET فعلى سبيل المثال :يمكنك تطوير صفحات ويب بأي لغة تدعم .NETFramework. بالاضافة الى ذلك التعليمات البرمجية الخاصة بك لم تعد بحاجة لتشاركها في نفس الملف النصي الخاص بك من نوع HTTP (على الرغم من أنها يمكن ان تستمر بالقيام بذلك اذا كنت تفضل هذا الامر) وصفحة الويب تقوم بتنفيذ لغة الآلة الأم لأنه كالتطبيقات البرمجية المدارة تقوم بالاستفادة الكاملة من وقت التشغيل .

وفي المقابل فإن صفحات ال ASP يتم كتابتها دائما وتفسيرها .وكما هو ملاحظ من التجارب فإن الصفحات من نمط ال ASP.NET هي اسرع واكثر وظيفية والاسهل هو تطوير صفحات ال ASP غير المدارة لانها تتفاعل مع وقت التشغيل مثل اي تطبيق آخر تتم إدارته.

ويقوم .NETFramework. بتوفير مجموعة من الطبقات والادوات للمساعدة في تطوير تطبيقات خدمات الويب ويتم بناء هذه الخدمات على معايير مثل SOAP (بروتوكول إجراء المكالمات عن بعد) و ال XML (تنسيق البيانات الموسعة) وال WSDL(لغة وصف خدمات الويب). حتى .NETFramework. يتم بناؤه على هذه المعايير لتعزيز قابلية التشغيل مع الحلول غير التابعة ل Microsoft فعلى سبيل المثال :أداة وصف خدمات الويب المتضمنة .NETFramework.SDK تمكنك من الاستعلام عن خدمة الويب المنتشرة على الشبكة وتعطي تحليلا وصفيا لها وتنتج #C أو Visual شيفرة المصدر الاساسي للتطبيق الخاص بك والتي يمكنك استخدامها لتصبح عميلا لخدمة الويب .ويمكن للشيفرة ايضا إنشاء صفوف مشتقة من الصفوف التابعة لمكتبة الصفوف التي تتعامل مع جميع الاتصالات الاساسية باستخدام SOAP و XML . يقوم .NETFramework. بتوفير مجموعة من الصفوف التي تتوافق مع كل معايير الاتصالات الاساسية مثل SOAP,WDSL,XML وباستخدام تلك الصفوف يمكنك التركيز على منطق الخدمة الخاص بك دون أن تشغل نفسك مع موضوع البنية التحتية للاتصالات التي يتطلبها تطوير البرمجيات الموزعة .

ثالثا: ACTIVE SERVER PAGE (ASP.NET)

ASP.NET هو بيئة عمل برمجية مبنية على وقت تشغيل اللغة العامة التي يمكن استخدامها لبناء تطبيقات ويب قوية [7].ويقدم ASP. NET العديد من المزايا الهامة ومنها:

- تعزيز الأداء : يتم تصنيف ASP.NET على أنه رمز للغة مشتركة وهي قيد التشغيل على المخدم وتمكن هذه اللغة المبرمج من الاستفادة من الربط المبكر وطبعا هذه اللغة لا ترقى الى تقديم اداء افضل ما لم تمضي بكتابة سطر برمجي وحيد.
- أداة دعم من الطراز العالمي : ويتكامل ASP.NETframework عن طريق مجموعة ادوات غنية ومصممة في برنامج بيئة التطوير المتكاملة VisualStudio مثل التحرير وعناصر التحكم بمخدم السحب والافلات والنشر التلقائي وكلها تعتبر مجرد عدد قليل من السمات التي توفرها هذه الاداة .
- الطاقة والمرونة : ولان ال ASP.NET قائم على وقت تشغيل اللغة المشتركة فإن القوة والمرونة لمطوري تطبيقات الانترنت ستكون متاحة بالاضافة الى مكتبة NET Framework والتراسل وحلول الوصول الى البيانات فكلها يمكن الوصول اليها بسهولة من الويب . ASP.NET هو لغة مستقلة بحيث يمكنك من اختيار اللغة التي تنطبق وتلائم التطبيق الذي تعمل عليه أو يمكنك تقسيم العمل لديك الى عدة لغات.
- البساطة : ASP.NET تجعل من السهل تنفيذ المهام المشتركة وذلك يبدو جليا اثناء تقديم نموذج بسيط من قبل العميل الى اللحظة التي يتم فيها نشر التطبيق على الموقع او تكوين موقع ما فعلى سبيل المثال فإن إطار صفحة ASP.NET يسمح لك ببناء واجهات للمستخدم ستكون منفصلة وواضحة وذلك بالتعامل مع تعليمات برمجية للعرض والتعامل مع الاحداث بطريقة بسيطة .
- الادارة : وتقوم ASP.NET بتوظيف نظام هرمي قائم على النص للقيام بعملية الادارة ولانه يتم تخزين معلومات التكوين كنص عادي ويمكن تطبيق الاعدادت الجديدة دون مساعدة من ادوات الادارة المحلية .
- قابلية التطوير والتوفر : وقد تم تصميم ASP.NET مع ميزات مصممة خصيصا لتحسين الاداء في بيئات العمل المتفاوتة وفي حالات المعالجة المتعددة علاوة على ذلك فإنه يتم رصد العمليات عن كثب بحيث اذا اساء احد ما التصرف يمكنك انشاء عملية جديدة مكانها مما يساعد في الحفاظ على التطبيق الخاص بك متاحا باستمرار للتعامل مع الطلبات .

- التفصيل والتمدد : ASP.NET توفر بنية معينة تسمح للمطورين بربط التعليمات البرمجية الخاصة بهم على المستوى المناسب .
- الأمن : مع الاندماج في شروط وبندوز وتطبيق إعدادت التكوين لكل تطبيق فهذا يجعلك متأكدا من ان التطبيقات الخاصة بك أصبحت آمنة .

1 - دعم اللغة :

توفر مايكروسوفت ببيئة ال NET. القيام بعملية بناء اي تطبيق برمجي بدعم من ثلاث لغات برمجية وهي :C# ,Visual Basic , JavaScript .

ما هو نموذج الويب ASP.NET ؟

هو اطار عمل لانشاء الصفحات عن طريق تقليل وقت تشغيل اللغة العامة والتي يمكن استخدامها من قبل المخدم لانشاء صفحات ويب بطريقة أكثر حيوية ,ويقصد به ان يكون التطور المنطقي لل ASP.NET يقوم بتوفير التوافق مع تركيب الصفحات الموجودة وقد تم تصميم اطار صفحة ال ASP.NET خصيصا لمعالجة عدد من الاوجه الرئيسية للصفحات .ويمكننا تلخيص عمل ASP.NET بالشكل التالي :

- القدرة على إنشاء واستخدام عناصر التحكم في واجهة المستخدم والقابلة لاعادة الاستخدام وبالتالي تقليل كمية التعليمات البرمجية التي يحتاج مطور الصفحات ان يكتبها .

- قدرة مصممي ومطوري الصفحات على هيكلة صفحاتهم بشكل منظم وجميل .

- القدرة على تطوير الادوات لتوفير الدعم وزيادته لمصممي الصفحات وجعل التصميم اكثر قوة من الناحية العملية.

صفحات نماذج الويب ASP.NET هي عبارة عن ملفات نصية مع لاحقة تكون عادة من نوع aspx. وعندما يطلب المستخدم الموارد من لاحقة aspx. ففي وقت تشغيل ASP.NET يقوم بتوزيع صفوف من مكتبة ال NETFramework. عندها يمكن استخدام هذه الصفوف لمعالجة الطلبات الواردة بشكل حيوي .

وصفحة ASP.NET يمكن ان تنشأ ببساطة عن طريق اخذ ملف من نوع HTML وتغيير لاحقة الملف الى aspx.

الرمز الخفي في نماذج الويب : يدعم ASP.NET طريقتين لإنشاء صفحات ديناميكية .الاول هو الأسلوب حيث أقوم بكتابة كود الصفحة داخل ملف ال aspx الاصلي .أما الطريقة الثانية تعرف باسم طريقة التعليمات البرمجية الخلفية وذلك بفصل الكود البرمجي بصفحة خاصة به .

مقدمة لمخدم التحكم بال ASP.NET : بالاضافة لاستخدام كتل التعليمات البرمجية لبرمجة المحتوى الديناميكي يمكن للمطورين استخدام صفحة ASP.NET للتحكم ببرمجة صفحات الويب .ويتم كتابة مخدم التحكم ضمن ملف aspx يحتوي العلامات الجوهرية من نمط HTML والتي تحتوي على RUNAT المخدم.

تقارير الكريستال: تعتبر تقارير الكريستال ل VisualBasic.NET هي الاداة القياسية للإبلاغ عن ال Visual.NET كما وأن لها القدرة على التفاعلية وذات جودة عالية في عرض المحتوى ولتقديم البيانات الى المستخدمين يمكنك كتابة التعليمات البرمجية لتنفيذ حلقات عبر مجموعات السجلات وطباعتها داخل الويندوز الخاص بك او ضمن صفحة الويب ومع ذلك فإن أي عمل يتجاوز التنسيقات الاساسية يمكن ان يكون معقدا .ومع تقارير الكريستال لبرنامج ال VisualBasic.NET يمكنك إنشاء تقارير معقدة ذات مظهر احترافي وبدلا من الترميز يمكنك استخدام واجهة مصممة بتقارير الكريستال لخلق وتهيئة التقرير التي تحتاج إليه .

Chapter 5

تكنولوجيا التطبيق

أولا: لمحة تاريخية عن التكنولوجيات المتبعة في هذا العمل

بداية سنتكلم بإيجاز عن MicrosoftSQLServer والذي هو لغة استعلام هيكلية مبنية على اساس قاعدة بيانات تتكون من عميل ومخدم وكل هذه المصطلحات تصف جزءا اساسيا من بنية ال SQL Server .

قاعدة البيانات : وهي مشابهة للملف المتعلق بالبيانات لانها مكان لتخزين البيانات وكما هو الحال في ملف البيانات فإن قاعدة البيانات لا تقدم المعلومات مباشرة الى المستخدم فيتوجب على المستخدم القيام بتطبيق للوصول الى البيانات ضمن قاعدة البيانات ويقدمها بعد ذلك للمستخدم بشكل مفهوم .وقاعدة البيانات تتألف عادة من مكونين : ملفات قاعدة البيانات الفعلية ونظام إدارة قاعدة البيانات والبرمجيات التي تستخدم في إنشاء تطبيق معين للوصول لقاعدة البيانات .بما في ذلك :

- الحفاظ على العلاقات والربط بين البيانات وقاعدة البيانات .
- التأكد من البيانات يتم تخزينها بشكل صحيح .
- استعادة كافة البيانات الى نقطة معروفة ويتم تحديدها في حال فشل النظام .

قاعدة البيانات المرتبطة : هناك طرق مختلفة لتنظيم البيانات في قاعدة البيانات ولكن قواعد البيانات المرتبطة هي الأكثر فعالية على الإطلاق .ونظم قواعد ابيانات المرتبطة هي تطبيق لنظرية المجموعات الرياضية على مشاكل تنظيم البيانات ويتم ذلك على نحو فعال وفي قاعدة البيانات المرتبطة يتم جمع البيانات في الجداول وعند تنظيم البيانات في الجداول يمكنك ان تجد عادة العديد من الطرق المختلفة لتحديد الجداول وتعرف نظرية قواعد البيانات المرتبطة عملية التطبيع والتي تضمن لك ان مجموعة الجداول التي تحددها وتنظيم البيانات داخلها سيتم على نحو فعال .

