

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ACKNOWLEDGEMENT	i
	ABSTRACT	ii
	LIST OF FIGURES	iv
	LIST OF ABBREVIATIONS	v
1.	INTRODUCTION	01
	1.1 Background	01
	1.2 Problem statement	01
	1.3 Specific Objectives	01
	1.4 Scope and importance of studies	02
	1.5 Limitations	02
2.	LITERATURE SURVEY	03
3.	REQUIREMENT SPECIFICATIONS	08
4.	PROPOSED SYSTEM AND ARCHITECTURE	09
5.	RESULTS AND DISCUSSION	15
6.	CONCLUSION AND FUTURE ENHANCEMENT	24
	REFERENCES	25

ABSTRACT

In today's world using papers are the least preferred way of maintaining records or data or anything for that matter. Scheduling an appointment with a doctor by a clerk is one among the situations where using papers become the most inefficient way of doing the job. A computerized solution to it does not only solve the problem of maintaining and working with data easy but also helps environment in many ways. The use of making a computer application for the clerk to schedule appointment which by the way is the purpose of the project entitled as "BookAppoint: An appointment scheduling application" is to computerize the management of appointments, booking new ones registering new patients to the hospital , modifying existing patients details and also helps in detecting/Avoiding the collision between appointments. The main factor in consideration is to develop software which is user friendly, simple, fast. The application itself is built for the clerks in the hospitals, so the clerks are given username and password to authenticate and proceed with their job which is made simple with this application.

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1	System architecture diagram	09
2	Database Diagram	10
3	Logic model/Flow diagram	11
4	Jenkins build success	12
5	Unit testing Output(successful test cases)	13
6	Login window	15
7	Login error pop up	15
8	Home window	16
9	Book Appointment window	17
10	Book Appointment List doctors	17
11	Book Appointment success	18

12	Book Appointment unsuccessful	18
13	Validation error: Start time under current time	19
14	Validation error: patient Id required	19
15	Validation error: end time before start time	20
16	Patient registration window	20
17	Modify/Update window	21
18	Edit details window	22
19	Jenkin build console output	23

LIST OF ABBREVIATIONS

DFD	Data Flow Diagram
DBD	Database Diagram
DBMS	Database Management System
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
IT	Information Technology
MFC	Microsoft Foundation Class
SQL	Structured Query Language
VC++	Visual C Plus Plus

CHAPTER 1

INTRODUCTION

1.1 Background

Aim of this project is to create an appointment scheduling windows application for the clerk which helps clerk to book doctor appointment for the patients without any collision between other appointments. It also helps the clerk in modifying/Updating the existing patient's details , registering new patients to the hospital which helps the patient to proceed with booking appointments. Upon Confirmation of appointment, any time later in the future, patients can view (enquire about) their appointment with the help of the clerk. Once the appointment is booked the appointment status is marked as "Confirmed". Once the patient gets consulted by the doctor , the appointment status changes to "Seen" , if the patient did not show up to the hospital then the status is marked as "No Show" and if the patient wants to cancel the appointment then the status is marked as "Cancelled".

1.2 Problem Statement

To create a windows application, which helps in scheduling an appointment with a doctor. Existing patient's details can be modified/updated, New patients can register and go ahead with booking appointments. Upon confirmation of appointment, any time later in the future, patients can view (enquire about) their appointment.

1.3 Specific Objective

In this application there are four modules. The first one is the home page where all the appointments show up, the second one is the appointment booking window, where the appointments are booked. The third one is the Registration window where new patients to the hospital are registered and the last one is Modify/Update window , where existing patient details are updated.

1.4 Scope and Importance of the study

This project , if scaled to bigger application , can incorporate the storage of patients medical record (EMC) and the next time the patient comes , he/she doesn't have to bring the previous medical record , the stored data is sent to the doctor directly to his machine. And also a new window for doctors can be created where the doctors can update their availability themselves instead of letting the clerk know of their unavailability. With this we can achieve a seamless transition between appointment booking and transferring patient details to the doctors window.

1.5 Limitations

This project is limited to the hospital clerk right now. Implementing an online appointment scheduling application which would help patients book their appointment themselves sounds really good but practically to deploy a system like this in a hospital that is still on papers (small scale hospitals) to book their appointments is not feasible.

CHAPTER 2

LITERATURE SURVEY

Waiting Time

Waiting time simply means a period of time which one must wait in order for a specific action to occur, after that action is requested or mandated (Fernandes et al., 1994). Patients' waiting time has been defined as "the length of time from when the patient entered the outpatient clinic to the time the patient actually received his or her prescription" (Jamaiah, 2003). It is defined as the total time from registration until consultation with a doctor. There were two waiting times, the first is time taken to see a physician and the second is time to obtain medicine (Suriani, 2003). This paper deals with the waiting time to see physicians. Long waiting times are a serious problem for patients using urban health centres in developing countries (Bachmann, 1998). A block appointment system was introduced and evaluated in a large South African health centre. Waiting times of all patients were measured over one-week period before and after the implementation of appointments. Focus groups and individual interviews were conducted with staff and patients. After introducing appointments, patients with acute and chronic illnesses and having appointments had significantly shorter waits time than similar patients without appointments (Mahomed, 1998). Appointments had no benefits for patients not seeing doctors or collecting repeat medication. There was, however, an overall increase in patients' waiting times after introducing the system, mainly due to one typical day in the follow-up study. Focus groups and interviews revealed that staff were skeptical at baseline but at follow-up were positive about the system. Patients were enthusiastic about the appointment system at all stages. The study shows that block appointments can reduce patients' waiting times for acute patients, but may not be suitable for all patients. Staff and patients had different views, which converged with experience of the new system (Mahomed, 1998).

