

University of Southern Queensland
Faculty of Engineering and Surveying

Theatre Booking System

A dissertation submitted by

Derek Pejakovic

in fulfilment of the requirements of

Courses ENG4111 and ENG4112 Research Project

towards the degree of

Graduate Diploma of Geomatic Studies (Surveying)

Submitted: October, 2006

Abstract

A computerised reservation system possesses numerous advantages over its manual counterpart, but full endorsement by smaller organisations might be hampered by the cost factor. This project examines the booking requirements of theatres, as well as the type of information required by their managers, and incorporates some of the desirable features within a Microsoft Access database.

A database has been designed to cater for a 200 seat theatre with a maximum of two performances per day. In order to keep costs to a minimum it makes use of a PC which is practically standard office equipment. The database has been tested, and suggestions on how the specific features of the system could be improved are made where appropriate. While the system processes manager requests effectively and provides the treasurer with many advantages compared to a manual version, more work is required for it to achieve a comparable level of patron orientated functionality.

University of Southern Queensland
Faculty of Engineering and Surveying

ENG4111/2 *Research Project*

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled “Research Project” is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Prof R Smith

Dean

Faculty of Engineering and Surveying

Certification

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Derek Pejakovic

Student Number: 0031222532

Signature

Date

Acknowledgements

This research was carried out under the principal supervision of Associate Professor Frank Young.

Appreciation is also due to the DocEx team of the USQ library and to my family.

Contents

Abstract	i
Acknowledgements	iv
Chapter 1 Introduction	1
1.1 Background	2
1.2 Research Objectives	2
1.3 Achievements	3
1.4 Conclusions: Chapter 1	3
Chapter 2 Literature Review	4
2.1 Traditional Theatre Booking Systems	4
2.2 Contemporary Theatre Booking Systems	6
2.3 Conclusions: Chapter 2	8
Chapter 3 Methodology	9
3.1 Choice of database	9
3.2 Database Construction	10
3.3 Database Structure	12
3.4 Queries	14

3.5 Query Construction	15
3.6 Direct Queries	15
3.7 Indirect Queries	18
3.8 Reports	19
3.9 Conclusions: Chapter 3	19
Chapter 4 Testing	20
4.1 Navigating through the Database	20
4.2 Query Testing	21
4.3 Reports	28
4.4 Chapter Summary	29
Chapter 5 Analysis	30
5.1 Benefits	30
5.2 Weaknesses	31
5.3 Conclusions: Chapter 5	33
Chapter 6 Conclusions	34
6.1 Suitability of the System	34
6.2 Further Work	35

6.3 Summary of Chapter 6	35
References	36
Appendix A Project Specification	38

Chapter 1 Introduction

From the onset of the computer age, the major Airlines recognised the efficiency a computer reservation system would bring to their operations. With costs being prohibitive to all but the largest of companies, the airline industry discarded their manual booking systems in favour of permanently utilising the electronic systems. Several decades later, computers are affordable to even the smallest of businesses, allowing them to enjoy the benefits of enhanced productivity using Information Technology (IT). The major theatres have set up computer booking systems to increase their efficiency, but smaller operations might be hampered to follow suit due to budgetary constraints.

This research investigated the construction of a computer-based *Theatre Booking System* appropriate to the niche theatre business. The completed system aims to enable it to be utilised by patrons willing to book theatre seats online in addition to theatre managers wanting to obtain various real time ticket sale reports. The designed system assumes that the theatre has a seating capacity of 200 and there are no more than two performance sessions per day for example, morning and evening.

A discussion of traditional and more modern booking systems is presented, with a proposal to make use of IT to design a system which is fast, convenient, informative, accurate and consistent with the modern day conveniences consumers have come to expect. The various components of the system will be tested and evaluated for efficiency, accuracy, convenience, ease of use and possible future developments.

1.1 Background

The financial centre of a theatre is called the *box office*. An ineffective box office can cause failure of the whole theatre (Schneider and Ford 1993; Grippo 2002), so its correct functioning is vital. The box office is operated by the *treasurer* who books theatre seats for patrons and sells them the appropriate tickets. The treasurer is accountable to the theatre manager, who requires frequent reports on ticket sales and finances. Since this project was concerned with booking systems, the box office was the prime area of focus.

For a small theatre to process 200 seat sales at different prices for each performance can be a very tedious undertaking. One misplaced ticket stub has the potential to cause havoc when accounting for performance sales. Appropriate computer use can reduce the time taken to accomplish these duties, as well as reduce the labour cost, rendering the theatre more competitive.

To prevent theatre sales from dwindling, booking seats and paying for tickets should be made as simple and convenient as possible. Numerous enterprises provide customers with the possibility of purchasing their goods and services online. Since consumers have come to expect such facilities, providing an online booking service can only serve to boost performance attendance.

1.2 Research Objectives

This researched involved:

- acquiring information on the functioning of various theatre booking systems through a literature review,
- designing of a 200 seat theatre booking system,
- computerising the system using Microsoft Access,
- testing the computerized system.

1.3 Achievements

The aim of the system proposed was to allow patron internet access to performance information, and enable them to book seats for the chosen performance, while theatre managers would be able to utilise it to obtain reports on ticket sales, and other information of interest. However, computerisation of the system proved to be far more difficult than had been anticipated, requiring at the least, a fairly substantial appreciation of database theory which could not be acquired within a reasonable time limit. While manager requirements were successfully catered for, difficulties were encountered which prevents the system from becoming patrons' *one stop shop* with regards to theatre bookings.

1.4 Conclusions: Chapter 1

While it is almost certain that smaller theatres can benefit from a computerised booking system, affordability is a factor which must always be considered by management. This research has contributed towards an eventual system that would offset the need for significant investment. The mechanisms of booking systems are examined in the next chapter.

Chapter 2 Literature Review

Introduction

This chapter examines several texts in order to establish the fundamentals of a theatre booking system. Several features that used to be processed manually have been incorporated in the database design.