العميل والمخدم : في نظام العميل والمخدم ,المخدم هو جهاز كمبيوتر كبير نسبيا في موقع مركزي يتم ادارته من قبل عدد كبير من الناس وعندما يحتاج الافراد لاستخدام الموارد فإنهم ينشؤون اتصالا من اجهزتهم وباستخدام شبكة الانترنت يتم ايصالهم الى المخدم الرئيسي ومن الأمثلة على المخدمات يمكن ان نرى بإنه بأي بنية قاعدة بيانات عميل ومخدم توجد هناك ملفات لقاعدة البيانات ونظم ادارة لهذه القواعد والبرمجيات وكلها ستكون موجودة على المخدم .ويتم توفير عنصر الاتصالات بحيث يمكن للعملاء تشغيل التطبيقات بشكل منفصل والتواصل مع المخدم من خلال شبكة اتصال معينة .

ويسمح عنصر الاتصال SQLServer للتواصل بين تطبيق قيد التشغيل على المخدم و SQLServer. وتطبيقات المخدم عادة ما تكون قادرة على العمل مع العديد من العملاء في نفس الوقت . SQLServer يعطي القدرة لجعل آلاف من تطبيقات العملاء تعمل في نفس الوقت والمخدم لديه المزايا لمنع المشاكل المنطقية التي تحدث في حالة حاول المستخدم قراءة او تعديل البيانات المستخدمة حاليا من قبل الآخرين .في حين تم تصميم SQL Server للعمل كمخدم لشبكة العميل والمخدم كما أنها قادرة على العمل كقاعدة بيانات قائمة بذاتها مباشرة للعمل مع العميل .وإن سهولة استخدام ميزات SQLServer تجعلها قابلة للعمل بكفاءة في حال طلب العميل ذلك وبدون استهلاك الكثير من الموارد.

لغة الاستعلام الهيكلية : للعمل مع البيانات في قاعدة البيانات يجب استخدام مجموعة من الأوامر والبيانات التي يحددها برنامج نظم إدارة قواعد البيانات .وهناك العديد من اللغات المختلفة التي يمكن استخدامها مع قواعد البيانات العلائقية والأكثر شيوعا هي SQL .

ميزات SQL Server :

يعتمد Microsoft SQL Server على مجموعة من المزايا التي تحقق الفوائد التالية :

- سهولة التركيب والنشر والاستخدام : يتضمن SQLServer مجموعة من الأدوات الإدارية والتنموية التي تعمل على تحسين قدرتك على تثبيت ونشر وإدارة واستخدام SQLServer عبر عدة مواقع .

- التدرجية (القياسية) : محرك قواعد البيانات نفسها يمكن استخدامها عبر منصات تتراوح بين اجهزة الكمبيوتر المحمول بنظام التشغيل Windows-vista/7/8 والمخدمات الكبيرة والمتعددة المعالجات بأنظمة تشغيل أقدم .

- تخزين البيانات : يتضمن SQLServer أدوات لاستخراج وتحليل البيانات الموجزة عن المعالجة التحليلة التي تقام عبر الانترنت ويتضمن أيضا أدوات لتصميم قواعد البيانات المرئية وتحليل البيانات باستخدام اسئلة معينة .

ثانيا: تكامل النظام الحالي مع برامج المخدمات الأخرى

قواعد البيانات : تتكون في MicrosoftSQLServer من مجموعة من الجداول التي تحتوي على البيانات وغيرها من الأشياء مثل طرق عرض الفهارس وإجراءات التخزين ويعرف لدعم أنشطة التنفيذ مع البيانات وعادة ما ترتبط البيانات المخزنة في قاعدة البيانات بموضوع معين أو عملية مثل معلومات الجرد لمستودع ما .

ويقوم SQLServer بتقديم دعم للعديد من قواعد البيانات وكل قاعدة بيانات يمكنك تخزين البيانات ضمنها إما بشكل مترابط أو حتى البيانات المستقلة التي ليس لها علاقة بقواعد البيانات الأخرى . فعلى سبيل المثال يمكن أن يكون لمخدم ما قاعدة بيانات أولى لتخزين بيانات الموظفين وأخرى تقوم بتخزين البيانات ذات الصلة بالمنتجات وبدلا من ذلك يمكنك تخزين كافة البيانات ضمن قاعدة بيانات واحدة . فمن المهم أن نفهم أجزاء قاعدة البيانات وكيفية تصميم هذه الاجزاء لضمان ان قاعدة البيانات تؤدي عملها بشكل جيد وخاصة بعد عملية التنفيذ .

نظرية التطبيع :

العلاقات في العادة هي عملية تطبيع لتجنب الحالات الشاذة أثناء إدراج وتحديث وحذف العمليات .وقد تم بناء نظرية حول مفهوم التطبيع من الاشكال الطبيعية ويقال إن العلاقة ستكون بشكل معين اذا استوفت مجموعة محددة من القيود .

التكنولوجيا الوسيطة :

ADO.NET: وهو نموذج الوصول للبيانات التي تعالج مباشرة احتياجات المستخدمين لتطوير تطبيقات قابلة للتطوير وتم تصميمه خصيصا لشبكة الانترنت في حالات التدرج .ويستخدم أيضا بعض الأغراض مثل الاتصال وأغراض الأوامر ويشتمل مفتاح ال ADO.NET على تقديم مجموعة جديدة من البيانات وقارئ لهذه البيانات ومحول لهذه البيانات. ومحول البيانات في التعريف هو الغرض الذي يربط قاعدة البيانات لملء البيانات داخلها فهو يقوم بالاتصال بقاعدة البيانات بكل مرة يقوم المستخدم بإدخال البيانات لتحديث البيانات داخلها .

في الماضي كانت معالجة البيانات تقوم في المقام الأول على أساس الاتصال أما الآن وفي محاولة لجعل التطبيقات المتعددة المستويات اكثر كفاءة فتحولت معالجة البيانات الى النهج القائم على الرسالة التي يدور حولها جزء من المعلومات . والعامل الأساسي لجعل

هـذا الـنهج يعمـل هـو وجـود محـول البيانـات والـذي يـوفر جسـرا لاسـترداد وحفـظ البيانـات وتخزينها في المصدر .

وننتقـل الآن الـى الغـرض Dataset الـذي يسـتند الـى نمـوذج ال XML ويـوفر برمجـة منسـقة تعمـل مـع جميـع نمـاذج تخـزين البيانـات وذلـك فـي حـال عـدم تـوفر معرفـة مسـبقة عـن مصـدر البيانـات الخاصـة بـه والتـي تمثـل البيانـات التـي تحـتفظ بهـا علـى شـكل معلومـات .وبغـض النظـر عـن مصـدر البيانـات ضـمن مجموعـة البيانـات فإنـه يـتم التلاعـب مـن خـلال مجموعـة مـن واجهـات التطبيقـات البرمجيـة ,فـي حـين ان مجموعـة البيانـات المخزنـة فـي قاعـدة البيانـات لا يتـوفر لـديها معرفـة مسـبقة عـن مصـدر البيانـات المخزنـة داخلهـا .وقـد تـم فصـلها بمعلومـات محـددة .وتسـتمر مجموعـة البيانـات بالانتقـال مـن وإلـى مخـازن البيانـات ضـمن System.Data.Sqlclient والتـي هـي جـزء مـن ال Framework وتقـوم بتـوفير أربعـة أشياء أساسية وهي : القيادة , الاتصال , قارئ البيانات ومحول البيانات .

الإتصـالات والـروابط : هـذه الـروابط تسـتخدم للتواصـل والتخاطـب مـع قواعـد ويـتم تمثيلهـا عـن طريـق مـزود مثـل SQLConnection ويـتم إرجـاع الأوامـر عبـر الاتصـال وتنـتج عنهـا مجموعـات بشـكل تيـارات ولقـراءة هـذه المجموعـات يتوجـب علينـا اسـتخدام قـارئ البيانـات .

الأوامـر : وهـي أوامـر تتضـمن المعلومـات التـي يـتم إرسـالها الـى قاعـدة البيانـات وهـذه البيانـات ويـتم تمثيلهـا عـن طريـق مـزود خـاص مثـل SQLCommnd ويمكـن أن يكـون إجـراء التخـزين هـو عبـارة عـن عمليـة تحـديث وبعـدها يـتم إرجـاع النتـائج كمـا ويمكنـك ايضـا مـن اسـتخدام مـدخلات و مخرجـات المعلومـات والقـيم التـي يـتم ارجاعهـا هـي جـزء مـن بنـاء جملـة الاوامر الخاصـة بـك .

قارئ البيانات : يتم استدعاء غرض قارئ البيانات في قاعدة البيانات بعد تنفيذ أمر ما ضمن قاعدة البيانات وقارئ البيانات عادة يكون مختلفا بين المجموعات القياسية فمثلا يمكنك استخدام قارئ البيانات لإظهار نتائج البحث في قائمة صفحة ويب .

مجموعات البيانات : إن الغرض Dataset أو مجموعة البيانات هو مماثل لغرض مجموعة السجلات ADO ولكنه أكثر قوة ,كما ويوفر عدة ميزات أخرى ومنها :

إن غرض مجموعة البيانات دائما غير متصل , يعتبر هذا الغرض كمخبأ توضع بداخله البيانات ويتعامل مع معظم هياكل قواعد البيانات مثل الجداول والاعمدة والعلاقات والقيود ومع ذلك فإن مجموعة البيانات لا يمكننا اعتبارها كالكثير من قواعد البيانات . ومن المهم الذكر هنا أن أغراض Dataset لا تتفاعل مباشرة مع قواعد البيانات او اي مصدر بيانات آخر وهذا يسمح للمطور أن يعمل مع نموذج البرمجة المحدد مسبقا وبغض النظر عن المكان الذي توجد فيه البيانات فإنه يمكن وضع جميع البيانات القادمة من قاعدة البيانات ضمن ملف من نوع XML أو إدخال المستخدم الى أغراض البيانات .

ويمكننا إجراء تغييرات على مجموعة البيانات التي يمكن تعقبها والتحقق من صحتها قبل تحديث بيانات المصدر ويمكن للمطور الحصول على هذه التغييرات من غرض Dataset والذي يخلق في الواقع بواسطة فهرس والذي يحتوي ضمنيا على التغييرات المطلوبة وبعد ذلك يتم استخدام هذا الغرض بواسطة محول البيانات لتحديث مصدر البيانات الاصلي .وال Dataset لديها الكثير من خصائص ال XML بما في ذلك القدرة على إنتاج واستعمال البيانات من نوع XML ومخططات ال XML وهذه المخططات يمكن استخدامها لوصف المخططات المتبادلة عبر خدمات الويب المتنوعة .

