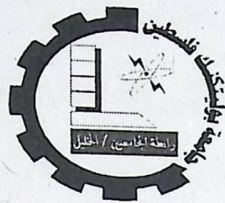


Palestine Polytechnic University
Collage of Administrative Sciences and Informatics
Department of Information Technology



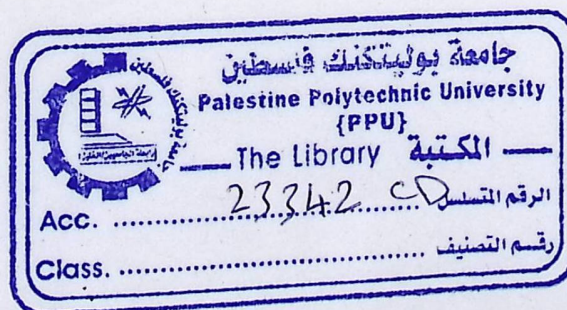
Diabetes Patient Follow up System (DPFS)

Alaa rjoub
Rihab namoura
sajida hijazi

Supervisor:
Mr. Wesam Herbawi

**A final project submitted in partial fulfillment of the requirements for the degree of
B.Sc. in Information Technology**

2009



الإهداء

إلى من أحبته الجبال فتهاقتت، و اهتزت، ثم سكنت عندما قال لها اثبتي!

إلى الأمين الصادق!

محمد صلى الله عليه وسلم!

إلى من احتضنت كل من لم يجد له أما وأبا... إلى من علا ويعلو فيها صوت الحق

رغما عن أنوف الظالمين...

إلى من شربنا ماءها، وعشقنا سماءها، وتشابكت أيدينا تعلي بناءها إلى من تحنو

علينا حنو المرضعات على الفطيم

إلى فلسطين الحبيبة

إلى من بدعائها فتحت لي أبواب السعادة

أمي الغالية

إلى من رباني فأحسن تربيتي و علمني فأمعن في تعليمي

والدي العزيز

إلى كل من شهد الوطن على إخلاصهم إلى من علت أرواحهم فداء لك يا قدس

إلى شهداءنا الأبرار

إلى الصرح الشامخ جامعة بوليتكنك فلسطين

إلى اساتذتنا الافاضل الذين علمونا ان الشمعة لا تحترق لتذوب... بل لتتير الدرب للاخرين

الشكر والتقدير

نتقدم بالشكر والامتنان أولاً لله نحمده ونستغفره ونتوب إليه
ثم كل من أعاننا على إنجاز هذا المشروع نذكرهم فنشكرهم:
إلى أساتذتنا الكرام موطن الثقة والاحترام والتقدير، اللذين علموا، فعملوا، فحققوا بذلك رضوان
الله ورسوله

وكانوا أهلاً للثناء، وأساساً للبناء، وداعماً للأبناء
والى كل الكرام الذين ساعدونا وزودونا بما مكننا من إخراج هذا البحث
والى الزملاء الذين ساهموا بكلمة أو أشاروا برأي
إن لكل منا لسان مليء بالثناء،، رطب بالدعاء،، يشكر لذي الفضل فضله،، ويردد دائماً قول
الشاعر:

وللأشكرنك ما حييت فان أمت فلتشكرنك في التراب عظامي

نخص بالشكر مشرف المشروع الاستاذ وسام الحرباوي والى الدكتور يعقوب الاشهب وفريق
مشروعه والى الاطباء والممرضين في المركز الصحي (الكرنتينا) على تعاونهم معنا في
اعطائنا المعلومات اللازمة لبناء المشروع ونشكر كل من ساهم في انجاز هذا المشروع .

ولله المنه من قبل ومن بعد

Abstract

Diabetes is one of the chronic diseases that affect a considerable portion of the society. Diabetes centres spread through out the country to track and provide cure for diabetic patients who are supposed to visit the centres at regular intervals.

Researchers, who are interested in diabetes, consider the diabetes centres as a valuable source of information for their researches. However, diabetes centres, uses manual paper-based archiving for diabetic patient visits that makes the data subject to loss and the accessibility for the data is low.

The system is built using Object Oriented Approach; Component roles within the system are clearly defined within a N-tier framework, The term N-Tier refers to the various levels of responsibility in a system's design. The N in N-tier can be any number from 2 on up.

A very common design is the 3-Tier model, In the 3-tier model the application is divided into 3 distinct tiers of responsibility, the user interface, the business logic, and the database.

Each of these tiers can be implemented using one or more objects that are dedicated to the responsibilities of that tier.

- **Presentation tier:** This is the topmost level of the application. The presentation tier displays information related to such services. It communicates with other tiers by outputting results to the screens
- **Business Logic/Logic Tier:** It controls an application's functionality by performing detailed processing
- **Database tier:** contains both Database management system and Data sets. Information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic.

In this project, the project team is trying to solve this problem by building a system to computerize the archiving of diabetic patients visits using windows application. In addition to the archiving, the system provides data visualization tools which can be used by researchers. One more feature that the system provides is the reminding service; which reminds the patient about the next scheduled visit to the centre using emails and SMS.

Future Work

- Build a web interface to enable the patient to enter/view his tests by himself.
- Add decision support system modules to the system that helps in taking decisions for doctors and researchers.

ملخص المشروع

مرض السكري من الأمراض المزمنة التي يحتاج المريض إلى مراجعة دورية لمراكز السكري المنتشرة في أنحاء الوطن. يواجه الباحثين مشكلة في تجميع البيانات عن مرضى السكري وصعوبة الوصول إليها وذلك بسبب النظام الذي يعتمد على أرشفة البيانات يدويا في مراكز السكري الذي يسبب فقد لبعض بيانات مرضى السكري. كما أن عدد المرضى المراجعين يشكل ضغط على هذه المراكز وهذا يؤدي إلى عدم أخذ كل مريض حقه في العلاج.

في هذا النظام تم استخدام طريقة برمجة الكيانات في تصميم وتطبيق هذا النظام كما استخدمنا التصميم متعدد الطبقات N-tier وهو يقسم التطبيقات إلى طبقات مستقلة، وإن العدد الصحيح لهذه الطبقات تبدأ من 2 فما فوق.

في هذا النظام تم استخدام 3 مستويات وهي:

- طبقة العرض (presentation Tier): تمثل طبقة العرض واجهة الاستخدام User Interface الرئيسية للبرنامج، والتي تكون بالعادة إما تطبيق Windows Application أو موقع Web Application ، وهي تحتوي على جميع الأجزاء المرئية التي تستخدم لتنفيذ الوظائف والأوامر المختلفة للتطبيق.
- طبقة الأعمال (Business logic Tier): تمثل طبقة الأعمال لجميع الوظائف والأوامر التي يقوم بها التطبيق، وقد تشمل أيضا القواعد والقوانين Rules الخاصة بالبرنامج.
- طبقة البيانات (Data Tier): طبقة البيانات هي الطبقة المسؤولة عن حفظ وتخزين كافة وحدات التطبيق.

يقوم نظام متابعة مرضى السكري بما يلي:

- بتخزين وتعديل وحذف والبحث عن بيانات مريض
- بتخزين وتعديل وحذف والبحث عن بيانات طبيب وممرض
- إرسال رسالة قصيرة وإيميل للمريض لتذكيره بموعد المراجعة التالية
- يزود المستخدم برسوم بيانية تساعد أكثر في اتخاذ القرار المناسب
- يزود المستخدم بتقارير عن حالة المريض

مستقبلا يمكن تطوير هذا النظام ليصبح موقع ويب يستطيع من خلاله المريض متابعة حالته الصحية مع الطبيب كما يستطيع الطبيب أو الممرض الوصول إلى بيانات المرضى من خارج مركز السكري

Project content

In this document we use RUP standard .we prefer to submit each document independently.

RUP: It is abbreviation of Rational Unified Process
It's a software engineering process, aimed at guiding software development organizations in their endeavors [IBM]

For clarity these documents are:

- Vision document
- Planning document
- Software requirement specification (SRS) document.
- Design document
- Testing document

Palestine Polytechnic University

**Diabetes Patients Follow up System
Vision Document**

Version <1.2>

Diabetes Patients Follow up System	Version: <1.2>
vision document	Date: <18/3/2009>

Revision History

Date	Version	Description	Author
<6/3/2009>	<1.0>	<first version of vision documentation >	Alaa rjoub
<10/3/2009>	<1.1>	<More details added>	Alaa rjoub
<18/3/2009>	<1.2>	<adding definitions ,abbreviation and references>	Alaa rjoub

Diabetes Patients Follow up System	Version: <1.2>
vision document	Date: <18/3/2009>

Table of Contents

1. Introduction	5
1.1 Purpose	5
1.2 Scope	5
1.3 Definitions, Acronyms, and Abbreviations	5
1.4 References	5
1.5 Overview	6
2. Positioning	6
2.1 Problem Statement	6
2.2 Product Position Statement:	7
3. Stakeholder and User Descriptions	7
3.1 Stakeholder Summary	7
3.2 User Summary	8
3.3 User Environment	9
3.4 Summary of Key Stakeholder or User Needs	9
3.5 Alternatives and Competition	9
3.5.1 System alternatives	9
3.5.2 Software alternatives:	10
4. Product Overview	11
4.1 Product Perspective	11
4.2 Assumptions and Dependencies	11
5. Product Features	11

Diabetes Patients Follow up System	Version: <1.2>
vision document	Date: <18/3/2009>

List of tables

Table 1: problem that solved by this project	6
Table 2: positioning for the final product.....	7
Table 3: stakeholder summary	8
Table 4: the users of the system	8

Diabetes Patients Follow up System	Version: <1.2>
vision document	Date: <18/3/2009>

1. Introduction

1.1 Purpose

The purpose of this document is to collect, analyze, and define high-level needs and features of the Diabetes Patients Follow up System (DPFS). It focuses on the capabilities needed by the stakeholders and the target users, and why these needs exist.

1.2 Scope

This Vision Document applies to DPFS which will be developed by the graduation project team. The development team will develop this system to solve the problems of the existing system, traditional paper system that is used in health centers. Health centers need this system to arrange patient files in computerized way to prevent data loss and to increase data accessibility.

1.3 Definitions, Acronyms, and Abbreviations

DPFS: Diabetes Patients Follow up System.

SMS: Short Message Service

Lab-test: tests that the patient makes them in lab.

Doctor-test: tests that made only by doctor in the clinic.

Physical test: tests that made by the doctor inside the clinic such that: examine the patient eyes and knees.

GSM: Global System for Mobile Communications

1.4 References

Thomas Ana, 2008, advantages and disadvantages of java and SQL, http://www.geekinterview.com/question_details/31111

IBM Software Group, 2004, RUP Vision Document, <http://www.lsi.upc.es/events/recots/04/UTDCS-17-04.pdf>

Freetutes.com, 2008, Role of System Analyst, weak Reference-manage to changre, <http://www.freetutes.com/systemanalysis/role-of-system-analyst.html>

FOLDOC, 2009, GSM, <http://encyclopedia2.thefreedictionary.com/GSM>

KUDOS, 2007, Advantages of using vb.net, http://www.kudos-india.com/technology/vb_dot_net1.htm// weak Reference-manage to changre

AdvantagesofUsingOracle,2006,<http://www.learn.geekinterview.com/databas e/oracle/advantages-of-using-oracle.html>// weak Reference-manage to changre

Diabetes Patients Follow up System	Version: <1.2>
vision document	Date: <18/3/2009>

Ruggieri Denise, 2009, Advantages of Microsoft Access, // weak Reference-
 manage to change http://www.ehow.com/about_4760017_advantages-microsoft-access.html

1.5 Overview

This document contains problem statement, Product Position Statement stakeholder, and user description for the system, alternative and competition, Product Perspective, Assumptions and Dependencies and Product Features

2. Positioning

2.1 Problem Statement

The problem of	loss of information of diabetes patients in public health centers
affects	diabetes patients and health centers
the impact of which is	Difficult to access patients' information to make decisions or to search about a patients' illness history
a successful solution would be	Make a computerized system that stores patients' information and make it easy to access the information

Table 1: problem that solved by this project

Diabetes Patients Follow up System	Version: <1.2>
vision document	Date: <18/3/2009>

2.2 Product Position Statement:

For	Diabetes centers
Who	Need data archiving
The DPFS	is an archiving software with data visualization and patients reminding facilities
That	provides the ability to save, update, delete, and search diabetes patient's data, provides different data representation techniques like reporting and data visualization, provides patients reminding through emails and SMS
Unlike	Current manual paper archiving system that makes data subject to loss and data access is hard. The current manual system does not provide data representation other than independent documents. No reminding services for diabetes patients
Our product	Is computerized

Table 2: positioning for the final product

3. Stakeholder and User Descriptions

This system can be used by doctors and nurses to create and update patient's files, view reports and visualized patients data. The system might be upgraded to provide a web interface to be used by patients. There are a number of stakeholders with an interest in the development and not end users

3.1 Stakeholder Summary

Name	Responsibilities
System Analyst	describing problems <ul style="list-style-type: none"> ✓ Investigates such problems and opportunities to determine the feasibility of a system solution and to identify the general kinds of system solution that appear appropriate
Requirements Specifier	<ul style="list-style-type: none"> ✓ Detailing business requirements and provides functional and non-functional requirements