Patients' Appointment System

A patient appointment system or appointment schedule for health care center started long time ago (Harper, 2003). Management of patients' appointments has earlier works and has developed simplified queuing models and fairly static scheduling conditions. Another attempt was made to calculate the waiting time between patient and doctor

using the mathematical queuing models to minimize waiting time (Gamlin, 2003). However; traditionally the appointment system has considered that the doctor time is more important than patient time (Wijewickrama, 2005). So an appointment system was designed to minimize the doctor idle time but current designing of an appointment system is based on decisive factors with respect to both the patient and doctor (Takakuwa, 2005). The patient appointment system has complex structures because it represents the patient appointment time in the healthcare center and controls the patient waiting time based on the type and the period of patient appointment (Gamlin, 2003). Moreover, a patient appointment system is International Journal of Computer Science & Information Technology (IJCSIT) Vol 6, No 4, August 2014 62 meant for: managing doctor's time, reducing patient's waiting time, reducing doctor's idle time, reducing nurse's idle time, and improving the quality of service in the health care (Harper, 2003).

Appointment Delay

Past research shows that the longer the appointment delay which is defined as the time between the day a patient requests an appointment and her actual appointment date, the higher the chances that he/she will cancel or not show up (Gallucci et al. 2005). This suggests an obvious way of minimizing no-shows and cancellations: this is done by asking the patients to come right away or make appointment requests on the day they want to be seen (Murray, 2000). This is called an open access (OA) or advanced access policy (Tantau, 2000), and of late it has become a popular paradigm in practice and the subject of active research. Several authors report on their experiences in implementing OA, both positive and negative (Dixon et al. 2006). Some practitioners strongly advocate OA (Murray and Tantau 2000), and there are some who are strongly against it (Lamb, 2002).

Managing Patients' Appointment system

According to Dexter (1999), managing patient appointment system is a computer application used to manage and reduce the patient waiting time in the health care center. Some health care centers do not use any appointment system. So it has a longer average patients' waiting time than the health care center that adopts the patients' appointment system. While patients can wait for more than one hour to be attended to by a physician in a health care center, they also can feel that they are being disregarded and treated unfairly. So when patients are given the time of appointment in a health care centre,

they can evaluate the quality of service in the centre (Dexter, 1999). Hence, developing patients' appointment process for health care centres necessitates the use of a sophisticated queuing model that captures much of the real system's features (saving time, reducing idle time, etc). Therefore the appointment schedule represents the real situation in the health care centre faced by patient appointment schedulers (Rohleder, 2002). On the other hand, the standard practice for scheduling and processing patient appointments are based on the nature of treatments of the patients and that better approaches more sensitive to patient needs are desirable (Klassen, 2002).

Existing Hospital Appointment Schemes

One application developed to manage patients' appointment scheduling has used exponential enter arrival times. This model assumes that the exponential enter arrival times could not be directly validated by date, and it is limited due to the nature of the appointment scheduling (Rohleder, 2002). Since appointments are scheduled in the future, the exact model of call arrivals will only have limited impact on measures related to the time between the call and the appointment time. For this reason, the challenge for making appointment system is designing a suitable system based on the health care procedure environment (Klassen, 2002). Hence, the appointment provider in the health care center can schedule a patient into an appropriate time slot on a given day. Klassen (2004) developed another method for managing patients' appointment using multiple schedule appointment in multiple period environments. Patients can call for any appointment time but if the period time is full, they should replace the appointment to another time. Moreover, various combinations for multi appointment and double booking are measured and recommended for different operational use depending on the health care environment because the varying appointment request has little effect on appointment system performance, especially maintaining acceptable performance, except when the system has the overloaded option (Rohleder, 2004). Many studies about patients' appointment have found that there are rules or policies for scheduling appointment system such as no scheduling for more than 20 or 30 clients and the best schedule is to place two patients in the first appointment and spread the rest consistently over a period based on average service times (Klassen, 2004). On the other hand, a patient can call for an appointment without knowledge of the type of appointment and appointment queue number and the patient is not aware whether the appointment is variable or not. Sometimes the exact duration for each patient can be known but at other

times this is unknown (Rohleder, 2004). Another system developed by Mustafa, (2004) allows a registered patient, having user name and password, to access and explore the list of physicians alphabetically and select a physician whose email contact and profile are also provided. A patient can also view the physician working calendar to find out his/her working and non working day to make an appointment. When the patient selects, view calendar the patient can then choose any valid day in any month to make an appointment (Mustafa, 2004). After that, the patient will receive an e-mail from the system to confirm the appointment time or to inform the patient that the selected time is already taken by another patient or blocked by the physician. In general, the patient appointment system provides all the choices and the capabilities to the patients, such as selecting a physician, selecting the time of appointment, and allows them to access the health care system day or night and schedule their own appointments using the Internet without spending time holding for a nurse or having lengthy phone calls. Wijewickrama and Takakuwa (2005) opine that the health care operating time (due time) is from 8:30 am to 5:30 pm during the week days. Throughout this period, four types of patients arrive to have a consultation appointment in the health care center-appointed patients, same day appointment patients (walk-ins), patients who come for a medical test and new patients (Wijewickrama, 2005). Patients who have appointments are given priority over those who walkin for consultation. Consequently, these latter patients have to wait a long time in the waiting room to meet a doctor even if the consultation time only last few minutes (Takakuwa, 2005). Porta-Sales et al. (2005) developed another system. The main concept of the system is contacting, screening and scheduling appointment with the health care center initially by an expert nurse and the patient initiating contacting with the health care center using the telephone. Moreover, the health care center can be accessible from different places. So there should be PC resources and PC consultations to be accessed from different sources, from other hospitals, from general practitioners, or even from the patients themselves. Porta-Sales et al. (2005) studied 534 patients for a period exceeding one year. After the first visit, 195 patients did not return for the second scheduled appointment and 203 patients had progressed on to the third scheduled visit. The main reason given for the scheduled visits was admission into the health care; the median time-lapse between the first and second visit was 21 days, between the second and third was 27.5 days and between the first and third was 48 days. Comparing patients, who did not attend the three consecutive visits with those who did, indicated that the former had (at the first visit) a lower performance status. Su et al.