محولات البيانات :

محول البيانات يعمل كجسر بين مجموعة البيانات وبيانات المصدر ويمكن استخدام sqlDataAdapter كموفر إلى جانبSqlCommand المرتبط به و SqlConnection ولزيادة الأداء العام عند العمل مع بيانات Microsoft SQL Server ولقواعد البيانات الاخرى بدعم DB_OLE يمكن استخدام الغرض OleDbDataAdapter وما يرتبط بها من كائن OleDbCommand و OleDbConnection ويستخدم محول البيانات لتحديث مصدر البيانات بعد ان تم إجراء تغييرات على ورقة العمل .باستخدام أسلوب التعبئة من محول البيانات Select واستخدام الأسلوب update نقوم بعملية تحديث بالاضافة الى الأسلوب delete , insert وهذه الاوامر كلها لتغيير البيانات في الصفوف او التعديل بها .وبما في ذلك استخدام الاجراءات المخزنة لسيناريوهات مخصصة ويمكن لكائن معين تم انشاؤه مسبقا ان يقوم بتوليد هذه الأوامر في وقت التشغيل على أساس بيان محدد ومع ذلك هذا يتطلب وقتا إضافيا من وقت التشغيل أثناء عملية استيراد البيانات من المخدم وفي عملية جمع البيانات الوصفية المطلوبة .

وإن أوامر ال ,insert ,update, select, تؤدي الى تحسين أداء وقت التشغيل .

جافا سكريبت :

وهي لغة البرمجة الجديدة لصفحات الويب والمخططات المكتوبة مع جافا سكريبت يمكن تضمينها في صفحات HTML الخاصة بك .مع جافا سكريبت لديك الكثير من الاحتمالات لتعزيز صفحة HTML الخاصة بك مع عناصر مثيرة للاهتمام .على سبيل المثال ستكون قادرا على الاستجابة للأحداث التي بدأها المستخدم بسهولة تامة .كما ويمكنك إنشاء صفحات متطورة حقا بمساعدة جافا سكريبت على شبكة الانترنت .

كيف يمكن تشغيل مستندات الجافا سكريبت ؟

كان أول متصفح لدعم الجافا سكريبت من Navigator-Netscape بالإضافة الى متصفح المايكروسوفت انترنت اكسبلورر وقد ندرك انه من السهل كتابة البرامج النصية بلغة الجافا سكريبت ويعتمد ذلك على مدى معرفتنا المسبقة وإطلاعنا على بعض التقنيات الاساسية ومنها فهم HTML قبل قراءة هذا البرنامج التعليمي ويمكنك العثور في الانترنت على بعض المصادر لتتعلم أساسيات ال HTML وسنعطي على سبيل المثال هذا البرنامج او الفورم البرمجي البسيط باستخدام لغة ال HTML :

```
<html>
<head>
 My first JavaScript
</head>
<body><br>
This is a normal HTML document
<br>
<script language=”JavaScript”>
Document. Write (“this is a java script”)
</script><b r>
Backing HTML again
</body>
</html>
```

ثالثا: الوظائف

وتكـــون عبـــارة عـــن وظـــائف معلنـــة بـــين العلامـــة <HEAD> مـــن صـــفحة الHTML ويبـدو انـه مـن المعقـول الحفـاظ علـى هـذه الوظـائف بـين العلامـات <HEAD> وذلـك بتحميلهـا قبـل ان يمكـن للمسـتخدم القيـام بـأي شـيء .ويمكـن وضـع البـرامج النصـية بـين أو داخـل حقـول تعليـق على الشكل لضمان أن المتصفحات القديمة لن تعرض النص نفسه .

```
<html>
<head>
<script language=”JavaScript”>
 function pushbutton ()
{
        alert (“Hello!”);
}
</script>
</head>
<body>
<form>
<input type=”button” name=”Button1” value=”push me” onclick=”pushbutton ()”>
</form>
</body>
</html>
```

إذا كنـا نريـد اختبـار هـذا الكـود علـى الفـور وكنـا نسـتخدم متصـفح تمكـين جافـا سـكريبت ثـم نقـوم بالضـغط علـى الـزر وننفـذ البرنـامج وهـذا السـيناريو سـوف يقـوم بإنشـاء زر وعنـد الضـغط عليـه سـتظهر نافـذة مكتـوب فيهـا الكلمـة المـراد طباعتهـا ,وفـي الواقـع لـدينا الكثيـر مـن الإمكانيـات فقـط عن طريق اضافة وظائف الى البرامج النصية لدينا .

عناصر المدخلات : إن استخدام العلامة <input> هو لتحديد أحد عناصر النماذج المشتركة بما في ذلك حقول النص والقوائم المتعددة الخيارات الموجودة فوق الصور وهناك العديد من الصفات الخاصة بهذه العلامة ولكل نوع من عناصر المدخلات هناك مجموعة فرعية وقد تكون هناك سمات اضافية للعلامة<input> وذلك استنادا الى اي نوع من السمات تريد ان تستخدم.

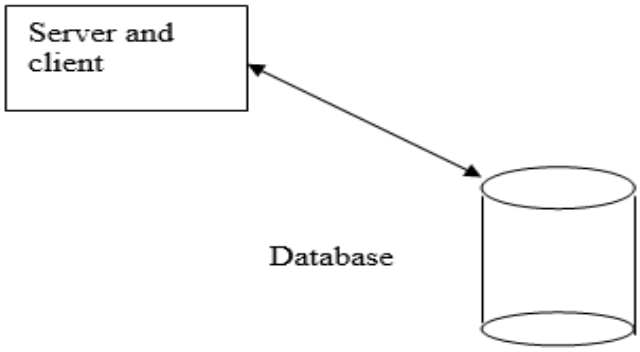
زر الإرسال : يقوم زر الإرسال (<input type=submit>) بعمل ما يوحي به اسمه من اعدادات في الحركة لتقديم النموذج الى المخدم من قبل المستعرض .

زر إعادة الضبط : إن زر اعادة الضبط لا يحتاج إلى الشرح بل يتيح للمستخدم اعادة تعيين القيم او تعيين قيم افتراضية لجميع العناصر في النموذج كما أن المتصفح يعرض زر اعادة الضبط تحت مسمى (reset) .

رابعا: أنماط قواعد البيانات

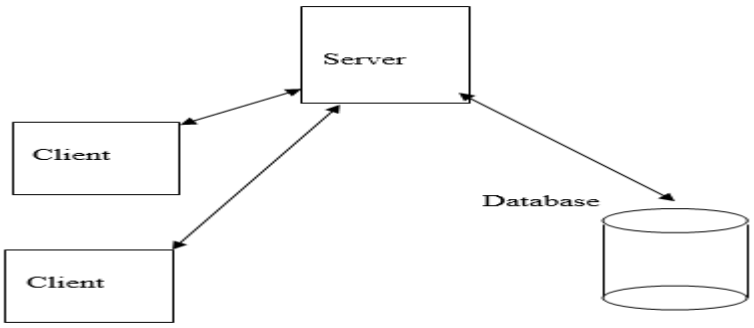
أدى الوصول إلى قاعدة البيانات من خلال التطبيقات والمعرف ب (ADO.NET) إلى ظهور نوع جديد من نماذج قواعد البيانات والتي تختلف عن نماذج خدمة العملاء واستنادا الى عدد الخوادم الوسيطة فإنه إما ان تكون وحيدة الصف او ثنائية الصف او متعددة الصفوف [10].

وحيدة الصف : في هذا المستوى يكون المخدم و العميل هو نفسه بمعنى أن برنامج العميل الذي يحتاج المعلومات مصدر هذا النوع من التركيب الهندسي يكون بالجافا في حال استخدام ملفات قريبة لتخزين البيانات ولكن هذا مفيد فقط في حالة التطبيقات الصغيرة وميزة هذا النوع هي البساطة وقابلية التطبيق المتقدمة [11].



الشكل 5.1 - قاعدة بيانات وحيدة الصف

ثنائيــة الصــف : ضــمن هــذا المســتوى تكــون قاعــدة البيانــات فــي جهــاز والعميــل يكــون علــى جهــاز آخــر ويــتم الــربط بينهمــا مــن خــلال الشــبكة وفــي هــذا النــوع ايضــا تأخــذ ادارة قواعــد البيانــات السـيطرة علــى قاعــدة البيانــات وتــوفر الوصــول الــى العمــلاء فــي الشــبكة وتســمى ايضــا حزمــة البرمجيات والمخدم ويطبق على البرمجيات في اجهزة مختلفة .



الشكل5.2 – قاعدة بيانات ثنائية الصف

Chapter 6

لغة البرمجة C#

وهي لغة برمجية متعددة النماذج والتي تشمل الوظيفية والحتمية العمومية وتخصصات البرمجة غرضية التوجه وتم تطوير هذه اللغة بواسطة شركة مايكروسوفت كجزء من برمجة النت وهي واحدة من 44 لغة برمجة تدعمها ال NET Framework.

وهي لغة بسيطة وحديثة للاغراض العامة وتحتوي على جمل غرضية التوجه ومأخوذة بالأساس من لغة ال C++ وتتأثر أيضا بلغات البرمجة الأخرى ومنها الدلفي والجافا .

1 - ميزات لغة البرمجة C# :

بحسب التصميم فإن لغة البرمجة C# هي لغة البرمجة التي تنعكس بشكل مباشر على البنية التحتية الاساسية للغة العامة (CLI) ومعظم انواع C# تتوافق مع انواع القيم التي تنفذها (CLI) وهذا لا يعني أنه لم يتم الذكر بأنه لا يمكن لهذه اللغة ان تتعامل مع وقت تشغيل اللغة العامة (CLR) أو توليد اللغة المشتركة الوسيطة أو إنشاء اي صيغة محددة اخرى ونظريا يمكن لمترجم اللغة ان يولد رمز الجهاز مثل اللغات التقليدية .

بعض الصفات المميزة والبارزة في لغة ال C# :

- لا يوجد هناك متغيرات أو وظائف عالمية ويجب تعريف كل الأساليب والأعضاء في الصفوف ومن الممكن مع ذلك استخدام أساليب ثابتة أي استخدام المتغيرات في الطبقات بدلا من المتغيرات العالمية .
- المتغيرات المحلية هنا لا يمكن ارفاقها بالتظليل ,خلافا لما يحدث في ال C,C++ وغالبا ما يعتبر التظليل امرا مربكا .
- C# هي لغة تدعم المنطقية في موضوع نوع البيانات وخاصة البيانات التي تهتم بالظروف مثل الوقت فهي تتطلب تعبير من نوع منطقي بينما ال C++ حققت هذه المنطقية ولكن ليس بهذه المرونة بالرغم من اننا نستطيع تحويلها بحرية من والى اعداد صحيحة أما مؤشر ال#C لا يسمح بهذا ففي هذه اللغة الصحيح قابل لان يأخذ معنى منحى الصحة أو منحى الخطأوهذا ما يجبر المبرمجين على استخدام التعبيرات التي تقوم بإرجاع الامور بشكل منطقي, كما ويمكن منع انواع معينة من اخطاء البرمجة.
- مؤشرات عناوين الذاكرة في لغة ال#C لا يمكن إستخدامها على كتل معينة بحد ذاتها لأنها تعتبر غير آمنة والبرامج اذا كانت غير آمنة فهي بحاجة الى إذن لتكون قابلة للتشغيل ويتم ذلك من خلال الوصول الى المراجع الأكثر أمانا وإن التعليمات البرمجية التي لم يتم علامة غير آمنة عليها ستبقى غير قابلة للتخزين والتلاعب بها من خلال مؤشرات من نوع System.Intptr ولكن لا يمكننا إلغاء مرجعيتها .
- ويوفر C# ايضا الدعم المباشر لوضع الصيغة النهائية القطعية مع العبارة .
- التوريث المتعدد غير مدعوم على الرغم من أنه يمكنك تنفيذ عدد من الواجهات وكان هذا القرار من تصميم مهندسي اللغة لمنع المضاعفات التي تحدث اثناء عملية التوريث مما يزيد في دقة التصميم والتنفيذ.