Diabetes Patients Follow up System	Version: <1.2>
vision document	Date: <18/3/2009>

System Architect	<ul style="list-style-type: none"> ✓ Promote architectural goals such as expandability. ✓ Understand the (business) problem domain, identifying requirements and constraints. ✓ Understand technological possibilities. <p>Understand technological limits</p>
Programmers	<ul style="list-style-type: none"> ✓ Determine the programming needs for the system ✓ Meet the client's needs for programming by creating programs

Table 3: stakeholder summary

3.2 User Summary

Name	Responsibilities
Doctor	<ul style="list-style-type: none"> ✓ Update patient file to store doctor-test results ✓ View previous tests ✓ View reports and View charts
Nurse	<ul style="list-style-type: none"> ✓ Create, update, delete patient file ✓ Store lab-test results <p>Set next visit date and time patient file</p>
Administrator	Add, delete, update doctor or nurse

Table 4: the users of the system

Diabetes Patients Follow up System	Version: <1.2>
vision document	Date: <18/3/2009>

3.3 User Environment

Two users can complete the task; doctor, and nurse, this may change later to add the patient himself to participate to accomplish the task. The task cycle consists of storing lab-tests and physical-tests by the nurse followed by storing doctor-tests by the doctor. This cycle might change to let the nurse perform the whole cycle. The existing platform is windows platform and the end users are familiar with this platform, it's expected that this platform will remain for a long period. No internet connectivity is available at the time of this document. No existing computerized systems are present at the user environment so no need for integration

3.4 Summary of Key Stakeholder or User Needs

The existing system used now in health centers depends on papers and files with handwritten entries which might not be for other persons. Existing system might cause data loss and data redundancy. The solution is to make a computerized system that store the information about patients and staff in a database that has no redundancy and no data loss

3.5 Alternatives and Competition

3.5.1 System alternatives

The alternatives to apply the functions that the system must perform might be:

- ✓ Building the system as desktop application.
- ✓ Building the system on the web.

3.5.1.1 Building the system as desktop application

Advantages:

- ✓ High response.
- ✓ More security and privacy.
- ✓ Easy to deal with.

Disadvantages:

- ✓ Hard to access from anywhere

3.5.1.2 Building the system on the web

Advantages:

- ✓ Doctors can easily following up patients from anywhere.

Diabetes Patients Follow up System	Version: <1.2>
vision document	Date: <18/3/2009>

- ✓ The flexibility to update the system requirements after operating the system.

Disadvantages:

- ✓ There is no internet connectivity in the diabetes centers.
- ✓ The workers don't know how to use the internet.
- ✓ Data is subject to access from outsiders.

Based on the previous comparison, windows application is more suitable to serve the diabetes centers and could be updated later on to have web facilities.

3.5.2 Software alternatives:

Following are three software alternatives to build the desktop application:

- ✓ Database programming using Oracle
- ✓ Database programming using Microsoft Access
- ✓ Visual basic windows application with SQL server.

3.5.2.1 Database programming using Oracle

Advantages:

- ✓ high productivity

Disadvantages:

- ✓ Complex system screens
- ✓ Used to handle very huge data which is not needed in this system.

3.5.2.2 Microsoft Access

Advantages:

- ✓ Easy to deal with.
- ✓ high productivity

Disadvantages:

- ✓ Used to handle small amount of data which might not serve the system

Diabetes Patients Follow up System	Version: <1.2>
vision document	Date: <18/3/2009>

3.5.2.3 Visual Basic windows application with SQL server

Advantages:

- ✓ produce simple screens
- ✓ SQL server provides intermediate storage capability between Oracle and Microsoft Access.

Disadvantages:

- ✓ less productivity than the other alternatives

VB.NET windows application with SQL server is chosen, because it meets the system requirements.

4. Product Overview

This section provides a high level view of the product capabilities, interfaces to other applications, and system configurations. This section consists of two subsections, Product perspective and Assumptions and dependencies

4.1 Product Perspective

This system is undependable of other systems and it is easy to use, the interface is friendly and comfortable to its users.

4.2 Assumptions and Dependencies

This system works in windows platform and need some special hardware to provide full functionality like GSM modem or mobile phone to send SMS to patients.

5. Product Features

- ✓ This system store staff and patient data in a database
- ✓ The capability to add, delete , update, search about patient this done by nurse or doctor
- ✓ The capability to add, delete, update, search about staff user(doctor and nurse)this done by administrator
- ✓ The ability to view patient reports through this system
- ✓ The ability to alert the patient in SMS and email

Palestine Polytechnic University

Diabetes Patients Follow up System Software Development Plan

Version <1.1>

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

Revision History

Date	Version	Description	Author
4/2009	1.0	First version of planning document	rihab numora
4/2009	1.1	Updating the document by adding more detailed information	Alaa rjoub

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

Revision History

Date	Version	Description	Author
4/2009	1.0	First version of planning document	rihab numora
4/2009	1.1	Updating the document by adding more detailed information	Alaa rjoub

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

Table of Contents

1.	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	Definitions, abbreviations	5
1.4	References	5
2.	Project Overview	6
2.1	Project Purpose, Scope, and Objectives	6
2.1	Assumptions and Constraints	6
3.	Project Organization	6
3.1	Organizational Structure	6
3.2	External Interfaces	7
3.3	Roles and Responsibilities	7
4.	Management Process	7
4.1	Project Estimates	7
	Printer HP LaserJet M5035 MFP (Q7829A)	9
4.2	Project Plan	10
4.2.1	Project Time Scheduling	10
4.2.2	Project Resourcing	11
4.3	Project Monitoring and Control	11
4.3.1	Requirements Management Plan	11
4.3.2	Schedule Control Plan	12
4.3.3	Quality Control Plan	12
4.3.4	Reporting Plan	12
4.4	Risk Management Plan	12

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

List of tables

Table 1 : cost of Hardware resources in Development stage	7
Table 2: Cost of software resources in Development stage.....	8
Table 3: Cost of human resources in Development stage	8
Table 4: Total development cost.....	8
Table 5 :Cost of hardware resources in operating stage.....	9
Table 6: Cost of software resources in operating stage.....	10
Table 7: Cost of human resources in operating stage.....	10
Table 8: Total operating cost.....	10
Table 9: Total project cost.....	10

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

1. Introduction

1.1 Purpose

The purpose of this document is to plan to the system and clarifies planning process for the project and defines the resources that might be needed to develop the system

1.2 Scope

This software development plan applies to DPFS that will be built by the graduation project team. In this document, the project team will provide the planning for the project including estimates, resources, and management planning. The rest of this document is structured as follows; section 2 provides a project overview followed by project organization in section 3 and finally section 4 provides the project management process

1.3 Definitions, abbreviations

DPFS: Diabetes Patients Follow up System.

1.4 References

- James R., 1997, Working with Microsoft Project, http://www.hyperhot.com/pm_msp1.htm
- Barry W. Boehm and others ,2000,what is RUP, http://www.augustana.ab.ca/~mohrj/courses/2000.winter/csc220/papers/rup_best_practices/rup_bestpractices.html#1
- Rational SoftwareCorporation , 1998, Rational Unified ProcessBest Practices for SoftwareDevelopment Teams, http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

- Russell Norlund, 2005, integrating the Rational Unified Process with Managing Successful Programmes, <http://www.ibm.com/developerworks/rational/library/jun05/norlud/>

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The proposed system intends to:

- ✓ Computerize the archiving of the diabetes patient's data.
- ✓ Provide a graphical representation of the patient illness history
- ✓ Provide SMS service to remind the patient about the review.
- ✓ To facilitate dealing with the patient's record.
- ✓ Reduce the time and the effort in recalling and saving the patients data

2.1 Assumptions and Constraints

There are many constraints forced on building the system:

- ✓ Build the system in a short period does not exceed 14 weeks.
- ✓ Defining the system requirements needs to visit the diabetes center that requires time and workers collaboration.
- ✓ The system might face change resistance from the existing diabetes centers staff.
- ✓ The shortage in the diabetes center resources to computerize the system.
- ✓ The shortage in the workers skills.

3. Project Organization

3.1 Organizational Structure

The project team is not that big to enforce a rigid structure, so the project team is informally structured

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

3.2 External Interfaces

This project is not part of larger project, so no interface with any other group or project is available

3.3 Roles and Responsibilities

The whole team will work on one task at a time but the work will be distributed within the task.

4. Management Process

4.1 Project Estimates

This part explains the cost and the required resources for the development and operating the system which includes

- ✓ Hardware resources
- ✓ Software resources
- ✓ Human resources

4.1.1 Development Cost

4.1.1.1 Hardware resources

Hardware resources	Number of units	Unit cost	Total
HP Computer Core Duo T2600 / 2.16 GHz Centrino Duo RAM 1 GB HDD 120 GB DVD±RW (+R double layer) / DVD-RAM WLAN : 802.11a/b/g	2	600\$	1200\$
Flash memory 1GB	3	5\$	15\$
Total cost			1215\$

Table 1 : Cost of Hardware resources in Development stage

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

4.1.1.2 Software resources

Software resource	Number of units	Unit cost	Total
Windows XP	1	40\$	40\$
PhotoShop	1	70\$	70\$
Microsoft Visual Studio.Net 2005	1	0	0
SQL server 2000	1	0	0
Microsoft Office 2003	1	160\$	160\$
Total cost			270\$

Table 2: Cost of software resources in Development stage

4.1.1.3 Human resources

Human resource	Number of workers	Cost/month	Total
Designer	1	700\$	700\$
DataBase developer	1	700\$	700\$
Programmer	1	700\$	700\$
Total cost			2100*4=8400

Table 3: Cost of human resources in Development stage

4.1.1.4 Total development cost:

Hardware resource cost	Software resource cost	Human resource cost	Total cost
1215\$	270\$	8400\$	9885\$

Table 4: Total development cost

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

4.1.2 Operating Cost

4.1.2.1 Hardware resources

hardware resource	Number of units	Unit cost	Total
Server Core Duo T2600 / 2.16 GHz Centrino Duo RAM 1 GB HDD 120 GB DVD±RW (+R double layer) / DVD-RAM WLAN : 802.11a/b/g	1	600\$	600\$
Printer HP LaserJet M5035 MFP (Q7829A)	1	250\$	250\$
Desktop Core Duo T2600 / 2.16 GHz Centrino Duo RAM 1 GB HDD 120 GB DVD±RW (+R double layer) / DVD-RAM WLAN : 802.11a/b/g	1	600\$	600\$
Total cost			1450\$

Table 5 :Cost of hardware resources in operating stage

4.1.2.2 Software resources

Software resource	Number of units	Unit cost	Total
Windows XP	1	40\$	40\$

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

.net frame work	1	0\$	0\$
SQL server 2000	1	0\$	0\$
Total cost			40\$

Table 6: Cost of software resources in operating stage

4.1.2.3 Human resources

Human resource	Number of workers	Cost/month	Total cost per year
Worker	2	500\$	1000\$*12
Total cost			12000\$

Table 7: Cost of human resources in operating stage

4.1.2.4 The total operating cost:

Hardware resource cost	Software resource cost	Human resource cost	Total cost
1450\$	40\$	12000\$	13490\$

Table 8: Total operating cost

4.1.2.5 The overall cost of the project:

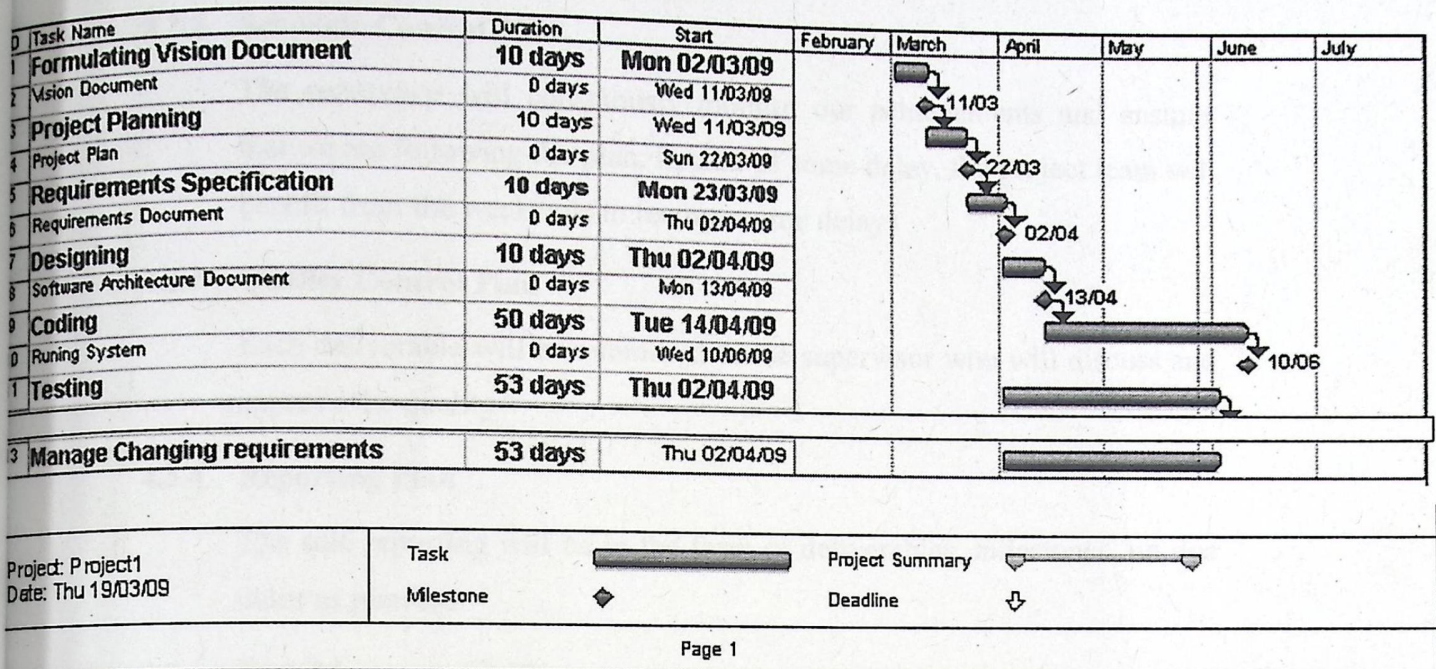
The development cost	The operating cost	Total sum
9885\$	13490\$	23375\$