(2003) studied in a private hospital which has several clinics. For each clinic, the average patient load is 20 per consultation section (morning or afternoon) and the health care system adopts both a patient appointment model and patient registration model. The system allows patients to have self-selected specific physicians for consultation and registration (Shih, 2003). The management appointment system studied by Su and Shih, (2003) is based on the first 20 reserved for scheduled patients, after that, only seven are offered for scheduling. Odd numbers after 20 are left for walk-ins. The arrival time of the first patient is assumed to be the same as the clinic starting time. The scheduled patients are assigned based on 3- main intervals and are also informed about their appointed arrival times (Su, 2003). If the scheduled patient does not appear on time, the next available patient receives consultation immediately. The management operating philosophy of services here is based on “first in, first seen” to limit patient waiting time. Therefore, patients can walk-in to see a physician, when patient shows up at the appointed time (Shih, 2003). Some of the existing appointment booking system have some limitations and the system developed in this research eradicate the limitations of the existing system in confirming patient medical appointment by sending an email to the patient if the appointment have being confirmed or not. It will also enable the patients to view and monitor their medical records online.

CHAPTER 3

REQUIREMENT SPECIFICATIONS

- Visual studio professional 2015.
- MFC library
- MySql workbench
- Jenkins (Continuous integration)

System specifications:

As the application is build in MFC/VC++ , it is a windows specific applications. That is, it runs only on windows pc's.

- Windows OS
- Min. 2GB RAM
- Processor: 2.4 Ghz processing speed and above

CHAPTER 4

PROPOSED SYSTEM AND ARCHITECTURE

System Design

After interpretation of the data, tables were drawn and process of data determined to guide the researcher of the implementation stage of the project. The tools, which were employed during this methodology stage, where mainly tables, Data Flow Diagrams and Entity Relationship Diagrams. The design ensures that only allows authorized users to access the systems information.

System Architecture

This gives a high level view of the new system with the main components of the system and the service they provide and how they communicate. The system is implemented using a three-tier architecture that comprises of our interface, process management and DBMS as illustrated bellow.