- هـذه اللغـة هـي اكثـر امانـا مـن ال C++ باعتبـار ان التحـويلات الضـمنية تعتبـر اكثـر امانـا مثـل اتسـاع الأعـداد الصـحيحة والتحويـل مـن نـوع مشـتق الـى نـوع أساسـي وطبعـا يـتم فـرض هـذا الامـر فـي وقـت التحويـل البرمجـي خـلال (JIT) وفـي بعـض الحـالات اثنـاء وقت التشغيل ولا توجد تحويلات ضمنية بين القيم المطلقة والأعداد الصحيحة

2 - نظام النمط المشترك :

نظـام الـنمط المشـترك أو الموحـد (CTS) هـو نظـام موحـد لـلC# بمـا فـي ذلـك الأولويـات مثـل الأعـداد الصـحيحة وهـي أقسـام فرعيـة مـن فئـة System.Object فعلـى سـبيل المثـال كـل نـوع يـرث مـن الـنمط ToString فلأسـباب تتعلـق بـالأداء سـوف يـتم تخصيصـها داخليـا ضمن مكدس.

فئات وأنواع البيانات :

(CTS) يفصل أنواع البيانات الى فئتين:

- أنماط القيم .
- أنماط المراجع .

أنمـاط القـيم هـي تجميعـات عاديـة للبيانـات وهـذه الأنمـاط لـيس لـديها الهويـة المرجعيـة ولا دلالات المقارنـة المرجعيـة والمقارنـات لأنمـاط القـيم مـع البيانـات الفعليـة تـتم عـن طريـق المسـاواة وعـدم المسـاواة وتسـتمد أنمـاط القـيم مـن System.ValueType ودائمـا قيمهـا افتراضية وبالإمكان خلقها أو نسخها .

أمـا بالمقابـل فـإن أنمـاط المراجـع لـديها مفهـوم الهويـة المرجعيـة وكـل نمـط مـن أنمـاط المراجـع يختلـف بطبيعتـه عـن كـل حالـة مـن الحـالات الأخـرى حتـى ولـو كانـت البيانـات فـي كلتـا الحـالتين هـي نفسـها ويـنعكس هـذا فـي مبـدأ المسـاواة وعـدم المسـاواة لأنـواع المقارنـات المرجعيـة .وفـي المجمـل لا يمكننـا دائمـا انشـاء مثيـل مـن نـوع مرجعـي ولا نسـخ مثيـل آخـر موجـود أو إجـراء مقارنـة للقيمـة الحاليـة علـى حـالتين فـي القائمـة نفسـها ومـن الأمثلـة علـى أنـواع أنمـاط المراجـع هـي System.String وال System.Array وكلتـا الفئتـين هـي مـن الانواع الواسعة والمعروفة لدى المستخدم .

Chapter 7

الدراسة ومنهجية البرمجيات

أولا: دراسة الإمكانية

دراسة الإمكانية نقوم بها في حال المشاكل لكي يتم فهم هذه المشكلة بشكل واضح ودراسة الامكانية هي نسخة عالية المستوى لعملية التصميم وتحليل الأنظمة والهدف من دراسة الإمكانية هو كيف سيتم حل هذه المشكلة بشكل سريع والتقرير إذا كانت المشكلة قابلة للحل أم لأ وليس هدف هذه الدراسة حل المشكلة بحد ذاتها .

والنظام سوف يتم اختبار الجدوى منه وفق النقاط التالية :

- العملية التقنية .
- العملية الاقتصادية .
- العملية سارية المفعول .

العملية التقنية : المشروع يؤهل نظام خدمة الساعي وتكون العملية تقنية بسبب الصفة المذكورة وهذا المشروع تم تطويره بلغة C# بواجهات المستخدم الرسومية .ويزود هذا المشروع المستوى العالي للثقة والتوفر والتوافق وكل هذا يجعل لغة C# هي اللغة الملائمة لهذا المشروع .

العملية الاقتصادية : النظام الالكتروني سيساعد في أتمتة الاختيار الذي سيقود الارباح وتفاصيل المنظمة وبهذه البرمجيات فإن استخدام القوة البشرية والآلة قابلة للارتفاع بنسبة 80-90% تقريبا .

العملية سارية المفعول : في هذا المشروع الإدارة ستعرف بكل التفاصيل التي ستقدم ضمن هذا المشروع والبيانات ستبقى مصانة .

التطبيق :

التطبيق هي المرحلة التي يتحول خلالها التصميم النظري الى نظام عملي ,وهي المرحلة الأكثر حسما لإنجاز النظام الناجح الجديد وفي إعطاء الثقة للمستخدمين بأن النظام الجديد سيعمل بشكل عملي وبكفاءة عالية . ويمكن للنظام أن يطبق وينفذ فقط بعد إنهاء عملية الاختبار والتأكد من أنه يعمل ضمن المواصفات المحددة له . وهي تتضمن مخطط موضوع بشكل فعال بالإضافة لتحقيق النظام الحالي والقيود الموضوعة له على التطبيق وتصميم الطرق لإنجاز التغيير وتقييم التغيير على الطرق هو جزء من التخطيط .وهناك مهمتان رئيسيتان لتحضير وتهيئة التطبيق وتتجلى في تعليم وتدريب المستخدمين وإختبار النظام .

وهنا يجدر القول بأن العملية الأكثر تعقيدا هي قابلية النظام للتطبيق بالإضافة لتحليل الأنظمة وجهد التصميم المتطلب لتطبيق هذا النظام .

وتشتمل مرحلة التطبيق على عدة مراحل ونشاطات والأجهزة المطلوبة وامتلاك البرمجيات التي يجب تنفيذها والنظام قد يتطلب بعض البرامج والتي يجب ان تكون متطورة وفي نفس الوقت يجب ان تكون مكتوبة ومجربة ,والنظام المستعمل يجب ان يتم تبديله بنظام مجرب وجديد وأما بالنسبة للنظام القديم يجب أن يتم إيقافه .

الإختبار :

إن مرحلة الاختبار هي جزء مهم من تطوير البرامج وهو نظام موضوع للمساعدة في أتمتة عملية معينة لإيجاد الأخطاء وضياع العمليات وأيضا تساعد هذه المرحلة في إكمال التحقق من الحساب سواء في عملية جمع الاهداف او في تحديد متطلبات المستخدمين .

واختبار البرمجيات يتم تنفيذه على مراحل ثلاث وهي :

- أولا يتضمن اختبار الوحدة حيث أنه يتم الاختبار في كل وحدة لزيادة الصواب والصلاحية ويقرر اي العمليات هي المفقودة ايضا .بعد إكتشاف الأخطاء سيتم تصحيحها فورا واختبار الوحدة هو الجزء المهم والرئيسي للمشروع لذلك الاخطاء ستصحح بسهولة ,وفي هذا المشروع سنلاحظ أن كامل النظام منقسم الى عدة وحدات ومتطورة بشكل منفرد ,لذلك فإن اختبار وحدة معينة سوف يجري على الوحدات الفردية ايضا .

- تتضمن الخطوة الثانية اختبار التكامل وإن الوحدات الفردية مقصوصة تحت هذه الوحدة الرئيسية والمجربة ثانية .ويمكن أن يتسبب التوصيل السيء الى فقدان البيانات وهذا يسبب مشاكل خطيرة .
- تتضمن الخطوة النهائية التصديق والإختبار واللذان بدورهما يقرران أي الوظائف البرمجية متوقعة من قبل المستخدم .

الصيانة والبيئة :

وضعت هذه الحالة بسبب إنتشار الأخطاء وذلك نسبة للتوسع في عدد الأنظمةالمرتبطة بالحاسب .وإنشاء مكتبات عامة للبرامج أتت مواكبة لتطوير المشاريع الذي أنتج آلاف البرامج المصدرية وكل تلك البيانات المصدرية وكل تلك البرامج كان لزاما عليها أن تصحح في حال إكتشاف وجود أي خطأ .

وتركز مرحلة الصيانة على التغيير الذي يرتبط بتصحيح الخطأ والتغييرات والتحسينات التي تظهر بحسب متطلبات الزبون .وهناك أربعة من أنواع التغييرات التي تصادف مرحلة الصيانة وهي : التصحيح , والتكيف , والتحسين , والمنع .

التصحيح : حتى بأفضل نشاطات التأمين الممتازة فإن الزبون سيكشف العيوب في البرامج وتقوم الصيانة التصحيحية للبرامج بتصحيح العيوب . والصيانة هي مجموعة نشاطات هندسة البرامج والتي تحدث بعد تسليم البرامج الى الزبون .ونحن يمكن أن نعرف الصيانة بوصف أربعة نشاطات متعهدة ومسؤولة عن البرامج الجاهزة للاستعمال :

- ❖ الصيانة التصحيحية .
- ❖ الصيانة التكيفية .
- ❖ صيانة وتحسين الماضي .
- ❖ صيانة و إعادة هندسة الوقائية .

ويجدر بالذكر هنا ,أنه فقط 20% من عمل الصيانة تكون لتصحيح الاخطاء وال 80% المتبقية ,تكون فقط من أجل تكيف الانظمة الحالية مع التغيير في بيئتهم الخارجية ,وصناعة التحسينات المطلوبة من قبل المستخدمين ,وإعادة هندسية التطبيقات للاستعمال.

التكيف : لا بد أن تتغير البيئة الأصلية للبرامج المتطورة مع مرور الوقت لذلك تؤدي الصيانة التكيفية في تعديل البرمجيات لضمان ملائمة التغييرات مع البيئة الخارجية .

التحسين : الزبون والمستخدم سيعرف وظائف إضافية جديدة والتي تقوم بزيادة الفائدة . وتقوم الصيانة الإدراكية للبرامج بتمديد متطلبات الوظيفة الأصلية كالبرمجيات المستعملة .

المنع : تتدهور برامج الكمبيوتر بسبب التغيير وكنتيجة لذلك فإن الصيانة الوقائية والمسماة إعادة هندسة البرامج ,يجب ان تجرى لتمكين البرامج من خدمة حاجيات المستعملين وجوهريا ,تجعل الصيانة الوقائية التغييرات في الحاسوب تبرمج بشكل جوهري لتكون قابلة للتصحيح بسهولة أكبر .