2.1 Traditional Theatre Booking Systems

The theatre provides infrastructure and facilities for a performance to take place, while allowing an audience to enjoy this for a fee. A booking system is used to ensure patrons can purchase tickets for a given performance well in advance, and avoid being turned away at the last minute. Theatre management also prefer tickets to be sold as early as possible, both for financial reasons (Hillenbrand 2001), and to avoid a long queue at the entrance just before the performance is due to start. The theatre box office is the section where bookings are made, being the first point of contact between the theatre and the public (Schneider & Ford 1993; Grippo 2002; Hillenbrand 2001).

Pre-electronic era box offices consisted of a theatre ticket sales point where a patron could purchase one or more tickets and have some preference of seat location. The box office treasurer would have a paper plan of the theatre for each performance, with a corresponding ticket book. The patron would have access to a diagram of the theatre to indicate preferred seats, but usually not be allowed to actually see the treasurer's plan, as part of the treasurer's job is to distribute the audience around the theatre for both increased comfort (not all patrons in the same corner while many other seats are vacant), and to give the impression of a well-attended performance (even if the theatre was not full) for

publicity reasons (Reid 1983; Langley 1980). In the process, the treasurer would attempt to sell the premium (and most expensive) seats first (Langley 1980). Booking a seat and buying a ticket in this manner would be a face to face negotiation between the patron and treasurer. The treasurer was also expected to be knowledgeable about the performance in question, so that patron questions could be dealt with in a positive manner (Sweeting 1969). The patron would finally pay for the agreed seats, and receive tickets printed with the respective seat numbers. The treasurer would have to carefully mark off the seats on the theatre seating plan for every ticket sold, to avoid the possibility of the same seat being sold twice by mistake.

Performance seats were also sold by telephone or mail order to those patrons who could not readily pay a visit to the box office. Of this, mail order was the most tedious and time consuming method. In the most straight forward scenario, the patron would post a cheque with an order for the desired theatre seats, date and time, with a stamped, self-addressed envelope (Pick 1985). If the order had been received early enough, the box office would book the seats requested, and post the appropriate tickets back to the patron, who would receive them at least two days after the order was first sent. However, if the order had been sent after the desired seats had been sold, the box office would have to send the patron a refund, with a list of the remaining seats (Sweeting 1969, p. 206), advising the patron to re-order promptly. This process would then be repeated until the patron either obtained satisfactory seats, or would be sent a refund (Sweeting 1969, p.206) if all desirable seats had been sold out. Therefore, mail order could easily take up two weeks, and still result in disappointment! To simplify the mail ordering process, theatres also sold seats for a whole season of performances (Sweeting 1969; Pick 1985), in which case the patron would post a blank cheque to be completed by the box office, according to the cost of seats successfully reserved.

Inevitably, not all performances would have been sell-outs, and indeed, a certain percentage of ticket holders would not attend. In order to maximise profits, the box office would sell any unreserved seats at concession rates the final few hours before the performance (Pick 1985; Carmack Celentano &

Marshall 1998), and these would be clearly indicated on the ticket stub (which is retained by the box office) so as to avoid accounting problems at the end of the day (Langley 1980). Sales were sometimes deduced by counting the number of unsold tickets after the performance had begun (Collins 2003), but due consideration would have to be given regarding any discounted tickets sold.

The theatre manager would be concerned with various ticket sale reports, such as master statements showing the number of seats at each price and the total potential ticket income, daily reports (for the specific performance or performances for the day), weekly reports (giving a summary of sales for the week), seasonal box office statements (a summary of figures for the season) and season reports (the complete record for each performance in the series) (Langley 1980; Grippo 2002). Reid (1983) gives an example of a *Weekly Return* (which gives the seat occupation as a percentage and total money collected for each day of the week) and *Final Return* (which gives the total money collected in advanced booking, at the doors and percentage of seats occupied). Sweeting (1969) also gives examples of a *Weekly Summary* and *Final Return*, but these do not contain percentages.

So it is clear that the theatre manager made use of ticket sales reports for each performance, each end of day, each week in summary form and each series of performances, and that the reporting style varied from one theatre to another.

2.2 Contemporary Theatre Booking Systems

Although the fundamental principles of a booking system have remained unchanged over time, developments in communication and other technologies have had a phenomenal impact on their implementation. Sweeting (1969) speculated that perhaps theatres could benefit from computerised ticketing systems, similar to what airlines at the time had started to adopt. In the late 1960s, the personal computer did not exist. Computer systems were bulky, expensive industrial machines and would be set up to execute tasks specific to

the organisation in question, putting them beyond the economic reach of most, if not all theatres. Langley (1980) asserted that computer ticket systems were only feasible for large theatre operations. However, with today's comparatively low hardware cost, the computer has become standard equipment in virtually all offices. Even the smallest of theatres would opt for a computerised system, if only because the two week mail order negotiation described in the previous section can now be condensed to two minutes. Nevertheless, Hillenbrand (2001) cautions against a computer and software purchase for small operations, suggesting that a dedicated telephone answering machine be utilised. Reid (1983) observed that computers were replacing the traditional paper and pencil systems, allowing several telephonists to simultaneous access the same seating plan, thus avoiding a patron queue and improving on customer services, compared to a manual system. Hillenbrand (2001) proposes that theatres should match the conveniences provided by the numerous other service providers such as toll free numbers, credit card processing, internet access to products and services, mail order, all of which today's consumers have come to expect.

In addition to knowing about the performance, the modern day box office treasurer is expected to be able to give patrons advice on restaurants and parking, as well as operate it until well after normal working hours, so as to provide convenient opening hours to the public (Carmack Celentano & Marshall 1998).

Tickets may be paid for in advance, and either mailed to the patron, or kept at the box office to be collected when the patron arrives at the theatre for the performance (Reid 1983). The box office can make use of voice mail to reserve tickets for subsequent purchase, but should sell any unclaimed tickets fifteen minutes prior to the beginning of the performance. At least 10% of tickets reserved for a performance are not collected (Langley 1980, p.280), so reserving them while there is demand at the door would be unsound economics.

Since the theatre of today has to compete with a multitude of other established entertainment options available to busy patrons, marketing is as important as

ever. A number of complimentary tickets may be made available to the press, producer, etc, as a means of promoting the theatre.