Table 9: Total project cost

4.2 Project Plan

4.2.1 Project Time Scheduling

Diabetes Patients Follow up System		Version: <1.1>
Software Development Plane		Date: <7/4/2009>



4.2.2 Project Resourcing

4.2.2.1 Staffing Plan

The project development team consists of three software developers

4.2.2.2 Resource Acquisition Plan

The staff is selected straightforward as they are a graduation project team

4.2.2.3 Training Plan

No training is needed

4.3 Project Monitoring and Control

4.3.1 Requirements Management Plan

The first step after finalizing the plan will be defining and specifying requirements. Any newly emerging requirement will be incorporated in the system either by creating new classes or by modifying existing ones

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

4.3.2 Schedule Control Plan

The supervisor will consciously monitor our achievements and ensures that we are following the plan. In case of some delay, the project team will benefit from the weekends to recover some delays

4.3.3 Quality Control Plan

Each deliverable will be submitted to the supervisor who will discuss and approve the quality

4.3.4 Reporting Plan

The sole reporting will be in the form of deliverables, milestones, on due dates as planned

4.4 Risk Management Plan

Risks:

- ✓ lose the program during the work as result of system crash
- ✓ New requirement appear after development stage Change the requirement during the working
- ✓ Fear of deliver the project late and The required time to improve the system more than available
- ✓ the team may not have the skills for programming the system
- ✓ The diabetes center rejects the system.

Risks solution:

- ✓ Backup.
- ✓ Collect the detailed information to define the requirement accurately.
- ✓ Assigning tasks and dividing it to all project team.
- ✓ Collect the resources to improve team project skills
- ✓ Display the advantage of this computerize system over the traditional system that is used in diabetes center

Diabetes Patients Follow up System	Version: <1.1>
Software Development Plane	Date: <7/4/2009>

4.3.2 Schedule Control Plan

The supervisor will consciously monitor our achievements and ensures that we are following the plan. In case of some delay, the project team will benefit from the weekends to recover some delays

4.3.3 Quality Control Plan

Each deliverable will be submitted to the supervisor who will discuss and approve the quality

4.3.4 Reporting Plan

The sole reporting will be in the form of deliverables, milestones, on due dates as planned

4.4 Risk Management Plan

Risks:

- ✓ lose the program during the work as result of system crash
- ✓ New requirement appear after development stage Change the requirement during the working
- ✓ Fear of deliver the project late and The required time to improve the system more than available
- ✓ the team may not have the skills for programming the system
- ✓ The diabetes center rejects the system.

Risks solution:

- ✓ Backup.
- ✓ Collect the detailed information to define the requirement accurately.
- ✓ Assigning tasks and dividing it to all project team.
- ✓ Collect the resources to improve team project skills
- ✓ Display the advantage of this computerize system over the traditional system that is used in diabetes center

Palestine Polytechnic University

Diabetes Patients Follow up System Software Requirements Specification

Version <1.2>

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

Revision History

Date	Version	Description	
17/3/2009	<1.0>	First requirement specifications	Sajida hijazi
2/4/2009	<1.1>	More requirement added	Rehab namora
5/5/2009	<1.2>	More requirement added	Alaa rjoub

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

Table of Contents

1.	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	Abbreviations	5
1.4	References	5
1.5	Overview	5
2.	Overall Description	6
3.	Specific Requirements	6
3.1	Functionality	6
3.1.1	Use cases:	6
3.1.2	Use-Case Name: Login	7
3.1.2	Use-Case Name: Add treatment	8
3.1.3	Use-Case Name: Patient Administration	9
3.1.4	Use-Case Name: Add Appointments	9
3.1.5	Uses-Case Name: View Treatments	10
3.1.6	Use-Case Name: Insert experimental results	11
3.1.8	Uses-Case Name: handle alert use case	11
3.1.9	Uses-Case Name: View Report	12
3.2.1	Usability:	13
3.2.2	Reliability	13
3.2.4	Supportability	13
3.2.5	Design Constraints	13
3.2.6	Purchased Components	14
3.2.7	Interfaces	14
3.2.7.1	User Interfaces	14
3.2.7.2	Hardware Interfaces	14

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

Tables of figures

Figure 1: login use case.....	7
Figure 2: add treatment use case	8
Figure 3: Patient administration	9
Figure 4: Add appointment use case	9
Figure 5: View treatment use case	10
Figure 6: Insert experimental results use case.....	11

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

1. Introduction

1.1 Purpose

This Software Requirements Specification document (SRS) provides a complete description of all the functions and specifications of the DPFS.

The expected audience of this document is the faculty of Administrative science and Informatics; including Mr. Wesam Herbawi the project supervisor and the projects evaluation committee. The staff of the diabetes centers that will use this system also is an expected audience of this document in addition to the project team as supposed to be the designers of the project

1.2 Scope

The DPFS is designed to run as a desktop application to allow diabetes centers to fill diabetes patient's information on a database server to allow handling patient's data in a computerized way, create illness history charts, and send reminding SMS and emails.

1.3 Abbreviations

SRS: Software Requirements Specification
DPFS: Diabetes Patients Follow up System

1.4 References

- Barry W. Boehm and others ,2000 , What Is the Rational Unified Process, http://www.augustana.ab.ca/~mohrj/courses/2000.winter/csc220/papers/rup_best_practices/rup_bestpractices.html
- Rational Website: www.rational.com, rational rose tutorials , <http://emhain.wit.ie/~mmcmahon/CC3/HOut2.pdf>
- Russell Norlund, 2005, integrating the Rational Unified Process with Managing Successful Programmes, <http://www.ibm.com/developerworks/rational/library/jun05/norlund/>

1.5 Overview

The rest of the document is as follow, section 2 provides an overall description of the system followed by section 3 which specifies functional and non-functional requirements.

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

2. Overall Description

The DPFS stores patient's information in a database and allow processing it, makes reminding Short Message Service (SMS), and reminding emails. This system will be a desktop application, linked with database server. An Internet connection is optional.

3. Specific Requirements

3.1 Functionality

Functional requirements are those that refer to the functionality of the system, i.e., what services it will provide to the user. Nonfunctional requirement pertain to other information needed to produce the correct system and are detailed separately. The functional requirements will be defined in the form of use cases in the subsequent subsections

3.1.1 Use cases:

General scenario for Diabetes Patients Follow up System

The users of the system are doctors and secretaries. Any user of the system should login to the system before using it by providing his/her user name and password.

The doctor can chose to enter a prescription for a specific patient. When the doctor takes this selection, the system will display a screen to enable him/her to provide the patient identifier and the details of the prescription.

Another thing the doctor can do is to view a patient history report. The doctor selects to view patient history report, and then the system will display a screen asking for the patient identifier. After that, a textual report will be displayed describing the patient status over a period of time.

One more thing the doctor can do is to view the patient status history in a visualized form. The doctor selects to view patient history chart, and then the system will display a screen asking for the patient identifier. After that, a chart will be displayed describing the patient status over a period of time.

The secretary can insert patient's experimental results, made in medical labs to the system. He/She can do this by selecting the add experimental results, the system will display a screen that enables the secretary to add the patient's identifier and experimental results.

The secretary also can add appointment. The diabetic patients must visit the diabetes centers on regular intervals; hence the system enables the secretary to add appointment indicating the time of the next visit. When the secretary chooses

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

to add appointment, the system will display a screen to asking for the patient identifier and the date and time of the next visit.

In addition to the aforementioned use cases, the secretary can view the prescription provided by the doctor. When choosing to view the treatment, the system will ask for the patient identifier to view the prescription and then provide a list of all prescriptions provided to the patient. The secretary then selects one of the prescriptions based on the required date.

The last use case of the system is the patient reminding. After the secretary adds an appointment, the system will automatically sends a reminding message for the patient two days before the date of the appointment. The reminding message will have two form; email and SMS

3.1.2 Use-Case Name: Login

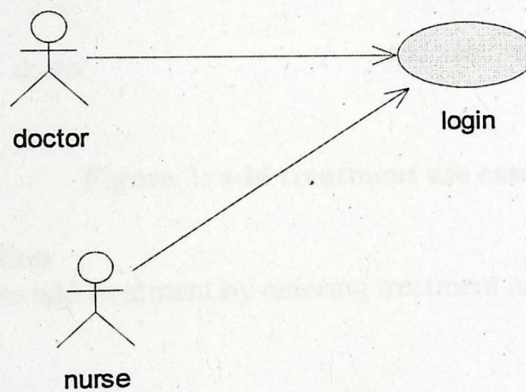


Figure 1: login use case

1. Brief Description

This use case is used by DPFS user to login to the DPFS to get roles.

2. Actors

Doctor and Nurse.

3. Flow of Events

3.1 Basic Flow

- The user chooses login to the system
- The system asks for user name and password
- The user enters username and password
- The system compare the entered username and password against already defined ones

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

- If the entered username and password are corrected, the user is given access to the DPFS.
- If the entered username and/or password are incorrect, error message will display.

4. Special Requirements

Login must be safety and fast process and store login information in login file

5. Precondition

Syntactically valid username and password

6. Post condition

User is authenticated

3.1.2 Use-Case Name: Add treatment

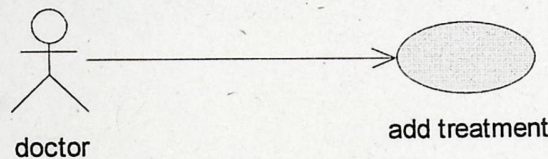


Figure 2: add treatment use case

1. Brief Description

Enables doctor to add treatment by entering treatment name, description, and the date.

2. Actors

Doctor .

3. Flow of Events

3.1 Basic Flow

- Doctor login to his account.
- Doctor select patient file by typing patient file number.
- System find the patient record
- Doctor select the appropriate treatment
- The system display treatments to the patient report include description and date

4. Special Requirements

- Use treatment name in medical prescription
- Store the name of treatment to write it in medical prescription before print the prescription.

5. Precondition

Doctor must be logged on to his account and logged on to patient account.

6. Post condition

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

The detail of Treatment stored in patient file and doctor can print medical prescription.

3.1.3 Use-Case Name: Patient Administration

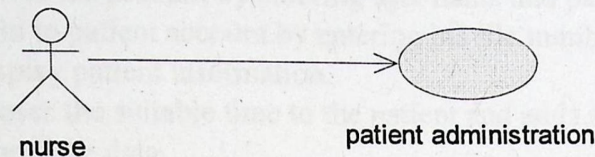


Figure 3: Patient administration

1. Brief Description

Enables nurse to administrate patient files.

2. Actors

Nurse

3. Flow of Events

3.1 Basic Flow

- Nurse login to his account.
- Nurse select the specific patient by his file number
- System show patient record
- Nurse select the appropriate action, he can add patient or change the classification of the patient.
- System will save the changes

5. Precondition

Nurse must be logged on and have the authorization

6. Post condition

Every patient set to a specific classification regarding to his status

3.1.4 Use-Case Name: Add Appointments

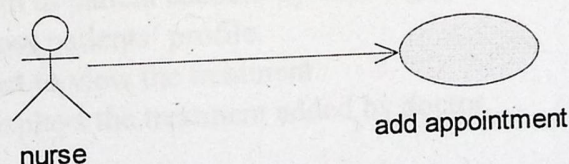


Figure 4: Add appointment use case

1. Brief Description

Enables nurse to add appointment for the next visit.

2. Actors

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

Nurse.

3. Flow of Events

3.1 Basic Flow

- Nurse login to his account by entering user name and password.
- Nurse login to patient account by entering his file number
- System display patient information
- Nurse chooses the suitable time to the patient and adds appointment.
- System stores the data.

5. Precondition

Nurse must be logged on to his account and logged on to patient account.

6. Post condition

Appointment saved and used in handling alert.