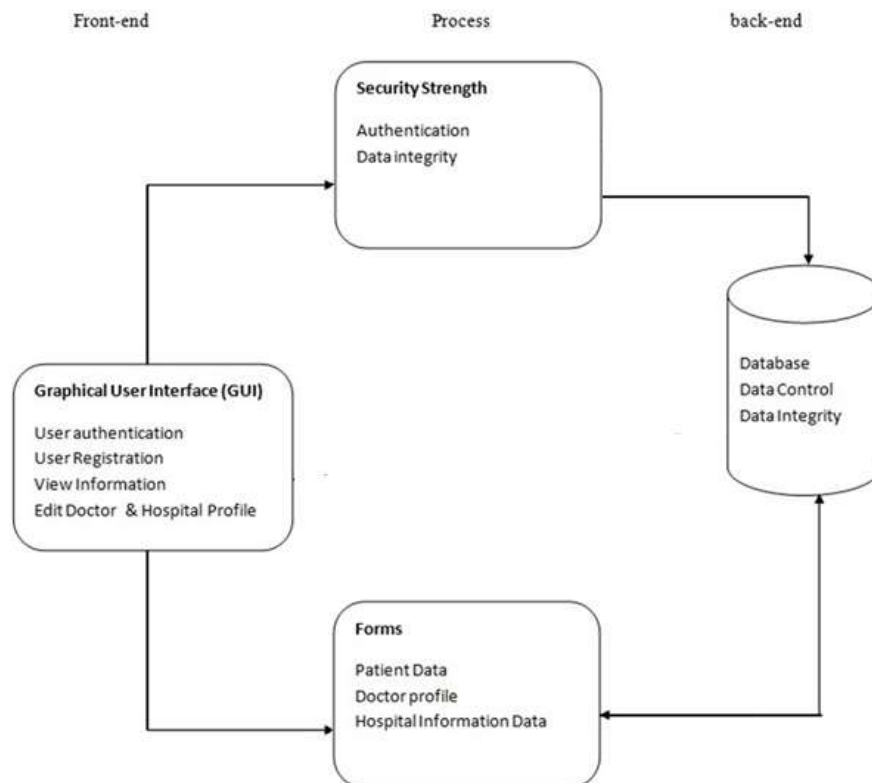


Fig 4.1 System architecture diagram

Database Diagram

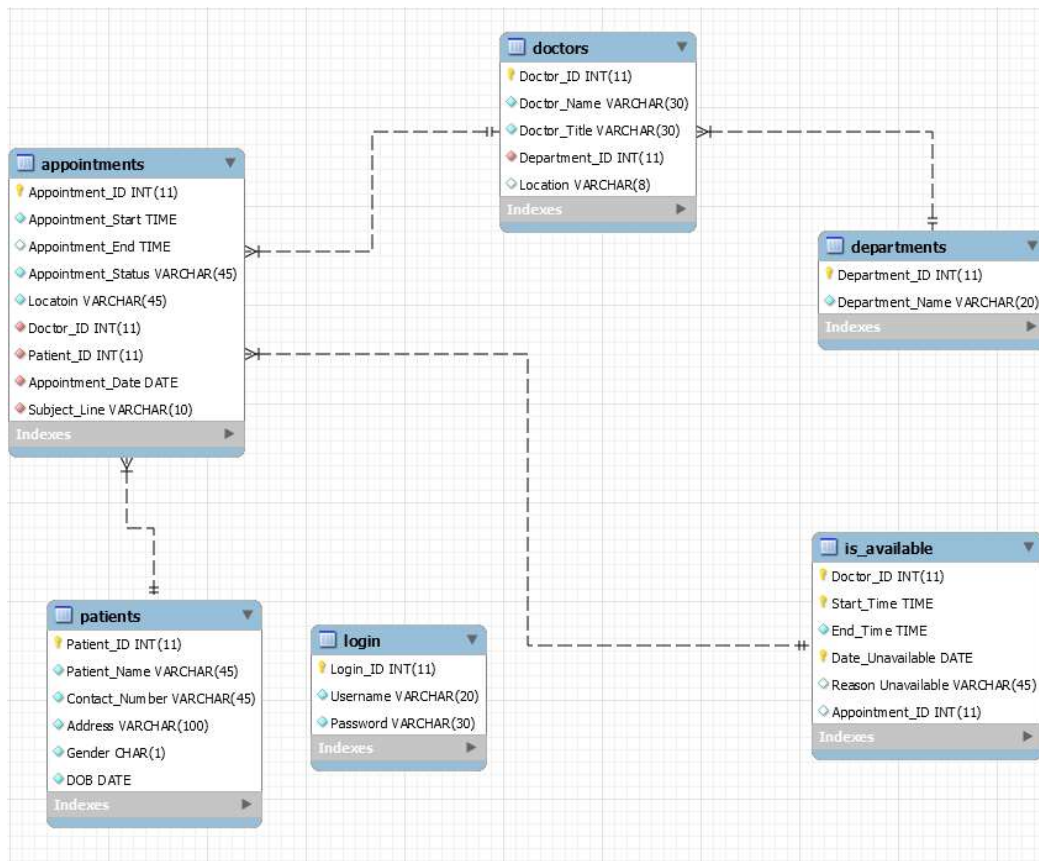


Fig 4.2 Database Diagram

Proposed system

In this application , the clerk is authenticated using the login window, where he must enter his username and password correctly. Wrong username or password pops up an error message. Every clerk working in the hospital is given his/her own credentials to login. Upon successful login, the home window opens up where the clerk gets to see the appointments that are booked for the selected date in the calendar.

There are 3 options on the top to either book a new appointment or Register new patient or Update existing patients details.Each of the above mentioned options upon clicking opens a new window where more options for the specific task is provided to proceed further.In addition to the options at the top of the window in the home window there are so many other filters that are provided in the home window itself to search for the appointments based on the filters.

4.3 Project Flow

Logical Model

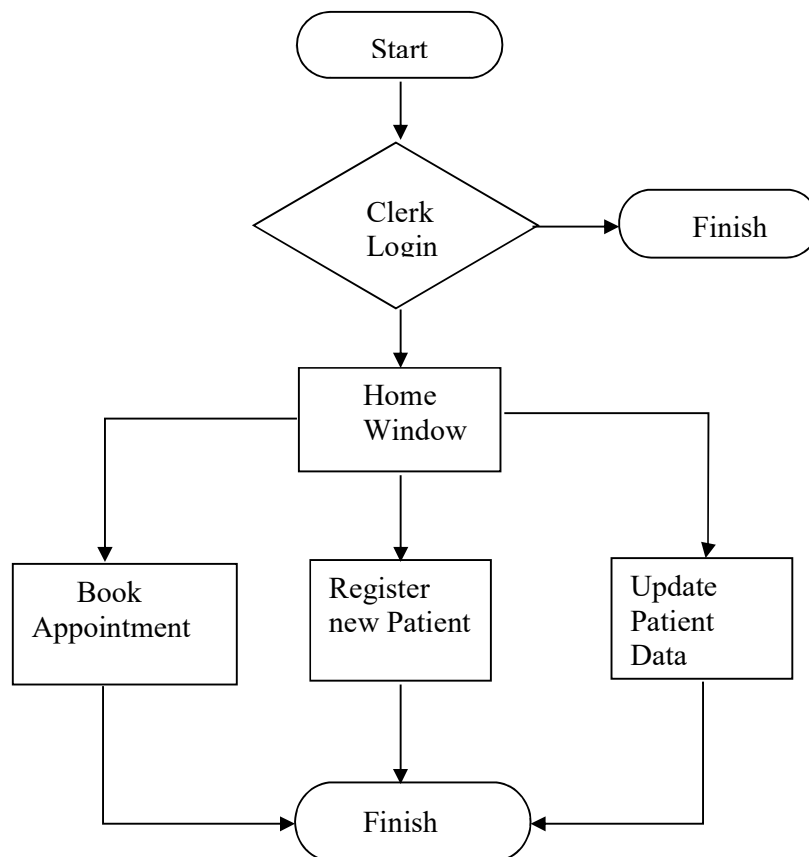


Fig 4.3 Logical model / flow diagram

Testing

Testing an application that has GUI as the major part of the code is not completely possible. But when the logic and GUI is separated from each, Unit testing the logic is possible. Hence in this project the logic is separated from the GUI components and the basic test cases for the logic is performed using the native c++ testing framework that comes with visual studio professional 2015. The test cases are the ones that are written for the filters, input boxes that take the input and validates the input to provide corresponding output. The result of the tests are attached below.

Continuous Integration

For continuous integration an open source tool called Jenkins is used, Jenkins is an open source mechanization server written in Java. Jenkins mechanizes the non-human piece of the product improvement process, with consistent reconciliation and encouraging specialized parts of persistent conveyance. It is a server-based framework that keeps running in servlet compartments, for example, Apache Tomcat. It bolsters adaptation control instruments, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, TD/OVS, ClearCase and RTC, and can execute Apache Ant, Apache Maven and SBT based ventures just as self-assertive shell contents and Windows clump directions.

