

Software Systems Development

EVENTS UNLIMITED

Alicia Smith
2017/18

Contents page

Page	Contents
3	<u>Analysis</u>
4	Background
5	Current Problems
6	Development Tools
7	Project Management – Gantt Chart
9	Project Management – PERT Chart
10	Project Management – Development Diary
15	User Requirements
16	Hardware and Software Requirements
17	Aims of the Project
18	<u>Design</u>
19	System Flow Diagram
20	ER Diagram
21	Data Structures Dictionary
23	Validation
26	Normalisation
27	Program Overview
28	Form Sketches
36	System Components
38	Class Diagram
43	Pseudo Code
47	Report Design
48	<u>Testing</u>
49	Types of Testing
50	Test Plan
63	Acceptance Testing
67	Testing
78	Corrective Action

79	<u>User Guide</u>
80	Installation Guide
82	Hardware and Software Required
83	Operating Instructions
90	Backup Procedures
94	<u>Evaluation</u>
95	Testing Evaluation
96	User Requirements Evaluation
97	Aims of Project Evaluation
98	Development Tools Evaluation
99	Project Management Evaluation
100	Personal Performance Evaluation
102	
103	<u>Appendix</u>

Analysis

Background

Events Unlimited is a successful business which began as a small enterprise, initially offering only a catering facility run by Doreen McAllister. The demand for catering rose from family events and parties to business events and conferences. The success of the catering has resulted in the realisation that Doreen must develop more efficient administrative practices. At busy times the administration and recording of orders became quite lax, with details being regularly omitted in the order book.

Doreen then expanded from basic catering to on-site catering which involved the hiring of part-time staff. Staff availability had to be ensured and the logistics carefully planned; information such as items returned, items damaged, and items lost all had to be carefully checked upon event completion. The new catering facility was extremely successful, but the extra work and meticulous checks placed an additional strain on the business.

The next expansion for Events Unlimited was a themed party service, piloting the new venture with the organisation of children's parties. On top of catering, many customers wanted activity based entertainment, including activities such as bowling, swimming, baking and clay modelling. Other customers desired entertainers for their events, which involved Doreen researching appropriate artists and commissioning their services. This new development meant that Doreen needed to maintain records for a wider range of resources, and while she succeeded in satisfying customers' requests, she struggled to manage the associated work.

The final expansion stage of Events Unlimited was wedding planning – Doreen's ultimate goal. Over the last year the expansion has exceeded her expectations, but it has confirmed for her that she needs professional help to assist in business management. Due to the wide range of wedding facilities and the significant increase in work load, this part of the business has become extremely demanding.

Events Unlimited currently offers basic catering services, on-site catering, themed parties and wedding planning. Managing all aspects of the business is extremely difficult and Doreen needs a complete review of her administrative practices and systems; she requires a system which can be built with various separate sub systems, including: catering; on-site catering; themed parties; and weddings.

Current Problems

Doreen is aware of the administrative deficiencies of Events Unlimited and ultimately she would like a computer system which would incorporate all individual aspects of her business, which can be managed separately, but linked with an overarching structure. The main issues with Doreen's current system are the logistics of recording information and the overall efficiency of the business. Doreen has an excellent professional reputation and wants to project the best image possible.

Catering

Customers placing orders over the phone were required to pay by credit or debit card, card receipts should then have been stapled to order receipts. If this did not happen then unfortunately some customers were asked for full payment again upon event completion.

All orders and full details were recorded in an order book, but with the volume of work increasing, details were often accidentally omitted. This could also lead to customers being asked for payment numerous times, or having to provide personal details on repeated occasions.

Another aspect of administration which raised many issues was the orders not being in chronological order, resulting in staff having to constantly check the order book. On some occasions Doreen has been overbooked and this led to her struggling to complete up to three events in a single day

Stock ordering was haphazard, and Doreen soon realised that she could be saving much more money if she had ordered certain stock online, however she did not have the facility to do this. She requires a detailed and prioritised list of suppliers in order to keep stock updates consistent.

On-site catering

It was difficult to manage additional stock and resources, resources had to be recorded on a resource sheet upon arrival at event meaning that mistakes could have easily occurred, or details omitted. This made it impossible to keep an accurate inventory.

Extra staff were required to staff the parties rather than just prepare catering services, scheduling staff correctly proved to be difficult.

Themed parties

Doreen relied on a written list of entertainers, which was not practical. Organising this type of party meant that she had to maintain records for a whole range of resources, from entertainers, venues, costume and prop providers and script providers

Staff has to complete their own time sheets which were collected and logged into a computer system once per week, time sheets could easily be completed incorrectly or lost, meaning wage calculations would be wrong.

Weddings

Doreen needs a set list of suppliers and products which she can easily refer to and reorder from, currently she sources serviced and products for each individual customer, but this method would save time

Having a finite list of certain items such as cars and hotels would be beneficial to the business also, this would not only establish a professional network with other businesses but also simplify this service for Doreen and her staff.

Development Tools

I have used two different methodologies to create my project; RAD (Rapid Application Development) and the Waterfall Methodology.

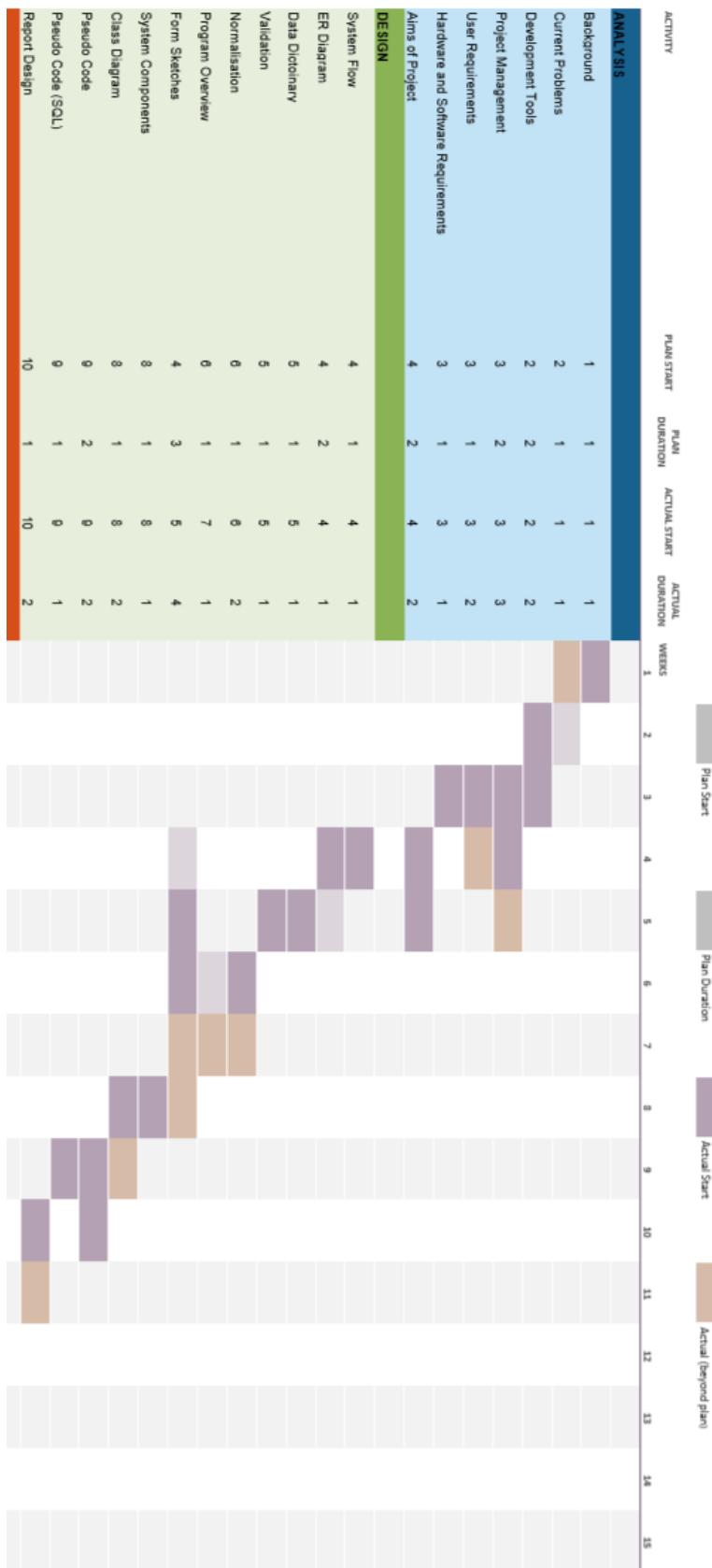
Firstly, one methodology I have chosen to create my sub system is Rapid Application Development which is an iterative process. An iterative process means that it cycles through iterations of design and development until either the project is completed or, time and/or money run out. The aim of RAD is to create software products at increased speed (6 months or less) and with improved quality due to higher user involvement. RAD is less concerned with setting requirements at the beginning and focuses on the development of the application based around the prototype, which is known as the interface model. It is flexible, and application can change based on new or extended requirements uncovered during workshop phases of iteration, this factor makes RAD very suitable for systems that are interface heavy. The advantages of using RAD are that there is reduced development time due to automation tools and use of reusable components, there is also better quality of software due to the on-going involvement of the user and the early detection of errors. Furthermore, due to the level of user involvement, there is a reduced risk of user rejection of the finished project. Although the RAD methodology is very suitable for my project, I do recognise that there are some disadvantages to the method. For example, there is less control of the project due to malleability of it and in some cases, there can also be poor design of the overall project as developers focus more on the prototype which can lead to less time spent on the overall architecture and functionality of the project. Overall, I have chosen this methodology to implement my project as I had a time constraint to develop the sub system and RAD is very suitable for form driven systems like mine.

The other methodology I used to implement my project is the Waterfall Model, which is a traditional software development methodology which involves using rigid procedures and requires deep and precise up-front planning. In this method, each phase is fully completed before the next phase is started and it is possible to move back through the phases if problems occur, though the methodology is mainly in one direction. The waterfall model is very simple to understand and use and is mainly used for projects that are small and there are no specific requirements. The advantages to this method are that it is easy to stay on schedule and stay within budget, there is also a clear understanding of what has to happen at each phase and the model is easy to understand and use. The management is also easier than many other methodologies due to the rigidity of the model and because model phases are processed and completed one at a time. I recognise the negatives to the method such as, if the requirements for the project change, you have to restart the whole project and if problems do occur it is difficult to move back through the phases. Furthermore, if mistakes are made in the early phases of the project, they can be carried on through to the end project and it very difficult at that stage to fix the problems. There are also high possibilities of risks and uncertainty due to the limited user involvement and ultimately it is not a good model for long and on-going projects. Although I am aware of the disadvantages of using this method, I still believe it was one of the best methodologies to use for my project as it is not a long and on-going one, I have a full understanding of the requirements and they unlikely to change as I have used a mixture of the RAD and Waterfall methods which entailed a high level of user involvement from Doreen therefore reducing the likelihood of risk and uncertainty from the project overall.

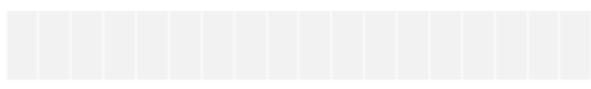
Project Management

Gantt Chart

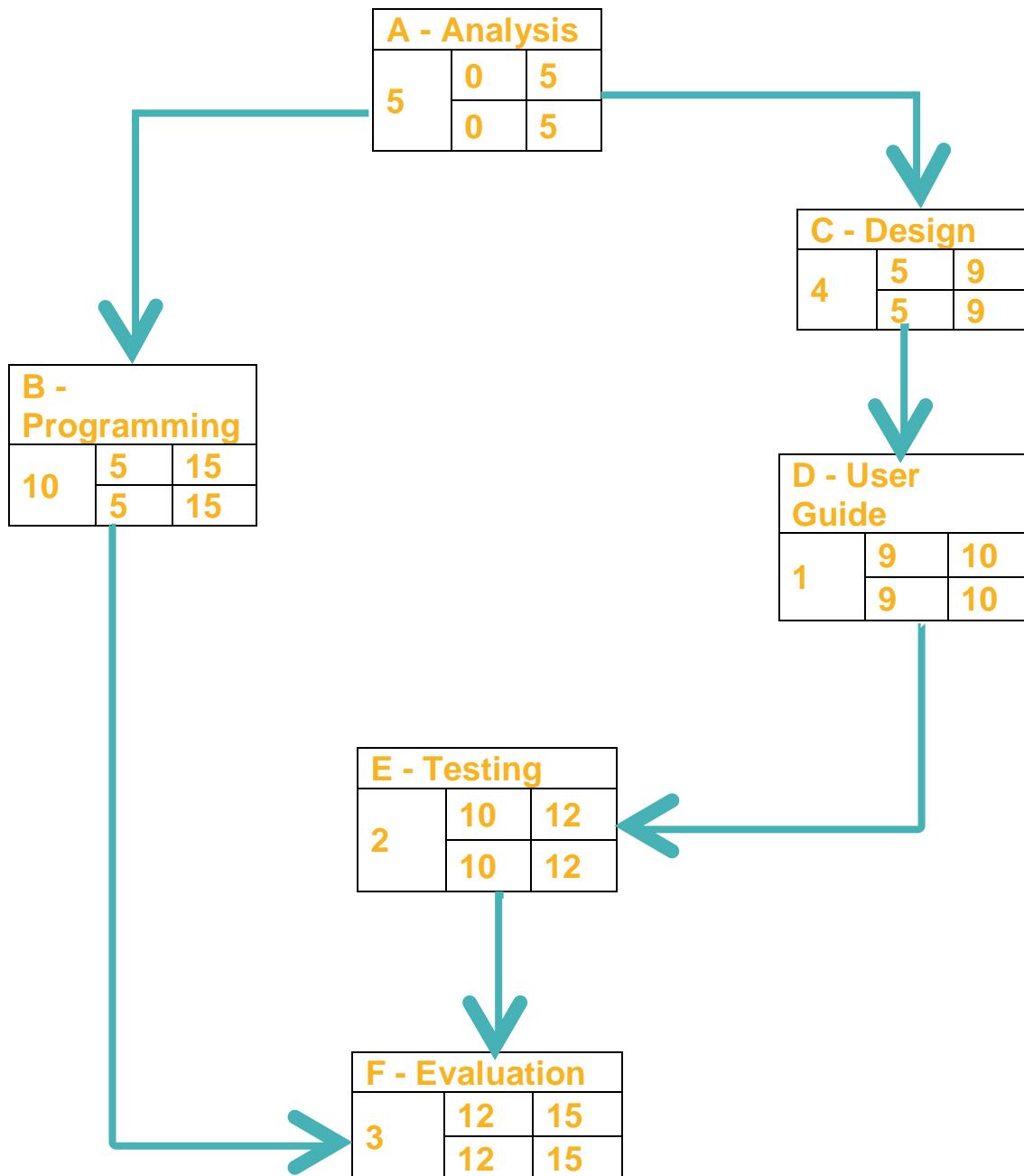
Events Unlimited



Pseudo Code	9	2	9	2
Pseudo Code (SQL)	9	1	9	1
Report Design	10	1	10	2
TESTING				
Types of Testing	10	1	10	1
Test Plan	10	3	10	4
Acceptance Testing	11	2	11	1
Corrective Action	11	2	11	3
USER GUIDE				
User Guide	12	2	12	3



Pert Chart



Development Diary

	To Do	Summary
Week 1:	Background	I completed my investigation into Events Unlimited within my estimated time frame and gathered a lot of information regarding the issues within Doreen's business.
Week 2:	Current Problems	In order to get a better idea of the issues with Doreen's business, I spent some time carefully dissecting her business as a whole, and into various sub-systems.
	Development Tools	During this section I aimed to research various methodologies, both Agile and Waterfall methods, in order to determine which would be most beneficial to my project's needs. I decided to use both the Waterfall model and RAD as I think they are best suited to my project. I completed this section on time and am content with the result.
Week 3:	Project Management	For this section I created a Gantt Chart, Pert Chart and Project Diary. However, I struggled with anticipating correct timescales, often having to recreate plans due to being too unrealistic about deadlines and the quantity of work I could complete.
	User Requirements	In order to determine the User Requirements, I outline the current problems in the business in more detail and tried to relate them to a user, rather than a business as a whole. I also tried to analyse the problems and determine solutions which then influenced which sub system I would try to tackle in my project. Due to Week 3 being particularly busy, this section taking some time longer than expected and as a result ran into the beginning of Week 5.
	Hardware and Software Requirements	I outlined the minimum hardware and software requirements for my system. I determined the specifications hastily and feel like I could have made the requirements more thorough.
Week 4:	Aims of project	The identification of user requirements ran on to week 4 as I had to identify more

		functional user requirements such as a score counter.
	ER Diagram	I had initially anticipated spending more time on creating my ER Diagram so that I could amend it as I developed. However, I completed during Week 4 and was happy with the outcome.
	System Flow	I found creating my System Flow Diagram to be quite straightforward and therefore didn't spend too much time on it.
Week 5:	Data Dictionary	I found this section to be particularly tedious as it was quite forthright and monotonous, yet this task didn't take up too much time, meaning I had it finished well within the week.
	Validation	Similar to the Data Dictionary section, I found this task rather monotonous yet straightforward and had it completed in the week as planned.
	Form Sketches	From the beginning of the project I had been planning and deciding on form designs, however I planned to begin the task of finalising them in Week 4 – but due to the volume of other work which I hadn't anticipated for, the start of this task was delayed and took one week longer than I expected.
	Normalisation	The normalisation of my project as well as the written evidence took a week longer than I anticipated. However, I am glad that I dedicated extra time to this section in order to identify issues and rectify them.
Week 6:	Program Overview	This task involved me simply creating a program overview diagram which will show the basic components of my system. This was a small task and I completed it well within a week, however due to my Normalisation section carrying through an extra week, I was unable to start this task until week 7, while I initially intended on starting and completing it in week 6.
	System Components	For this task I had to break down the various components of my system into

		more detail. This was a straightforward task which I completed on time.
Week 9:	Class Diagram	In order to create a class diagram I had to analyse the aspects of my different forms, identifying the various methods and variable I utilised. I began this task in week 8 and while I initially intended on completing it within a week, the task ran on slightly.
	Pseudo Code	I struggled slightly with this task as I was unsure of what to actually include in my pseudo code, at time including too much detail or actual code at points, and at other times writing pseudo code which was far too vague. Despite this issue, I got the task finished over the two week time frame I allotted.
Week 10:	Pseudo Code (SQL)	As I mentioned above, I struggled to write pseudo code as I was unsure of the correct details to include and such was also true for the SQL pseudo code. However, I also managed to complete this task within the correct time.
	Report Design	For creating my report designs I allotted a week, however I ended up spanning this task over two weeks due to indecisiveness and frequent revisions to my design as I wanted to maintain my aesthetic theme without losing professionalism. The overrunning of this project didn't really affect my development with other tasks as I was able to compete it between other sections
	Types of Testing	In this section I outlined the types of testing I would perform on my system, discussing details of who should perform the tests and what the tests entailed. I began this task on week ten and finished it promptly within a week as I found it to be rather straightforward.
	Test Plan	Devising a test plan involved me analysing the program I created and outlining what parts of my program I should test. I intended on spending three weeks on this task; however I found it to be quite difficult as many of my forms contained many textboxes of the same nature, as well as the same events and functionality, it was therefore very repetitive and monotonous - as a result this task began to run on into another week so I decided to thoroughly test one of the repeating forms and any

		differenced between the forms. (See test plan item 3)
Week 11:	Acceptance Testing	For this section of the testing process I devised two different acceptance questionnaires, tailored towards a business manager and a system user. In order to complete these I gave each potential user my system and asked them to complete the questionnaire in regards to my system. I initially allotted two weeks to this task to allow for the potential users to take time to assess my system, however, this task turned out to only take one week.
	Corrective Action	At this point, my project was beginning to become backlogged due to issues which arose from my program. I assigned two weeks to the completion of my corrective procedures of my program, however as I previously mentioned I was having issues with my program development, resulting in this section instead spanning over three weeks.
Week 12:	User Guide	For this section I aimed to create a straightforward guide to using my program, which contained all basic functional aspects. The aspects which I included were an installation guide, the required hardware and software, operating instructions, troubleshooting advice, and backup procedures. This task took longer than I anticipated, as this task was relatively straightforward I spread it out over three weeks as opposed to two in order to get other work complete.
Week 13:	Testing Evaluation	I began this task on time, however I spent less time than I intended, spending only one week on this section rather than two. I found this task to be quite straightforward, and prioritised other aspects of my evaluation section.
	Project Aims Evaluation	This section involved me assessing my fulfilment of both the functional and non-functional user requirements which I outlined in the analysis section of my documentation. I intended on spending a single week completing this task, however this week turned out to be very hectic and this section of my documentation spanned over two weeks as a result.