3.1.5 Uses-Case Name: View Treatments

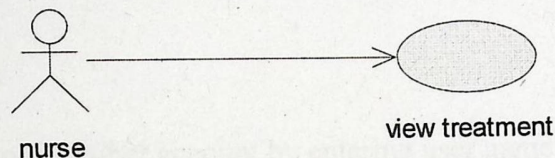


Figure 5: View treatment use case

1. Brief Description

This use case enables nurse to view patient treatment that doctor added to patients.

2. Actors

Nurse.

3. Flow of Events

3.1 Basic Flow

- Nurse login to his/her account by entering user name and password.
- Nurse login to patient account by entering patients' file number
- System show patients' profile
- Nurse select to view the treatment
- System displays the treatment added by doctor.

4. Special Requirements

View treatment must be consistent with treatment doctor add.

5. Precondition

Nurse must be logged on to his/her account and logged on to patients' account and treatment must be added by doctor.

6. Post condition

Nothing

3.1.6 Use-Case Name: Insert experimental results

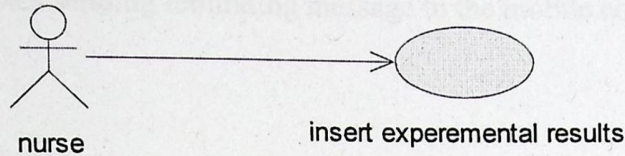


Figure 6: Insert experimental results use case

1. Brief Description

The user selects a specific patient in system, selects lab result from this patient's record and reads specific lab result based on test purpose

2. Actors

Nurse.

3. Flow of Events

3.1 Basic Flow

- Nurse login to his/her account by entering user name and password.
- Nurse selects a specific patient that he/she is in charge of
- System finds the patient record
- Nurse selects Lab Reports in the record
- System displays a list of lab result of this patient
- Then insert experimental results for the patient in the specified location..

5. Precondition

Nurse must be logged on to his account and logged on to patient account.

6. Post condition

Experimental results stored in database and processed to asses users decisions.

3.1.8 Uses-Case Name: handle alert use case

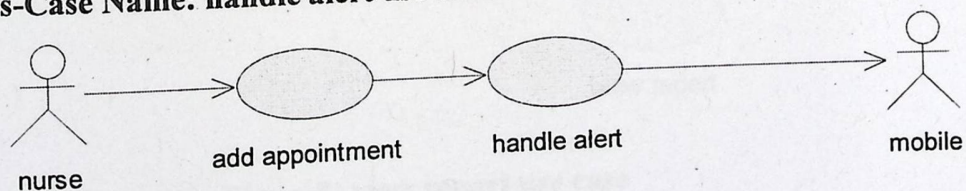


Figure 7: Handle alert use case

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

1. Brief Description

This use case enables sending reminding message to the mobile connected with system.

2. Actors

Nurse .

3. Flow of Events

3.1 Basic Flow

- Nurse add appointment
- System save appointment
- System sends the appointment for handling alert(alert may be email or SMS or both)
- Before tow days of the visit handle alert sends message to mobile

4. Special Requirements

Patient mobile number stored in the database.

5. Precondition

Nurse adds appointment.

6. Post condition

Reminding message sends to mobile

3.1.9 Uses-Case Name: View Report

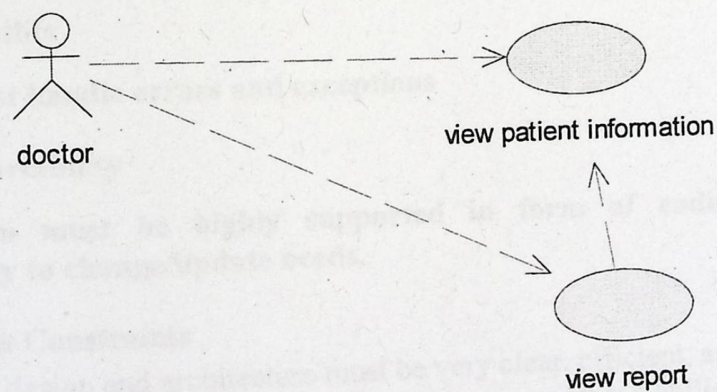


Figure8: view report use case

1. Brief Description

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

Enable to read reports from a specific patient's record.

2. Actors

3. Flow of Events

3.1 Basic Flow

- Doctor logs in
- System presents Doctor's patient list
- Doctor selects patient
- System shows patient page
- Doctor selects 'Reports'
- System shows all reports
- Doctor selects report
- System displays report

5. Precondition

Doctor Account must exist, Patient Account must exist

6. Post condition

None

3.2 Non functional requirements

3.2.1 Usability:

The system must provide friendly interfaces and easy to be used across all its functions

3.2.2 Reliability

System must handle errors and exceptions

3.2.4 Supportability

The system must be highly supported in form of coding styles and responsively to change/update needs.

3.2.5 Design Constraints

The system design and architecture must be very clear, efficient, and the code must be well structured. The chosen design must support scalability.

3.2.6 Purchased Components

DXperience for charting

Diabetes Patients Follow up System	Version: <1.2
Software Requirements Specification	Date: <5/5/2009

3.2.7 Interfaces

3.2.7.1 User Interfaces

We used graphical user interface not command interface to be easy of use

3.2.7.2 Hardware Interfaces

We will use IP address to communicate with DB server

3.2.7.3 Communication interface

Local area network

Appendix

Appendix

محاضر الجلسات

محضر 1:

التاريخ	10\3\2009 م
اليوم	الثلاثاء
الوقت	من الساعة التاسعة صباحا الى الساعة العاشرة والنصف صباحا
المكان	مركز السكري(الكرنتينا)
الحضور	فريق المشروع(رحاب النمورة, الاء رجوب وساجدة حجازي) والمرضة ابتسام الطيبي والموظفة فياز الجعبة
الموضوع	الحصول على معلومات حول مرضى السكري
	<ul style="list-style-type: none"> • تم التعرف على البيانات التي يتم تسجيلها للمريض في اول زيارة له لمركز السكري • تم التعرف على البيانات التي يسجلها الطبيب في كل زيارة للمريض • تم التعرف على البيانات التي تسجلها الممرضة • تم مناقشة امكانية تطبيق المشروع على هيئة ويب او ويندوز • تم التعرف على الفحوصات التي تجري بشكل دوري للمريض كل ست شهور

محضر 2:

التاريخ	2009\3\12 م
اليوم	الخميس
الوقت	من الساعة الثامنة صباحا الى الساعة التاسعة والنصف صباحا
المكان	مركز السكري(الكرنتينا)
الحضور	فريق المشروع(رحاب النمورة, الاء رجوب وساجدة حجازي)و الدكتور نعيم الصرصور والموظفة فياز الجعبة
الموضوع	الحصول على معلومات حول مرضى السكري
	<ul style="list-style-type: none">• تم التعرف على البيانات التي يتم تسجيلها للمريض عند تحويله من العيادات الخارجية الى مركز السكري• تم التعرف على البيانات التي يتم تسجيلها للمريض عند تحويله من مركز السكري الى المستشفى• تم الحصول على نموذج للاستيبيان الذي يتم تعبئته لمريض السكري

محضر 3:

التاريخ	2009\3\15 م
اليوم	الاحد
الوقت	من الساعة الثانية عشر صباحا الى الساعة الواحدة والنصف صباحا
المكان	مركز السكري(الكرنتينا)
الحضور	فريق المشروع(رحاب النمورة,الاء رجوب وساجدة حجازي) والدكتور عارف ابو ارميلة والمرضة ابتسام الطيبي والموظفة فياز الجعبة
الموضوع	الحصول على معلومات حول مرضى السكري
	<ul style="list-style-type: none"> • تم التعرف على الفحوصات التي يقوم الطبيب باجرائها لمريض السكري ويتم تسجيلها • تم التعرف على الفحوصات التي تقوم الممرضة باجرائها لمريض السكري ويتم تسجيلها • تم التعرف على انواع مرض السكري • تم التعرف على انواع علاج مرض السكري • تم التعرف على الفحوصات المخبرية المهمة

Palestine Polytechnic University

**Diabetes Patients Follow up System
Software Architecture Document**

Version <1.1>

Diabetes Patients Follow up System	Version: <1.1>
Software Architecture Document	Date: <10/5/2009>

Revision History

Date	Version	Description	Author
/2009	<1.0>	First version of design document	Rihab namuora
5/2009	<1.1>	Detailed version of design document	Alaa rjoub

Table of Contents

1. Introduction		6
1.1 Purpose		6
1.3 References		6
1.4 Abbreviations		6
2. Architectural Representation		7
3. Architectural Goals and Constraints		7
4. Use-Case View		7
4.1 Significant Use Case Descriptions		8
4.1.1 Login	8	
4.1.2 Add Treatment	8	
4.1.3 View Reports	8	
4.1.4 Insert experimental results	9	
4.1.5 Add Appointments	9	
4.1.6 View Treatments	9	
4.1.7 Send Alert	9	
4.2 Use-Case Realizations		10
5. Logical View		11
5.1 Overview		11
5.2 Architecturally Significant Design Packages	12	
5.1.1 Presentation tier:		16
5.1.2 Business Logic/Logic Tier	26	
5.1.4.1 Database design tables		36
6. Deployment View		37
6.1 Server		37
6.2 DPFS Application		37
6.3 GSM modem		38
7. Implementation View		38
7.1 Overview		38
7.2 Layers		38
7.2.1 Client layer		38
7.2.2 Server layer		38
8. Size and Performance		38
9. Quality		

List of tables

Table 1: staff user table	26
Table 2: city table.....	27
Table 3: degree table	27
Table 4: disease table	27
Table 5: Dr Monthly test table	28
Table 6: Education level table.....	28
Table 7: Martial status table.....	29
Table 8: NUmonthly test.....	30
Table 9: patient table.....	31
Table 10: Physical activity table	31
Table 11: region	31
Table 12: Role.....	32
Table 13: RS table.....	33
Table 14: Six month center test.....	34
Table 15: Six month lab test.....	35
Table 16: staff martial status	36

Table of figures

Figure 1: Architectural Use-cases	10
Figure 2: Logical View	12
Figure 3: login screen.....	12
Figure 4: main doctor screen.....	13
Figure 5: add Patient screen	14
Figure 6: Patient search screen.....	15
Figure 7: delete patient screen.....	15
Figure 8: class diagram for the system.....	16
Figure 9: system-user class.....	17
Figure 10: patient class.....	18
Figure 11: staff user class.....	18
Figure 12: Relativness history class	19
Figure 13: Tests class.....	19
Figure 14: Doctor monthly tests.....	20
Figure 15: Six month lab test	21
Figure 16: Six month center test	22
Figure 17: nurse month lab test.....	23
Figure 18: Alarm class	23
Figure 19: SMS alarm.....	24
Figure 20: Email alarm.....	24
Figure 21: database diagram	25

Diabetes Patients Follow up System	Version: <1.1>
Software Architecture Document	Date: <10/5/2009>

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

This Software Architecture Document provides an architectural overview of the DPFS. The DPFS is being developed to help the doctors and patient in the diabetes center.

1.3 References

- Tony Marston, 2002, the 3-Tier Architecture <http://www.tonymarston.net/uniface/3tiercriticisms.html>
- Barry W. Boehm and others ,2000 , What Is the Rational Unified Process, http://www.augustana.ab.ca/~mohrj/courses/2000.winter/csc220/papers/ru_p_best_practices/ru_p_bestpractices.html
- Rational Website: www.rational.com, rational rose tutorials , <http://emhain.wit.ie/~mmcmahon/CC3/HOut2.pdf>
- Russell Norlund, 2005, integrating the Rational Unified Process with Managing Successful Programmes, <http://www.ibm.com/developerworks/rational/library/jun05/norlund/>

1.4 Abbreviations

SAD: Software Architecture Document
DPFS: Diabetes Patients Follow up System

1.5 Overview

Diabetes Patients Follow up System	Version: <1.1>
Software Architecture Document	Date: <10/5/2009>

In this document we briefly want to explain the use case view of the whole system, logical view showing the most important classes, process view, deployment view that describes the various physical nodes for the most typical platform configurations, and the implementation view.

2. Architectural Representation

This document presents the architectural as a series of views; use case view, deployment view, and implementation view. These views are presented as Rational Rose Models and use the Unified Modeling Language (UML).

3. Architectural Goals and Constraints

1. The All requirements, as discussed in the Requirement Specification must be taken into consideration as the architecture is being developed.
2. The DPFS will be implemented as a client-server system.
3. The DPFS must ensure complete protection of data from unauthorized access.
4. All doctors and nurses must be available from local campus PCs.

4. Use-Case View

A description of the use-case view of the software architecture. The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage (that exercise many architectural elements) or that stress or illustrate a specific, delicate point of the architecture.

Diabetes Patients Follow up System	Version: <1.1>
Software Architecture Document	Date: <10/5/2009>

4.1 Significant Use Case Descriptions

Following are the possible use cases of the system

4.1.1 Login

The users of the system are doctors and secretaries. Any user of the system should open the login screen before using the system then providing his/her user name and password.