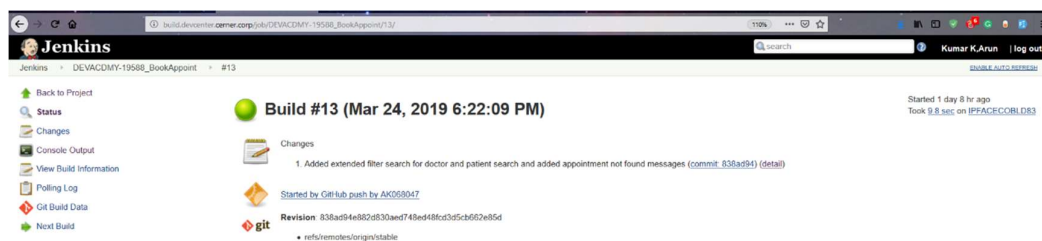


Fig 4.4 Jenkins build success

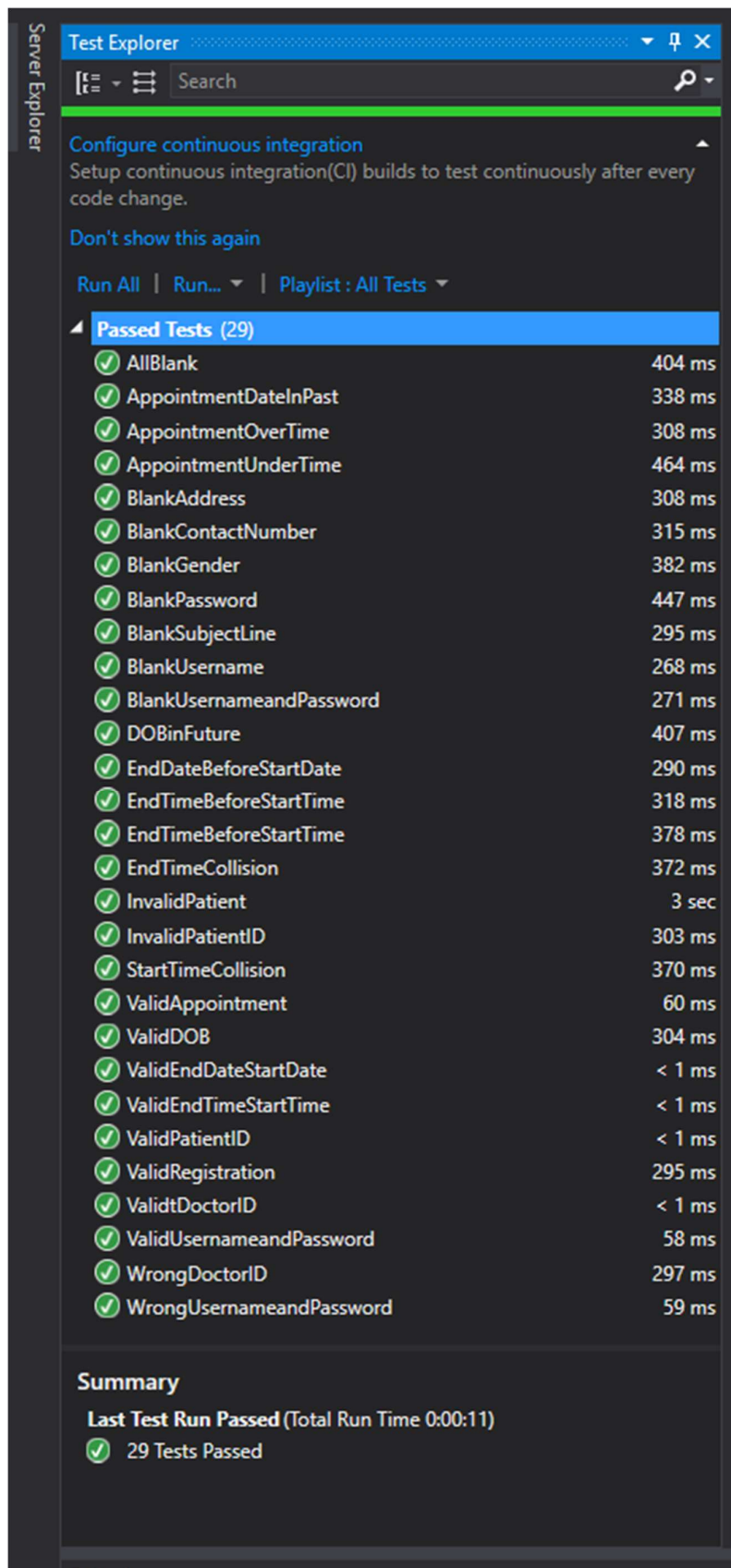


Fig 4.5 Unit testing output (successful tests)

Exception Handling

Three categories of outcomes can occur when a function is called during program execution: normal execution, erroneous execution, or abnormal execution. Each category is described below.

- Normal execution

The function may execute normally and return. Some functions return a result code to the caller, which indicates the outcome of the function. The possible result codes are strictly defined for the function and represent the range of possible outcomes of the function. The result code can indicate success or failure or can even indicate a particular type of failure that is within the normal range of expectations. For example, a file-status function can return a code that indicates that the file does not exist. Note that the term "error code" is not used because a result code represents one of many expected outcomes.

- Erroneous execution

The caller makes some mistake in passing arguments to the function or calls the function in an inappropriate context. This situation causes an error, and it should be detected by an assertion during program development.

- Abnormal execution

Abnormal execution includes situations where conditions outside the program's control, such as low memory or I/O errors, are influencing the outcome of the function. Abnormal situations should be handled by catching and throwing exceptions.

Using exceptions is especially appropriate for abnormal execution.