Week 14:	User Requirements Evaluation	I intended on beginning this task during week 13, however, this week proved to very hectic and I was unable to start this section until the following week. Despite the delay, this task didn't take very long and I managed to complete it within a week.
	Development Tools Evaluation	For my evaluation of the development tools I implemented, I discussed both RAD and the Waterfall Model, outlining the pros and cons of each and how they affected my development. I completed this task well within a week as I found it to be rather simple as I had a well-grounded knowledge of both methodologies before I began the project and it was therefore easy to evaluate their implementation.
	Project Management Evaluation	I initially intended on spending two weeks on this task, however managed to complete this in under a week. I found this evaluation to be quite difficult as when I initially wrote it, I had missed including the positive aspects of my project management skills, instead focussing on the negatives.
	Personal Performance Evaluation	Due to completing the evaluation of my project management skills a week early, I was able to start this task a week before I intended and complete it a week early also. Much like the project management evaluation, I struggled with this task as my self-evaluation was very critical and the harshness of my evaluation was not symbolic of the work and effort I put into this project.
Week 15:	Further Developments	This task required me to reassess Events Unlimited, and to consider which aspects of the business I could develop in the future, as well as what other functionality I could implement upon revision of my project. This task was straightforward as I already analyses the business problems in the analysis section of my documentation.

User Requirements

Catering

- Customers placing orders over the phone had to pay by card, card receipts had to be stapled to order receipts, if this did not happen then some customers were asked for full payment upon event completion.
- All orders and order details were recorded in an order book, with the volume of work increasing, details were often omitted, this could also lead to customers being asked for payment numerous times or having to provide personal details on repeated occasions.
- Another aspect of administration which raised many issues was the orders not being in chronological order, resulting in staff having to constantly check the order book. On some occasions Doreen had overbooked events and this led to having to complete up to three events in a single day
- Stock ordering was haphazard, and Doreen soon realised that she could be saving much more money if she had ordered certain stock online, however she did not have the facility to do this. She requires a detailed and prioritised list of suppliers in order to keep stock updates consistent.

On-site catering

- It was difficult to manage additional stock and resources, resources had to be recorded on a resource sheet upon arrival at event meaning that mistakes could have easily occurred, or details omitted
- Extra staff were required to staff the parties rather than just prepare catering services, scheduling staff correctly proved to be difficult.

Themed parties

- Doreen relied on a written list of entertainers, which was not practical. Organising this type of party meant that she had to maintain records for a whole range of resources, from entertainers, venues, costume and prop providers and script providers
- Staff must complete their own time sheets which were collected and logged into a computer system once per week, time sheets could easily be completed incorrectly or lost, meaning wage calculations would be wrong.

Weddings

- Doreen needs a set list of suppliers and products which she can easily refer to and readers, currently she researches the needs of each individual customer, but this method would save time
- Having a finite list of certain items such as cars and hotels would be beneficial to the business also.

Hardware and Software Requirements

HARDWARE REQUIRED	
CPU: <ul style="list-style-type: none">• Intel Core i7-2600 with 3.40 GHz	Monitor: <ul style="list-style-type: none">• 17" LCD (minimum)• 19" LCD (recommended)
Memory: <ul style="list-style-type: none">• 8 GB (minimum)• 16GB (recommended)	Graphics Card: <ul style="list-style-type: none">• NVIDIA GeForce 9200
Hard Disk: 100GB (minimum) 500GB (recommended)	

SOFTWARE REQUIRED	
Operating System: <ul style="list-style-type: none">• Windows 7/8/10 (64-bit OS required)	Antivirus: <ul style="list-style-type: none">• Norton Security 2016 (recommended)

Aims of the Project

Functional

- Create a program which will help Doreen manage her business – facilitate customer and employee bookings as well as making orders from suppliers
- Create a facility to update customer records
- Create a facility to update employee records
- Create a facility to update supplier records
- Create a facility to update product records
- Create a facility to add and delete customer records
- Create a facility to add and delete employee records
- Create a facility to add and delete supplier records
- Create a facility to add and delete product records
- Ensure that Doreen will be able to generate reports
- Implement an employee login facility so that bookings can be automatically linked to an employee
- Implement a customer login facility so that customers can login and review their booking details
- Create a help form so that new employees can check how to create new bookings etc.
- Have a menu form which allows the employee to choose from customer, employee, and supplier

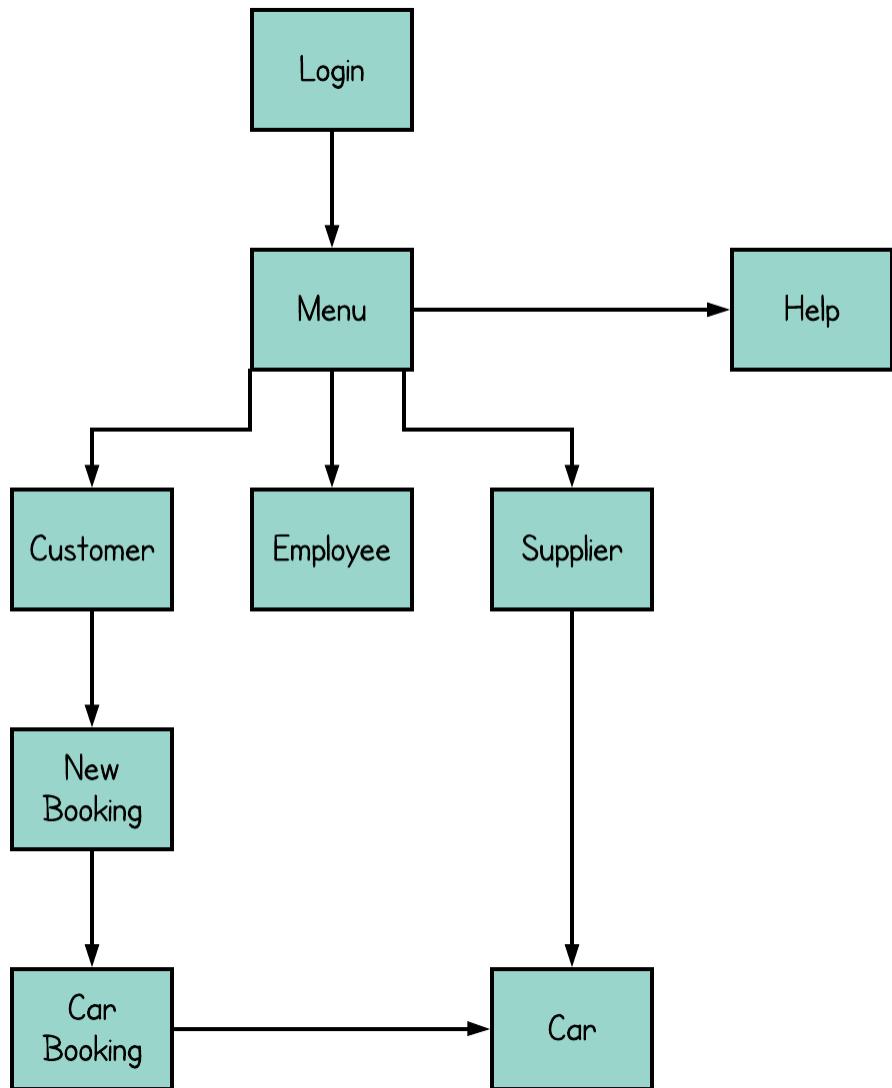
Non-Functional

- Ensure the overall aesthetic of the program is professional and attractive
- Make the program user friendly and accessible
- Implement simple navigation between forms and tables

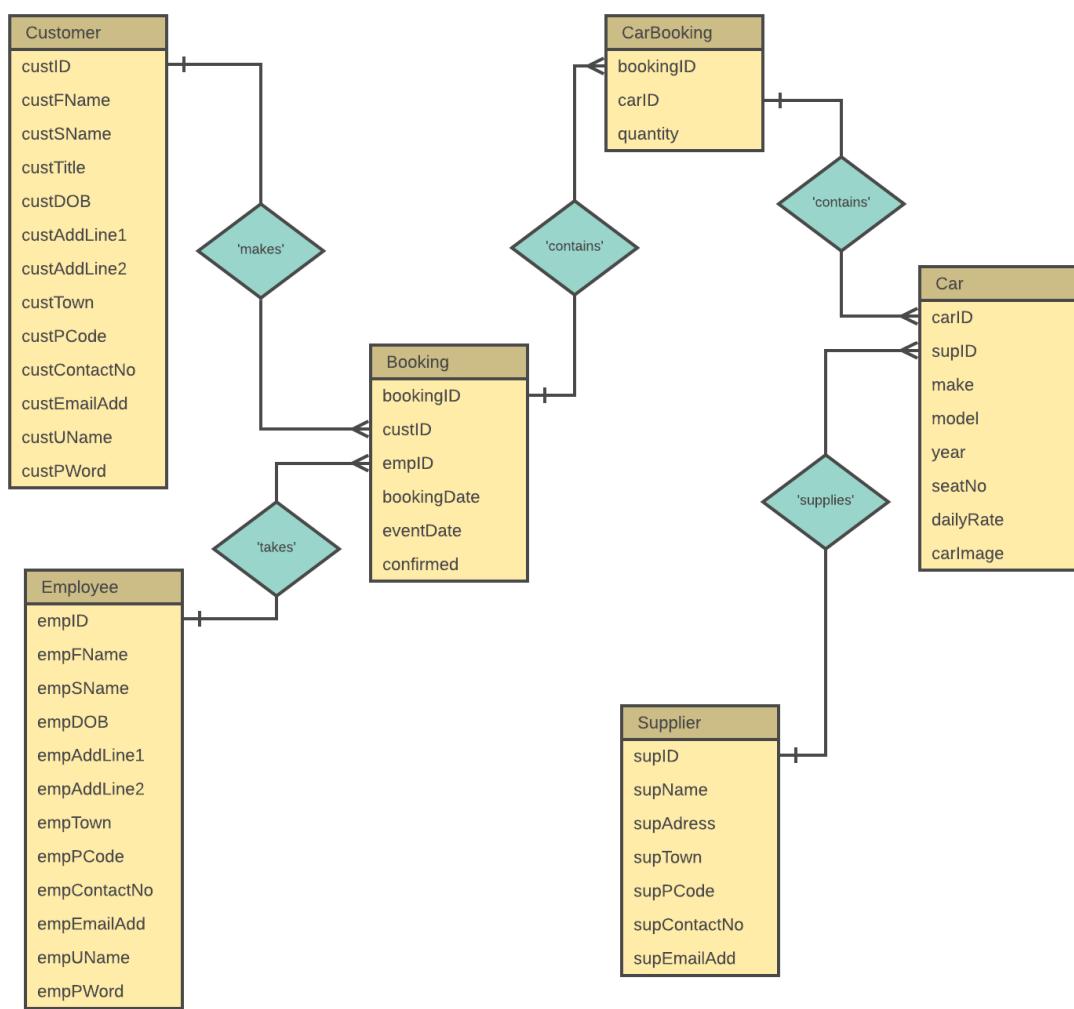
The user requirements I have previously identified are too vast, therefore due to the time constraints of the project I will focus my investigation on the rental of wedding cars and general management in that area. In the future I would ideally be able to design and develop a program which fully encompasses the issues raised with Doreen's business, Events Unlimited, and I would aim to facilitate all other aspects of her work. Unfortunately, this is not possible at this time and I will therefore aim to design and develop this area as best I can.

Design

System Flow Diagram



ER Diagram



DATA STRUCTURES DICTIONARY

CUSTOMER

Field name	Primary key	Data type	Required?	Length	example
CustID	Yes	autonumber	Y	-	42
custFName	No	varchar	Y	30	'John'
custSName	No	varchar	Y	30	'Doe'
custTitle	No	varchar	N	10	'Mr'
custDOB	No	date	Y	8	07/07/1989
custAddLine1	No	varchar	Y	30	'42 Drumman Road'
custAddLine2	No	varchar	N	30	'Apt 3B'
custTown	No	varchar	Y	20	'Armagh'
custPCode	No	varchar	Y	8	'BT60 8EF'
custContactNo	No	varchar	Y	11	'02837988765'
custEmailAdd	No	varchar	N	30	'jd89@yahoo.com'
custUName	No	varchar	N	20	'Jdoe8942'
custPWord	no	varchar	N	20	'Drumman77'

EMPLOYEE

Field name	Primary Key	Data type	Required?	Length	Example
empID	Yes	autonumber	Y	-	13
empFName	No	varchar	Y	30	'Sarah'
empSName	No	varchar	Y	30	'Laney'
empDOB	No	date	Y	8	12/03/1981
empAddLine1	No	varchar	Y	30	'64 Woodford Close'
empAddLine2	No	varchar	N	30	n/a
empTown	No	varchar	Y	20	'Newry'
empPCode	No	varchar	Y	8	'BT59 4TF'
empContactNo	no	varchar	Y	11	'077976234087'
empEmailAdd	No	varchar	Y	30	'slaney@gmail.com'
empUName	No	varchar	Y	20	sarahL81
empPWord	no	varchar	Y	20	Wddfrd641981

SUPPLIER

Field name	Primary key	Data type	Required?	Length	Example
supID	Yes	autonumber	Y	-	5
supName	No	varchar	Y	50	'cars R us'
supAdd	No	varchar	Y	50	'14 Railway Street Business Park, Block 7'
supTown	No	varchar	Y	20	'Armagh'
supPCode	No	varchar	Y	8	'BT61 4LM'
supContactNo	No	varchar	Y	11	'028377498'
supEmailAdd	no	varchar	Y	30	'carsrusltd@googlemail.com'

CAR

Field name	Primary key	Data type	Required?	Length	Example
carID	Yes	autonumber	Y	-	29
supID	No	varchar	Y	-	5
Make	No	varchar	Y	20	'Dodge'
Model	No	varchar	Y	20	'Charger'
Year	No	varchar	Y	4	'1979'
seatNo	No	varchar	Y	-	'3'
dailyRate	No	decimal	Y	-	'£150'
carImage	no	varbinary(MAX)	Y	-	

BOOKING

Field name	Primary key	Data type	Required?	Length	Example
bookingID	Yes	autonumber	Y	-	36
custID	No	varchar	Y	-	'42'
empID	No	varchar	Y	-	'13'
bookingDate	No	date	Y	8	23/12/2017
eventDate	No	date	Y	8	12/08/2018
confirmed	No	bit	Y	1	1

CAR BOOKING

Field name	Primary key	Data type	Required?	Length	example
bookingID	Yes	int	Y	-	36
carID	Yes	int	Y	-	29
quantity	No	int	Y	-	2

Validation

CUSTOMER

Field Name	Validation Used	Example of Validation
CustID	-	-
custFName	Presence check, type check (text only)	-
custSName	Presence check, type check (text only)	-
custTitle	Presence check, type check (text only)	-
custDOB	Presence check, type check (numbers only), format check (date formatting), length check (8), range check	DD/MM/YYYY Between 04/08/1900 AND current date
custAddLine1	Presence check, type check (alphanumeric)	-
custAddLine2	Presence check, type check (alphanumeric)	-
custTown	Presence check, type check (text only)	-
custPCode	Presence check, type check (alphanumeric), length check (8)	LL00 0LL
custContactNo	Presence check, type check (numbers only), length check (11), format check	000 0000 0000
custEmailAdd	-	-
custUName	Presence check, length check (5>)	-
custPWord	Presence check, length check (5>)	-

EMPLOYEE

Field Name	Validation Used	Example of Validation
empID	-	-
empFName	Presence check, type check (text only)	-
empSName	Presence check, type check (text only)	-
custDOB	Presence check, type check (numbers only), format check (date formatting), length check (8), range check	DD/MM/YYYY Between 04/08/1900 AND current date
custAddLine1	Presence check, type check (alphanumeric)	-
custAddLine2	Presence check, type check (alphanumeric)	-
custTown	Presence check, type check (text only)	-

custPCode	Presence check, type check (alphanumeric), length check (8)	LL00 OLL
custContactNo	Presence check, type check (numbers only), length check (11), format check	000 0000 0000
custEmailAdd	-	-
custUName	Presence check, length check (5>)	-
custPWord	Presence check, length check (5>)	-

SUPPLIER

Field Name	Validation Used	Example of Validation
supID	-	-
supName	Presence check, type check (text only)	-
supAdd	Presence check, type check (alphanumeric)	-
supTown	Presence check, type check (text only)	-
supPCode	Presence check, type check (alphanumeric), length check (8)	LL00 OLL
supContactNo	Presence check, type check (numbers only), length check (11), format check	000 0000 0000
supEmailAdd	-	-

CAR

Field Name	Validation Used	Example of Validation
carID	-	-
supID	-	-
Make	Presence check, type check (text only)	-
Model	Presence check, type check (alphanumeric)	-
Year	Presence check, type check (numbers only), range check	Between 1920 AND current year (2018)
seatNo	Presence check, type check (numbers only), range check	Between 1 AND 30
dailyRate	Presence check, type check (decimals only), range check	Between 100.00 AND 1000.00
carImage	Presence check	-

BOOKING

Field Name	Validation Used	Example of Validation
bookingID	-	-
custID	-	-
empID	-	-
bookingDate	Presence check, type check (numbers only), format check (date formatting), length check (8), range check	DD/MM/YYYY Between 04/08/1900 AND current date
eventDate	Presence check, type check (numbers only), format check (date formatting), length check (8), range check	DD/MM/YYYY Between 04/08/1900 AND current date + 5 years
confirmed	Presence check, type check(bit / 0,1), length check (1)	-

CAR BOOKING

Field Name	Validation Used	Example of Validation
bookingID	-	-
carID	-	-
quantity	Presence check, type check (numbers only), range check	Between 1 AND 5 (drop down menu)

Normalisation

0NF

Booking-0(custID, custFName, custSName, custTitle, custDOB, custAddLione1, custAddLine2, custTown, custPCode, custContactNo, custUName, custPWord, empID, empFName, empSName, empDOB, empAddLine1, empAddLine2, empTown, empPCode, empContactNo, empUName, empPWord, bookingID, custID, empID, bookingDate, eventDate, confirmed, bookingID, carID, quantity, carID, make, model, year, seatNo, dailyRate, carImage, supID, supName, supAdress, supTown, supPCode, supContactNo, supEmailAdd).