ثانيا: منهجية البرمجيات

إن منهجيـــة البرمجيـــات أتبعـــت فـــي هـــذا المشـــروع , وتتضـــمن المنهجيـــة غرضـــية التوجـــه وتطبيـــق المنهجيـــات المطـــورة للنظـــام , وبـــالرغم مـــن وجـــود العديـــد مـــن التطبيقـــات المنهجيـــة فإنـــه يتوجب علينا توصيف بعضا منها على الشكل التالي:

مطور أنظمة التطبيقات (نظام دورة الحياة)

يعنـــي بـــأن هنـــاك عـــددا متزايـــدا مـــن التطبيقـــات (مثـــل أنظمـــة دعـــم القـــرارات) التـــي ينبغـــي تطويرهـــا باســـــــتخدام اســـتراتيجية عمليـــة تجريبيـــة مثـــل النمـــاذج وكميـــة كبيـــرة مـــن أعمـــال التطـــوير الجديـــدة , و تواصـــل التطبيقـــات إشـــراك العمليـــة الرئيســـية فيهـــا بشـــكل واســـع . نظـــم التطبيـــق تشـــمل مســـتوى عـــالي مـــن التنظـــيم وخاصـــة فـــي مجـــال مهمـــات المســـتخدم ومهمـــات المطـــور وعـــادة مـــا يكونـــا بدرجـــة عاليـــة ومرتفعـــة , وتشـــير هـــذه العوامـــل الاســـتراتيجية إلـــى ضـــمان خطـــي أو تكـــراري . الأســـلوب الأكثـــر شـــيوعا لهـــذه الفئـــة مـــن مشـــاكل المرحلـــة هـــو تطـــوير نظـــام الوســـائط لـــدورة الحيـــاة وفيهـــا كـــل مرحلـــة مـــن مراحـــل التنميـــة واضـــحة المعـــالم ولهـــا متطلبـــات واضحة لإنجازها.

توصـــف دورة حيـــاة تطـــوير النظـــام فـــي التفصـــيل لأنهـــا عبـــارة عـــن منهجيـــة ملائمـــة لجـــزء كبيـــر مـــن أعمـــال التطـــوير الجديـــدة. الفكـــرة الأساســـية لـــدورة حيـــاة تطـــوير النظـــام هـــو أن هنـــاك عمليـــة واضـــحة المعـــالم التـــي يـــتم مـــن خلالهـــا تطبيـــق تصـــورها المســـتقبلي وتطويرهـــا وتنفيـــذها, و تضـــيف دورة الحيـــاة الهيكـــل الأساســـي إلـــى العمليـــة الإبداعيـــة.و مـــن أجـــل الإدارة والـــتحكم فـــي جهـــود التنميـــة , فإنـــه مـــن الضـــروري أن تعـــرف مـــا كـــان ينبغـــي القيـــام بـــه و مـــا تـــم إنجازه وما لم يتم إنجازه.

التعليمـــات فـــي دورة حيـــاة تطـــوير النظـــام تـــوفر أســـاس لـــلإدارة و المراقبـــة, لأنهـــا تحـــدد شـــرائح تـــدفق العمـــل، والتـــي يمكـــن تحديـــدها لأغـــراض إداريـــة . يـــتم وصـــف المراحـــل فـــي دورة حيـــاة تطـــوير نظـــم المعلومـــات بشـــكل مختلـــف مـــن قبـــل الكتّـــاب ، ولكـــن لابـــد مـــن وجـــود بعـــض الاختلافـــات فـــي ضـــرورة وطريقـــة التصـــنيف. هنـــاك اتفـــاق عـــام علـــى تـــدفق خطـــوات التنميـــة وضرورة إجراءات المراقبة في كل مرحلة.

تتكون مراحل دورة تطوير نظام المعلومات للتطبيق من ثلاث مراحل رئيسية وهي :

- التعريف
- التنمية
- التركيب و التشغيل

المرحلة الأولى من هذه العملية هي التي تحدد متطلبات المعلومات المجدية لنظام فعّال من حيث التكلفة , ثم يتم تحويل الاحتياجات إلى النظام المادي للأشكال والإجراءات والبرامج وما إلى ذلك, وذلك من خلال تصميم النظام وبرمجة الكمبيوتر والتطوير الداخلي .

النظام الناتج هو اختبار وضع موضع التنفيذ , و لا يوجد نظام مثالي حتى لا يكون هناك دائما حاجة لإجراء تغييرات الصيانة , ولإنهاء الدورة كاملة يجب أن يكون هناك تدقيق آخر للنظام لتقييم مدى التنفيذ , وكيف تجتمع التكلفة وأداء المواصفات .

يمكن تقسيم مراحل التعريف و التطوير و التركيب و التشغيل إلى خطوات أو عبارات أصغر على النحو التالي:

1- التعريف

- التعريف المقترح: هو إعداد طلب التطبيقات المقترحة .
- تقييم الجدوى: هو تقييم الجدوى والتكاليف والفوائد من النظام المقترح .
- تحليل الاحتياجات من المعلومات: يعني تحديد المعلومات المطلوبة .

2- التصميم

- التصميم النظري: التصميم الموجه للمستخدم ويكون مأخوذ من تطوير التطبيقات.
- تصميم النظام الفيزيائي : تصميم مفصل لتدفقات العمليات في نظام معالجة وإعداد مواصفات البرنامج.

3- التطوير

- تطوير البرامج: أي ترميز واختبار برامج الكمبيوتر .
- تنمية الإجراء: و هو تصميم الإجراءات وإعداد تعليمات المستخدم .

4- التركيب و التشغيل

- التحويل: هو اختبار النظام النهائي وتحويله.
- التشغيل والصيانة: يفضل أن تكون عملية الصيانة بشكل شهري.

آخر مراجعة: وتعني بتقييم عملية التنمية, وكذلك فإن الموافقة الرسمية في التعديل على المشروع مطلوبة من العاملين عليه وكذلك من مدير تطوير المشروع.

Chapter 8

الإختبار

الاختبار هو عملية تنفيذ برنامج مع مراعاة العثور على خطأ , وهو العنصر الحاسم في ضمان جودة البرمجيات, ويقدم الاستعراض النهائي للمواصفات و الترميز. اختبار النظام هو مرحلة هامة حيث يمثل اختبار حالة شاذة و مثيرة للاهتمام في البرنامج , وبالتالي يتم تنفيذ سلسلة من الاختبارات المقترحة للنظام قبل أن يصبح جاهزا لقبول المستخدم.

هناك حالة اختبار واحدة جيدة حيث تكمن في أن النظام له احتمال كبير في العثورعلى الخطأ الغير مكتشف ويمكن تصنيف حالات اكتشاف الأخطاء كما في الحالات التالية:

1- اختبار الأهداف:

- الاختبار بشكل عام هو عملية تنفيذ برنامج بقصد إيجاد خطأ.
- حالة الاختبار الجيد هي أن لديه احتمال العثور على أخطاء كما لم تكتشف بعد.
- الاختبار الناجح هو أن يكشف الأخطاء غير المكتشفة.

2- مبادئ الاختبار:

- يجب أن تكون جميع الاختبارات يمكن تعزيزها باللوازم المطلوبة لإنهاء متطلبات المستخدم.
- ينبغي التخطيط للاختبارات قبل فترة طويلة من بدء الاختبار.
- يجب أن يبدأ الاختبار على نطاق صغير و التقدم نحو إجراء العملية على نطاق أكبر.
- حالة الإختبار الشاملة غير ممكنة.
- ليكون الاختبار أكثر فعالية ينبغي إجرائه من قبل طرف ثالث مستقل.

الهدف الرئيسي لتصميم حالة الاختبار هو لاستخلاص مجموعة من الاختبارات التي لديها أعلى نشاط ,وذلك لكشف العيوب في البرمجيات , و لتحقيق هذا الهدف من حالة الاختبار فهناك تقنيتين مختلفتين من تقنيات تصميم حالة الاختبار و هما:

- اختبار الصندوق الأبيض.
- اختبار الصندوق الأسود.

حيث أن حالة اختبار الصندوق الأبيض تركّز على بنية تحكم البرنامج , وتستمد حالات الاختبار للتأكد من أن جميع البيانات في البرنامج تم تنفيذها مرة واحدة على الأقل خلال الاختبار وأن كافة الشروط المنطقية تمّ تنفيذها أيضا.أما في حالة اختبار الصندوق الأسود فوضعت هذه الحالة للتحقق من صحة المتطلبات الوظيفية دون النظر إلى الأعمال الداخلية للبرنامج, و يركّز اختبار الصندوق الأسود أساسا على مجال المعلومات في البرنامج, و يجدر بالذكر أن حالات الاختبار السابقة مستمدة عن طريق تقسيم المدخلات والمخرجات بطريقة تغطي حالة الاختبار بشكل صحيح, ووظيفة الاختبار في حالة الصندوق الأسود وضعت للتحقق من الوظائف الغير صحيحة ومن البيانات المفقودة, و للتحقق من أخطاء الواجهة, والأخطاء المنطقية في هياكل البيانات.

3- استراتيجيات الاختبار:

يجب على استراتيجية اختبار البرمجيات استيعاب التجارب على مستوى منخفض والتي تعتبر ضرورية للتحقق من أن كل رمز مصدر تم تنفيذه بشكل صحيح وكذلك في استراتيجية الاختبارات الرفيعة المستوى والتي تحقق وظائف النظام بشكل كبير ضد متطلبات العملاء .

4- أساسيات الاختبار:

كما ذكرنا سابقا فإن الاختبار هو عملية تنفيذ البرنامج بهدف إيجاد الخطأ وأن حالة الاختبار الجيدة تكمن في احتمال العثور على الخطأ الغير مكتشف, لذلك فإن تم إجراء الاختبار بنجاح فإنه سيكشف الأخطاء في البرنامج, والاختبار لا يمكن أن يظهر عدم وجود عيوب, و يمكنه فقط إظهار العيوب في البرمجيات .

5- اختبار تدفق المعلومات:

تدفق المعلومات للاختبار يعطي حالة النمط, وهناك فئتين من صفوف المدخلات تزود لاختبار العملية في النظام , ومن ضمن هذه الصفوف يوجد صف يعرف باسم التكوين و الذي يتضمن مواصفات و متطلبات البرنامج , و مواصفات التصميم و شفرة المصدر, و يشمل صف التكوين المواصفات التالية:

- حالات الاختبار.
- خطة الاختبار.
- أدوات الاختبار.

وتجرى جميع هذه الاختبارات و يتم تقييم كل النتائج, وبالطبع فجميع ما ذكر هو عبارة عن مقارنة نتائج الاختبار مع النتائج المتوقعة, وفي حال اكتشاف بيانات خاطئة, فهذا يعني أن الخطأ موجود ضمنيا ويبدأ التصحيح تلقائيا.

6- اختبار الوحدة :

اختبار الوحدة ضروري للتحقق من الرمز الذي أنتج خلال مرحلة الترميز, وبالتالي كان الهدف هو اختبار المنطق الداخلي من الوحدات وذلك باستخدام صفا مفصلا وضع كدليل لا أكثر، يتم اختبار مسارات مهمة لكشف الأخطاء في حدود الوحدات, أجريت هذه الاختبارات من خلال مرحلة البرمجة نفسها, وتم اختبار جميع وحدات قاعدة البيانات المعرفة باسم (hospital) بنجاح.

7- اختبار التكامل:

يركز على أنماط وحدة الاختبار وبناء هيكل البرنامج الذي تمليه مرحلة التصميم.

8- اختبار النظام:

اختبار النظام يكمن في اختبار كل وحدات التكامل في النظام, وتختبر أيضا للعثور على التناقضات بين النظام وهدفه الأصلي, و بين المواصفات الحالية ووثائق النظام , والهدف الرئيسي هو التوافق بين الوحدات الفردية.إنّ مسار الاتصال المحدد ODBC سيعلمنا إن كان مسار النظام صحيح أم لا, والنتائج والمخرجات في النظام ستعطينا حالة الوثوقية والصلاحية وذلك بمقارنتها بالمدخلات الموجودة في النظام أساسا و بمقارنتها بالمخرجات المتوقعة أيضا.