4.1.2 Add Treatment

The doctor can chose to enter a prescription for a specific patient. When the doctor takes this selection, the system will display a screen to enable him/her to provide the patient identifier and the details of the prescription.

4.1.3 View Reports

Another thing the doctor can do is to view a patient history report. The doctor selects to view patient history report, and then the system will display a screen asking for the patient identifier. After that, a textual report will be displayed describing the patient status over a period of time.

One more thing the doctor can do is to view the patient status history in a visualized form. The doctor selects to view patient history chart, and then the system will display a screen asking for the patient identifier. After that, a chart will be displayed describing the patient status over a period of time.

Diabetes Patients Follow up System	Version: <1.1>
Software Architecture Document	Date: <10/5/2009>

4.1.4 Insert experimental results

The secretary can insert patient's experimental results, made in medical labs to the system. He/She can do this by selecting the add experimental results, the system will display a screen that enables the secretary to add the patient's identifier and experimental results.

4.1.5 Add Appointments

The secretary also can add appointment. The diabetic patients must visit the diabetes centers on regular intervals; hence the system enables the secretary to add appointment indicating the time of the next visit. When the secretary chooses to add appointment, the system will display a screen to asking for the patient identifier and the date and time of the next visit.

4.1.6 View Treatments

In addition to the aforementioned use cases, the secretary can view the prescription provided by the doctor. When choosing to view the treatment, the system will ask for the patient identifier to view the prescription and then provide a list of all prescriptions provided to the patient. The secretary then selects one of the prescriptions based on the required date.

4.1.7 Send Alert

The last use case of the system is the patient reminding. After the secretary adds an appointment, the system will automatically sends a reminding message for the patient two days before the date of the appointment. The reminding message will have two form; email and SMS.

4.2 Use-Case Realizations

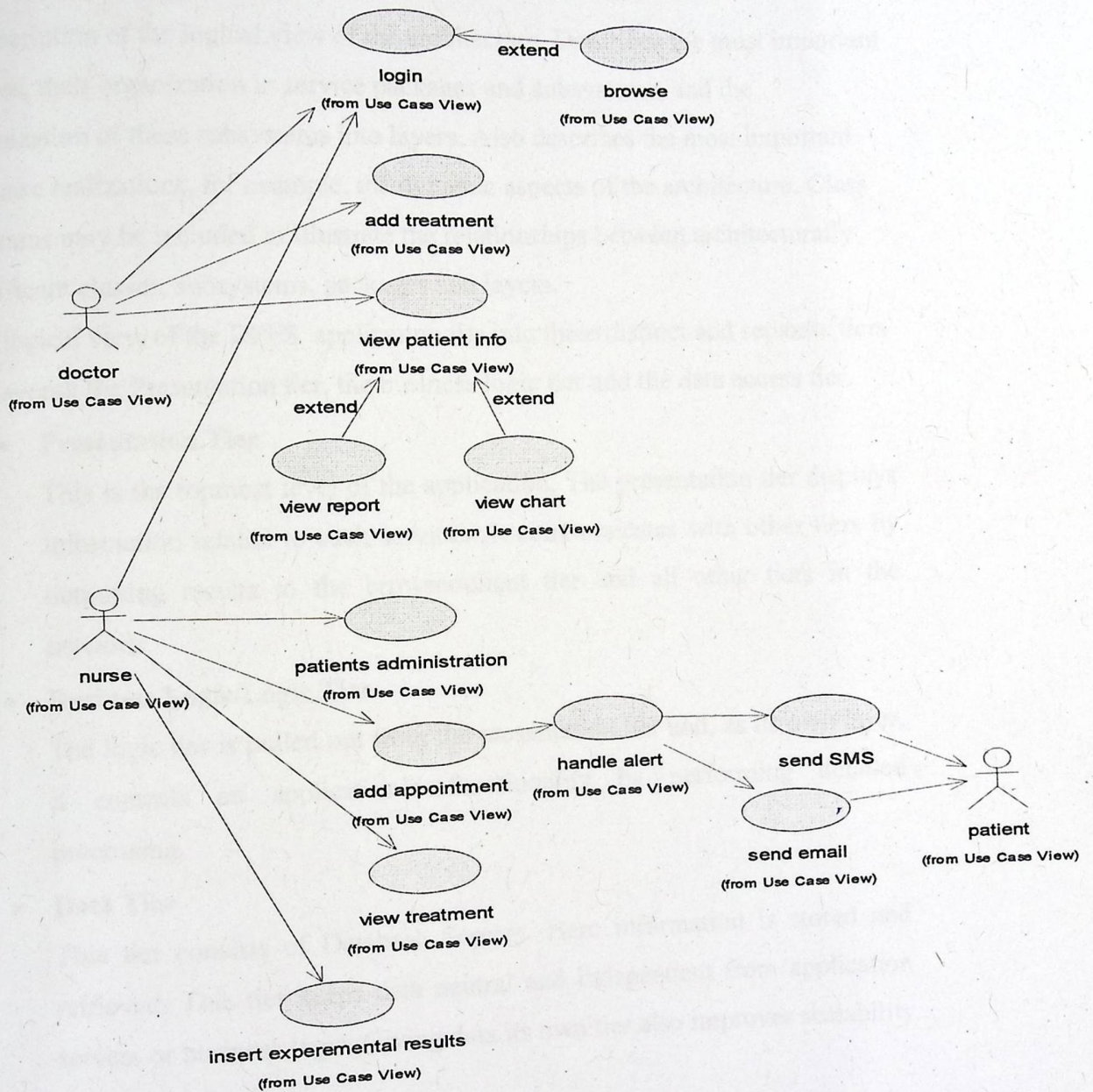


Figure 1: Architectural Use-cases

Diabetes Patients Follow up System	Version: <1.1>
Software Architecture Document	Date: <10/5/2009>

5. Logical View

5.1 Overview

A description of the logical view of the architecture. Describes the most important classes, their organization in service packages and subsystems, and the organization of these subsystems into layers. Also describes the most important use-case realizations, for example, the dynamic aspects of the architecture. Class diagrams may be included to illustrate the relationships between architecturally significant classes, subsystems, packages and layers.

The logical view of the DPFS application fits into three distinct and separate tiers (or layers): the Presentation tier, the business logic tier and the data access tier.

- **Presentation Tier**

This is the topmost level of the application. The presentation tier displays information related to such services. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network

- **Business Logic/Logic Tier**

The logic tier is pulled out from the presentation tier and, as its own layer, it controls an application's functionality by performing detailed processing.

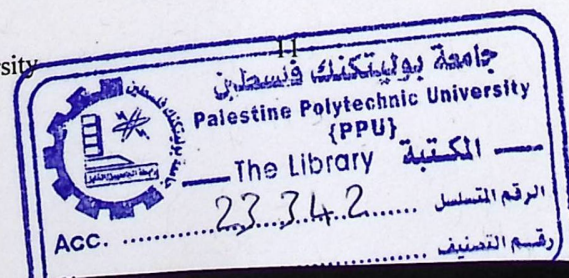
- **Data Tier**

This tier consists of Database Servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance

5.2 Architecturally Significant Design Packages

Confidential

Palestine Polytechnic University



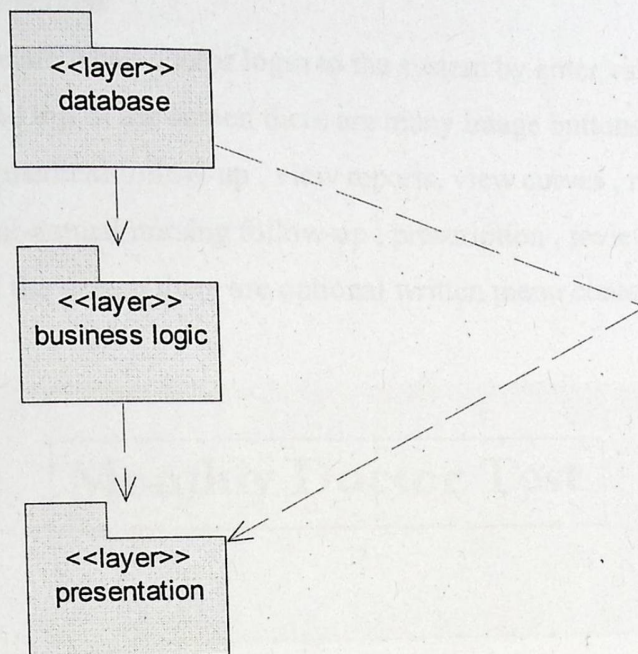


Figure 2: Logical View

5.1.1 Presentation tier:

5.1.1.1 Screens:

a. Main screen

This screen represent main screen in the system where the user login here by enter user name and password in the text box.

User Name

Password

Figure 3: login screen

b. Main doctor screen

This screen appears when doctor login to the system by enter valid username and password , in the top of the screen there are many image buttons (new patient file , login ,search ,medical follow-up , view reports, view curves , monthly nursing follow-up , semi-annual nursing follow-up , prescription , review , insulin therapy) , in the left of the screen there are optional written menu contains the same items.

Monthly Doctor Test

Complaints

Physical Examination (NBI DPAP)

Paraclinical Data

Treatment

Submit

Figure 4: main doctor screen

c. New patient file screen

When to create a new patient file for the first time we use this screen

The screenshot displays a form for adding a new patient. The form is organized into two columns of fields. The left column includes: Name (text input), FileNo. (text input), DOB (three dropdown menus for day, month, and year), Marital Status (dropdown menu), educational level (dropdown menu), Region (dropdown menu), and MobileNO. (text input). The right column includes: IDNo. (text input), Gender (dropdown menu), no.of family members (text input), Job (text input), City/village/Camp (dropdown menu), and Insurance No. (text input). A 'Next' button is located at the bottom right of the form area.

Figure 5: add Patient screen

f. Patient search screen:

The user can search about patient through patient ID or patient name

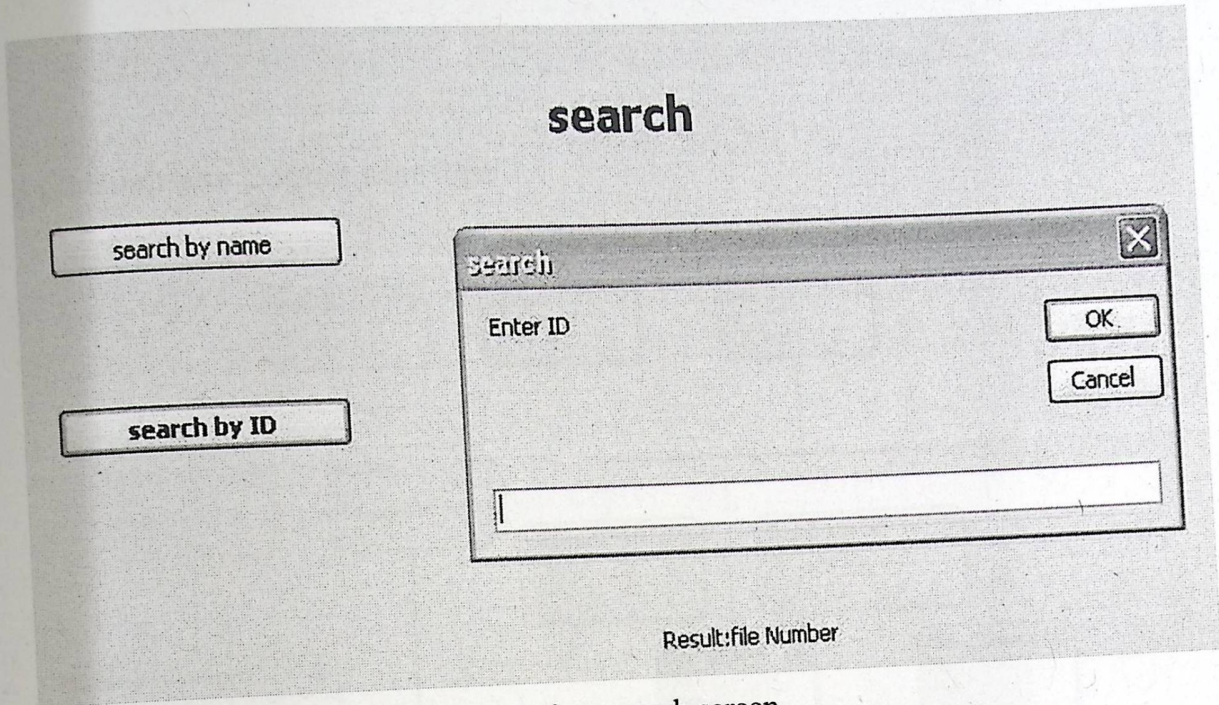


Figure 6: Patient search screen

J. Delete patient screen

Nurse can delete patient through patient ID

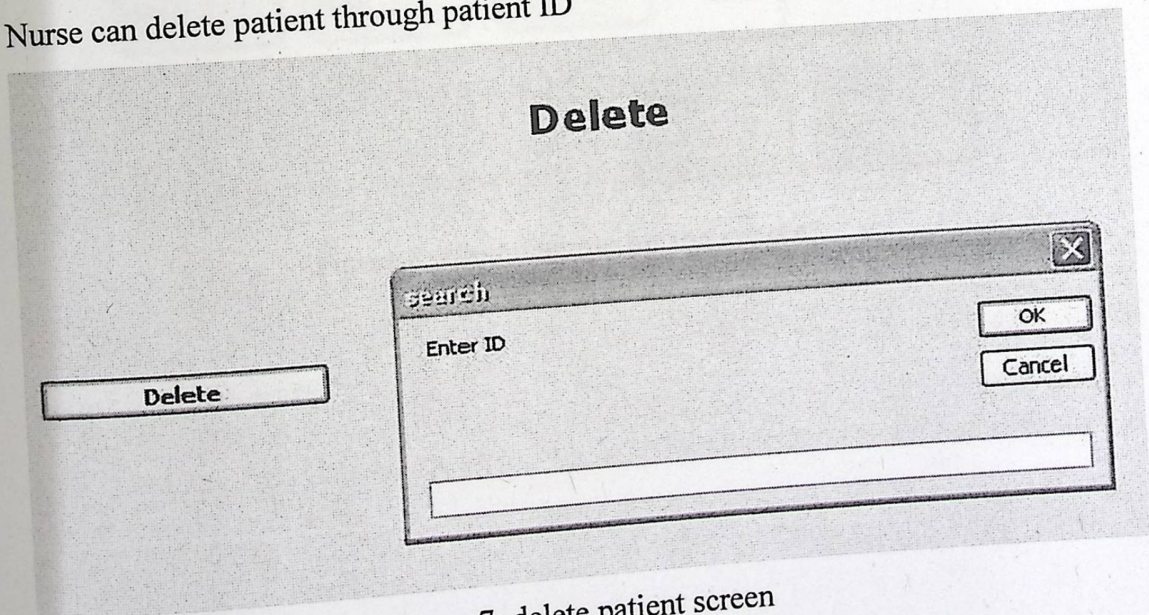


Figure 7: delete patient screen

5.1.2 Business Logic/Logic Tier

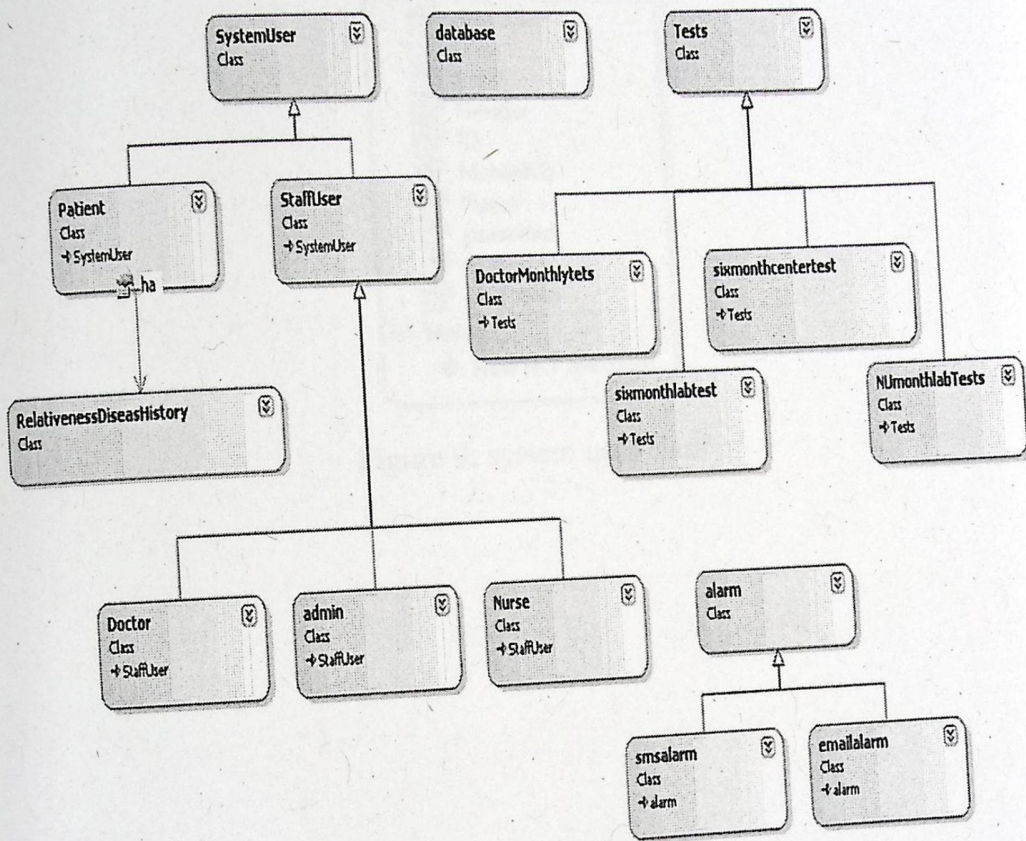


Figure 8: class diagram for the system

5.1.3 Classes in details

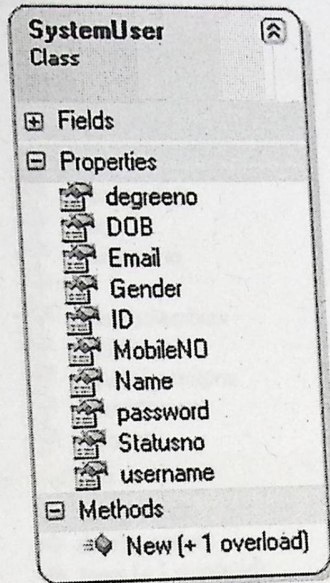


Figure 9: system user class

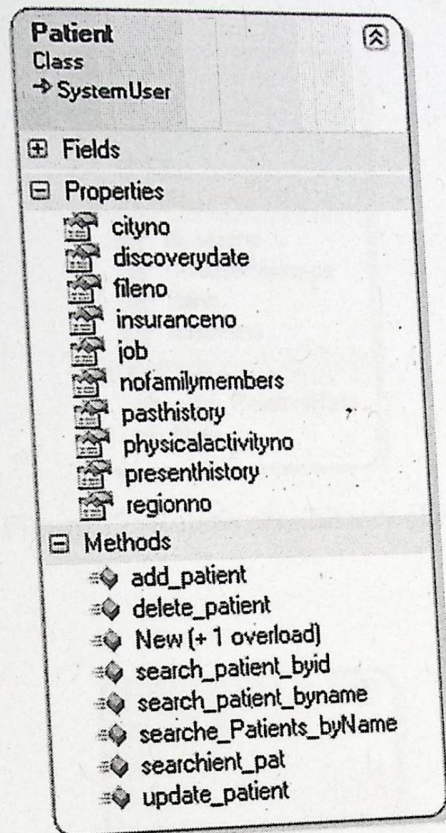


Figure 10: patient class

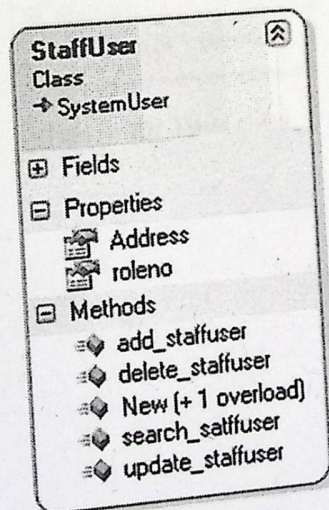


Figure 11: staff user class

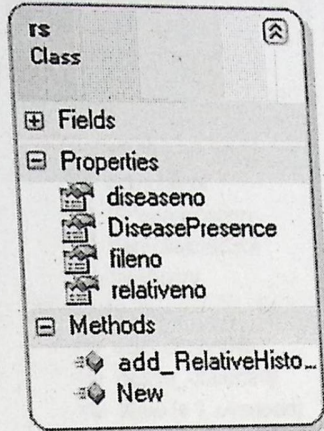


Figure 12: Relativeness history class

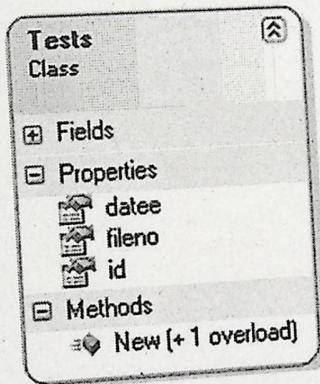


Figure 13: Tests class

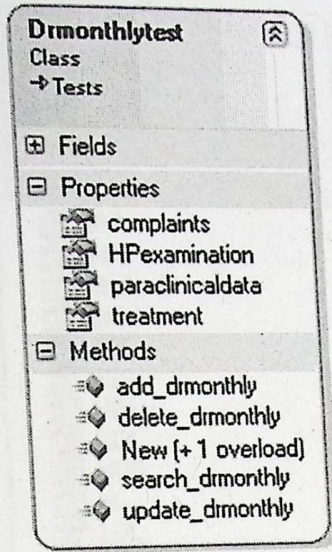


Figure 14: Doctor monthly tests

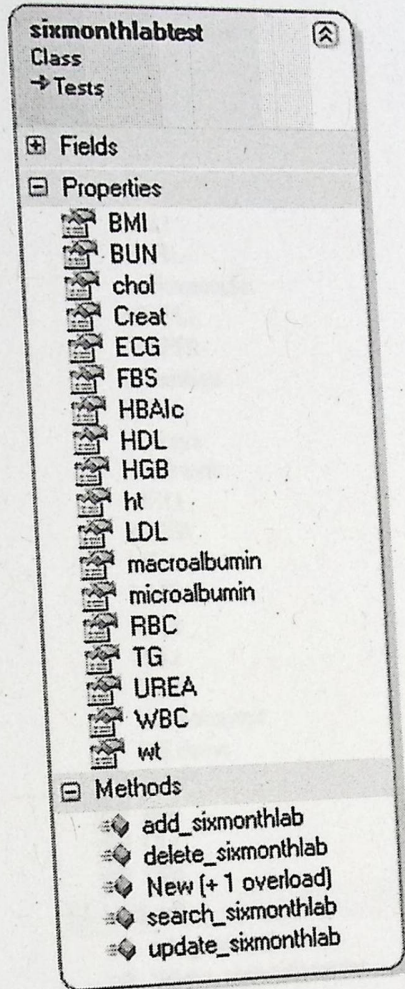


Figure 15: Six month lab test

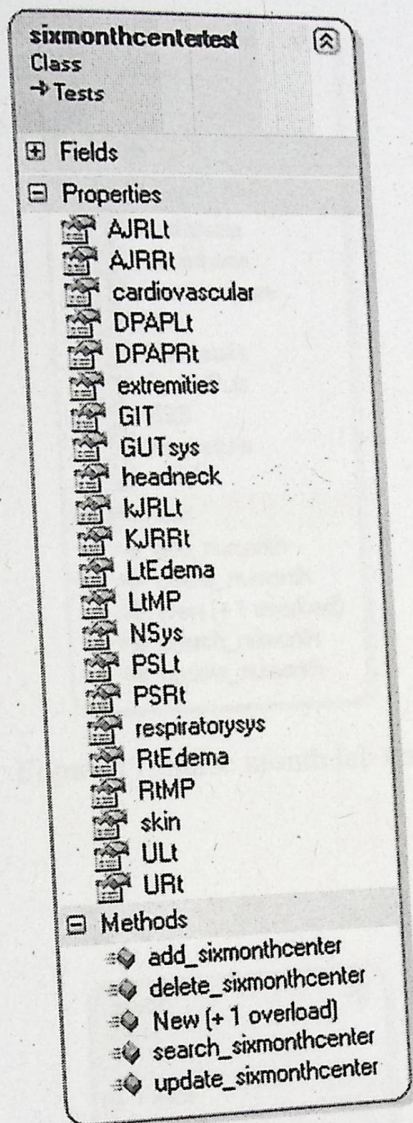


Figure 16: Six month center test

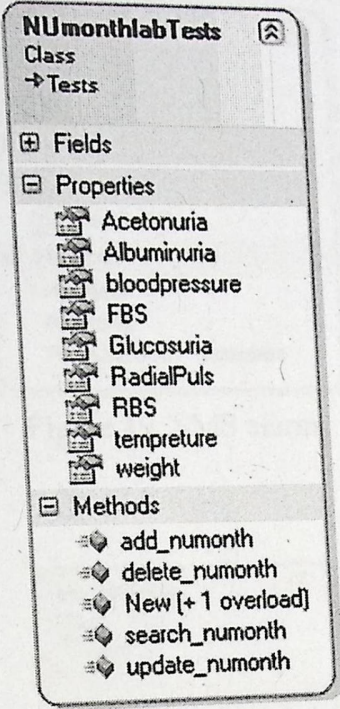


Figure 17: nurse month lab test

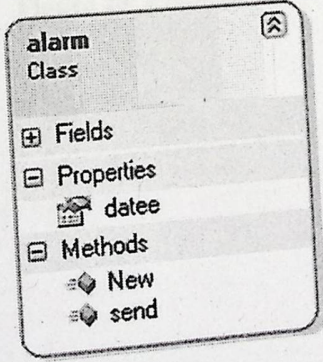


Figure 18: Alarm class

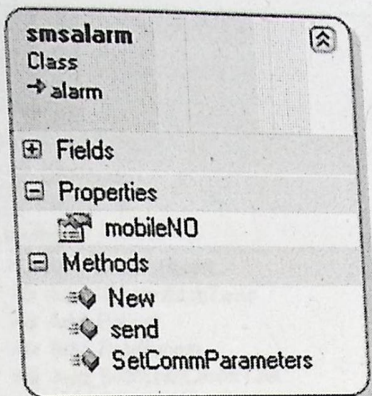


Figure 19: SMS alarm

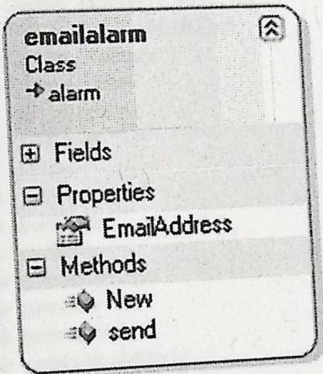


Figure 20: Email alarm

5.1.4 Database tier

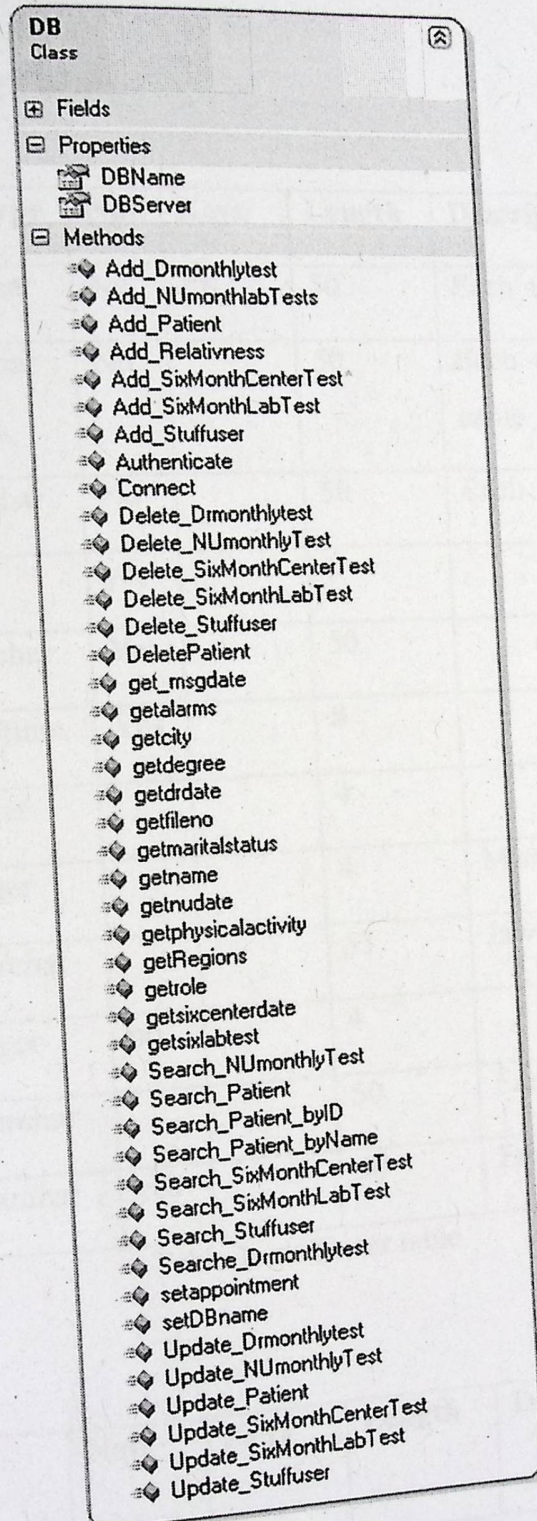


Figure 21: database diagram

5.1.4.1 Database design tables

a. Staff user table

Field name	Data type	Null	Keys	Length	Description
Id	Nvarchar	No	PK	50	Each user has Id
Username	Nvarchar	No		50	Each user has user name
Password	Nvarchar	No		50	Each user has password
Gender	Bit	yes		1	
Address	Nvarchar	Yes		50	
Date of birth	Date/time	Yes		8	
Degree NO	integer			4	
Status no	integer			4	Martial status
Mobile no	Nvarchar			53	Jawwal number
Role no	integer	no		4	
name	Nvarchar			50	Each user has name
email	Nvarchar	yes		50	Email address

Table 1: staff user table

b. City table

Field name	Data type	Null	Keys	Length	Description

City name	Nvarchar	No		50	The name of the city for address
City number	integer	No	PK	4	Each city has number

Table 2: city table

c. Degree table

Field name	Data type	Null	Keys	Length	Description
Degree NO	integer	No	PK	4	Each degree has number
Degree name	Nvarchar	No		50	Level of education

Table 3: degree table

d. Disease table

Field name	Data type	Null	Keys	Length	Description
Disease NO	integer	No	PK	4	Each disease has number
Disease name	Nvarchar	No		50	Each disease has name

Table 4: disease table

e. Dr.monthly test

Field name	Data	Null	Keys	Length	Description
------------	------	------	------	--------	-------------

	type				
File no	integer	No	PK	4	File number
Id	Nvarchar	No	PK	50	Personal ID
Drdate	Date/time	NO	PK	8	Date that the doctor write treatment
Complaints	Nvarchar	Yes		50	Type of tests
PHexamination	Nvarchar	Yes		50	Test of parts of the body
Para clinical data	Nvarchar	Yes		50	Doctor notes
Treatment	Nvarchar	yes		50	The medicine

Table 5: Dr Monthly test table

f. Education level table

Field name	Data type	Null	Keys	Length	Description
Education level NO	Integer	no	PK	4	Tawjihi, B.Sc,
Education level name	Nvarchar			50	

Table 6: Education level table

g. Martial status table

Field name	Data type	Null	Keys	Length	Description
Status	Nvarchar	NO		50	Married or not
Status NO	integer	NO	PK	4	

Table 7: Martial status table

h. NUmonthly test table

Field name	Data type	Null	Keys	Length	Description
file NO	integer	No	PK	4	
ID	Nvarchar	No	PK	50	
Nudate	Date/time	No	PK	8	The date that nurse write in patient information
Weight	decimal	Yes		9(10,0)	
Blood pressure	decimal	Yes		9(18,0)	
Temperature	decimal	Yes		9(10,0)	
Radial plus	decimal	yes		9(18,0)	
FBS	decimal	yes		9(18,0)	Type of tests
RBS	decimal	yes		9(18,0)	Type of tests
glucosuria	decimal	yes		9(18,0)	Type of tests
albuminuria	decimal	yes		9(18,0)	Type of tests

acetonuria	decimal	yes		9(18,0)	Type of tests
------------	---------	-----	--	---------	---------------

Table 8: NUmonthly test

Patient table

Field name	Data type	Null	Keys	Length	Description
File no	Integer	No	PK	4	
Id	nvarchar	NO		50	
Date of birth	Date/time	No		8	
Gender	bit	Yes		1	
Past history	nvarchar	Yes		100	His health was bad or not
Present history	nvarchar	Yes		100	Now his health bad or not
No family members	integer	Yes		4	Number of family member
Job	nvarchar	Yes		50	
Mobile no	nvarchar	Yes		50	
Name	nvarchar	No		50	Patient name
Insurance no	nvarchar	No		50	