1NF

Customer-1(custID, custFName, custSName, custTitle, custDOB, custAddLione1, custAddLine2, custTown, custPCode, custContactNo, custUName, custPWord).

Employee-1(empID, empFName, empSName, empDOB, empAddLine1, empAddLine2, empTown, empPCode, empContactNo, empUName, empPWord).

Booking-1(bookingID, custID, empID, bookingDate, eventDate, confirmed, quantity, carID, make, model, year, seatNo, dailyRate, carImage, supID, supName, supAdress, supTown, supPCode, supContactNo, supEmailAdd).

2NF

Customer-1(custID, custFName, custSName, custTitle, custDOB, custAddLione1, custAddLine2, custTown, custPCode, custContactNo, custUName, custPWord).

Employee-1(empID, empFName, empSName, empDOB, empAddLine1, empAddLine2, empTown, empPCode, empContactNo, empUName, empPWord).

Booking-2(bookingID, custID, empID, bookingDate, eventDate, confirmed, carID, quantity).

Car-2(carID, make, model, year, seatNo, dailyRate, carImage, supID, supName, supAdress, supTown, supPCode, supContactNo, supEmailAdd).

3NF (see pg 20 (ER diagram))

Customer-1(custID, custFName, custSName, custTitle, custDOB, custAddLione1, custAddLine2, custTown, custPCode, custContactNo, custUName, custPWord).

Employee-1(empID, empFName, empSName, empDOB, empAddLine1, empAddLine2, empTown, empPCode, empContactNo, empUName, empPWord).

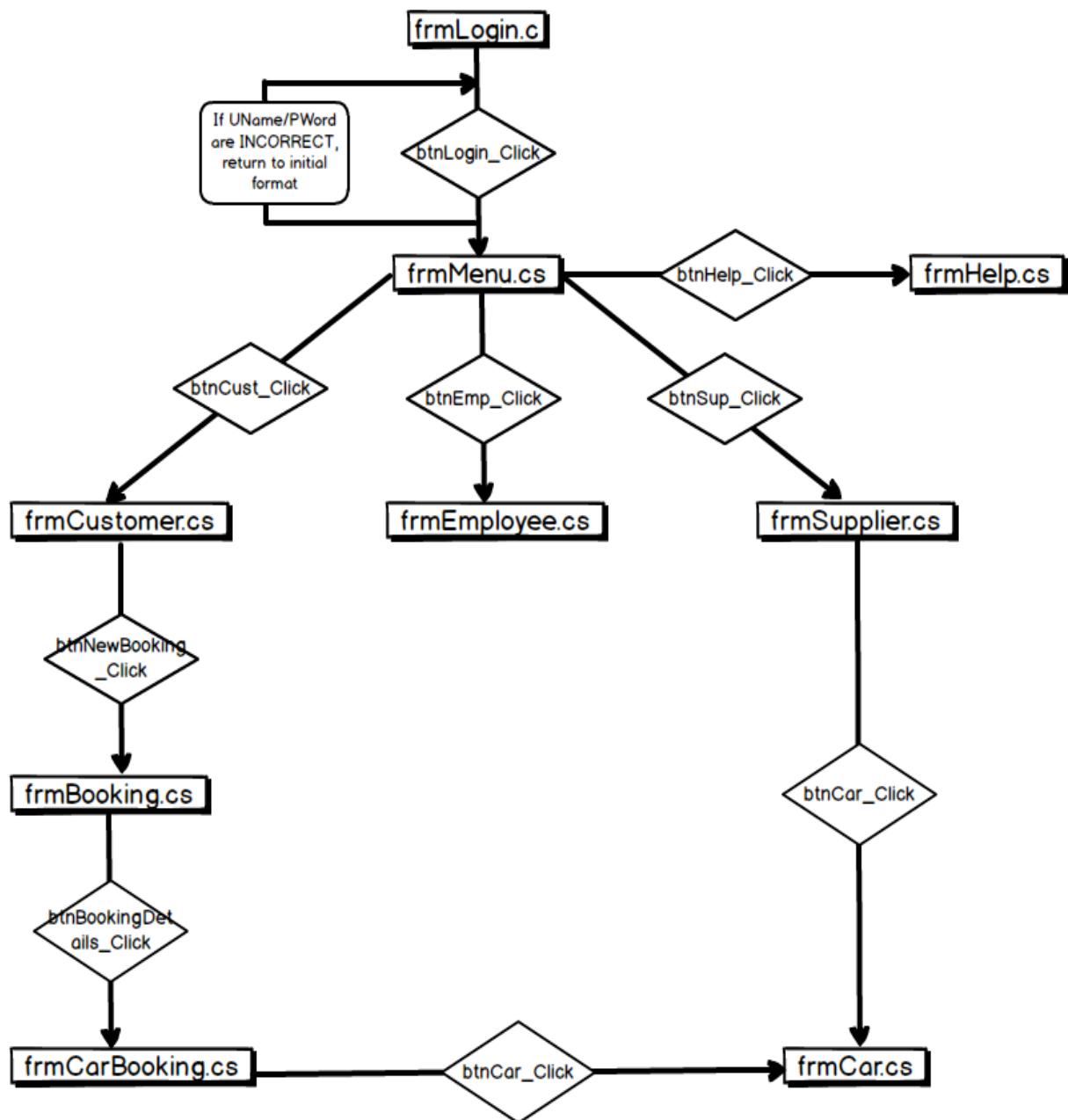
Booking-3(bookingID, custID, empID, bookingDate, eventDate, confirmed).

CarBooking-3(bookingID, carID, quantity).

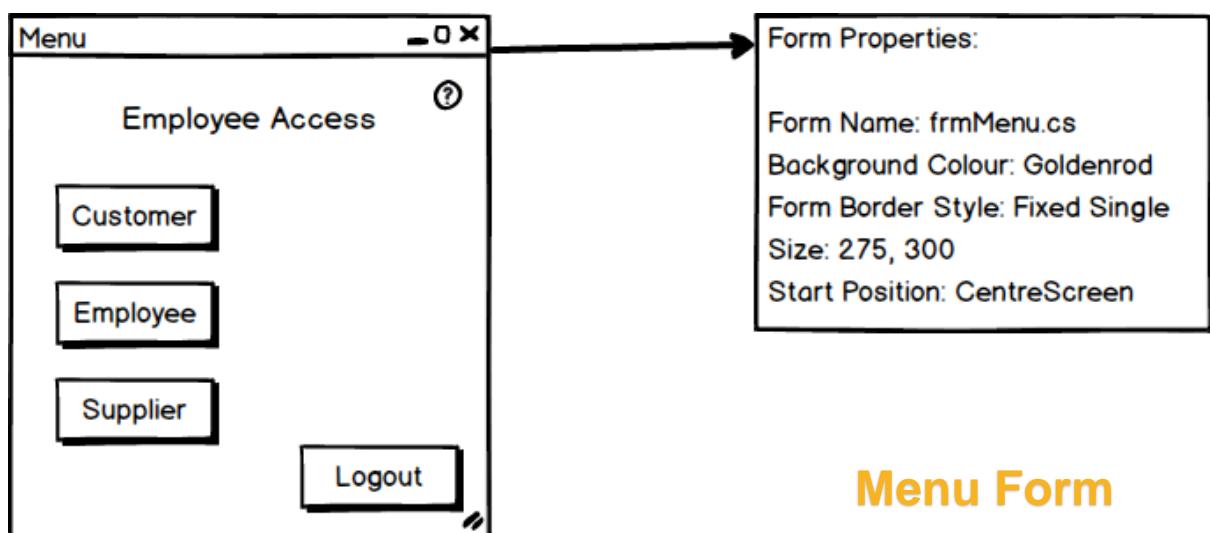
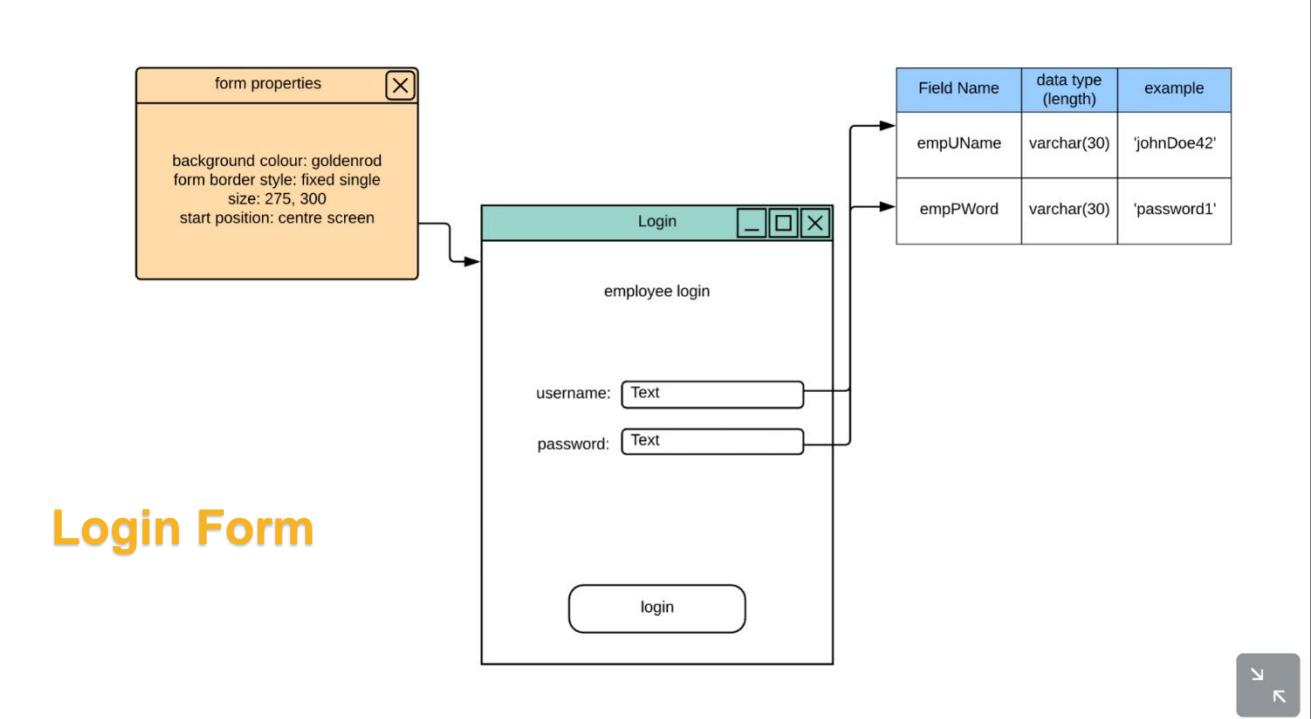
Car-3(carID, supID, make, model, year, seatNo, dailyRate, carImage).

Supplier-3(supID, supName, supAdress, supTown, supPCode, supContactNo, supEmailAdd).

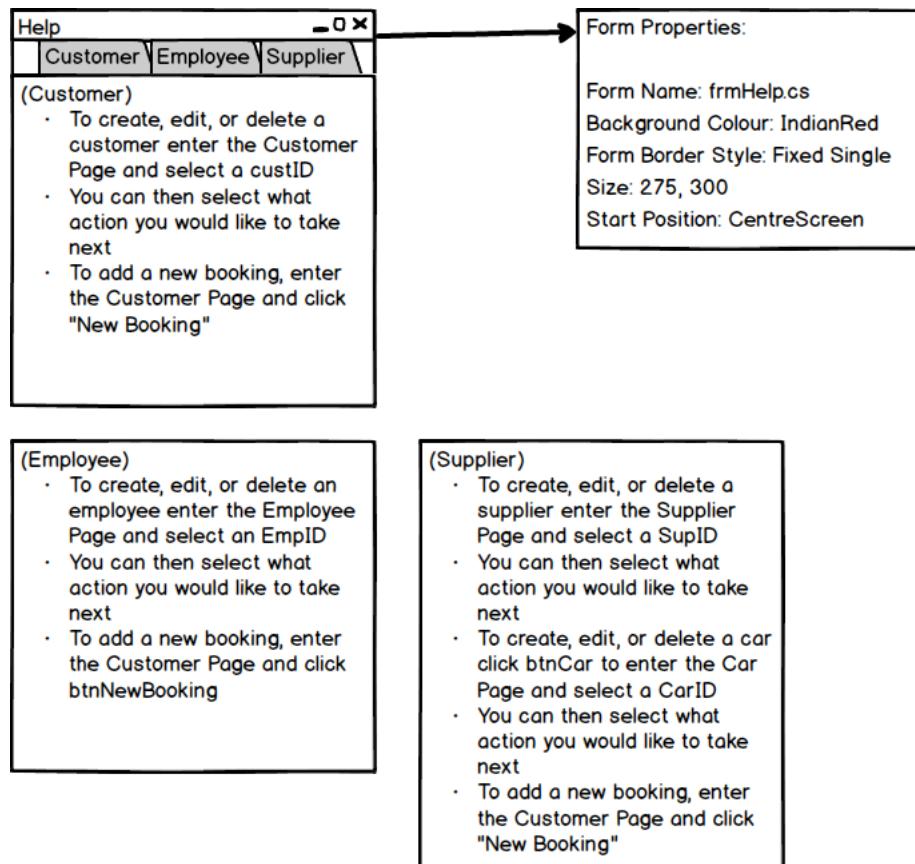
Program Overview



Form Sketches



Help Form



Customer Form

Form Properties:

Form Name: frmCustomer.cs
Background Colour: DeepSeaGreen
Form Border Style: Fixed Single
Size: 785, 500
Start Position: CentreScreen

Customer

ID FName SName Title DoB Adress ContactNo Email UName PWord

Save AddNew Clear Delete NewBooking Menu

(Customer Table)

Field Name	Data Type (length)	Example
CustID	Autonumber(-)	42
CustFName	Varchar(30)	"Jane"
CustSName	Varchar(30)	"Doe"
CustTitle	Varchar(5)	"Miss"
CustDOB	Date(3)	07-03-1994
CustAddLine1	Varchar(30)	"12 Penny Lane"
CustAddLine2	Varchar(30)	"Apt 5"
CustTown	Varchar(30)	"Armagh"
CustPCode	Varchar(8)	"BT61 8EQ"
CustContactNo	Varchar(11)	"02837551083"
CustEmailAdd	Varchar(30)	"jane_doe@yahoo.com"
CustUName	Varchar(30)	"jane94"
CustPWord	Varchar(30)	"Penny5?"

Employee Form

Form Properties:

Form Name: frmEmployee.cs
Background Colour: DeepSeaGreen
Form Border Style: Fixed Single
Size: 785, 500
Start Position: CentreScreen

The Employee form interface includes fields for ID, FName, SName, DoB, Address, ContactNo, Email, UName, and PWord, along with Save, AddNew, Clear, Delete, and Menu buttons. Below the form is a table titled '(EmployeeTable)' with columns for ID, FName, SName, DoB, Adress, ContactNo, Email, UName, and PWord.