9- اختبار القبول:

ويتم هذا الاختبار للتحقق من جاهزية النظام للتنفيذ, إنّ اختبار القبول يبدأ عندما يكون النظام كامل وجاهز للتنفيذ. والغرض منه هو توفير كل المتطلبات للمستخدم النهائي مع الثقة بأن النظام جاهز للاستخدام, و اختبار القبول مبني على التخطيط وتنفيذ الاختبارات الوظيفية، وأيضا مبني على اختبارات الأداء واختبارات الضغط من أجل إثبات أن النظام المطبق يرضي جميع متطلباتها.

تتضمن أدوات اختبار القبول المواصفات المهمة التالية:

- محلل تغطية الاختبار -يسجل مسارات التحكم المتبعة لكل حالة اختبار.
- محلل الوقت أو يسمى تقرير الوقت المتواجد في عدة مناطق عالميا ليعمل ذلك على تحسين أداء النظام.
- يستخدم الفاحصين الشيفرة القياسية لتفقد أوامر البرنامج من الانحرافات عن المعايير والمبادئ التوجيهية.

10- حالات الاختبار:

تستخدم حالات الاختبار للتأكد من أن جميع البيانات في البرنامج تم تنفيذها مرة واحدة على الأقل خلال الاختبار و أنّ تنفيذ الشروط المنطقية تمّ اختبارها أيضا, و يمكن لمهندس البرمجيات اكتشاف عدة أمور باستخدام طريقة اختبار الصندوق الأبيض والتأكد من التالي :

- ضمان أن القرارات المنطقية على الجانب الصحيح.
- ممارسة جميع القرارات المنطقية في كلتا الحالتين الصحيحة والخاطئة.
- تنفيذ جميع الحلقات في حدودها و مراحل عملها.
- ممارسة بنية قاعدة البيانات لضمان صحتها.

حالة الاختبار للنظام يجب التأكد من عرضها وتقديمها للمراجعة قبل بدء اختبار النظام.

Chapter 9

التنفيذ العملي

أولا: شاشات الإخراج



(الشكل 9.1) الواجهة الرئيسية



(الشكل9.2)الواجهة الرئيسية (2)



(الشكل 9.3) واجهة الإدارة



(الشكل 9.4) واجهة تسجيل دخول الأطباء



(الشكل 9.5) واجهة تسجيل طبيب جديد



(الشكل 9.6) واجهة تسجيل دخول الموظفين



(الشكل 9.7) واجهة تسجيل موظف جديد



(الشكل 9.8) واجهة التذكير بكلمة السر



(الشكل 9.9) واجهة المختبرات



(الشكل 9.10) واجهة مسرح العمليات



(الشكل 9.11) واجهة الاختبارات الطبية (فحص الدم)



(الشكل 9.12) واجهة الاختبارات الطبية (فحص البول)



(الشكل 9.13) واجهة تسجيل المرضى



(الشكل 9.14) واجهة تسجيل المرضى الداخليين في المستشفى



(الشكل 9.15) واجهة تسجيل المرضى الخارجين في المستشفى



(الشكل 9.16) واجهة بيانات تخريج المرضى من المستشفى



(الشكل 9.17) واجهة مخزن الأدوية (الصيدلية)

ثانيا: الشبكة

الشبكة الحاسوبية هي الوسيلة المثلى للإستغادة من موارد أكثر من جهاز حاسوبي عن طريق مشاركة هذه الموارد مثل مساحة التخزين والبيانات وأيضا مشاركة الأجهزة الأخرى مثل الطابعات وبالطبع مشاركة الدخول الى الإنترنت , وتتنوع الشبكات من حيث الشبكات , كما وتتنوع بروتوكولات نقل البيانات عبر الشبكات ولكن اهمها وأكثرها انتشارا هو ال Ethernet.

لكي يكون لديك شبكة داخلية فإنه من البديهي أن يكون لديك أكثر من جهاز حاسب بالإضافة الى آلات طابعة وسيرفرات في مكان العمل [12]. وبدأت الشبكات خلال الستينات من القرن العشؤين كنتاج طبيعي من صناعة واكتشاف الحاسب الآلي في تلك الفترة , كما وأن التقدم الكبير والسريع في عالم التكنولوجيا والرقميات أدى الى ظهور بروتوكولات لنقل البيانات و أحد أهم هذه البروتوكولات هو بروتوكول ال TCP/IP والذي يعتبر حتى يومنا هذا بمثابة الربوتوكول الأساسي لجميع الشبكات العاملة في العالم ابتاء من الشبكات المحلية حتى الشبكات العنكبوتية . واستمر تطور الشبكات حتى أصبحت جزء لا يتجزأ من أي نظام تشغيل .

وعن التقنية المتبعة في هذا المشروع فقد تم ربط الشبكة باستخدام نمط شبكة داخلية تتألف من :

- عدد من الحواسب الشخصية وتم توظيف أحدها كحاسب أساسي للشبكة والنظام ككل .
- مخدم أو سيرفر أساسي لتخزين البيانات داخله مع وجود مخدم إضافي يستخدم في حال الأعطال ويحتوي على نسخة طبق الأصل من النظام الحالي لسهولة إسترجاع البيانات .

- الكابلات اللازمة لتوصيل الشبكة داخليا ويختلف نوع ونمط الكابل بحسب الحاجة ونذكر هنا بعض أنواع الكابلات (الكابلات الزوجية Twisted pair cable وهي أكثر الأنواع استخداما على الإطلاق حيث يتم استخدامها في

أغلب الشبكات المحلية ومن أهم أنواعها كابلات ال CAT5 والتي تتميز بالمرونة وسهولة الاستعمال وتغطي حتى مساحة 100 متر وهناك أنواع أخرى من الكابلات المستعملة في توصيل الشكات ومنها الكابلات المحورية وهي أحد أنواع الموصلات النحاسية المستعملة عادة في الشبكات الكبيرة الحجم ومميزاتها تنحصر في امتداد التوصيل وقوة البيانات لمسافة أطول من الكابلات الزوجية وبالتالي يمكن استعمالها أيضا ضمن هذا المشروع)

نظرة عامة عن البروتوكولات

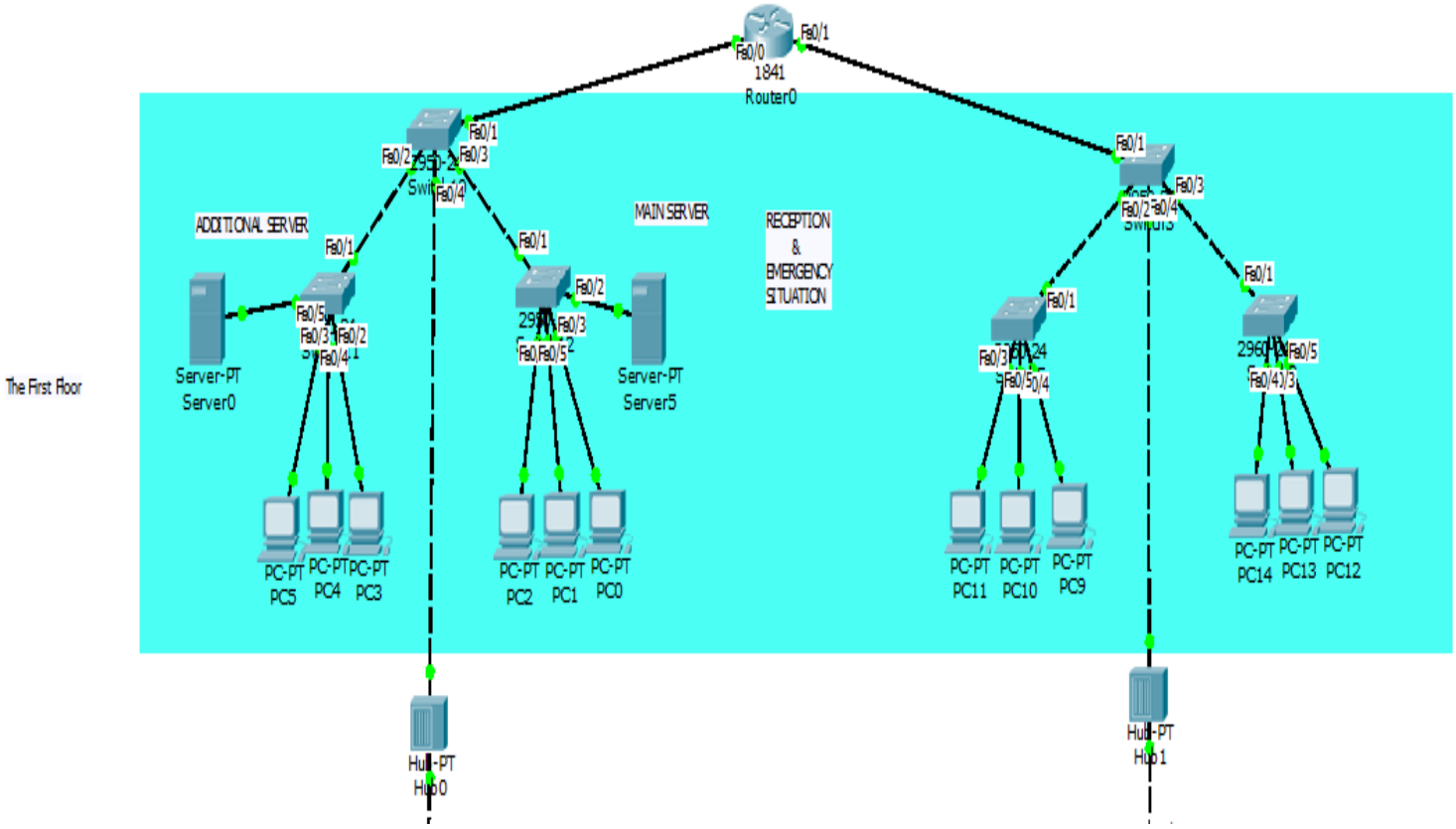
البروتوكول هو الطريقة التي تتصل بها الأجهزة ببعضها كما ويعتبر أحد أنواع التشفير التي تسمح للجهاز بترجمة محتويات الرسالة دون غيره ومن أشهر أنواع البروتوكولات المنتشرة اليوم هو بروتوكول ال TCP/IP .