Modern day theatre managers still make use of the same types of reports that were sought decades ago, but have come to expect them instantly, and without error, due to widespread computer usage. A computerised reservation system has the potential to provide the precise type of information required with just a mouse click. Comparing manual and computerised booking systems, Collins (2003, p. 125) quotes a theatre manager describing them as ‘labor intensive, bulky and hard to make changes’ as opposed to ‘fast, easy and sleek’ respectively.

2.3 Conclusions: Chapter 2

The fundamental theatre booking principles have not changed over time but with the arrival of the electronic age there is pressure from patrons and society in general to modernise traditional systems, so that theatres are perceived to be providing conveniences on par with other entertainment and service providers. The next chapter discusses construction of the system making use of the Access database package.

Chapter 3 Methodology

Introduction

The major requirements of a theatre booking system were reviewed in Chapter 2 and these have greatly influenced the discussions that follow about the design of the booking system. This chapter discusses how the database was set up in order to incorporate these features, as well as the difficulties encountered.

3.1 Choice of Database

In terms of convenience and speed, there is no doubt that a computer booking system is vastly superior to a manual one. However, some authors (Langley 1980; Hillenbrand 2001) discourage smaller theatres from using such a system due to their limited budgets. This booking system was designed with this constraint as a fundamental consideration, therefore it makes use of 'off the shelf' equipment, namely a PC running the almost standard Microsoft Office Professional suite (which includes Microsoft Access database). It is hoped that a small theatre which already owns a PC with Access installed would perceive this system as a cost effective solution to enhancing productivity. Also, a windows-based interface with the system would prove to be less intimidating to patrons willing to take advantage of an eventual online booking facility.

Ease of use was also an important consideration. Unlike word processors which most computer users are familiar with, database programs are comparatively rarely used and would require the box office treasurer to undergo more intense training. In the hierarchy of databases, Kroenke (2006) rates Access as an entry level product while Mannino (2001) associates PC databases with low cost and limited capabilities. However, it has built in features that can be set up (eg

forms, reports, etc.) designed to interface between the user and the database, such that the most common tasks can be executed with minimum effort by a non-expert user. This user friendly aspect of Access is another reason why it was selected for system construction.

Access also permits construction of web pages which allow an internet user (without necessarily having Access on his/her computer) to make changes on the database. This is an ideal scenario whereby a patron can view performance information online and click a box to enter his/her booking into the system, without having to know anything about databases. This database was constructed using Access 2003, which by default saves files in Access 2000 format to enhance backward capability, enabling computers installed with the older version of the software to open and use the file.

3.2 Database Construction

Kroenke (2006) states that the world is full of poorly designed databases. In order to avoid the more common pitfalls of database design, several texts on the subject matter were examined before commencing the booking system. One of the fundamental design rules is that databases should avoid repetition of the same data. Consider, for example, a table of performances showing dates:

Performance Name	Description	Performance Date	Performance Time
Goose Pond	Brisbane Philharmonic in action	12/12/06	pm
Goose Pond	Brisbane Philharmonic in action	13/12/06	pm
Goose Pond	Brisbane Philharmonic in action	14/12/06	am
Blue Tree	A play by Swedish performers	16/12/06	pm
Blue Tree	A play by Swedish performers	17/12/06	pm

This table should be broken down as follows:

Performance Name	Description
Goose Pond	Brisbane Philharmonic in action
Blue Tree	A play by Swedish performers

Performance Date	Performance Time
12/12/06	pm
13/12/06	pm
14/12/06	pm
16/12/06	pm
17/12/06	pm

In the latter configuration, the two separate tables avoid repeating the performance names and their respective descriptions. To view and manipulate data from the two tables, they have to be 'joined' in a one-to-many relationship

(Goose Pond has many corresponding dates in the second table). The addition of a serial number (which Access can do automatically) ensures that Goose Pond does not get linked to the wrong dates. In the same way, because each performance provides 200 seats, the date/time table is also joined to the seats table in a one-to-many relationship. A query is used to extract information from the joined tables, and they behave as if they are a single table.

Each table within the database stores information with related attributes. For the database to function properly, all the tables must be in the 'third normal form' (3NF), which involves a three step restructuring of the original table so as to identify and eliminate data anomalies (Adamski & Finnegan 2006). Powell (2006) asserts that not all database designers fully understand normalisation, so it is unlikely that a superficial study of this topic would lead to the ultimate design.

Leading to this database, several versions, starting with a 10 to 20 seat theatre were constructed, tested and refined before adding new tables and progressing to the next iteration. This method allowed proven features from the earlier versions to be implemented in the latest version. However, as the database became more complex, it was found that many features would no longer function as they would in simpler cases. For example, a graphical representation of the theatre seat configuration was devised showing seat numbers in their respective locations and corresponding check boxes using an Access form. To book, a patron or treasurer would just click the box of the desired seat, and the booking would also appear in the 'Seats' table. The idea had to be abandoned when it was discovered that Access can only cater for 20 seats (the maximum number of database fields allowed), while 200 seats were the requirement.

3.3 Database Structure

This database consists of 5 main and 2 supportive tables. The first table named 'Performance' contains information about the performances, such as the name

(abbreviated to PerfName, as is the convention in SQL for multi-worded field names) and description (PerfDescription). For convenience, the table also contains a Performance ID (PerfID) column. Using database conventions, this table can be written as:

Performance (PerfID, PerfName, PerfDescription).

PerfID is underlined because it is the 'primary key' of the table (it is never repeated within the table and therefore can uniquely identify every row). The other tables are:

Date (DateID, Date, Time, PerfID);

Seats (SeatID, SeatNo, DateID, Reserved, Cost, Discount%, Paid, TransactionDate);

Patron (PatronID, SeatID, PatronName, Address1, Address2, Postcode);

Orders (OrderID, PatronID, PerfID, PerfName, Seats, OrderTotal, MailTicket).

Certain fields, such as PerfID are included in more than one table for convenience where various tables need to be joined. The 'Performance' table is linked to 'Date' in a one-to-many relationship, and in turn, 'Date' is linked to 'Seats' in similar form. 'Patron' is also linked to 'Orders' in a one-to-many relationship, but no satisfactory relationship could be generated between 'Seats' and 'Orders,' as Access would only permit a one-to-one. (In a previous version of this file, and under seemingly 'identical' conditions, a one-to-many relationship had been set up). As a result, patron order information cannot be modified from within the 'Performance' table, in the way that the seating status can. The most likely cause is that not all these tables have been fully reduced to the 3NF, and therefore cannot make full use of Access functions.