Field Name	Data Type (length)	Example
EmpID	Autonumber(-)	17
EmpFName	Varchar(30)	“Frank”
EmpSName	Varchar(30)	“Castle”
EmpDOB	Date(3)	12-08-1984
EmpAddLine1	Varchar(30)	“4 Drummad Walk”
EmpAddLine2	Varchar(30)	N/A
EmpTown	Varchar(30)	“Banbridge”
EmpPCode	Varchar(8)	“BT60 4LJ”
EmpContactNo	Varchar(11)	“02832998551”
EmpEmailAdd	Varchar(30)	“castle_frank84@gmail.com”
EmpUName	Varchar(30)	“frankieC84”
EmpPWord	Varchar(30)	“drummad84”

Supplier Form

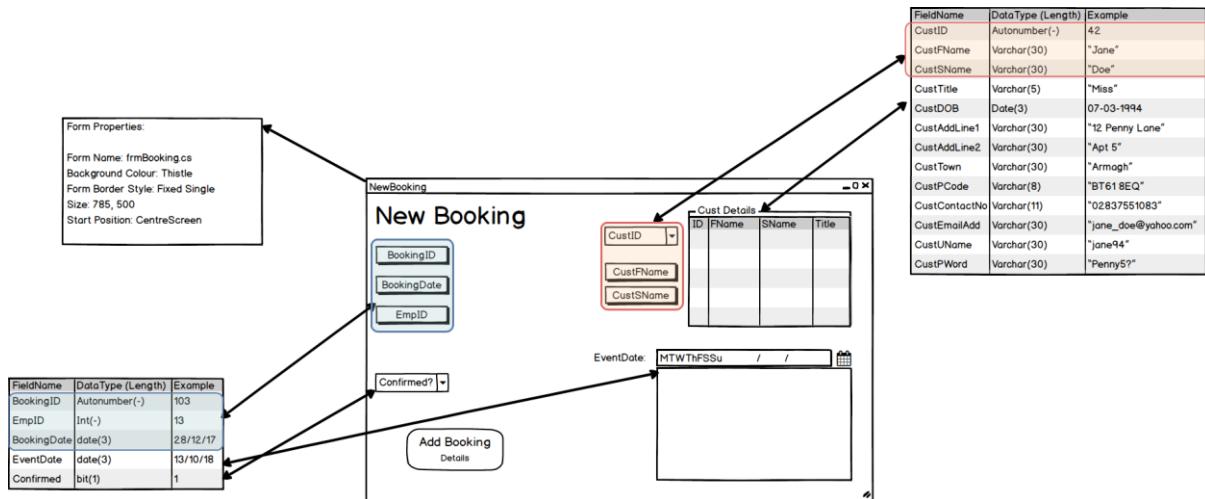
Form Properties:

- Form Name: frmSupplier.cs
- Background Colour: DeepSeaGreen
- Form Border Style: Fixed Single
- Size: 785, 500
- Start Position: CentreScreen

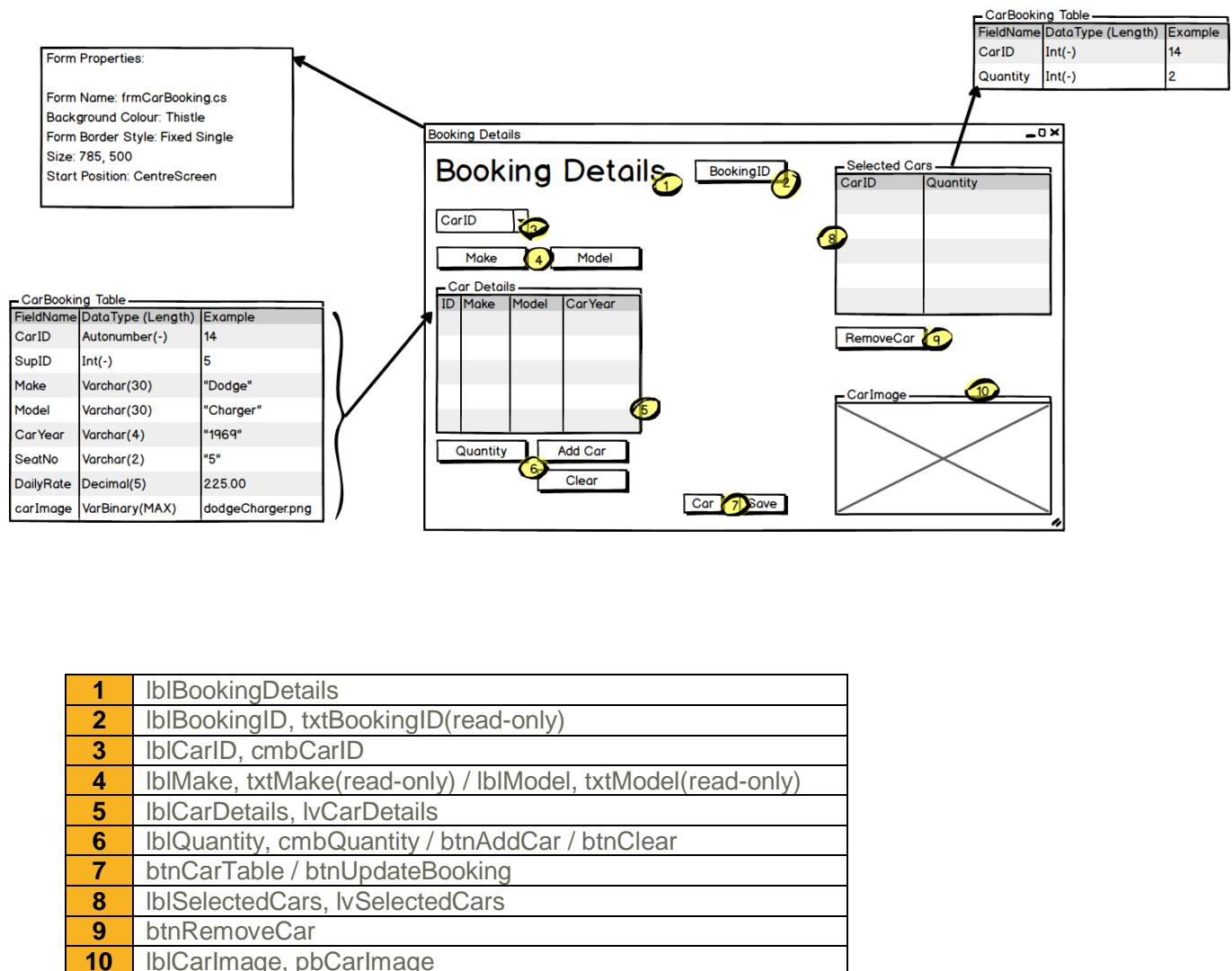
The screenshot shows a Windows application window titled "Supplier". The form contains fields for ID, Name, Address, ContactNo, and Email, along with buttons for Save, AddNew, Clear, Delete, and Menu. Below the form is a table titled "(Supplier Table)" with columns for ID, Name, Address, ContactNo, and Email.

Field Name	Data Type (length)	Example
SupID	Autonumber(-)	1
SupName	Varchar(30)	“Tennyson Luxury Cars”
SupAdd	Varchar(30)	“38 Moy Road”
SupTown	Varchar(30)	“Armagh”
SupPCode	Varchar(8)	“BT62 6KE”
SupContactNo	Varchar(11)	“07935096035”
SupEmailAdd	Varchar(30)	“tennysonlc_business@googlemail.com”

Booking Form



Car Booking Form



Car Form

Form Properties:

Form Name: frmCars
Background Colour: DeepSeaGreen
Form Border Style: Fixed Single
Size: 785, 500
Start Position: CentreScreen

Car

ID SupID Make Model CarImage
CarYear SeatNo DailyRate Menu
Save AddNew Clear Delete

(CarTable)

ID	SupID	Make	Model	CarYear	SeatNo	DailyRate	CarImage

Field Name **Data Type (length)** **Example**

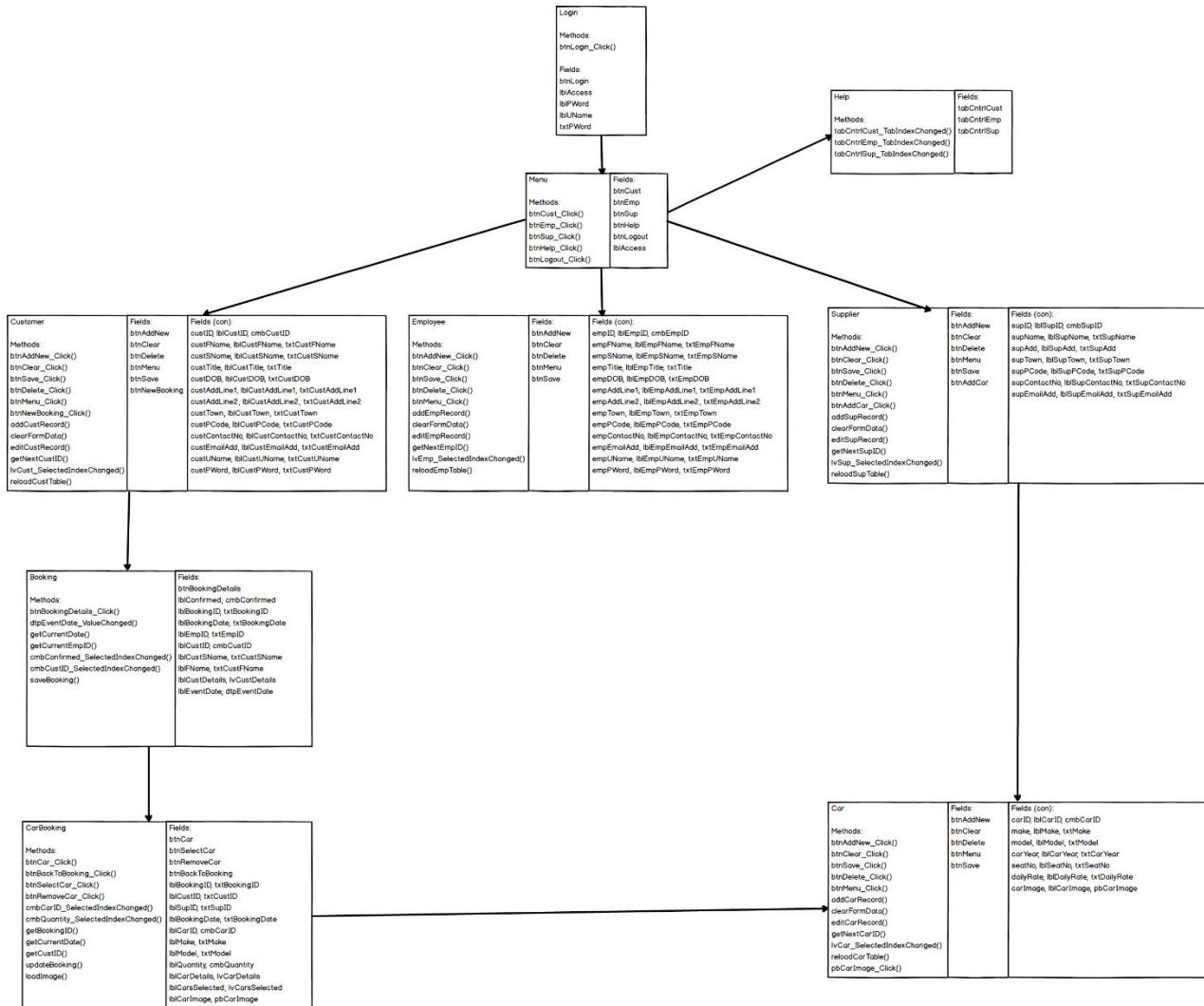
Field Name	Data Type (length)	Example
CarID	Autonumber(-)	14
SupID	Int(-)	5
Make	Varchar(30)	"Dodge"
Model	Varchar(30)	"Charger"
CarYear	Varchar(4)	"1969"
SeatNo	Varchar(2)	"5"
DailyRate	Decimal5)	225.00
CarImage	VarBinary(MAX)	dodgeCharger.png

System Components

Component	Details
Events_Unlimited.mdf	This is the database file wherein all information is stored on bookings, customers, cars, employees and suppliers.
Program.cs	This class will create and run the new instance of the first form (frmLogin.cs)
dBConnection.cs	This is the class where the database connection string is stored. I created this class so that if the file path is changed, only one instance of it in the program will need to be altered.
frmBooking.cs	This is the form in which a user (employee) may create a booking for a specific customer. This form links to both the customer and booking table in the database.
frmCar.cs	This is the form which all details of car records are displayed. On this form a user may alter records, add new records, delete records and query the car table of the database.
frmCarBooking.cs	This is the form in which a user may enter specific details of a booking for a customer, such as carID, quantity, and eventDate. This form links to both the car table and the carBooking tables of the database.
frmCustomer.cs	This is the form which all details of customer records are displayed. On this form a user may alter records, add new records, delete records and query the customer table of the database.
frmEmployee.cs	This is the form which all details of employee records are displayed. On this form a user may alter records, add new records, delete records and query the employee table of the database.
frmHelp.cs	This form will contain very basic instructions on how to use the main forms and how to create a booking. Further instructions for the system can be found in the User Guide.

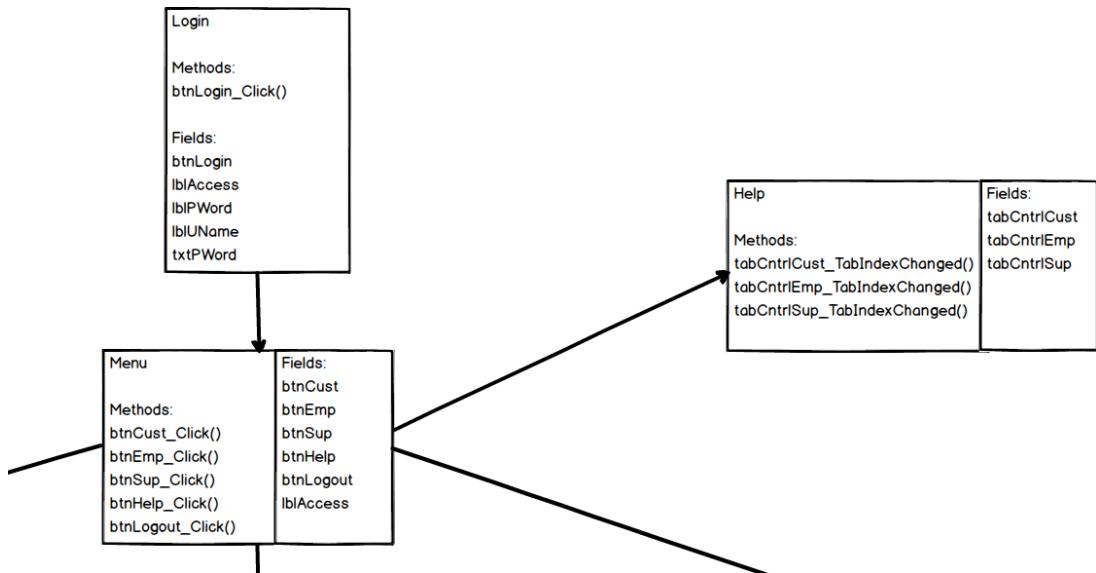
frmLogin.cs	This is the first form which loads, the user will use their employee credentials to login to the system. This form links to the employee table of the database.
frmMenu.cs	This form will contain buttons to the main forms from which the user can choose the appropriate path for their task. Buttons on this form will include Customer, Employee, Supplier, Help and Logout.
frmSupplier.cs	This is the form which all details of supplier records are displayed. On this form a user may alter records, add new records, delete records and query the supplier table of the database.
reportDS.xsd	This file contains the report of a booking which can be generated once a booking is complete.

Class Diagram



Class diagrams enlarged:

1. Login, Menu, Help



2. Customer

Customer Methods: <code>btnAddNew_Click()</code> <code>btnClear_Click()</code> <code>btnSave_Click()</code> <code>btnDelete_Click()</code> <code>btnMenu_Click()</code> <code>btnNewBooking_Click()</code> <code>addCustRecord()</code> <code>clearFormData()</code> <code>editCustRecord()</code> <code>getNextCustID()</code> <code>lvCust_SelectedIndexChanged()</code> <code>reloadCustTable()</code>	Fields: <code>btnAddNew</code> <code>btnClear</code> <code>btnDelete</code> <code>btnMenu</code> <code>btnSave</code> <code>btnNewBooking</code>	Fields (con): <code>custID, lblCustID, cmbCustID</code> <code>custFName, lblCustFName, txtCustFName</code> <code>custSName, lblCustSName, txtCustSName</code> <code>custTitle, lblCustTitle, txtTitle</code> <code>custDOB, lblCustDOB, txtCustDOB</code> <code>custAddLine1, lblCustAddLine1, txtCustAddLine1</code> <code>custAddLine2, lblCustAddLine2, txtCustAddLine2</code> <code>custTown, lblCustTown, txtCustTown</code> <code>custPCode, lblCustPCode, txtCustPCode</code> <code>custContactNo, lblCustContactNo, txtCustContactNo</code> <code>custEmailAdd, lblCustEmailAdd, txtCustEmailAdd</code> <code>custUName, lblCustUName, txtCustUName</code> <code>custPWord, lblCustPWord, txtCustPWord</code>
---	--	--

3. Booking, Car Booking



4. Employee, Supplier

Employee Methods: btnAddNew_Click() btnClear_Click() btnSave_Click() btnDelete_Click() btnMenu_Click() addEmpRecord() clearFormData() editEmpRecord() getNextEmpID() lvEmp_SelectedIndexChanged() reloadEmpTable()	Fields: btnAddNew btnClear btnDelete btnMenu btnSave	Fields (con): empID, lblEmpID, cmbEmpID empFName, lblEmpFName, txtEmpFName empSName, lblEmpSName, txtEmpSName empTitle, lblEmpTitle, txtTitle empDOB, lblEmpDOB, txtEmpDOB empAddLine1, lblEmpAddLine1, txtEmpAddLine1 empAddLine2, lblEmpAddLine2, txtEmpAddLine2 empTown, lblEmpTown, txtEmpTown empPCode, lblEmpPCode, txtEmpPCode empContactNo, lblEmpContactNo, txtEmpContactNo empEmailAdd, lblEmpEmailAdd, txtEmpEmailAdd empUName, lblEmpUName, txtEmpUName empPWord, lblEmpPWord, txtEmpPWord
---	---	---

Supplier Methods: btnAddNew_Click() btnClear_Click() btnSave_Click() btnDelete_Click() btnMenu_Click() btnAddCar_Click() addSupRecord() clearFormData() editSupRecord() getNextSupID() lvSup_SelectedIndexChanged() reloadSupTable()	Fields: btnAddNew btnClear btnDelete btnMenu btnSave btnAddCar	Fields (con): supID, lblSupID, cmbSupID supName, lblSupName, txtSupName supAdd, lblSupAdd, txtSupAdd supTown, lblSupTown, txtSupTown supPCode, lblSupPCode, txtSupPCode supContactNo, lblSupContactNo, txtSupContactNo supEmailAdd, lblSupEmailAdd, txtSupEmailAdd
--	--	---

5. Car

Car Methods: btnAddNew_Click() btnClear_Click() btnSave_Click() btnDelete_Click() btnMenu_Click() addCarRecord() clearFormData() editCarRecord() getNextCarID() lvCar_SelectedIndexChanged() reloadCarTable() pbCarImage_Click()	Fields: btnAddNew btnClear btnDelete btnMenu btnSave	Fields (con): carID, lblCarID, cmbCarID make, lblMake, txtMake model, lblModel, txtModel carYear, lblCarYear, txtCarYear seatNo, lblSeatNo, txtSeatNo dailyRate, lblDailyRate, txtDailyRate carImage, lblCarImage, pbCarImage
---	---	--