Discovery date	Date/time	Yes		8	When he discover the disease
Region no	integer	Yes		4	

City no	integer	Yes		4	
Degree no	integer	No		4	
Status no	integer	No		4	
Physicalactivityno	integer	No		4	
Email	nvarchar	Yes		50	
Password	nvarchar	No		50	
username	nvarchar	No		50	

Table 9: patient table

j. Physical activity table

Field name	Data type	Null	Keys	Length	Description
Physicalactivityno	Integer	no	PK	4	
activity name	nvarchar	No		50	

Table 10: Physical activity table

k. Region table

Field name	Data type	Null	Keys	Length	Description
R no	Integer	no	PK	4	
R name	nvarchar	No		50	Region name that he live in

Table 11: region

l. Relativness table

Field name	Data type	Null	Keys	Length	Description
Relative no	Integer	no	PK	4	
relative relationship	integer	yes		4	Who in the family has the same disease

m. Role

Field name	Data type	Null	Keys	Length	Description
Role no	integer	no	PK	4	
Role name	nvarchar			50	e.g.: Doctor or nurse

Table 12: Role

n. Rs table

Field name	Data type	Null	Keys	Length	Description
Disease no	Integer	No	PK	4	
Relative no	Integer	No		4	
File no	Integer	No		4	
Disease	bit	No		1	If this disease presence

Presence					or not
----------	--	--	--	--	--------

Table 13: RS table

o. Six month center test

Field name	Data type	Null	Keys	Length	Description
File no	Integer	No	PK	4	
Id	Nvarchar	No	PK	50	
Drsexdate	Date/time	No	PK	8	The date written by doctor when patient come in every six month
head neck	Nvarchar	Yes		100	The head and neck have problems or not
cardiovascular	Nvarchar	Yes		100	Physical test
respiratorysys	Nvarchar	Yes		100	Physical test
GIT	Nvarchar	Yes		100	Physical test
GUTsys	Nvarchar	yes		100	Physical test
skin	Nvarchar	Yes		100	Physical test
NSys	Nvarchar	Yes		100	Physical test
Extremities	Nvarchar	Yes		100	Physical test

RtMP	bit	Yes		1	Physical test
LtMP	bit	Yes		1	Physical test
KJRRt	bit	Yes		1	Physical test
KJRLt	bit	Yes		1	Physical test
AJRRt	bit	Yes		1	Physical test
AJRLt	bit	Yes		1	Physical test
URt	bit	Yes		1	Physical test
ULt	bit	Yes		1	Physical test
RtEdema	bit	Yes		1	Physical test
LtEdema	bit	Yes		1	Physical test
DPAPRt	bit	Yes		1	Physical test
DPAPLt	bit	Yes		1	Physical test
PSRt	bit	Yes		1	Physical test
PSLt	bit	Yes		1	Physical test

Table 14: Six month center test

p. Six month lab test

Field name	Data type	Null	Keys	Length	Description
File no	Integer	No	PK	4	
id	Nvarchar	No	PK	50	
nusixdate	Date/time	No	PK	8	The date that written by nurse every six month

					patient visit
Wt	decimal	Yes		9(10,0)	weight
Ht	decimal	Yes		9(10,0)	Height
BMI	decimal	Yes		9(18,0)	Lab test
FBS	decimal	Yes		9(18,0)	Lab test
HBA1c	decimal	Yes		9(18,0)	Lab test
Micro albumin	decimal	Yes		9(18,0)	Lab test
macro albumin	decimal	Yes		9(18,0)	Lab test
WBC	decimal	Yes		9(18,0)	Lab test
RBC	decimal	Yes		9(18,0)	Lab test
HGB	decimal	Yes		9(18,0)	Lab test
BUN	decimal	Yes		9(18,0)	Lab test
UREA	decimal	Yes		9(18,0)	Lab test
Creat	decimal	Yes		9(18,0)	Lab test
Chol	decimal	Yes		9(18,0)	Lab test
TG	decimal	Yes		9(18,0)	Lab test
HDL	decimal	Yes		9(18,0)	Lab test
LDL	decimal	Yes		9(18,0)	Lab test
ECCG	nvarchar	Yes		50	Lab test

Table 15: Six month lab test

					patient visit
Wt	decimal	Yes		9(10,0)	weight
Ht	decimal	Yes		9(10,0)	Height
BMI	decimal	Yes		9(18,0)	Lab test
FBS	decimal	Yes		9(18,0)	Lab test
HBA1c	decimal	Yes		9(18,0)	Lab test
Micro albumin	decimal	Yes		9(18,0)	Lab test
macro albumin	decimal	Yes		9(18,0)	Lab test
WBC	decimal	Yes		9(18,0)	Lab test
RBC	decimal	Yes		9(18,0)	Lab test
HGB	decimal	Yes		9(18,0)	Lab test
BUN	decimal	Yes		9(18,0)	Lab test
UREA	decimal	Yes		9(18,0)	Lab test
Creat	decimal	Yes		9(18,0)	Lab test
Chol	decimal	Yes		9(18,0)	Lab test
TG	decimal	Yes		9(18,0)	Lab test
HDL	decimal	Yes		9(18,0)	Lab test
LDL	decimal	Yes		9(18,0)	Lab test
ECG	nvarchar	Yes		50	Lab test

Table 15: Six month lab test

S. staff martial status

Field name	Data type	null	key	Length	Description
Staff martial status NO	integer	No	PK	8	
Staff martial status name	Nvarchar	No		50	Martial status for staff

Table 16: staff martial status

6. Deployment View

A description of the deployment view of the architecture describes the various physical nodes for the most typical platform configurations. Also describes the allocation of tasks (from the Process View) to the physical nodes.

This section is organized by physical network configuration; each such configuration is illustrated by a deployment diagram, followed by a mapping of processes to each processor.