The table 'Seating Plan 2' is not actively used in the booking process. To avoid having to repeatedly enter 200 seats for each show, an original 200 (labelled A1

to J20 and representing a 20 by 10 seating configuration) were typed into a table named 'Seating Plan.' 'Seating Plan 2' was derived by applying the Append Query on 'Seating Plan' 20 times, producing 20 seat sets for this same number of performances. In turn, 'Seating Plan 2' was used to populate the 'Seats' table with 4000 new seats per mouse click. For reasons discussed in 3.5, 'Seating Plan 2' is no longer being used in the system.

'Seat Cost' is a small table of ticket price figures, which can be amended as and when necessary. This table is used by the 'Cost' lookup column featured in the 'Seats' table, allowing the treasurer to pick an appropriate seat price from a list rather than typing it in each time.

3.4 Queries

Queries are used to extract information from the database, and perform calculations. For example, the treasurer can check for unsold seats for a given performance via a query, while the manager can obtain ticket sale statistics for the day, week, month or year. Tables can also be updated using queries, but there are limits to this in Access (Microsoft, 2003).

A special type of query known as the Append Query in Access can be used to add new data to an existing table. This was used to generate the seat sets as they appear in the 'Seating Plan 2' and 'Seats' tables. The intention was that for each new performance, the treasurer would add a set of 200 seats to the database with the click of a button. However, it was found that joining the 'Seats' table to the 'Date' table caused the Append Query to stop functioning. That is why it was necessary to pre-load the 'Seats' table with enough seat sets to last over seven years (assuming one performance per day, every day).

Using the Append Query to add 20 seat sets (4000 seats) at a time, it was discovered that at certain regular intervals, inexplicably, some seat sets contained 192 instead of 200 seats, followed by sets with over 200. Some of the

defective sets were identified and deleted, but due to time constraints, this anomaly remains beyond SeatID 30289.

The exact type and format of information requested by managers varies from one theatre to the next, so the queries in this database were designed to answer the most common questions its users are likely to have. However, it was sometimes necessary to design intermediate queries, not normally run by the user, but processed by yet another query to ultimately produce the required user information. The database remains ‘open’ in the sense that it is possible to add new queries to suit specific user requests while the system is in operation, and this is a normal part of a Systems Development Life Cycle (Mannino 2001).

3.5 Query Construction

With a few minor exceptions, Access is able to process commands from the standard database language, SQL. The queries contained within this database were built using both SQL and the Access query *wizard*. While SQL is text-based and requires exact punctuation and spelling not to return an error, the wizard provides a graphical design interface, facilitating references to table names and entering SQL commands. Quite literally, a mouse click would be the equivalent of several commands without syntax problems. The wizard provided rapid query construction, with greater Access compatibility. However, the automation involved resulted in some of the queries being too general, and these had to be fine-tuned to the specific requirements, by editing the code produced by Access in its ‘SQL view’ window.

3.6 Direct Queries

These are the queries summoned directly by the user in order to provide real time information about some aspect of the booking system. Since the various tables containing data are continually being updated, running the same query at

different times yields different results, and is therefore an ideal tool for keeping track of the system. The queries and their functions set up for this database are discussed below.

DISCOUNTED SEATS QUERY

It is not uncommon for theatres to sell performance tickets below the advertised price. VIPs get complimentary tickets, students and pensioners might be entitled to a discount while members of the public might obtain bargains in the final few minutes before the show begins. The theatre manager keeps track of these discounts, and the *Discounted Seats Query* allows him to see for which performance these discounts apply, which seats have been discounted, the discount percentage and the revenue collected.

MASTER STATEMENT

A Master Statement gives the manager the distribution pattern of ticket sales. This query prompts the user for a specific date and time (am or pm) for the performance in question and then displays a table showing the various ticket prices and how many have been sold at each price.

SINGLE PERFORMANCE POTENTIAL INCOME QUERY

Managers need to know what the potential income is for a given performance. This query prompts the user for date and time information, then adds the tickets sold to all the reserved, but as yet, unpaid seat prices. The amount displayed is the theoretical sales figure for the performance in question. Such information is useful in determining whether more marketing efforts are needed for a particular performance.

SINGLE PERFORMANCE SALES QUERY

This query prompts the user for the performance date and time, and then displays the total amount of ticket sale money collected to date, including discounted tickets.

TIME INTERVAL SALES

Managers require total sale figures for the day, the week, the month and the performance season. This query allows the user to generate all this and more by prompting for a beginning and ending date corresponding to whatever the manager requests. The query displays the total revenue collected within the date limits specified, irrespective of performance.

UNPAID RESERVED SEATS QUERY

In practice, a number of patrons who have reserved seats for a given performance will fail to attend. In order to maximise theatre profits, these reserved tickets are sold to patrons without a reservation in the final few minutes before the beginning. This query enables the treasurer to quickly verify which reserved tickets have not been claimed so that they may be sold.

VACANT SEATS QUERY

This query prompts the treasurer for the date and time, and then displays all unreserved seats for the particular performance. The treasurer can enter new bookings from the table that appears, and the 'Seats' table is automatically updated. A web version of this query could be used for patrons to book themselves for a performance, but they would have no option to assign themselves discounts or change the ticket price, parameters which the treasurer has control over.

WHOLE PERFORMANCE SALES QUERY

This displays financial quarter sales information, ordered according to the performance name. The manager can see how much revenue each performance has brought. In the case where a performance series spans two quarters (eg December 30, 2006 to January 5, 2007), the two entries displayed can simply be added together.

3.7 Indirect Queries

These are not meant to be run by the user, but rather provide relevant information to other queries. Often, not all the required information can be obtained in one step, so it is necessary to build a query which gathers data from other queries.

EXTENDED PRICE QUERY

This query calculates the final cost of a seat, taking into consideration any discounts. All queries that produce a financial output refer to this one in order to perform their own calculations, as they make use of the cost after discounts have been applied.