Pseudo Code

frmLogin.cs

- User will type in their username and password
- User will click submit button
- Program will query database to check username is valid
- If username is valid, program will query if the password entered is the correct password associated with the username
- Form will close and frmMenu.cs will open

frmMenu.cs

- User will click on one of the five buttons on the form;
 - **Logout button** – form will close and the frmLogin.cs will be reloaded; user must login again to access the menu
 - **Help button** – form will remain open; help frmHelp.cs will load
 - **Customer button** – form will remain open; frmCustomer.cs will load
 - **Employee button** – form will remain open; frmEmployee.cs will load
 - **Supplier button** – form will remain open; frmSupplier.cs will load

frmHelp.cs

- User can choose from five tabs (labelled: Customer, Employee, Supplier, Car, and Booking)
- On tab click, the screen will change to the relevant section and help text will be displayed
- User can then exit frmHelp.cs via the form's 'X' button

frmCustomer.cs

- On load listView(customer) will display all information on customers from the Customer database in relevant columns
- the Customer ID field will also auto-increment in the customerID textbox which will be a read-only field, so the user can create a new customer record easily
- On SelectedIndexChanged, the selected CustomerID will load the remaining customer details into the appropriate textboxes, this facility will allow the user to edit and update existing records.
- Form will have six buttons
 - **Save** – on click any alterations of new customers details will be saved to the database, the listView will then reload, allowing for the changes to be seen
 - **Add New** – if there is already a customer selected, textboxes will be cleared and the CustomerID correctly incremented
 - **Clear** - any typing in the textboxes will be cleared, saved or unsaved
 - **Delete** – this facility will allow for the deletion of a record. A dialogue window will pop-up, confirming your decision to delete the customer record. When confirmed a new form will open, displaying the currently deleted records.
 - **New Booking** - this button will open the form for a new booking
 - **Menu** - on click frmCustomer will close, with frmMenu remaining open in the background.

frmEmployee.cs

- On load, listview(employee) will display all information on employees from the Employee database in tabular format
- The EmployeeID field will also auto-increment in the EmployeeID textbox which will be a read-only field, so the user can create a new employee record easily
- On SelectedIndexChanged, the selected EmployeeID will load the remaining customer details into the appropriate textboxes, this facility will allow the user to edit and update existing records.
- Form will have six buttons
 - **Save** – on click any alterations of new employee's details will be saved to the database, the listview will then reload, allowing for the changes to be seen
 - **Add New** – if there is already an employee selected, textboxes will be cleared and the EmployeeID correctly incremented
 - **Clear** - any typing in the textboxes will be cleared, saved or unsaved
 - **Delete** – this facility will allow for the deletion of a record. A dialogue window will pop-up, confirming your decision to delete the employee record. When confirmed a new form will open, displaying the currently deleted records.
 - **New Booking** - this button will open the form for a new booking
 - **Menu** - on click frmEmployee will close, with frmMenu remaining open in the background.

frmSupplier.cs

- On load, listview(Supplier) will display all information on suppliers from the Supplier database in tabular format
- The SupplierID field will also auto-increment in the SupplierID textbox which will be a read-only field, so the user can create a new supplier record easily
- On SelectedIndexChanged, the selected SupplierID will load the remaining supplier details into the appropriate textboxes, this facility will allow the user to edit and update existing records.
- Form will have six buttons
 - **Save** – on click any alterations of new supplier's details will be saved to the database, the listview will then reload, allowing for the changes to be seen
 - **Add New** – if there is already a supplier selected, textboxes will be cleared and the SupplierID correctly incremented
 - **Clear** - any typing in the textboxes will be cleared, saved or unsaved
 - **Delete** – this facility will allow for the deletion of a record. A dialogue window will pop-up, confirming your decision to delete the supplier record. When confirmed a new form will open, displaying the currently deleted records.
 - **New Product** - this button will open the form for adding a new product (Car)
 - **Menu** - on click frmSupplier will close, with frmMenu remaining open in the background.

frmNewBooking.cs

- on load, BookingID value will be auto-incremented and displayed in a read-only textbox this value will increase with every booking created, not every booking confirmed (so as to avoid data duplication); this value will be used to reference which booking it is (in case of numerous booking in the same name or on the same date etc), it will be saved into a relevant field in the BookingTable and visible on reports.

- on load, the current date will be displayed in a label on the form, date found using dateTime selector (e.g. Mon 01-01-18); this value will be used to reference which date a particular booking was made on, it will be saved into a relevant field in the BookingTable and visible on reports.
- on load, current EmployeeID will be loaded into a read-only textbox (this will be derived from the employee's initial login); this value will be used to reference which employee took a particular booking, it will be saved into a relevant field in the BookingTable and visible on reports.
- Customer DropDown Menu
 - On click, a dropdown menu of all customerIDs will display, when a value is selected, the textbox will display the value and the CustomerListView will load the customer's details – this will help to ensure that the correct customer is selected for the booking.
 - The selected CustomerID will be used to reference which Customer the booking is for, it will be saved to the BookingTable and visible on reports.
- CustomerListView
 - Listview will display all details related to the selected CustomerID from the CustomerTable
- EventDate dateTime selector
 - dateTime selector will allow user to choose a date between the current date this date + 18 months
 - when date is clicked it will display into a textbox; the selected date will be used to reference the date the event will take place, it will be saved to the BookingTable and visible on reports.
 - The date will also determine whether or not a certain Car is available, each car will have a maximum booking quantity of 5, this can mean that a car can be either booked singly on 5 separate occasions or booking once at a quantity of 5
- Confirmed Button (CheckBox)
 - on click, buttonIMG will display green tick, on repeat click, buttonIMG will display red cross – this will indicate whether or not the booking has been saved / confirmed
- Booking Details button
 - On click, frmCarBooking.cs will open, frmBooking.cs will remain open in background

frmCarBooking.cs

- On load, BookingID value will be displayed in a read-only textbox this value will be taken from the value assigned to frmBooking.cs, not frmCarBooking.cs; this value will be used to reference which booking it is, linking it to customer details which are declared on the booking form, it will be saved into a relevant field in the CarBookingTable and visible on reports.
- On load, the date from the previous booking form will be visible on the form but not saved to the CarBookingTable.
- CarID DropDownMenu
 - On click, a dropdown menu of all CarIDs will display, when a value is selected, the textbox will display the value and the CarListView will load the car's details – this will help to ensure that the correct car is selected for the booking.
 - On selectedIndexChanged, pictureBoxCar will load the image from the CarTable where the CarID matches that of the CarID selected from the DropDownMenu
 - The selected CarID will be used to reference which Customer the booking is for, it will be saved to the CarBookingTable and visible on reports.

- AddCar button
 - On click, CarID and quantity will be displayed in LvAddedCars
 - The carID and quantity will be used to determine if the car is available and to reference the customer's full order details. It will be saved to the CarBookingTable and it will also be visible on reports.
- **BookingReturnButton**
 - On click, changed values on frmCarBooking.cs will be updated to CarBookingTable
 - frmCarBooking.cs will close

frmCar.cs

- On load, listview(Car) will display all information on all products from the Car database in a tabular format
- The CarID field will also auto-increment in the CarID textbox which will be a read-only field, so the user can create a new car record easily
- On SelectedIndexChanged, the selected SupplierID will load the remaining supplier details into the appropriate textboxes, this facility will allow the user to edit and update existing records.
- Form will have six buttons
 - **Save** – on click any alterations of new supplier's details will be saved to the database, the listview will then reload, allowing for the changes to be seen
 - **Add New** – if there is already a car selected, textboxes will be cleared and the SupplierID correctly incremented
 - **Clear** - any typing in the textboxes will be cleared, saved or unsaved
 - **Delete** – this facility will allow for the deletion of a record. A dialogue window will pop-up, confirming your decision to delete the customer record. Confirmed a new form will open, displaying the currently deleted records.
 - **New Product** - this button will open the form for adding a new product (Car)
 - **Menu** - on click frmCar will close, with frmMenu remaining open in the background.

Report Design

Booking Report

Events Unlimited - Booking Report

Booking No:	[REDACTED]	Date:	[REDACTED]
Booking Details...	[REDACTED]	Cust Details...	[REDACTED]

CarID	Make	Model	SupID	SupName	DailyRate	Quantity	Total
Product Booking Details...							
Final Totals...							

Testing

Types of Testing

When testing my project, I will implement numerous types of testing, including the following: Alpha testing; black box testing; white box testing; system testing; and acceptance testing.

Alpha testing will provide a good idea of the reliability of the software at an early stage and will help to simulate real user behaviour and environments. Alpha testing will help to identify errors at an early stage which will result in them being addressed before any serious issues can emerge. A downside of alpha testing is that in-depth functionality can not be tested as the system is still very much in the development stage.

Black box testing is a testing technique where no knowledge of the internal functionality and structure of the system is available. As I have created the system and have knowledge of the internal workings, I aim to consult my peers and ask them to test my system and its external functionality; this will help me to understand how my system would work implemented in a business.

White box testing involves looking inside the software and recreating instances of exceptions and breaks. White box testing requires knowledge of the system and programming skills and provides an internal perspective of the software. I think this will be essential to my project as it will allow me to test real conditions.

At the end of my project I will perform thorough system testing, this is a level of testing where a complete and integrated piece of software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements, I will therefore assess my project in accordance with the user requirements which I outlined in the analysis section of my documentation. System testing requires my system to be fully finished as it must emulate an exact workplace, while also testing internal functionality. I feel that this is ultimately one of the most important types of testing which must be performed on a system and I will therefore try to complete it thoroughly.

I will also perform acceptance testing which is a level of software testing where a system is tested for acceptability, this is a key part of software development as it assesses whether or not the system truly suited to the business. If the acceptance testing results are poor while other tests are prosperous, the system is still not beneficial to staff or businesses. For this section of my thorough testing, I will request that my peers complete various questionnaires based on their use, perception and review of my system.

Test Plan

Test no.	Description	Test Data	Expected Outcome	Requirement
1 (Login form)	-	-	-	
1.1	frmLogin.cs loads		Login form will load first	
1.2.1	Text entered into txtUName CORRECT	“frankieC84”	Upon btnLogin_click this text will be accepted frmMenu will load frmLogin will close	
1.2.2	Text entered into txtUName INCORRECT	“fr@nkieC_84”	Upon btnLogin_click this text will not be accepted and a message box will load stating “Please enter numbers and letters only”	
1.2.3	Text entered into txtUName EXTREME	“frankieC84...” (max 30 char. reached)	Upon btnLogin_click this text will not be accepted and a message box will appear stating “Maximum characters: 30”	
1.3.1	Text entered into txtPWord CORRECT	“drummad84”	Upon btnLogin_click this text will be accepted frmMenu will load frmLogin will close	
1.3.2	Text entered into txtPWord INCORRECT	“drummad_84!”	Upon btnLogin_click this text will not be accepted	

			and a message box will load stating “Please enter numbers and letters only”	
1.3.3	Text entered into txtPWord EXTREME	“drummad84...” (max 30 char. reached)	Upon btnLogin_click this text will not be accepted and a message box will appear stating “Maximum characters: 30”	
1.4	btnLogin is clicked	-	Text entered into txtUName and txtPWord will be validated (alphanumeric exception) Employee table of Evets-Unlimited database will be queried in order to verify that both UName and PWord are both valid	
2 (Menu Form)	-	-	-	
2.1	frmMenu.cs loads	-	Menu form will load	
2.2	btnLogout is clicked	-	Message box will show stating “Are you sure you wish to exit?” (message box buttons: ‘Exit’, ‘Cancel’) If yes: frmMenu.cs will close and frmLogin.cs will open	

			If no: message box will close	
2.3	btnHelp is clicked	-	frmMenu.cs will close, frmHelp.cs will load	
2.4	btnCustomer is clicked	-	frmMenu.cs will close, frmCustomer.cs will load	
2.5	btnEmployee is clicked	-	frmMenu.cs will close, frmEmployee.cs will load	
2.6	btnSupplier is clicked	-	frmMenu.cs will close, frmSupplier will load	
3 (Customer Form)	<i>As my Customer, Employee and Supplier forms are duplicates of one another with only minor differences, I have decided to record the Test Plan for my Customer form only – along with any differences that need further testing.</i>			
3.1	frmCustomer.cs will load	-	Customer form will load Customer details will be loaded into lvCustomer from database custoemrID will increment in txtCustID	
3.2	Selected index is changed in Customer listview	-	Selected customer's details will load into associated textboxes (e.g. custFName will be inserted into txtCustFName)	
3.3.1	Text entered into txtCustFName CORRECT	"jane"	Upon btnAddNew _click this text will be accepted frmMenu will load frmLogin will close	

3.3.2	Text entered into txtCustFName INCORRECT	“jan3”	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters only. Error found in txtCustFName”	
3.3.3	Text entered into txtCustFName EXTREME	“jane...” (max 30 char. reached)	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtCustFName”	
3.4.1	Text entered into txtCustSName CORRECT	“Doe”	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database	
3.4.2	Text entered into txtCustSName INCORRECT	“D0e”	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters only. Error found in txtCustSName”	
3.4.3	Text entered into txtCustSName EXTREME	“Doe...” (max 30 char. reached)	Upon btnAddNew_click this text will not be accepted and a message box	

			will appear stating “Maximum characters: 30. Error found in txtCustSName”	
3.5.1	Text entered into txtCustTitle <i>CORRECT</i>	“Miss”	Upon btnAddNew _click this text will be accepted And lvCustomer will reload with updated data from database	
3.5.2	Text entered into txtCustTitle <i>INCORRECT</i>	“M15s”	Upon btnAddNew _click this text will not be accepted and a message box will load stating “Please enter letters only. Error found in txtCustTitle”	
3.5.3	Text entered into txtCustTitle <i>EXTREME</i>	“Miss...” (max 30 char. reached)	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 5. Error found in txtCustTitle”	
3.6.1	Text entered into txtCustDOB <i>CORRECT</i>	07-03-1994	Upon btnAddNew _click this date will be accepted And lvCustomer will reload with updated data from database	

3.6.2	Text entered into txtCustDOB INCORRECT	seventh-March-1994	Upon btnAddNew_click this date will not be accepted and a message box will load stating “Please enter numbers only. Error found in txtCustDOB”	
3.6.3	Text entered into txtCustDOB EXTREME	07-03-1994 (max 8 char. reached)	Upon btnAddNew_click this date will not be accepted and a message box will appear stating “Maximum characters: 8. Error found in txtCustDOB”	
3.7.1	Text entered into txtAddLine1 CORRECT	“12 Penny Lane”	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database	
3.7.2	Text entered into txtAddLine1 INCORRECT	“12 Penny Lane?”	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters and numbers only. Error found in txtAddLine1”	
3.7.3	Text entered into txtAddLine1 EXTREME	“12 Penny Lane” (max 30 char. reached)	Upon btnAddNew_click this text will not be accepted and a	

			message box will appear stating “Maximum characters: 30. Error found in txtAddLine1”	
3.8.1	Text entered into txtAddLine2 CORRECT	“Apt 5”	Upon btnAddNew _click this text will be accepted And lvCustomer will reload with updated data from database	
3.8.2	Text entered into txtAddLine2 INCORRECT	“Apt-5”	Upon btnAddNew _click this text will not be accepted and a message box will load stating “Please enter letters and numbers only. Error found in txtAddLine2”	
3.8.3	Text entered into txtAddLine2 EXTREME	“Apt 5” (max 30 char. reached)	Upon btnAddNew _click this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtAddLine2”	
3.9.1	Text entered into txtTown CORRECT	“Armagh”	Upon btnAddNew _click this text will be accepted And lvCustomer will reload with	

			updated data from database	
3.9.2	Text entered into txtTown INCORRECT	“@rmagh4”	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters only. Error found in txtTown”	
3.9.3	Text entered into txtTown EXTREME	“Armagh” (max 30 char. reached)	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtTown”	
3.10.1	Text entered into txtPCode CORRECT	“BT61 8EQ”	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database	
3.10.2	Text entered into txtPCode INCORRECT	“BT61_8EQ”	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters and numbers only. Error found in txtPCode”	
3.10.3	Text entered into txtPCode EXTREME	“BT61 8EQ” (max 8 char. reached)	Upon btnAddNew_click this text will	

			not be accepted and a message box will appear stating “Maximum characters: 8. Error found in txtPCode”	
3.11.1	Text entered into txtContactNo CORRECT	“02837551083”	Upon btnAddNew _click this text will be accepted And lvCustomer will reload with updated data from database	
3.11.2	Text entered into txtContactNo INCORRECT	“a028-3755-1083”	Upon btnAddNew _click this text will not be accepted and a message box will load stating “Please enter numbers only. Error found in txtContactNo”	
3.11.3	Text entered into txtContactNo EXTREME	“02837551083” (max 11 char. reached)	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 11. Error found in txtContactNo”	
3.12	Text entered into txtEmailAdd CORRECT	“jane_doe@yahoo.com”	Upon btnAddNew _click this text will be accepted And lvCustomer will reload with	

			updated data from database	
	Text entered into txtEmailAdd INCORRECT	"jane_doe@yahoo.com"	Upon btnAddNew_click this text will not be accepted and a message box will load stating "Please ensure email address contains '@'. Error found in txtEmailAdd "	
	Text entered into txtEmailAdd EXTREME	"jane_doe@yahoo.com" (max 30 char. Reached)	Upon btnAddNew_click this text will not be accepted and a message box will appear stating "Maximum characters: 30. Error found in txtEmailAdd "	
3.13.1	Text entered into txtUName CORRECT	"jane94"	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database	
3.13.2	Text entered into txtUName INCORRECT	"jane_94"	Upon btnAddNew_click this text will not be accepted and a message box will load stating "Please enter letters and numbers only. Error found in txtUName"	

3.13.3	Text entered into txtUName EXTREME	“jane94” (max 30 char. reached)	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtUName”	
3.14.1	Text entered into txtPWord CORRECT	“Penny5?”	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database	
3.14.2	Text entered into txtPWord INCORRECT	“penny”	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter a combination of letters, numbers and other characters to strengthen your password. Error found in txtPWord”	
3.14.3	Text entered into txtPWord EXTREME	“Penny5?” (max 30 char. reached)	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30.	

			Error found in txtPWord"	
3.5	btnAddNew is clicked	-	Text entered into [all textboxes on form] will be validated (alphanumeric, numeric, presence check) Customer table of Events- Unlimited database will be updated lvCustomer will reload with updated content from database	
4 (Booking Form)	-	-	-	
4.1	frmBooking.cs will load		Booking form will load	
5 (Car Booking Form)	-	-	-	
5.1	frmCarBooking.cs will load		Car Booking form will load	
6 (Employee Form)	-	-	-	
6.1	frmEmployee.cs will load		Employee form will load	
7 (Supplier Form)	-	-	-	
7.1	frmSupplier.cs will load		Supplier form will load	
8 (Car Form)	-	-	-	
8.1	frmCar.cs will load		Car form will load	

9 (Help Form)	-	-	-	
9.1	frmHelp.cs will load		Help form will load	
9.2	tabCustomer is selected	-	tabCntrl tab page changes to “Customer” tab	
9.3	tabEmployee is selected	-	tabCntrl tab page changed to “Employee” tab	
9.4	tabSupplier is selected	-	tabCntrl tab page changed to “Supplier” tab	
10.1	frmReport.cs will load	-	Report form will load txtBookingDate will be populated with today's date (current date)	
10.2				

Acceptance Testing

Employee Acceptance Questionnaire:

How satisfied were you with the following? :

---	1 ☺	2	3 ☹	4	5 ☻	Comments
Overall functionality of the program?						
Overall aesthetic of the program?						
How the program was laid out?						
How the contents of the forms were displayed?						
The navigation of the system?						
The Customer form?						
The Employee form/facility?						
The Supplier form/ facility?						
The Car form?						
The Booking form/ facility?						
The CarBooking form/facility?						
The help form/facility?						
The login form/ facility?						
The nature of the program?						