نظرة عامة عن ال IP address

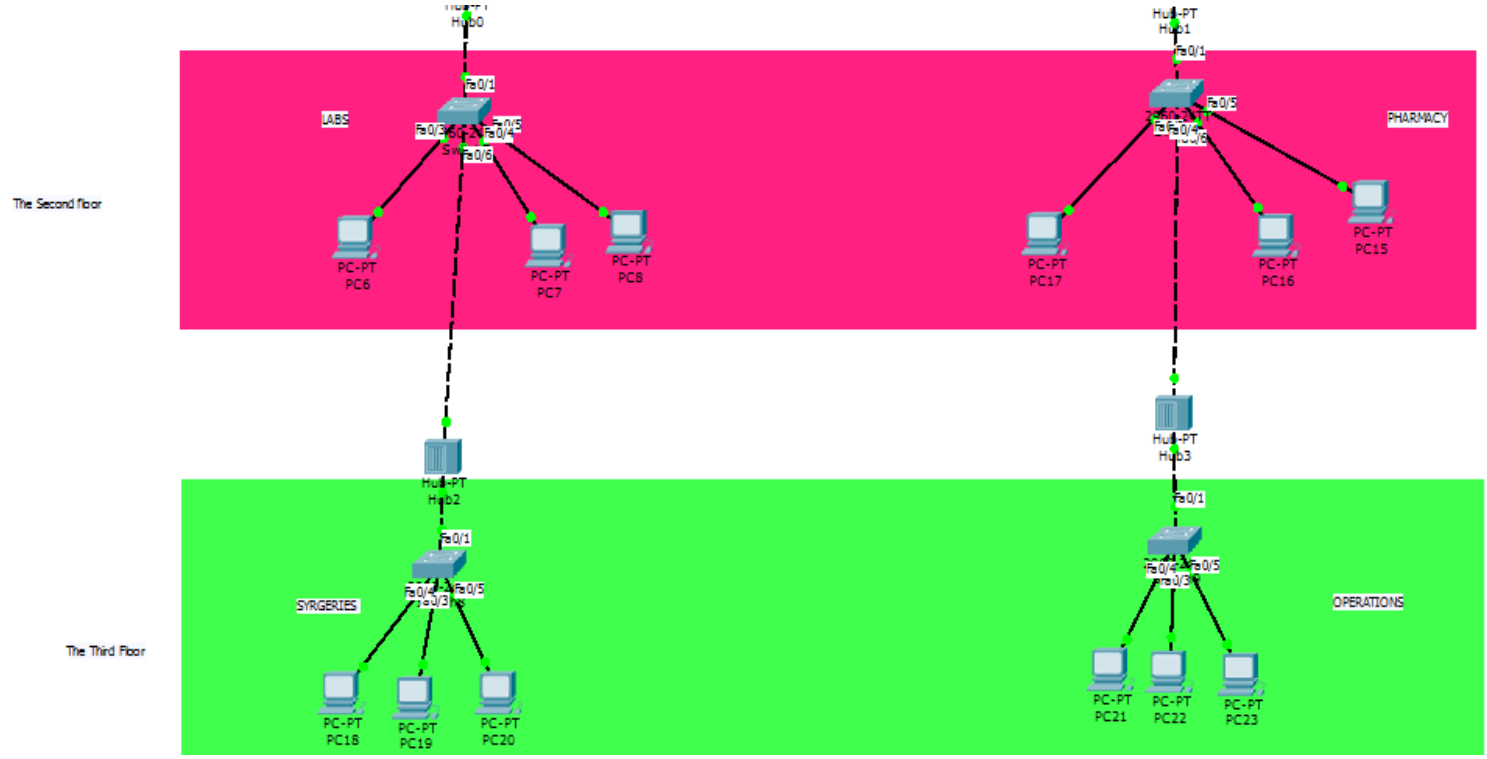
وهو رقم يتم إعطائه لكل شخص يقوم بعمل إتصال على شبكة الإنترنت بغض النظر إذا كانت الشبكة التي يتم التعامل معها هي شبكة محلية أو شبكة عنكبوتية .

تنفيذ الشبكة :

يبين الشكل التالي بيئة تصورية وتنفيذية للشبكة المتبعة في هذا المشروع باستخدام عدد من الحواسب الشخصية الموزعة في أقسام المستشفى والتي تشكل لنا العميل بالإضافة إلى مخدم أساسي أو سيرفر والذي يعتبر بمثابة الخلية الحيوية لأي شبكة وطبعا الكابلات اللازمة للتوصيل مع مزود خدمات للشبكة وهو عبارة عن الراوتر الذي يتم استخدامه في حال وصل الشبكة المحلية مع الشبكة العنكبوتية وذلك لتوسيع نطاق العمل , بعض الإجهزة الخاصة بتوصيلات الشبكات كال (SWITCHES-HUBS) وذلك للمحافظة على قوة نقل البيانات وثباتها ما بين أقسام الشبكة المتنوعة .



(الشكل 9.18) بيئة الطابق الأول من المستشفى



(الشكل 9.19) بيئة الطابق الثاني والثالث من المستشفى

Chapter 10

الخلاصة و الأفكار المستقبلية

أولا: الخلاصة

تم تصميم المشروع في طريقة قابلة للتعديل والإضافة عليه في المستقبل والقيام بهذا التعديل بشكل سهل وبسيط. ويمكن الاستنتاج من خلاصة المشروع ما يلي:

- أتمته النظام بشكل كامل وذلك يؤدي إلى تحسين الكفاءة.
- توفر واجهة المستخدم حالة استخدام أفضل بالمقارنة مع النظام المعتاد استخدامه في أغلب المشافي.
- يعطي حق إعطاء المعلومات المناسبة للمستخدمين او المعنيين اعتمادا على الصلاحيات الخاصة بهم.
- يتغلب على حالة التأخير في الاتصالات بين الأقسام.
- مجال تحديث المعلومات الموجود في النظام يؤدي لاستخدامه بشكل أسهل.
- نظام الضمان، وأمن البيانات والوثوقية هي السمات البارزة.
- إن النظام لديه نطاق كاف للتعديل في المستقبل إذا كان ذلك ضروريا.

ثانيا: الأفكار المستقبلية

هـذا التطبيـق يتجنـب العمـل اليـدوي ويتجنـب مشـاكله الشـائعة أيضـا, و هـو وسـيلة سـهلة للحصول على المعلومات بشأن حالات المرضى المختلفة والتي تكون موجودة في النظام.

فريـق العمـل عمـل بجـد مـن أجـل تقـديم موقـع علـى الشـبكة بشـكل محـاكي للواقـع ومراعيـا لعـدة جوانـب أيضـا مثـل تخـزين وجلـب المعلومـات حـول عـدد لا بـأس بـه مـن النشـاطات الموجـودة داخـل معظـم المشـافي, ولابـد مـن الـذكر أن مشـروعنا يمكـن أن يـتم بطريقـة أفضـل فـي حال تم التطوير عليه.

أمـا بالنسـبة للنظـرة المسـتقبلية للمشـروع تكمـن فـي القيـام بتطـوير الخـدمات عبـر الإنترنـت, ذلـك يعنـي أنـه إذا كـان المـريض لديـه أي مشـاكل فسـيمكنه إرسـال مشـكلته إلـى الطبيـب مـن خـلال شـبكة الانترنـت فـي منزلـه ثـم سـيقوم الطبيـب بإرسـال الـرد إليـه, ولابـد أن يكـون لـدى المريض اسم مستخدم وكلمة سر خاصة به.

يمكـن أن نتوصـل أيضـا لاسـتراتيجية فـي المسـتقبل القريـب نسـتطيع مـن خلالهـا أن نعمـل نظـام مشـابه لنظامنـا الحـالي ولكـن للهواتـف الذكيـة وربطهـا مـع النظـام الحـالي ليـتمكن الموظفين والأطباء والمرضى أن يتواصلوا مع المشفى بشكل أسهل.

هـذه الفكـرة أتـت مواكبـة للحالـة الشـائعة فـي أيامنـا هـذه, وذلـك بـأن معظـم الأشـخاص فـي وقتنـا الـراهن أصـبحوا يتعـاملوا مـع الهواتـف الذكيـة أكثـر مـن التعامـل مـع أي شـيء آخـر , لـذلك سـيتم ربـط كـل مسـتخدم موجـود فـي هـذا التطبيـق وسـيمكنه بـأن يبقـى علـى تواصـل دائـم من خلال وضع آلية تنبيه مربوطة مع الالتزامات الخاصة به .

References

مواقع النت و نشرات علمية:

[1] -http://srv137.mans.edu.eg/mus/system نظام ابن سينا لادارة المستشفيات

[2]http://advacsys.com/index.php?option=com_content&view=article&id=3:hims&catid=18&Itemid=143&lang=ar

أنظمة أدفاك المحدودة

[3]http://advacsys.com/index.php?option=com_content&view=article&id=3:hims&catid=18&Itemid=143&lang=ar

الأساليب الحديثة في إدارة المستشفيات

[4] http://projectabstracts.com/4207/hospital-management-system-in-asp-net.html

[5] http://codemyne.net/Projects/hospital-management-system-developed-in-aspdotnet-csharp-and-sqlserver.aspx?visitid=10&type=4

[6] http://codemyne.net/Projects/hospital-management-system-developed-in-aspdotnet-csharp-and-sqlserver.aspx?visitid=10&type=4

/[7] http://www.asp.net

[8]- http://en.wikipedia.org/wiki/.NET_Framework

[9] http://arabhardware.net/forum/showthread.php?t=202826

الكتب:

[10] Introduction to SQL. Fourth edition.

[11] The c# programming language third edition.

[12] The history of networks by Adam leman.

APPENDIXES

جداول قاعدة البيانات

Doctor Registration

Name	NULL / NOTNULL	TYPE	KEY
Name	Null	Varchar (20)	
Loginid	Null	Varchar(20)	
Password	Null	Varchar(20)	
Department	Null	Varchar(20)	
Specialization	Null	Varchar(20)	
Phonenumber	Null	Bigint	
Address	Null	Varchar(20)	
Email	Null	Varchar(20)	

Employee Registration

Name	NULL / NOTNULL	TYPE	KEY
Name	Null	Varchar (20)	
Loginid	Null	Varchar(20)	
password	Null	Varchar(20)	
Department	Null	Varchar(20)	
Address	Null	Varchar(20)	
Email	Null	Varchar(20)	

Inpatient Registration

Name	NULL / NOTNULL	TYPE	KEY
Patientname	Null	Varchar (20)	
Patientid	Not Null	Varchar(20)	Primary
Gender	Null	Varchar(20)	
Age	Null	Varchar(20)	
Phoneres	Null	Varchar(20)	
Phonemob	Null	Varchar(20)	
Maritual	Null	Varchar(20)	
Occupation	Null	Varchar(20)	
Admid	Null	Varchar(20)	
Admdate	Null	Varchar(20)	
Admtime	Null	Varchar(20)	
Status	Null	Varchar(20)	
Symptoms	Null	Varchar(20)	
Department	Null	Varchar(20)	
Wardno	Null	Varchar(20)	
Bedno	Null	Varchar(20)	
Doctor	Null	Varchar(20)	

Out Patient Registration

Name	NULL / NOTNULL	TYPE	KEY
Patientname	Null	Varchar (20)	
Patientid	Null	Varchar(20)	Primary
Gender	Null	Varchar(20)	
Age	Null	Varchar(20)	
Address	Null	Varchar(20)	
Assigndoctor	Null	Varchar(20)	
Phoneres	Null	Varchar(20)	
Phonemob	Null	Varchar(20)	
Opdate	Null	Varchar(20)	
Department	Null	Varchar(20)	

For Patient Information

Name	NULL / NOTNULL	TYPE	KEY
Pid	Null	Varchar(20)	Foreign
Patientname	Null	Varchar(20)	
Age	Null	Varchar(20)	
Department	Null	Varchar(20)	
Doctor	Null	Varchar(20)	

For Surgery Information

Name	NULL / NOTNULL	TYPE	KEY
Pid	Null	Varchar(20)	Foreign
Patientname	Null	Varchar(20)	
Age	Null	Varchar(20)	
Department	Null	Varchar(20)	
Doctor	Null	Varchar(20)	
Surgerydate	Null	Varchar(20)	