POTENTIAL SALES QUERY

This query searches for all reserved seats, whether they have been paid for or not. It is called upon by the *Single Performance Potential Income Query* which is described in section 3.6.

3.8 Reports

If necessary, all the above queries can be printed as they appear on the display. However, Access provides a Report tool which optimises a query's output for printing. Both the *Single Performance Sales Report* and the *Whole Performance Sales Report* provide the exact information that their respective similarly named queries do, but do so in a format ready to be printed. The manager thus has a choice as to which of these printout styles he prefers.

3.9 Conclusions: Chapter 3

Access was chosen for this booking system due to its low cost, wide availability and aspects of user-friendliness. Although facilities for patrons to book online could not be set up within time limits, a number of useful functions making use of tables, queries and reports have been built, which will provide both the treasurer and manager with quick, accurate answers in a convenient form. These are tested in the following chapter.

Chapter 4 Testing

Introduction

The structure of the database has been discussed in Chapter 3, and in this chapter, the various functions of the database tested have been described. The database consists of fictitious data for a generic 200 seat theatre. By making changes to the database content to simulate real world events, it could be verified whether the various queries and reports responded to the changes in the manner expected. Aspects of user-friendliness and skill level required by the user have also been considered for each one.

The tests which follow have been conducted and the outcomes documented. Should the reader wish to confirm the results, the file *Theatre bookings.mdb* must first be copied to a computer hard drive with at least 50MB of free space. Testing cannot be conducted directly from the accompanying CD-R, since Access automatically saves changes to tables as and when they are made, and this is not possible with this storage medium. As the reader gains familiarity with the system, the level of detail describing the steps involved in the tests have been gradually reduced.

4.1 Navigating through the Database

Once the file has been copied and opened in Access 2003, a small window labelled *Theatre bookings: Database (Access 2000 file format)* appears. The left hand side of this window has a heading called *Objects*, and under this are several selectable tab buttons, i.e. *Tables*, *Queries*, *Forms*, etc. When *Tables* is clicked, a list of the seven tables which constitute the database appear on the right (Date, Orders, Patron, Performance, Seat Cost, Seating Plan 2 and Seats),

and these tables can be opened from this window. Likewise, when the *Queries* tab is selected, a list of ten queries appears on the right, and they can be launched by double clicking the relevant query name. Note that the table names are no longer visible and cannot be opened from here. By clicking the *Reports* tab, only two reports are visible, but no tables or queries. By closing this small window, the *Theatre bookings* file closes.

4.2 Query Testing

The steps described below are for the benefit of a Windows user who may not be familiar with Access. The procedures involved quickly become routine and reference to the initial examples is advised if the latter tests are found to be inadequately explained.

TOTAL PERFORMANCE SALES

Click the *Queries* tab in the *Objects* column, and then double click *Whole Performance Sales Query*. A table appears showing the revenue collected for each performance according to the financial quarter. From this the manager can see the total money collected to date for the whole attraction (Jose \$100, Julio \$980). Any changes to the number of seats sold should affect this table, and this is verified below. The user needs only a very basic knowledge of Access to run this query. Click the close control for the *Whole Performance Sales Query: Select Query* window.

INDIVIDUAL PERFORMANCE SALES

With the *Queries* tab still selected, double click *Single Performance Sales Query*. The user is prompted for the date of the performance of interest. Type '1/12/06' then press enter. The user is now prompted for the time. Type 'am'

and press enter. Access now displays 180 under the heading 'TotalSales.' This means that so far, \$180 worth of tickets have been sold for the morning performance of 1/12/06. Click the window's close control.

We now enter a new sale for the very first performance. Click the *Tables* tab (next to *Queries*), then double click 'Performance.' The table is now open showing six performances, giving 'PerfID', 'PerfName' and 'PerfDescription.' Any new performance can be added to this table whenever it is required.

Click the + sign in the first row. The 'Date' table appears, showing 'DateID,' Date and Time for five shows. Click the + sign at DateID 1 (1/12/06 am). A list of 200 seats (A1 to J20) appears, with editable fields 'Reserved,' 'Cost,' etc. Most of the seats that appear here have not yet been reserved or paid for.

Click the 'Reserved' box next to seat B1. This books seat B1 for this performance. Click the cell under 'Discount%' and select 50 from the drop down list that appears. Seat B1 is being sold with a 50% discount. Click the 'Paid' box in the same row. As far as the database is concerned, the seat has now been paid for. Enter the current date (shortcut Ctrl ;) under 'TransactionDate.' Repeat the above instructions to pay for seat B2 with 0% discount (the full ticket price). Click the *Performance: Table* close control. Access has already saved the changes made.

Now both queries reporting the revenues for the first performance, and the quarter should increase by $40 \times 50\% + 40 = 60$, i.e. $180 + 60 = \$240$ and $980 + 60 = \$1040$ respectively. Click the *Queries* tab. Double click *Single Performance Sales Query* and type '1/12/06' followed by 'am' when prompted. Press enter or click OK. The total of 240 appears as expected. Click close. Double click *Whole Performance Sales Query*. The sales figure for 'Julio's Best' is now \$1040. Click close.

This shows that both the *Single Performance Sales Query* and the *Whole Performance Sales Query* respond to the changes made in the Seats table, and require minimal Access skills to use.

SALES OF DISCOUNTED SEATS

With the *Queries* tab selected, double click the *Discounted Seats Query*. A table appears showing which seats have been discounted to date (A1, A2, A11 and B1 for the 1/12/06 am performance). Click close.

We will now add two new discounted seat sales. Click the *Tables* tab. Double click 'Performance.' Click + next to PerfID 2 (Jose in Action). Click + next to DateID 10 (9/12/06 am). Click 'Reserved' next to seats A1 and A2. For both seats select \$25 under 'Cost', select 25 under 'Discount%,' click 'Paid' and enter the current date. Click close.

Select the *Queries* tab and double click the *Discounted Seats Query*. The table now shows seats A1, A2, A11 and B1 for 1/12/06 am, and A1 and A2 for 9/12/06 am. This shows that the query is updated in accordance with the changes made in the Seats table. Click close. This query is easy to use and quickly produces the required information.