Please Tick Yes or No

	Yes	No	Comments
Do you think this program is suitable for employees of all abilities?			
Do you think this program is suited to the nature of the addressed business problem?			
Would you recommend this program?			
Did you find this program helpful on a day-to-day basis?			
Do you think this program has all the functional elements a system of this nature should have?			

Manager Acceptance Questionnaire:

How satisfied were you with the following? :

---	1 ☺	2	3 ☹	4	5 ☻	Comments
<i>Overall functionality of the program?</i>						
<i>Overall aesthetic of the program?</i>						
<i>How the program was laid out?</i>						
<i>How the contents of the forms were displayed?</i>						
<i>The navigation of the system?</i>						
<i>The Customer form?</i>						
<i>The Employee form/facility?</i>						
<i>The Supplier form/ facility?</i>						
<i>The Car form?</i>						
<i>The Booking form/ facility?</i>						
<i>The CarBooking form/facility?</i>						
<i>The help form/facility?</i>						
<i>The login form/ facility?</i>						
<i>The nature of the program?</i>						

Please Tick Yes or No

	Yes	No	Comments
<i>Do you think this program is suitable for employees of all abilities?</i>			
<i>Do you think this program is suited to the nature of the addressed business problem?</i>			
<i>Would you recommend this program?</i>			
<i>Did you find this program helpful on a day-to-day basis?</i>			
<i>Did you find that this program benefitted your business?</i>			
<i>Have you received any feedback (positive or negative) about the updated system?</i>			
<i>Have you found that there are less problems following the implementation of this system? (e.g Customer records not being duplicated, booking details not being misplaced, etc.)</i>			
<i>Do you think this program has all the functional elements a system of this nature should have?</i>			
<i>Will you use this program in your workplace?</i>			

(1)

Employee Acceptance Questionnaire:

	How satisfied were you with the following?					Comments
	1 ☺	2 ☻	3 ☻	4 ☻	5 ☺	
Overall functionality of the program?		✓				there were very few issues.
Overall aesthetic of the program?	✓					Loved the colour scheme, simplicity worked well.
How the program was laid out?			✓			Navigation could be clearer.
How the contents of the forms were displayed?		✓				I think the employee and customer forms were too crowded.
The navigation of the system?				✓		Could have been more straightforward, e.g. if the booking forms were accessible from main menu
The Customer form?	✓	✓		✗		Would have been more accessible from main menu if crammed our otherwise fine.
The Employee form/facility?	✓					
The Supplier form/ facility?	✓					
The Car form?	✓					
The Booking form/ facility?		✓				
The CarBooking form/facility?		✓				
The help form/facility?	✓					
The login form/ facility?	✓					
The nature of the program?		✓				

Please Tick Yes or No

	Yes	No	Comments
Do you think this program is suitable for employees of all abilities?	✓		A simpler navigation would be easier to use.
Do you think this program is suited to the nature of the addressed business problem?	✓		
Would you recommend this program?	✓		
Did you find this program helpful on a day-to-day basis?	✓		The program is much easier to use than manually writing for each booking.
Do you think this program has all the functional elements a system of this nature should have?	✗	✓	I think the synchronised most elements but a profile facility could be helpful.

(2)

Employee Acceptance Questionnaire:

How satisfied were you with the following? :

	1 ☺	2	3 ☻	4	5 ☺	Comments
Overall functionality of the program?	✓					
Overall aesthetic of the program?		✓				I liked the layout and colour scheme but some of the theme was too harsh
How the program was laid out?		✓				I think the individual forms were good but Cust. + emp. were slightly messy
How the contents of the forms were displayed?	✓					↗
The navigation of the system?	✓					I thought the navigation was easy to follow, especially with the help facility
The Customer form?			✓			A bit too messy/crammed
The Employee form/facility?			✓			↗ "
The Supplier form/ facility?	✓					
The Car form?	✓					
The Booking form/ facility?		✓				
The CarBooking form/facility?		✓				
The help form/facility?	✓					
The login form/ facility?	✓					
The nature of the program?		✓				

Please Tick Yes or No

	Yes	No	Comments
Do you think this program is suitable for employees of all abilities?	✓		Along with the user guide, the program is easy to use
Do you think this program is suited to the nature of the addressed business problem?	✓		
Would you recommend this program?	✓	✗	The system is obviously tailored to Events Unihired but could be customised
Did you find this program helpful on a day-to-day basis?	✓		
Do you think this program has all the functional elements a system of this nature should have?	✓		

Testing

<u>Test No</u>	<u>Expected Outcome</u>	<u>Actual Outcome</u>	<u>Success? (Y/N)</u>	<u>Evidence</u>
1 (Login form)	-	-	-	-
1.1	Login form will load first	Login form loads	Y	See fig. 1.1
1.2.1	Upon btnLogin_click this text will be accepted frmMenu will load frmLogin will close	Text is accepted - frmMenu loads and frmLogin closes	Y	See fig.1.2.1
1.2.2	Upon btnLogin_click this text will not be accepted and a message box will load stating “Please enter numbers and letters only”			See fig. 1.2.2
1.2.3	Upon btnLogin_click this text will not be accepted and a message box will appear stating “Maximum characters: 30”			See fig 1.2.3
1.3.1	Upon btnLogin_click this text will be accepted frmMenu will load frmLogin will close	Text is accepted – frmMenu loads and frmLogin closes	Y	See fig. 1.2.1
1.3.2	Upon btnLogin_click this text will not be accepted and a message box will load stating “Please enter numbers and letters only”			See fig. 1.2.2

1.3.3	Upon btnLogin_click this text will not be accepted and a message box will appear stating "Maximum characters: 30"			See fig. 1.2.3
1.4	Text entered into txtUName and txtPWord will be validated (alphanumeric exception) Employee table of Evets-Unlimited database will be queried in order to verify that both UName and PWord are both valid			See fig. 1.2.1
2 (Menu Form)	-			
2.1	Menu form will load	Menu form loads	Y	See fig. 2.1
2.2	Message box will show stating "Are you sure you wish to exit?" (message box buttons: 'Exit', 'Cancel') If yes: frmMenu.cs will close and frmLogin.cs will open If no: message box will close	Message box shows – confirmation for exit required	Y	See fig. 2.2
2.3	frmMenu.cs will close, frmHelp.cs will load	frmMenu closes	Y	See fig.2.3
2.4	frmMenu.cs will close, frmCustomer.cs will load	frmCustomer loads and frmMenu closes	Y	See fig. 2.4

2.5	frmMenu.cs will close, frmEmployee.cs will load	frmEmployee loads and frmMenu closes	Y	See fig. 2.5
2.6	frmMenu.cs will close, frmSupplier will load	frmSupplier loads and frmMenu closes	Y	See fig.2.6
3 (Customer Form)	<i>As my Customer, Employee and Supplier forms are duplicates of one another with only minor differences, I have decided to record the Test Plan for my Customer form only – along with any differences that need further testing.</i>	-	-	-
3.1	Customer form will load Customer details will be loaded into lvCustomer from database customerID will increment in txtCustID	Customer form loads	Y	See fig. 3.1
3.2	Selected customer's details will load into associated textboxes (e.g. custFName will be inserted into txtCustFName)	Selected customer's details load into correct textboxes	Y	See fig. 3.2
3.3.1	Upon btnAddNew _click this text will be accepted frmMenu will load frmLogin will close			See fig. 3.3.1
3.3.2	Upon btnAddNew _click this text will not be accepted			See fig. 3.3.2

	and a message box will load stating “Please enter letters only. Error found in txtCustFName”			
3.3.3	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtCustFName”			See fig. 3.3.3
3.4.1	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database			See fig. 3.4.1
3.4.2	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters only. Error found in txtCustSName”			See fig. 3.3.2
3.4.3	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtCustSName”			See fig. 3.3.3
3.5.1	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database			See fig. 3.5.1

3.5.2	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters only. Error found in txtCustTitle”			See fig. 3.3.2
3.5.3	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 5. Error found in txtCustTitle”			See fig. 3.3.3
3.6.1	Upon btnAddNew_click this date will be accepted And lvCustomer will reload with updated data from database			See fig. 3.6.1
3.6.2	Upon btnAddNew_click this date will not be accepted and a message box will load stating “Please enter numbers only. Error found in txtCustDOB”			See fig. 3.3.2
3.6.3	Upon btnAddNew_click this date will not be accepted and a message box will appear stating “Maximum characters: 8. Error found in txtCustDOB”			See fig. 3.3.3
3.7.1	Upon btnAddNew_click this text will be accepted			See fig. 3.7.1

	And lvCustomer will reload with updated data from database			
3.7.2	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters and numbers only. Error found in txtAddLine1”			See fig. 3.3.2
3.7.3	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtAddLine1”			See fig. 3.3.3
3.8.1	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database			See fig. 3.8.1
3.8.2	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters and numbers only. Error found in txtAddLine2”			See fig. 3.3.2
3.8.3	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30.			See fig. 3.3.3

	Error found in txtAddLine2”			
3.9.1	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database			See fig. 3.9.1
3.9.2	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters only. Error found in txtTown”			See fig. 3.3.2
3.9.3	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtTown”			See fig. 3.3.3
3.10.1	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database			See fig. 3.10.1
3.10.2	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter letters and numbers only. Error found in txtPCode”			See fig. 3.3.2
3.10.3	Upon btnAddNew_click this text will not be accepted and a message box			See fig. 3.3.3

	will appear stating “Maximum characters: 8. Error found in txtPCode”			
3.11.1	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database			See fig. 3.11.1
3.11.2	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please enter numbers only. Error found in txtContactNo”			See fig. 3.3.2
3.11.3	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 11. Error found in txtContactNo”			See fig. 3.3.3
3.12.1	Upon btnAddNew_click this text will be accepted And lvCustomer will reload with updated data from database			See fig. 3.12.1
3.12.2	Upon btnAddNew_click this text will not be accepted and a message box will load stating “Please ensure email address contains '@'. Error found in txtEmailAdd ”			See fig. 3.3.2
3.12.3	Upon btnAddNew_click			See fig. 3.3.3

	this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtEmailAdd ”			
3.13.1	Upon btnAddNew _click this text will be accepted And lvCustomer will reload with updated data from database			See fig. 3.13.1
3.13.2	Upon btnAddNew _click this text will not be accepted and a message box will load stating “Please enter letters and numbers only. Error found in txtUName”			See fig. 3.3.2
3.13.3	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtUName”			See fig. 3.3.3
3.14.1	Upon btnAddNew _click this text will be accepted And lvCustomer will reload with updated data from database			See fig. 3.14.1
3.14.2	Upon btnAddNew _click this text will not be accepted and a message box will load stating “Please enter a combination of			See fig. 3.3.2

	letters, numbers and other characters to strengthen your password. Error found in txtPWord”			
3.14.3	Upon btnAddNew_click this text will not be accepted and a message box will appear stating “Maximum characters: 30. Error found in txtPWord”			See fig. 3.3.3
3.15	Text entered into [all textboxes on form] will be validated (alphanumeric, numeric, presence check) Customer table of Events-Unlimited database will be updated lvCustomer will reload with updated content from database			
4 (Booking Form)	-	-	-	-
4.1	Booking form will load	Booking form loads	Y	See fig. 4.1
5 (Car Booking Form)	-	-	-	-
5.1	CarBooking form will load	CarBooking form loads	Y	See fig. 5.1
6 (Employee Form)	-	-	-	-
6.1	Employee form will load	Employee form loads	Y	See fig. 6.1

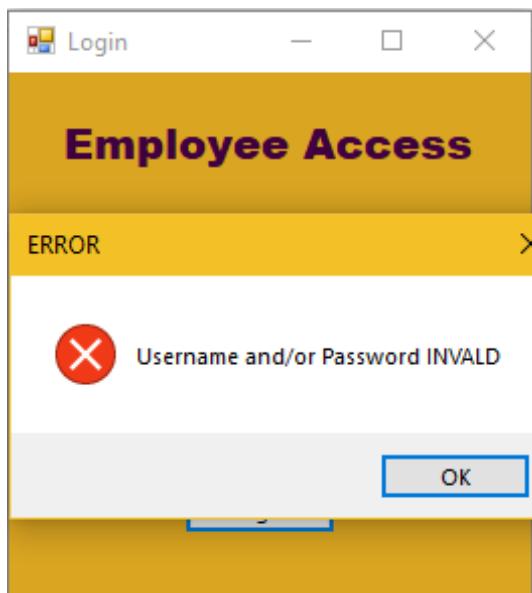
7 (Supplier Form)	-	-	-	-
7.1	Supplier form will load	Supplier form loads	Y	See fig 7.1
8 (Car Form)	-	-	-	-
8.1	Car form will load	Car form loads	Y	See fig. 8.1
9 (Help Form)	-	-	-	-
9.1	Help form will load	Help form loads	Y	See fig. 9.1
9.2	tabCntrl page will change to “Customer” tab	tabCntrl tab page changes to “Customer” tab	Y	See fig. 9.2
9.3	tabCntrl page will change to “Employee” tab	tabCntrl tab page changed to “Employee” tab	Y	See fig. 9.3
9.4	tabCntrl page will change to “Supplier” tab	tabCntrl tab page changed to “Supplier” tab	Y	See fig. 9.4
10	-	-	-	-
10.1				See fig. 10.1

Corrective Action

Due to time constraints and poor time management, I was unable to complete my project, particularly to the extent of which I initially assessed my scope. While I would ideally like to add much more functionality to my program, at this stage of development it was more realistic to make more minor changes to increase user acceptance and lower level functionality. Such minor changes included the implementation of text box validation where I had initially forgone it due to timing issues.

For the Login form, I implemented the validation of both the Username and Password textboxes separately, I felt that this was better functionality to use than so simply have a textbox which displays that either username OR password is incorrect.

Before:

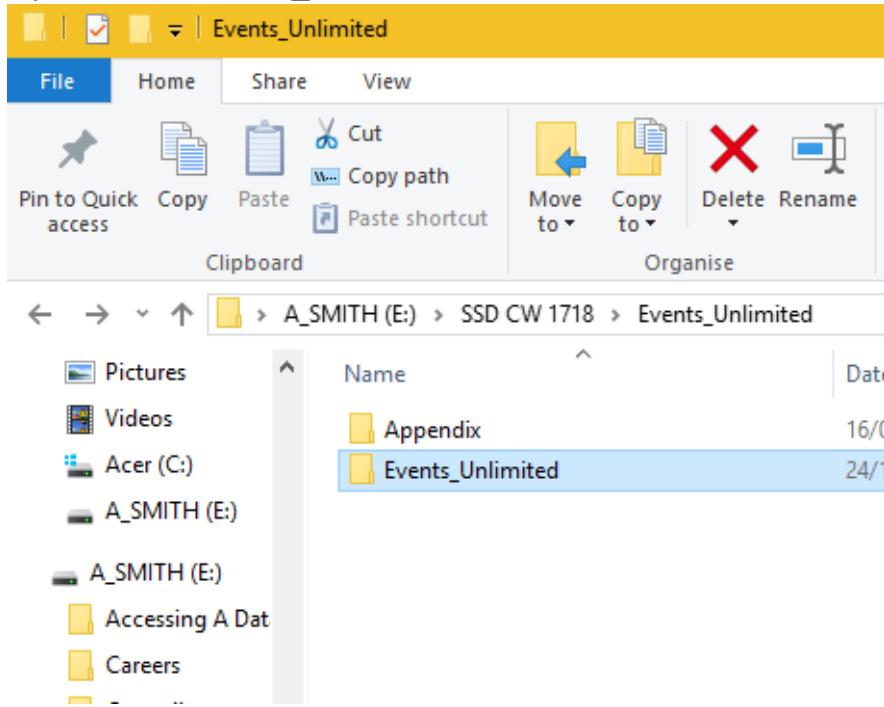


After:

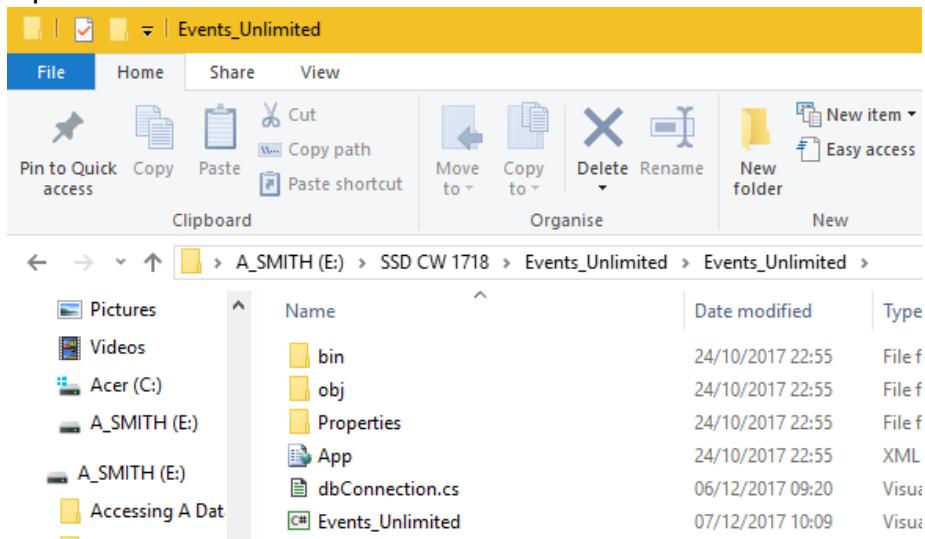
User Guide

Installation Guide

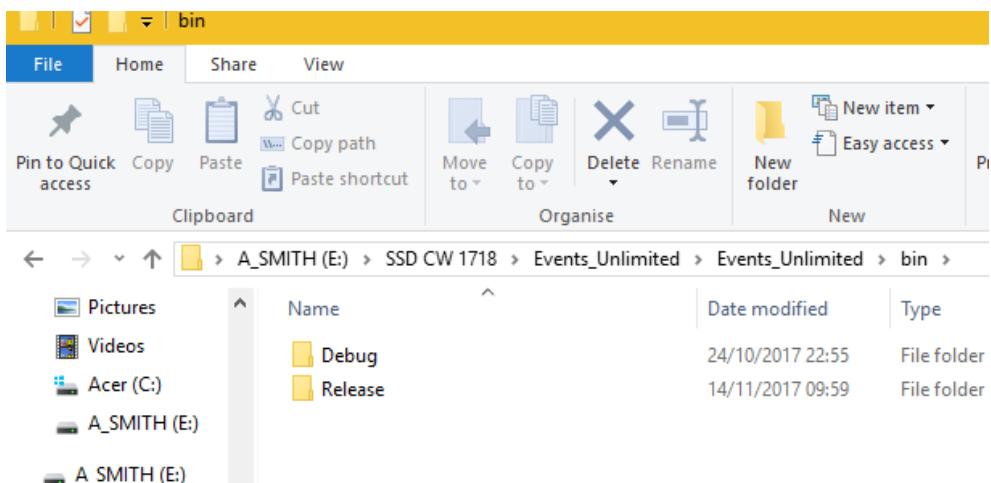
- To install this program onto your PC / laptop, insert installation disc into disc drive.
- Open folder “Events_Unlimited”