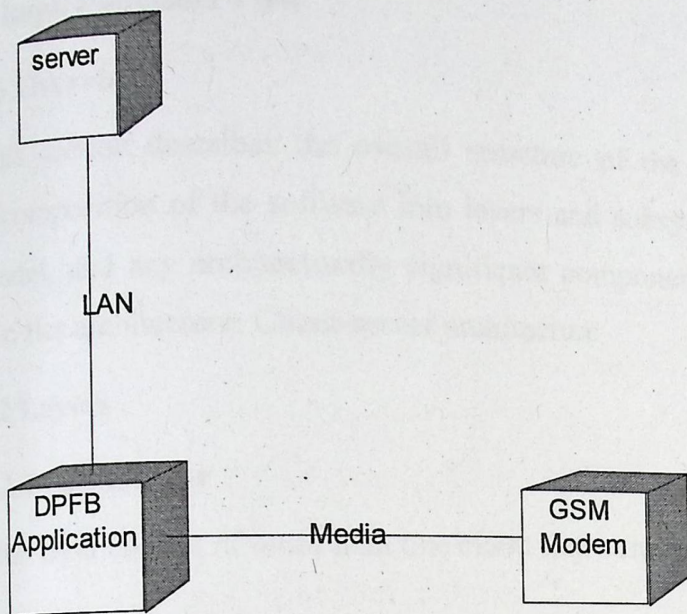


Diagram Name: Deployment View

6.1 Server

It contains the database that only the doctor and nurse can access to it.

6.2 DPFS Application

Connecting to the server in which the nurse can add patient, insert tests and so on, and the doctor can view report, add treatment and so on.

6.3 GSM modem

Connecting to the FSDP Application that can send SMS and email to the patient to remind him/her the next appointment

7. Implementation View

7.1 Overview

This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components. This system implements two tier architecture: Client-server architecture

7.2 Layers

7.2.1 Client layer

This layer consist of more than one client that connect to the server layer via LAN

7.2.2 Server layer

Hardware that contain the database of the system.

8. Size and Performance

The system implements the two tier architecture where a client talks directly to a server; it can scale up by simply adding more users to the server. Client-server architecture is responsiveness that means the system can be developed to a web application in the future.

All the data are stored on the servers, which generally have far greater security controls than most clients.

9. Quality

By breaking down an application into 3 distinct and separate tiers (or layers) - the presentation tier, the business logic tier and the data access tier - you gain advantages in several areas:-

- ✓ Scalability and deployment flexibility - component roles are specialized, improving maintainability, networking, and I/O

7. Implementation View

7.1 Overview

This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components. This system implements two tier architecture: Client-server architecture

7.2 Layers

7.2.1 Client layer

This layer consist of more than one client that connect to the server layer via LAN

7.2.2 Server layer

Hardware that contain the database of the system.

8. Size and Performance

The system implements the two tier architecture where a client talks directly to a server; it can scale up by simply adding more users to the server. Client-server architecture is responsiveness that means the system can be developed to a web application in the future.

All the data are stored on the servers, which generally have far greater security controls than most clients.

9. Quality

By breaking down an application into 3 distinct and separate tiers (or layers) - the presentation tier, the business logic tier and the data access tier - you gain advantages in several areas:-

- ✓ Scalability and deployment flexibility - component roles are specialized, improving maintainability, networking, and I/O

overheads. An application constructed using the 3-Tier software architecture is an ideal candidate for being deployed on 2 layers of hardware.

- ✓ Component roles are clearly defined within a 3-tier framework. This provides a good basis for component-based development and reusability. Components in the business layer can be shared by any number of components in the presentation layer.
- ✓ Infrastructure independence is enhanced by the use of 3-tier architecture. This is because presentation and data access - areas that are often infrastructure-dependent are separated from the application's business logic.
- ✓ A specific set of skills is required for the development of each tier, so tiers can be developed independently of each other.

Palestine Polytechnic University

Diabetes Patient Follow Up System

Test Plan

Version <1.1>

Test Plan	Version: <1.1>
Diabetes Patient Follow up System	Date: <20/5/2009>

Revision History

Date	Version	Description	Author
<15/5/2009>	<1.0>	<first system test>	Alaa Rjoub
<20/5/2009>	<1.1>	<second system test>	Alaa Rjoub

Table of Contents

1.	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	Abbreviations	6
1.4	References	6
1.5	Project Identification	7
2.	Test Strategy	8
2.1	Testing Types	8
2.1.1	Data and Database Integrity Testing	8
2.1.2	Function Testing	8
2.1.3	User Interface Testing	10
2.1.4	Security and Access Control Testing	11
2.1.5	Installation Testing	12
2.2	Roles	13
3.	Test Milestones	16

Test Plan	Version: <1.1>
Diabetes Patient Follow up System	Date: <20/5/2009>

List of tables

Table 1:project identification.....7

Table 2: integrity testing.....8

Table 3: function testing.....9

Table 5: user interface testing.....10

Table 8: Security and Access Control Testing12

Table 9: Installation Testing.....12

1. Introduction

1.1 Purpose

- ✓ This document provides a test plan for the developed system in order to produce a robust system without an unpredictable behavior. In this document, testing strategies will be identified for each part of the system.

1.2 Scope

1.2.1 Stage of testing:

1.2.1.1 Unit testing:

- ✓ we develop the database and test it
- ✓ we develop class diagram and test it
- ✓ we develop the interface and test it
- ✓ we develop crystal reports and charting then we test it
- ✓ we develop the system to send SMS from a computer to mobile phone and test it
- ✓ we develop mobile phone to send SMS from mobile phone to other mobiles and test it

1.2.1.2 Integrated testing:

- ✓ attaching the database with classes and test them
- ✓ attach interface with classes and test them
- ✓ attach crystal report with database and test them
- ✓ attach SMS with database and test them
- ✓ attach all previous modules together and test the system

1.2.1.3 System testing:

Bring the whole component and sub systems together to produce successful system that achieve system goals

1. Introduction

1.1 Purpose

- ✓ This document provides a test plan for the developed system in order to produce a robust system without an unpredictable behavior. In this document, testing strategies will be identified for each part of the system.

1.2 Scope

1.2.1 Stage of testing:

1.2.1.1 Unit testing:

- ✓ we develop the database and test it
- ✓ we develop class diagram and test it
- ✓ we develop the interface and test it
- ✓ we develop crystal reports and charting then we test it
- ✓ we develop the system to send SMS from a computer to mobile phone and test it
- ✓ we develop mobile phone to send SMS from mobile phone to other mobiles and test it

1.2.1.2 Integrated testing:

- ✓ attaching the database with classes and test them
- ✓ attach interface with classes and test them
- ✓ attach crystal report with database and test them
- ✓ attach SMS with database and test them
- ✓ attach all previous modules together and test the system

1.2.1.3 System testing:

Bring the whole component and sub systems together to produce successful system that achieve system goals

1.3 Abbreviations

DPFS: Diabetes Patients Follow up System

1.4 References

- http://hep-proj-grid-fabric.web.cern.ch/hep-proj-grid-fabric/admin_procedures/docs/others/rup.doc,2009

✓

1.5 Project Identification

The table below identifies the documentation and availability used for developing the test plan:

Table 1: project identification

Document (and version / date)	Created or Available	Received or Reviewed	Notes
Requirements Specification	Yes	Yes	Requirement specification was covered in SRS document.
Functional Specification	Yes	Yes	Functional requirement
Use-Case scenario	Yes	Yes	
Project Plan	Yes	Yes	Project scheduling
Design Specifications	Yes	Yes	
Data Model or Flow	Yes	Yes	UML diagrams
Business Functions and Rules	Yes	Yes	
Project or Business Risk Assessment	Yes	Yes	

2. Test Strategy

2.1 Testing Types

2.1.1 Data and Database Integrity Testing

Test Objective:	Database access methods and processes function properly and without data corruption or errors, and module functions are correctly implemented. ✓
Technique:	✓ We use query for data from database ,and we send correct data to database the system accept it and the invalid data the system reject them by error messages We review the returned data to ensure that the correct data was retrieved
Completion Criteria:	✓ All database access methods and processes function as designed and without any data corruption

Table 2: integrity testing

2.1.2 Function Testing

Test Objective:	Test application navigation, data entry, processing and retrieval ✓
-----------------	--

Technique:	<p>We Execute each use case or function using valid and invalid data, to verify the following:</p> <ul style="list-style-type: none"> . The expected results occur when valid data is used. . The appropriate error/warning messages are displayed when invalid data is used. . Each business rule is properly applied. <ul style="list-style-type: none"> ✓ for example : ✓ login use-case check the username and password ✓ Exception occur when telephone number not equal 10 numbers and when name less than 6 characters and insurance number not equal 5 numbers <p>When user enter characters in phone numbers or in insurance number</p> <ul style="list-style-type: none"> ✓
Completion Criteria:	<ul style="list-style-type: none"> ✓ All planned tests have been executed. ✓ All identified defects have been addressed.

Table 3: function testing

2.1.3 User Interface Testing

Test Objective:	<ul style="list-style-type: none">✓ The interface is easy to move around✓ Navigation is easy
Technique:	<ul style="list-style-type: none">✓ Testing each window to verify proper navigation and object states for each application window and objects.
Completion Criteria:	<ul style="list-style-type: none">✓ Each window successfully verified to remain consistent with benchmark version or within acceptable standard

Table 4: user interface testing

2.1.4 Security and Access Control Testing

- ✓ Application-level security: including access to the Data or Business Functions
- ✓ System-level Security: including logging into or remote access to the system.

Test Objective:	<p>To Ensure that the system provides appropriate access and denial of access to certain Users.</p> <ul style="list-style-type: none"> ✓ Application-level Security: nurse cannot enter to the system as doctor .there permission to nurse differ from doctor permission ✓ System-level Security: because the system is windows application not any one from outside can access the system and if he access the system he must have the user name and password to enter to the system
Technique:	<ul style="list-style-type: none"> ✓ Application-level Security: <ul style="list-style-type: none"> nurse account : he can access patient data and insert only lab tests ✓ doctor account : he can access all patient data and add patient physically tests ✓ System-level Access: the system is windows application it cannot be accessed from outside

Completion Criteria:	<ul style="list-style-type: none"> ✓ For each actor type the appropriate function or data are available, and all transactions function as expected and run in prior Application Function tests.
----------------------	--

Table 5: Security and Access Control Testing

2.1.5 Installation Testing

- ✓ Installation testing has two purposes:
- ✓ The first is to ensure that the software can be installed under different conditions such as a new installation, an upgrade, and a complete or custom installation under normal and abnormal conditions. Abnormal conditions include insufficient disk space, lack of privilege to create directories, and so on.
- ✓ The second purpose is to verify that, once installed, the software operates correctly.

Test Objective:	<ul style="list-style-type: none"> ✓ Verify that the system properly installs onto each required hardware configuration under the following conditions: ✓ new installation, a new machine, never installed previously with follow up diabetes system
Technique:	<ul style="list-style-type: none"> ✓ Manually or develop automated scripts, to validate the condition of the target machine new Diabetes patient follow up system never installed
Completion Criteria:	<ul style="list-style-type: none"> ✓ DPFS transactions execute successfully without failure.

Table 6: Installation Testing

Resources

2.2 Roles

This table shows the staffing assumptions for the project.

Human Resources		
Worker	Minimum Resources Recommended (number of full-time roles allocated)	Specific Responsibilities or Comments
Test Manager, Test Project Manager	One	Provides management oversight. <ul style="list-style-type: none"> ✓ Responsibilities: ✓ provide technical direction ✓ acquire appropriate resources ✓ provide management reporting
Test Designer	One	Identifies, prioritizes, and implements test cases. Responsibilities: <ul style="list-style-type: none"> ✓ generate test plan ✓ generate test model ✓ evaluate effectiveness of test effort

Tester	One	<p>Executes the tests.</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> ✓ execute tests ✓ log results ✓ recover from errors ✓ document change requests
Test System Administrator	One	<p>Ensures test environment and assets are managed and maintained.</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> ✓ administer test management system ✓ install and manage access to test systems
Database Administrator, Database Manager	One	<p>Ensures test data (database) environment and assets are managed and maintained.</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> ✓ administer test data (database)

Designer	One	<p>Identifies and defines the operations, attributes, and associations of the test classes.</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> ✓ identifies and defines the test classes ✓ identifies and defines the test packages
Implementer	One	<p>Implements and unit tests the test classes and test packages.</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> ✓ creates the test classes and packages implemented in the test model

3. Test Milestones

Milestone Task	Effort	Start Date	End Date
Plan Test	1 day	14-5-2009	14-5-2009
Design Test	1day	15-5-2009	15-5-2009
Implement Test	2days	16-5-2009	17-5-2009
Execute Test	1day	18-5-2009	18-5-2009
Evaluate Test	1day	19-5-2009	19-5-2009

3. Test Milestones

Milestone Task	Effort	Start Date	End Date
Plan Test	1 day	14-5-2009	14-5-2009
Design Test	1day	15-5-2009	15-5-2009
Implement Test	2days	16-5-2009	17-5-2009
Execute Test	1day	18-5-2009	18-5-2009
Evaluate Test	1day	19-5-2009	19-5-2009