THE MASTER STATEMENT

With the *Queries* tab selected, open (or double click) the *Master Statement* query. Enter '1/12/06' and 'am' when prompted. The table displayed indicates that 2 complimentary (100% discount) tickets have been issued, 2 have been sold at \$20 and 5 have been sold at \$40. Close the query window.

We now add some new sale information for this performance. With the *Tables* tab selected, open the 'Performance' table, expand PerfID 1 (by clicking +) and expand DateID 1 (click +). Scroll down this innermost table until seat D16 is visible (this is the row numbered 251 in the SeatID column). Reserve seat D16 by clicking the adjacent box, click on the blank cell in the 'Cost' column and select \$30 from the drop down list. In the same way, select 0 for 'Discount%,'

click the 'Paid' box and enter the current date (Ctrl ;) under 'TransactionDate.' Reserve and pay for seats D17 and D18 in the same way. When complete, close the *Performance:Table* window (the modifications are saved automatically).

Run the *Master Statement* query again (as described above), entering '1/12/06' and 'am.' In addition to the seats and prices described earlier, the query now yields 3 seats at \$30, which reflects the modifications made to the 'Seats' table. This query is simple to use, requiring the user to possess very basic Access skills.

POTENTIAL INCOME BY PERFORMANCE

With the *Queries* tab selected, open the *Single Performance Potential Income Query*. When prompted, enter '1/12/06' and 'pm.' The potential income, which is the total of all the revenue collected from sold seats and the sum of the value of all the seats that have been reserved, but not yet paid for, displayed is \$800. Close the window.

We will now modify the bookings for the second performance. Select the *Tables* tab, open the 'Performance' table, expand PerfID 1 and expand DateID 2 (1/12/06 pm). Set discounts to 0 for the reserved seats B1 to B10.

Even with the built-in drop down list, selecting 0 for ten seats is a bit tedious and is not very convenient or quick when reserving seats for a large group of patrons at a given time. Since most seats are sold at full price, a useful function would be to set the default discount rate at 0, reducing the treasurer's work load. Click close.

Running the *Single Performance Potential Income Query* (as per the instructions above), the value \$1200 is returned, and not the \$800 expected. This exposes a major inconvenience of the system, with the potential to generate errors. Unless a value is selected (or entered) in the 'Discount%' column of the 'Seats' table, the reserved seat is totally ignored in calculations.

Unlike in Microsoft Excel, the value of a blank numeric cell is not taken to be 0, so no calculation is performed. Again this inconvenience can be avoided if the default discount rate is set to 0. Click close.

If some of these reserved seats are now sold, the query should not report a new sales figure. Making sure the *Tables* tab is selected, open 'Performance,' PerfID 1 and DateID 2. Click the 'Paid' box for seats B1, B2 and B3, and enter the current date under 'TransactionDate.' Close the window. Running the *Single Performance Potential Income Query* entering '1/12/06' and 'pm' still gives \$1200, as expected.

We now reserve two new seats for this performance. Open the 'Performance' table, PerfID 1 and DateID 2. Reserve seats B11 and B12 (SeatID 406 and 407) and set their discounts to 50%. The potential income should now increase by $2 \times (40 \times 50\%) = \40 . Running the *Single Performance Potential Income Query* produces \$1240, which is the expected figure.

While this query is easy to execute providing the manager with a real time theoretical sales figure for the chosen performance, great care must be taken to ensure that for each unpaid reserved seat, a value is also entered in the 'Discount%' column of the 'Seats' table. Setting the default discount value to 0 would reduce the possibility of errors, and lessen the treasurer's work load.

SALES OVER A SPECIFIC PERIOD

When the *Time Interval Sales* query is run, the user is prompted for a beginning and ending date to define the time interval of interest. Suppose the manager requires the sales figure for the first week of August 2006. We open the *Time Interval Sales* query and enter the dates '1/8/06' and '7/8/06' when prompted. The total sales figure returned is \$980. Close.

We will now enter a new sale for this period. Open the 'Performance' table, PerfID 3 and DateID 11. Reserve seat A1, select \$25 under 'Cost,' set

'Discount%' to 0, click 'Paid' and enter 7/8/06 for the 'TransactionDate.' Repeat these instructions for seat A2, and then close the window. The sales figure for the first week of August 2006 should now be $980 + 50 = \$1030$ and this is confirmed by running the *Time Interval Sales* query with dates 1/8/06 and 7/8/06.

The revenue for a single day is obtained by using the same beginning and ending dates. Running the *Time Interval Sales* query and entering '1/8/06' both times when prompted displays \$820. Similarly by entering '1/1/06' and the current date, the sum of \$1473.50 is returned, being the revenue collected from January 1, 2006 to the current date.

If a patron decides to cancel his reservation and is issued a refund, this should also affect the total sales for the year. Open the 'Performance' table, PerfID 1 and DateID 1, and deselect 'Paid' and 'Reserved' for seat A7, deleting the transaction date to leave a blank cell. Close.

Running the *Time Interval Sales* query for the first week of August is found to be $1030 - 40 = \$990$ and the yearly sale to date is also reduced by \$40 to give \$1397.50. This query requires only basic Access skills to use, responds to both new sales and refunds as they are made and is extremely flexible regarding time intervals. However, it might be made more user-friendly if a drop down list of 'standard' time intervals, such as day, week, month, etc. could be incorporated at the prompting stage, while still allowing the manager any awkward time interval entry.

LAST MINUTE SEATS

The *Unpaid Reserved Seats Query* shows the treasurer which seats have been reserved, but not claimed for the particular show. Open the *Unpaid Reserved Seats Query* and enter '1/12/06' and 'am' when prompted. A list of 12 reserved seats appears. Suppose seats A3, A4 and A5 are now sold to another patron

because the performance is due to begin in 30 minutes. The 'Paid' boxes are clicked for these seats, and the current date entered. Click close.

When the *Unpaid Reserved Seats Query* is run again, seats A3 to A5 no longer appear on the list. Furthermore, checking the seat status for the very first performance (open 'Performance,' PerfID 1 and DateID 1) reveals that these seats have indeed been sold, and the Transaction Date is the current date. Changes made in this query are automatically updated in the 'Seats' table. The query is easy to use and provides the treasurer with a quick, convenient path to sell tickets that would otherwise go to waste. If these seats need to be sold at a discount, the query also allows this to be incorporated in the sale. Close the window.