MFC exception support that are used in this project are :

CDBException class and COLException Class

CHAPTER 5

RESULTS AND DISCUSSIONS

Login window:

The very first window that opens upon executing the application is the login window where the clerk has to enter his credentials to login successfully. Successful authentication leads the clerk to proceed further i.e., the next window, the home window where the clerk can see all the appointments that are booked for a specified data and their status. Upon unsuccessful login the following error message pops up,

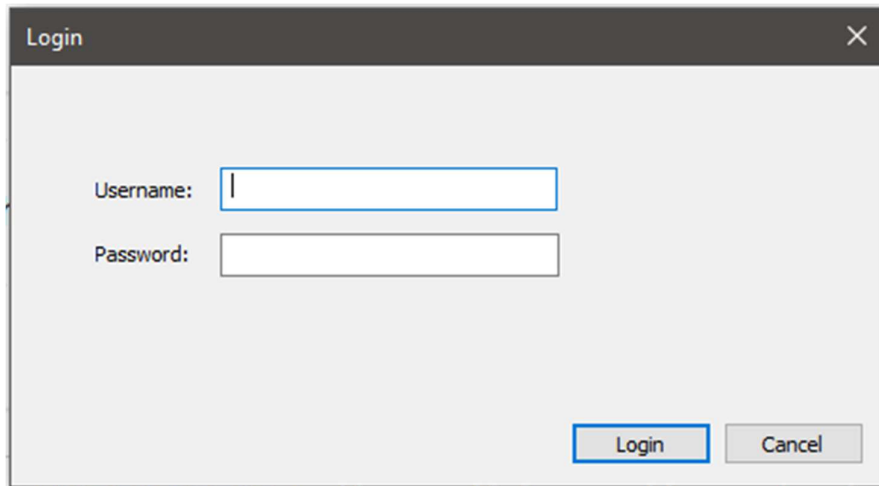


Fig 5.1 Login Window

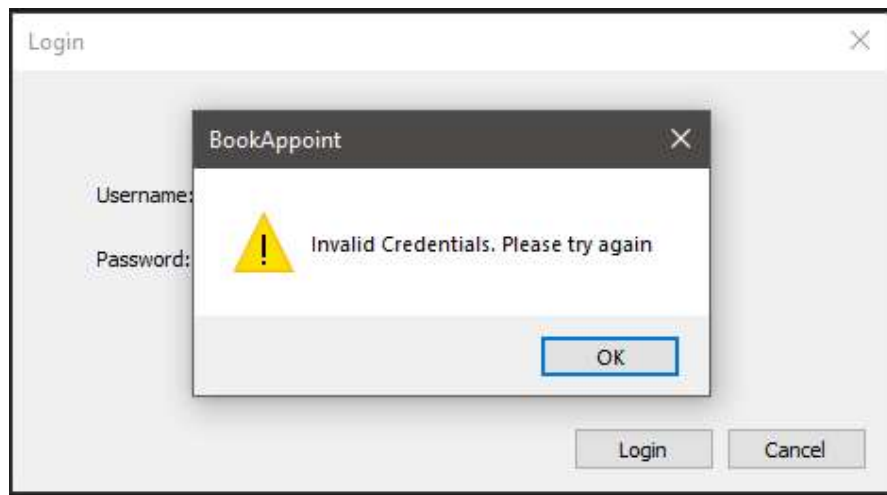


Fig 5.2 Login error pop up

[illegible]

The home window is where the clerk inputs all sorts of filters that are available to the left of the screen to view the appointments. Example, filter appointments based on DoctorID, PatientID , Date interval or Time interval .

Book Appointment window

The below show figures are the appointment booking windows in different stages of its operation. The first one is the state of the window when a new appointment is to be booked.

Book Appointment

Date: 6/ 6/2019

From: 12:05:55 PM To: 12:05:55 PM

Department: List Doctors Add

Doctor ID: 0 Location: Location:

Patient ID: 0

Subject Line:

OK Cancel

Fig 5.3 Book appointment window

Upon entering the right values in the columns, i.e., Entering the date when the appointment needs to be booked and then the time , and then there is a drop down , which displays all the departments that are there in the hospital eg, cardiology, dermatology etc. So the clerk has to pick one and then click list Doctors button to list the all the available doctors in the hospital under that department.

Book Appointment

Date: 6/ 8/2019

From: 5:00:00 PM To: 5:30:00 PM

Department: Diabetology List Doctors Add

Doctor ID: 19 Location: Q2-3B5

Patient ID: 0

Subject Line:

Doctor ID	Doctor Name	Doctor Posting	Location
47	Dr. Shanthy	Asst. Professor/Surgeon (M.S)	G3-
19	Dr. Soundharam	Senior resident(M.S)	Q2-
14	Dr. Kosuksi pasapugal	Senior resident(M.S)	F2-

OK Cancel

Fig 5.4 Book Appointment List doctors

Upon selecting the desired doctor the clerk needs to add the doctor to the appointment by clicking the add button and then the value in the columns DoctorID and Location are autopopulated based on the doctors details.

Upon completion of inputting the details proceed with clicking the ok button which helps in booking the appointment successfully.

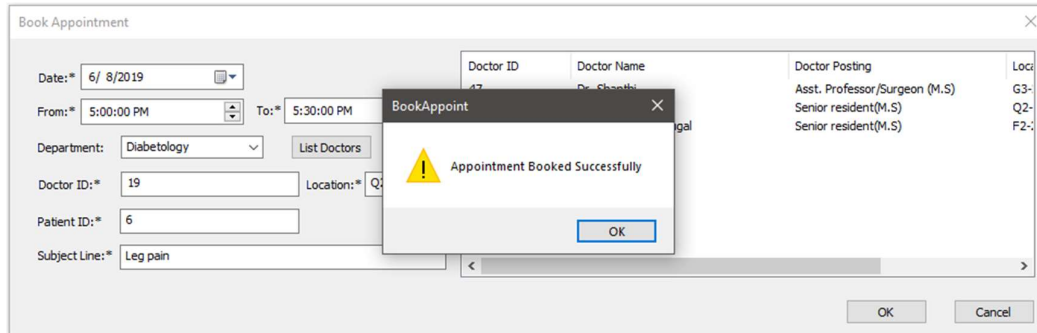


Fig 5.5 Boood Appointment success

If there is any collision between the timing of the appointments the following message is shown to the clerk which lets the clerk know the reason for unsuccessful appointment booking. In the following the figure for example, the reason for unsuccessful appointment booking is that the doctor is already in another appointment on the specified time.