For Blood test

Name	NULL / NOTNULL	TYPE	KEY
Patienttype	Null	Varchar(20)	
Patientid	Null	Varchar(20)	
Patientname	Null	Varchar(20)	
Mediclatestype	Null	Varchar(20)	
Bloodgroup	Null	Varchar(20)	
Haemoglobin	Null	Varchar(20)	
Bloodsugar	Null	Varchar(20)	
Sacid	Null	Varchar(20)	
Description	Null	Varchar(20)	

For Urine test

Name	NULL / NOTNULL	TYPE	KEY
Patienttype	Null	Varchar(20)	
Patientid	Null	Varchar(20)	
Patientname	Null	Varchar(20)	
Mediclatestype	Null	Varchar(20)	
Color	Null	Varchar(20)	
Clarity	Null	Varchar(20)	
Odor	Null	Varchar(20)	
Specificgravity	Null	Varchar(20)	
Glucose	Null	Varchar(20)	
Description	Null	Varchar(20)	

For Operation

Name	NULL / NOTNULL	TYPE	KEY
Patienttype	Null	Varchar(20)	
Patientid	Null	Varchar(20)	
Patientname	Null	Varchar(20)	
Refdoctor	Null	Varchar(20)	
Operationtype	Null	Varchar(20)	
Operatonresult	Null	Varchar(20)	

For Pharmacy Information

Name	NULL / NOTNULL	TYPE	KEY
Patienttype	Null	Varchar(20)	
Patientid	Null	Varchar(20)	
Department	Null	Varchar(20)	
Patientname	Null	Varchar(20)	
Medicine	Null	Varchar(20)	

For Discharge Summary

Name	NULL / NOTNULL	TYPE	KEY
Patientid	Null	Varchar(20)	
Patientname	Null	Varchar(20)	
Joiondate	Null	Varchar(20)	
Dischargedate	Null	Varchar(20)	

ASP CODE:

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="surgeries.aspx.cs"
Inherits="surgeries" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

<asp:Label ID="Label1" runat="server" Font-Bold="False" Font-
Names="Modern" Font-Size="24pt"

        ForeColor="White" Style="z-index: 100; left: 146px; position:
absolute; top: 184px; color: red;"

        Text="Surgery Information  "></asp:Label>

<table style="width: 306px; z-index: 101; left: 239px; position:
absolute; top: 289px; color: white; font-family: 'Lucida Console';
height: 113px;">

<tr>

    <td style="width: 682px; height: 20px">

        Patient Id

    </td>

    <td style="width: 115px; height: 20px">

        <asp:DropDownList ID="pidddl" runat="server" Width="155px"
AutoPostBack="True"
OnSelectedIndexChanged="pidddl_SelectedIndexChanged" >

        </asp:DropDownList>

        </td>

</tr>

<tr>

    <td style="width: 682px; height: 20px">

    Patient Name</td>
```

```
    <td style="width: 115px; height: 20px">
    <asp:TextBox ID="pntxt" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td style="width: 682px; height: 20px">
    Age
    </td>
    <td style="width: 115px; height: 20px">
    <asp:TextBox ID="agtxt" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
<td style="width: 682px; height: 6px">
    Surgery Date
    </td>
    <td style="width: 115px; height: 6px">
    <asp:TextBox ID="sdtxt" runat="server"></asp:TextBox>
    </td>

</tr>
<tr>
    <td style="width: 682px; height: 20px">
    Department
    </td>
    <td style="width: 115px; height: 20px">
```

```
    <asp:TextBox ID="deptxt" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td style="width: 682px; height: 20px">
    Doctor
    </td>
    <td style="width: 115px; height: 20px">
    <asp:TextBox ID="doctxt" runat="server"></asp:TextBox>
    </td>
</tr>
</table>
<table style="z-index: 102; left: 289px; position: absolute; top:
485px">
        <tr>
        <td style="width: 82px; height: 26px">
          <asp:Button ID="Button1" runat="server" Text="Save"
Width="85px" OnClick="Button1_Click"  />
        </td>
       <td style="width: 67px; height: 26px">
             <asp:Button ID="Button2" runat="server" Text="Cancel"
Width="92px" OnClick="Button2_Click" />
       </td>
       </tr>
       </table>
    <asp:Calendar ID="Calendar1" runat="server" style="z-index: 102;
left: 548px; color: white; position: absolute; top: 376px"
```

```
BackColor="White" BorderColor="#999999" CellPadding="4"
DayNameFormat="Shortest" Font-Names="Verdana" Font-Size="8pt"
ForeColor="Black" Height="180px"
OnSelectionChanged="Calendar1_SelectionChanged" Width="200px">

        <SelectedDayStyle BackColor="#666666" Font-Bold="True"
ForeColor="White" />

        <TodayDayStyle BackColor="#CCCCCC" ForeColor="Black" />

        <SelectorStyle BackColor="#CCCCCC" />

        <WeekendDayStyle BackColor="#FFFFCC" />

        <OtherMonthDayStyle ForeColor="#808080" />

        <NextPrevStyle VerticalAlign="Bottom" />

        <DayHeaderStyle BackColor="#CCCCCC" Font-Bold="True" Font-
Size="7pt" />

        <TitleStyle BackColor="#999999" BorderColor="Black" Font-
Bold="True" />

    </asp:Calendar>

</asp:Content>
```

C# Code:

```
using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Data.SqlClient;


public partial class surgeries : System.Web.UI.Page

{

    SqlConnection cn = new SqlConnection("server=DATASERVER;user
id=sa;password=sqlserver;database=arun");

    protected void Page_Load(object sender, EventArgs e)

    {

        if (!Page.IsPostBack)

        {

            filldrop();

            pidddl.Items.Insert(0, "...Select...");

        }
```

```
    }
    void filldrop()
    {
        cn.Open();
        string qry;
        qry = "select patientid from hospital_inpatient";
        SqlDataAdapter da = new SqlDataAdapter(qry, cn);
        DataSet ds = new DataSet();
        da.Fill(ds, "hospital_inpatient");
        pidddl.DataSource = ds;
        pidddl.DataTextField = "patientid";
        pidddl.DataBind();

        cn.Close();
    }
    protected void Calendar1_SelectionChanged(object sender, EventArgs
e)
    {
        sdtxt.Text = Calendar1.SelectedDate.ToShortDateString();
    }
    protected void pidddl_SelectedIndexChanged(object sender, EventArgs
e)
    {
        cn.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
```

```
        cmd.CommandText = "sp_hospital_patientinfo";
        cmd.Connection = cn;

        SqlParameter p = new SqlParameter("@pid", SqlDbType.Int);
        p.Value = pidddl.Text;
        cmd.Parameters.Add(p);

        SqlParameter p1 = new SqlParameter("@patientname",
SqlDbType.VarChar, 20);
        p1.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(p1);

        SqlParameter p2 = new SqlParameter("@age", SqlDbType.Int);
        p2.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(p2);

        SqlParameter p3 = new SqlParameter("@department",
SqlDbType.VarChar, 20);
        p3.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(p3);

        cmd.ExecuteReader();
        pntxt.Text = cmd.Parameters["@patientname"].Value.ToString();
        agtxt.Text = cmd.Parameters["@age"].Value.ToString();
        deptxt.Text = cmd.Parameters["@department"].Value.ToString();
        cn.Close();
```

```
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        cn.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "sp_hospital_surgeryinfo";
        cmd.Connection = cn;

        SqlParameter pd = new SqlParameter("@pid", SqlDbType.VarChar,
20);
        pd.Value = pidddl.SelectedItem.Text;
        cmd.Parameters.Add(pd);

        SqlParameter pname = new SqlParameter("@patientname",
SqlDbType.VarChar, 20);
        pname.Value = pntxt.Text;
        cmd.Parameters.Add(pname);

        SqlParameter age = new SqlParameter("@age", SqlDbType.Int);
        age.Value = agtxt.Text;
        cmd.Parameters.Add(age);

        SqlParameter dep = new SqlParameter("@department",
SqlDbType.VarChar, 20);
        dep.Value = deptxt.Text;
        cmd.Parameters.Add(dep);
```

```
        SqlParameter doct = new SqlParameter("@doctor",
SqlDbType.VarChar, 20);
        doct.Value = doctxt.Text;
        cmd.Parameters.Add(doct);

        SqlParameter sdate = new SqlParameter("@surgerydate",
SqlDbType.DateTime);
        sdate.Value = sdtxt.Text;
        cmd.Parameters.Add(sdate);

        cmd.ExecuteNonQuery();
        cn.Close();

        Response.Redirect("receptionhome.aspx");
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        Response.Redirect("receptionhome.aspx");
    }
```

- Referral Doctor

Pharmacy:

This module deals with all medical items. This module helps in maintaining Item Master, Receipt of Drugs/consumables, issue, handling of material return, generating retail bills, stock maintenance. It also helps in fulfilling the requirements of both IPD and OPD Pharmacy.

Laboratory:

This module enables the maintenance of investigation requests by the patient and generation of test results for the various available services, such as clinical pathology, X-ray and ultrasound tests. Requests can be made from various points, including wards, billing, sample collection and the laboratory receiving point. The laboratory module is integrated with the in-patient/ outpatient registration, wards and billing modules.

Registration:

This module helps in registering information about patients and handling both IPD and OPD patient's query. A unique ID is generated for each patient after registration. This helps in implementing customer relationship management and also maintains medical history of the patient.

typically involves; patient personal information and medical history, staff information, room and ward scheduling, staff scheduling, operating theater scheduling and various facilities waiting lists. All of this information must be managed in an efficient and cost wise fashion so that an institution's resources may be effectively utilized HMS will automate the management of the hospital making it more efficient and error free. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies.

SYSTEM DESIGN:

In this software we have developed some forms. The brief description about them is as follow:-

Reception:

The reception module handles various enquiries about the patient's admission and discharge details, bed census, and the patient's movements within the hospital. The system can also handle fixed-cost package deals for patients as well as Doctor Consultation and Scheduling, Doctor Consultancy Fees and Time Allocation.

- Doctor visit schedule
- Doctor Appointment Scheduling
- Enquiry of Patient
- Find History of Patient Enquired.

Administration:

This module handles all the master entry details for the hospital requirement such as consultation detail, doctor specialization, consultancy fee, and service charges.
Employee

- Employee Detail Recording.
- Doctor Type .
- Doctor Master

ABSTRACT

OBJECTIVE :

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information (on forms) is incomplete, or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves; patient personal information and medical history, staff information, room and ward scheduling, staff scheduling, operating theater scheduling and various facilities waiting lists. All of this information must be managed in an efficient and cost wise fashion so that an institution's resources may be effectively utilized HMS will automate the management of the hospital making it more efficient and error free. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies.

PROJECT OVERVIEW :

The Hospital Management System (HMS) is designed for Any Hospital to replace their existing manual, paper based system. The new system is to control the following information; patient information, room availability, staff and operating room schedules, and patient invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks.

A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. This information

CERTIFICATION OF APPROVAL

HOSPITAL MANAGEMENT SYSTEM

By
Ismaeel Hindawi
Mohammad Alafandi

Degree Bachelor of Engineering in
[Department of communication and networking]