AVAILABLE SEATS

Running the *Vacant Seats Query* for 1/12/06 am displays a list of 177 (indicated at the bottom of this window) unreserved seats for this show. We now reserve seat A6 (without paying for it), setting the discount to 0 and entering the current date. The same is done for seat A7. Close the query.

Running the query again, we observe that there are now 175 remaining seats, with row A of the theatre being entirely booked. Opening the 'Performance' table, PerfID 1 and DateID 1, it can be seen that seats A1 to A20 have been reserved.

The *Vacant Seats Query* has the ability to update the 'Seats' table directly, simplifying user interaction with the database, and requires a minimal Access skill level. This query would probably prove to be one of the most fundamental to the treasurer as it provides the remaining seat status for a performance, which he must be aware of whenever making a fresh booking.

4.3 Reports

As described in Chapter 3, an Access report simply presents the information obtained from a query in a printer-optimised format. It does not provide any editing facilities, but should change in response to modifications of the database.

THE SINGLE PERFORMANCE SALES REPORT

Select the *Reports* tab and double click *Single Performance Sales Report*, entering '1/12/06' and 'am' when prompted. A print preview appears (which can be magnified by clicking on the window) displaying Total Sales of \$450. Close the window. Running the *Single Performance Sales Query* for 1/12/06 am also produces \$450.

We now add a new sale to the very first performance. Click the *Tables* tab, open 'Performance,' open PerfID 1 and then DateID 1. Click the 'Paid' box for seats A12 and A13, entering the current date. Click close. The total sales figure should now be $450 + 80 = \$530$. Previewing the *Single Performance Sales Report* with '1/12/06' and 'am' confirms this. This report is easy to use, providing the manager with an instant hard copy of ongoing sales.

THE WHOLE PERFORMANCE SALES REPORT

When the *Whole Performance Sales Report* is previewed, an alphabetical list of performances is displayed, with corresponding sales figures according to the financial quarter. Note that the performance Goose Pond has accumulated \$50 in quarter 3 of 2006. We will now add a sale to one of its shows.

Click *Tables*, open 'Performance,' PerfID 3 and DateID 11. Reserve and pay for seats A3 to A6, setting the cost at \$25, discount at 0 and the current date for the

Transaction Date column. Click close. \$100 worth of sales should appear for Goose Pond in the current quarter. Opening the *Whole Performance Sales Report* confirms this. The manager can now deduce that the total revenue collected over the two quarters was \$150. This report is simple to use and print, and provides the manager with a hard copy of sales figures for all the attractions. Over time there will be multiple such pages, in which case the manager can choose only the relevant pages, using the print dialogue box provided by windows, in the usual way.

4.4 Chapter Summary

This chapter has explored the functioning of the various tables, queries and reports of the database, and has exposed several problems and inconveniences associated with certain of them. The tests that were conducted were not exhaustive, as they made use of reasonable, straight-forward information requests, since it is not possible to cater for the endless deviations from the norm which can be expected in real-life operating conditions. The advantages and deficiencies of the system with proposals to correct them are discussed in Chapter 5.

Chapter 5 Analysis

Introduction

The various queries and reports of the database were tested in chapter 4. In this chapter, the system in general is analysed, with its features being classified into what works to a satisfactory level and what needs improving.

5.1 Benefits

USEFULNESS TO THE TREASURER

Through the use of queries, the system provides the treasurer with rapid seat status information. Once they have been run, the queries also permit instant changes to be made (additional bookings and payments as well as cancellations and refunds) which are automatically updated on the 'Seats' table. The system also has the ability to quickly determine which seats have been reserved and unclaimed, so that they may be sold just prior to performance commencement, contributing towards maximising the theatre's revenue. The task of ticket sale tallying can be done with barely more than the click of a button, eliminating the need to count all ticket stubs collected, provided of course, that all sales were entered correctly into the system. The database does not diminish the need for the treasurer to be accurate, but reduces the scope for errors with automated calculations.

As its functions are easy to use, even though Access is not the most widely used program of the Microsoft Office suite, a treasurer would not require much training in order to operate the system. To further reduce the potential

requirement of being familiar with database conventions, Access allows field names to be renamed using captions. For example, the field named *TotalSales* from the *Single Performance Sales Query* (written using a technique employed in programming languages where two or more words need to be entered in a position where only single words are accepted) can be renamed *Total of Sales* or *Total Performance Sales* to give a more descriptive heading.

USEFULNESS TO THE MANAGER

Managers require a variety of financial output, such as the total sales for the day, week, whole season, etc. The system is capable of calculating all these figures, given the beginning and ending dates, using the *Time Interval Sales* query. Other queries provide rapid responses to the state of ticket sales for a particular show, or the total sales figure of a whole series of performances, or a list of discounted tickets sold, amongst other information of managerial interest.

While the queries displayed can be printed, there are two reports which present and print the relevant sales information, so a hard copy can be maintained. These can help consolidate the other accounting processes the manager might have.

5.2 Weaknesses

In constructing the database, difficulties were encountered while setting up some of the relationships required between tables. As a result, patron details, their preference as to whether to receive the tickets purchased by mail or to have them kept at the box office for collection and other order information, cannot be entered while bookings are being made in the 'Seats' table. The treasurer would have to close the 'Performance' table, and then proceed to open the 'Patron' table to enter the required information, which is not an efficient method.

One of the primary considerations in opting for a modern day theatre booking system is to offer patrons the choice of booking online. Access allows its tables to be updated via a *page* which is an HTML document saved for browser viewing and modification. Unfortunately, no such functioning interface could be implemented within time limits. Without this, the database has no *internet ready* capability, and consequently, no online booking facility.

Traditionally, the treasurer has made use of a paper theatre seat plan to mark off booked seats for each performance. An electronic version of the plan would complement the electronic booking system, facilitating deletion of cancelled bookings. Also, seats could be selected from a graphical representation of the theatre displaying a spatial perspective, rather than a long list of seat numbers from the 'Seats' table, conveying a better sense of position to patrons who book online. It is not obvious if this can be accomplished within Access, or whether a supporting program would have to be written to provide the necessary graphic display.