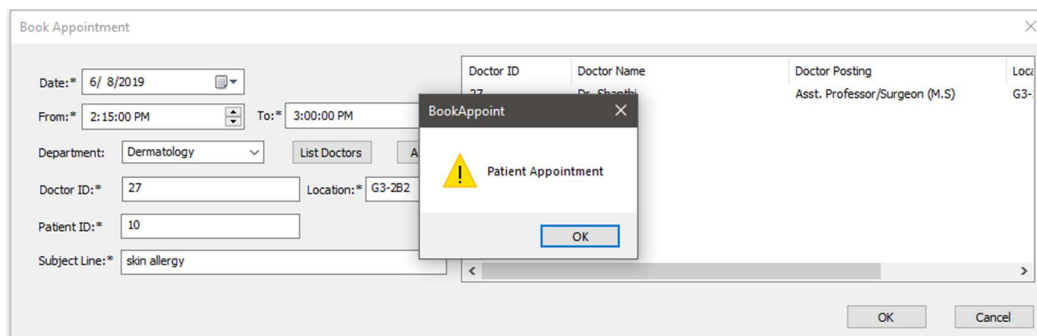


Fig 5.6 Book Appointment unsuccessful

Some validation errors :

The below images shows that the start time cannot be before the current time , which helps in avoiding the error of booking the appointment in the past.

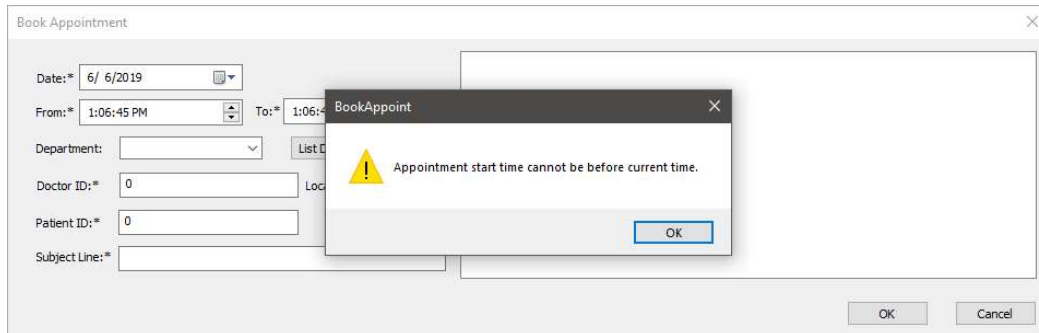


Fig 5.7 Validation error: start time under current time

The below image shows that you must provide a valid patient ID , which helps in avoiding the error of booking the appointment without a patient.

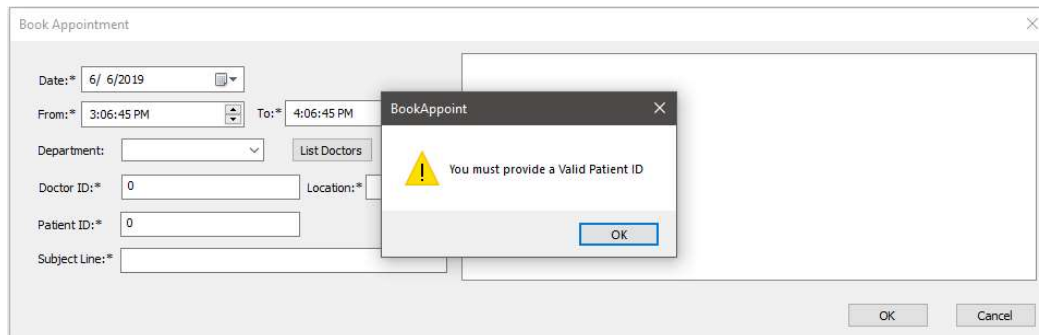


Fig 5.8 Validation error: Patient ID required

The below image shows that the end time cannot be before the start time, which helps in avoiding the error of booking an appointment with wrong timings.

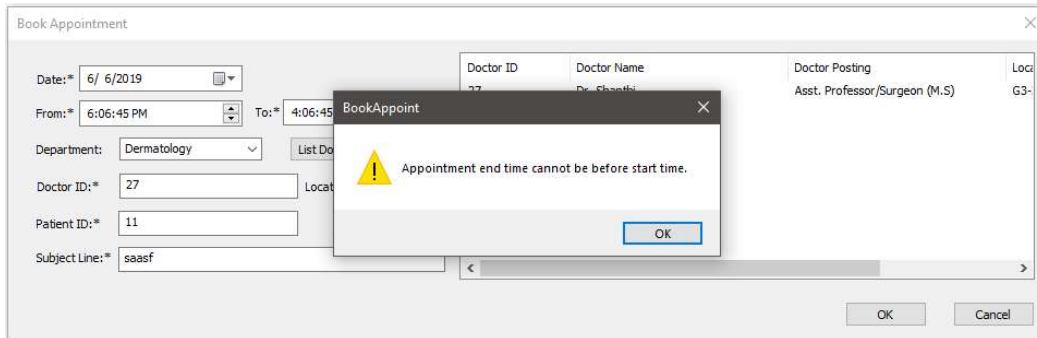


Fig 5.9 Validation error: end time before start time

Patient registration window:

This is the Patient registration window , where new patients to the hospital are registered by the clerk. The clerk will get information such as the patient name, DOB, gender , contact number and address. These information are then confirmed by the clerk . The generated patient ID is given to the patient.