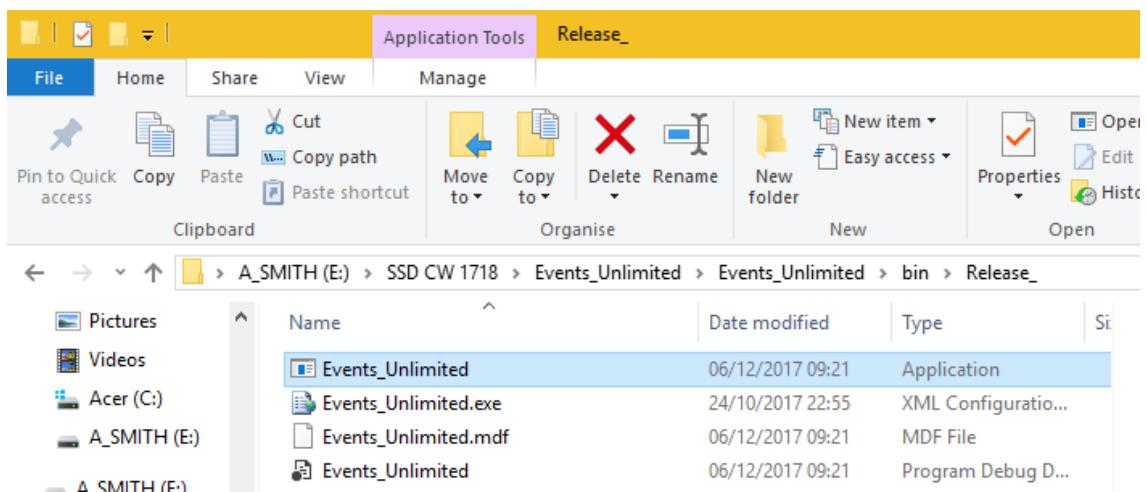
- Open folder “Bin”



- Open folder “Release”



- Copy file “Events-Unlimited” of file type Application into your chosen location by right-clicking and choosing either Copy or Send To



Hardware and Software Required

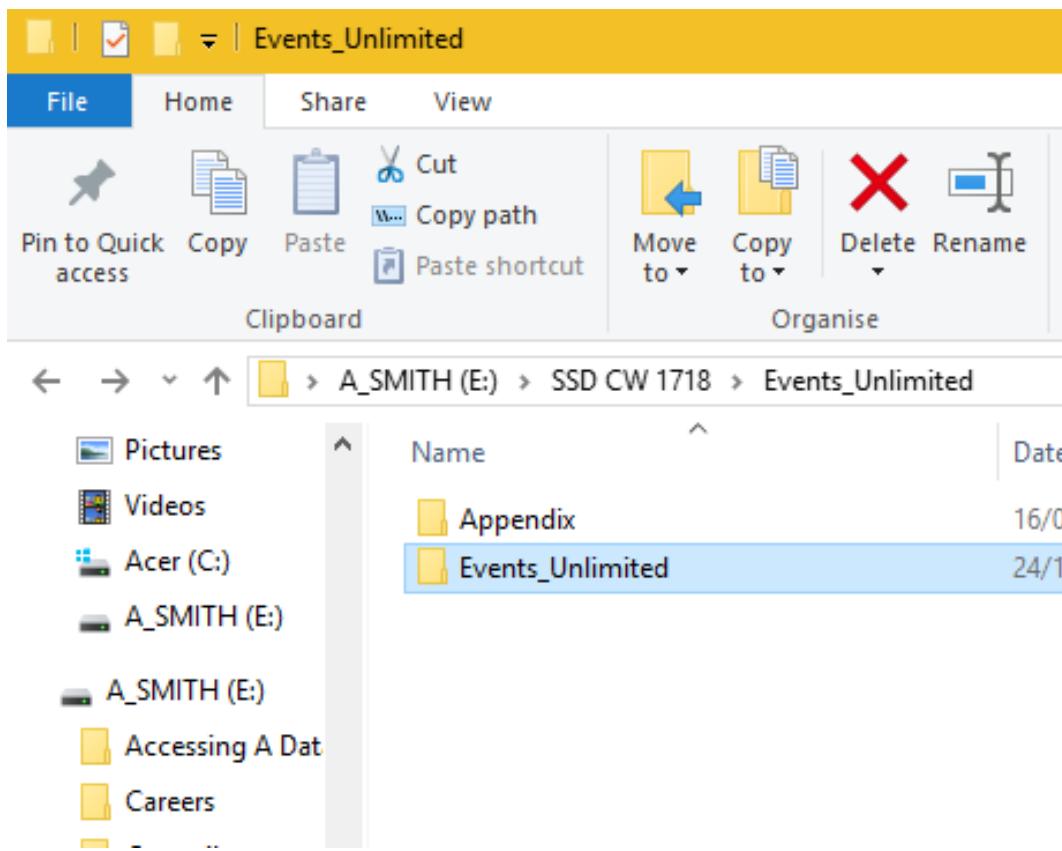
HARDWARE REQUIRED	
CPU: <ul style="list-style-type: none">• Intel Core i7-2600 with 3.40 GHz	Monitor: <ul style="list-style-type: none">• 17" LCD (minimum)• 19" LCD (recommended)
Memory: <ul style="list-style-type: none">• 8 GB (minimum)• 16GB (recommended)	Graphics Card: <ul style="list-style-type: none">• NVIDIA GeForce 9200
Hard Disk: 100GB (minimum) 500GB (recommended)	

SOFTWARE REQUIRED	
Operating System: <ul style="list-style-type: none">• Windows 7/8/10 (64-bit OS required)	Antivirus: <ul style="list-style-type: none">• Norton Security 2016 (recommended)
Other: <ul style="list-style-type: none">• Microsoft Office Suite (2010-2016)	

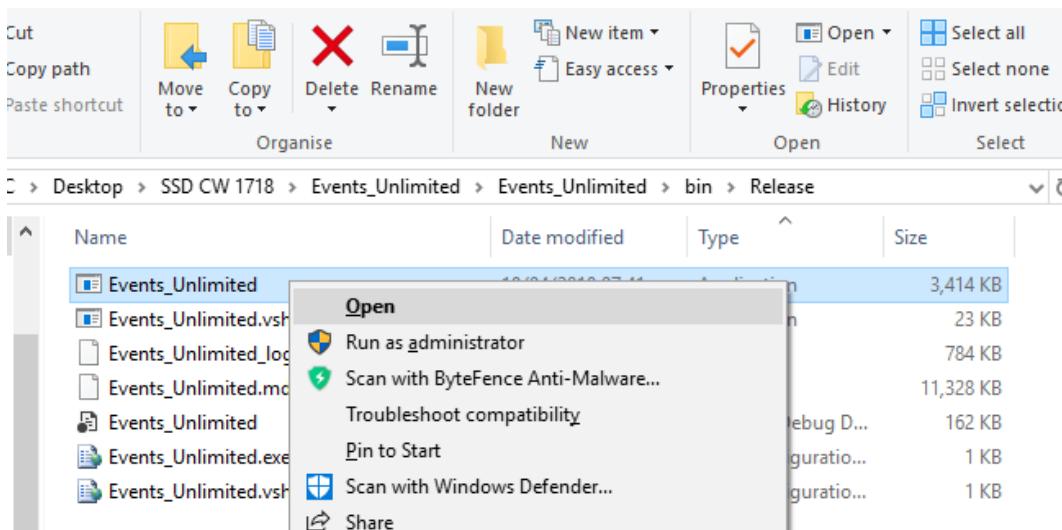
Operating Instructions

*** basic operating instructions can be found in the HELP form within the system ***

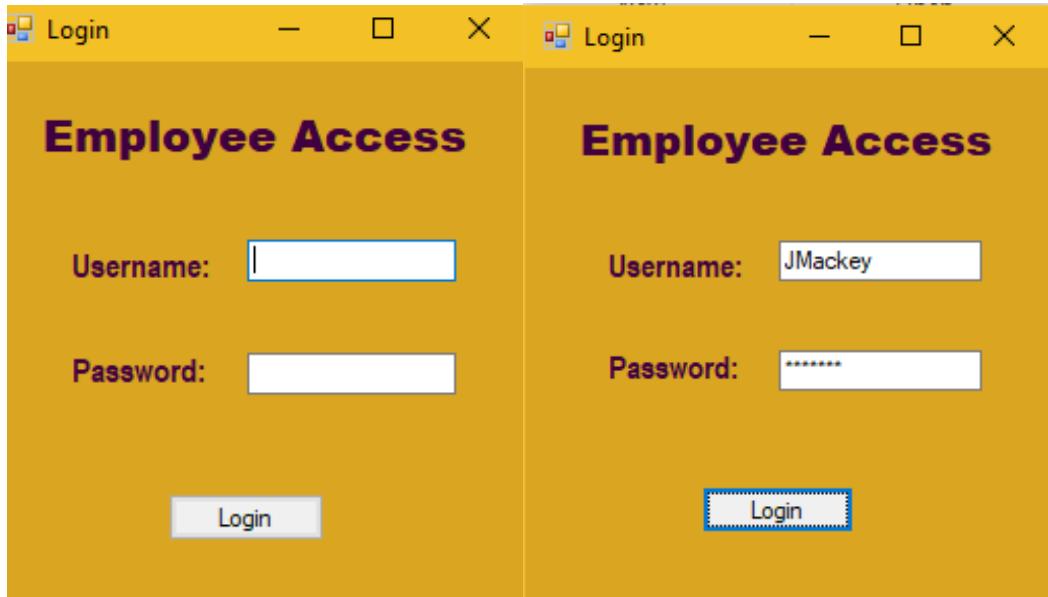
- To run the program, open folder Events Unlimited



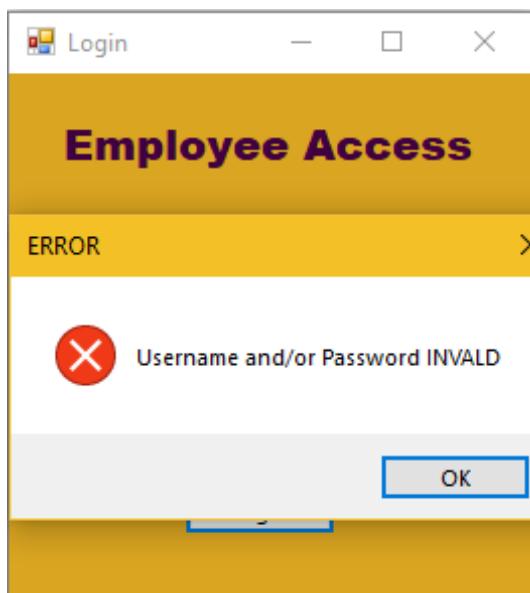
- Right-click Events Unlimited of file type Application and choose the 'Open' function



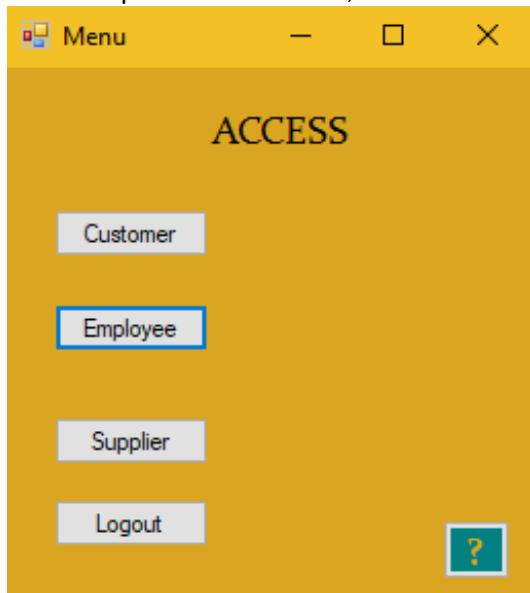
- The system should then load a login screen, using your employee credentials (e.g. username: jdoe, password: unlimited1997), log in to the system by clicking the 'Login' button



- If the credentials are not entered or are entered incorrectly an error message will show. If you see this message, the user is required to enter their credentials again. If the credentials are not accepted further, seek technical support.

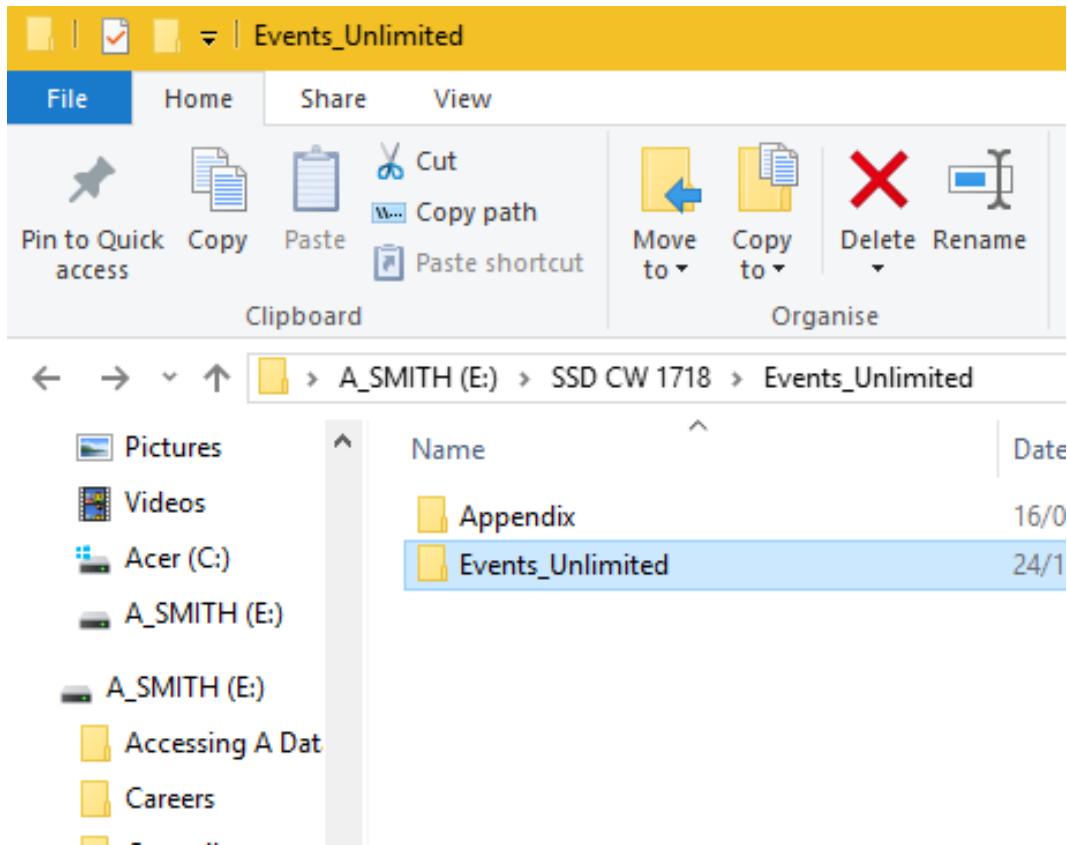


- When login details are accepted, the system will then load a menu form, choose one of the five options on the form; from customer, employee, supplier, logout, and help