Entering data in the database tables can be a repetitive process, for example, in the case of seat prices and discount rates. Additional automation would help streamline the data entry process. With fields such as 'Discount%' in the 'Seats' table, setting the default discount rate to zero would be helpful, since most seats are normally sold without discounts, leaving only a few that would require treasurer intervention. Seat cost would be more difficult to automate since not all seat rows in the theatre are equally priced, and ticket prices fluctuate as attractions change.

The airline TWA used to discard the records of its reservations 24-48 hours after the flight had departed (Gifford & Spector 1984, vol. 7, p.654) to free up as many computer resources as possible. With today's aviation security concerns and the relatively low cost of computing power and storage, post flight reservation records are probably retained for much longer. This theatre booking system has no provision for deleting old records, the accumulation of which will

slow down the system, frustrating both the treasurer and patrons attempting to book online. A macro (which is a sequenced package of commands built to automate tasks) could be devised to delete all records beyond a certain age as deemed reasonable by the theatre.

5.3 Conclusions: Chapter 5

The manager and treasurer can obtain instant outputs from this database for many of their routine information requests. The treasurer's work load can be further reduced with certain improvements regarding data entry. The lack of a functioning web page linked to the database prevents it from allowing patrons to make their own bookings online. The structure of the database tables may prevent it from attaining a more comprehensive level of performance. Proposals are made in the next chapter as to how the system might be improved.

Chapter 6 Conclusions

Introduction

Analysis of the theatre booking system in the previous chapter has highlighted its strengths and weaknesses. In this chapter some concluding remarks are presented, with recommendations for further work with a view to improving the system.

6.1 Suitability of the System

Although it could not be fully demonstrated in this case, from the work discussed in the previous chapters, it can be concluded that a 200 seat theatre booking system can be built using an Access database. It is imperative that database design principles be implemented to ensure that all inter-table relationships be correctly and adequately established, so that all Access features can perform without restriction. A suitable graphic representation of the theatre seats might be beyond the capabilities of Access, and therefore not form part of the system, so a traditional paper plan would still have to be used.

As it makes use of a common, low cost database package, with the prospect of rapid and accurate information output, combined with online booking capabilities, smaller theatres with limited budgets would have much to gain by implementing a completed version of the system.

6.2 Further Work

It is obvious that this theatre booking system is far from being fully functional. One of the fundamental steps towards this goal is to have all the database tables linked by the appropriate relationship, which in turn requires that all tables be normalised. A substantial understanding of database table design principles, namely *normal forms* and *functional dependency* would be required, and would most certainly result in the re-structuring of the tables.

Setting up a functioning Access web page would also require some effort. Once the database can be modified from a *page*, this can simply be posted on the theatre's web site, allowing online users to book their seats for the chosen performance.

6.3 Summary of Chapter 6

For a fully functional theatre booking system, further work is required with regards to database table design, as well as Access web page design. The queries and reports which have been found to be easy to use and effective should be maintained, although they would have to be modified as the table structure is adjusted. Efforts to streamline otherwise effective procedures should be maintained. Only then would the system become suitable for a trial real life implementation.

References

Adamski, JJ & Finnegan, KT 2006, *New Perspectives on Microsoft Office 2003*, 2nd edn, Thomson Learning, Inc. USA.

Carmack Celentano, S & Marshall, K 1998, *Theatre Management*, Players Press, Inc., USA.

Cavanaugh, J 1973, *Organization and Management of the Non-Professional Theatre*, Richards Rosen Press, Inc., New York.

Cohen, GP 2003, *The Community Theatre Handbook*, Heinemann, Portsmouth, NH, USA.

Collins, T 2003, *How Theatre Managers Manage*, The Scarecrow Press, Inc, Lanham, Maryland, USA.

Gifford, D & Spector, A 1984, 'The TWA Reservation System', *Communications of the ACM*, vol. 7, no. 7, p. 654.

Grippio, C 2002, *The Stage Producer's Business and Legal Guide*, Allworth Press, New York.

Hillenbrand, M 2001, *Produce Your Play Without a Producer*, Smith and Kraus, Inc, Hanover, NH, USA.

Kroenke, DM 2006, *Database Processing Fundamentals, Design and Implementation*, Pearson Education, Inc, New Jersey.

Langley, S 1980, *Theatre Management in America Principle and Practice*, Drama Book Publishers, New York.

Mannino, MV 2001, *Database Application Development and Design*, McGraw-Hill Higher Education, New York.

Microsoft Corporation 1992-2003, Microsoft Office Access 2003 SP1 help files.

Powell, G 2006, *Beginning Database Design*, Wiley Publishing, Inc., Indianapolis, Indiana.

Pick, J 1985, *The Theatre Industry*, Comedia Publishing Group, UK.

Raymond, C 1999, *Essential Theatre: The Successful Management of Theatres and Venues with Present the Performing Arts*, Arts Council of England, London.

Reid, F 1983, *Theatre administration*, A&C Black (Publishers) Ltd, London.

Schneider, RE & Ford, MJ 1993, *The Well-Run Theatre*, Drama Book Publishers, New York.

Sweeting, E 1969, *Theatre Administration*, Sir Isaac Pitman and Sons Ltd, London.

Appendix A Project Specification

University of Southern Queensland
Faculty of Engineering and Surveying

PROJECT SPECIFICATION

FOR: **Derek Pejakovic**

TOPIC: Theatre Booking Management

SUPERVISOR: Associate Professor Frank Young

SPONSORSHIP: Faculty of Engineering and Surveying

PROJECT AIM: Develop and evaluate a seat booking system for a theatre for patrons to access performance information and make bookings and managers to monitor ticket sales and other statistics.

PROGRAMME: **Issue B, 12th April 2006**

1. Literature review on manual and electronic systems: Research how typical booking systems function, identifying the crucial as well as desirable aspects that can contribute to patron satisfaction and are important to theatre managers.
2. Formulate a booking system incorporating the crucial features from both perspectives, as well as any desirable ones that may enhance the system.
3. Computerise the system in the context of a small, two session theatre.
4. Test and evaluate the system.
5. Submit the Project dissertation.

AGREED:..... (student).....(supervisor)

12/04/ 2006

..../..../ 2006