The screenshot shows a 'Patient Registration' window with the following fields: Patient Name:*, Date of Birth:*, Gender:*, Country Code:*, Contact Number:*, and Address:*. The 'Date of Birth' field is set to 3/25/2019. The 'Confirm' button is highlighted with a blue border, and there is a 'Cancel' button next to it.

Fig 5.10 Patient Registration window

Modify/Update patient details window:

This window is where the clerk is allowed to modify or update the details of the patient upon the request of updation by the patient . Updating the details can be done in 3 ways. The patient can either give the Patinet ID to the clerk to find his records, or he can give his DOB to the clerk or he can give his contact number to the clerk to find his record and then edit it.

Modify/Update Data

Search By

☐ Date of Birth

☐ Contact Number

☐ Name

☒ All

Search by DOB

Date of Birth:

3/25/2019

Search by Contact Number

Country Code:

Contact Number:

Search by Name


Patient Name:

Reset Search Edit Cancel

Fig 5.11 Modify/Update window

The above shown figure is the window that opens when the modify/update tab is clicked. It doesn't show any details without giving any input filter to search for. Once the filters are specified the window looks like the below image. Upon clicking the search button the Edit Details window opens up , where the clerk can modify the details of the patients with their approval.

Jenkins build success report:



Jenkins

Jenkins

DEVACDMY-19588_BookAppoint

#1

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[View as plain text](#)

[View Build Information](#)

[Git Build Data](#)

Timesamps


☒ System clock time

☒ Use browser timezone

☐ Elapsed time

☐ None

[View as plain text](#)



Console Output

00:03:33 Started by user Kumar K.Arun

00:03:33 Running as SYSTEM

00:03:33 Building remotely on IPFACE0BLD83 (JAVA LINUX NODE RUBY) in workspace /home/devcenter/workspace/DEVACDMY-19588_BookAppoint

00:03:33 No credentials specified

00:03:33 Cloning the remote git repository

00:03:33 remote: Enumerating objects

00:03:33 Receiving objects

00:03:38 Resolving deltas

00:03:39 Updating references

00:03:39 Checking out Revision a0b269b59b59b2cd213701fac516b16590bbb68 (refs/remotes/origin/stable)

00:03:42 Commit message: "member variables in separate lines"

00:03:42 First time build. Skipping changelog.

00:03:42 Started calculate disk usage of build

00:03:42 Finished calculation of disk usage of build in 0 seconds

00:03:42 Started calculate disk usage of workspace

00:03:42 Finished calculation of disk usage of workspace in 0 seconds

00:03:42 [WS-CLEANUP] Deleting project workspace...

00:03:42 [WS-CLEANUP] Deferred wipeout is disabled by the job configuration...

00:03:42 [WS-CLEANUP] done

00:03:42 Finished: SUCCESS

Fig 5.13 Jenkins Build Console output

23

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

The core reason for the establishment of computerizing Doctor Appointment System is to enable the hospital administrators in a convenient, fair and timely manner. Therefore the IT used should support the core objective of the system if it is to remain relevant to the hospital. A lot still needs to be done in the IT department in order to make available technology effective. This may involve training of the hospital staffs on how to enter data in the right and relevant data in the system and the management to keep updating the hardware and software requirements of the system. IT and computer systems need to be kept being upgraded as more and more IT facilities software are introduced in to days IT market. The researcher acknowledges the fact this system does not handle all patient doctor and hospital. The researcher therefore suggests that for further research, the following can be researched on. The most cost effective way of handling all Hospital Patient management system process.

This project , if scaled to bigger application , can incorporate the storage of patients medical record (EMR) and the next time the patient comes , he/she doesn't have to bring the previous medical record , the stored data is sent to the doctor directly to his/her machine. And also a new window for doctors can be created where the doctors can update their availability themselves instead of letting the clerk know of their unavailability. With this we can achieve a seamless transition between appointment booking and transferring patient details to the doctors window.

REFERENCES

- [1] www.pluralsight.com for c++ tutorials
- [2] <https://docs.microsoft.com/en-us/office/vba/fl/> for any syntax related issues in MFC or any errors that occur while execution
- [3] <https://www.codeproject.com/> for any MFC related queries (errors, syntax)
- [4] <http://msgroups.net/vc.mfc/> is a community for MFC developement
- [5] <http://www.functionx.com/visualc/>
- [6] <https://www.tenouk.com/visualcplusmfc/visualcplusmfc17.html>
- [7] <https://stackoverflow.com/questions/> for any query related to errors.
- [8] <https://app.pluralsight.com/paths/skills/cplusplus>
- [9] <http://www.codersource.net/2010/02/14/c-ado-select-sample/>
- [10] <https://www.tenouk.com/Module4.html>
- [11] <https://www.geeksforgeeks.org/> for all c++ related issues.
- [12] <https://www.experts-exchange.com/questions/23136685/CRecordset-Open-throws-exception-querying-MS-Access-database-Too-few-parameters-Expected-1.html> for all MFC related issues.
- [13] <http://documentation.microfocus.com/help/index.jsp?topic=%2Fcom.borland.silkt est.install.doc%2FSILKTEST-413CCBCD-CREATINGDATASOURCEACCESSDATABASE-TSK.html>

- [14] https://www.youtube.com/watch?v=-FS_gVFH6BQ

- [15] <https://social.msdn.microsoft.com/Forums/vstudio/en-US/01d5bf12-fc7e-46fe-b939-a41dc6a9d1ee/>

- [16] <https://msdn.microsoft.com/library/en-us/vcsample98/html/samplemfcrowlist.asp>

- [17] <http://forums.codeguru.com/showthread.php>

- [18] <https://bytes.com/topic/net/answers/281717-coledatetime-problem-converting-date>

- [19] <https://access-programmers.co.uk/forums/showthread.php?t=206028>

- [20] <https://wiki.cerner.com/display/public/ConsultingControlledDocs/Custom+CCL+Coding+Standards+and+Best+Practices>