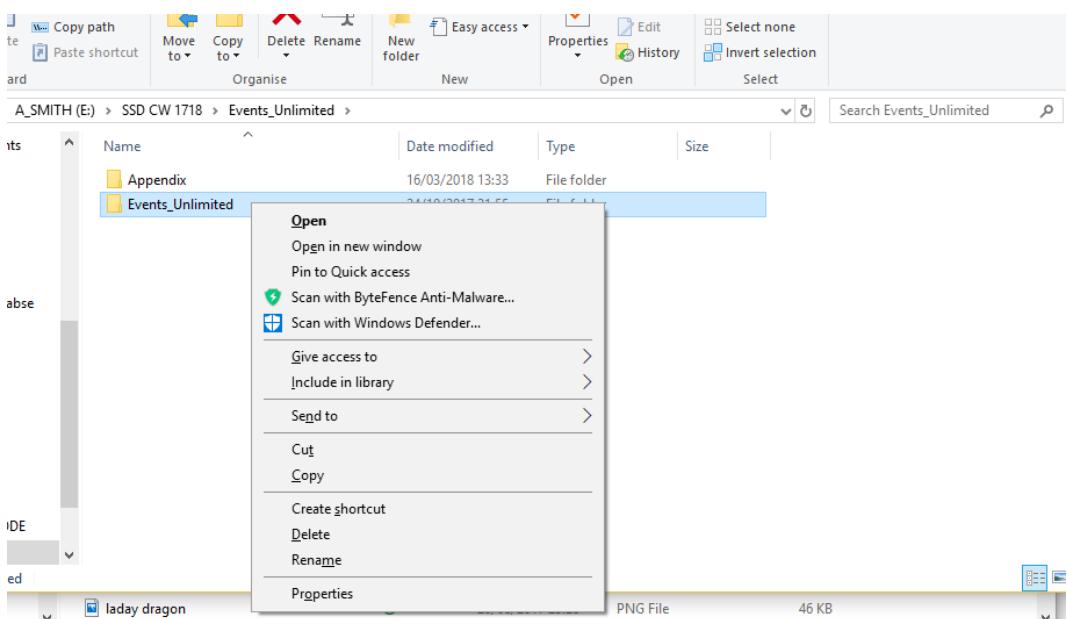


Backup Procedures

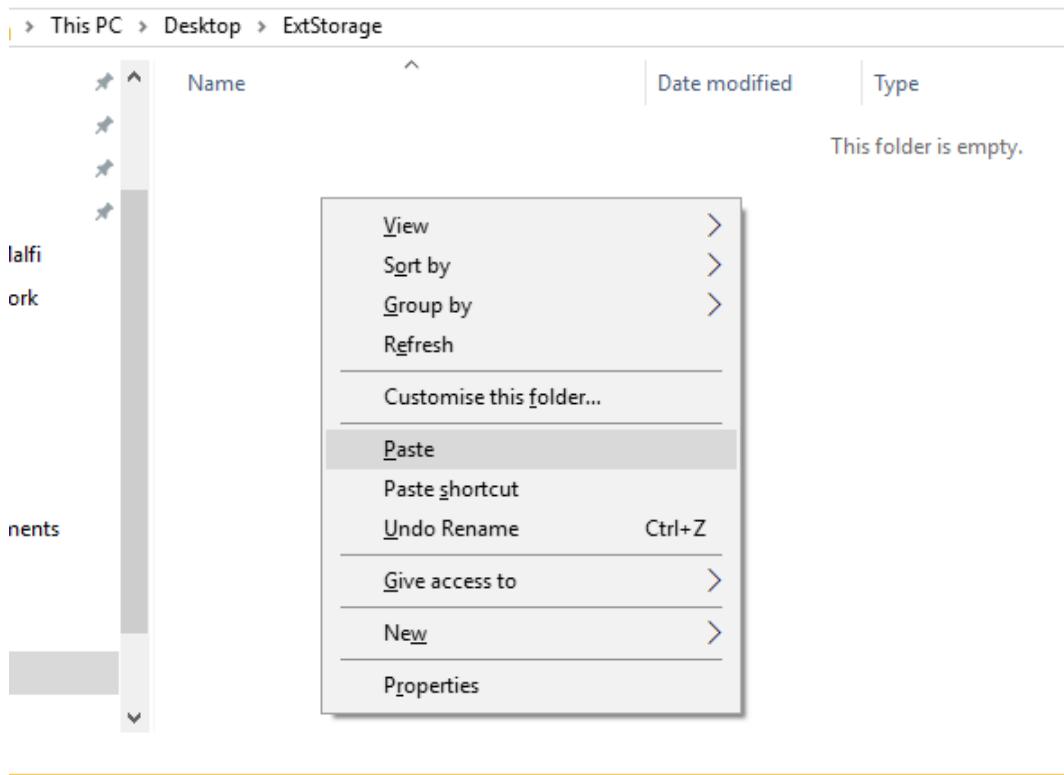
- To backup your system, open folder containing Events Unlimited folder



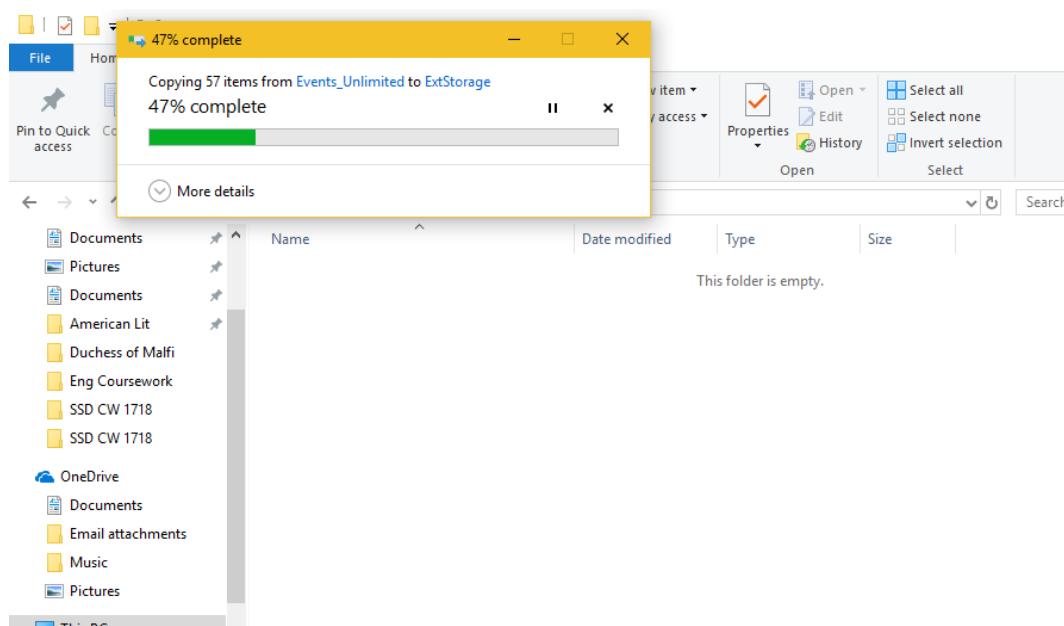
- Right-click the Events Unlimited folder and choose the 'copy' function



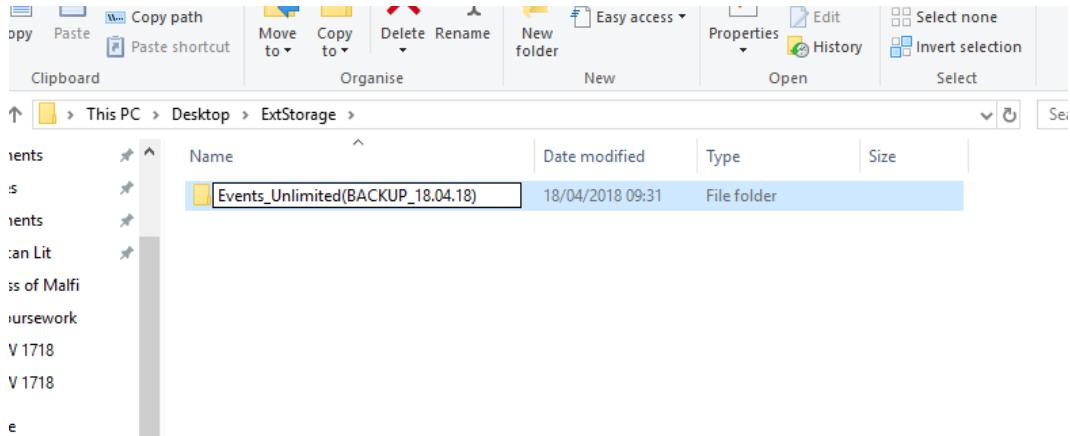
- Copy this file into a secure location where the system is not already stored, e.g. an external hard-drive – do this by right-clicking your chosen location and clicking the ‘paste’ function



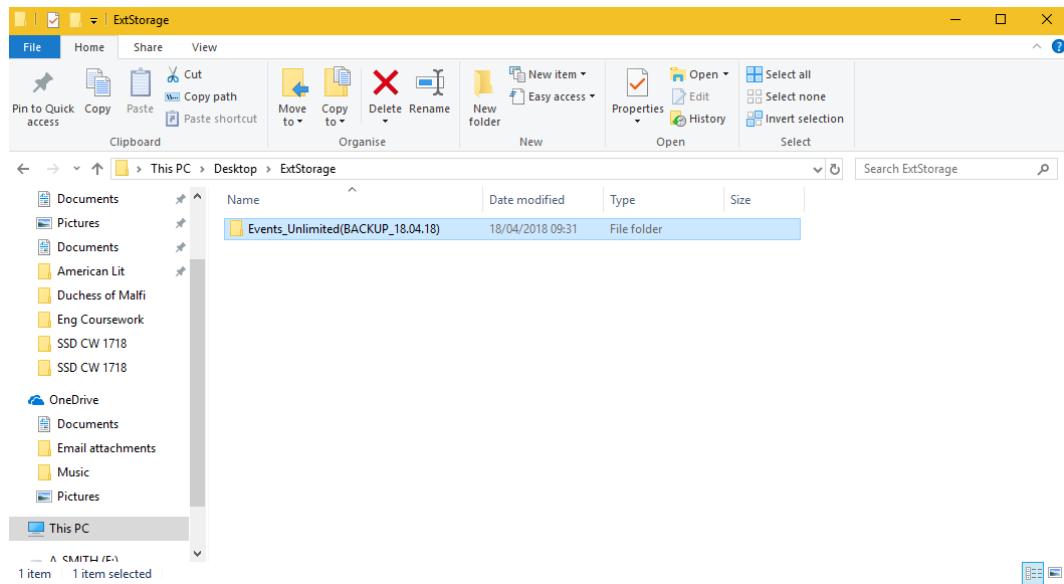
- Once the file has copied into its new location, open it and run the system to ensure the backup was successful



- If the backup was a success, it is good practice to rename the file to [filename]_backup, and possibly a date or edition, this will be useful in the future if anything happens to the program and needs to be amended at a certain point or restored to a particular edition.



- If the backup was, for any reason, unsuccessful, retry the steps previously – if still failing to back up, seek further technical assistance.



Evaluation

Evaluation of User Requirements

During the project I tried to meet all of the user requirements set out at the beginning of the task, including both functional and non-functional aspects. The main aim which I tried to meet and believe I have is to create a program which will help Doreen manage her business and to facilitate bookings – I had initially intended on creating a system which allowed both customers and employees to make bookings but due to the time constraint I thought it better to restrict the system to only allowing for bookings to be made by employees.

Another important requirement was to create a facility which allowed Doreen and her employees to create, edit and delete records for customers, employees and suppliers, as well as create, edit and delete records for the products (cars) themselves. I feel that I have succeeded in this requirement as employees are able to manage database records with ease as the process is straightforward.

Following the beginning of my development, I decided that a login feature for employees would be beneficial as it would mean that employees would not have to enter their employeeID repeatedly to reference bookings. I was also determined to create this feature as I intended to implement it in a previous project but couldn't due to time constraints. I was satisfied with this form overall as it functioned correctly and added functionality to the system. As I previously mentioned, I forewent a customer login facility due to time constraints, despite missing this feature I still think that it will benefit Events Unlimited by speeding up the booking process and making bookings easier for employees and therefore portraying a more professional reputation to customers.

I created a simple help form in the system also, this will be helpful for employees who need a reminder of how to use the system, although I have also created a more in-depth user guide which iterates through each step of running the system. The help form is extremely basic, explaining simply how to make bookings and manage records - therefore I was satisfied with the turnout.

Another key functional requirement was to ensure that Doreen will be able to generate reports of bookings to reference and to give to customers. I am satisfied with the turnout of this facility as it works well in correspondence to the system and can generate a basic report, along with calculations for totals and discounts.

In terms of non-functional requirements, I feel that I have succeeded in designing and developing a system which is aesthetically professional and attractive, although I think that I could have reorganised the customer, and employee forms so that they would have appeared less cramped and messy. I also think that the overall program is very user friendly and accessible; however, I recognise that my navigation system could have used some improvement as the path to making a booking was overcomplicated.

In conclusion I think that I have successfully addressed the majority of the user requirements established at the beginning of the project. Despite some complications throughout the development I believe that my effort has been reflected in my work and has been appreciated by the business, as can be seen in the Acceptance Testing section of my documentation.

Evaluation of Aims

- By the end of this project I do believe that I have created a program which will help Doreen to manage her business by facilitating a way for employees at Events Unlimited to easily make bookings for customers. I feel that I could have developed other elements to the system which would have aided her business further, such as a way to make a booking with a supplier, or also be able to log in as a customer and request a booking.
- I have also created a facility for Doreen to amend customer, employee, supplier and car records. This is a practical amendment for the business as it means that information for customers, employees, suppliers, and cars has to only be stored once, reducing data duplication, and therefore data redundancy.
- In addition to being able to amend records through the system, it is also possible to add and delete records, this will allow employees to quickly add customers who wish to make a booking. It also means that adding new cars is very simple.

I also included a login facility which will allow employees so easily take charge and link themselves to bookings.

- Implement an employee login facility so that bookings can be automatically linked to an employee
- Implement a customer login facility so that customers can login and review their booking details
- Create a help form so that new employees can check how to create new bookings etc.
- Have a menu form which allows the employee to choose from customer, employee, and supplier

Non-Functional

- Ensure the overall aesthetic of the program is professional and attractive
- Make the program user friendly and accessible
- Implement simple navigation between forms and tables

Evaluation of Development Tools

In my project I used two different methodologies to create my project including RAD (Rapid Application Development) and the Waterfall Methodology. I feel that RAD was a good methodology for me to implement as its aim is to develop functional software in a short time period. The flexibility of the RAD model meant that I was able to change requirements and aims throughout the project, for example implementing a Login feature - I think this aspect of the methodology was beneficial to me as I like to constantly review my plan and my work, even altering it at certain points, I also found that it was preferable to the Waterfall model as RAD is an iterative process and less sequential. I think the RAD methodology also helped my project by allowing to identify errors and fix them at an early stage in the development.

Although I found RAD to be very beneficial overall, I recognise that there are some disadvantages to using this method, I found that there is less control over the development due to the flexibility to change scope and requirements during development. The main reason I chose this methodology was due to the rigid time constraint on developing my system, therefore this development tool was suited to a form driven system like mine.

For my project I also used the Waterfall methodology, I believe that the implementation of this development tool helped me to stay on track and within rough time-constraints - even though I still struggled to manage and optimise my time. I found that this model was very easy to understand as there is a clear understanding of what should happen at each phase. Another aspect which I think benefitted my project was the rigidity of the model, of phases being completed sequentially; in collaboration with the RAD model, which allowed me to make changes to earlier phases at a later stage, the management portion of my project was made easier. Similarly to RAD, the Waterfall model is also suited to short-term development projects such as mine.

Comparable to the RAD model, I acknowledge that there are also downfalls of this method, the main one being the lack of user involvement during the development, I think that it is important for the user to be involved in the project's development, this would mean that a project is less likely to be rejected upon completion and it would also mean that the system would likely have more necessary functionality along with more non-functional requirements set out by the user. Despite the negative aspects of this methodology, I retain that it was one of the best development tools I could have implemented for my project. In conclusion I think that the combination of the RAD methodology and the Waterfall method were a good combination to use for my project and ultimately aided my project by reducing the amount of risk and uncertainty from the task overall.

Evaluation of Project Management

In evaluating my Gantt chart, dairy and PERT chart, I can conclude that I did get my project completed on time, however I still have some of improvement to make on my time management skills as there were times when I was running tasks on extra time or having to delay the start of a task in aid of finishing another. I feel that my Gantt chart helped me to manage my time as I had a structured development plan and I was aware that certain tasks had to be completed on time. My PERT chart was also useful in that it reminded me that each section was dependent on another, even if only partially of certain tasks, due to this I tried to keep each section within a set time, even if specific tasks were delayed. In the majority of cases where I found myself running behind, it was due to either setting myself unrealistic deadline – for example when I estimated that all of my form designs would take 3 weeks, this was unrealistic as my designs were constantly being reviewed and updated to ensure they worked well and that the system looked professional. On the weeks I intended to get more done than was possible, however I attempted to rectify this by spending less time than I had dedicated to another task, I discussed these issues in my project diary and they are also visible on my Gantt chart. Another aspect which delayed me was the actually programming section of the project, I often found myself changing aspects which I had already developed and running into errors.

Ultimately, I think my project managements skills have improved greatly in comparison to my project from last year and while I know that I made mistakes, the next time I plan a project I will be aware of rigid deadlines and the importance of being harsh and realistic when setting goals for development.

Evaluation of Personal Performance

Throughout the duration of this project, many of my personal skills have been tested and improved, these skills included problem-solving, time management, communication and project management, and programming. I feel that these skills have all been honed while developing this project. My problem-solving skills were initially tested when I received the case study of Events Unlimited, on reception I had to analyse the business to determine the most important issues and from there, I had to decide ways in each problem, could be solved, finally choosing which issue I would tackle in order to help the business. Problem-solving is also a very important part of programming as each time my program encountered an error I had to work out what aspects weren't working correctly and what measures I could take in order to fix the problem. This also helped to improve my programming skills greatly because not only did it teach me how to fix the problem but it also helped my understanding of *why* my program has encountered certain errors, enhancing my knowledge of the code. Although this experience has helped my problem-solving and programming skills immensely, I know that I still have a lot of improvement to make and will try to improve more in future developments.

Time management was also a key part of this project, it is also a personal weakness which I had to test rigorously in order to meet deadlines set by myself and my tutor. I put a lot of effort into improving my time management and organisational skills throughout the duration of this coursework, at times I was disappointed that I wasn't meeting my own deadlines however I am satisfied that my overall deadlines were met and there has been an improvement in my project management skills. The use of a Gantt chart, and PERT chart helped me to stay on track even though I still struggled with timing, I was able to complete important tasks on time by prioritising dependant tasks.

Throughout the project, I also had to work on my communication skills, I found this difficult as it involved listening to criticisms from both my tutor and classmates in order to improve my system. I dedicated a significant amount of time to designing and developing my project and found it tough to take people's criticisms and use it constructively rather than defend my own shortcomings. I tried to listen to my tutor's and peers' suggestions for improvements in order to get an outside opinion to eliminate bias, I understand that this is a key aspect when working for a business, especially working as part of a team and is therefore necessary for optimal development and will continue to work on my interpersonal skills for the future.

When programming, commenting code is an important step in understanding exactly what you've written and even improve your efficiency. I often forgot this while programming which meant that I had to return to comment my code while completing other tasks, I found this tedious and this resulted in me prioritising tasks. I regret doing this as it left me with patches of commenting that ended up not being very helpful. When I finished my project, I attempted to fill in the gaps but found this difficult as I had since moved on and forgotten why I had done certain things. In future projects I will *definitely* comment my code as I now fully understand the importance of doing it as well as the impact of neglecting it.

Through my constant self-assessment, I was able to improve on my self-evaluation skills, this helped me to understand where I should place focus on in effort to improve for future

developments, my time management skills being a focal point. If I had have made more effective use of my time I could have included more functionality in my system, which I will discuss in my Further Developments section.

In evaluation of my project as a whole, I can say that I worked to the best of my ability, despite the pitfalls I am content that I have developed a project which I have analysed, designed, and tested – a project which is now a functioning system. When I began this project I had limited programming skills, in that I was still learning SQL and to say now that I can confidently design and develop a booking is an important personal achievement. Throughout this project I put in significant effort and I am overall satisfied with the outcome, even though I know I have much room for improvement. I hope to continue further developing my skills through personal and academic projects in the future.

Future Developments

Due to time constraints and poor time management, I was unable to complete my project, particularly to the extent of which I initially assessed my scope. If I had more time I would like to implement a facility for which customers could also log into the system to request booking for themselves, such a function could be web-based for ease of use.

As I outlined at the beginning of my documentation in the analysis section, Events Unlimited have various areas of business concern which I would also ideally like to address in future developments.

Appendix

Screenshot Evidence

